ANALYTICAL STUDY OF LAUNCH VEHICLE COMPONENT LEVEL SIMULATION

GPO PRICE $ ________

DECEMBER 1965

CFSTI PRICE(S) $ ________

Hard copy (HC) 2.00

Microfiche (MF) 2.00

ff 653 July 65

Prepared under Contract No.

NAS 8-20060, DCN 1-5-60-00036-01 by

APOLLO SUPPORT DEPARTMENT
MISSILE AND SPACE DIVISION
GENERAL ELECTRIC
DAYTONA BEACH, FLORIDA

FACILITY FORM 602

N66-23834
(ACCESSION NUMBER)

(THRU)

389
(PAGES)

1
(CODE)

CR-71796
(NASA CR OR TMX OR AD NUMBER)

11
(CATEGORY)

Marshall Space Flight Center

National Aeronautics and Space Administration

FOREWORD

This report was prepared by the Apollo Support Department, General Electric Company, Daytona Beach, Florida, under NASA Contract NAS8-20060, control number DCN 1-5-60-00036-01 (IF). The work reported was performed during the period 26 March 1965 through 26 November 1965. Work performed after 26 November 1965 under a supplemental agreement will be reported upon completion on 31 March 1966. This work was under the guidance of Mr. J.R. Mitchell of MSFC Quality and Reliability Assurance Laboratory.

Principal technical contributors were Messrs. R.O. Harper, D.G. Chappell, W.P. Crownover, E.D. Fisher, E.A. Houser, J.W. Johnson, T.J. Little, E.W. Reinhardt, A.L. Ruiz, H.A. Torchia, and C.A. Tsonis of the General Electric Company, Apollo Support Department and Professor T.R. Hoffman of Union College, Schenectady, New York.

This report has been approved for release to qualified requestors through the NASA Scientific and Technical Aerospace Reports system.

ABSTRACT

This study was initiated by MSFC in order to establish a sound technical basis for the implementation of a digital computer simulation of the Saturn launch vehicle and its support equipment at a level compatible with effective utilization of the simulàtion outputs. Analysis has shown that a complete discrete and dynamic simulation is both practical and desirable. Recommendations are made to complete the design and implementation of this digital simulation.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONT.)

LIST OF TABLES

LIST OF TABLES (CONT.)

SECTION 1

INTRODUCTION

This Launch Vehicle Component Level Simulation (LVCLS) study was initiated by MSFC to establish a sound technical basis for the implementation of a digital computer simulation of the Saturn launch vehicle and its support equipment at a level compatible with effective engineering utilization of the simulation outputs. The study encompassed analyses of the present data organization and simulation methods, the available and anticipated computer hardware, and the potential utilizations in order to select an optimum combination of equipment and methods. The intent of the study was to insure meaningful results in time for utilization, and to point out and develop problem-solving methods that will continue to be efficient and useful for systems that follow. In order for the study to be practical, it was necessary and advisable to give careful consideration to the ultimate implementation concept and to identify and explore its technical problem areas. It is felt that the objectives enumerated in Section 2 have been met successfully.

Implementation of this simulation will provide a complete, integrated, dynamic computer simulation of the launch vehicle and its support equipment that will aid in:

- Equipment design.
- Verification of equipment functional relationships.
- Reliability and safety evaluations of equipment and procedures.
- Test and evaluation of the functional operation of equipment.
- Generation and optimization of checkout operations and procedures.
- Validation of manufacturing tests.
- Configuration management, identification, accounting, and control.
- Growth and development of advanced systems.

SECTION 2

ANALYTICAL STUDY OBJECTIVES

The objectives of the LVCLS study were to identify and solve problems dealing with the general nature of the simulation - its formulation, operation, maintenance, and effectiveness. The approaches and solutions yielded by this study are essential for implementing the simulation.

The intention was not to develop a complete simulation program, rather selected portions were programed and actual computer solutions were obtained. Only in this way was it possible to test the feasibility of the new approaches to problems and assess their salient features and accuracy.

The aim of this study has been to provide MSFC with specific recommendations covering the following areas:

a. Initial computer hardware requirements.
b. Computer hardware required for future needs.
c. Compatibility techniques.
d. Individual programs to be developed.
e. Order of implementation of individual programs.
f. Overall simulation requirements.
g. Data requirements for outputs from the simulation.
h. Data requirements for inputs to the simulation.
i. Procedures to insure that the simulation continues to receive current data and contributes to configuration management decisions.

In order to reach these objectives, the study was divided into interrelated tasks and organized sequentially as shown on Figure 2-1. Section 4, "Simulation Description," gives a comprehensive discussion of these tasks as well as the basis for the study recommendations defined in Section 3, "Summary of Results and Recommendations."

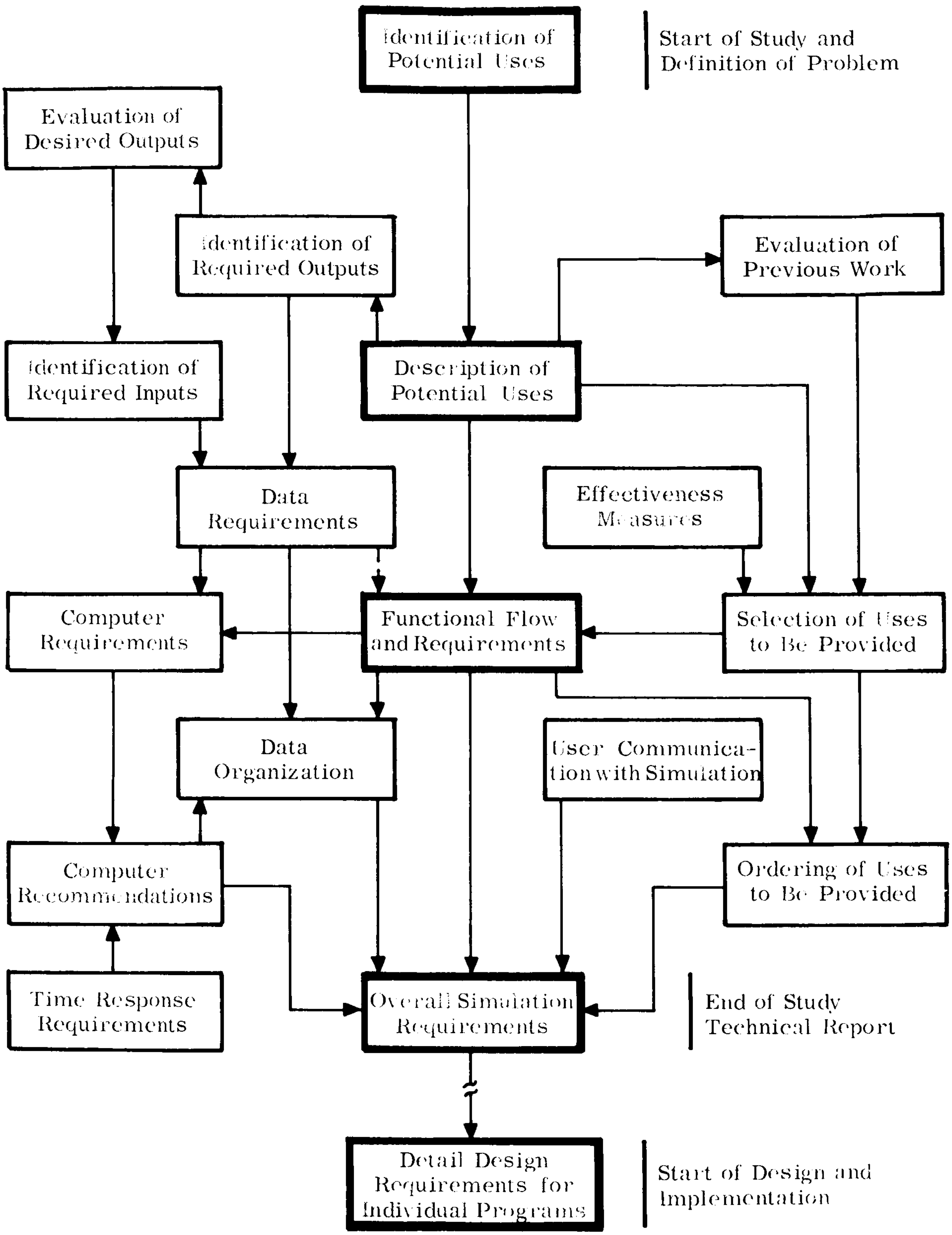


Figure 2-1. Task Flow Diagram for Analytical Study of Launch Vehicle Component Level Simulation

SECTION 3

SUMMARY OF RESULTS AND RECOMMENDATIONS

3.1 GENERAL

This study has shown that launch vehicle simulation at the component level is practical and desirable. Usually, when considering the problem of system simulation, the configuration is modeled in terms of a well-defined set of equations describing the system response to prescribed stimuli. The problem is then one of solving these equations. The Launch Vehicle Component Level Simulation (LVCLS) goes another step. Here, it is desired to be able to automatically build up both the discrete and dynamic model equations for vehicle subsystems, as well as obtain a solution to these equations. The users will desire to simulate different combinations of a variety of configurations. Some may want a refined simulation of a relatively small composite of components, while others may want a cursory simulation of a large network of components. The user generally will not need nor want a sophisticated simulation involving in-depth representations of several major functional systems. For instance, a study of the structures problems associated with wind loading will require a loading profile using a representative trajectory generator rather than one employing a sophisticated model of the propulsion and flight control systems. In-depth simulation of a propulsion system will include a rather cursory representation of the flight control system and vice-versa.

There are a number of solutions available for application to the mechanical and electrical problems involved in this study. Many of these have been written for special problems and are often difficult to use for anything else, while others have been written to handle very general problems. However, the price paid in using these "ready-made" solutions is excessive memory requirements and excessive running time. Ideally, then, the flexibility of the general program and the efficiency of the special-purpose program are desired. The development of a user-oriented simulation monitor system seems to provide a means to obtain this end.

Thus, the LVCLS is planned to have the following characteristics:

- The simulation system will contain a compiler that accepts a user-oriented language.
- It will utilize a large program library and launch vehicle data base to compile discrete and/or dynamic simulations as called for by the user.

- Only those programs necessary for the particular simulation at hand will be used.
- The program library will contain programs that simulate common functions, but at varying degrees of sophistication.
- There will be provisions to add and delete programs to the library.
- Once a simulation program has been compiled, it may be kept for later use.
- The output from one simulation may be stored for use with another simulation at a later time.
- The capability to be used in a time-sharing mode will be maintained throughout the development of the system.
- Provisions will be made to include existing simulation programs in the library.
- The strength of these existing simulations will be enhanced by the ability to write programs under the simulation monitor that automatically will prepare the input data from the launch vehicle data base.

There undoubtedly will be several techniques available for structuring system models in the dynamic portion of the simulation system. The suitability of several approaches has been investigated and the relative merits of these and other techniques will be resolved by the user needs. Generally, highly nonlinear functions will be handled best with differential equations, while the transfer function or impulse response will offer a number of advantages for more linear functions.

3.2 IMPLEMENTATION

There are three basic areas which must be implemented for a successful simulation of the launch vehicle at the component level. The first of these is the system description or engineering data base with its logical and functional description of vehicle hardware in terms of discrete connections and in terms of differential equations. The second is a discrete simulation to provide logical and functional analyses. The third is a dynamic simulation to provide dynamic analysis of the overall system. These items, together with the computer system on which they are run, form the core of the simulation. In addition, other supporting routines detailed in paragraph 3.4 must be provided to maintain a current and specified data base and to provide the required outputs.

3.3 COMPUTER HARDWARE

3.3.1 COMPUTER

The various program modules of this simulation should be programed and run on an existing computer system, such as the IBM 7044 with a disc file, during the initial stages of implementation. Experience with the discrete simulation system for use on the Electrical Support Equipment showed that the initial system ability of simulating some 3,500 components could be improved to the point where it now performs a discrete simulation using some 28,000 components. Final selection of the computer hardware should be postponed until sufficient operating experience with the modular parts of the simulation has firmly established the best equipment compromise for economy and efficiency. Paragraph 4.5 discusses the simulation requirements which are expected to limit the choice of computer configuration to one or more of the following modern digital systems:

a.	General Electric Company	GE - 625/635/645
b.	Burroughs Corporation	B - 8500
c.	Control Data Corporation	CDC - 6600/6800
d.	International Business Machines Corporation	IBM - 360-65/75/95

The system should include large, random-access disc units, such as the DS-25, capable of storing 200 x 10^6 characters. It is desirable to have three of these disc units in order to provide the capability of simulating different configurations without excessive waiting time. To provide ready machine access for engineers at the different MSFC laboratories, the computer system should be capable of operation in a time-shared mode.

3.3.2 PERIPHERAL EQUIPMENT

The following peripheral equipment is a partial listing necessary to fulfill system requirements:

a. Digital plotter.
b. Tape units.
c. Card readers.
d. Printers.
e. CRT displays.

3.4 ORDER OF IMPLEMENTATION

Paragraph 4.15 discusses the rationale used for development of the recommended order of implementation. Basically, the functional interdependencies of the simulation program modules and their estimated effectiveness have established the recommended order of implementation so that, as computer hardware and software is built up and specific functions become operational, individual outputs and their associated uses will become available at the earliest practicable time. In this way, operational portions of the simulation will provide useful output before the complete implementation of the overall simulation system. Unlike the basic discrete and dynamic simulation modules, the output routines may be implemented conveniently in other sequences so that MSFC may assign higher priorities to some outputs and uses than to others, thereby changing the sequence. Each of the 20 uses examined is comprised of one or more specific outputs. A total of 32 of these outputs and the associated uses are illustrated in Table 3-1 in the recommended order of implementation as shown in Appendix D.

3.5 OPERATIONAL SUPPORT PROCEDURES

To insure that the simulation accurately reflects the true system and provides optimum usefulness, it is essential to:

- Maintain software.
- Implement changes.
- Schedule use.

Maintenance of software and implementation of changes are specifically referred to as configuration control of the simulation. The methods to be applied during implementation will be similar to those published and followed for the current Electrical Support Equipment simulation. The salient feature of the existing simulation configuration control system are described in paragraph 4.4.

Configuration control of the launch vehicle simulation system will have to be carried out at a very high technical level. It is essential that the simulation at all times does in fact meet the objectives of vehicle system mathematical definition and user requirements. To this end, then, it is essential that control emphasize:

a. Limiting the data base information to that which is of specific interest.

b. Converting the data base as quickly as possible to the mathematical definition required.

c. Being informed of changes by design file indices.

d. Using a building-block approach in the growth of each simulation objective.

Outputs*	
Code	Description
AH	Listing of new approved permanent data being entered.
AI	Listing of new approved permanent changes being entered.
AG	Listing of changes which affect LVCLS operations.
P	List of equipments by panel.
Q	List of equipments by drawing numbers.
R	Listing of equipments by function.
AJ	Listing of equipments involved within specified bounds.
B	Listing of sequence of operations by time.
C	Listing of component status changes.
K	Listing for comparison run.
U	Listing of equipments activated with time or number of activations.
AF	Listing of transient response.
L	Listing of delay times for selected system portion.
M	Listing for comparison of delay times.
A	Function sequence chart.
O	Plot of transient response.
N	Listing of equipments unstable in operation.
AE	Listing of conflicts in connections.
AK	Listing of initial conditions.
V	Listing of differences between equipments expected to be activated and those actually activated.
W	Listing of effective failure rates for selected system portions.
X	Listing of probabilities of not failing for selected system portions.
F	Listing of inconsistencies.
D	Listing of redundancies.
T	Listing of equipments and fault conditions leading to fault indicator symptoms.
AD	Listing of points required for detection and isolation of faults.
H	Listing of components contributing to questionable operation.
Y	End item approved configuration indices.
Z	Approved ECP end item indices.
AA	End item quantitative requirements schedule.
AB	End item modification status.
AC	Spares status.

* See Table D-3, Page D-9.
**See paragraph D.2, Page D-27.

lementation

	Applicable Uses**																			
	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII	XIX	XX
			X																	
			X																	
		X																		
	X			X				X												X
	X			X				X												X
	X			X				X												X
													X							
	X			X			X													X
	X			X			X													X
	X			X			X										X			X
	X			X											X			X		X
	X			X		X														X
	X			X	X															X
	X			X	X															X
	X			X			X													X
	X			X		X														X
	X			X		X														X
	X			X																X
														X						
	X			X												X		X		X
	X			X														X		X
	X			X														X		X
	X			X						X										X
	X			X							X									X
	X			X								X								X
	X			X																X
	X			X					X											X
																			X	
																			X	
																			X	
																			X	
																			X	

e. Checking out each stage of the simulation against specification performance requirements and test results.

f. Demonstrating that numerical methods do not introduce computational inaccuracies and instabilities.

g. Following complete mathematical definition, machine programing, and program debugging and verification, the changes must be limited to those required by design ECP of the actual system and those required for user satisfaction.

SECTION 4

SIMULATION DESCRIPTION

4.1 GENERAL

The Launch Vehicle Component Level Simulation (LVCLS) is a part of the total Saturn V information system. It is essentially an engineering-type tool which performs a mathematical simulation of the Saturn V launch vehicle and its ground support equipment on a digital computer. The results of this simulation are presented in various forms which best support the real engineering evaluation to be performed by men.

4.2 POTENTIAL USES

During the course of the present study, there were some 20 potential uses examined. Many of these have multiple outputs which were also considered in determining the recommendations contained in this report. Use number VI, "Perform transient analysis of a selected portion of the launch vehicle and ground support systems," and number VII, "Follow signals through a selected portion of the launch vehicle on a discrete basis," form the backbone of the simulation. Other potential uses either add capability in the analysis of simulation results or support the simulation itself. All of the uses examined to date can be implemented. In fact, there are examples of each type in use in other applications now. The chief difference and advantage of LVCLS is its wide applicability to Saturn V systems and its availability on a demand basis. The following list of the uses is not arranged by priority but only to maintain the numerical sequence previously used during the study, in Appendix D, and in Table 3-1:

I. <u>Define the effect of a proposed change on operation of a selected portion of the launch vehicle and ground support systems</u> - The objective is to allow the user to change data temporarily, corresponding to a proposed change, and determine its effect on the operation of the system portion under consideration; reference to Appendix D and Tables D-5 and D-12 indicates that most of the features of LVCLS will be required.

II. <u>Keep track of approved change orders, drawing changes, and hardware changes made in the simulation data file and the resultant configurations</u> - The objective is to allow a user to keep a record of the changes that have been made, so that administrative and technical control can be maintained to prevent the invalidation of some or all of the system data. This provides a method of keeping track of changes made in the data used by LVCLS in simulating equipment operations, but it does not check the effect of such a change or the equipment being simulated, and thus it does not duplicate Use I. Further, since Use II pertains to changes in data employed by LVCLS in simulating equipment operations, it does not duplicate Use XIX.

III. Insert approved changes into the central data file - This central data file is a computerized source of all descriptive data necessary to support the simulation, such as advanced system schematics, identification, location, and characteristics of components, etc. The objective is to allow a user to update the data employed by LVCLS such that it reflects the latest system configuration. This updating consists of inserting new information for all items of data associated with:

- Connection statements.
- Logical statements.
- Advanced schematics.
- Panel schematics.
- Delays.
- Element parameters.

These changes will be made only under carefully controlled procedures and a record will be kept as outlined for Use II.

IV. Change data temporarily to simulate a fault condition and follow its effect through a selected portion of the system - The objective is to allow the user to make such temporary changes in the data that a fault condition or a number of fault conditions may be simulated. Through the use of all the features of the simulation, the effect of such fault conditions may be traced through the vehicle and/or ground system, or a selected portion of it. Changes in the data may consist of the following, singly or in consistent combinations:

- Changes in connection statements.
- Changes in delays.
- Changes in element parameters.
- Changes in logical statements.
- Elimination of elements or signals.

This feature closely parallels that whereby the effect of a proposed change may be traced.

V. Calculate expected times for events of the sequential operation of a selected portion of the launch vehicle and ground support systems - The objective is to allow a user to determine the times at which discrete operations may occur, based on delays calculated from parameters of the elements rather than on assumed delays. This pertains to electrical, mechanical, hydraulic, and pneumatic elements in the portion of the vehicle and/or ground system under consideration.

VI. Perform transient analysis of a selected portion of the launch vehicle and ground support systems - The objectives are:

- To determine the time of operation of the elements based on their parameters.
- To determine stability characteristics for the portion of the system under consideration.
- To determine transient response.

The analysis will encompass discrete and continuous operation of electrical, mechanical, hydraulic, and pneumatic elements.

VII. Follow signals through a selected portion of the launch vehicle on a discrete basis - The objective is to allow the user to follow the sequence of operations, on a discrete basis, through a selected portion of the vehicle and/or ground support system. The system includes electrical, mechanical, hydraulic, and pneumatic equipment. The time intervals may either have predefined values or be the result of calculation (such as from Use V).

VIII. Relate the simulation to the racks, equipment numbers, etc., as given on panel schematics, interconnection diagrams, and advanced system schematics - The objective is to provide, for the user, the ability to relate the equipment used in the vehicle and/or ground support equipment to the drawings. The comparison can be made by:

- Coding equipment designations so that they reflect type and location.
- Tying the equipment to panel and rack.
- Tying the equipment to drawing and page number.

IX. Search out closely timed operations and identify the equipments involved to eliminate areas of questionable operation where chance plays a significant role in the operation of a system - Closely timed functions (e.g., relay races) may sometimes appear inadvertently in a system, so this use is being considered as a means of calling attention to such a condition, if it exists. Changes in the data which may result from such a disclosure are not automatic - they would require evaluation and then use of procedures for updating the data. The objectives are:

- To search out and define closely timed operations which may result in questionable operation of parts of the vehicle and ground systems.
- To define the equipment involved in such operations.

X. Check for inconsistencies such as conflicting signals and component operations which lead to inconsistent functions - Inconsistent functions are not intentionally designed into a system, so this is a means for emphasizing such a condition, if it exists. Changes in the data which can accompany such a disclosure are not automatic, they would require evaluation and then use of procedures necessary for updating data. The objectives are:

- To check for signals which conflict.
- To check for components, the operation of which lead to inconsistent functions.

XI. Check for redundancies to detect unintentional multiple methods of obtaining individual signals or modes of operation and also to verify the presence of intended redundant signals or modes included to improve reliability - Redundancies in a system may appear inadvertently or may be included intentionally (e.g., to increase reliability). This is a means for calling attention to the fact that a redundancy may exist. Changes which may result in the data will not be automatic - they would require evaluation and then use of procedures for updating the data. The objectives are:

- To check for multiple methods of obtaining individual signals.
- To check for components that lead to duplicate functions when operated.

XII. Define areas of possible malfunctions given a set of symptoms - This feature is essentially the reverse of Use XIV. Its objective is to define possible malfunctions which can give rise to a given set of symptoms. It would be restricted to the case where the symptoms could result from single rather than multiple simultaneous equipment malfunctions.

XIII. Allow a user to set up conditions which identify a portion of a proposed or actual checkout or countdown sequence - The objective is to allow the user to set up the conditions which identify a portion of a proposed or actual checkout or countdown. This is related to Uses I and IV, and it provides a means for specifying the time interval or equipment bounds of the system portion to be investigated, along with the necessary initial conditions.

XIV. Allow a set of simulated fault conditions to be superimposed on a list of conditions defining a planned checkout or countdown sequence - The objective is to allow the user to simulate a system fault during a simulated checkout or countdown. This is related to Uses I, IV, and XIII, and provides a means for specifying simulated faults in a simulated checkout or countdown sequence.

XV. Define and keep track of equipments that have been activated and maintain a record for output - For example, during a specified period of checkout or countdown, a record could be kept of which elements were activated, along with either the number of activations or time during which each was in the active state.

XVI. Define equipments which have not been activated - For example, given a list of equipment which should be activated during some period of a checkout or countdown sequence, define those equipments which were not activated.

XVII. Compare resulting sequences with desired ones - The objective is to permit a user to compare equipment states resulting from a simulation run with expected states either for checkout or countdown activities. It employs outputs from other uses and compares them with expected results.

XVIII. Determine expected reliability factors for a selected portion of the system - The objectives are:

- To determine the effective failure rate for a selected portion of the system based on checkout or countdown usage.
- To allow the user to predict the probability of a failure in an equipment during a selected portion of a test sequence.

XIX. Configuration management documentation data center and control - The objective is to allow a user to follow the procedural concepts required for identification, control, and accounting for all systems, equipments, and components of the Saturn V launch vehicle as outlined in NPC 500-1.

Specifically, this is accounting information directed toward keeping track of:

- Specifications for contract end items.
- Changes to and maintenance of specifications.
- Engineering documentation required for:
  - Design releases.
  - Design changes.
  - Design reviews.
  - Test acceptance and reviews.

This entails storage of document identification indices into the computer bulk memory rather than the documents themselves.

XX. Development of checkout and countdown procedures - The objective is to aid in preparing test procedures for checkout and countdown. It is not intended that this usage write the procedures, but, by simulating the consequences of a procedural step, it acts as a tool for individuals responsibile for writing such procedures.

4.3 ORGANIZATION OF IMPLEMENTATION

Modularity is the key to the organization for development, for the data base, and for utilization. During development, each computer program will be essentially complete in itself. A program will not, in general, be dependent on other programs for essential functions. This will insure that output will be available from separate programs when they are programed without the necessity of waiting for the completion of programing for the entire simulation. However, the structure and functional relationships of the complete simulation will be designed into each separate program so that they will fit together into an integrated simulation system.

The data base will be stage and engineering oriented. It will be preprocessed from basic random disc files into linked and sequentially organized working disc files to provide working data for various portions of the vehicle system bounded by time and by hardware relationships. This preprocessing will insure that the bulk working data base actually used in computations will be available in minimum access time and the simulation can operate efficiently. These data will be file protected and verified. Temporary changes or working data will not be placed in the main data base without going through proper change procedures and being input through the control program. Permanent change data will be provided through flagging of the engineering data base through the configuration management data base system and subsequently entering the new engineering data into the random access disc file and processing it for entry into the working disc files.

Utilization will be possible on a time-shared basis. The monitor system will select the portion of the data base and those computer program modules necessary for a particular request and duplicate them for use during computation without destroying the capability of the simulation to set up other combinations or repeat the same combinations for other users. Dynamic and/or discrete simulations may be processed concurrently and their results saved for possible later use if desired.

4.4 SIMULATION CONFIGURATION CONTROL

It will be necessary to set up a configuration control procedure to insure that the simulation and its data base are protected at all times. Protection will also be maintained during development and for changes to insure evaluation prior to any implementation of a change. There is an existing control procedure for the Electrical Support Equipment simulation and the salient features of it are given below as an example of what must be developed. The control procedure written for LVCLS will be more complete and in accordance with the forthcoming computer program configuration management document, NPC 500- .

1.0 CHANGE DEFINITION

For purposes of this document the two types of changes considered are defined as follows:

- Error Correction - A computer program change that is necessary because a program bug is discovered after a routine is in production.
- Design Change - A computer program change that is desirable, to change a program function after a routine is in production.

2.0 CHANGE REQUEST FORMS AND PROCEDURES

All program error corrections must be made by a programer and in most cases the programer will be the first to discover a coding error. However, when users discover inconsistencies in program operation they should discuss the situation with the Programing Task Leader who will then inform the cognizant programer of the apparent error. If the programer finds a program error he must make the appropriate changes and fill out a Program Error Notice form shown in Figure 1. The completed form must be submitted to the Programing Task Leader for approval of the program correction and distribution of the notice to members of the Digital Simulation Group.

The elements of the Program Error Notice form are as follows:

Program Name identifies the specific routine of interest.

Version Number indicates the system version, e.g. Version I.

Modification indicates the modified program by modification number.

Name of Programer is the name of the individual responsible for correcting the program error.

Date is the month/day/year the program error is corrected.

Description of Error is a precise technical description of the program error with an indication of impact on past user outputs.

PROGRAM ERROR NOTICE

Program Name____

Version Number____________________ Modification____________________

Name of Programer____________________ Date____________________

Description of Error: ____

Corrective Action Taken: ____

Correction Approval____________________

Program Task Leader

Figure 1. Sample Program Error Notice Form

Corrective Action Taken is a description of program changes made in coding, compilation, source deck, and documentation.

Correction Approval indicates the program error has been identified correctly and proper corrective action has been taken.

2.2 DESIGN CHANGE

To simplify the operational computer program design change process, a change request form was designed and is shown in Figure 2. The form is to act as the change action record as each change is processed through the steps shown in Figure 3.

The change request forms will be available from the Methodology Group Leader or they may be prepared by a typist. With the form available any individual on the simulation project may request a design change by filling out Section A of the form and submitting it to the Methodology Group Leader. The elements of Section A are described below:

Change Request No. will be assigned by the Methodology Group Leader.

Date is month/day/year on which the request is made.

Type of Change is the desired program function change.

Program Identification: the specific program to be changed e.g. Test Procedure Output Program (GETS)/Version I/unmodified.

Description of Design Change: a detailed description of the new program capability that should result from the change.

Change Required By is month/day/year change is needed.

Change Requestor is the name of the individual requesting the change.

After the design change request is submitted it becomes an action item and is distributed for evaluation. Usually the Programming Task Leader and the cognizant programer will evaluate the desired change. When the evaluation is completed the individual responsible for the evaluation must complete Section B of the change request form and submit it to the Methodology Group Leader. The elements of Section B are described below:

Change Recommendation and Comment: a yes/no recommendation and general comment regarding the scope and worth of the change.

Esitmated Cost of Change: a cost summary giving the total cost, the manpower needed, computer hours (including keypunch and tab room) and documentation costs.

OPERATIONAL COMPUTER PROGRAM CHANGE REQUEST FORM

Change Request No. ____________________ Date ____

Type of Change: ____

Program Identification: Name ____

____________________ Version ________________ Modification ____________________

Description of Design Change: ____

____

____

____

Change Required By: ____________________ Change Requestor: ____________________

Change Recommendation and Comment: ____

____

____

____

____

Estimated Cost of Change: Total ____

Manpower ____

Computer Hours ____

Documentation ____

Estimated Completion Date ____________________ Change Evaluator ____________________

Change Authorized ______________________________ Date ____________________
Project Manager

Change Completed ______________________________ Date ____________________
Methodology Group Leader

Figure 2. Sample Operational Computer Program Change Request Form

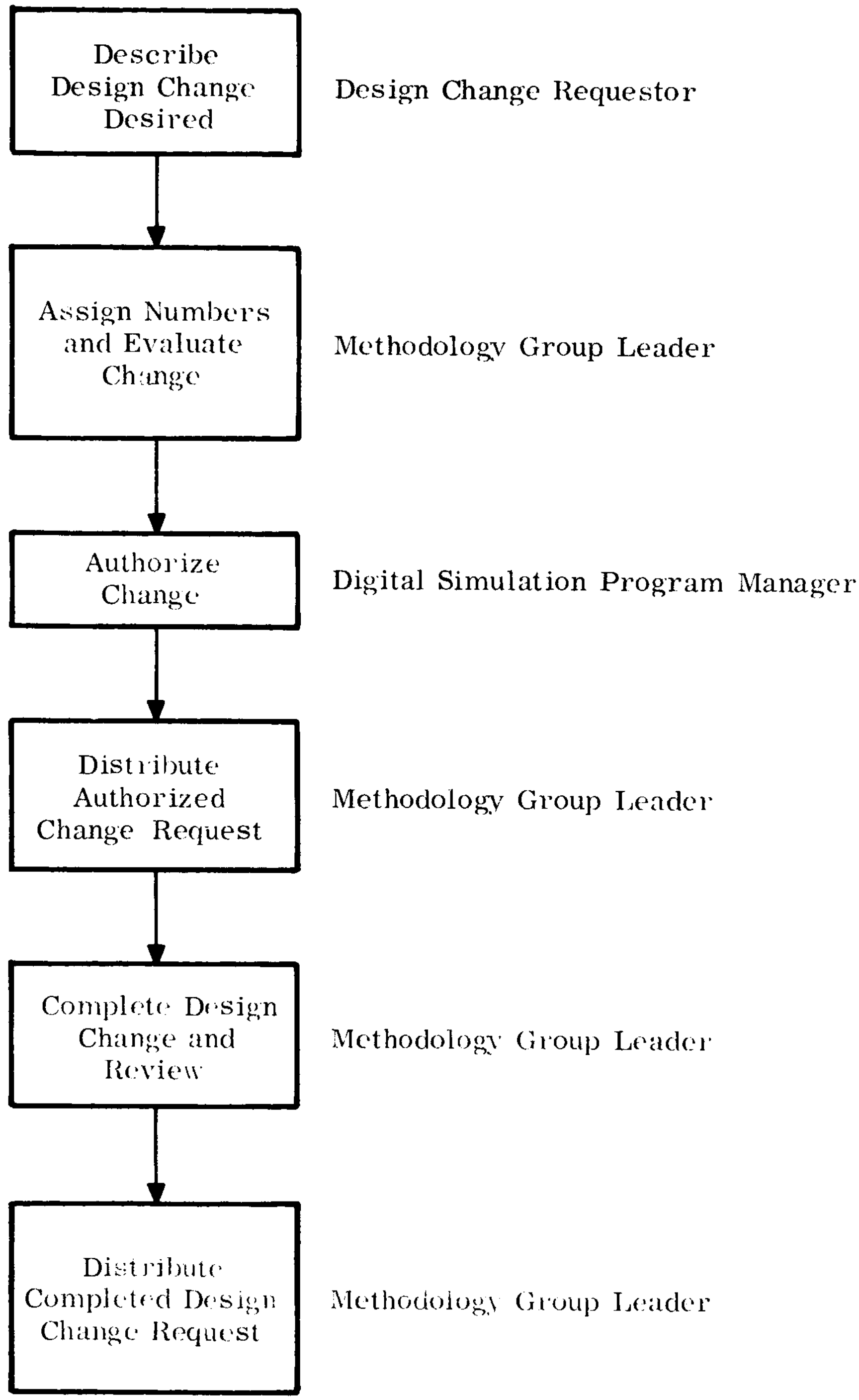


Figure 3. Flow Diagram of Design Change Procedure

Estimated Completion Date: the calendar weeks required to implement the change assuming an immediate go-ahead.

Change Evaluator: the name of the individual making the evaluation.

After the evaluation is complete a decision must be made to authorize the change. If the change is authorized Section C is filled out by the Digital Simulation Program Manager and the Methodology Group Leader will send a copy of the change request to the requestor notifying him of the action taken.

Upon completion of the change (specifications revised, code changed, new listings obtained, source deck updated, description documents revised, etc.) Section D will be filled out and the change request form will be distributed as notification of change completion.

Scheduling and user communication tie together in terms of the particular requirements of the user, i.e., his priority, the area of simulation equipment required, the status of model, output formats, and simulation development needs:

a. Priority - User priority will be based on the level of urgency of the output data in relation to its application. Factors such as design, manufacturing, test, and launch schedules will be used in the estimate.

b. Simulation equipment required - The simulation equipment required will depend on the nature of the problem. For example, a dynamic simulation of some part of the system will require main processor time for assembly and run, in addition to peripheral equipment for outputs. Problems in the areas of configuration accounting, file search, file maintenance, etc., can be performed on off-line equipment.

c. Status of model - Users requesting service in areas where model development is required must submit test and/or design information pertinent to the system to be simulated. Model status will also have to be considered in areas where the simulation is being up-dated to reflect design changes. User needs may not be satisfied until the model truly reflects his system.

d. Output formats - Output formats must conform to user requirements. These will be in the form of graphs, printed output, special forms (such as for configuration accounting), punched cards, paper tape, magnetic tape, and direct CRT displays.

e. Simulation development - Simulation development follow-on procedures will be instituted. This will insure that future customer needs will be met.

4.5 COMPUTER CONFIGURATION

The recommendation of this study is to develop the individual modules of the simulation using existing computer systems with disc capability and make a selection of operational computer hardware based on computing experience. However, there is sufficient basis to expect the final configuration to have the following capabilities:

- Core memory of 131K words or greater.
- Access time of one microsecond with compatible add, multiply, and divide times.
- Multiple disc memories each having a capacity of 33M words.
- Time-sharing capability.
- Remote terminals two of which should have local processors.
- Digital plotters.
- CRT displays.
- High-speed tape, card reader, printer, etc.

Experience with the current discrete simulation of the Electrical Support Equipment has shown that computer running time is sensitive to small changes in the functional description of the hardware and even to the sequence of its internal processing of equations. Runs which were expected to take 20 minutes on an IBM 7044 computer have been completed in 10 minutes and almost identical problems have taken 40 minutes. However, the following projections are made based on the "most typical" run experience and allow for recent improvements in computational technique.

As an example, to develop a discrete event test procedure for LVCLS would require the following time on a GE-635 or equivalent digital computer:

• Number of components simulated (Current active components/stage = 3400)	9000
• Number of sections in procedure	10
• Number of steps initiated per section	250
• Computational speed of simulation	25 steps/minute
• Computational speed of output	50 steps/minute
• Number of complete runs required	3

$$\text{Time per discrete test procedure} = \left[\frac{(10\ \text{sections/procedure})(250\ \text{steps/section})}{25\ \text{steps/minute}} + \frac{(10\ \text{sections/procedure})(250\ \text{steps/section})}{50\ \text{steps/minute}}\right] 3\ \text{runs}$$

$$= (100\ \text{minutes/simulation run} + 50\ \text{minutes/output run})\ 3\ \text{runs}$$

$$= 450\ \text{minutes} = 7\ 1/2\ \text{hours for 3 runs of a 9000-component subsystem}$$

Based on the number of additional iterations, it would take 30 hours for 18,000 active components and it would take 67 1/2 hours for 27,000 active components.

On the other hand, a typical dynamic simulation is expected to require the solution of some sixty different equations. There will also be some five to eight different sets of coefficients for possibly thirty of these equations. This would indicate some 200 to 300 equations to evaluate. The anticipated types of equations include the following forms:

- LOX Tank

$$P_{LH} = P_{gLO}\left[\frac{1 + \frac{1}{M_{gLO}}\int(\omega_{gL} - \omega_{VL})\,dt}{1 + \frac{1}{\rho_L V_{gLO}}\int \omega_L\,dt}\right] + \frac{\rho_L\left(1 + \frac{a}{g}\right) h_{LO}}{144} - \frac{1 + \frac{a}{g}}{144 A_L}\int \omega_L\,dt$$

- Engine Fuel Inlet Line

$$P_{FH} - P_{FO} = \left(R_{F1} + \frac{R_{FV1}}{\Omega_{FV1}} + C_F + R_{F2}\right)\omega_{F1}^2 + (L_{F1} + L_{F2})\frac{d\omega_{F1}}{dt} - \frac{\rho_F\left(1 + \frac{a}{g}\right)h_{F10}}{144} - H_{FO}N^2 + f_F\,(\mathrm{NPSH})$$

- Thrust Chamber Pressure

$$\frac{dP_T}{dt} + \frac{1}{\tau_T} P_T = \frac{1}{\tau_T A_T T_T} \left\{ \sqrt{\left[\frac{R}{\sqrt{\gamma \left(\frac{2}{\gamma H} \right)^{\frac{\gamma + 1}{\gamma - 1}}}} \right] \left[K_1 + K_2 \left(\frac{\omega_{LT}}{\omega_{FT}} \right) \right]} \right\} (\omega_{LT} + \omega_{FT})$$

- Turbine/Pump Configuration

$$\frac{dN^2}{dt} + \frac{2}{In_o^2} \left[\frac{K_L}{\epsilon_L} W_{L1} + \frac{K_F}{\epsilon_F} W_{F1} \right] N^2 = \frac{\epsilon_T}{778} (\Sigma \omega_g) \Delta H$$

$$+ \left[\frac{C_F}{\epsilon_F} W_{F1}^3 + \frac{C_L}{\epsilon_L} W_{L1}^3 \right]$$

- Heat Balance Equations (Plenum)

$$C_x \frac{d\overline{T}_x}{dt} = C_{P_g} (\Sigma \omega_g) (T_{x1} - T_{xo}) - (uA)_{xL} (\overline{T}_x - \overline{T}_L) - (uA)_{xH} (\overline{T}_x - \overline{T}_H)$$

- Turbine Back Pressure

  Assume square law flow and perfect gas law so that

$$P_d^2 - P_{atm} P_d = R_x \overline{T}_x (\Sigma \omega_g)^2$$

Another example would be to perform a dynamic simulation of the S-IC stage propulsion system which would require the following time on a GE-635, or equivalent, computer:

- Number of locations at which data are required (from 30 to 300 depending on parameter) 60
- Number of parameters required (temperature, energy, velocity, density, static pressure, total pressure) 6
- Number of output time steps desired 200
- Number of calculations per output time step 25

- Number of engine systems 5
- Computational speed of simulation 1500 calculations/second

$$\text{Time per dynamic simulation} = \frac{(60\ \text{locations})(6\text{parameters})(200\ \text{time steps})}{(1500\ \text{cal/sec})(60\ \text{sec/minute})}$$

$$\times\ (25\ \text{cal/time step})(5\ \text{systems/simulation})$$

$$= \frac{9{,}000{,}000\ \text{cal/dynamic simulation}}{90{,}000\ \text{cal/minute}}$$

$$= 100\ \text{minutes per dynamic simulation}$$

There are four classes of computers available now or in the near future that are capable of performing the LVCLS and are all classed as information processors. They are the General Electric GE-625/635/645, Burroughs B-8500, Control Data Corporation CDC-6600/6800, and the International Business Machine IBM 360-65/75/95 for which characteristic data are included in Table 4-1. Selection of the computer system that best satisfies all simulation requirements should be postponed until after the detail design requirements specification is developed and more detailed information becomes available from the development and implementation of individual program modules. The implementation of these program modules may be done on an available IBM 7044 machine with a 32K core memory and disc files. The knowledge and experience gained during initial development will provide a better measure of the data storage, buffering, and timing requirements for subsequent selection of final computer hardware.

The processing system for LVCLS should be capable of simulating an estimated 300,000 components. Each component will be described by about 36 words of 36 bits each for a total of 1300 bits of storage. The anticipated words are:

- 3 words for drawing and identification number.
- 1 word for item number.
- 12 words for item name.
- 2 words for component identification.
- 8 words for logical expressions.
- 10 words for dynamic expressions.

For ease of processing and data acquisition requirements of the overall simulation, several of these words should be listed multiply, thereby expanding the data base to twice or possibly three times its minimum size.

Table 4-1

Characteristic Machine Data

Computer	Word Size (bits)	Full Core (megabits)	Exponent (bits)	Mantissa (bits)	Decimal Equivalent (digits)	Characters/Word
GE-625/635/645	36	9.4	8	61	18	6
B-8500	48	12.6	12	73	21	8
CDC-6600/6800	60	7.9	11	95	28	10
IBM 360-65/75/95	32	134	8	54	15	4

One minimum description of 300,000 components uses 10,800,000 words and will require three DS-20 disc storage units (3.9×10^6 words each). The DS-25 disc storage unit on the other hand is capable of storing 33×10^6 words each and could contain a full description of 300,000 components with up to three times the minimum size description. Use of the DS-25 or equivalent disc unit will enable the simulation to operate on multiple configurations, up to four disc files at one time, and perform operations on one configuration while simultaneously loading the data base for another. Computer time chargeable to loading a configuration on a DS-25 disc will range from 6.3 minutes to 7.1 minutes depending on what other operations are taking place on the computer at the time. This is exclusive of the initial ten hours of preprocessing time required to organize the working data base from random storage on source disc files. Preprocessing is required only once per configuration.

The computer characteristic data presented in Table 4-1 for comparison includes exponent and mantissa double precision sign bits but does not include redundant representations. In the case of the IBM 360, a mantissa of 54 bits is more representative than 56 bits because this machine does not fully normalize floating point numbers.

Each of the above computer systems is equipped with the standard supervisory software including FORTRAN IV and COBOL. The CDC version of FORTRAN IV is nonstandard and programs written therein are not directly transferable to other computers. The Burroughs B-8500 computer is available with ALGOL in addition to the other languages. Also, compilation and information handling times on the B-8500 should be lower because of its list-processing structure.

It is anticipated that portions of the dynamic simulation of the LVCLS will require special programing in order to avoid numerical instabilities. There will be fewer of these problem areas which will affect processing time and programing complexity if the available object computer has a greater number of significant digits to work with.

With the exception of the CDC machines, the recommended computer systems have fully compatible I/O configurations. The CDC machines are associated with 12-bit-word satellite modules which will impose programing problems with respect to data handling, since an additional manipulation will be required between the central processor and bulk data, which will not be necessary on the other computers.

An Electrical Support Equipment simulation presently operates within the capabilities and limitations of an IBM 7044. Many of the same capabilities will be required to realize component level simulation. Because of the larger data base and more pressing numerical stability problems, considerable thought should be given to fixed-point arithmetic and machine-dependent programing.

If more than one class of computer is used during initial development of LVCLS, machine-independent programing may be justified during development. However, the operational simulation should be programed to take advantage of the computer hardware installed at MSFC by machine-dependent programing. This will improve the efficiency of the simulation with respect to running time and storage requirements. With frequent utilization, these improvements become more important.

4.6 DATA BASE

4.6.1 DESCRIPTION

4.6.1.1 General

The data necessary to support LVCLS are functional. They are the logical expressions, the connection statements, the dynamic characteristics, dynamic equations, and similar engineering data used in current analysis. These data may be related to the administrative data such as cost, manhours, location, date, quantity, proposal number, etc., as used by configuration management through drawing numbers or part numbers but the contract end-item linkages used for configuration management are not the same as the functional linkages used in LVCLS and engineering analysis. However, the simulation will support the engineering evaluation of proposed changes to contract end items, and the implementation of changes through the configuration management system should flag parallel changes in the engineering data base where they are applicable.

Source engineering data should be gathered concurrently with configuration management data from the stage contractors and other basic sources. There should be a protected random access disc file system set up for these engineering data required by LVCLS. From the source engineering data file, there will be protected working engineering data gathered. These working bulk storage files will also be on disc systems but will be organized by vehicle configurations. The working file will be preprocessed and organized into linked families for computational use utilizing multiple listing of data where advantageous.

For each simulation run there will be a temporary problem-oriented data file selected from the working bulk storage files. The selection criteria will be applicable hardware bounds, time bounds, and type of simulation run. Although results of each simulation run may be saved, they will not reflect directly back on the working data nor on source data. The applied failures and/or temporary changes inserted into problem data for evaluation purposes will be isolated. Changes to source data and to working bulk storage files will be controlled through both hardware and simulation configuration management procedures and inserted only through application of the proper change routine of the simulation.

The essential differences between the two systems are described in the following paragraphs.

4.6.1.2 Configuration Management Design File

This is a business-oriented system whose purpose is to maintain:

a. Assembly records.
b. Parts records.
c. Usage records.

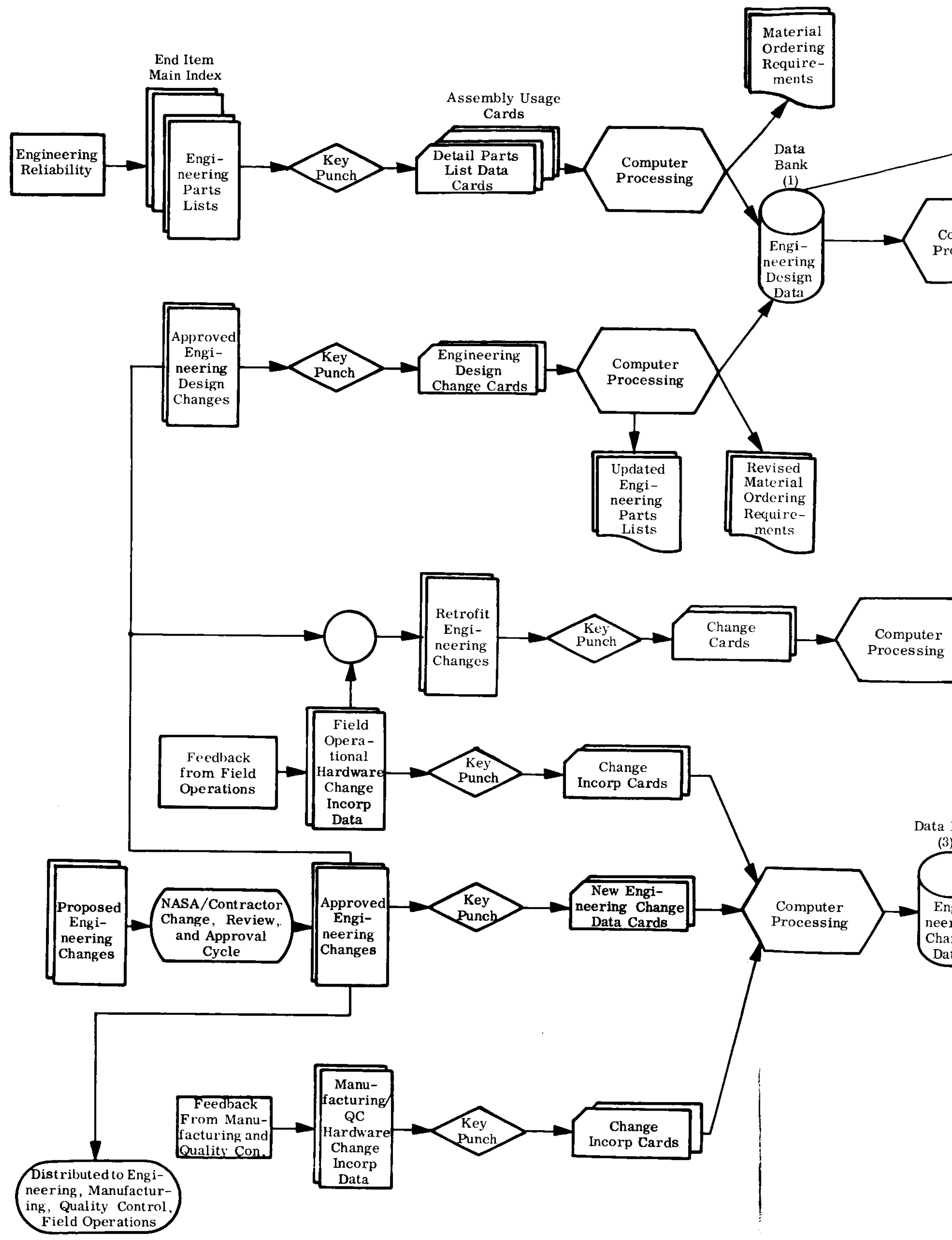
The system is maintained in detail in the form of:

a. Engineering design data.
b. As-is configuration data.
c. Engineering change data.

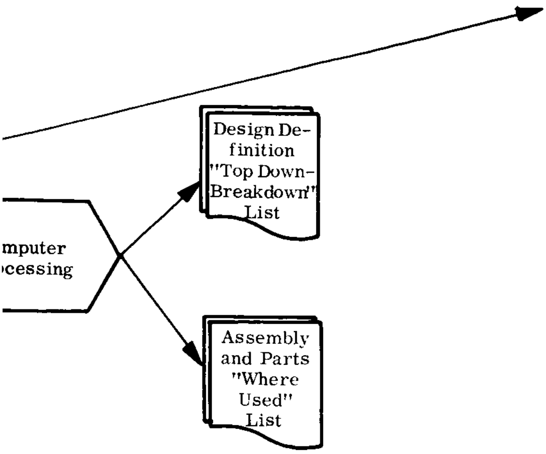
This system is maintained for customer reference upon request. Configuration management reports (see Figure 4-1), which serve as system status indices, are submitted at periodic intervals to the customer.

4.6.1.3 Simulation System

This is an engineering-oriented system whose purpose is to assimilate system functional relationships and to provide design and test validation outputs.



End Item Main Index
Engineering Reliability
Engi-neering Parts Lists
Key Punch
Assembly Usage Cards
Detail Parts List Data Cards
Computer Processing
Material Ordering Require-ments
Data Bank (1)
Engi-neering Design Data
Approved Engi-neering Design Changes
Key Punch
Engineering Design Change Cards
Computer Processing
Updated Engi-neering Parts Lists
Revised Material Ordering Require-ments
Retrofit Engi-neering Changes
Key Punch
Change Cards
Computer Processing
Feedback from Field Operations
Field Opera-tional Hardware Change Incorp Data
Key Punch
Change Incorp Cards
Data
(3)
Proposed Engi-neering Changes
NASA/Contractor Change, Review, and Approval Cycle
Approved Engi-neering Changes
Key Punch
New Engi-neering Change Data Cards
Computer Processing
Feedback From Manu-facturing and Quality Con.
Manu-facturing/ QC Hardware Change Incorp Data
Key Punch
Change Incorp Cards
Distributed to Engi-neering, Manufactur-ing, Quality Control, Field Operations



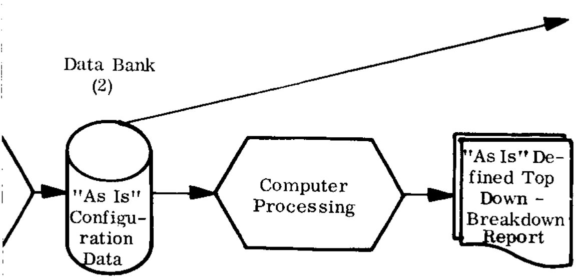
(1) ENGINEERING DESIGN DATA

ASSEMBLY USAGE RECORDS

1. Assembly Drawing Number and Revision Letter
2. Assembly Name
3. Quantity
4. Next Higher Assembly Drawing Number and Revision Letter
5. End Item Number
6. Assembly Reference Designator (MILSTD 16C)

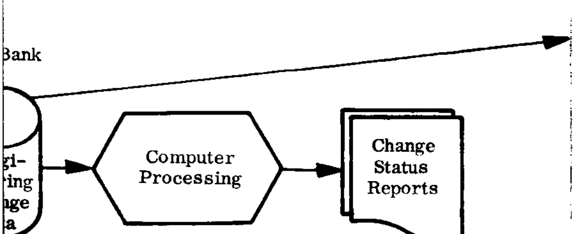
DETAIL ASSEMBLY PARTS LIST RECORDS

1. Drawing or Part Number and Revision Letter
2. Parts List Item Number
3. Part Name
4. Quantity
5. FSCM Code (If Applicable)
6. Change Identification Number
7. Procurement Code
8. Parent Assembly Drawing Number and Revision Letter



(2) "AS IS" CONFIGURATION DATA

Same Data Elements as Engineering Design Data Records Except Add Serial Numbers as Applicable.



(3) ENGINEERING CHANGE DATA

CONTRACT END ITEM RECORDS

1. Identification Number
2. Name
3. Spec Number
4. Drawing Number
5. Serial Number
6. Location Code

ENGINEERING CHANGE RECORDS

1. ECP Number
2. ECP Title
3. CCN Number
4. Retrofit Number
5. Change Document Number
6. Drawing or Part Number and Revision Letter
7. New Drawing or Part Number Created (If Applicable)
8. Effectivity (End Item Numbers)
9. Approval Date
10. Scheduled Incorporation Date
11. Actual Incorporation Date

Figure 4-1. Configuration Accounting

Specifically, then, the design file is parts-list-oriented, while the simulation is function-oriented. The design file would serve to some extent as an up-to-date library reference index for the simulation.

4.6.2 PROCEDURES

4.6.2.1 Data Base, Programs, and Output Requirements

The procedure for determining the simulation requirements is:

a. Scope out system to be simulated and/or users applications.

b. Determine types of output required by user.

c. From a and b, determine the nature and extent of data elements that must be fully defined and processed.

d. Determine computer programs required to handle data elements and supply required outputs.

Since the data base, computer programs, and output requirements are intimately related, the above four areas would be developed concurrently or with maximum overlap to produce programs and output formats suitable for expanded utilization.

4.6.2.2 Design File

The gross outline for the configuration accounting system is shown in Figure 4-2 which follows standard business system practices and is consistent with the requirements of NPC-500-1.

There are four paths shown:

a. Engineering design data.

b. As-is configuration data.

c. Engineering change data.

d. Configuration management report.

Establishment of the file would proceed as follows:

a. Examine manufacturing parts list, assembly, and usage records to determine overall accounting system requirement.

b. Arrange above data for key punch operations.

c. Transfer data to tape (or disc) storage.

d. Set up data link with engineering, manufacturing, quality control, and field operations for change identification and up-date cycle.

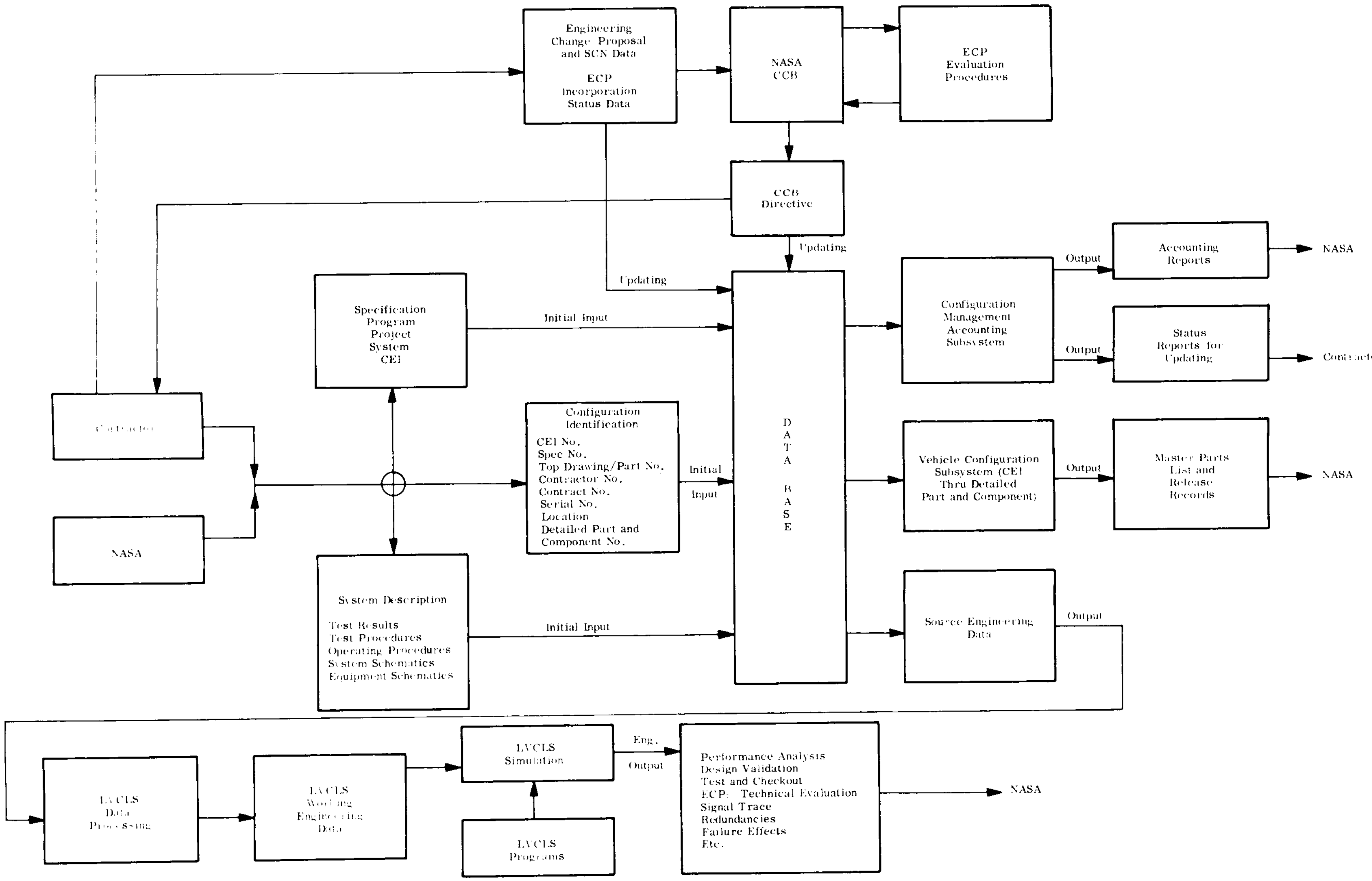


Figure 4-2. Overall Information Flow Functional Block Diagram

4.6.2.3 Simulation

The procedures required to establish and maintain the simulation system are not as clear cut as for the business system. This arises from the intent to provide a system having considerable scope and flexibility. Appendix D presents 20 usages and their functional relationships. Note that all usages tend to overlap in terms of functions and programs, presenting the question of program arrangement and use - that is, can the overlap be handled by a single set of tapes or should complete usage program tapes be supplied even though it means duplication? It is felt that the latter procedure should be followed when one considers that time-shared or multiple processors might be available so different programs can be run at the same time. Individual packages also are desirable from the standpoint of freedom to duplicate for remote utilization.

Another area that will require careful handling is that of data base. Specifically, the data base is that set of functional information completely describing the component or system in question in terms of the intended usage and outputs required. The combination of usage/data base/and output can be lumped under the term modeling activity. The activity will determine the precise nature of the data required for a given usage. For example, an area that has proven useful is the ESE discrete simulation. The modeling activity has concentrated on defining the ESE primarily in terms of its logical sequence of events - with some additional sophistication in terms of pure delay time between command and reaction. Thus, the sequence of modeling events is:

a. Data base:
   (1) Logic equations.
   (2) Delay times.
b. Computer program:
   (1) Compiler program to convert logic expressions into machine language.
   (2) Evaluate logic expressions by means of Boolean arithmetic.
c. Outputs:
   (1) Relay states as a function of time.
   (2) Lamp states as a function of time.
   (3) EA pen printout.

Another area that is completely different in nature, but follows the same pattern, would be that of the mechanical systems as described in Appendix A. Briefly, the modeling activities are:

a. System definition:
   (1) Hydraulic system.
   (2) Gas system.
   (3) Valve actuating system.
   (4) Indicator system.
   (5) Propellant tankage.
   (6) Turbo pumps.
   (7) Engine thrusters.
   (8) Sequence control.
   (9) Pressurant system.
   (10) Thrust control.
   (11) Regulators.
b. Data base:
   (1) Differential equations (linear and nonlinear) to describe system dynamics.
   (2) Arithmetic equations and function generation.
   (3) Discrete elements.
c. Computer programs:
   (1) FORTRAN or JOVIAL languages.
   (2) Numerical integration.
d. Outputs: System variables versus time.

It is evident that the latter case is far more complex and will require a much higher level of sophistication in the identification and definition of data base, program, and outputs.

4.7 USER ACCESS

The chief users of LVCLS will be engineers located at the various laboratories of MSFC. They must have ready access to the simulation through computer hardware and through their ability to communicate their particular set of requirements to the simulation system. The computer hardware recommended includes remote terminals and time-shared capability to satisfy the first of these two requirements. The second requirement may be satisfied by the development of a user-oriented language which will allow each engineer to express himself in terms with which he is familiar and, at the same time, translate these instructions into other terms which will operate most efficiently on the computer.

Component level simulation will result in a software package of considerable scope and extent. It is anticipated that there will be a large number of users, each with his own input/output requirements upon the simulation. Each user of the simulation will have his own needs with respect to report generations. And, unless the problem is anticipated, he will have his own communication problems with the computer.

Any user-oriented language is a device which collates programs and subprograms as well as identifying data and output formats. This must be done using a language in which each user has some degree of familiarity. The vocabulary needed should be defined by the individual users. Of course, processing considerations will result in compromises, but the objective of communication ease should not be lost.

Such a language becomes an integral part of the simulation. It represents data to the supervisor programs which enable proper execution according to user needs. Because this language selects programs rather than generates them, the problem is one of cross-indexing by which a program associated with a computation-oriented identifier can be fetched using a user-oriented identifier. It is anticipated that any given program may have more than one name. In the event the usage calls for multiple program execution, the ordering of these executions will be controlled by the user language.

It must be pointed out here that such a language is in no way in the category of FORTRAN. It will not result in the generation of any new programs, and hence will not require fancy programing techniques. It is believed that a user-oriented language of the ATOLL type can be incorporated into the simulation without any significant increase in manpower. That is, the simulation package imposes essentially the same programing difficulties with or without such a language. It will, however, be of considerable help if each user were to submit the type of words he would like to use.

4.8 OPERATING PROCEDURES FOR SIMULATION

During the detailed design and coding of the simulation, there will be a set of operational support procedures developed specifically for LVCLS. These will be similar to existing procedures discussed under paragraph 3.4. In addition, the basic structure of the simulation and its engineering data base are modular in order to facilitate future development of the simulation. Changes, additions, and deletions will be made in individual modules without necessitating complete reprograming of the simulation. The support procedures will insure that each proposed change to the simulation or its data base are evaluated fully prior to implementation and that each implemented change is properly authorized.

4.9 TURN-AROUND TIME

Prior to the period when turn-around time is germaine, there is a very large effort necessary and much time to be expended in providing a data base or system description for utilization by the simulation. The estimated 300,000 components for the LVCLS will require a tremendous initial data collection effort. Current estimates indicate that once these data are organized and placed upon key punch sheets there will then be 3000 key punch man-days, 20 hours card-to-digital-magnetic-tape transfer time, 10 hours data processing and organization on a computer, and 1/2 hour digital-tape-to-binary-disc-files transfer time required. However, this is done only once per vehicle configuration for all uses of the simulation. During operational use, a configuration may be loaded from binary tape or portable disc to the binary disc file on the computer in from 6.3 to 7.1 minutes.

Turn-around time is dependent not only on the computational time for the problem but also on the work load of the computer system. To accurately determine running times, problem size, etc., we should have operational programs in use which are still in the future. However, from other computer programs and problem sizes such as encountered in the Electrical Support Equipment simulation and dynamic simulations now operational, an engineering estimate can be made as follows:

a. There is no lost time in loading the next job.
b. Each discrete simulation run is equivalent to two sections of test procedure (paragraph 4.5) and takes 1/2 hour. Four per day will be run for a total of 2 hours.

c. Each dynamic simulation run is equivalent to two input conditions for a propulsion system, takes 3 1/3 hours, and one per day will be run for a total of 3 hours and 20 minutes.

d. There are 6 hours available for computing each day.

There would be one dynamic simulation, four discrete simulations, and two miscellaneous jobs run on an average day. This will result in approximately 12 magnetic tapes of output per day with 25,000 lines per tape to be printed at a rate of 1000 per minute. This would require five hours of off-line listing which could be done on second shift for a one-day turn-around. If listing is done on prime shift, then turn-around will increase to two days.

Output list generation times depend upon the ordering and identification properties of the data base. That is, it is faster to retrieve a connected set of information than to randomly access a scattered set of information. This is because a connected block lying on a single disc track requires only a single access, while randomly located information calls for an access for each item. Where the output calls for updating the component configuration in time, information must be processed before the output listings are completed.

The following capabilities and requirements were considered in arriving at the above estimates: Disc access requires about 50 milliseconds. Information in an average block or track is about 480 words. Data transfer is about 10,000 words per second or 10 words per millisecond. Hence, 480 words stored on the same track should require about 100 milliseconds to access, while if each word is on a different track, 24,000 milliseconds would be required. Assuming nominal efficiency of storage, that is 48 words stored in ten tracks, access would require about 500 + 50 or 550 milliseconds. Since 48 words of input data represent 1 1/3 components, random component access will take about 0.4 second each.

An average disc track allows for storing 13 components, with 12 additional words for miscellaneous information (not used in any presently anticipated use). One might anticipate that 6 of these 13 are pertinent (on the average) for any given use. Acquisition of all 13 can be accomplished in 0.1 second. One can reasonably anticipate a retrieval time of 17 milliseconds per component from the LVCLS data base.

If, on the average, a computer instruction takes 4.25 microseconds, then 4000 computer instructions may be executed during the access time required for retrieving a single component. Many outputs will not require extensive computational operations but will be limited by storage and retrieval capabilities rather than computational speed.

Tape speeds of today (800 bpi/150 ips) allow for storing about 2.5 megawords in 192 seconds. As an example of a typical output listing, a list of 50,000 components of 50 words per component will fill a magnetic tape. Processing requirements, neglecting computer time, come to 850 + 190 = 1040 seconds or 17 1/3 minutes. It must be pointed out that this 17 1/3-minute figure implies a high degree of program efficiency. A safer figure would probably be 30 minutes.

4.10 MAINTAINABILITY

The various computer programs and routines of LVCLS will be organized in such a way that errors of omission or commission within one routine will not interfere with successful operation of other routines and programs. This modular structure will improve maintainability by providing for corrections, additions, or deletions which may become necessary in the future. That is, a particular requirement or capability has a principal location which is referenced whenever it is used. Then whenever it is necessary to change a specific capability, only one section of the simulation need be changed to keep all uses of the particular capability current. Engineering working bulk data will be preprocessed and stored on disc files in related lists for efficient computational use. All basic data, simulation programs, and historical run results will be file protected.

Each simulation problem will draw data from the working bulk data and make necessary temporary additions and/or deletions. These temporary data will not affect the working file unless later processed through the simulation configuration management procedures and implemented by the proper change program or routine. An executive routine or simulation monitor will control the modular operation of the simulation. There will be provisions for insertion of future new simulation modules, new data, and changes tnereto. It is necessary to consider maintainability throughout the entire design and implementation and establish change procedures which will insure its inclusion after implementation and later modification.

4.11 EFFECTIVENESS OF THE SIMULATION

4.11.1 GENERAL

For a measure of effectiveness to be significant, it must be based on a measurable performance. Even relative value may be used for this purpose. In the present study, it was possible to produce numerical values of effectiveness using relative values. However, this value is only an expression of engineering judgement since it is dependent upon the value to the user and the cost to him of obtaining this result, which in the present case would be engineering judgements themselves. The various uses of the LVCLS are so closely interrelated that the cost of any one use is highly dependent upon the number and sequence of development of other uses. When the user has selected the set of outputs which will actually be provided and the detail design requirements for this set have been developed, then a more meaningful measure may be calculated and compared with other alternatives. The selection of the recommended order of development has, therefore, been made largely on the basis of the interdependencies between functions and between uses so that early useful results will become available from the expenditure of a minimum of effort. Also, sequential development is organized so that most of the preceding work will contribute directly to the next following computer program. The discussion which follows gives some insight into measures of effectiveness and how they were applied in guiding the selected order and evaluating the various portions of the proposed simulation system.

4.11.2 METHODOLOGY

4.11.2.1 Introduction

An analytical approach to cost effectiveness* is assumed necessary to the decision process that is used to control the development and use of the LVCLS at all stages, from conception through final implementation, operation, and improvement. During the design phase, analysis will provide guidelines for selection of attributes to be implemented and the appropriate order of implementation. During the implementation phase, the effectiveness/cost-effectiveness analysis will be an iterated comparison of current status with requirements, to be used in considerations of optimal progress available within discrete time intervals, depending upon the state already obtained and the resources available at such times.

4.11.2.2 The LVCLS Mission Definition

The mission of the LVCLS is stated in terms of:

a. Functional description (purpose).

b. System quantitative requirements.

*See Reference 21 for a recent Government/industry review of applications status.

The statements cover the end-item functions of the LVCLS (configuration performance, data processing, discrete simulation, dynamic simulation, analysis, output presentations, etc.) and the locations and operating constraints under which these are to take place. Since minimum requirements for some factors of effectiveness are not derivable from strictly objective criteria of value, relative cost effectiveness of alternative requirements may frequently be considered as a basis for requirement changes as system performance data materializes. Effectiveness/system effectiveness is essential to a viable rationale and economic and timely progress for such a general system as the LVCLS, but its methodology, though using tools often applied to the intrinsic theory of physical systems, is at best a quasi-science blending mathematical logic and subjective value arguments that are extrinsic to the physical theory of the system. It is highly dependent upon the provision of value judgements by the LVCLS sponsoring authorities.

4.11.2.3 Resource Identification

Resources represent constraints upon the LVCLS implementation program, principally in terms of:

a. Budget.
b. Manpower.
c. Contract requirements.
d. State-of-the-art limitations.

These factors can receive increasing clarification as formal ground rules for considering the alternatives to LVCLS refinements, in terms applicable to successive program achievements.

4.11.2.4 LVCLS System Description

The system description can achieve increasing depth of detail as the system is materialized. The description will progress through:

a. Identification of alternative configurations.
b. Configuration documentation, followed by system summary description.

The LVCLS will be considered in terms of the basic general configuration planned, along with some alternatives of physical or performance characteristics at various levels of detail. Alternatives which affect attributes of system value will furnish the focal points for emphasis in the description development. These include reliability, accuracy, consumption of time and labor, maintainability, availability, etc., and a special account of these for successive stages of system implementation to be considered. Generally, it will be necessary to consider attributes at one level of description, across the overall system, then revise the system as suboptimizations at lower levels of detail may bring out overriding considerations. Thus, it is necessary that the descriptive procedure will allow for a flow up and down levels of detail, in serving the overall optimization process.

4.11.2.5 Figures of Merit

Cost effectiveness optimization entails a selection of a combination of resources and attained effectiveness that is best by some figure of merit. The establishment of the figures of merit (FOM) is a judgement process, which should be based upon a consensus of experts and responsible agencies; it depends upon considerable use of engineering, economic, and operational judgement. The important feature, in addition to authoritative consensus, is that an adequate spectrum of system attributes are considered to achieve a well-balanced tradeoff. A problem of considerable significance is the establishment of system boundaries, i.e., <u>what</u> is being optimized, since some arbitrary cutoff of interdependent factors is always necessary.

The most general figures of merit are system effectiveness and system cost effectiveness. System effectiveness prediction and evaluation utilizes probabilistic rationalization. System effectiveness is developed in terms of the probability that the system can accomplish a mission directive at a random point in time (after operational implementation). System effectiveness is formulated as:

$$E = A \times D \times C$$

where

E = effectiveness.

A = availability.

D = dependability.

C = capability.

4.11.2.6 Availability

This is a measure of the simulator condition at the start of a desired period of usage. There are two states: either it is ready for use, or it is shut down. The latter must be interpreted broadly. It is obvious that the system may be inoperative because of a requirement for repair or maintenance. But it also may be unavailable for useful outputs because of key portions being tied up with receiving and processing new inputs, such as changes. The total time of unavailability is the sum of these two. The probability that the system will be inoperable at the start of a period of desired usage may be estimated to be the ratio of total time of unavailability to total clock time.

To put this concept in mathematical form, let

$\overline{A'}$ = Availability matrix (row vector).

a_1 = Probability of system being available.

a_2 = Probability of system not being available.

t_a = Expected period of time system is available.

t_i = Expected period of time system is tied up for new inputs.

t_r = Expected period of time system is shut down for repairs or maintenance.

Then

$$a_1 = \frac{t_a}{t_a + t_i + t_r}$$

$$a_2 = \frac{t_i + t_r}{t_a + t_i + t_r}$$

$$a_1 + a_2 = 1$$

$$\overline{A'} = [a_1 a_2]$$

4.11.2.7 Dependability

This is a measure of the simulation condition during a period of usage and its ability to perform as desired. It is related to reliability. For an availability vector of two states, there are four elements of the dependability matrix. Let:

$\overline{D}$ = Dependability matrix.

d_{11} = Probability that simulator is in operable condition at end of period, given that it was operable at the beginning.

d_{12} = Probability that simulator is not operable at end of period, given that it was operable at the beginning.

d_{21} = Probability that simulator is in operable condition at end of period, given that it was not operable at the beginning.

d_{22} = Probability that simulator is not operable at end of period, given that it was not operable at the beginning.

Then

$$\bar{D} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$$

where

$$d_{11} + d_{12} = 1$$

$$d_{21} + d_{22} = 1$$

Note that d_{11} is the conventional reliability of the system, and that all the d's may be calculated from known or estimated failure rates and repair times using the standard techniques developed for reliability theory. If the times for system shut down for receiving new inputs do not occur completely randomly, some modification of these techniques may be needed.

4.11.2.8 Capability

This is a measure of the ability of the simulation to achieve its desired objectives, given the system conditions. Essentially, this means the probability of completing a problem within the required accuracy. In this definition, the emphasis is on accuracy, since noncompletion may be treated as a failure and considered in the dependability matrix.

For a dynamic simulation, the concept of accuracy is a familiar one and the determination or estimation of the probability of its being within specified limits requires no new methods or techniques. The chosen limits may be on instantaneous errors, on terminal errors, on average error, on average absolute error, or on whatever function of error that may be selected.

With other usages of the simulator, the definition of accuracy may be somewhat modified. For example, in printing a list of components in a given subsystem, each character printed will be right, or it will be wrong - there will be no gradation. Being wrong includes failure to print something that should be. The system error can be defined as the percent of incorrectly printed characters. However, every error is not of the same importance. In printing the word "relay," any one of the five letters may be incorrect and the meaning will still be intelligible. But in printing the drawing number of that relay, an incorrect digit completely alters the meaning. To take care of this, it is recommended that weighting factors be used. Incorrect letters may be assigned weights less than unity because groups are still intelligible with one wrong, while numbers may be weighted heavily, because one incorrect digit may destroy the meaning of a whole group. The system error, then, would be the weighted percent error. Its distribution function will be found from experience with this or similar simulators.

Finally, for some usages there is no such thing as a single, isolated, or small error. For example, in plotting a sequence diagram, if the simulator says a particular relay closes when it really should be opening, then everything beyond that point that depends on that relay will be wrong. This type of error might better be treated as a failure and considered in the dependability matrix.

With accuracy defined and limits chosen, the probability of keeping within those limits may be determined or estimated by standard techniques and a capability matrix set up. There are two cases: the probability of keeping within limits when the system is operable, and the probability when the system is not operable. In all the foreseeable cases, the second of these will be zero, but, to keep the mathematics general and complete and to allow for extension of the theory, a matrix element will be included for this probability. Thus, let

$\overline{C}$ = Capability matrix (column vector).

c_1 = Probability of keeping within prescribed limits given that the system is operable at the end of the period of use.

c_2 = Probability of keeping within limits given that the system is not operable at the end = 0 for most cases.

Then

$$\overline{C} = \begin{bmatrix} c_1 \\ \\ c_2 \end{bmatrix}$$

The system effectiveness may now be expressed as the product of the three matrices for availability, dependability, and capability. Let

$\overline{E}$ = Effectiveness matrix.

Then

$$\overline{E} = \overline{A'}\,\overline{D}\,\overline{C}$$

In this discussion, the tacit assumptions have been made that the entire system operated or failed on a unit, and that there was a single effectiveness figure for it. This is not strictly true. It will be possible for some portions of the simulator to be operable and turning out useful results while other portions are undergoing repairs. The matrix formula for effectiveness is easily used for this situation by adding more elements to each of the component matrices. Each element of the expanded matrices is then defined as the appropriate probability applied to a selected subsystem, rather than to the entire system.

4.11.3 ANALYTICAL EXAMPLES

4.11.3.1 Relative Effectiveness

The concept of cost effectiveness as a per-unit value of a designated attribute is a many faceted one. The kinds of costs are several, such as capital investment or operating cost, direct or indirect, or some combination. The designated attribute may be achieving a specified level of system effectiveness, or it may be simply the ability to perform a given task. Clearly, the number of possible combinations of the several kinds of costs with the many kinds of attributes is large. Which is picked out as the cost effectiveness depends on the use to which it is to be put. It is not feasible to discuss all such combinations. This report will concern itself with two examples.

In Appendix D are listed 20 expected uses for this simulator, 35 different outputs, and 64 functions required to accomplish these. Suppose it is desired to determine the cost of achieving each use. One way is to assume the entire simulator will be built, use the implementation order proposed in Table D-20, and estimate the cost of adding each use. Examination of the table discloses that four functions must be constructed for use III, two additional ones for use II, ten more for use XIII, and so on. Of course, the several functions vary widely in their construction costs, but to simplify the illustration, it is assumed that each has the same cost, denoted as one unit. On this basis, the cost of each may be taken as given in Table 4-2.

Obviously, changing the order of implementation will change the unit cost of some of the uses and, therefore, the cost effectiveness. This may clearly be seen by taking a second example.

Table 4-2

Cost Per Use

Use	Unit Cost
III	4
II	2
VIII	8
XIII	2
XVII	3
XV	1
V	4
VII	2
VI	7
XIV	1
XVI	2
XVIII	1
X	2
XI	2
XII	5
I, IV, XX	2
XIX	5

Suppose that a management decision is made that use XX is the most important, and that the simulator will be built to accomplish this as a primary objective. On this basis, the cost of each use may be taken as given in Table 4-3. From the table, it is seen that 12 uses are assigned a zero unit cost, because they are automatically accomplished by a simulator built for use XX.

Naturally, the numerical values given in these examples will change when appropriate weights are given to the costs of the functions and the importance of the uses. When these are done, the type of cost effectiveness illustrated below may be determined.

Here, engineering judgements were used to assign relative values to each output and relative cost to development of each function required to obtain the output. Where the function was required for more than one output, its cost was prorated to each output. The value of each output is then compared with its total proportional cost to arrive at

Table 4-3

Cost Per Use

Use	Unit Cost
XX	48
I, IV, V, VI, VII, VIII, IX, X, XI, XII, XV, XVI, XVII, XVIII	0
XIII, XIV	1
II, III	2
XIX	5

a "figure of merit" for each output. The results of this analysis are illustrated in Table 4-4. Cumulative values versus cumulative costs are plotted in Figure 4-3. This figure correctly illustrates the fact that some portions of the simulation produce greater value than other portions. However, since all the development costs are prorated, it does not provide a basis for selecting portions of the simulation to be developed.

Table 4-4

Outputs in Order of Merit

Rank	Output	Figure of Merit	Relative Value	Relative Cost
1	AG	3.9	4.7	1.2
2	U	2.3	3.5	1.5
3	AK	2.0	1.8	0.9
4	V	1.9	4.7	2.5
5	AH	1.8	1.8	1.0
6	AI	1.8	1.8	1.0
7	AA	1.6	2.9	1.8
8	P	1.6	1.8	1.1
9	Q	1.6	1.8	1.1
10	AC	1.5	1.8	1.2
11	AJ	1.5	1.8	1.2
12	O	1.4	4.7	3.4
13	Z	1.3	2.4	1.8
14	AE	1.3	4.7	3.6

Table 4-4

Outputs in Order of Merit (Cont.)

Rank	Output	Figure of Merit	Relative Value	Relative Cost
15	AF	1.3	5.9	4.6
16	R	1.2	1.8	1.5
17	A	1.2	2.9	2.4
18	W, X	1.2	5.3	4.6
19	B	1.1	2.9	2.7
20	S	1.1	4.1	3.7
21	C	1.0	1.8	1.8
22	G	0.9	2.9	3.4
23	N	0.9	5.3	5.9
24	I	0.8	2.9	3.6
25	AD	0.8	5.3	6.4
26	M	0.7	1.2	1.8
27	AB	0.7	1.2	1.8
28	L	0.7	1.2	1.8
29	Y	0.7	2.4	3.3
30	H	0.7	2.9	4.4
31	D	0.6	1.8	3.0
32	F	0.6	1.8	3.0
33	E	0.5	1.8	3.4
34	K	0.4	1.2	3.1
35	T	0.3	3.5	11.0

4.11.3.2 Use Effectiveness

It is, perhaps, redundant to refer to use effectiveness as being in a third category. In reality, it is a type of cost effectiveness, but it does have an additional aspect. It is a comparison of the cost effectiveness of the simulator, or a portion of it, with the cost of accomplishing the same things by other means currently in use.

The comparison is meaningful when other methods do exist. The determination of use effectiveness is fairly straightforward in this case. But when other means do not exist, and the simulator achieves results that cannot be achieved in any other way, a

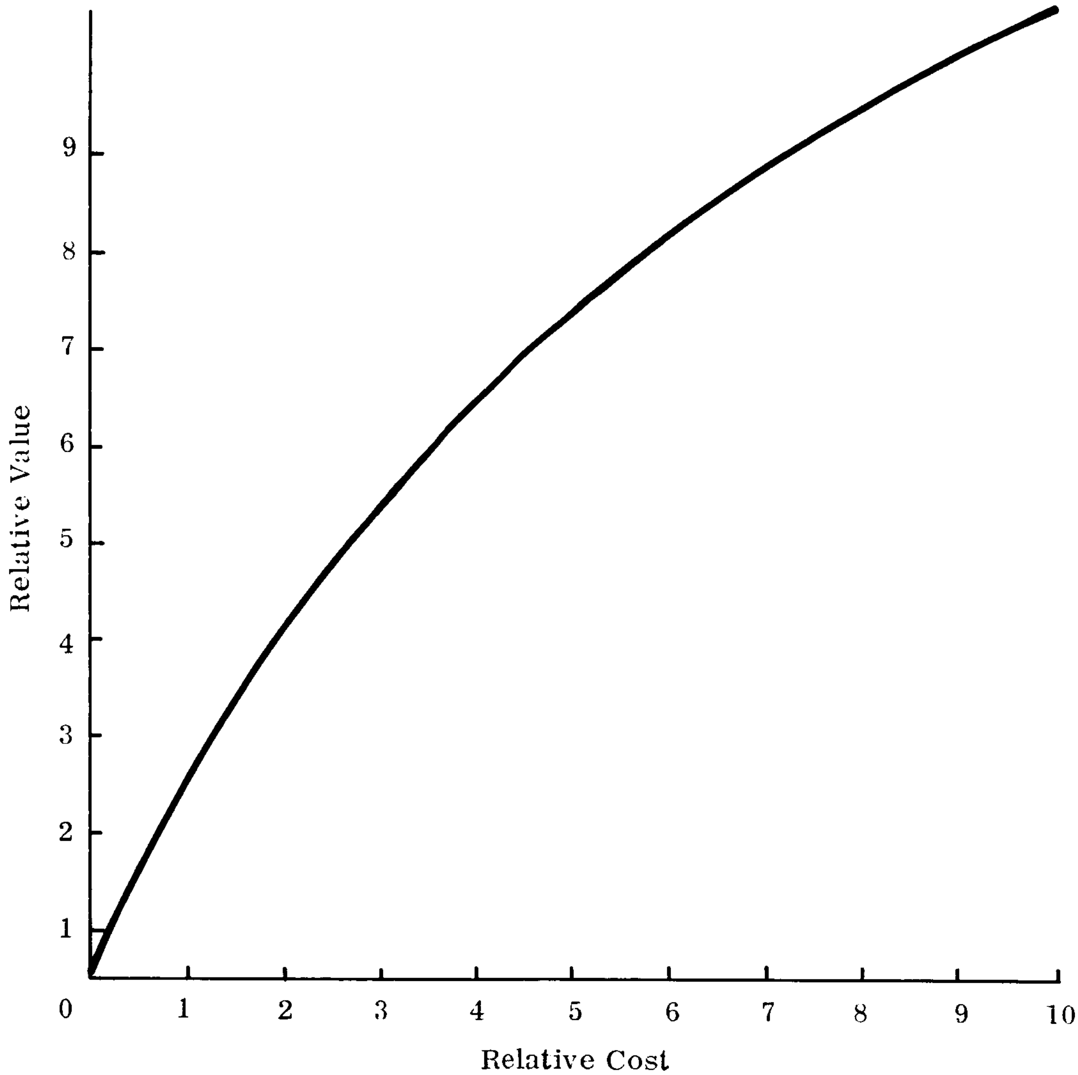


Figure 4-3. Simulator Outputs, Relative Value versus Relative Cost

modified definition must be used. Some estimate must be made of the value to the user of being able to achieve these particular results. This is a subjective matter depending closely on judgment. For example, if one use is selected as prime, it is possible to show the relative values of other outputs as illustrated in Table 4-4.

The concept of relative value versus relative cost described in the closing portion of paragraph 4.11.3.1 may be further extended to the use categories described in Table D-5. This was done and the outputs ranked by figure of merit in Table 4-5.

Table 4-5
Uses versus Order of Merit

Rank	Use	Figure of Merit	Relative Value	Relative Cost
1	II	2.5	4.7	1.9
2	XV	2.3	0.7	0.3
3	XIV	2.0	1.8	0.9
4	XVI	1.8	0.9	0.5
5	III	1.8	3.6	2.0
6	XIII	1.5	1.8	1.2
7	XVIII	1.4	2.9	2.0
8	VIII	1.3	2.5	1.9
9	VI	1.2	4.1	3.5
10	XIX	1.9	10.7	9.5
11	I	0.9	19.9	21.8
12	IV	0.9	19.9	21.8
13	XX	0.9	19.9	21.8
14	VII	0.9	2.0	2.3
15	V	0.8	0.6	0.8
16	X	0.8	1.2	1.6
17	IX	0.7	1.4	2.0
18	XI	0.6	0.9	1.6
19	XVII	0.3	0.2	0.6
20	XII	0.3	0.9	2.8

This table illustrates that while some of the intended uses have higher or lower figures of merit, a large proportion of the simulation value and cost is associated with four uses. They are:

Use I	-	Define the effect of a proposed change or operation of a selected portion of the launch vehicle and ground support system.
Use IV	-	Change the data temporarily to simulate a fault condition and follow its effects through a selected portion of the system.
Use XIX	-	Configuration Management documentation data center and control.
Use XX	-	Development of checkout and countdown procedures.

Again, in this case, the developmental costs are prorated and it is quite obvious that different weighting factors of value cost or proration could produce significantly different results. Furthermore, it should be emphasized that these results do not provide complete criteria for order or extent of implementation. Technical considerations provide the basis for the implementation order herein.

During the course of the design and implementation of this simulation, effectiveness studies will be conducted to establish sequential courses of action dependent on the then current state of progress, requirements, and available resources.

4.12 EDUCATION OF USERS

As used here, education is taken to be a two-way exchange of technical information. The simulation must provide output with information content and format which satisfies current needs. The user must understand both the capabilities and limitations of the simulation in order to have confidence in the results and to obtain optimum utilization. The demonstration of capability should include descriptive presentations, actual usable sample results, and discussion of technical limitations. While an informed user tends to be more receptive, this will not be a selling activity as such.

4.13 EVALUATION OF PREVIOUS WORK

There are a number of existing computer programs which have discrete or dynamic capabilities related to those required of LVCLS. These programs for the most part are either so general in capability as to be inefficient for LVCLS applications or so specialized and limited in the size of hardware systems which may be simulated as to be incapable of representing the launch vehicle at the component level. The successful application of these programs to sample problems during the study has however shown the practicality of LVCLS as a simulation system. The individual programs of LVCLS will undoubtedly use significant portions of existing computer programs, but these must be modified to fit and operate efficiently together in a complete simulation system.

4.14 SELECTION OF CAPABILITIES TO BE DEVELOPED

4.14.1 GENERAL

LVCLS is composed of three basic parts: the system description or engineering base data, the discrete simulation program, and the dynamic simulation program. The other functions recommended for implementation are principally in support of, use, or interpret basic capabilities. These additional functions maintain a current data base and edit output for specific purposes. The program functions and the engineering data required are closely interrelated to the point where a basic capability may be expanded in usefulness manyfold by the expenditure of small increments of additional programing effort. The ease with which a particular capability may be provided is sensitive to the selection of capabilities which have previously been provided. This dependency was largely the basis for our selection and ordering of capabilities recommended for design and implementation. This dependency influences the measure of effectiveness discussed in paragraph 4.11. There are examples of computer programs which have capabilities similar to each of those considered during this study. However, there is apparently no existing simulation system having the overall capability of LVCLS. Several techniques for solving dynamic problems developed during the study will provide accurate dynamic representation of vehicle systems on demand and at reduced cost. Should individual users have pressing requirements for particular outputs, the sequence of implementation may be changed without excessive penalty in total cost or time.

4.14.2 SOURCE DATA

Basic to the requirements of all the simulation uses is a complete engineering system description of all the components of the system, their type, location, function, status, interconnections, and other pertinent information. This inventory must be kept up to date as changes are made in the system.

Once this engineering data base has been established, the addition of comparatively few functions and administrative data inputs to the computer will permit the preparation of the various reports necessary for effective configuration management decisions. The problems that arise for this portion of the simulator do not differ in nature from those solved by existing simulations. They differ only in that LVCLS will be much larger.

4.14.3 DISCRETE SIMULATION

The very nature of the problem of simulating a system having a data base consisting of many elements calls for efficient methods of manipulating the data base to achieve the various objectives. For the discrete portion of the LVCLS, three methods of handling the data base were considered:

a. The iterated equation approach - This is the technique used in the ESE simulation, where it has been found quite feasible to handle a large, discrete data base. This has the advantage of being an operating technique, already programed and tested.

b. The matrix reduction method - In this method, the Boolean equations of the system are solved explicitly. While the method has not been programed on a trial basis, it apparently will require more memory and time than the other two.

c. The path tracing via adjacent nodes - This is a new method devised during the study. Preliminary estimates indicate that it will require fewer instructions than the other two and is to be preferred for the implemented simulation.

It is recommended that sample problems be programed on a computer to provide a sound basis for determining the relative advantages of the three methods at an early date.

4.14.4 DYNAMIC SIMULATION

In the study of dynamic simulation, new methods and new applications of established techniques were devised. All of these will be useful, and it is recommended that no single one be adapted to the exclusion of the others. Three phases were studied: the generation of system equations, the solution of these equations, and combinations of both.

For each change of state, caused by such events as the opening or closing of relays and switches, a new set of system equations must be generated from subsystem equations and solved. Several methods of achieving this dual result were considered, but none was found to be superior to all others for all situations. For different portions of the system, different combinations of the methods will give optimum results considering such factors as accuracy attainable, computer memory and instruction requirements, and speed of solution.

Methods studied for linear equations were:

a. Analytical operations on subsystem transfer functions to obtain the system transfer function in analytical form. This is a standard, straightforward method. Any method of solution may be used with it.

b. Numerical operations on subsystem transfer functions to obtain numerical values of the system transfer function, from which the analytical form is found by curve-fitting techniques. This is a new application of established methods and, in many cases, will require less computer capacity for the same problem as the analytical method.

c. Direct numerical inversion of Laplace transforms for the solution of equations. This is an adaptation of a technique found in the literature on orthogonal polynomials near the end of the contract period. The characteristics of this approach have not yet been evaluated.

d. The use of number series to both generate and solve the equations directly. This is an adaptation of a method from circuit theory, in which the dynamic responses of subsystems are stored as a sequence of numbers. Arithmetic operations on these yield, directly, a number series representing the dynamic response of the system. This holds promise of being a useful tool.

An important break-through was achieved with a method, believed to be new, of numerically solving differential equations. It is applicable to any equation, linear or non-linear, whose form is known analytically and is differentiable. First trials on sample problems have been very encouraging.

4.15 DEVELOPMENT OF RECOMMENDED ORDER OF IMPLEMENTATION

4.15.1 GENERAL

The recommended order of implementation developed during this study is given in Table 3-1 under the preceding summary of results and recommendations. The analytic rationale which resulted in the ordering and the major problem areas which influenced decisions are discussed in the following paragraphs.

4.15.2 RATIONALE

The order of implementation recommended is dependent on the following considerations:

a. Functional interrelationships.

b. Problem areas.

c. Processing equipment requirements.

d. Customer considerations.

Analysis of these factors has resulted in the tradeoff elements that were used in developing the implementation plan.

4.15.3 FUNCTIONAL INTERRELATIONSHIPS

Analysis of the requirements for each use and the extent to which a requirement in one shows up in all others has led to the conclusion that discrete simulation is indigenous to the largest number of uses. Hence, implementation in this area first will provide essential elements for a large number of other areas. This building-block idea is considered a desirable element in the ordering. In fact, modular development, organization, and utilization is basic to all of LVCLS. Proper modularity aids future changes, allows simultaneous unrelated usage, and reduces computer requirements.

4.15.4 PROBLEM AREAS

The problem areas that have been identified and examined in relation to uses and outputs are also considered to be essential factors in the selection of implementation order. These areas are discussed in detail in Section 5 and the extent to which they affect uses and outputs are shown in Tables 5-1 and 5-2.

The characteristics of each problem area and how they affect each use and output have been translated in terms of total difficulty facing each use implementation. One consideration in order selection is to implement first those uses which provide immediate results and are presented with the least problem difficulty. This allows favorable development timing so that the broad scope of problems can be attacked and resolved concurrently, thus permitting the implementation sequence to proceed at the best possible rate.

4.15.5 PROCESSING EQUIPMENT REQUIREMENTS

Processing equipment requirement specifications are based on the functional requirements of each use, the problem areas, and equipment availability. It is considered desirable, particularly in meeting the early implementation milestones, to minimize the need for extensive inventive development in the equipment area. It is planned that any unusual equipment needs will be solved concurrently with the main program.

4.15.6 USER CONSIDERATIONS

The recommended order has taken into account our estimate of the user's present and projected needs. It is expected that, as the program proceeds and develops, user needs will change out of necessity. Thus, it is essential that accurate forecasts be made and updated at periodic intervals.

SECTION 5

IMPLEMENTATION PROBLEMS

5.1 GENERAL

Problems identified during this study are principally related to the data base, simulation programs, and computer hardware. They result largely from the exceptionally large size of the data base, the many iterations required for discrete simulation, the many calculations required for solution of dynamic system equations, and input/output limitations of computer hardware. Almost without exception, similar problems for each program module have been successfully solved on a smaller scale but have not been combined into a general simulation of a complete launch vehicle system. The size of the simulation system and time constraints for utilization dictate a large computer system and the use of machine-dependent programing to improve the running efficiency during operational usage. However, the individual program modules may be developed on existing computer hardware such as the IBM 7044 digital computer with disc files and combined at a later date under a simulation monitor developed along with the modules. To operate the complete simulation system on a continuous demand basis will require sophisticated computer systems as discussed in paragraphs 4.5 and 3.2.

5.2 PROBLEM AREAS

During the study, a number of specific problem areas were identified. Solutions have already been found for some of these problems. The remaining problems appear to be capable of solution but will require work during the implementation phase. The latter group is listed below:

a. Data base and storage and retrieval system development.
   - Time and manpower requirements.
   - Read in and update functions needed to insure ready availability and correctness of data.
   - Priority ratings between engineering functional data and configuration accounting data.

b. Compilation to process connection statements into logic statements.
   - Elimination of process loops.
   - Multiple storage requirements for connection statement and logic statement approach.

c. Dynamic computation.
   - Computational accuracy.
   - Computational stability.
   - Processing time.

d. Algorithms for determining test points required for fault detection and isolation procedures.

e. Information formatting requirements for plotter.

f. Identification of equipments leading to questionable operation.

g. Supervisory program to control computer outputs.

h. Identifying and listing unstable equipments.

5.3 USES AND OUTPUTS VERSUS PROBLEM AREAS

Tables 5-1 (Use) and 5-2 (Outputs) summarize the extent to which the problem areas identified above affect the proposed uses and outputs.

Table 5-1
Problem Area Summary for Potential Use

Potential Use	a	b	c	d	e	f	g	h
I	x	x	x	x	x	x	x	x
II	x						x	
III	x						x	
IV	x	x	x	x	x	x	x	x
V	x	x	x				x	
VI	x	x	x				x	
VII	x	x			x		x	
VIII	x	x			x		x	
IX	x	x				x	x	
X	x	x			x		x	
XI	x	x			x		x	
XII	x	x					x	
XIII	x	x	x				x	
XIV	x						x	
XV	x	x					x	
XVI	x	x					x	

Table 5-1

Problem Area Summary for Potential Use (Cont.)

Potential Use	a	b	c	d	e	f	g	h
XVII	x	x					x	
XVIII	x	x					x	
XIX	x						x	
XX	x	x	x	x	x	x	x	x

Table 5-2

Problem Area Summary for Potential Outputs

Potential Output	a	b	c	d	e	f	g	h
A	x	x			x		x	
B	x	x					x	
C	x	x					x	
D	x	x					x	
E	x	x			x		x	
F	x						x	
G	x	x			x		x	
H	x	x				x	x	
I	x	x			x	x	x	
K	x	x					x	
L	x	x	x				x	
M	x	x	x				x	
N	x	x	x				x	x
O	x	x			x		x	
P	x						x	
Q	x						x	
R	x	x					x	
S	x	x			x		x	
T	x	x					x	
U	x	x					x	
V	x	x					x	
W	x	x					x	
X	x	x					x	

Table 5-2

Problem Area Summary for Potential Outputs (Cont.)

Potential Output	a	b	c	d	e	f	g	h
Y	x						x	
Z	x						x	
AA	x						x	
AB	x						x	
AC	x						x	
AD	x	x		x			x	
AE	x						x	
AF	x	x	x				x	
AG	x						x	
AH	x						x	
AI	x						x	
AJ	x	x	x				x	
AK	x						x	

SECTION 6

RECOMMENDED STATEMENT OF WORK FOR IMPLEMENTATION

6.1 INTRODUCTION

The following statement of work describes the design and implementation of the Launch Vehicle Component Level Simulation. The scope of this work is based on the system requirements baseline established during the Analytical Study Contract NAS-8 20060. During this implementation, the actual programing of computers to perform the functions developed during the analytical study will be accomplished in their recommended sequence. There will be the development of a data base adequate to support the simulation and prove out its functions. This data will cover those systems supplied by MSFC, but should be at least one complete stage and its support equipment, such as the S-IVB. The product of this work will be an operational simulation system with a functional data base together with the operational support procedures necessary to utilize the simulation, expand the data base, and maintain both in an accurate and current condition.

6.2 SCOPE OF WORK

6.2.1 GENERAL

This initial implementation will include three basic functional capabilities together with the supporting routines to provide the capability for maintaining a current data base and the capability to provide the specific initial outputs detailed under Phase I. That is:

a. Design and implement configuration management accounting to establish an engineering data base and system description.

b. Design and implement a discrete simulation to provide logical and functional analysis.

c. Design and implement a dynamic simulation to provide dynamic analysis of the overall system.

d. Design and implement those supporting routines detailed under Phase I to maintain a current and specified data base.

e. Design and implement those supporting routines detailed under Phase I to provide the required initial outputs.

The work shall be carried out in three phases as described in the following paragraphs.

6.2.2 PHASE I

During this phase, a Functional Design Specification will be developed for the individual computer programs to provide the following outputs (the reference Roman numerals, capital letters, and small letters are the same as those used in the Functional Flow and Requirements Report of 15 September 1965 and Appendix D):

a. Configuration Management Documentation Data Central and Control (XIX)

- End-Item Approved Configuration Indices (Y)
- Approved ECP End-Item Indices (Z)
- End-Item Quantitative Requirements Schedule (AA)
- End-Item Modification Status (AB)
- Spares Status (AC)

b. Follow Signals Through a Selected Portion of the Launch Vehicle on a Discrete Basis (VII)

- Listing of Sequence of Operations by Time (B)
- Listing of Component Status Change (C)
- Listing for Comparison Run (K)
- Function Sequence Chart (A)

c. Perform Transient Analysis of a Selected Portion of the Launch Vehicle and Ground Support Systems (VI)

- Listing of Equipments Unstable in Operation (N)
- Plot of Transient Response (O)
- Listing of Transient Response (AF)

d. Keep Track of Approved Change Orders, Drawing Changes, and Hardware Changes Made in the Simulation Data File and the Resultant Configuration (II)

- Listing of Changes Which Affect Simulation Operation (AG)

e. Insert Approved Changes Into Central Data File (III)

- Listing of New Approved Permanent Data Being Entered (AH)
- Listing of New Approved Permanent Changes Being Entered (AI)

f. Calculate Expected Times for Events of the Sequential Operation of a Selected Portion of the Launch Vehicle and Ground Support Systems (V)

- Listing of Delay Times for Selected System Portion (L)
- Listing for Comparison of Delay Times (M)

g. Relate the Simulation to the Advanced System Schematics Through Actual Circuit Connection Statements so that Components may be Identified With Terminals, Racks, Equipment Numbers, etc., as Given on Panel Schematics, Interconnection Diagrams, and Advanced System Schematics (VIII)

- List of Equipments by Panel (P)
- List of Equipments by Drawing Number (Q)
- Listing of Equipments by Function (R)
- Panel Schematic (S_1)
- Drawing Schematic (S_2)
- Function Schematic (S_3)

h. Allow a User to Set Up Conditions Which Identify a Portion of a Proposed Actual Checkout or Countdown Sequence (XIII)

- Listing of Equipments Involved Within Specified Bounds (AJ)

i. Define and Keep Track of Equipments Which Have Been Activated and Maintain a Record for Output (XV)

- Listing of Equipments Activated With Time or Number of Activations (U)

j. Compare Resulting Sequences With Desired Ones For Checkout or Countdown Activities (XVII)

- Listing for Comparison Run (K)

The detailed functions to provide the above outputs will be organized into subprograms and into an overall simulation program matrix. The specification and design activities will occur as detailed in Phases II and III.

6.2.3 PHASE II, SUBPHASE A

The tasks performed in this subphase will pertain to each of the subprograms which make up the operational, utility, and test support software subsystems.

- The individual programs shall be designed in detail.
- All interfaces shall be identified and designed in detail.
- Test plans for the individual programs shall be prepared.
- Input/output specifications and operating procedures shall be detailed.
- Special coding rules and input/output subprograms of the operational system shall be defined.
- Individual computer program Design Specifications Baseline shall be developed.

6.2.4 PHASE II, SUBPHASE B

The task performed in this subphase shall include:

- Coding, assembly, checkout, and documentation of the individual computer subprograms.
- Implementing the data base.
- Acceptance testing of the subsystems.
- Establishing the final System and Individual Computer Program Design Specifications.

6.2.5 PHASE III

During this phase, the individual programs and data base shall be integrated into the overall Launch Vehicle Component Level Simulation System.

- The system shall be brought into an operational state.
- Final detail documentation shall be completed.
- User guides and user education material shall be completed.
- Other user education activities shall be performed.
- Detail plans for implementing additional areas of application shall be made.
- Recommendations for expanded capability shall be formulated.

6.3 REPORTS REQUIREMENTS

During the implementation, a number of reports shall be prepared and submitted for NASA review. Listed by Phase, they are:

a. Phase I

- Launch Vehicle Component Level Simulation.
  Individual Computer Program Functional Design Specifications.

b. Phase II

- Launch Vehicle Component Level Simulation.
  Individual Computer Program Detail Design Specification, Volume I.
- Launch Vehicle Component Level Simulation.
  Individual Computer Program Design Specification, Volume II.
- Preliminary User's Manual.
- Preliminary Launch Vehicle Component Level Simulation System Description.

c. Phase III

- Suggested Expanded Capability Plan.
- Final User's Manual.
- Test Specifications Manual.
- Launch Vehicle Component Level Simulation System Description.

APPENDIX A

DYNAMIC SIMULATION AT THE COMPONENT LEVEL

A1 INTRODUCTION

Usually when considering the problem of dynamic simulation, the system configuration to be simulated is modeled in terms of a well-defined set of equations that describe the system response to prescribed stimuli. The problem is then a question of solving these equations. The launch vehicle component level simulation goes another step. Here it is desired to be able to automatically build up the model equations for vehicle subsystems as well as obtain a solution to these equations. The user will want to be able to simulate different combinations of a variety of configurations. Some may want a somewhat refined simulation of a relatively small composite of components. Others may want a more cursory simulation of a large network of components. Perhaps a user will wish to specify as little as the input stimuli and output nodes at which the response is to be computed, having the simulation construct the pertinent program system from the program library and component level data base.

Examination of the major functional systems for the Saturn V launch vehicle and GSE (Table A-1) gives an indication of the breadth and depth of the problem. It is clear that the user generally will not need nor want a sophisticated simulation involving in-depth representations of several major functional systems. For instance, the user interested in studying the structures problems associated with wind loading will prefer to develop a loading profile using a representative trajectory generator rather than use a trajectory generator that employs a sophisticated model of the propulsion system and flight control system. In-depth simulation of a propulsion system will include a rather cursory representation of the flight control system and vice-versa.

There are a number of simulation programs available for application to mechanical and electrical problems. Many of these programs have been written for special problems and are often difficult to use for anything else. Others have been written to handle very general problems; however, the price paid is excessive memory requirements and excessive running time. Ideally, the flexibility of the general program along with the efficiency of the special purpose program is desired. The development of a user-oriented dynamic simulation monitor system seems to provide a means to obtain the latter.

Table A-1

Major Functional Systems

S-IC	S-II	S-IVB	IU
Saturn V Launch Vehicle			
Structures	Structures	Structures	Structures
Propulsion	Propulsion	Propulsion	
Flight Control	Flight Control	Flight Control	Guidance and Control
Electrical System	Auxiliary Power System	Electrical System	Electrical Power
Communication and Instrumentation	Instrumentation and Telemetry	Communication	Communication
Range Safety	Range Safety	Range Safety	Range Safety
Separation	Separation	Separation	Separation
Environmental Control	Environmental Control	Environmental Control	Environmental Control
Emergency Detection	Emergency Detection		Emergency Detection

Saturn V GSE	
S-IC, S-II, S-IVB	IU
Auxiliary Equipment	Monitoring and Control Equipment
Handling Equipment	Systems Integration Equipment
Transporting Equipment	Network Distribution and Control Equipment
Environmental Control or Protective Equipment	GETS
Service Equipment	Recording Group
Maintenance Equipment	OAT
Training Equipment	SIS
Other GSE	Simulation Equipment
	Peripheral Equipment
	Countdown Clock
	Signal Conditioning Equipment

The dynamic simulation system will contain a compiler that accepts a user-oriented FORTRAN-like language. It will utilize a large program library and launch vehicle data base to compile dynamic simulations as called for by the user. Only those programs necessary for the particular simulation at hand will be used. The program library will contain programs that simulate common functions but at varying degrees of sophistication. There will be provisions to add and delete programs to the library. Once a simulation program has been compiled it may be kept for later use. The output from one simulation may be stored for use in conjunction with another simulation at a later time. The capability to be used in a time-sharing mode will be maintained throughout the development of the system. Provisions will be made to include existing simulation programs in the library (such as DYNASAR, THTB, etc.). The strength of these existing simulations will be enhanced by the ability to write programs under the dynamic simulation monitor that automatically will prepare the input data from the launch vehicle data base.

There undoubtedly will be several techniques for structuring system models available in the Dynamic Simulation System. The suitability of several approaches has been investigated and will be discussed in depth. The relative merits of these and other techniques will be resolved by the user needs. Generally, highly nonlinear functions will be handled best with differential equations. The transfer function or impulse response will offer a number of advantages for more linear functions.

An over-all conceptual dynamic simulation system layout is shown in Figure A-1. The system is divided into multiple phases that need not occupy core memory simultaneously. The three phases in Figure A-1 can be further divided if necessary.

Phase I is essentially the simulation definition phase. A user's simulation specification data are the input. These specifications are interpreted by the Phase I program and a simulation description in terms of programs and data base required is prepared. The program library directory and master data base directory are searched to determine if the required programs and required data are available in the system. The programs not available in the system library are obtained from the appropriate input medium, compiled, and added to the system library. The data not available in the system data base also is obtained from the appropriate input medium and added to the master data base. A simulation control list and a simulation data base list are prepared to be used by Phase II.

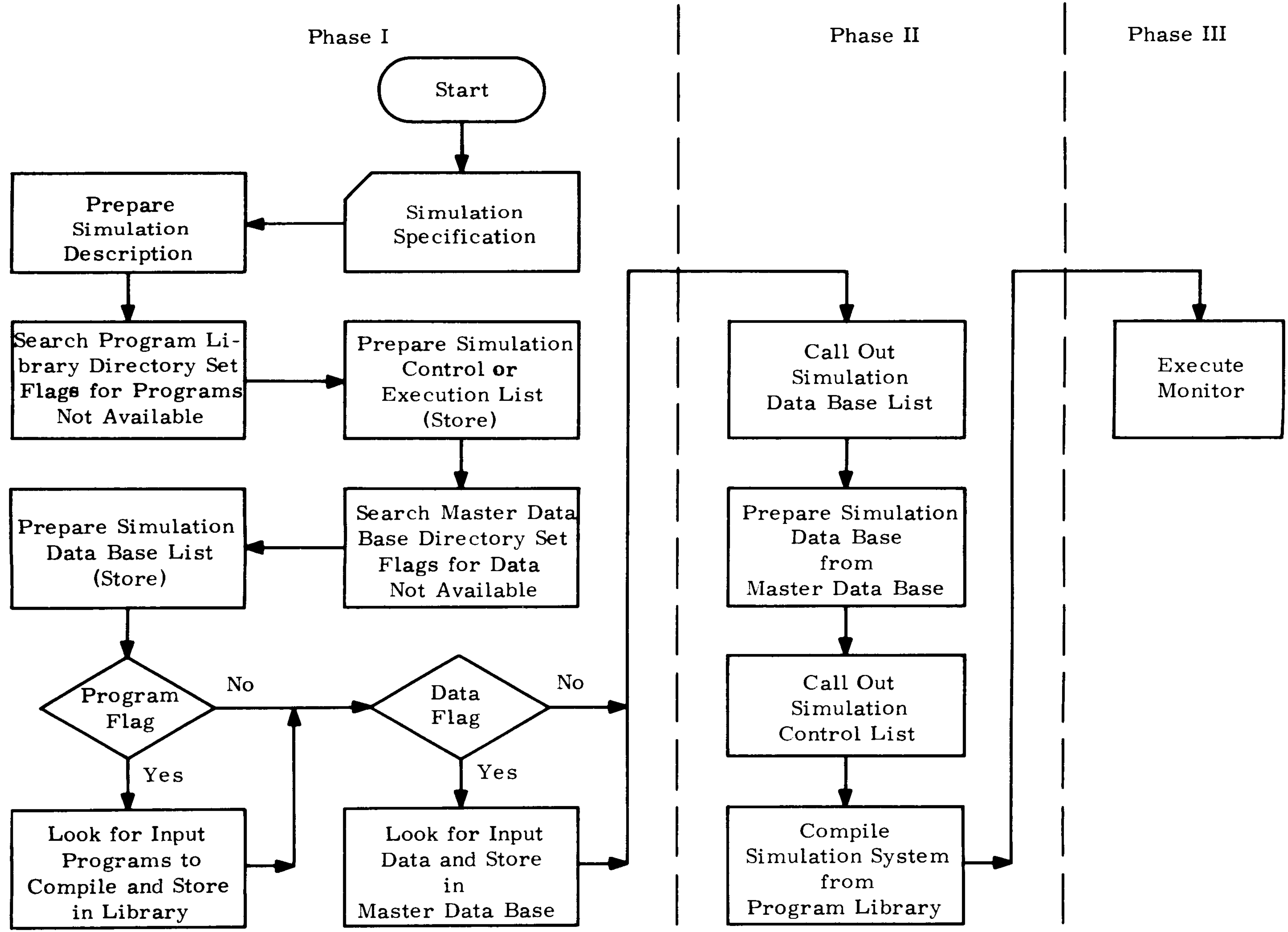


Figure A-1. Dynamic Simulation System Layout

Phase II uses the simulation data base list to extract data from the master data base file and prepare a simulation data base. The simulation control list is used to compile the simulation system from the program library. The latter includes the Execution Monitor Program.

Phase III is the execution monitor that loads the simulation system and supervises the simulation execution.

Implementation of the system will most logically proceed as follows:

(1) Implement Phase II and Phase III for a selected set of simulation capabilities.

(2) In parallel with Step 1 initiate the Phase I development. This will include the user language, storage and retrieval, library structure, etc.

(3) Implementation of Phases I, II, and III for a broad selection of simulation capabilities.

The system can have a high utility initially, but be capable of growth and expansion. Step 1 is relatively straightforward. Step 2 will require in-depth knowledge of the user's simulation interest. Step 3 is logically an initial implementation for a selected set of capabilities followed by a continuing effort as needed.

A2 TRANSITION MATRIX METHOD

A2.1 INTRODUCTION

The dynamic simulation of a system involves the solution of sets of simultaneous differential equations. These are in turn derived from sets of subsystem equations, which are made up of sets of component equations. When all the equations are linear, or can be approximated by linear equations, a convenient way of expressing them is as transfer functions (i.e., Laplace transforms). The system transfer function is obtained from algebraic operations on subsystem transfer functions, and they from operations on component transfer functions. One way of performing these operations in a systematic way is by reducing transition matrices as described in this section.

A2.2 NOMENCLATURE

In presenting the transition matrix method for dynamic simulation the following nomenclature will be used.

Network	A network is the total collection of systems to be considered.
System	A system is any collection of subsystems, $S_1, S_2, \ldots, S_i$.
Subsystem	A subsystem is that portion of the network lying entirely within the boundary defined by relay contacts, or their representative nodes, $x_1, x_2, \ldots, x_n$.
Component	Any element of the network lying within the domain bounded by the subsystem. In topological terms this would be the collection of edges defined by the subsystem network bounded by $x_1, x_2, \ldots, x_n$. The nodes at this level will be designated as internal nodes $y_1, y_2, \ldots, y_n$.
Nodes	The dependent or independent variable of the mathematical model representing the network. Nodes representing the system output may be designated as external nodes (x_i's) and nodes representing the subsystem configuration as internal nodes (y_i's).
T_n	Network transition matrix.
T_s	System transition matrix.
T_e	Transition matrix generating event.
T_{ss}	Subsystem transition matrix.
Ω	Absolute time vector.
E	Event marker.
V_n	Network state vector.
V_s	System state vector.
V_{ss}	Subsystem state vector.
A_n	System switching algorithm.
P_{ij}	Equivalent path through subsystem from node x_i to x_j or from node y_i to y_j.
x_c	Cross connection node or artificial external node.
x_i	Generic system input node or external node.
x_j	Generic system output node or external output node.
y_i	Generic subsystem input node or internal input node.
y_j	Generic subsystem output node or internal output node.

$y_i(0) \equiv y_i(IC)$	Initial condition at beginning of event interval.
$y_j(TC)$	Terminal condition at end of event interval.
$\hat{z}$	Serial representation of z.
τ	Time constant.
t_{ij}	Subsystem edge transfer function.
$t_{i,i}$	Edge transfer function from initial value node $x_{i'}$ (0) to x_i.
T_i, T_j	Some arbitrary system time.
α_{ij}	Boolean variable identifying node connection.

A2.3 DYNAMIC SIMULATION AT THE COMPONENT LEVEL

The mathematical modeling of a network interlaced with relays will be facilitated by a set of definitions.

a. Subsystem

A subsystem (see Figure A-2) is defined as that portion of the network bounded by relays and their representative nodes; x_i, x_k, and x_j:

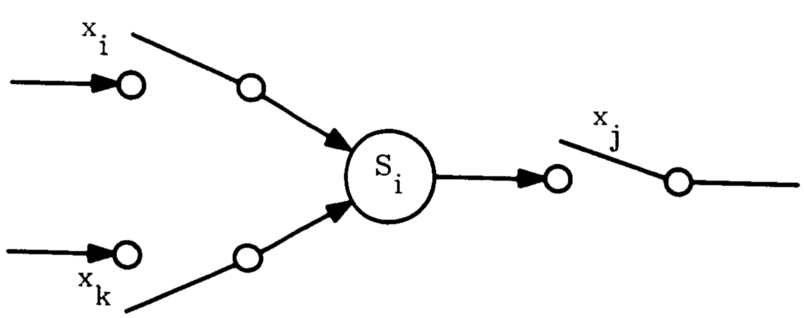


Figure A-2. Subsystem Configuration

The nodes x_i, x_j, etc., are external nodes to the subsystem. The total collection of x_i's defines the system state vector V_S.

b. System Component

A system component (see Figure A-3) is any element, or edge, bounded by the subsystem:

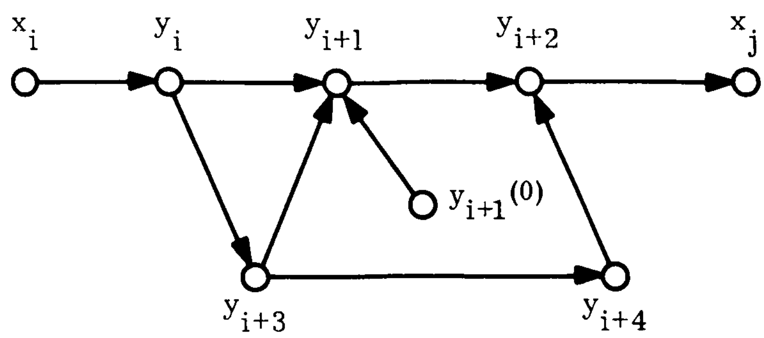


Figure A-3. Component Configuration

The nodes y_i, y_{i+1}, etc., are internal nodes of the subsystem. The total collection of y_i's defines the subsystem state vector V_{ss}.

c. System

A system (see Figure A-4) is any collection of subsystems S_1, S_2, S_3, S_4:

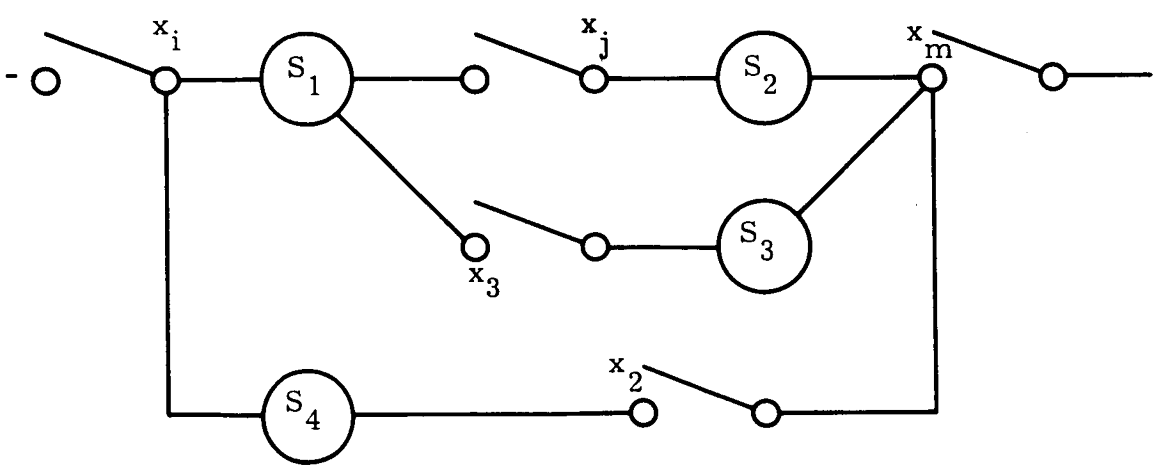


Figure A-4. System Configuration

d. Network

A network is defined as the total collection of systems. This definition is provided in order to recognize that not all portions of a network operate simultaneously.

A2.4 DYNAMIC SEQUENCING

The dynamic simulation of the previous type of network will be considered for the state dependent case. That is for the case where the position of the relay contact, open or closed, is a function of the vectors V_s and V_{ss}.

Consider some arbitrary collection of subsystems whose transfer functions are t_{ij}:

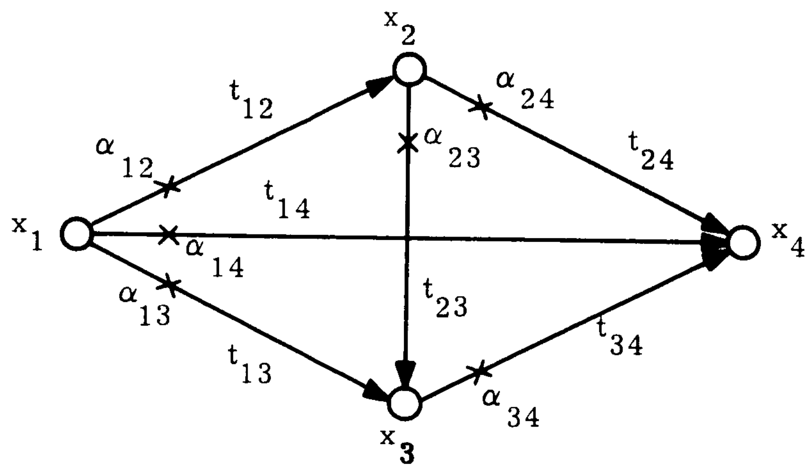


Figure A-5. Generic Sequential System

where t_{ij} represents the subsystem equivalent edge factor and α_{ij} is the contact element admitting the signal to t_{ij}. Figure A-5 yields a system state transition matrix:

$$T = \begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 1 & & & \\ x_2 & t_{12} & & & \\ x_3 & t_{13} & t_{23} & & \\ x_4 & t_{14} & t_{24} & t_{34} & \end{array} \qquad \text{(A-1)}$$

In Equation A-1 zero intersections are left blank. Input nodes appear at the top. Output nodes appear at the side.

A2.5 SUBSYSTEM EQUIVALENT EDGE

The subsystem edge factor, t_{ij} in T_{ss}, may be derived from a subsystem that is composed of a large collection of edges. Consider some arbitrary subsystem:

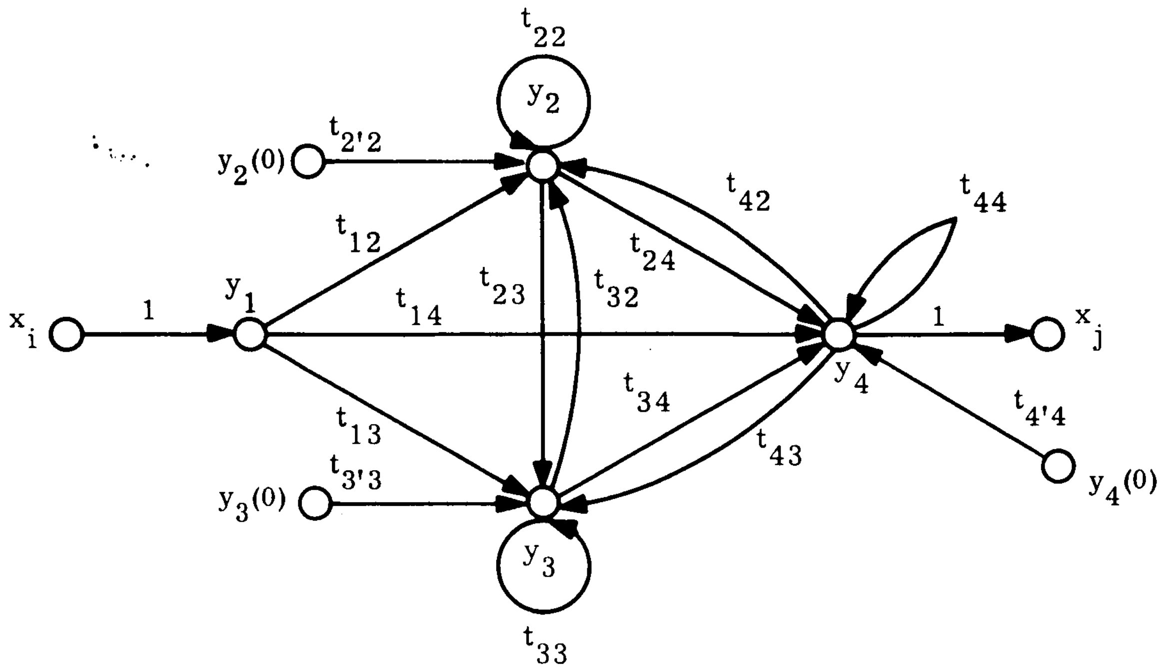


Figure A-6. Generic Subsystem

The reduction of Figure A-6 to generate equivalent edges is accomplished by a subsystem matrix that has the following form:

$$T_{ss} = T_o = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ \hline y_1 & 1 & & & \\ y_2 & t_{12} & t_{22} & t_{32} & t_{42} \\ y_3 & t_{13} & t_{23} & t_{33} & t_{43} \\ y_4 & t_{14} & t_{24} & t_{34} & t_{44} \end{array} \qquad \text{(A-2)}$$

which satisfies the following set of equations

$$[y_i] = T_o[y_i] \ . \qquad \text{(A-3)}$$

The matrix T_o is now reduced to the form

$$T_{ss} = T_{n-1} = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ \hline y_1 & 1 & & & \\ y_2 & t_{12}^{n-1} & & & \\ y_3 & t_{13}^{n-1} & & & \\ y_4 & t_{14}^{n-1} & & & \end{array} \qquad \text{(A-4)}$$

where the reduction algorithms for T_o are explained in Addendum A1.

The same process is applicable to the system matrix, T_s.

The intersections of the reduced matrix, t_{ij}^{n-1}, are designated as the equivalent path products through the system or between the subsystem nodes, P_{ij}.

A2.6 SWITCHING ALGORITHM

The state of the contact elements, α_{ij}, which establish the connections between the elements of V_s and V_{ss}, are controlled by algorithms. These algorithms may be related to the system state vector V_s, the subsystem state vector V_{ss}, to the time vector Ω, or to an event vector E. Several examples may be considered:

a. Event Switching (Figure A-7)

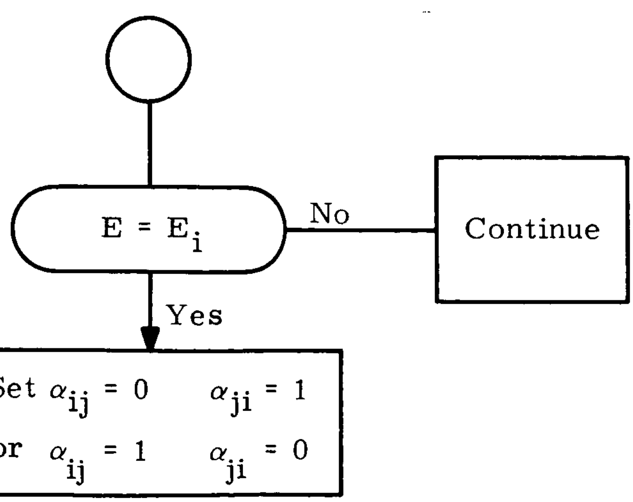


Figure A-7. Event Switching

b. Relay Switching with Delays (Figure A-8)

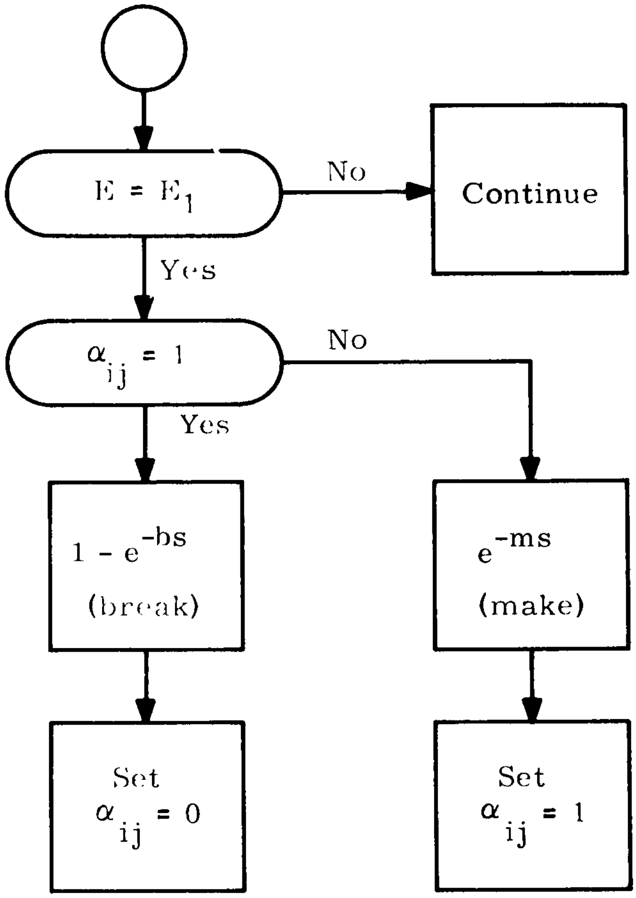


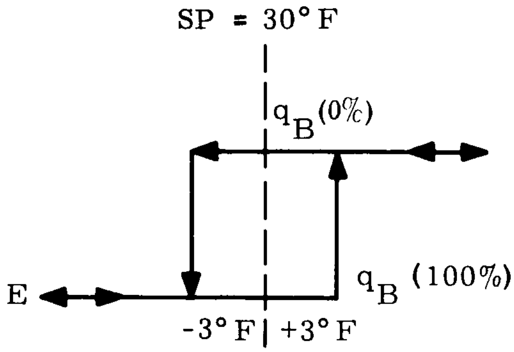
Figure A-8. Delayed Switching

c. Time Switching

Define event E_i as some time T_i.

d. State Vector Switching

Switching algorithms based on threshold levels of the state vectors V_s and V_{ss}, constitute the most important element or attribute of dynamic sequencing and provide the methodology of combining large systems connected by relays. As an example of a switching algorithm which is state dependent, the situation exhibited by a house thermostat control may be considered (see Figures A-9 and A-12). Here the subsystem state variable governing the switching operation is the room temperature θ_R. The thermostat is also considered to have a hysteresis of $\pm 3°$ F.



$$\alpha_{15} \equiv \alpha_{14} \qquad \alpha_{1'5} \equiv \alpha_{1'4}$$

$$\alpha_{15} = \alpha_{14} = 1 \qquad (q_B = 100\%)$$

$$\alpha_{1'5} = \alpha_{1'4} = 0 \qquad (q_B = 0\%)$$

Either

$$\alpha_{14} = 1 \qquad \alpha_{1'4} = 0$$

or

$$\alpha_{14} = 0 \qquad \alpha_{1'4} = 1$$

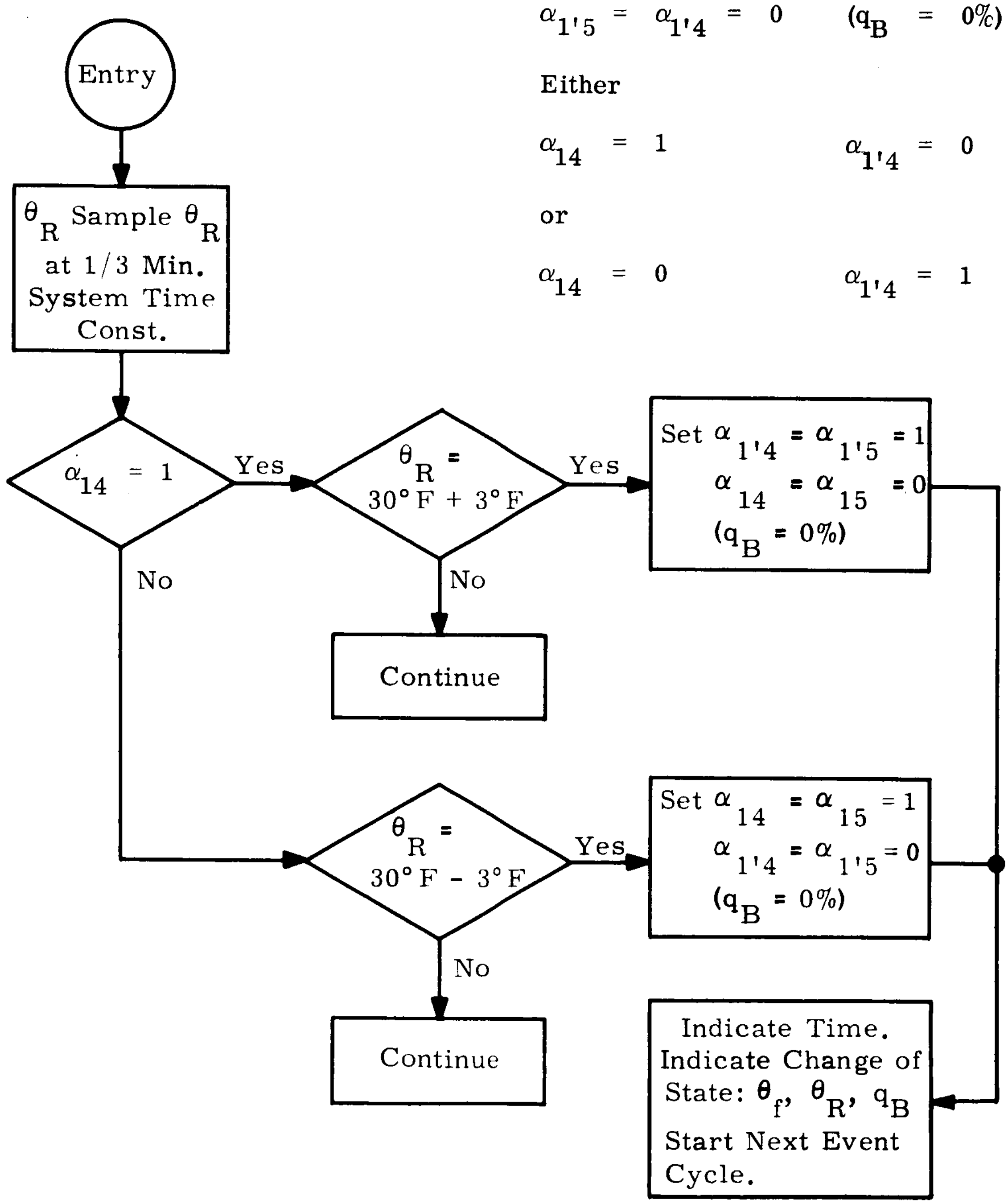


Figure A-9. Generic Switching Algorithm: Thermostat

A brief discussion of the thermostat action may be worthwhile. Here the subsystem requires two input nodes (x_i): q_B (100%) and q_B (0%). The subsystem has two paths leaving these nodes. Hence, there are four α's defining this contact element α_{14}, α_{15}, $\alpha_{1'4}$, and $\alpha_{1'5}$. An $\alpha_{14} = \alpha_{15} = 1$ represents the fact that the thermostat is closed and calling for heat (q_B = 100%).

An $\alpha_{14} = \alpha_{15} = 0$ and $\alpha_{1'4} = \alpha_{1'5} = 1$ represents the fact that the thermostat is open and the furnace turned off. If the state variable θ_R is less than 27° F, i.e., 57° F room temperature or 27° F above ambient, with the contact closed, the furnace turns on and heats up the room. As θ_R exceeds 30° F, the thermostat set point, the thermostat tries to open, but the temperature has to increase 3° F more to carry it through the hysteresis loop. At 33° F the thermostat opens and shuts off the furnace. As θ_R falls the action is repeated in the reverse direction.

The above example is presented in detail because it demonstrates an important aspect of the system modeling rule that must be followed in this methodology; i.e., that all states of the system must at all times be accounted for. Hence, in the thermostat problem, q_B (heat input) has two separate states: q_B = 100% (on) and q_B = 0% (off). This point will be illustrated in a subsequent example.

A2.7 NONLINEAR SYSTEMS

The above thermostat switching action introduces points of discontinuity into the dynamic response of the system and consequently is termed a nonlinear device. The system response, however, is linear within the domain defined by the respective switching states.

The same principle must be applied to systems described by a set of nonlinear differential equations. For such systems the representation suggested here is that the system be piece-wise linearized into as many sections as are necessary to give satisfactory results. This means that as many additional nodes, or states, must be introduced into the systems as required by the additional piece-wise regions introduced in the model. Switching algorithms similar to those discussed above must be issued to take the system from one section to the next.

A2.8 SYSTEM FLOW CHART

The mathematical structure relating the elements (nodes) of the system state vector V_s and the subsystem state vector V_{ss} is presented in Figure A-10.

a. Model - Total Network

The first objective is to construct a model of the total network. This model would consist of all the contact elements in the network and the nodal values, or state variables, appearing at the contact elements. The formulation is such that the contact element is always connected to a state variable regardless of its position. This means in effect that both positions are assigned a state variable designation. See for example, the illustrative problem. The sum total of all these nodal values, x_i's, constitutes the network transition matrix, T_n, and delineates what contact element (relay) is connected to what contact element. This yields a description of the network in terms of paths between the contact elements.

b. System Transition Matrix

It is doubtful if all systems in this network will be operated simultaneously. Therefore, it is conceivable that only a portion of the total network need be examined at a time. Hence, the network transition matrix and its associated state vectors V_s and V_{ss} could be reduced to some smaller size by adjusting the necessary nodes, x_i's and y_i's, to accommodate a given finite time interval say from T_i to T_j. This would be accomplished by placing the rest of the network in storage. This operation establishes what might be called the system transition matrix and the system state vector.

c. Switching Algorithm

Switching algorithms are now established as previously demonstrated to cause paths to be selected through the network.

d. Event Marker

At the instant any contact element is caused to change state, $\alpha_{ij} = 0$ to $\alpha_{ij} = 1$, or vice versa, the event interval is terminated and a new interval initiated. This operation requires that the terminal conditions (TC's) of the system and subsystems state vectors V_S and V_{SS}, must become the initial conditions (IC's) for the next interval. And the system is updated for the next interval.

e. Event Interval Matrix

The time dependency of V_S and V_{SS} between events is determined by constructing a finite, or smaller, transition matrix from the system matrix stored for reference above. This matrix may be manipulated as desired and is reconstructed anew for each event interval. The resulting reduction yields the frequency sensitive path product P_{ij}'s; i.e., $P_{ij} \equiv F(t_{ij}$'s).

f. Time Constant Search

The transfer functions, identified as P_{ij} above, are now examined for the shortest time constant. The transfer function is then serialized (number series) based on this time constant and reduced to a single serial number. This operation is explained subsequently. Initial value of the system state vector, involved in the interval transition, is convolved with the appropriate path product serial number $\hat{P}_{ij}$ to generate a system output $\hat{P}_j = X_i \hat{P}_{ij}$.

g. Subsystem Contribution

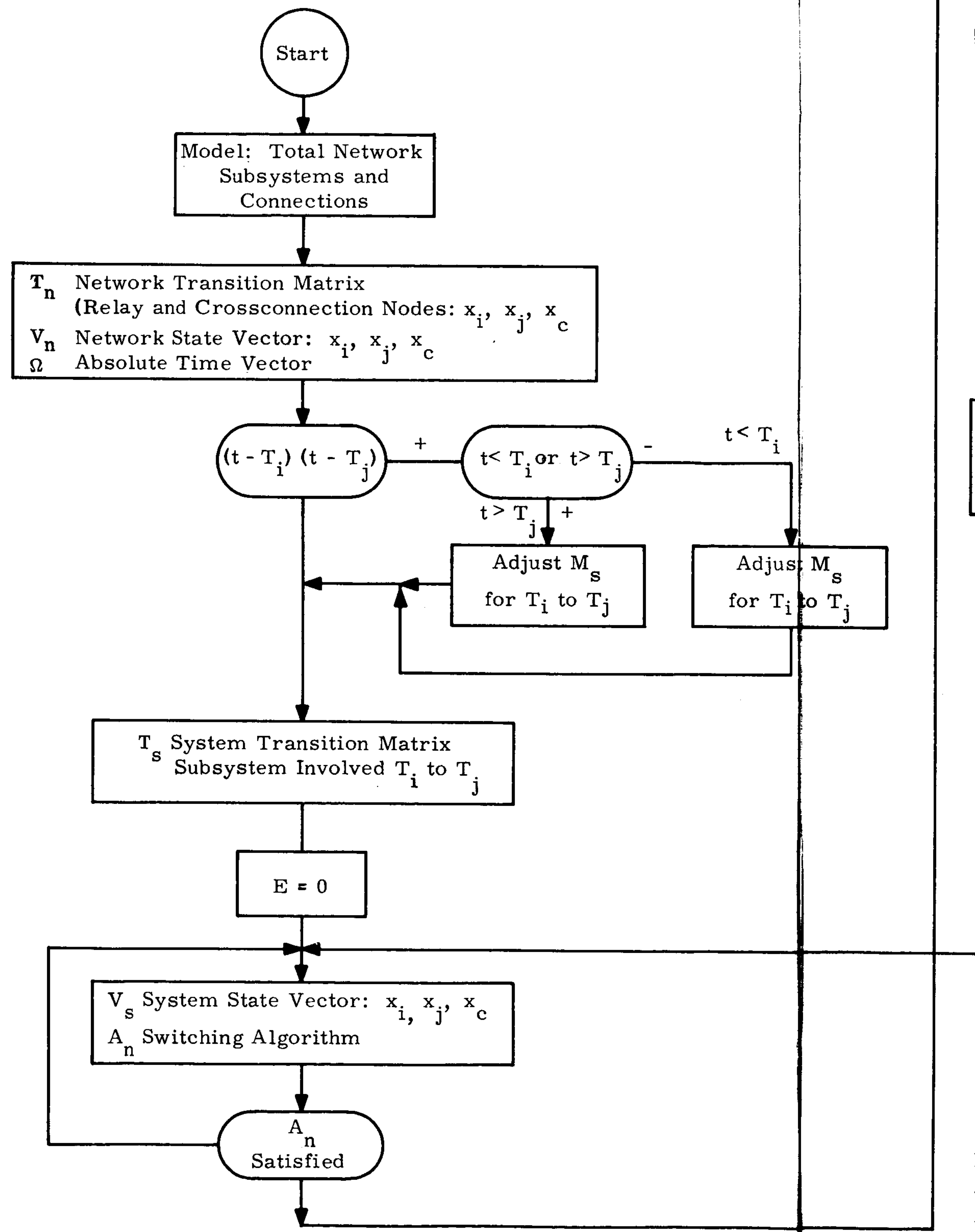
The subsystem also contributes to the dynamic behavior of the system state vector, V_S or $[x_j]$. This effect is now obtained by convolving the input vector with the path serial number. To this result is added the signal contributions from the subsystem internal sources; i.e., the initial condition sources, $y_i(0)$'s, convolved with the path product to the same external sink node.

h. Subsystem Contribution

At the subsystem level it may be desirable to follow its behavior in this time interval. This is accomplished with a subsystem state vector, V_{SS}, in the same manner as the system state vector V_S. A transition matrix is created and reduced to yield the equivalent paths from all input nodes, both external, x_i's, and internal, $y_i(0)$'s, to all output nodes, both external and internal.

It will be observed that at the time of switching for the subsequent interval the external source signals will be removed but that the subsystem internal nodes may continue to contribute to the system state vector and to the subsystem state vector. These two signals must be sorted and shunted to the appropriate location.

(A)



Start
Model: Total Network
Subsystems and
Connections
T_n Network Transition Matrix
(Relay and Crossconnection Nodes: x_i, x_j, x_c
V_n Network State Vector: x_i, x_j, x_c
Ω Absolute Time Vector
$(t - T_i)(t - T_j)$
+
$t < T_i$ or $t > T_j$
-
$t < T_i$
$t > T_j$
+
Adjust M_s
for T_i to T_j
Adjust M_s
for T_i to T_j
T_s System Transition Matrix
Subsystem Involved T_i to T_j
E = 0
V_s System State Vector: x_i, x_j, x_c
A_n Switching Algorithm
A_n
Satisfied

(B)

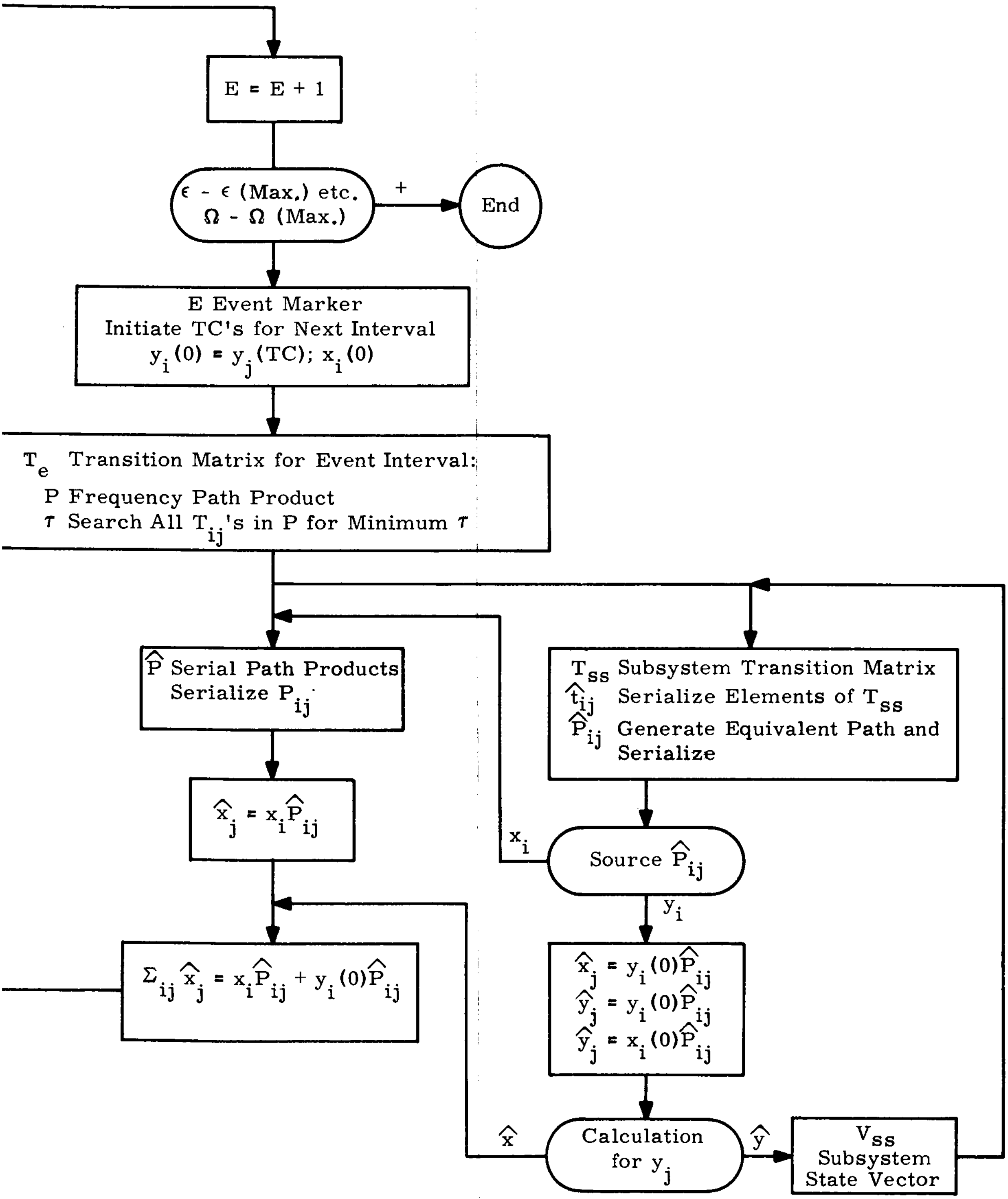


Figure A-10. System Flow Chart

A2.9 SERIAL NUMBERS/NUMBER SERIES

The system and subsystem matrix reduction schemes seek to enumerate all source and sink, or input and output, relationships as a collection of single equivalent edges or transfer functions. If this reduction is carried out algebraically, large order polynomials may be generated and consume a large amount of storage.

In order to improve this situation it is proposed to replace all transfer functions in the interval matrix, T_e, and the subsystem matrix, T_o, with serial numbers or number series as they are sometimes called (see Addendum A2). This serial number represents the impulse response of the transfer function in the time domain; i.e., as a series of values at discrete intervals $\Delta\tau$. This is a completely equivalent representation if $\Delta\tau$ is chosen on the basis of yielding a satisfactory integration scheme. For example, if the trapezoidal rule is observed, the number of terms required is of the order of at least 20 or slightly greater. The current program is based on a 40-term serial number and is generated by a slightly modified version of a program, described in Reference 1, in which a sample data of the time response is used to obtain the serial number.

Thus, the proposed matrix reduction schemes are reduced to cross multiplication of number series of fixed length. This process reduces the requirement of storing large order polynomials in the computer and providing the paraphernalia to operate with them. The generation of large-order polynomials is the result of the continual cascading of edges to obtain the equivalent path.

The generation of the number series is a separate subroutine. Several other methods are being investigated to carry out this operation.

In particular, one possibility is to replace the s operator in the transfer function with a numerical equivalent. For example (see References 9 through 12):

$$s = \frac{2}{\Delta\tau}\left[\frac{1,\ -1}{1,\ 1}\right]$$

$$s^2 = \frac{4}{\Delta\tau^2}\left[\frac{1,\ -2,\ 1}{1,\ 2,\ 1}\right]$$

$$s^n = \left(\frac{2}{\Delta\tau}\right)^n \left[\frac{1,\ -1}{1,\ 1}\right]^n$$

or in terms of 1/s:

$$\frac{1}{s} = \frac{\Delta\tau}{2}\left[\frac{1, \quad 1}{1, \ -1}\right]$$

$$\frac{1}{s^n} = \left(\frac{\Delta\tau}{2}\right)^n \left[\frac{1, \quad 1}{1, \ -1}\right]^n .$$

Such operators may have some advantages. However, the suitability of this approach is dependent upon the proper choice of $\Delta\tau$ and needs further investigation.

A2.10 MATRIX REDUCTION SCHEME

The formal matrix reduction algorithms, for the generation of equivalent paths, or edges, in the subsystem; i.e., of the T_o matrix (Equation A-2) is developed in Addendum A1. They have a topological significance which is of some interest to the system engineer.

This relationship may be demonstrated by considering the topological reduction of two examples. Basically, the operation has the following characteristics:

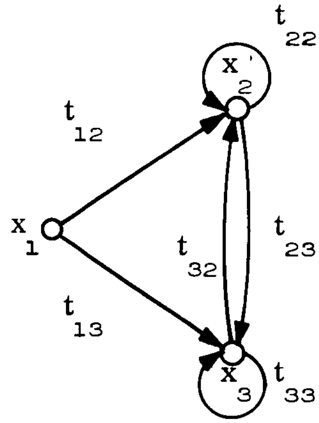
a. All nodes are numbered in a consecutive manner, say 1 to n.
b. Reduction is started at the furthermost node (n^{th}) and proceeds backward through the graph.
c. The first operation is the elimination of the furthermost self loop and all return paths from the furthermost node (n^{th}).
d. The next operation is the elimination of the self loop and return paths of the next to last node ($n\text{-}1^{th}$). Simultaneously the n^{th} and the $n\text{-}1^{th}$ node are exhibited as terminal nodes.
e. Process d is continued until all nodes are exhibited as terminal nodes. It is to be noted that this procedure is applied at the subsystem level.

Matrix Notation

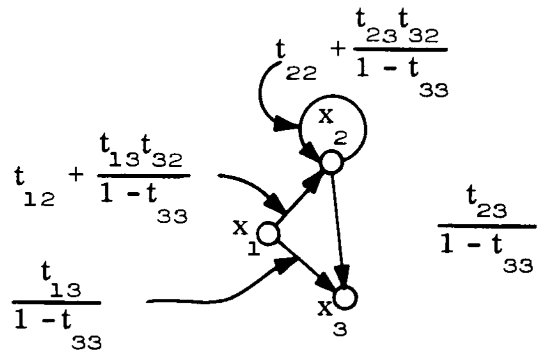
$[x_i] = [t_{ij}]\,[x_i]$ or $x_i = T^k x_i$, k = number of transformation.

Case 1.

$$T^{\circ} = \begin{bmatrix} 1 & 0 & 0 \\ t_{12} & t_{22} & t_{32} \\ t_{13} & t_{23} & t_{33} \end{bmatrix} \tag{A-5}$$



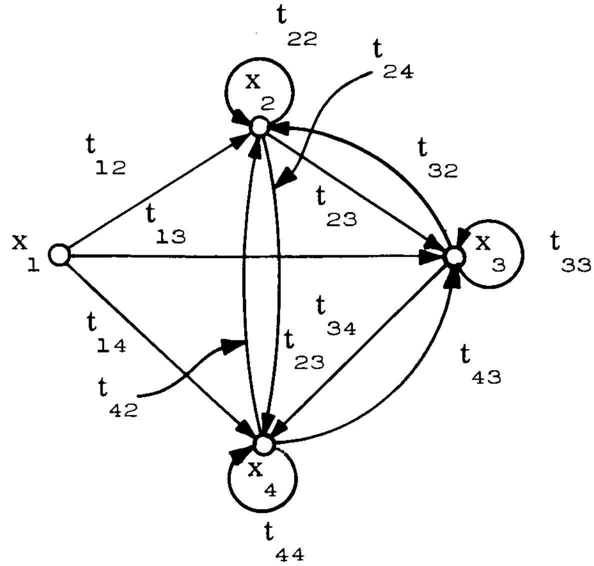
$$T^{1} = \begin{bmatrix} 1 & 0 & 0 \\ t_{12} + \dfrac{t_{13}t_{32}}{1 - t_{33}} & t_{22} + \dfrac{t_{23}t_{32}}{1 - t_{33}} & 0 \\ \dfrac{t_{13}}{1 - t_{33}} & \dfrac{t_{23}}{1 - t_{33}} & 0 \end{bmatrix} \tag{A-6}$$

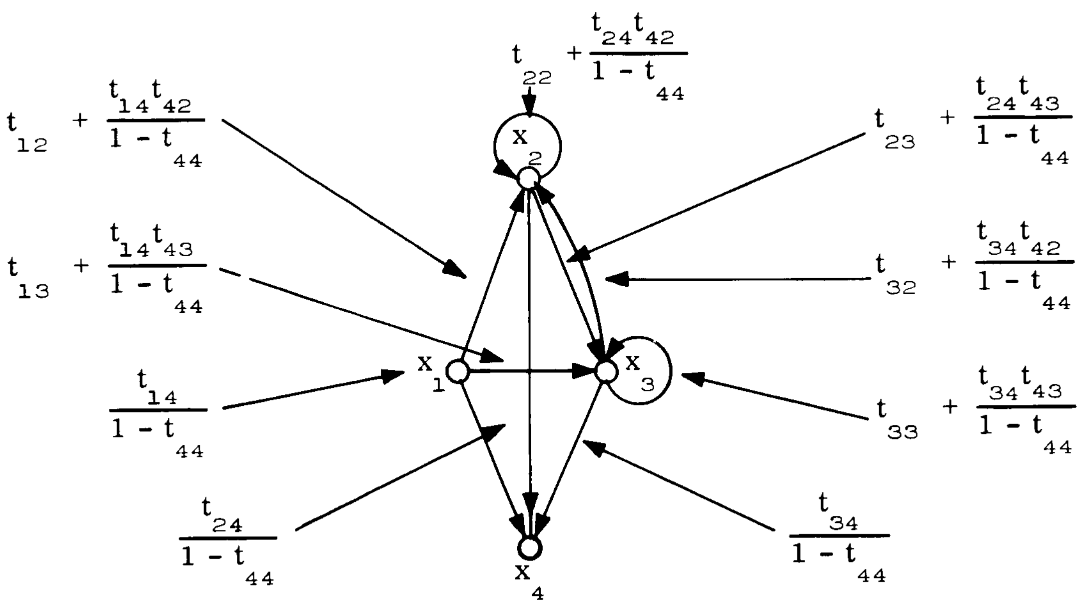


$$T^{2} = \begin{bmatrix} 1 & 0 & 0 \\ \left(t_{12} + \dfrac{t_{13}t_{32}}{1 - t_{33}}\right) \left[\dfrac{1}{1 - \left(t_{22} + \dfrac{t_{23}t_{32}}{1 - t_{33}}\right)}\right] & 0 & 0 \\ \dfrac{t_{13}}{1 - t_{33}} + \dfrac{t_{23}}{1 - t_{33}} \left(t_{12} + \dfrac{t_{13}t_{32}}{1 - t_{33}}\right) \left[\dfrac{1}{1 - \left(t_{22} + \dfrac{t_{23}t_{32}}{1 - t_{33}}\right)}\right] & 0 & 0 \end{bmatrix} \tag{A-7}$$

Case 2

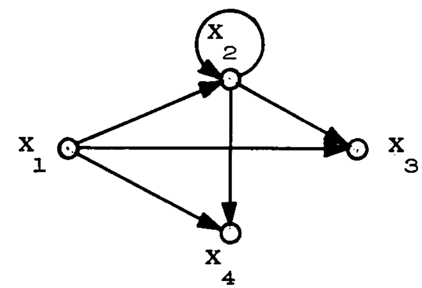
$$
T^{0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ t_{12} & t_{22} & t_{32} & t_{42} \\ t_{13} & t_{23} & t_{33} & t_{43} \\ t_{14} & t_{24} & t_{34} & t_{44} \end{bmatrix} \qquad \text{(A-8)}
$$





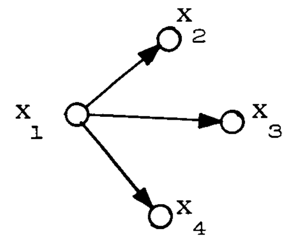
$$
T^{1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ t_{12} + \frac{t_{14}t_{42}}{1 - t_{44}} & t_{22} + \frac{t_{24}t_{42}}{1 - t_{44}} & t_{32} + \frac{t_{34}t_{42}}{1 - t_{44}} & 0 \\ t_{13} + \frac{t_{14}t_{43}}{1 - t_{44}} & t_{23} + \frac{t_{24}t_{43}}{1 - t_{44}} & t_{33} + \frac{t_{34}t_{43}}{1 - t_{44}} & 0 \\ \frac{t_{14}}{1 - t_{44}} & \frac{t_{24}}{1 - t_{44}} & \frac{t_{34}}{1 - t_{44}} & 0 \end{bmatrix} \qquad \text{(A-9)}
$$

$$
T^2 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
\left(t_{12} + \frac{t_{14}t_{42}}{1 - t_{44}}\right) + \frac{\left(t_{13} + \frac{t_{14}t_{43}}{1 - t_{44}}\right)\left(t_{32} + \frac{t_{34}t_{42}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \frac{t_{34}t_{43}}{1 - t_{44}}\right)} & \left(t_{22} + \frac{t_{24}t_{42}}{1 - t_{44}}\right) + \frac{\left(t_{23} + \frac{t_{24}t_{43}}{1 - t_{44}}\right)\left(t_{32} + \frac{t_{34}t_{42}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \frac{t_{34}t_{43}}{1 - t_{44}}\right)} & 0 & 0 \\
\frac{t_{13} + \frac{t_{14}t_{43}}{1 - t_{44}}}{1 - \left(t_{33} + \frac{t_{34}t_{43}}{1 - t_{44}}\right)} & \frac{t_{23} + \frac{t_{24}t_{43}}{1 - t_{44}}}{1 - \left(t_{33} + \frac{t_{34}t_{43}}{1 - t_{44}}\right)} & 0 & 0 \\
\left(\frac{t_{14}}{1 - t_{44}}\right) + \frac{\left(t_{13} + \frac{t_{14}t_{43}}{1 - t_{44}}\right)\left(\frac{t_{34}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \frac{t_{34}t_{43}}{1 - t_{44}}\right)} & \frac{t_{24}}{1 - t_{44}} + \frac{\left(t_{23} + \frac{t_{24}t_{43}}{1 - t_{44}}\right)\left(\frac{t_{34}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \frac{t_{34}t_{43}}{1 - t_{44}}\right)} & 0 & 0
\end{bmatrix}
$$



(A-10)

$$T^{3} = \begin{bmatrix} 1 \\[2ex] \dfrac{\left(t_{12} + \dfrac{t_{14}t_{42}}{1 - t_{44}}\right) + \dfrac{\left(t_{13} + \dfrac{t_{14}t_{43}}{1 - t_{44}}\right)\left(t_{32} + \dfrac{t_{34}t_{42}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \dfrac{t_{34}t_{43}}{1 - t_{44}}\right)}}{1 - \left[\left(t_{22} + \dfrac{t_{24}t_{42}}{1 - t_{44}}\right) + \dfrac{\left(t_{23} + \dfrac{t_{24}t_{43}}{1 - t_{44}}\right)\left(t_{32} + \dfrac{t_{34}t_{42}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \dfrac{t_{34}t_{43}}{1 - t_{44}}\right)}\right]} = t_{12}^{3} \\[4ex] \dfrac{t_{13} + \dfrac{t_{14}t_{43}}{1 - t_{44}}}{1 - \left(t_{33} + \dfrac{t_{34}t_{43}}{1 - t_{44}}\right)} + \dfrac{t_{23} + \dfrac{t_{24}t_{43}}{1 - t_{44}}}{1 - \left(t_{33} + \dfrac{t_{34}t_{43}}{1 - t_{44}}\right)} \times \dfrac{\left(t_{12} + \dfrac{t_{14}t_{42}}{1 - t_{44}}\right) + \dfrac{\left(t_{13} + \dfrac{t_{14}t_{43}}{1 - t_{44}}\right)\left(t_{32} + \dfrac{t_{34}t_{42}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \dfrac{t_{34}t_{43}}{1 - t_{44}}\right)}}{1 - \left[\left(t_{22} + \dfrac{t_{24}t_{42}}{1 - t_{44}}\right) + \dfrac{\left(t_{23} + \dfrac{t_{24}t_{43}}{1 - t_{44}}\right)\left(t_{32} + \dfrac{t_{34}t_{42}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \dfrac{t_{34}t_{43}}{1 - t_{44}}\right)}\right]} \\[4ex] \left(\dfrac{t_{14}}{1 - t_{44}}\right) + \dfrac{\left(t_{13} + \dfrac{t_{14}t_{43}}{1 - t_{44}}\right)\left(\dfrac{t_{34}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \dfrac{t_{34}t_{43}}{1 - t_{44}}\right)} + \left[\dfrac{t_{24}}{1 - t_{44}} + \dfrac{\left(t_{23} + \dfrac{t_{24}t_{43}}{1 - t_{44}}\right)\left(\dfrac{t_{34}}{1 - t_{44}}\right)}{1 - \left(t_{33} + \dfrac{t_{34}t_{43}}{1 - t_{44}}\right)}\right]\left(t_{12}^{3}\right) \end{bmatrix} \qquad \text{(A-11)}$$



A2.11 DYNAMIC SEQUENCING EXAMPLE: THERMOSTAT

A2.11.1 General

As an illustrative example of a simple network that incorporates most of the principles involved in dynamic sequencing, a bimetal thermostat, used in the temperature control loop of a house heating system, may be considered. The opening and closing of the bimetal thermostat being directly analogous to the action of a relay. The sequence of events is easily traced and the reduction process to remove self loops is included. Further, elements of the system must be treated as a subsystem and the nodal value appearing at the contact element as a system variable. That is to say, both the subsystem and system vectors, V_S and V_{SS}, are involved. The switching algorithm causing the change in state of the thermostat is an example of a subsystem state dependent situation.

The event profile generated by a typical house system will be investigated. Some effort is made to keep system parameters within representative ranges. However, the numbers which were chosen are arbitrary and may vary slightly from actual systems. For example, the hysteresis loop of the thermostat is taken as 6° F (±3° F). This may be larger than most cases, but it is of little consequence for the purpose here.

The system is started from rest at ambient conditions and put into operation at 60° F. Only three events are determined to illustrate the method. Further investigation can be carried out on a computer.

A2.11.2 Nomenclature

θ_f	=	furnace temperature
θ_R	=	room temperature
θ_a	=	ambient temperature = 30° F
q_B	=	heat from burner = 2×10^3 Btu/min.
q_R	=	heat transferred from furnace to room
q_a	=	heat loss to ambient
C_f	=	furnace heat capacity
C_R	=	room heat capacity
H_f	=	furnace heat transfer coefficient

H_R	=	room heat transfer coefficient
T_f	=	furnace time constant
T_R	=	room time constant
$\dot{\theta}$	=	$d\theta/dt$
s	=	Laplace operator
K_1, K_2	=	constants
E	=	error signal
q_f	=	furnace heat
SP	=	set point
TC	=	terminal condition
IC	=	initial condition

A2.11.3 Analysis

The temperature control model may be analyzed with the aid of Figure A-11.

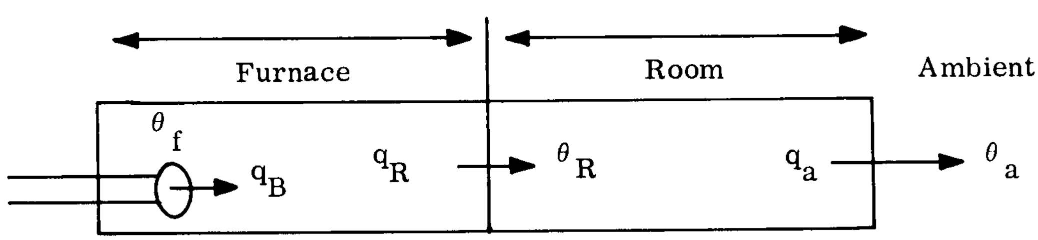


Figure A-11. Analysis of Temperature Control Model

The heat balance for the furnace is: $(dx/dt = \dot{x})$

$$q_f = q_B - q_R \tag{A-12}$$

$$C_f\dot{\theta}_f = q_B - H_f(\theta_f - \theta_R) \tag{A-13}$$

$$T_f\dot{\theta}_f + \theta_f = K_f q_B + \theta_R \tag{A-14}$$

where

$$T_f = \frac{C_f}{H_f} \tag{A-15}$$

$$K_f = \frac{1}{H_f} \quad . \tag{A-16}$$

And in a similar manner, the heat balance for the room is:

$$q_S = q_R - q_a \tag{A-17}$$

$$C_R\dot{\theta}_R = H_f(\theta_f - \theta_R) - H_R(\theta_R - \theta_a) \tag{A-18}$$

$$T_R\dot{\theta}_R + \theta_R = K_1\theta_f + K_2\theta_a , \tag{A-19}$$

where

$$T_R = \frac{C_R}{(H_f + H_R)} \tag{A-20}$$

$$K_1 = \frac{H_f}{(H_f + H_R)} \tag{A-21}$$

$$K_2 = \frac{H_R}{(H_f + H_R)} . \tag{A-22}$$

The coefficients required above may be obtained for a typical case from the following approximate data:

Ambient temperature, θ_a:	30° F
Maximum temperature rise obtainable with furnace on: $\Delta\theta_R$:	60° F
Maximum furnace temperature for $\Delta\theta_R$ = 60° F:	800° F
Furnace rating, q_B:	2×10^3 Btu/min
Furnace lag, T_f:	5 minutes
Room lag, T_R:	30 minutes

Whence, from the steady-state condition:

H_R = 2×10^3 Btu/min/60° F = 33.33 Btu/° F min.

H_f = 2×10^3 Btu/min/(800°-90°) = 2.82 Btu/° F min.

K_1 = 2.82/36.15 = 0.078 (num.)

K_2 = 33.33/36.15 = 0.922 (num.)

K_f = 1/2.82 = 0.355 °F min/Btu

Substituting in the above equations:

$$5\dot{\theta}_f + \theta_f = 0.355q_B + \theta_R \tag{A-23}$$

$$30\dot{\theta}_R + \theta_R = 0.078\theta_f + 0.922\theta_a \tag{A-24}$$

Whence transforming:

$$5s\theta_f(s) - 5\theta_f(0) + \theta_f(s) = 0.355q_B(s) + \theta_R(s) \tag{A-25}$$

$$30s\theta_R(s) - 30\theta_R(0) + \theta_R(s) = 0.078\theta_f(s) + 0.922\theta_a(s) \tag{A-26}$$

or

$$\theta_f(s) = \frac{5}{1 + 5s}\theta_f(0) + \frac{0.355}{1 + 5s}q_B(s) + \frac{1}{1 + 5s}\theta_R(s) \tag{A-27}$$

$$\theta_R(s) = \frac{30}{1 + 30s}\theta_R(0) + \frac{0.078}{1 + 30s}\theta_f(s) + \frac{0.922}{1 + 30s}\theta_a(s) . \tag{A-28}$$

For convenience Equations A-27 and A-28 are structured in Figure A-12. In Figure A-12(a) the ambient temperature θ_a will not be indicated since it can be considered as an infinite sink. The heat source $q_B(s)$ is added stepwise and consequently the edge in Figure A-12(b) includes this factor. Note that both values of q_B, 100% and 0%, are indicated.

A2.11.4 First Interval

For the first interval, or startup conditions, the system is assumed to be at equilibrium at 30° F (0° F reference value for heat load). The first event is generated by step changing the thermostat to 60° F.

The temperature response of the furnace is obtained from Figure A-12(b).

$$\theta_f(s) = \frac{2 \times 10^3}{s} \times \frac{0.355}{(1 + 5s)} \times \frac{(1 + 30s)(1 + 5s)}{150(s + 0.0303)(s + 0.2031)} \tag{A-29}$$

$$= \frac{710(1 + 30s)}{150s(s + 0.0303)(s + 0.2031)} \tag{A-30}$$

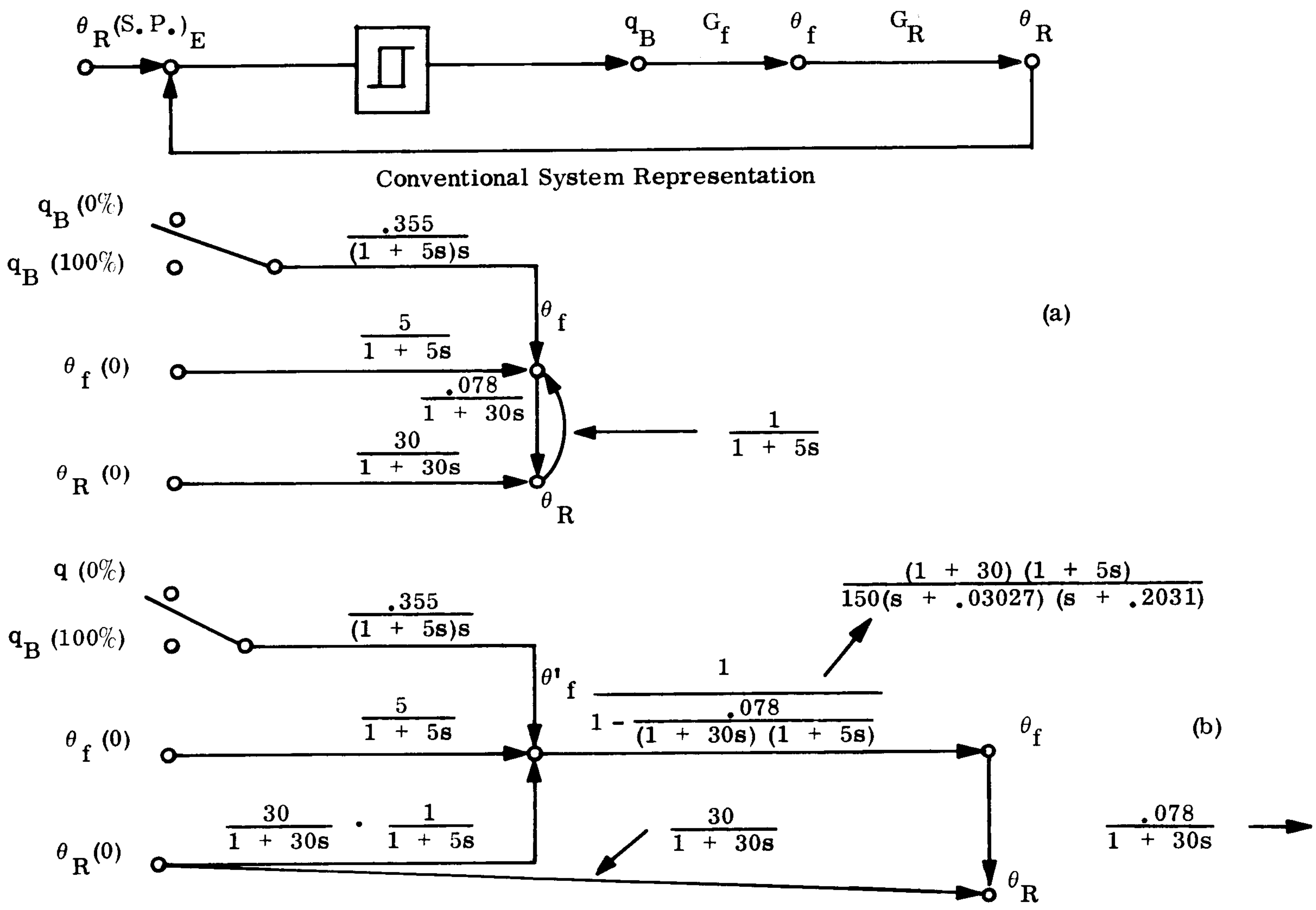


Figure A-12. Structuring of Equations A-28 and A-29

Equation A-30 gives a steady-state gain of 770° F (800°-30° F) which checks with the heat balance.

The time response of Equation A-30 is:

$$\theta_f(t) = 770 - 83.1\, e^{-0.0303t} - 686.9\, e^{-0.2031t} \tag{A-31}$$

and is presented in Table A-2 and Figure A-13.

Table A-2

Furnace Temperature: First Interval

Time (min)	83.1 Exp (-0.0303t) °F	686.9 Exp (-0.2031t) °F	θ_f °F	θ_f + 30 °F
0	83.1	686.9	0	30
10	60.6	91.3	618	648
15	52.2	33.0	685	715
30	33.1	1.37	735	765

The temperature response of the room is from Figure A-12(b):

$$\theta_R(s) = \frac{2 \times 10^3}{s} \times \frac{0.355}{1 + 5s} \times \frac{(1 + 30s)(1 + 5s)}{150(s + 0.0303)(s + 0.2031)} \times \frac{0.078}{1 + 30s} \tag{A-32}$$

$$= \frac{0.3688}{s(s + 0.0303)(s + 0.2031)} \tag{A-33}$$

which gives a steady-state gain of 60° F and checks the initial assumption.

The time response of Equation A-33 is:

$$\theta_R(t) = 60 + 10.5\, e^{-0.2031t} - 70.5\, e^{-0.0303t}. \tag{A-34}$$

Equation A-34 is plotted in Figure A-14 and represented in Table A-3.

Examination of Figure A-14 indicates that the thermostat relay opens at 63° F assuming a 6° F hysteresis loop for the mechanism (±3° F).

The time required for the first interval is 31.7 minutes.

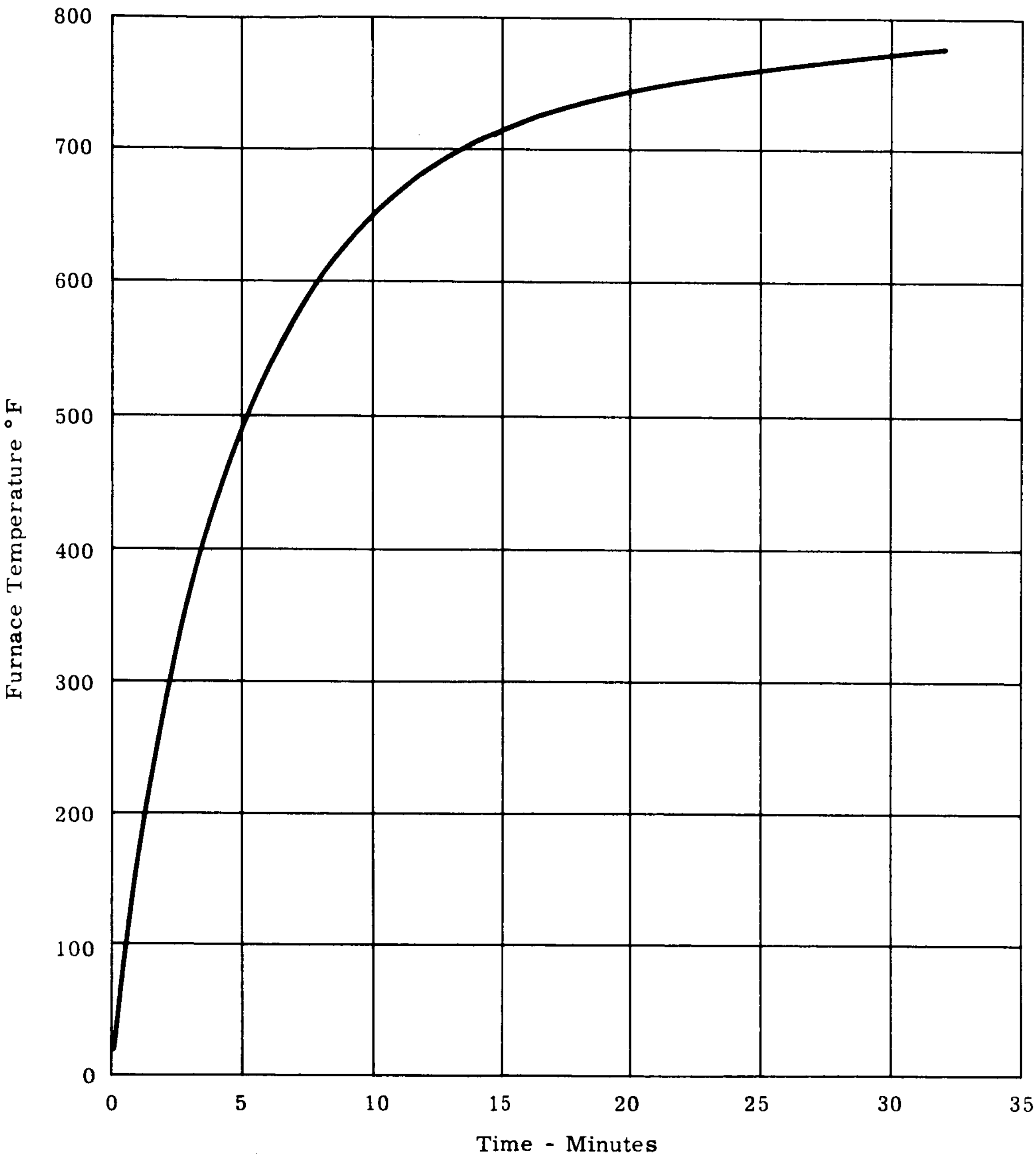


Figure A-13. Furnace Temperature Startup

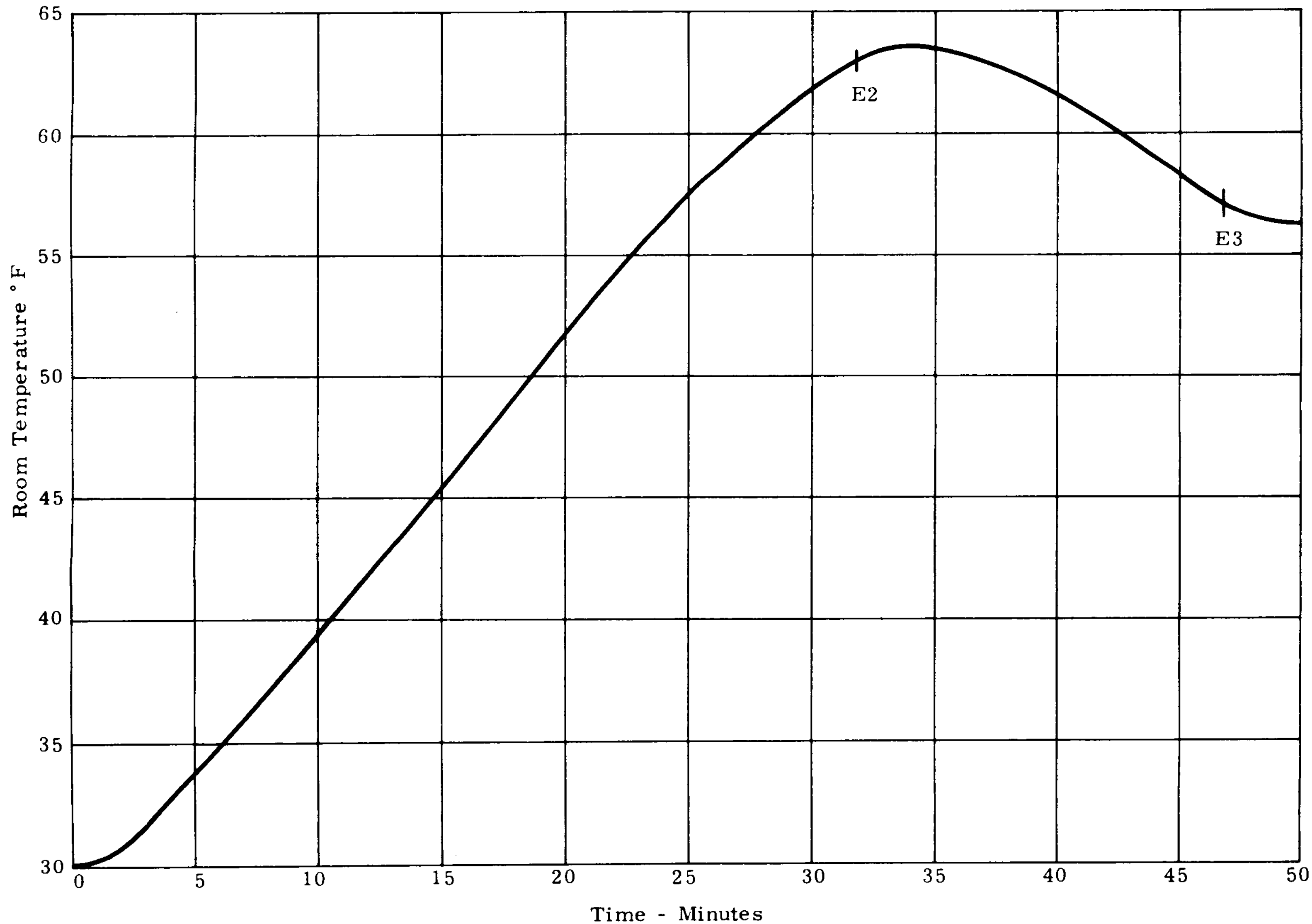


Figure A-14. Room Temperature Control

Table A-3

Room Temperature: First Interval

Time (min)	10.5 Exp (-0.2031t) °F	70.5 Exp (-0.0303t) °F	θ_R °F	θ_R + 30° F °F
0	10.5	70.5	0	30
10	1.41	52.1	9.3	39.3
20	0.19	38.5	21.7	51.7
30	0.021	28.5	31.5	61.6

A2.11.5 Second Interval

The second interval commences with the opening of the thermostat control unit - Event 2. The IC conditions for the start of the second interval are:

$$\theta_R(0) = 33° F \; (30° + 33° = 63° F \text{ room temperature})$$

$$\theta_f(0) = 737.1° F$$

$$q_B(0) = 0 \text{ Btu/min.}$$

The time response for θ_R is (from Figure A-12b):

$$\theta_R(s) = \frac{5.2 \times 10^{-4} (1 + 5s)}{(s + 0.0303)(s + 0.2031)} \left[\frac{5 \cdot 736}{1 + 5s} + \frac{30 \cdot 33}{(1 + 30s)(1 + 5s)}\right] + \frac{30 \cdot 33}{1 + 30s} \tag{A-35}$$

which has a steady-state gain of 0°, which checks, and hence:

$$\theta_R(t) = 43.5 e^{-0.0303t} - 10.5 e^{-0.2031t}. \tag{A-36}$$

Equation A-36 is plotted in Figure A-14 and presented in Table A-4.

Examination of Figure A-14 indicates that the thermostat turns on at 57° F or some 15.2 minutes later. This constitutes the generation of the third event.

Table A-4

Room Temperature: Second Interval

Time (min)	43.50 Exp (-0.0303t) °F	10.50 Exp (-0.2031t) °F	θ_R °F	$\theta_R + 30$ °F
0	43.5	10.5	33.0	63.0
5	37.3	3.8	33.5	63.6
10	32.1	1.4	30.7	60.8
15	27.7	0.5	27.2	57.1

The system now starts back up and goes on a limit cycle. However, we are not interested in this at the moment, and the generation of the first three events will be sufficient for our purpose.

The piece-wise solution of the thermostat action can be formulated with the methodology presented with little difficulty. To formulate the problem in this manner it is necessary to start with the oriented graph depicted in Figure A-12.

The first step involved in this process is to convert the graph of Figure A-12 to that of Figure A-15. This is easily done by observing the laws of linear graphs, as presented, for example, in References 2 and 3.

Several comments may be made with respect to this reduction.

a. The oriented graph is reduced to an equivalent graph with no return paths. For example, edge $t(\theta_R, \theta_f)$ in Figure A-12(a) must be eliminated. This is accomplished by splitting the θ_f node into a modified θ_f' node and into the real node θ_f. The elimination of the self loop which would have occurred at the θ_f node can be accomplished in a different manner. The alternative procedure is to absorb the self loop on all the incoming edges. However, this leads to more complex expressions and for this reason the node splitting procedure is adopted.

b. The reduction of the original graph must start at the output end and work backwards to preserve all internal node values. Reduction in any other manner is non-node preserving.

c. Any internal node may be exhibited as an output node by duplicating the node and joining the internal node with a unity edge. See, for example, nodes θ_f(TC) and θ_R(TC) in Figure A-15.

Figure A-15 is further reduced to Figure A-16, where the contact element α, which represents the thermostat, is a multiple value device controlling the signal from node 1 and 1' to node 4 and 5; i.e., α has the value α_{14}, α_{15}, $\alpha_{1'4}$, and $\alpha_{1'5}$ where $\alpha_{14} \equiv \alpha_{15}$ and $\alpha_{1'4} \equiv \alpha_{1'5}$. The algorithm producing the switching action is the algorithm previously presented in Figure A-9. The system response is now obtained by superposition of all the signals where $\theta_R(0)$ and $\theta_f(0)$ are subsystem internal source nodes and $q_B(100\%)$ and $q_B(0\%)$ are system input nodes. $\theta_R(s)$ and $\theta_f(s)$ represent subsystem nodes of interest and the switching algorithm is an example of a subsystem state, V_{ss}, dependency.

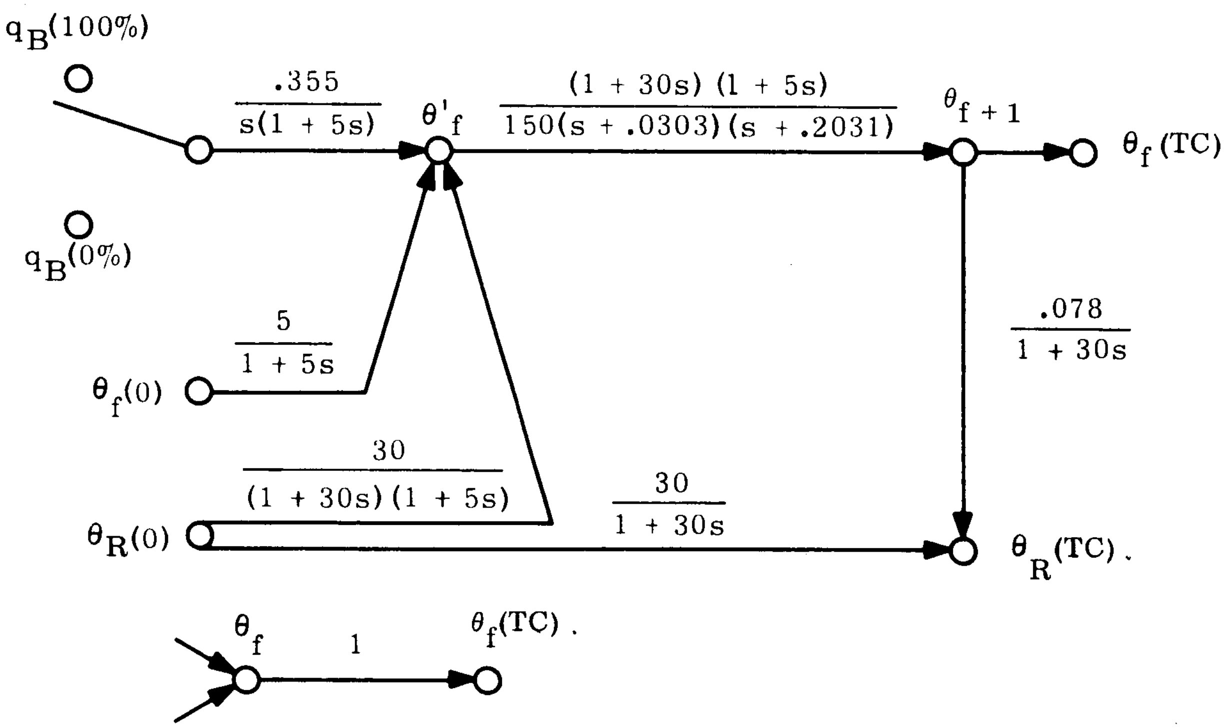


Figure A-15. Thermostat - Signal Flow Graph

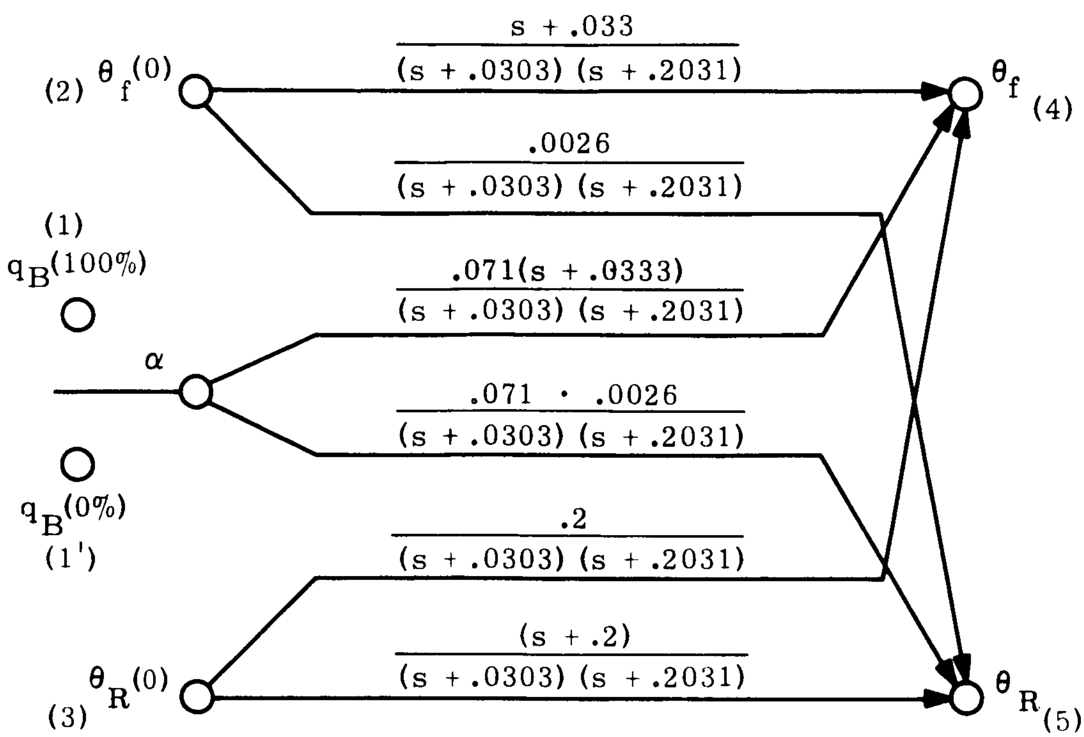


Figure A-16. State Flow Graph

A2.12 INVERSE NUMBER SERIES

It may be desirable to perform the inverse operation of generating the analytical time solution for some arbitrary number series. As a point of interest it may be noted that a program is currently under development which utilizes Prony's method, Reference 4, and which can accomplish this process for some types of functions.

An example of this inverse problem is solved by Prony's method.

Consider:

the transfer function:

$$G(s) = \frac{30}{s(s+1)(s+2)(s+15)} \tag{A-37}$$

the time solution is:

$$G(t) = 1 - 2.14286\, e^{-t} + 1.15385\, e^{-2t} - 0.010989\, e^{-15t} \tag{A-38}$$

and its serial number is presented in Table A-5.

Table A-5

Serial Number for Transfer Function

Time (sec)	Serial Number	
0.0000000000-31	0.3597888284-10	
0.1250000001+00	0.5865163328-02	
0.2500000003+00	0.3072377547-01	
0.3750000003+00	0.7223259903-01	
0.5000000006+00	0.1247566364+00	
0.6250000006+00	0.1835858991+00	
0.7500000006+00	0.2452349095+00	
0.8750000006+00	0.3072218927+00	
0.1000000001+01	0.3678303801+00	
0.1125000001+01	0.4259161863+00	
0.1250000001+01	0.4807584466+00	
0.1375000001+01	0.5319460164+00	
0.1500000001+01	0.5792913099+00	
0.1625000001+01	0.6227652686+00	
0.1750000001+01	0.6624485431+00	
0.1875000001+01	0.6984950893+00	
0.2000000002+01	0.7311052441+00	$\dfrac{30}{s(s+1)\,(s+2)\,(s+15)}$
0.2125000002+01	0.7605060140+00	
0.2250000002+01	0.7869368353+00	
0.2500000002+01	0.8318509648+00	$\Delta\tau$ = 0.125 second
0.2625000002+01	0.8507990206+00	
0.2750000002+01	0.8676989444+00	
0.2875000002+01	0.8827519531+00	
0.3000000002+01	0.8961443452+00	
0.3125000002+01	0.9080473236+00	
0.3250000002+01	0.9186172667+00	
0.3375000002+01	0.9279962992+00	
0.3500000002+01	0.9363130557+00	
0.3625000002+01	0.9436835554+00	
0.3750000002+01	0.9502121320+00	
0.3875000002+01	0.9559923758+00	
0.4000000005+01	0.9611080606+00	
0.4125000005+01	0.9656340365+00	
0.4250000005+01	0.9696370744+00	
0.4375000005+01	0.9731766573+00	
0.4500000005+01	0.9763057131+00	
0.4625000005+01	0.9790712890+00	
0.4750000005+01	0.9815151672+00	
0.4875000005+01	0.9836744258+00	
0.5000000005+01	0.9855819458+00	

Every other serial entry, in Table A-5, was utilized as data for the inverse Prony Program. The following time solution was obtained:

$$G(t)_E = 1 - 2.13871\,e^{-0.999347t} + 1.15004\,e^{-2.00325t} - 0.0113211\,e^{-14.0648t}. \quad \text{(A-38a)}$$

Comparison of Equations A-38 and A-38a shows the accuracy of the inversion to be quite good.

A3 A COMBINED METHOD OF FACTORING AND CURVE FITTING FOR REDUCTION OF LINEAR SYSTEMS IN DYNAMIC SIMULATION

A3.1 INTRODUCTION

For dynamic simulations the reduction of system matrices requires the performance of algebraic operations on a large number of polynomials. An analytical approach may, in some cases, require an inordinately large amount of computer memory. One way of alleviating this situation is the use of number series. Another is the use of a numerical technique for reducing a linear system without time delays and converting the results to an analytical form, which is described in this section.

Such a numerical approach is feasible with a high speed digital computer. Though the full advantages cannot be spelled out in a clear-cut fashion until some practical examples have been worked out, it may be pointed out that there are certain desirable features associated with this numerical method.

a. Being strictly a numerical method, it uses the full capabilities of a high speed digital computer.
b. It obtains the output in analytical form.
c. Storage requirements may be kept to a minimum.
d. Such problems as errors may be controlled satisfactorily.
e. Stability conditions may be investigated readily.
f. Partial fraction expansion of the output transforms is made possible.
g. The time response of the system can be obtained in explicit form in most cases.

These are some of the advantages of the proposed technique. It is true that increasing the size of the system to be treated makes the control of errors harder, but this is also true of any other approach, whether it be numerical or analytical. This can be best studied by means of actual examples of systems, which has not yet been done. Such studies also can be used to evaluate the advantages and disadvantages of the application to a particular situation of the several available techniques. It well may be that no one technique will be found to be best for all cases, but that different systems will require different techniques -- or appropriate combinations of techniques -- for optimum results in the system matrix reduction.

A3.2 SOLUTION OF EQUATIONS

Consider a linear system of equations that describes a subsystem of n nodes, such as x_i ($i = 1, \ldots, n$), with connections from node i to node j denoted by the transfer functions t_{ij}. For illustration purposes, assume that one node, say x_1, is an input node and the remaining n-1 nodes are output nodes, though in general there may be any number of input and output nodes. A typical connection system may be the following:

$$x_1 = x_1 \text{ (input node)}$$

$$x_i = \sum_{j=1}^{n} t_{ji} x_j \qquad i = 2, \ldots, n \text{ (output nodes)} . \tag{A-39}$$

It is desired to reduce the above system to the following form:

$$x_i = t^*_{1i} x_1 \qquad i = 1, \ldots, n \qquad t^*_{1i} = 1 . \tag{A-40}$$

The functions t^*_{1i} represent the equivalent transfer functions from the input node x_1 to each output node x . At this stage of the game, assume that the functions t_{ji}, and hence t^*_{1i}, are rational functions of the form P(s)/Q(s), where both P(s) and Q(s) are polynomials in s with real coefficients.

To make the system (Equation A-39) amenable to the present method of reduction, convert it into the following form:

$$- t_{1i} x_1 = \sum_{j=2}^{n} r_{ji} x_j , \tag{A-41}$$

where

$$\begin{aligned} r_{ji} &= t_{ji} & j &\neq i \\ &= t_{ji} - 1 & j &= i , \end{aligned} \tag{A-42}$$

and then to a form which is denoted by

$$\sum_{j=2}^{n} a_{ij} x_j = b_i x_1 \tag{A-43}$$

which is obtained from Equation A-41 by eliminating the denominators in r_{ji}'s.

Thus, Equation A-43 represents a linear system in x_j, with a_{ij} being polynomials in s and related to r_{ji} in Equation A-41 by:

$$
\begin{aligned}
a_{ij} &= r_{ji} R_i \\
b_i &= - t_{i1} R_i \, ,
\end{aligned}
\tag{A-44}
$$

where R_i are the appropriate polynomials (preferably of least degree) which effect the conversion from Equation A-41 to A-43. These equations are to be solved for the x_i.

In matrix form, let

$$A\vec{X} = \vec{B}x_1 \tag{A-45}$$

where

$$A = (a_{ij})_1^n$$

$$X = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \qquad B = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

and the a_{ij} are assumed to be polynomials with real coefficients. Then, the formal solution of Equation A-45 is:

$$\vec{X} = \frac{C}{q(s)} \vec{B}x_1 \tag{A-46}$$

where

$$C = (c_{ij})_1^n \, ,$$

$$\frac{C}{q(s)} = A^{-1} \quad ,$$

and

$$q(s) = |A| \, . \tag{A-47}$$

Thus, $q(s)$ as well as the elements c_{ij} of C are polynomials with real coefficients.

Then:

$$x_i = \frac{p_i(s)}{q(s)} x_1 \qquad i = 1, \ldots, n \tag{A-48}$$

where

$$p_i(s) = \sum_{j=1}^{n} c_{ij} b_j \qquad i = 1, \ldots, n \tag{A-49}$$

Explicit polynomial expressions of $q(s)$ as well as of $p_i(s)$ $(i = 1, \ldots, n)$ are sought.

In this method a number of discrete values of s are used in Equations A-47 and A-49 to find the polynomials as sets of numerical values, from which analytical expressions are to be derived by curve-fitting techniques. As a first step, it is necessary to know the degrees of the polynomials.

A3.3 DEGREES OF THE POLYNOMIALS

In simple cases, the degrees of the polynomials may be found by inspecting the forms of $A(s)$ and $\vec{B}(s)$. In large systems, however, this is no longer an easy matter, and some numerical tests are needed for the degree identification of the above polynomials.

To begin with let $g(s)$ be a polynomial representing $q(s)$ or any of the $p_i(s)$:

$$g(s) = d_n s^n + d_{n-1} s^{n-1} + \ldots + d_o \tag{A-50}$$

and let $\hat{g}(s)$ denote the numerical value corresponding to some assigned value of s. Clearly, for sufficiently large values of s,

$$\hat{g}(s) \rightarrow d_n s^n$$

Now let s_1 and s_2 be two such large values of s.

Then

$$\frac{\hat{g}(s_1)}{s_1^{\,v}} \div \frac{\hat{g}(s_2)}{s_2^{\,v}} \rightarrow \left(\frac{s_1}{s_2}\right)^{n-v}$$

If v is so chosen that v = n the above ratio is nearly 1. Hence to determine v, the following equation is solved:

$$\frac{\hat{g}(s_1)}{\hat{g}(s_2)} \left(\frac{s_2}{s_1}\right) = 1$$

from which is obtained

$$v = \log \frac{\hat{g}(s_2)}{\hat{g}(s_1)} \div \log \frac{s_2}{s_1} \tag{A-51}$$

It may happen that, though s_2 and s_1 may be very large, they are not large enough, due to the fact that the coefficient of the next power of s is very large. In this case, the value of v determined by Equation A-51 will be incorrect. This may be tested by taking even larger values of s_1 and s_2, say twice as large, and recalculating v. If there is a significant change, the process should be repeated. If there is still a change, the process should be repeated as often as necessary, until two successive values of v are essentially equal.

The value of v so calculated may not be precisely an integer, but the degree n of the polynomial should be taken to the nearest integer to the calculated value.

A word of caution in the above procedure for the degree determination of the polynomials is that the computer may impose a limitation as to the size of the numbers that can be used. Assume that the biggest number a computer can hold, before it starts an accumulator over flow, is 10^{38}. Further, assume that the polynomial g(s) is of the form

$$g(s) = d_{45}s^{45} + d_{44}s^{44} + \ . \ . \ . + d_0$$

Definitely, s cannot have a value as large as 10, which may not be large enough. Since it is known, however, that $p_i(s)$, for example, comes from the n^{th} order determinant $|(a_{ij})|$, this difficulty can be overcome by computing an equivalent one of the form

$$|(a_{ij}s^{-j/n})| .$$

This way such a limitation is avoided, and sufficiently large values of s necessary for the correct identification of the degree can be used.

It should be remarked that, since the x_i represent transfer functions, the following must be true:

$$x_i \to 0 \quad \text{as} \quad s \to \infty$$

Thus, the degree of q(s) must be greater than the degree of $p_i(s)$ for all i = 1, ..., n.

A3.4 CURVE-FITTING THE POLYNOMIALS

Finding the coefficients of a polynomial that has designated values is a standard and well-known procedure, as any interpolation formula represents such a process (see Reference 4, for example). However, errors may tend to build up if the degree is large. To minimize this build-up the present method combines a curve-fitting process with the calculation of numerical values of the polynomials g(s), in such a way as concurrently to determine the real roots. Thus g(s) is numerically expressed in the form

$$g(s) = g_1(s)g_2(s) \tag{A-52}$$

where

$$g_1(s) = (s-\sigma_1)(s-\sigma_2)\ldots(s-\sigma_r), \quad 0 \le r \le n \tag{A-53}$$

$$g_2(s) = e_{n-r}s^{n-r} + e_{n-r-1}s^{n-r-1} + \ldots + e. \tag{A-54}$$

First will be found the roots of g(s), thereby determining $g_1(s)$. The method to be used will be a finite difference analog of the Newton-Raphson method (Reference 4). Its application will be made clear by referring to Figure A-17. Essentially it is as follows:

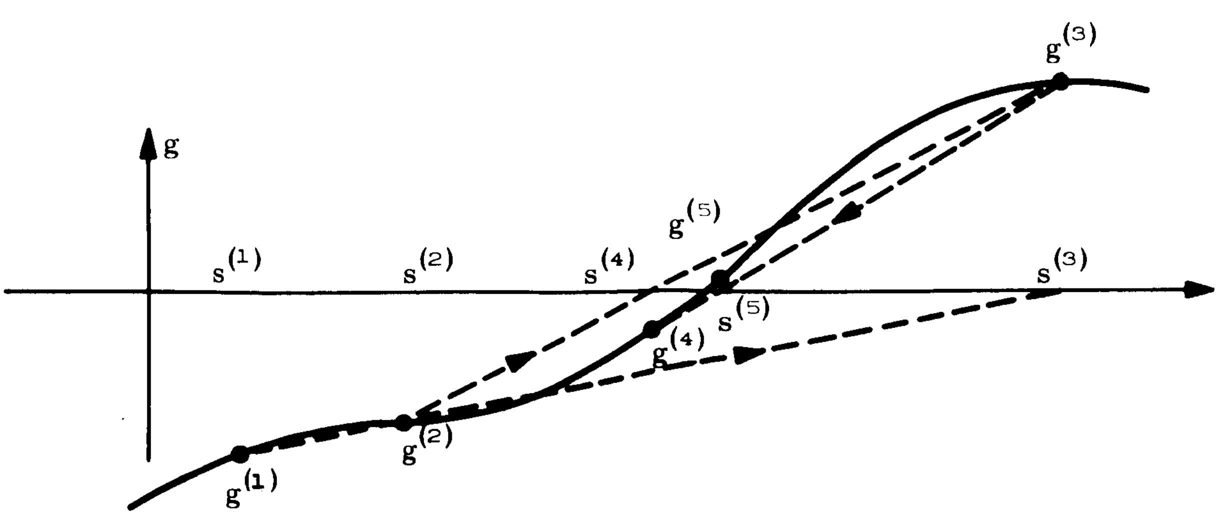


Figure A-17. Finding the Real Roots.

To start, choose two values of s, say $s^{(1)}$ and $s^{(2)}$, and calculate the corresponding values of g, say $g(s^{(1)}) = g^{(1)}$ and $g(s^{(2)}) = g^{(2)}$, from their determinantal definitions, Equation A-47 or A-49 (if $g^{(1)} = g^{(2)}$, pick a different value of one s). Then calculate $s^{(3)}$ from

$$s^{(3)} = s^{(2)} - \frac{(s^{(2)} - s^{(1)})g^{(2)}}{g^{(2)} - g^{(1)}} . \tag{A-55}$$

This is geometrically equivalent to passing a straight line through the points $(s^{(1)}, g^{(1)})$ and $(s^{(2)}, g^{(2)})$, Figure A-17, and finding its intersection with the s-axis.

For the next step, take $s^{(3)}$ with $s^{(2)}$ to calculate $s^{(4)}$ by the same procedure, using Equation A-55, with the superscripts raised by unity. Equation A-55 indicates that $g^{(k)} \to 0$ as k increases, so that $s^{(k)}$ approaches a root of g(s).

The figure also indicates that

$$\frac{s^{(k)} - s^{(k-1)}}{g^{(k)} - g^{(k-1)}} \to \frac{0}{0} , \tag{A-56}$$

which is difficult to compute accurately. In many cases the numerator of Equation A-56 will approach zero faster than the denominator, so that the difficulty will be more theoretical than real. In those cases, the iteration formula is:

$$s^{(k+1)} = s^{(k)} - \frac{(s^{(k)} - s^{(k-1)})g(s^{(k)})}{g(s^{(k)}) - g(s^{(k-1)})} \tag{A-57}$$

In those cases where difficulty might exist if the index k were increased one more step, it may be avoided by continuing the iteration with the formula:

$$s^{(k-1)} = s^{(k)} - \frac{(s^{(j)} - s^{(j-1)})g(s^{(k)})}{g(s^{(j)}) - g(s^{(j-1)})} \tag{A-58}$$

where

$$j < k$$

is the index number at which it becomes apparent that the coefficient of $g(s^{(k)})$ in Equation A-57 is becoming indeterminant.

In all cases, the iteration continues, using either Equation A-57 or A-58, until

$$|s^{(k+1)} - s^{(k)}| < e$$

where e is an assigned small positive number determined by the desired accuracy of the root. When this occurs, one root of g(s) has been found:

$$\sigma_1 = s^{(k+1)}$$

To find additional roots, repeat the procedure on numerical values of the function $n \rightarrow h$, where

$$n \rightarrow h = \frac{g(s)}{s - \sigma_1}$$

The procedure may be repeated continually until all n roots have been found, or until the iteration does not converge. When the latter happens, there are no more real roots of g(s), with the possible exception of some multiple roots, or several roots lying close together.

There are techniques for modifying the iteration scheme to find multiple, or bunched, roots, but seldom will they be required. It is not necessary to find <u>all</u> the real roots of g(s), although it is desirable to find as many as possible. If the number of roots calculated is r, then the polynomial to be fitted is $g_2(s)$, of degree n-r, rather than g(s), of degree n. If r is large, this will be an easier and more accurate procedure. (The technique is well-known, Reference 4, using an interpolation formula or least squares, so that a description of the details will not be given here.)

As many values of s as is desired may be used to calculate corresponding values of $g_2(s)$. First, g(s) is calculated from its definition, Equation A-47 or A-49. Then $g_1(s)$ is calculated from Equation A-53 and the r values of the located roots. Finally, $g_2(s)$ is calculated from Equation A-52. When the coefficients of $g_2(s)$ have been evaluated by the curve-fitting technique used, g(s) is constructed in analytical form, with numerical coefficients, by multiplying $g_1(s)$ and $g_2(s)$.

A3.5 SUMMARY

To recapitulate, this method determines a system transfer function by calculating numerical values for the polynomials in its numerator and denominator. The degree of each polynomial is determined, and the analytical expression for each found by curve-fitting. The accuracy is enhanced by arranging the calculation so as first to extract some or all of the real roots, and curve-fitting polynomials of lower degree. In most cases this method makes more efficient use of computer facilities than do algebraic methods of matrix reduction, and can, therefore, handle systems of more subsystems and components - perhaps several times as many.

A3.6 EXPLANATION OF FLOW CHARTS

Figure A-18 illustrates the iterative procedure for extracting real roots using the basic iteration procedure.

N = degree of polynomial.

N_1 = control number for either going into iteration and then into the curve-fitting or directly into the curve-fitting problem.

X_1, X_2 = two arbitrary initial predictions from where we start the iteration.

e = criteria parameter for convergence.

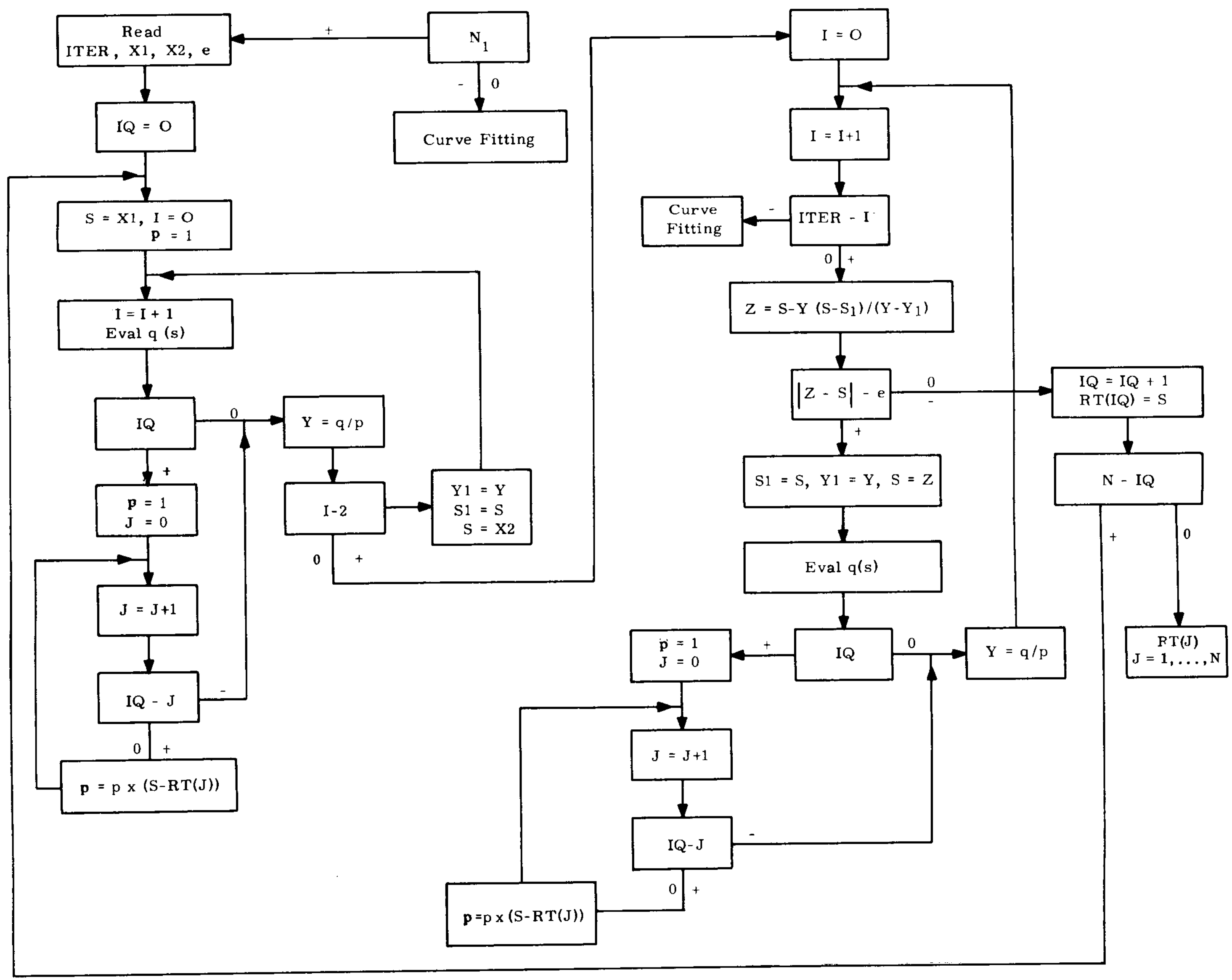


Figure A-18. Iterative Procedure for Extracting Real Roots

A4 DIFFERENTIAL EQUATIONS

A4.1 APPLICATION OF DIFFERENTIAL EQUATIONS TO MECHANICAL SYSTEMS

Mechanical systems simulation will be done by solving systems of differential equations, many of which may be nonlinear. Each equation or set of equations will represent the physical characteristics of each significant component in the system and the external constraints imposed on that component.

Simulation design will consider all sources of input as indicated in Reference 13. The primary types of input will be:

a. Valve actuation: both manual and remote automatic.
b. Pump startup and shutdown: both local electrical switching and remote automatic electrical switching.

The design will further account for the sequential requirements that will occur during checkout and countdown. This means, of course, that a sequence interface with external checkout and countdown equipment will have to be simulated. The most likely method at this point will be based on the established logic statement approach as presently used in ESE simulation.

Representative output from the simulation will be:

a. Flow distribution for all valve and pump configurations.
b. Pressure heads at all points as indicated.
c. Temperature levels at all points as indicated.
d. All valve and pump configurations as they occur during a sequence.
e. All temperatures, pressures and flows that are not indicated but are required for design validation and/or input to more complex component models.

The primary usage of this simulation will be in the following areas:

a. Mechanical system design validation.
b. Checkout and countdown sequence validation.
c. Phased test requirements.
d. Field operations engineer training.
e. Compatibility with other systems.
f. Establish boundary conditions for more complex component design models.
g. Safety analysis of high pressure piping systems.

The modeling (mathematical definition) would logically start from two points:

a. Detail requirements for each subsystem unit and the interfaces between units as indicated in Reference 13.
b. Phase-1 model to be entirely steady state.

The first statement is self-evident and needs no further elaboration. The second statement comes from simulation experience and is based on the following as objectives:

a. A steady-state model will accurately reflect system flow, temperature, and pressure distributions for all valve and pump configurations.
b. Precise computation of all steady-state initial conditions is an essential requirement for determination of dynamic simulation starting point.
c. As a phase effort, the steady-state model can be achieved in the shortest length of time and be put to use in the area of system checkout and countdown design validation.

The above statements imply a Phase-2 effort directed towards dynamic simulation. The most effective and efficient approach would be to overlap Phase 2 with Phase 1, so that they run concurrently but with Phase 2 lagging. The point in time at which Phase 2 would be started would coincide with a selected build up of data base and steady-state simulation pertinent to detailed definitions of subsystem units. It is recognized that in Phase 1 a considerable amount of the transient data required for Phase 2 will be acquired automatically. Phase 2 then would be demarked by active programming efforts and more intense efforts directed towards detailed system definitions in terms of transient functions.

In conclusion, it should be noted that the steady-state model will serve as:

a. A starting point.
b. A useful engineering design tool.
c. A source for dynamic model initial conditions.

The dynamic model will serve as:

a. A more sophisticated design tool.
b. A more life-like training tool.
c. A tool for extending studies into total system transient effects.
d. A tool for studying fail-safe requirements for such things as automatic shut-off in case of pipe rupture.

A4.2 MODELING AND INPUT DATA REQUIREMENTS FOR DYNAMIC SIMULATION OF LAUNCH VEHICLE

A4.2.1 Purpose

The purpose of this paragraph is to describe the input data requirements for dynamic simulation modeling of a launch vehicle.

A4.2.2 Scope

The scope of the description is limited to the cluster of Saturn V F-1 engines and associated component parts of the system that are required at ignition. The major areas of interest are:

a. Propellant tanks.
b. Propellant piping.
c. Propellant pumps.
d. Propellant valves.
e. Injection system.
f. Thrust chamber.
g. Gas generator.
h. Propellant pump turbine.
i. Pressurant system.
j. Ignition sequencing.
k. Thrust control.
l. ESE input/output.

The treatment here is necessarily brief. For more detailed information on actual F-1 engine simulation see Reference 1.

A4.2.3 Data Resources

In order to simulate a system it is essential to know what the system is and have access to detailed information that describes the component parts of the system, the functional characteristics of the component parts, and the functional interrelationships between component parts. In the conceptual and definition phase of a program, much definitive information is missing, hence systems and component parts must be proposed, analyzed and adjusted in an iterative process until a fully describable system is laid out. The Saturn V system is presently in the hardware stage, so the systems information is, or at least should be, available from which a simulation can be constructed.

The sources from which information must flow in order to have an accurate simulation are as follows:

a. Systems descriptions.
b. System flow charts.
c. Schematics (electrical and mechanical).
d. Locations.
e. Nomenclatures.
f. Change information.
g. Personal contacts.
h. Text books, etc., for theoretical and/or empirical information.
i. Test results.
j. Test procedures.
k. Operating description and procedures.

A4.2.4 Saturn V Propulsion System

The basic components and flow paths for the F-1 engine are shown in Figure A-19. The four remaining engines (not shown) are identical. The entire cluster has individual piping going to the propellant storage tanks. That is, the LOX tank and fuel tank act as propellant headers as well as storage tanks. Each engine supplies a proportionate share of hot Helium and GOX for the pressurant systems. That is, the turbopump discharge heat exchanger of each engine generates hot Helium and GOX, which is admitted to common gas headers. Helium is bled from its header to the fuel tank for controlled pressurization, and GOX is bled from its header to the LOX tank for controlled pressurization. In each case the bleed is regulated to hold proper pressurization levels.

The engine startup sequence is shown in Figure A-20. A similar sequence of course pertains for each engine. The start time for each engine is controlled from the launch sequencer. Figure A-21 shows generally how the ESE discrete events control engine startup from time for ignition to launch commit.

A4.2.5 Simulation

It is an objective to make the simulation as complete as possible within the confines of the part of the system chosen, and within the constraints imposed by limitations on time, systems information, and computational capability. The results will demonstrate: (1) Utilization of data resource; (2) Modeling methods; (3) Programming methods; and (4) Utilization of simulation.

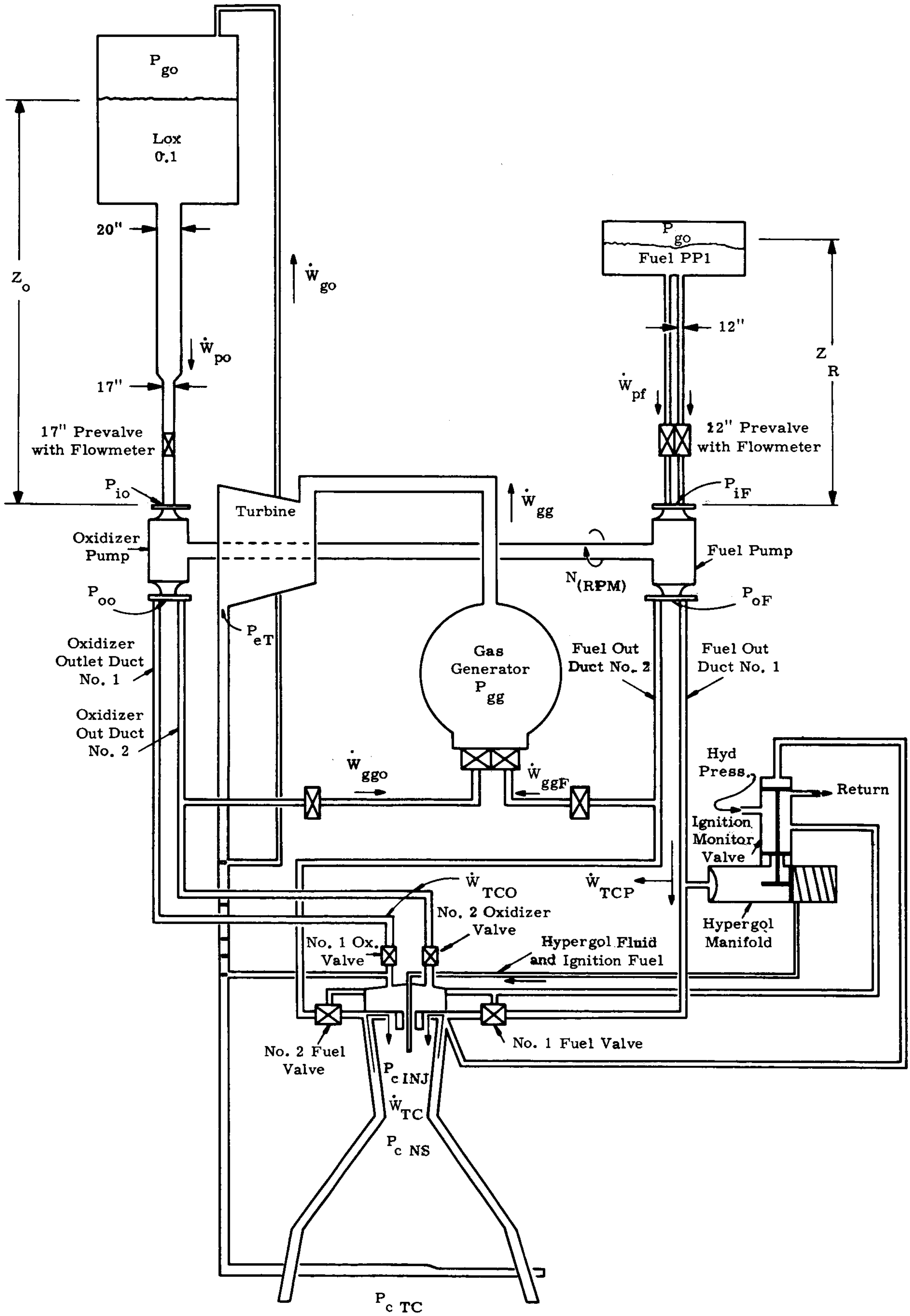


Figure A-19. Propulsion System

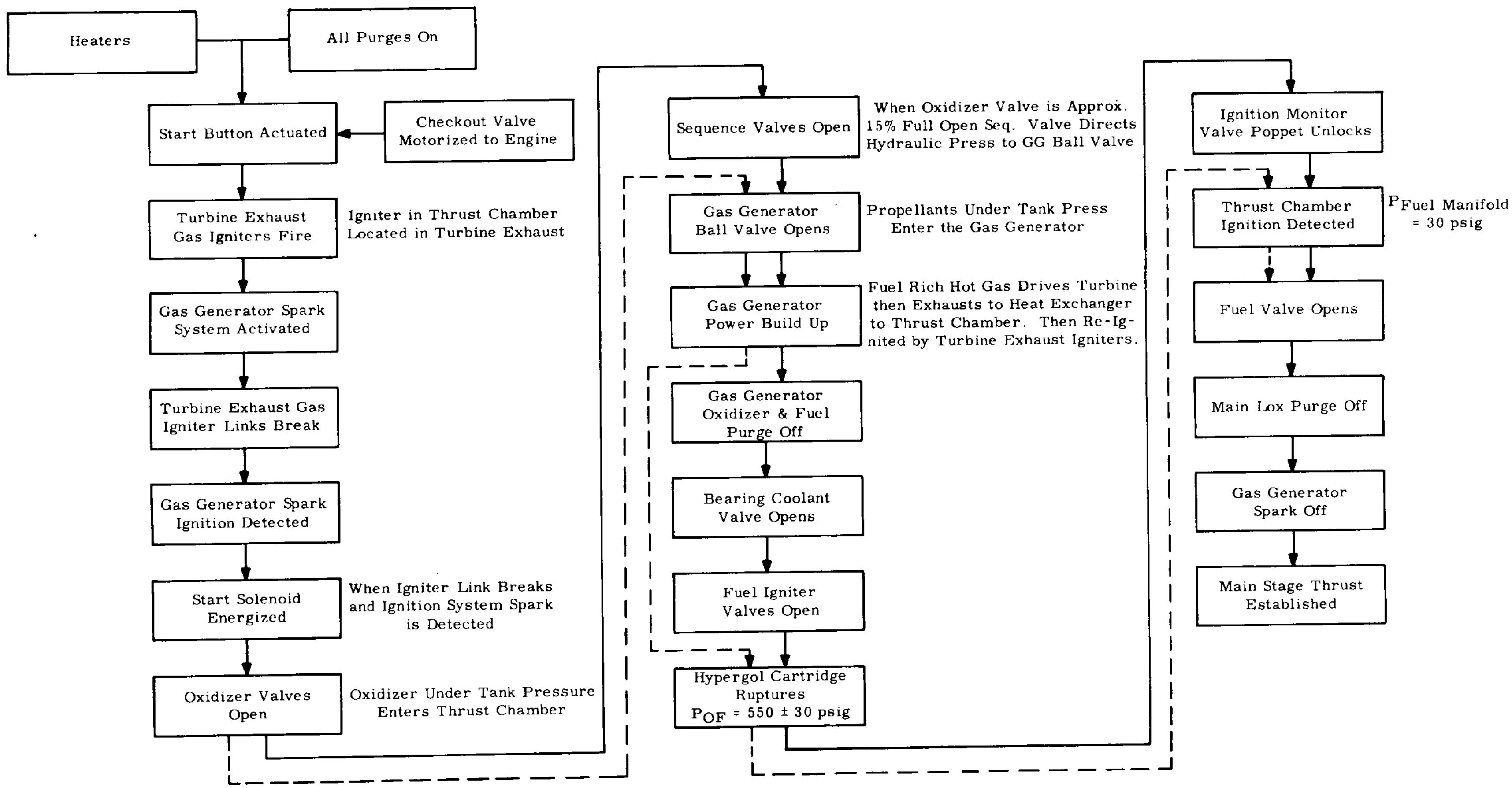


Figure A-20. Engine Starting Sequence

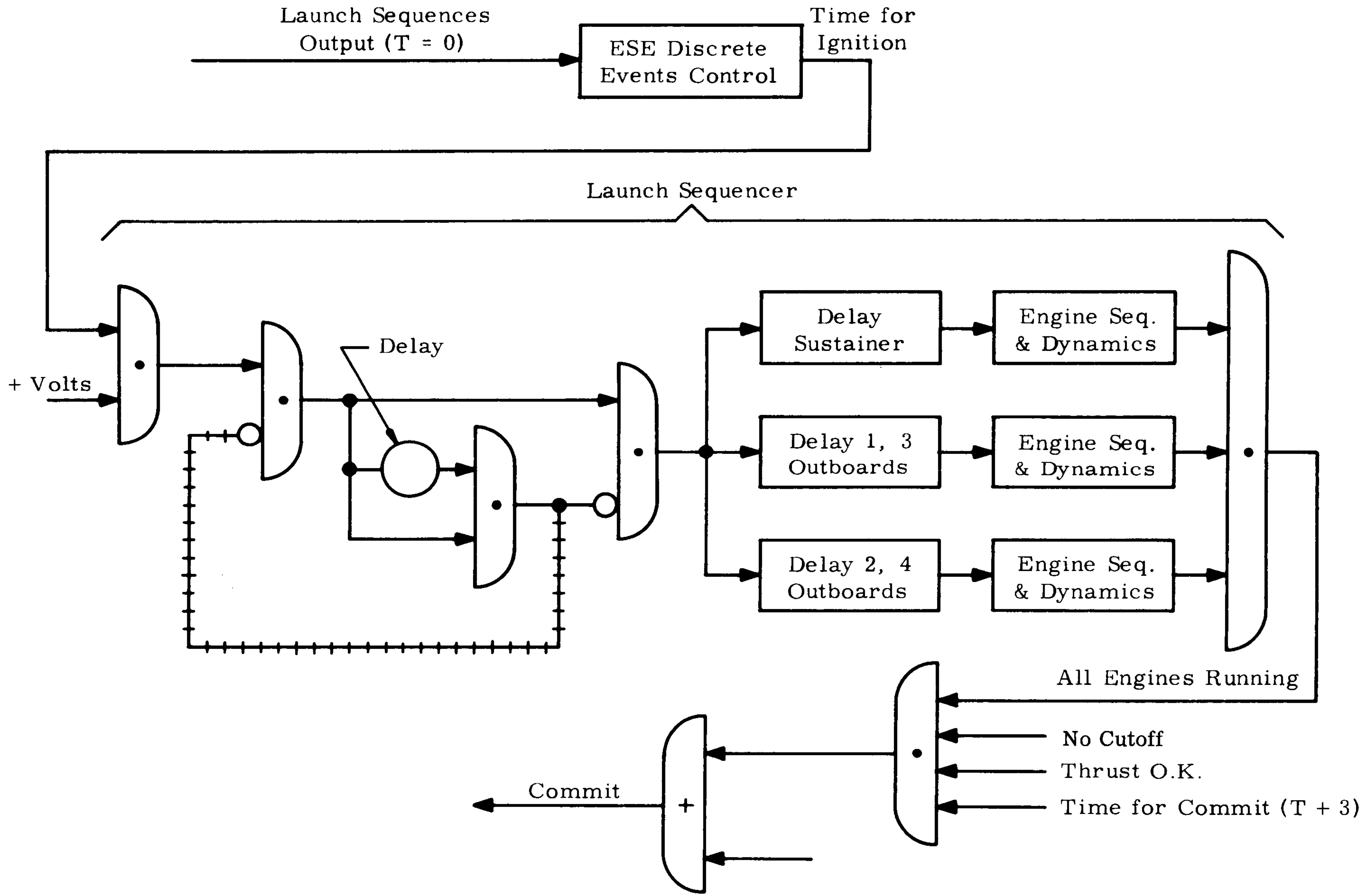


Figure A-21. ESE Ignition Control

Figure A-22 shows generally the confines of the system to be simulated. It will include ESE and engine sequencing from time for ignition to launch commit and engine static and dynamic characteristics in the areas of electrical, mechanical, hydraulic, and thermal systems. It will not include such things as structural dynamic interaction nor the effects of gimbaling and guidance. The gross effect of acceleration head in the fluid system following liftoff will of course be included, but the effects of propellant sloshing and gyroscopic effects in the fluid system could not realistically be considered at this time.

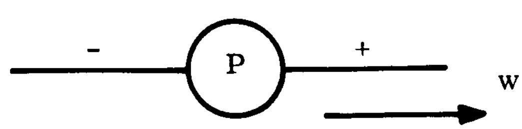
Figure A-23 is a flow diagram for the S-IC stage propellant fluid system. The transfer function (or impedance concept) notation, and network diagraming is used so as to emphasize the functional character of components and interrelationships. Figure A-24 shows the gas generator, turbine, heat exchanger system, and a formulation for power generation and transfer, and heat transfer to the pressurant system. Figure A-25 shows the engine combustion chamber and thruster along with a set of pressure, temperature, and thrust equations.

Figures A-23, A-24, and A-25 are presented to emphasize the source and nature of much of the input data required for launch vehicle simulation. The extent of the system presented here is of course represented in terms of a system of nonlinear differential equations. Additional features, such as sequencing, controls, etc., would of course introduce input requirements in such form as Boolean equations, function generation, and look-up tables.

Figures A-24 and A-25 are conventional presentations of thermal power systems. The system of equations is based on the listed references.

Figure A-23 is based on network form with each element identified in terms of its appropriate functional characteristic. The types of elements indicated are listed below:

a. <u>Pure Pressure Source</u>



P is the internal pressure level of the element. It acts as a driving source and has no internal impedance.

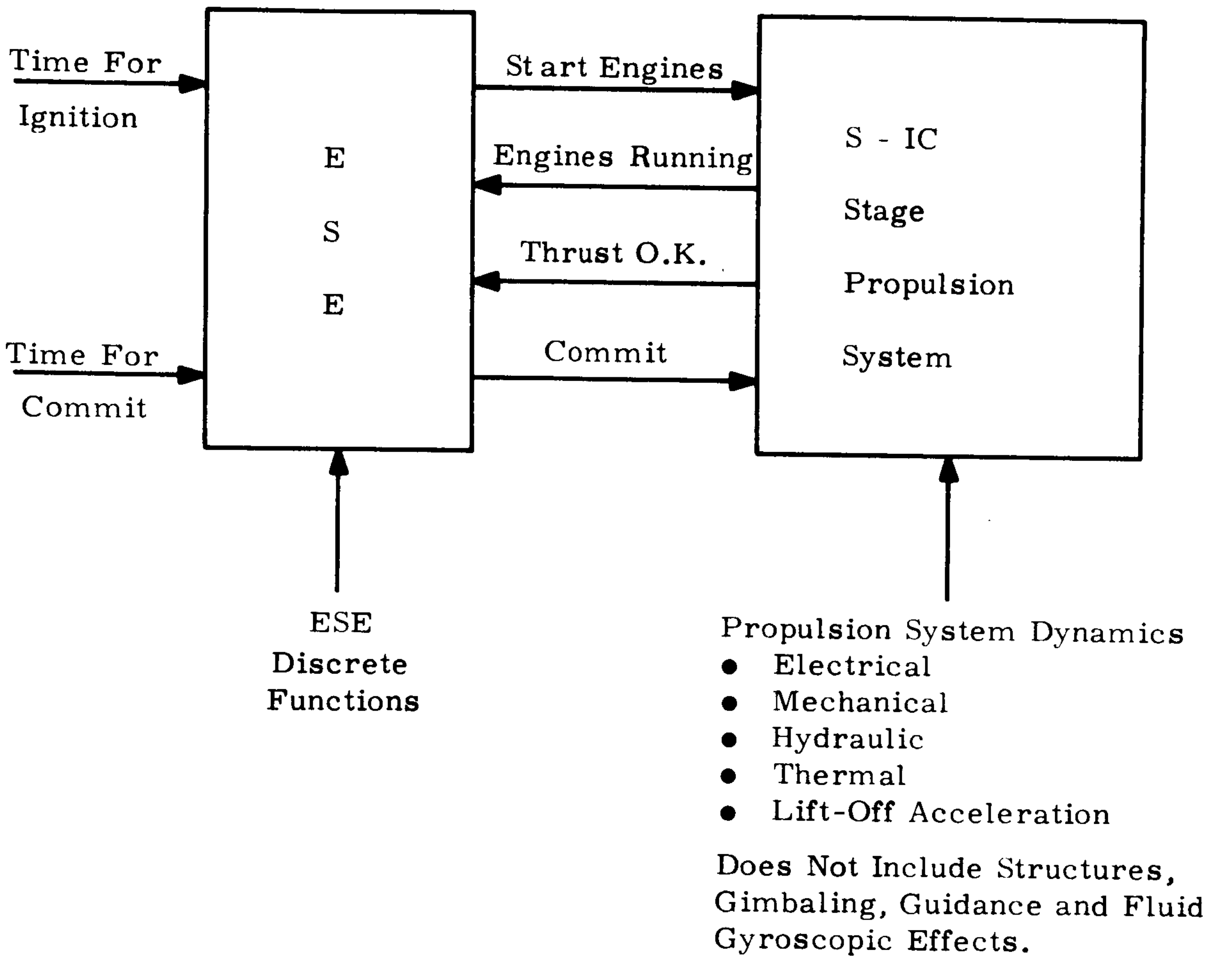


Figure A-22. System Boundaries

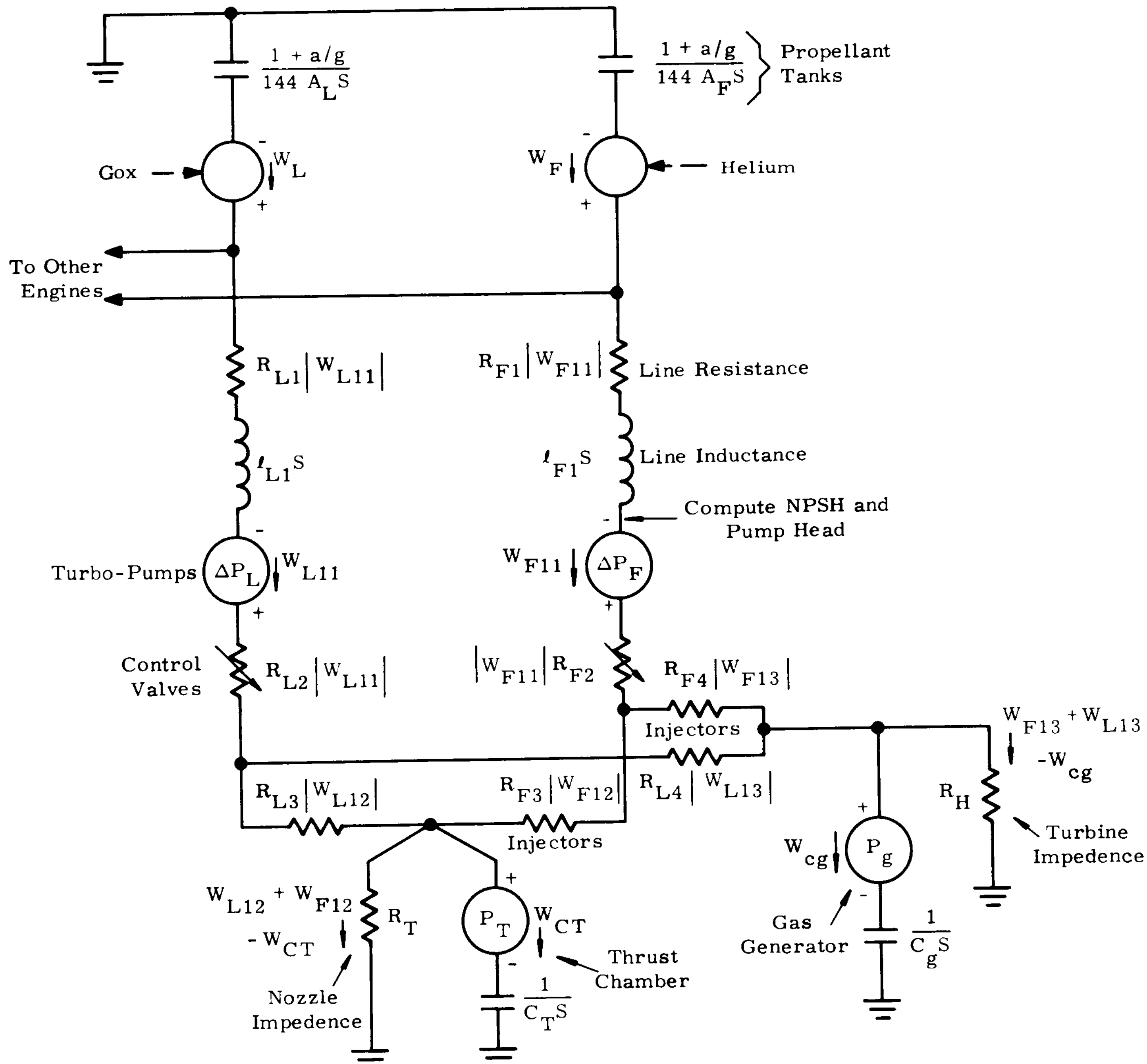
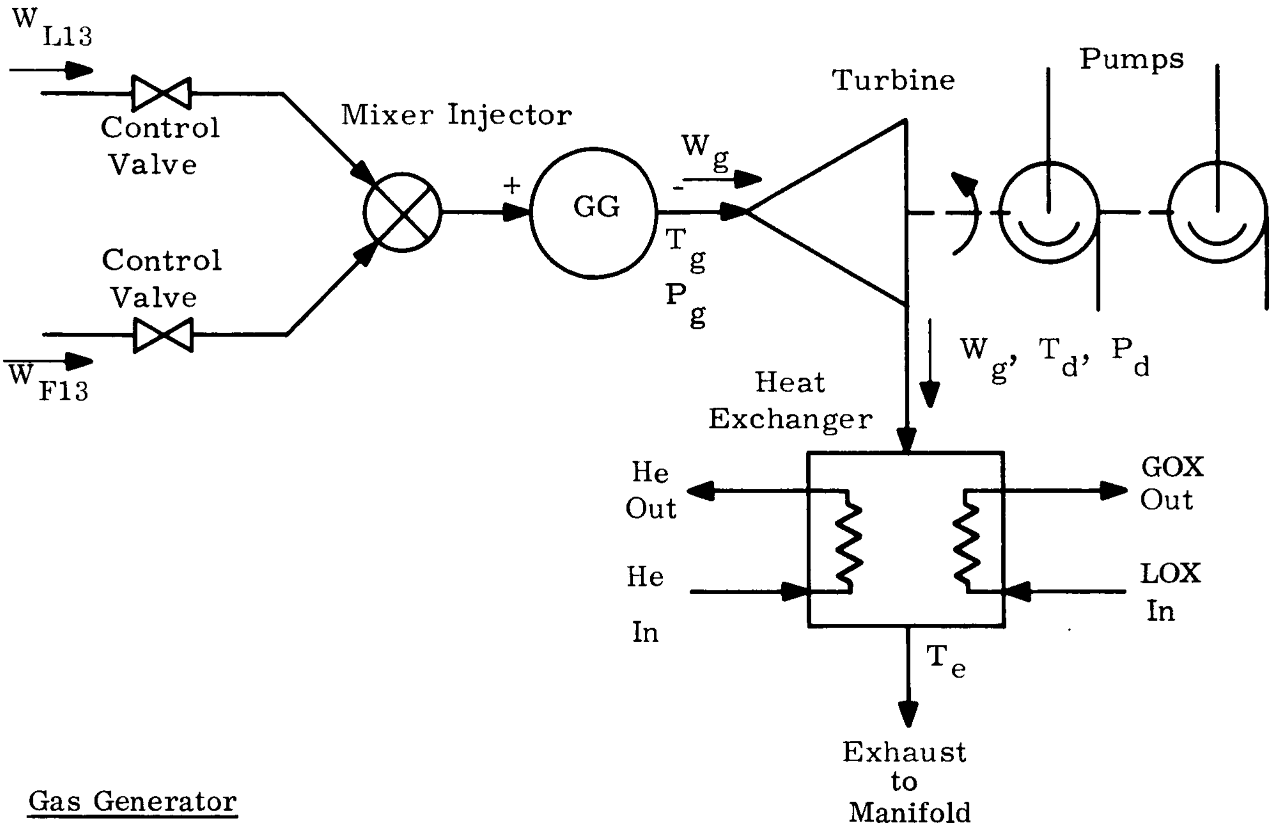


Figure A-23. S-IC Stage Propellant Fluid System



Gas Generator

$$T_g = 248 + 2810\ (W_{L13}/W_{F13})$$

$$P_g/P_d = 12.04 + 14.05\ (W_{L13}/W_{F13})$$

Turbine Power

$$I \frac{dn}{dt} = \text{Power In} - \text{Power Load} - \text{Losses}$$

$$\text{Power In} = (W_{F13} + W_{L13}) \left[C_p T_g \left(1 - \left[\frac{P_d}{P_g} \right]^{\frac{\gamma - 1}{\gamma}} \right) \right] \epsilon_t$$

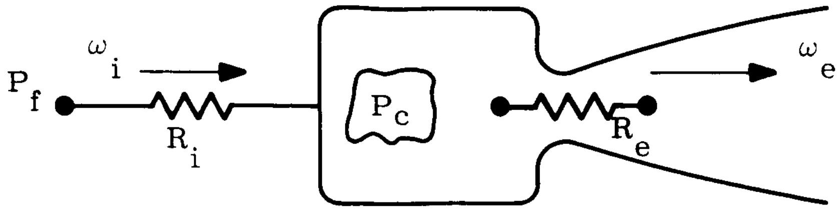
$$\text{Power Load} = \frac{(\Delta P_L)\ (W_{L13})}{778\ \epsilon_L} + \frac{(\Delta P_{F1})\ (W_{F13})}{778\ \epsilon_F}$$

$$\text{Losses} = \text{Windage} + \text{Friction}$$

$$C_p = 0.536 + 0.45\ (W_{L13}/W_{F13}) - 0.4\ (W_{L13}/W_{F13})^2$$

$$\gamma = 1.025 + 0.24\ (W_{L13}/W_{F13})$$

Figure A-24. S-IC Stage Turbopump Model



$$\frac{dP_c}{dt} + \frac{1}{\tau} P_c = \frac{1}{f_i \tau} \sqrt{\frac{R\,T_c}{\Gamma(\gamma)}}\ \omega_i$$

$$\Gamma(\gamma) = \gamma^{\frac{1}{2}} \left(\frac{2}{\gamma + 1}\right)^{\left[\frac{\gamma+1}{2\gamma - 1}\right]}$$

$$P_c v = R\,T_c$$

$$v = \frac{V_c}{M}$$

$$M = M_o + \int (\omega_i - \omega_e)\, dt$$

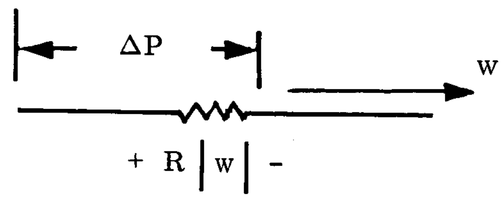
$$\omega_i = \sqrt{R_i (P_f - P_c)}$$

$$\omega_e = \sqrt{R_c P_c}$$

P_c	=	Chamber Pressure	ϵ	=	Efficiency
T_c	=	Chamber Temperature	a/g	=	Lift Acceleration Ratio
V_c	=	Chamber Volume	I	=	Turbine Moment of Inertia
M	=	Mass			
γ	=	C_p/C_v (Ratio of Specific Heats)			
R	=	Universal Gas Constant			
P_f	=	Line Pressure			
f_i	=	Nozzle Throat Area			

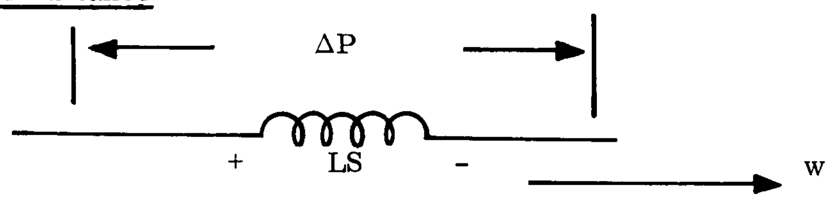
Figure A-25. S-IC Stage Thrust Chamber Model

b. Fluid Resistance



R is the resistance factor, w is fluid flow, P is the pressure drop induced by the element. The notation of resistance factor times absolute value of flow indicates that the impedance varies with flow. This in essence is square law hydraulic. The absolute value of flow must be carried to be able to handle flow reversal.

c. Fluid Inductance



The impedance LS specifically states that the pressure drop is proportional to the time rate of change of fluid flow. This is required to account for the change in kinetic energy storage in the fluid mass. The coefficient L is the equivalent length to area ratio of the piping system over the nodal length chosen.

d. Capacitance

Capacitance appears in two forms:

(1) That which is attributable to the static height of fluid in a standpipe.

(2) That which is attributable to fluid (or gas) compressibility.

The standpipe capacitance in this case is the surface area of the propellant tanks. It is of interest to note that the surface areas (A_L and A_F) may vary as a function of propellant level and that the apparent area varies as a function of missile acceleration (a/g) and gravity.

The other form of capacitance arises from the compressibility of the gases in the combustion chamber and in the gas generator. The coefficient is a function of the compressibility of the gas and completes the flow balance in the system.

The variable resistances indicate valves in the system that are operated through sequencing or continuous control.

The pressure heads indicated in the system arise from propellant tank pressurization (P_G and P_H), turbopump pressure head (ΔP_L and ΔP_F), and combustion pressure (P_T and P_g).

The representation is a true system balance in terms of conservation of mass, momentum and energy. It permits ready identification of the input functions and input data required for system simulation.

Furthermore, it accurately depicts interrelationships and connection points.

A4.3 RESOLUTION OF TWO PROBLEM AREAS THAT INVARIABLY OCCUR IN DYNAMIC SIMULATION

A4.3.1 Purpose

The purpose of this subsection is to point out two problem areas that invariably occur in dynamic simulation and how they will be handled in the proposed mechanical systems simulation.

A4.3.2 Problem Areas

The two problem areas are:

a. Numerical integration on digital computer.
b. The appearance of algebraic loops that cannot be handled by the computer.

These two areas are presently being investigated and the method of handling them will be recommended.

A4.3.3 Numerical Integration

The present methods of handling numerical integration in dynamic simulation generally do not perform satisfactorily in terms of response to very fast transients and total dynamic range that can be handled in a given problem. A proposed new method is discussed in detail in paragraph A4.6. A test problem has been solved using this method. The one selected was chosen for the specific purpose of addressing the method to a problem of maximum transient difficulty. Details of the test problem, and the various methods by which it has been solved, and the results, are presented in detail in paragraph A4.5.

It is interesting to note that the method relies on using the function and its functionally developed first three derivatives to give a stable and accurate calculation. The results of the test case substantiate the stability and accuracy of the method. Computation time cannot be compared readily at this point, because of the computer used and the fact that it was programmed in ALGOL.

Computer programming and processing requirements appear to be consistent with other methods.

A4.3.4 Algebraic Loops

The appearance of closed algebraic loops will arise in sets of coupled equations that represent such things as electric ladder networks and equivalent networks in fluid systems (see paragraph A4.4). These loops are not true physical quantities. They arise from improper programming techniques and lack of attention to the fact that when the computer computes a quantity from some function the quantities that make up that function must have been computed previously. When an algebraic loop is programmed per se, it in effect instructs the computer to compute a quantity from a function in which all quantities have not been computed previously. As a result the computer may:

a. Give erroneous results.
b. Oscillate between functions.
c. Become completely unstable.

In any event, the machine will not give a correct result. Equations programmed in this manner are called intransitive - they have no beginning and no end.

The DYNASAR code permits programming of this kind, except that a small time constant must be inserted into the loop. This in effect slows down the feedback information, stabilizes the computation, and will give correct results within the limits of the error criterion. The price paid for this is very significant. This becomes apparent when one realizes that the computer will bog down on these loop computations. That is, the numerical integration code will select a computing increment smaller than the time constant chosen. The time constant must be chosen to be far less than any time constant which truly exists in the system, otherwise the transient results will be very poor. As a consequence, the machine will spend more time computing a fictitious quantity than it spends on the real thing.

The programming technique that must be adhered to in order to prevent algebraic loops is discussed in detail in paragraph A4. 4.

A4. 3. 5 Conclusion

It is believed that the items discussed above are of maximum importance in the area of programming for dynamic simulations. The method of numerical integration, if not already being used elsewhere, may well represent a significant advancement in the state of the art. The test case chosen clearly demonstrates the power of the method.

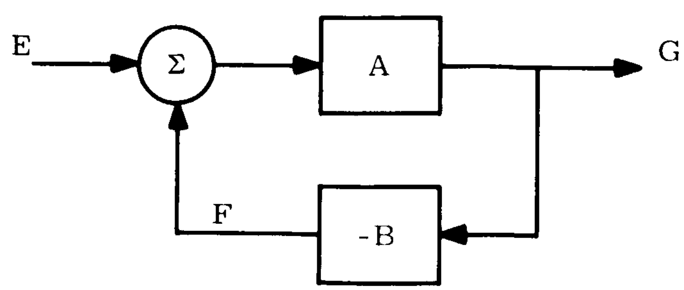
Attention to the appearance of closed algebraic loops is equally as serious as the matter of numerical integration. It is believed that if good programming practices are adhered to from the very start, they will present no problem.

A4. 4 ALGEBRAIC LOOPS

A4. 4. 1 General

Algebraic loops (that is, non-time dependent closed loops) will tend to appear whenever coupled sets of time dependent equations are solved. This generally arises from an attempt to solve the set without prior rearrangement. If the program is set up and algebraic loops are noted, then the program must be altered to remove the loops.

The problem with algebraic loops is that the computer is required to solve a set of equations, one at a time in which each equation has more than one unknown. This can be seen more clearly from the following example. Let the function be represented in block form as follows:



where

A and B = f(t)

Obviously one could write

$$G = \frac{A}{1 + AB} E \tag{A-59}$$

and program this equation. This would give the correct solution provided A and B are known, and $AB \neq -1$.

If on the other hand, the operation in Figure A-26 were programmed as shown, then the computer would have to solve the following set:

$$G = AE + AF$$

and

$$F = -BG \tag{A-60}$$

Each equation has two unknowns and the computer in general will not be able to find consistent values for F and G in both equations.

The block diagram or branch method of approach to formulation, which is frequently preferred by engineers, generally will lead to this situation. In the analog computer and DYNASAR type programming, this limitation will invariably appear. The only alternatives are:

a. Reprogram.
b. Insert a small time constant into each loop.

Methods of reprogramming are not always self-evident; hence, the second alternative is chosen. In analog computer programming, this problem is not always evident, since the manufacturer builds in a very small capacitor ($\approx$ 0.001 microfarad) around each amplifier to eliminate noise. This acts as a small time constant and permits calculation, provided a computational instability does not occur.

In the DYNASAR program (and in any digital computer for that matter) it is essential to insert a small time constant in order to force the computation to stabilize. This of course introduces time constants that are physically nonexistent. Worse than this, though, the small time constant will tend to control the integration step size for the entire program and hence slow down the computation rate to an undesirable point. This slowing down effect is not evident in the analog computer, since it computes on a parallel basis - it is very marked on digital computation because of the serial nature of performing operations.

An example of the type of systems that generate this problem and what procedures are required to avoid it is given in the following analysis.

A4.6.2 Simultaneous Equations

A common source of simultaneous equations is the electric ladder network. Figure A-26 shows a two-loop, R-L network with voltage generators.

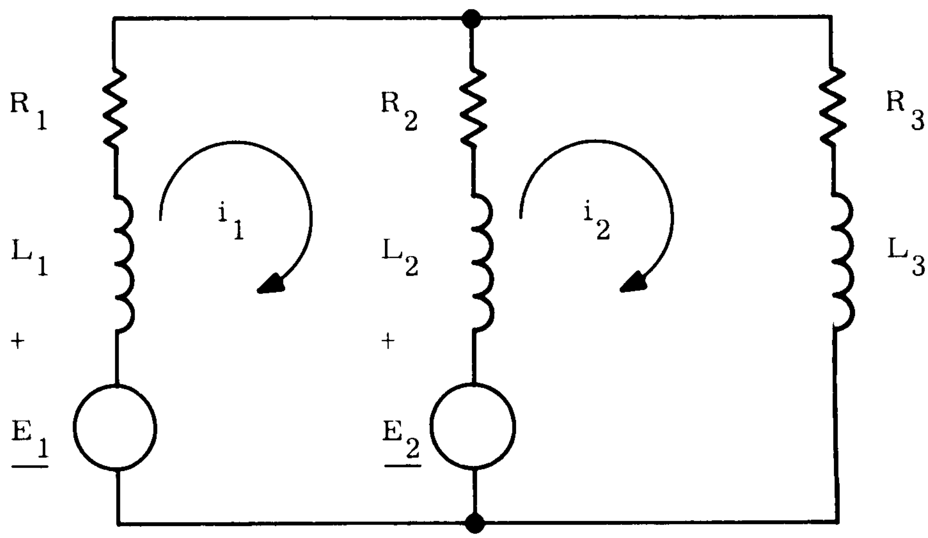


Figure A-26. Two-Loop, R-L Network

Two equations that properly describe the system are:

$$E_1 - E_2 = (R_1 + R_2)i_1 - (R_2)i_2 + (L_1 + L_2)\frac{di_1}{dt} - L_2\frac{di_2}{dt} \qquad \text{(A-61)}$$

$$E_2 = -R_2 i_1 + (R_2 + R_3)i_2 - L_2\frac{di_1}{dt} + (L_2 + L_3)\frac{di_2}{dt}. \qquad \text{(A-62)}$$

The typical analog approach would be to start by integrating di_1/dt in Equation A-61 and di_2/dt in Equation A-62. This gives i_1 and i_2, which can be fed back around, added to the voltages, and used to generate di_1/dt and di_2/dt. The primitive block diagram for this approach is shown in Figure A-27.

This leads to a closed algebraic loop that is considered undesirable. There are two alternatives:

a. Manipulate blocks to remove the loops.

b. Reform the first set of equations.

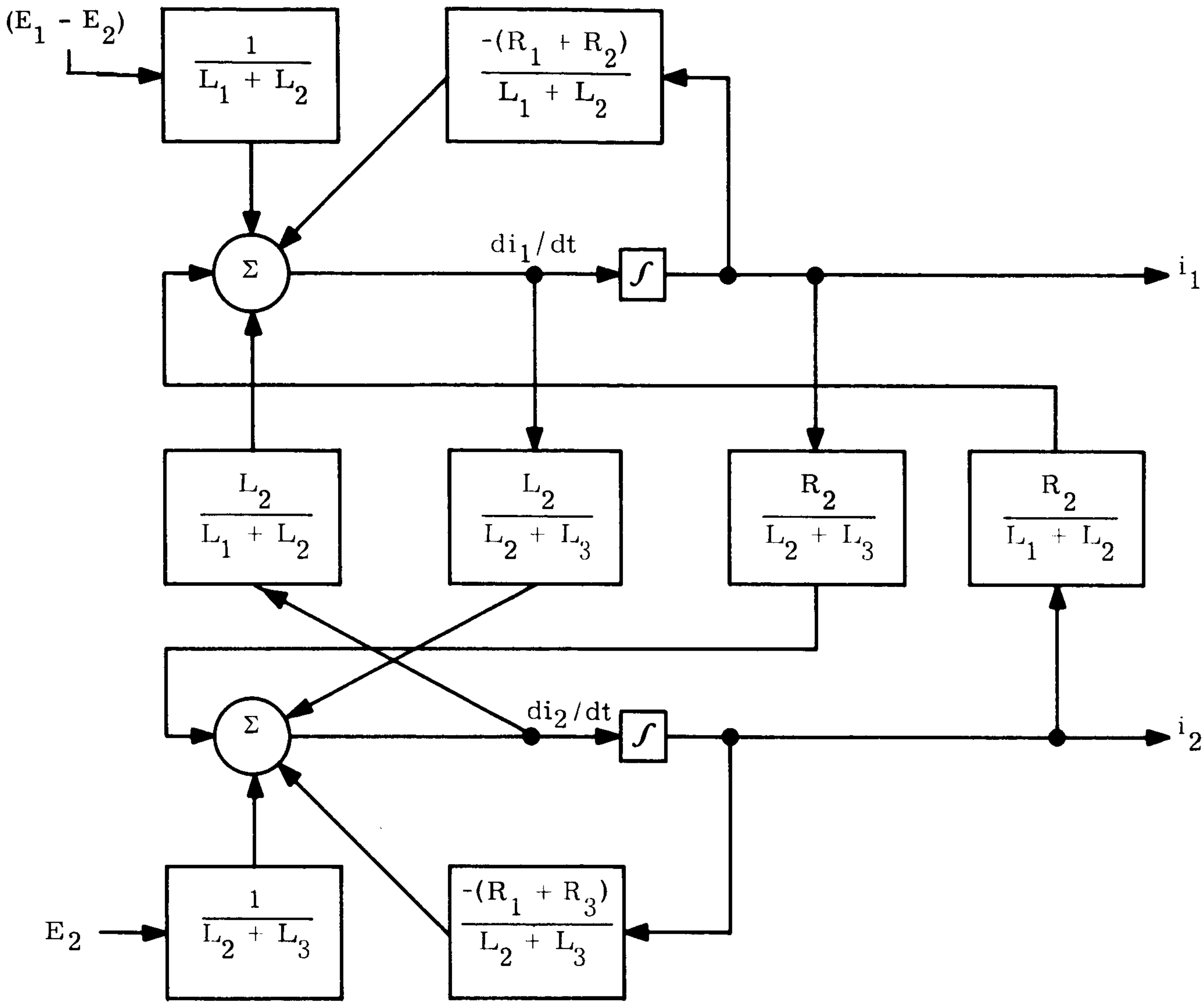


Figure A-27. Primitive Block Diagram for a Typical Analog Approach

The first approach is self-evident. The second is discussed below. (Note in either case much work is involved.)

a. Multiply Equation A-61 by $(L_2 + L_3)$ and Equation A-62 by L_2 and add:

$$
\begin{aligned}
(L_2 + L_3)\, E_1 - L_3 E_2 &= [\,(L_2 + L_3)\, R_1 + L_3 R_2]\; i_1 \\
&\quad + (L_2 R_3 - L_3 R_2)\; i_2 \\
&\quad + (L_1 L_2 + L_2 L_3 + L_3 L_1)\, \frac{di_1}{dt}
\end{aligned}
\qquad \text{(A-63)}
$$

b. Solve Equation A-63 for di_1/dt.

c. Insert this value of di_1/dt into either Equation A-61 or Equation A-62 and solve for di_2/dt.

The block diagram for this procedure is shown in Figure A-28.

There are no algebraic loops present in this system, hence the computer can handle it without having to resort to fictitious time constants.

It is evident that the problem becomes more laborious as the number of coupled equations increases, but, from the computer programming standpoint, there is no alternative.

It should be noted that the algebraic loop in Figure A-27 could be resolved by manipulating blocks. The procedure is indicated in Figures A-29a and A-29b (in which only the essential blocks are shown).

Figure A-29b shows that the algebraic loop has been eliminated. This setup is adequate for computer programming. This form can be further reduced to that of Figure A-28 by noting that the lower path couples to the upper path through block $L_2/(L_1 + L_2)$. This block can be eliminated by adding the inputs to the lower path to those of the upper path with the appropriate coefficients. The result is shown in Figure A-30.

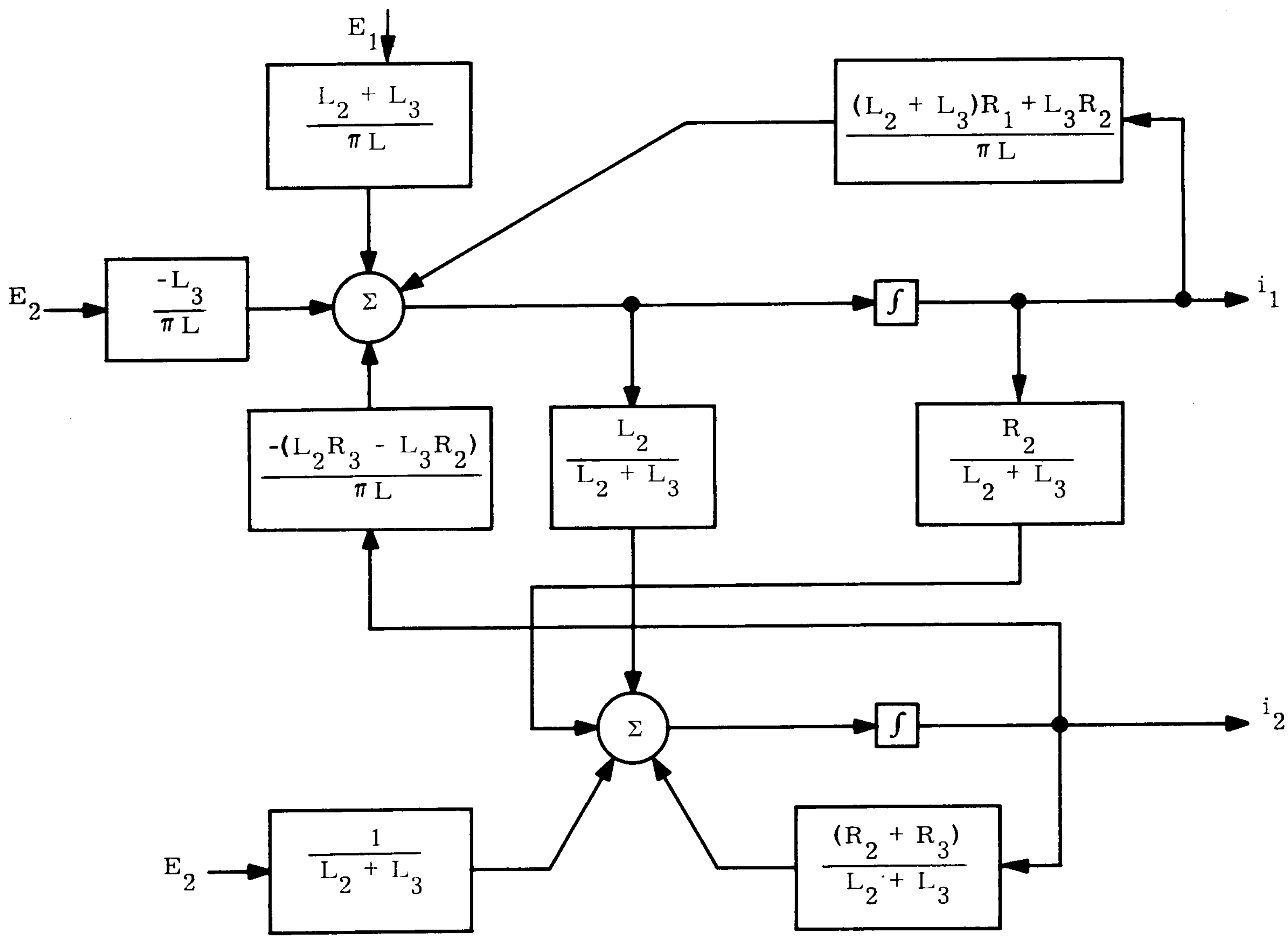
This can be reduced to an identity with Figure A-28 by moving the output coefficient $(L_1 + L_2)(L_2 + L_3)/\pi L$ backwards into each input branch. This is shown in Figure A-31.

The entire system may now be reduced to that shown in Figure A-32.

This probably represents the most symmetric form that can be achieved. The algebraic loops are completely eliminated and the coefficients are in the most easily maintainable form.

A4.5 NUMERICAL INTEGRATION

The literature of numerical analysis abounds in formulae for the solution of differential equations. In general these methods are derivable from manipulating a sequence of Taylor series approximations and applying the law of the mean, or by selecting an appropriate numerical integration technique and using Picard's theorem. The first class of methods is more popular probably because it does not require assuming the so called Lipschitz condition as does the second. Ince (Reference 14) shows that this



Note: $\pi L = L_1L_2 + L_2L_3 + L_3L_1$

Figure A-28. Block Diagram for Closed Algebraic Loop

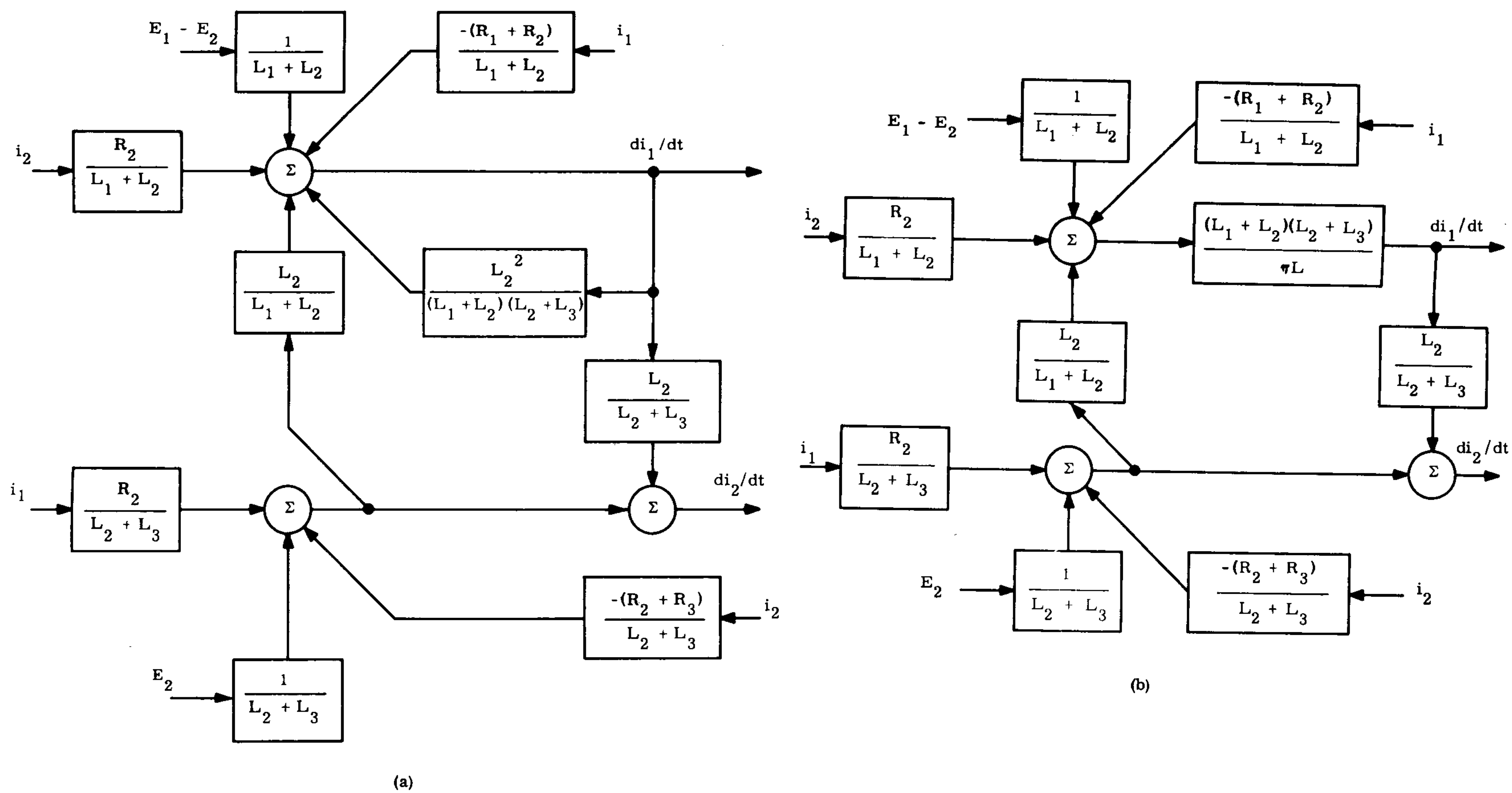


Figure A-29. Solving Algebraic Loop by Manipulating Inner Blocks

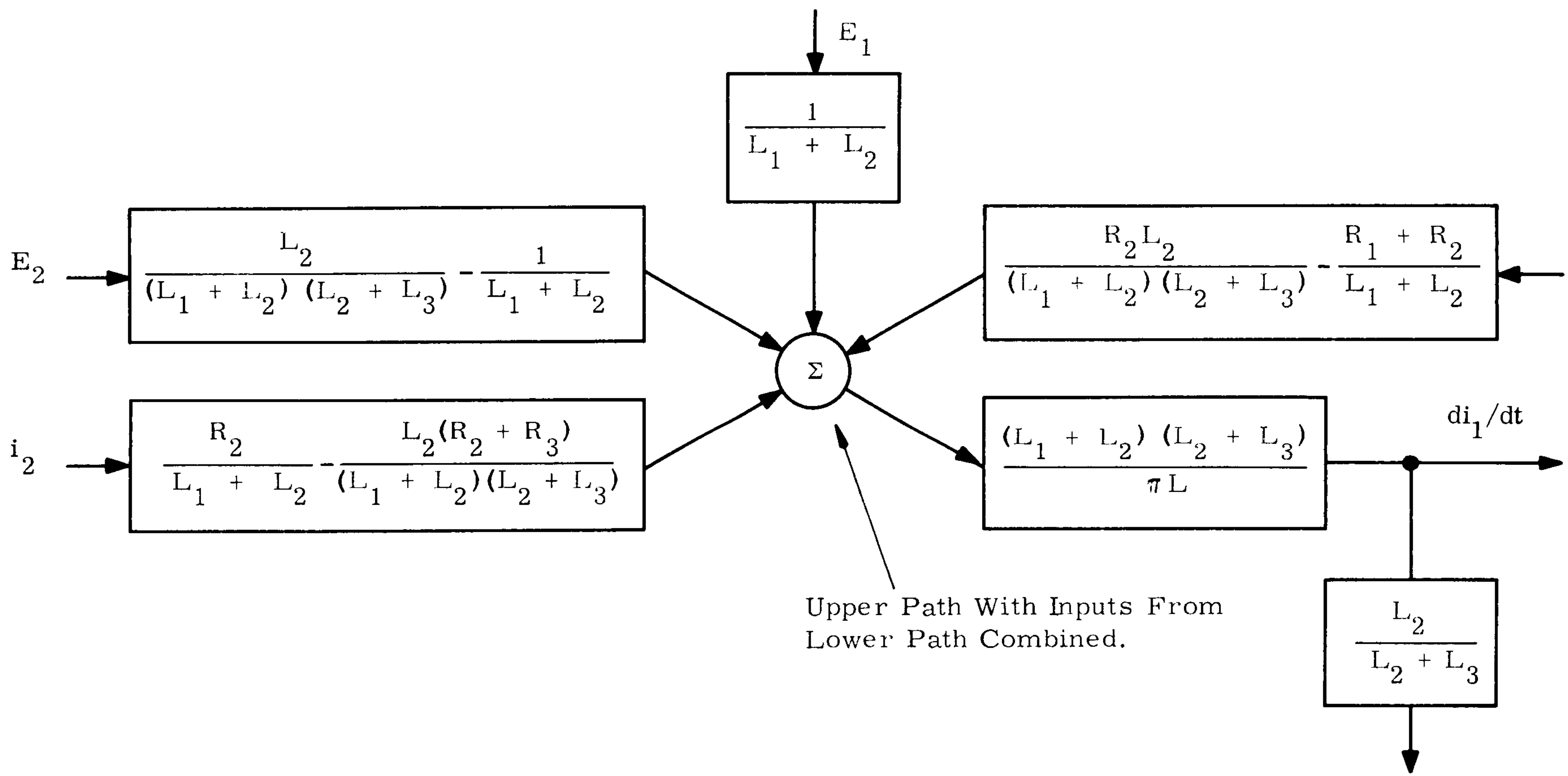


Figure A-30. Eliminating Loops by Combining Inputs

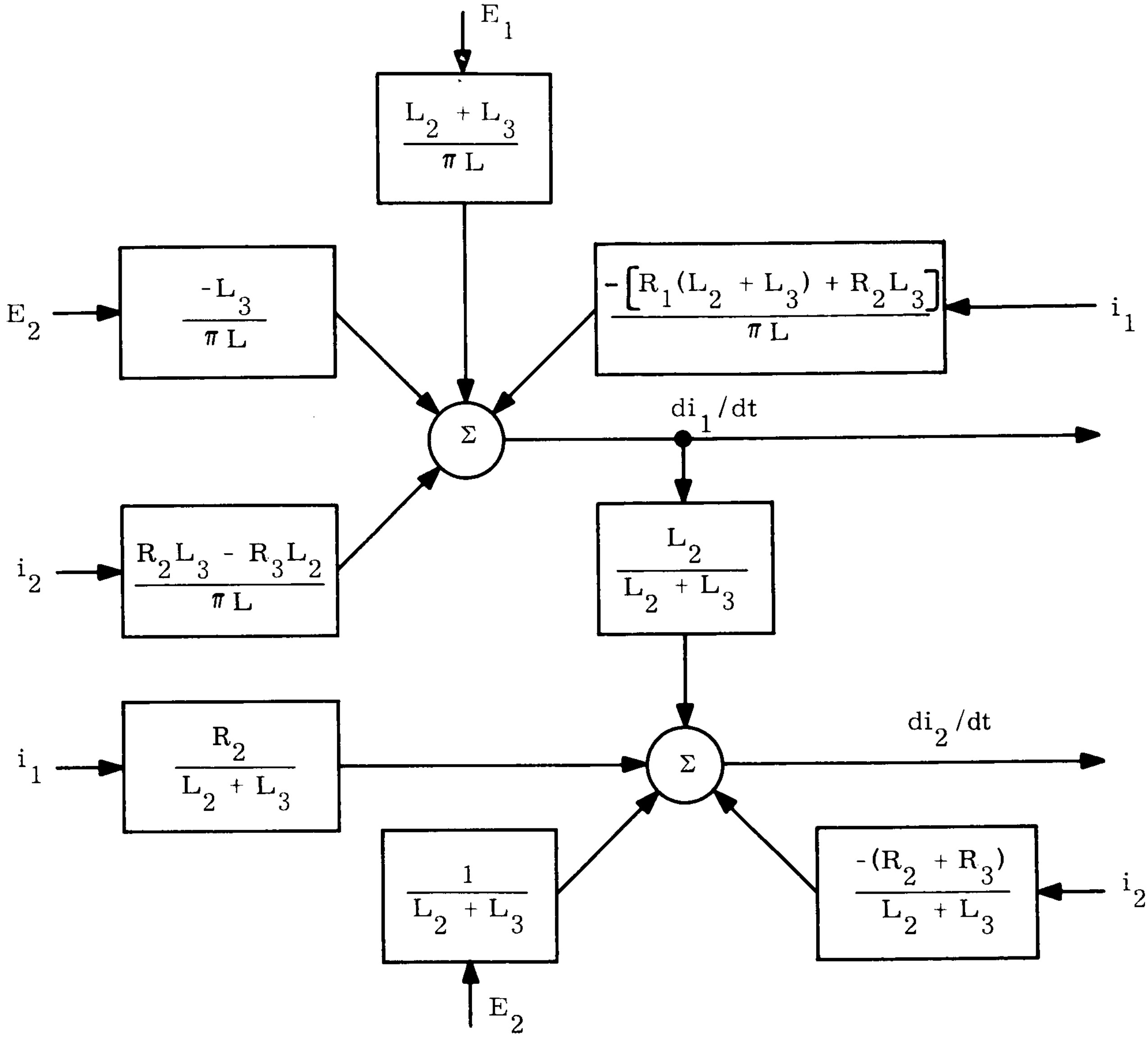


Figure A-31. Eliminating Loops by Algebraic Combinations

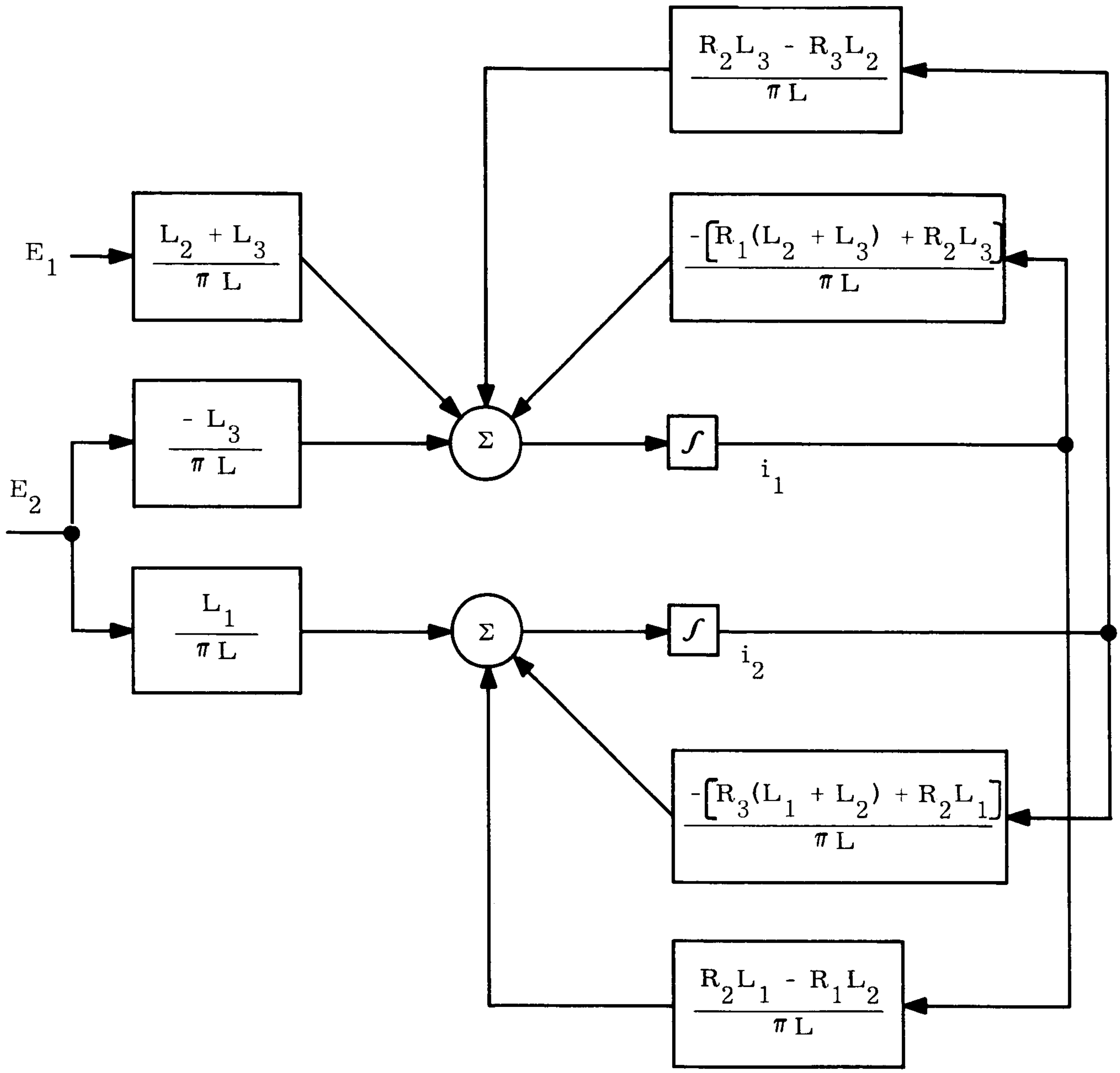


Figure A-32. Simplified Form of Diagram with Loops Eliminated condition can be weakened slightly, but this does not seem significant. Of minor consequence and justifiably beyond the scope of this section are the transform techniques that do not seem to generalize to nonlinear equations.

Most popular of the Class I techniques is the Runge-Kutta fourth order technique, probably because it is extremely economical with respect to computer storage. However, because the error term is of fifth order in the independent variable, multiplied by a very complicated function of second partial derivatives with respect to the dependent variables, one can often find oneself on treacherous ground by the indiscriminate use of this technique. The usual attempt to avoid this pitfall is to compute for h and h/2 simultaneously. It is not impossible for this procedure to lead to two wrong answers which check. This problem can be circumvented to some extent by suitable application of a corrector formula. However, no corrector seems to have the desirable storage saving properties.

The predictor-corrector techniques of Adams or Milne use first-order derivatives. Stability problems are about the same for each one. Neither is self starting and one is forced to compute several points initially before the continuation scheme can be used. However, the increased accuracy over Runge-Kutta four is often used to justify their use. A single application of a predictor-corrector scheme, however, does not promise a stable numerical solution. Doubling the interval size requires storing additional past history to accommodate potential use. Halving interval size is accomplished by some interpolation procedure at the expense of adding to the truncation error. Usually this is done by refining the last computed intervals. A much wiser approach would be to refine the central intervals of the past history set and integrate over the specified range using the reduced interval size.

A scheme that used higher derivatives has been chosen for study for the following reasons:

a. It is virtually self starting.
b. A halving formula of considerable accuracy is readily derived.
c. Doubling does not require carrying additional past history.
d. An optimum tradeoff seems to have been reached between accuracy and computation requirements if second and third derivatives are included in the integration formulae.

It is believed that the necessity of having available functional expressions for the second and third derivatives is not a valid objection. That is, predictor-corrector schemes using derivatives of order higher than the first must be considered along with the Milne and Adams methods. Also, iteration of the corrector seems to take less computer time.

In view of the infinite variety of integration formulae available for solving differential equations, it seems foolhardy for any one to make the statement that he has found a best method. Certain observations can be made, such as:

a. Higher derivative formulae are usually more accurate.
b. Corrector iteration can be used to monitor stability.
c. Formulae with high-order error terms have greater scope than low-order error formula.

The selection of a scheme for solving a system of one or more differential equation is almost entirely heuristic. Although it may be possible to find a better method for resolving a given set of equations, no one should ever assume he has found the best.

A4.6 AN ALGORITHM FOR SOLVING DIFFERENTIAL EQUATIONS USING A PREDICTOR/CORRECTOR TECHNIQUE BASED UPON DERIVATIVES OF HIGHER ORDER

The following paragraphs briefly describe an algorithm capable of solving nearly any system of differential equation. The algorithm consists basically of three fundamental formulas: a predictor, a corrector, and an interval refiner. These three formulas are derivable from the standard Taylor's series approximation. In the cases of the predictor and corrector, the origin is always taken as the last point computed. In the refiner, the computation results in a point at the origin.

The use of higher derivatives (second, third, etc.) with respect to predictor-corrector solution of differential equations has the primary advantage of being extremely economical with respect to the amount of computation involved for the accuracy attained. That is, the truncation error (which necessarily results from replacing an infinite process by a finite process) is considerably smaller than for the same amount of computation using any other approach (such as Adams-Moulton). For example: seventh order accuracy can be attained using third derivatives. The order increases by two for each additional derivative used for a 3 point predictor/2 point corrector technique.

One may look upon this algorithm as being concerned with three points: a previous value, a present value, and a next value. Like other predictor-corrector schemes, the algorithm assumes the previous y_0 and present y_2 values to have been computed. Using y_0 and y_2, the next value, y_4, is predicted This predicted value of y_4 is then corrected a number of times with respect to y_2. The difference between the predicted value and the (finally accepted) corrected value is compared. If this difference is less than k_1, the interval size may be doubled. And y_0 and y_4 are used for the next prediction to get y_8. If this difference is not less than k_1, but less than k_2, the interval is acceptable. y_2 and y_4 are used for the next prediction to get y_6. If both tests fail, then y_1 is computed and y_1 and y_2 are used to predict y_3. Figure A-33 illustrates this outer loop.

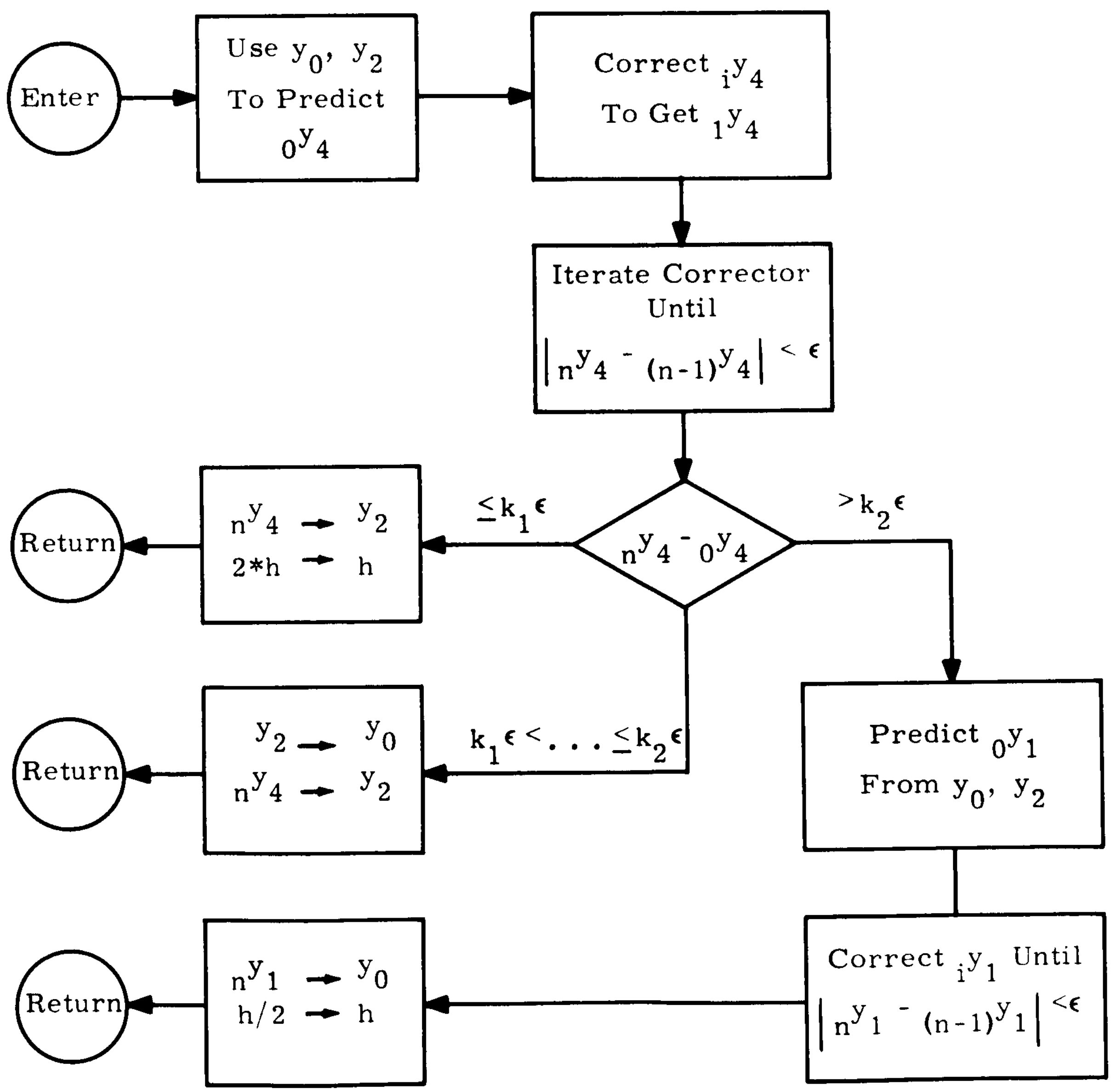


Figure A-33. Basic Outer Loop for a Predictor/Corrector Halver/Doubler Integration Algorithm

Let ϵ be a preassigned error value determined by the user of the algorithm. k_1 and k_2 depend upon the order, n, of the translation error: $k_2 \approx 2^n k_1$.

The following three formulas may be used.

The predictor:

$$y_{n+h} = y_{n-h} + 2h(4y'_n - 3y'_{n-h}) - \frac{2}{5}h^2(8y''_n + 7y''_{n-h})$$

$$+ \frac{2}{15}h^3(7y'''_n - 3y'''_{n-h}) + \frac{208}{100800}h^7 y^{vii}(\xi) \quad : \ P \quad (A-64)$$

The corrector:

$$y_{n+h} = y_n + \frac{1}{2}h(y'_{n+h} + y'_n) - \frac{1}{10}h^2(y''_{n+h} - y''_n)$$

$$+ \frac{1}{120}h^3(y'''_{n+h} + y'''_n) - \frac{1}{100800}h^7 y^{vii}(\xi) \quad : \ C \quad (A-65)$$

The refiner:

$$y_n = \frac{1}{32}(21y_{n+h} + 11y_{n-h}) - \frac{1}{32}h(16'_{n+h} - 6y'_{n-h})$$

$$+ \frac{1}{32}h^2(4y''_{n+h} + y''_{n-h}) - \frac{2}{96}h^3 y'''_{n+h} + \frac{20}{100800}h^7 y^{vii}(\xi) \ : \ R \quad (A-66)$$

If the independent variable is called x, then ξ is a value of x in the closed interval x_{n-h}, x_{n+h} and is usually different for each formula. Let the error from iterating C be ϵ. Then

$$P - C = 208\epsilon + \epsilon = 209\epsilon$$

$$k_2 = 209.$$

If C were to be applied to y_{n-h} instead of y_n to correct y_{n+h}, h could be replaced by $\frac{h}{2}$ in the predictor. In this case $P - C = \frac{208}{128}\epsilon + \epsilon = \frac{21}{8}\epsilon$. Under the assumption that ξ does not change very much between formulas $k_1 = \frac{21}{8} = 2.625$.

In the case of refinement $R - C = 21\epsilon$ using the above formulas. This does not seem significant since the coefficient of ϵ has just been reduced by a factor of $2^{-7} = 1/128$.

This method assumes that analytic formulas are available for each derivative in terms of lower order derivations and function values. Of course, decision functions may be included in the definitions in order to take into consideration nonanalytic points.

Storage requirements for this method are (per variable): 4 for y_{n-h}, 4 for y_n, 4 for ${}_o y_{n+h}$, and 4 for ${}_i y_{n+h}$ or 4 x 4 = 16 cells per variable. With proper programming no additional cells will be required for refinement.

Finally, since third order derivatives are supplied, one may use the assymptotic properties of the Taylor series. That is, assume the error to be approximated by the first neglected term, in this case $\frac{h^3 y'''}{6}$. Given the initial condition, y_o''', one can let $\epsilon = \frac{h^3 y_o'''}{6}$ and use the expression $y_1 = y_o + hy_o' + \frac{1}{2} h^2 y_o''$ as a predictor for y_1. That is, initially set $h = \sqrt[3]{\frac{6\epsilon}{y_o'''}}$. This should cause no difficulty since the algorithm will rapidly double to a reasonable interval size.

A4.7 TEST CASE FOR NUMERICAL INTEGRATION METHOD

The test case chosen was done so on the basis of presenting the method with a problem of maximum transient difficulty. The case was actually chosen in August 1963 at the Knolls Atomic Power Laboratory, Schenectady, N.Y. in order to check the capability of the AMSINT integration procedure used in the DYNASAR code. It is the typical nuclear reactor Kinetics equations as used in nuclear power plant systems studies. The model assumes two effective delay groups and zero prompt neutron lifetime. The forcing function simulates a 10-second withdrawal of the control rods at a constant rate. At 10 seconds the rods are suddenly returned to the starting point. Reactor power must take a sudden negative step at t = 10 seconds, then settle out to its final value.

The simplified set of equations that describe the reactor kinetics behavior are as follows:

$$(1 - \Delta k)\theta = \lambda_1 y_1 + \lambda_2 y_2 \qquad \text{(A-67)}$$

$$\frac{dy_1}{dt} + \lambda_1 y_1 = f_1 \theta \qquad \text{(A-68)}$$

$$\frac{dy_2}{dt} + \lambda_2 y_2 = f_2\theta \tag{A-69}$$

$$f_1 + f_2 = 1, \tag{A-70}$$

where:

θ = reactor power (per unit)

Δk = reactivity (per unit)

y_1, y_2 = delay group concentration

λ_1, λ_2 = delay group decay constant (sec^{-1})

f_1, f_2 = effective fraction of θ in delay group concentration

The form of the forcing function used is:

$$\Delta k = 0 \text{ for } t < 0$$

$$\Delta k = .01t \text{ for } 0 \leq t \leq 10 \text{ sec} \tag{A-71}$$

$$\Delta k = 0 \text{ for } t > 10 \text{ sec} \tag{A-72}$$

The exact solution for this equation is shown in Figure A-34. The result is from an analog computer that was set up at the time for a power plant study. The right-hand side of Figure A-34 shows the block diagram and equations. The lower left shows the input Δk. The upper left shows the response of θ, y_1, and y_2. The problem was run for for 40 seconds. The values of constants used are:

$$\begin{aligned} \lambda_1 &= 0.1352 \\ \lambda_2 &= 1.352 \\ f_1 &= 0.27 \\ f_2 &= 0.73 \end{aligned} \tag{A-73}$$

The λ_1 and λ_2 values used are larger by a factor of five than those found in nuclear tables. This results from the fact that the analog was scaled in time by 5:1 to increase the speed of computation. It does not affect the validity of the result.

The analog result indicates that reactor power increases on a generally exponential characteristic for 10 seconds. At 10 seconds, the power takes a step decrease because of the step change in Δk and then settles to its final value. Examination of the equations and block diagram show that the step change in power should occur and its magnitude is:

$$\Delta\theta = \theta\delta\Delta k, \tag{A-74}$$

where

$\Delta\theta$ = Step in power.

θ = Power at peak.

$\delta\Delta k$ = Step change in Δk.

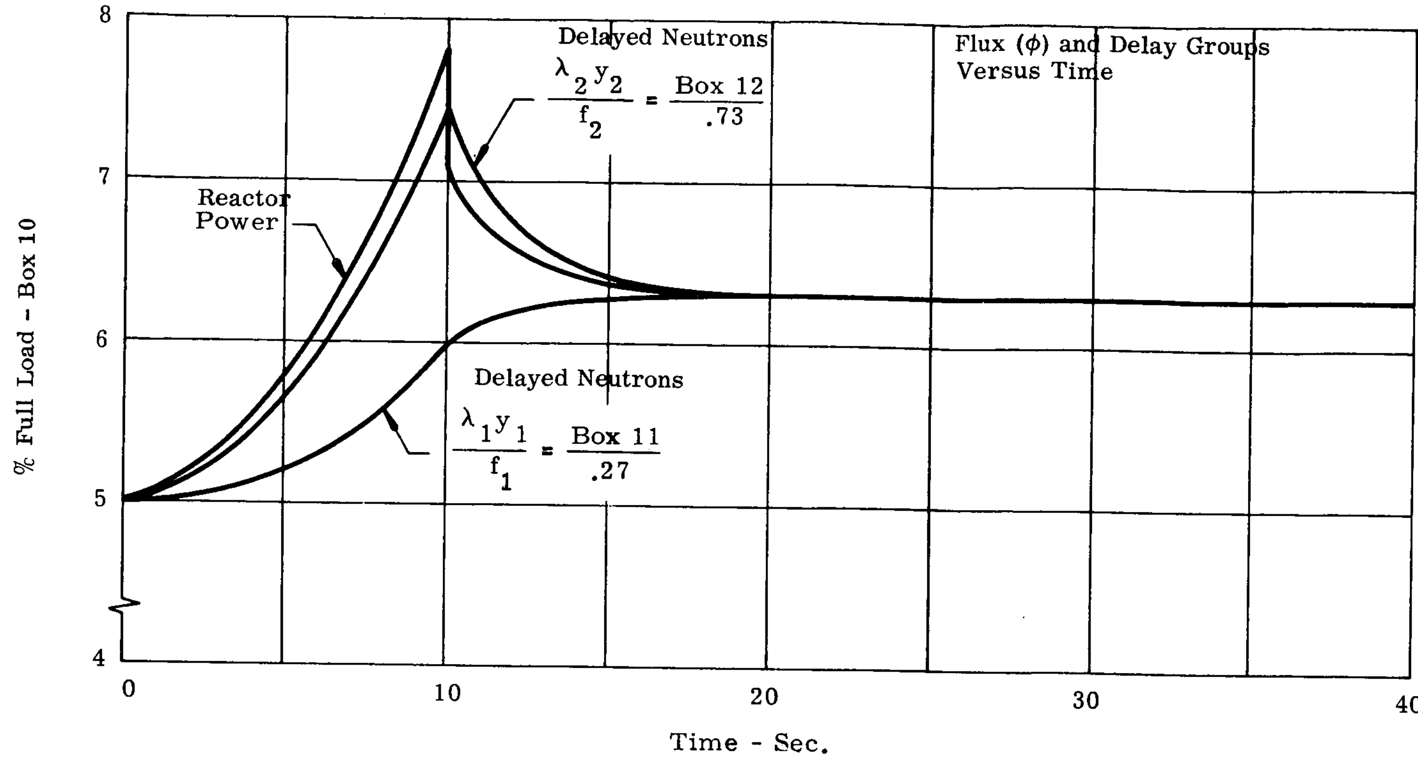
From Figure A-21, $\theta = 7.8$. The change in Δk is -0.1, therefore:

$$\Delta\theta = 0.1 \times 7.8 = 0.78 \tag{A-75}$$

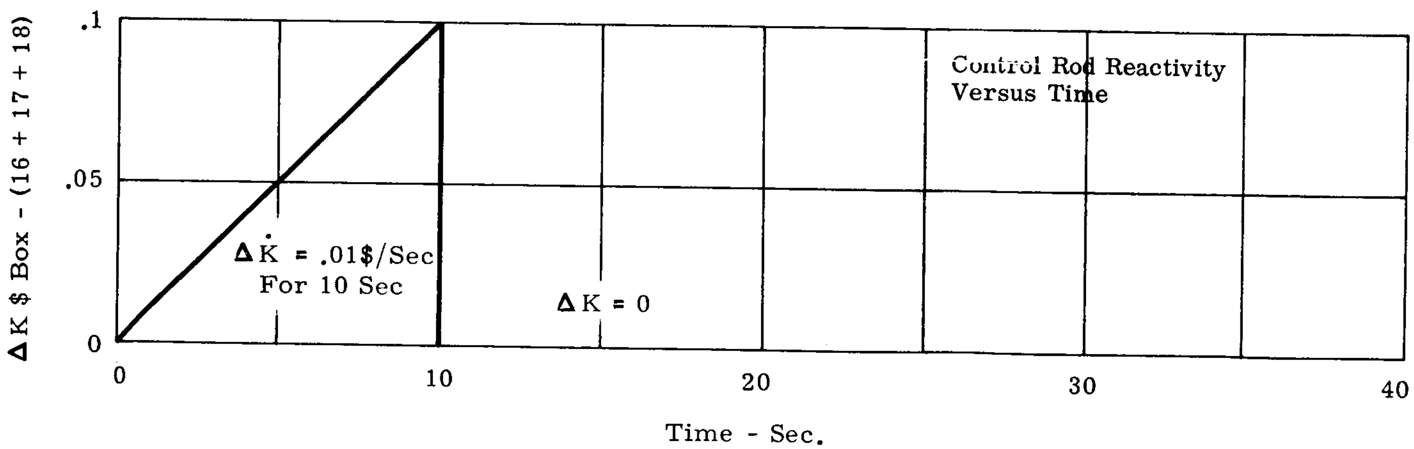
The value of θ just after the step is 7.02. The delay groups do not respond instantly because of the time constants in them. As time progresses from the step, θ, y_1, and y_2 decay exponentially to their final values as shown.

The problem also has been run on a digital computer using DYNASAR. The numbering system associated with the block diagram in Figure A-34 refers to the numbering system required by DYNASAR programming. The circled numbers refer to box type (time constant, summer, divider, etc.) while the noncircled numbers refer to box output number. The actual DYNASAR run was made on the IBM 704 located at General Electric Ordnance Department, Pittsfield, Mass., around February 1964. The results are not presently available, but it showed a truncation of the peak at $\theta = 7.6$ and a decided damped oscillation in response to the step. The truncation of the peak caused power to settle out erroneously at about 6.18. The time required to compute the function out to 20 seconds was estimated to be about 0.5 minutes. It was generally concluded at that time that the method of integration is not satisfactory for cases where steps (and near steps) are present. This was not an unexpected conclusion.

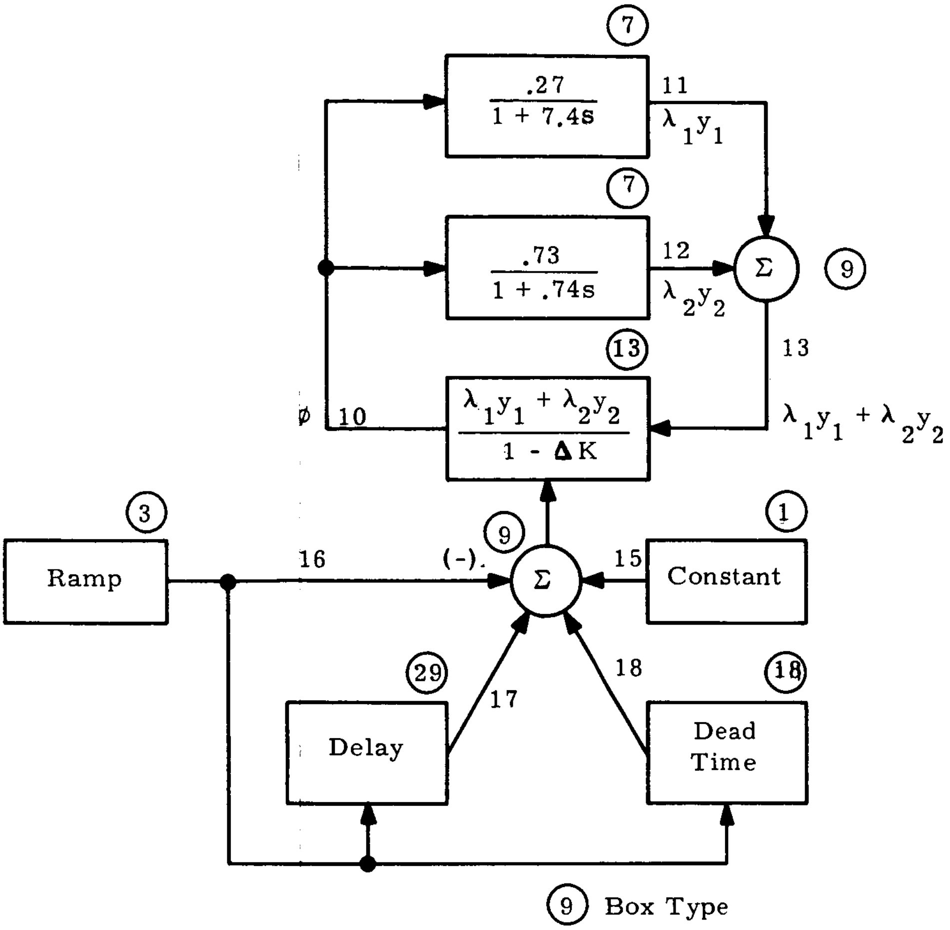
Finally, the method of numerical integration described in paragraph A4.6 of this report has been used to solve this problem. The result is shown in Figure A-35. Note that it is virtually identical with the analog result. The problem was programmed on a time-sharing GE-235/Datanet 30 configuration in Phoenix, Arizona, via teletype from Daytona Beach, Florida, using ALGOL language. Running time was about 0.5 minutes.



Flux (ϕ) and Delay Groups Versus Time
Delayed Neutrons
$\frac{\lambda_2 y_2}{f_2} = \frac{\text{Box 12}}{.73}$
Reactor Power
Delayed Neutrons
$\frac{\lambda_1 y_1}{f_1} = \frac{\text{Box 11}}{.27}$
% Full Load - Box 10
8
7
6
5
4
0
10
20
30
40
Time - Sec.



Control Rod Reactivity Versus Time
$\Delta \dot{K}$ = .01\$/Sec For 10 Sec
$\Delta K = 0$
ΔK \$ Box - (16 + 17 + 18)
.1
.05
0
0
10
20
30
40
Time - Sec.



Equations:

$$0 = \lambda_1 y_1 + \lambda_2 y_2 - (1-\Delta K)\,\phi \tag{1}$$

$$\therefore \phi = \frac{\lambda_1 y_1 + \lambda_2 y_2}{1 - \Delta K} \tag{2}$$

$$\frac{dy}{dt} = -\lambda y + f\phi \tag{3}$$

$$(S + \lambda)\,\psi(s) = f\phi(s) \tag{4}$$

$$\lambda\psi(s) = \frac{f}{1 + \frac{1}{\lambda}s}\,\phi(s) \tag{5}$$

$$\phi(t = 0) = 5;\ \lambda_1 y_1(t = 0) = 1.35;\ \lambda_2 y_2(t = 0) = 3.65 \tag{6}$$

$$f_1 + f_2 = 1 \tag{7}$$

$$\Delta K = 0 \text{ For } t < 0 \text{ and } t > 10 \tag{8}$$

$$\Delta K = .01\,t \text{ For } 0 \le t \le 10 \tag{9}$$

Figure A-34. Analog Results

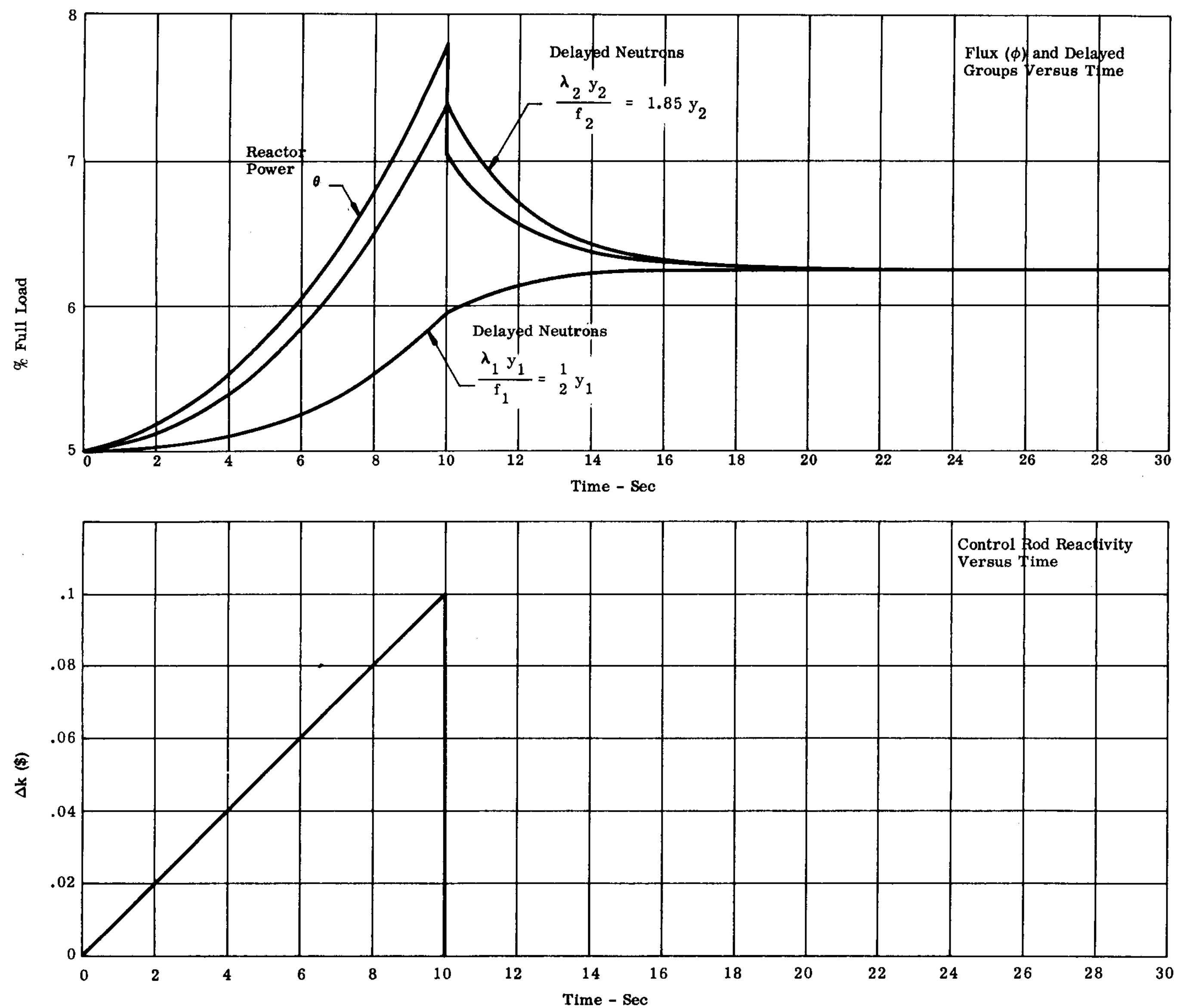


Figure A-35. Reaction Kinetics Response Using Numerical Integration Method of Appendix B

The machine computed the peak at:

$$t = 9.99951$$
$$\theta = 7.79836$$

The next computation was

$$t = 10.00000$$
$$\theta = 7.01825$$

There was no oscillation.

On the basis of this one problem, the indications are that the new method of paragraph A4.6 will prove to be superior.

A5 COMPONENT LEVEL SIMULATION DYNAMIC SYSTEM PROGRAM

A5.1 GENERAL

One of the techniques that shows promise of being an important factor in accomplishing dynamic simulation is the use of serial numbers or number series. To test its applicability quickly, a simple problem that could be checked by other methods was chosen. This was the thermostat problem, discussed in paragraph A2.10, for which serial numbers were generated. The results for the first interval are shown in Figures A-36 and A-37, which compare very well with their counterparts, Figures A-13 and A-14. Encouraged by this trial, work was started on a more sophisticated program for the number series application. Although not finished by the end of the contract period, the current status of the program is described in this section.

This program incorporates a main deck, six special subroutines, and one library subroutine. From the beginning of the programming task, a view to future usage was kept, but many portions of the program are tailored to the test case provided. Many techniques used are less elegant than desired and others are, quite candidly, "brute force" methods incorporated in the pragmatic sense that "they work." In each of these instances, it is visualized that improvements will be made.

The example problem supplied has been implemented and answers generated. The answers suffer some numerical loss, but this is being rectified.

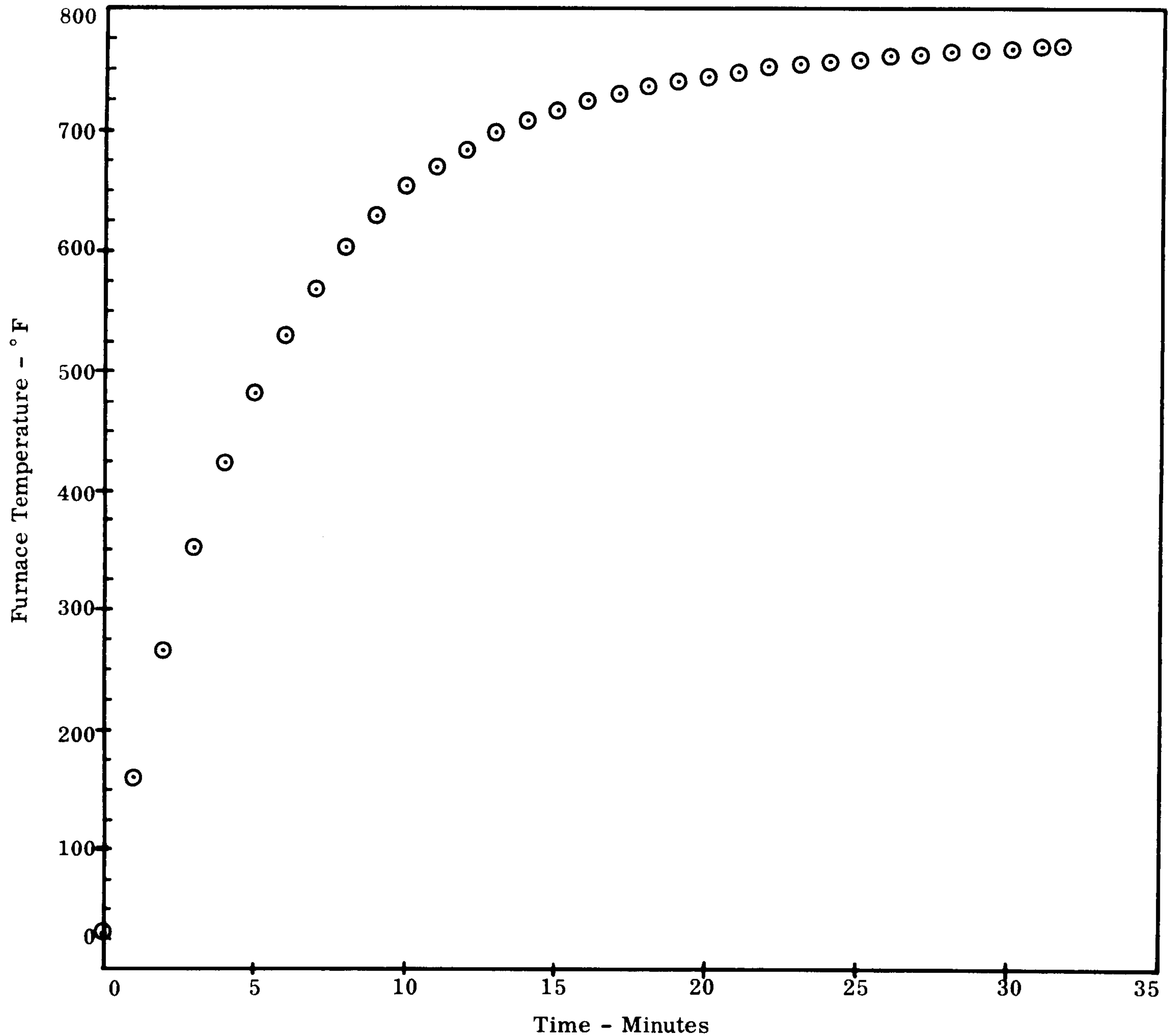


Figure A-36. Furnace Temperature Startup by Number Series

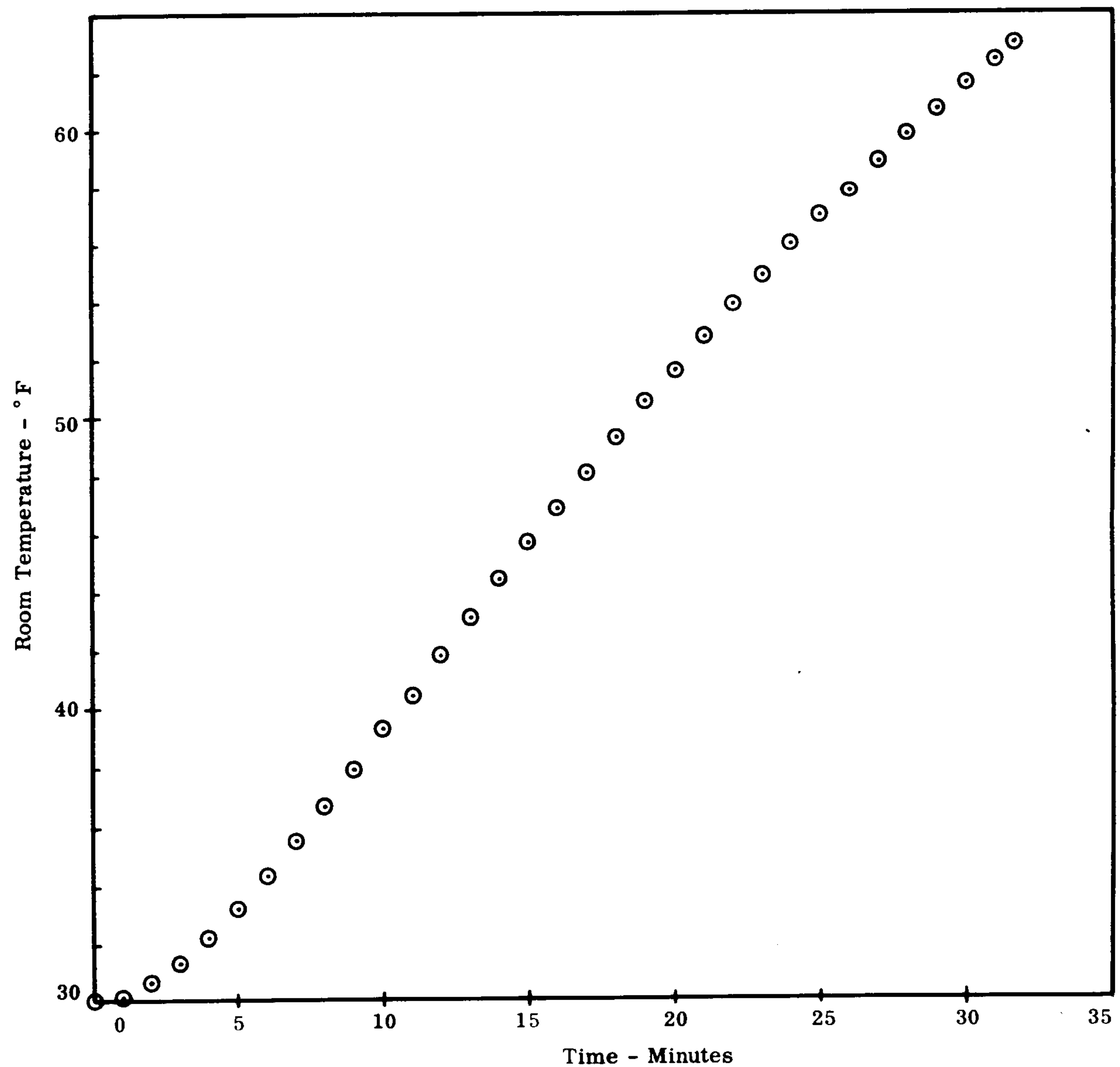


Figure A-37. Room Temperature Control by Number Series

The largest problem encountered is that of sufficient core storage. Storage used was 25,152 words including the IBSYS operating system. The example treated was, essentially, a seven node subsystem. Running time for solution was about 1.5 minutes.

The heart of the method is that of generating a transmittance path from an input node to an output node. This is accomplished in a subroutine named REDUCE. Inspection of the test case provided showed one or two interesting items.

a. Of the 49 possible transmittance branches in the 7 by 7 subsystem matrix only 10 were actually nonzero.

b. If the reduction algorithm were used in its present form, all elements of the subsystem matrix (hereafter referred to as the T matrix) would be manipulated. This would result in a great deal of wasted time inasmuch as the T matrix was (in this case) approximately 75 percent sparse (i.e., 75 percent of its elements were zero). Therefore, some method would need to be devised to manipulate only the nonzero elements. This was accomplished as follows.

   A one-dimensional array, the elements of which corresponded to a row of the T matrix, was generated. Then as an element of the T matrix, say t_{ij}, was read in and its equivalent number series formed, the j^{th} bit of the i^{th} word in the one-dimensional array would be turned on. This method is simplicity itself, but it does limit the size of the T matrix to a 34-node subsystem.

   The 75-percent sparseness also indicates a great deal of wasted core, so a method of storing only the nonzero elements in a linear array is being investigated for future work.

   The actual time sequence of calculation is as follows:

   (1) Read in the identification of the subsystem.
   (2) Read in initial conditions and testing values for the switching algorithm.
   (3) Read in a list of all external source nodes of the subsystem.
   (4) Read in a list of all internal source nodes of the system.
   (5) Read in an identification card containing the location of a transmittance branch in the T matrix.
   (6) Call the subroutine FORM and generate a number series representing that transmittance branch.

(7) Repeat steps (5) and (6) until all nonzero elements of the T matrix are formed.

(8) Store the T matrix on a peripheral device.*

(9) Form a subsystem submatrix for the first source (input) node.

(10) Reduce the submatrix to n equivalent transmittance branches, where n is the number of output nodes. Store these equivalent branches.

(11) Repeat steps (9) and (10) for all input nodes.

At this point of computation, the subsystem is completely described for further and future computational purposes. Unless the subsystem is physically altered, the previous steps will not have to be repeated. Therefore the equivalent transmittance branches may be stored permanently or until the subsystem is changed, in which case steps (1) through (11) will be used to update the description of the subsystem. For this reason, it is recommended that in the future the above steps be a separate program used only to generate the equivalent paths through a subsystem and to update the subsystem when necessary.

The remaining computational steps involve manipulating the generated equivalent paths, initial conditions, and the switching algorithm to describe the system response to a given set of inputs.

(12) Form all output node values for $t = t + k\Delta t$.

(13) If a switching condition has been met, go to step (15).

(14) Repeat steps (12) and (13) until switching is accomplished or other end conditions are met.

(15) Write out the output nodal values from $t = t_o$ until switching.

(16) Repeat steps (12) through (15) until a specified number of switchings have occurred.

The details of some of the 16 steps follow in the description of the subroutines.

A5.2 SUBROUTINE FORM

Subroutine FORM is essentially program 2313 of Reference 1. This program was modified for use on the IBM 7044 since it was originally intended for use on the

*Not implemented at this time but will be for larger problems.

IBM 7090. Certain other small modifications were made so as to make it compatible with the component level simulation main program.

The function of FORM is to read in the LaPlace transform of a transmittance branch in some subsystem matrix and to give an output of $L^{-1}\{F(s)\}$ sampled at some Δt between the limits $t = t_o$ and $t = t_f = t_o + k\Delta t$. The parameters t_o, t_f, Δt are specified in the input. The output of FORM will hereafter be referred to as number series. The output number series is assigned according to the location of the transmittance branch in the subsystem matrix. For example, suppose the transmittance from node i to node j is $P_{ij}(s)$, then FORM will read in $P_{ij}(s)$, compute the number series, and assign it to the i^{th} column and j^{th} row of the subsystem matrix T.

Subroutine READH is a card-reading subroutine allowing the data to be of mixed format with continuation cards.

Calling sequence is CALL FORM (P_{ij}, Δt, t_o).

A5.3 SUBROUTINE REDUCE

Subroutine REDUCE is designed to generate a number series representing the transmittance from a given input node to a given output node. REDUCE uses the following subroutines:

a. ADDNOS.
b. DIVNOS.
c. MULNOS.
d. SUBNOS.

The algorithm used to reduce an n-node subsystem matrix is given in Addendum A1.

The problem of multiple source nodes (indicated by $t_{ii} = 1$ for node i being a source node) is no problem as each source node is treated independently.

The method of accomplishing this is as follows. Suppose the original subsystem matrix is as appears below (this is the subsystem matrix used in the thermostat problem):

$$T = \begin{bmatrix} [t_{11}] & 0 & 0 & [0 & 0 & 0 & 0] \\ 0 & t_{22} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_{33} & 0 & 0 & 0 & 0 \\ t_{14} & t_{24} & t_{34} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t_{45} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_{56} & 0 & 0 \\ 0 & 0 & t_{37} & 0 & t_{57} & 0 & 0 \end{bmatrix} \qquad t_{56} = t_{11} = t_{22} = t_{33} = 1$$

Note that nodes 1, 2, and 3 are source nodes while nodes 6 and 7 are output nodes. Therefore the subsystem can be completely described by the following six equivalent paths.

$$P_{16},\ P_{17},\ P_{26},\ P_{27},\ P_{36},\ P_{37}\ .$$

To form the paths P_{16} and P_{17}, use the submatrix

$$G_1 = \begin{bmatrix} t_{11} & 0 & 0 & 0 & 0 \\ t_{14} & 0 & 0 & 0 & 0 \\ 0 & t_{45} & 0 & 0 & 0 \\ 0 & 0 & t_{56} & 0 & 0 \\ 0 & 0 & t_{57} & 0 & 0 \end{bmatrix} .$$

Then, upon reduction,

$$P_{16} = t_{14} \cdot t_{45} \cdot t_{56}$$

and

$$P_{17} = t_{14} \cdot t_{45} \cdot t_{57}\ .$$

Likewise for the source nodes 2 and 3,

$$G_2 = \begin{bmatrix} t_{22} & 0 & 0 & 0 & 0 \\ t_{24} & 0 & 0 & 0 & 0 \\ 0 & t_{45} & 0 & 0 & 0 \\ 0 & 0 & t_{56} & 0 & 0 \\ 0 & 0 & t_{57} & 0 & 0 \end{bmatrix} \quad G_3 = \begin{bmatrix} t_{33} & 0 & 0 & 0 & 0 \\ t_{34} & 0 & 0 & 0 & 0 \\ 0 & t_{45} & 0 & 0 & 0 \\ 0 & 0 & t_{56} & 0 & 0 \\ t_{37} & 0 & t_{57} & 0 & 0 \end{bmatrix}$$

In general if there are n nodes in a subsystem, i input nodes, and f output nodes, then:

a. It takes $i \cdot f$ equivalent paths to describe the subsystem.

b. The size of a G matrix is $K = n - i + 1$ by $K = n - i + 1$.

c. There will be exactly i G matrices; note that in each of the G matrices the second through the K^{th} row and the second through the K^{th} column are identical. Therefore this portion of the G matrix need only be formed once and only the first row and first column need be changed for each input node.

The calling sequence for REDUCE is CALL REDUCE (K, t_o, Δt).

A5.4 NUMBER SERIES ARITHMETIC PACKAGE

A5.4.1 Subroutine ADDNOS

Subroutine ADDNOS is used to add like terms of two number series. The algorithm used by ADDNOS is as follows. Given two number series,

$$X = \{X_i\}$$

$$Y = \{Y_i\} \qquad i = 1, 2, \ldots, N$$

where the braces indicate number series,

then

$$Z = X + Y = \{X_i + Y_i\} \qquad i = 1, 2, \ldots, N$$

Note that ADDNOS is merely vector addition in the sense that X being a vector may be thought of as a n-tuple in Euclidian n-space. Physically the summing algorithm is merely the superposition of two time responses.

Calling sequence is CALL ADDNOS (X, Y, N, Z).

A5.4.2 Subroutine DIVNOS

Subroutine DIVNOS is used to divide two number series. Using the notation above:

$$Z = \frac{X}{Y} = \frac{\{X_i\}}{\{Y_i\}} \ .$$

The algorithm used is merely one of synthetic division as is used in many root-finding programs, etc.

Calling sequence is CALL DIVNOS (Y, X, N, Z).

A5.4.3 Subroutine MULNOS

Subroutine MULNOS is used to multiply two number series. Physically, this amounts to a convolution of two time responses. Therefore using the notation as above:

$$Z = X \cdot Y = \{X_i\} * \{Y_j\} = \{Z_i\} \qquad i = 1, 2, 3, \ldots, N \ .$$

Since this multiplication amounts to a convolution, and convolution is integration, the numerical integration scheme employed is the trapezoidal rule. Fuller details on this scheme may be found in Addendum A2.

The calling sequence is CALL MULNOS (X, Y, N, Z, Δt, t_o).

A5.4.4 Subroutine SUBNOS

Subroutine SUBNOS functions exactly as ADDNOS except that the number series are subtracted. Using the above notation:

$$Z = X - Y = \{X_i - Y_i\} = \{Z_i\} \qquad i = 1, 2, 3, \ldots, N \ .$$

The calling sequence is CALL SUBNOS (X, Y, N, Z).

A5.4.5 Conclusion

The remaining work is to complete and debug the program, and use it on several numerical problems. In this way methods can be developed for verifying and controlling the accuracy of the solutions. There are several ways of verifying accuracy. One is to solve a problem by number series and by other means, and compare the results. Another is to predict the errors, as illustrated in Appendix E.

```
    ISN        SOURCE STATEMENT

      0 $IBFTC CLODS    FJLIST,REF
        C  MAIN USES THE SUBROUTINES READH,FORM, AND REDUCE
      1        INTEGER SWITCH
      2        INTEGER PISWT
      3        INTEGER PIWRD
      4        INTEGER ROW1,ROW2,ERASE1,AND
      5        REAL NODVAL
      6        DIMENSION PIWRD(7)
      7        DIMENSION TEMPG(7,35)
     10        DIMENSION STATE(7,2),START(7,2),TLEVEL(7),VALNOD(7,35),NODVAL(7)
     11        DIMENSION ONE(35),T(7,7,35),G(7,7,35),INLIST(7),LSTOUT(7),
             1           ROW1(7),ROW2(7),MASK(10), TEMP1(35)
     12        DATA TEMPG/245*0.0/
     13        DATA NODVAL/ 7*0.0/,TLEVEL/7*0.0/,START/14*1.0/,STATE/14*1.0/,
             1      VALNOD/245*1.0/
     14        DATA MASK/ 2,4,8,16,32,64,128,256,512,1024/,ONE/35*0.0/
     15        COMMON TCMIN,SYSTIM,NDTMAX
     16        COMMON NODES,NODEX,NODIN,NODOUT,ITROW,ITCOL,IFORM
     17        COMMON IST,JST,STAT1,STAT2
     20        COMMON/DUMMY/G
     21        CALL TRAPOK
     22        PISWT = 0
     23        WRITE (6,5068)
        C
        C         READ IN THE INITIAL CONDITIONS AND SWITCHING ALGORITHM TESTING
        C     VALUES.
        C
     24    104 CALL READH(IST)
     25        IF (STAT2 .EQ. 0.0) GO TO 105
     30        START(IST,JST) = STAT1
     31        STATE(IST,JST) = STAT2
     32        GO TO 104
     33    105 ITCNT = 0
     34        DO 3999 I = 1,2
     35        DO 3999 J = 1,7
     36   3999 WRITE (6,5066) J,I,STATE(J,I),J,I,START(J,I)
     41        DO 4000 I = 1,7
     42        NODVAL(I) = START(I,I)
     43   4000 ROW1(I) = 0
     45        CALL READH(TCMIN)
     46        T1PCNT = 2.0/TCMIN
     47        TNDT = NDTMAX
     50        DTIME = T1PCNT/TNDT
     51        TFINAL = T1PCNT + SYSTIM
     52        ONE(1) = 1./DTIME
     53        CALL READH(NODES)
     54        NODSUM = NODEX + NODIN
        C
        C         READ IN THE INPUT AND OUTPUT NUMBERS.
        C
     55        CALL READH(INLIST)
     56        CALL READH(LSTOUT)
     57    100 CALL READH(ITROW)
     60        IF (IFORM .EQ. 0) GO TO 99
     63        IF (IFORM .NE. 1) GO TO 98
```

```
CLODS                                        FORTRAN SOURCE LIST CLODS                    11/01/65      PAGE  2
          ISN       SOURCE STATEMENT

           66       DO 4001 K = 1,NDTMAX
           67  4001 T(ITROW,ITCOL,K) = ONE(K)
           71       T(ITROW,ITCOL,NDTMAX) = 1.0E20
           72       GO TO 101
              C  READ IN THE T MATRIX AND ASSIGN IT.
           73    98 CALL FORM (TEMP1,DTIME,TFINAL)
           74       DO 4002 K = 1,NDTMAX
           75  4002 T(ITROW,ITCOL,K) = TEMP1(K)
           77   101 ROW1(ITROW) = ROW1(ITROW) + MASK(ITCOL)
          100       ITCNT = ITCNT + 1
          101       GO TO 100
          102    99 WRITE (6,5050) ITCNT
              C
              C          FORM THE PERMANENT PART OF THE 'G' MATRIX.
              C
          103       ERASE1 = MASK(NODES+1) - 3
          104       IPLUS = NODSUM - 1
          105       NSIZE = NODES - IPLUS
          106       DO 4003 I = 2,NSIZE
          107       II = I + IPLUS
          110       DO 4003 J = 2,NSIZE
          111       JJ = J + IPLUS
          112       IF ( AND(ROW1(II),MASK(JJ)) .EQ. 0 ) GO TO 4003
          115       ROW2(I) = ROW2(I) + MASK(J)
          116       DO 4004 K =1,NDTMAX
          117  4004 G(I,J,K) = T(II,JJ,K)
          121  4003 CONTINUE
              C FORM THE VARIABLE PART OF THE A MATRIX AND BEGIN REDUCTION FOR EACH OF
              C THE INPUT NODES.
          124       DO 4005 I = 1,NODSUM
          125       IROW = INLIST(I)
          126       ROW2(1) = 2
          127       DO 4006 K = 1,NDTMAX
          130  4006 G(1,1,K) = T(IROW,IROW,K)
          132       DO 4007 JERAS = 2,NSIZE
          133  4007 ROW2(JERAS) = AND(ROW2(JERAS),ERASE1)
          135       DO 4008 J = 2,NSIZE
          136       JJ = J + IPLUS
          137       IF ( AND(ROW1(IROW),MASK(JJ)) .EQ. 0 ) GO TO 103
          142       ROW2(1) = ROW2(1) + MASK(J)
          143       DO 4009 K = 1, NDTMAX
          144  4009 G(1,J,K) = T(IROW,JJ,K)
          146   103 IF ( AND(ROW1(JJ),MASK(IROW)) .EQ. 0 ) GO TO 4008
          151       ROW2(J) = ROW2(J) + 2
          152       DO 4010 K = 1,NDTMAX
          153  4010 G(J,1,K) = T(JJ,IROW,K)
          155  4008 CONTINUE
              C
              C   REDUCTION SUBROUTINE NOW FOLLOWS.
              C
          157       CALL REDUCE  (NSIZE,IROW,LSTOUT,ROW2,NODOUT,SYSTIM,DTIME,
                   1             NDTMAX,TEMPG)
              C
              C          THE DO 4011 LOOP PUTS THE EQUIVALENT PATH NO. SERIES FOR NODE
              C   I TO NODE J INTO T(J,I).
```

```
     ISN        SOURCE STATEMENT

            C
     160          DO 4011 ICK = 1,NODOUT
     161          IPCOL = LSTOUT(ICK)
     162          PIWRD(IPCOL) = PIWRD(IPCOL) + MASK(I)
     163          DO 4011 KGO = 1,NDTMAX
     164          T(IPCOL,I,KGO) = TEMPG(IPCOL,KGO)
     165    4011  WRITE (6,5055) I,IPCOL,T(IPCOL,I,KGO)
     170    4005  CONTINUE
     172          DO 4012 I = 1,NODES
     173          TLEVEL(I) = 0.0
     174    4012  NODVAL(I) = START(I,1)
     176          SWICNT = 0.0
     177          SWITCH = 1
     200          NOSWT = 0
     201     200  TIMSYS = SYSTIM
     202          WRITE (6,5067) TIMSYS
     203          DO 4030 K = 1,NDTMAX
     204          KWRT = K
            C
            C        THE DO 4031 LOOP FORMS THE OUTPUT NODE VALUES FOR T=TO+K*DT.
            C
     205          DO 4031 IGET = 1,NODOUT
     206          I = LSTOUT(IGET)
            C  I IS THE OUTPUT NO. AND ROW NO. OF THE T MATRIX.
     207          SUM = 0.0
     210          DO 4032 JGET = 1,NODSUM
     211          J = INLIST(JGET)
            C  J IS THE INPUT NODE NO. AND THE COLUMN NO. OF THE T MATRIX.
     212    4032  SUM = SUM + T(I,J,K)*NODVAL(J)
     214    4031  VALNOD(I,K) = SUM + TLEVEL(I)
            C
            C        THE DO 4034 LOOP IS THE SWITCHING ALGORITHM.
            C
     216          DO 4034 IGET = 1,NODOUT
     217          I = LSTOUT(IGET)
     220          GO TO (201,202),SWITCH
     221     201  IF (VALNOD(I,K) .GE. STATE(I,1)) GO TO 203
     224          GO TO 4034
     225     202  IF (VALNOD(I,K) .LE. STATE(I,2)) GO TO 204
     230    4034  CONTINUE
     232    4030  CONTINUE
     234          KWRT = K-1
     235          KBAR = SWITCH
     236          GO TO 207
     237     203  SWITCH = 2
     240          GO TO 205
     241     204  SWITCH = 1
     242     205  SWICNT = SWICNT + 1.
     243     207  TK = K
     244          IF (KBAR .NE. 0) GO TO 208
            C
            C  UPDATE SYSTEM ABSOLUTE TIME.
            C
     247          WRITE (6,5056)
     250     209  SYSTIM = TIMSYS + TK*DTIME
```

A.96

```
     ISN        SOURCE STATEMENT

     251        WRITE (6,5057) TIMSYS,SYSTIM
     252        WRITE (6,5058)
     253        DO 4038 IGET = 1,NODOUT
     254        I = LSTOUT(IGET)
     255        WRITE (6,5059) I
     256        WRITE (6,5071) KWRT
     257        DO 4038 J = 1,KWRT
     260   4038 WRITE (6,5060) VALNOD(I,J)
          C
          C     UPDATE THE LEVEL OF THE NODES
          C
     263        DO 4035 IGET = 1,NODOUT
     264        I = LSTOUT(IGET)
     265   4035 TLEVEL(I) = VALNOD(I,KWRT)
          C
          C     UPDATE THE INTERNAL SOURCE NODES.
          C
     267        K1 = NODEX + 1
     270        DO 4036 IGET = K1,NODSUM
     271        I = INLIST(IGET)
     272        J = IGET - 2
     273        K2 = NODOUT - J
     274        K3 = LSTOUT(K2)
     275   4036 NODVAL(I) = TLEVEL(K3)
          C
          C     UPDATE THE EXTERNAL SOURCE NODES.
          C
     277        DO 4037 IGET = 1,NODEX
     300        I = INLIST(IGET)
     301        IF (SWITCH .EQ. 1) GO TO 206
     304        NODVAL(I) = START(I,2)
     305        GO TO 4037
     306    206 NODVAL(I) = START(I,1)
     307   4037 CONTINUE
     311        DO 4039 I = 1,NODES
     312   4039 WRITE (6,5064) I,TLEVEL(I),NODVAL(I)
     314        KBAR = 0
     315        GO TO 210
     316    208 WRITE(6,5061)
     317        NOSWT = NOSWT + 1
     320        IF (NOSWT .LT. 3) GO TO 209
     323        WRITE (6,5065)
     324        STOP
     325    210 CONTINUE
     326        IF (SWICNT .LE. 4.) GO TO 200
     331        WRITE (6,5062)
     332        WRITE (6,5063) SYSTIM
     333   5050 FORMAT (1H110X10HTHERE ARE ,I3,24H NON-ZERO ELEMENTS IN T.)
     334   5055 FORMAT (1H0,10X,7HEQPATH(,I2,1H,,I2,4H) = ,F18.8)
     335   5056 FORMAT (1H1,10X,35HA SWITCHING STATE HAS BEEN REACHED.)
     336   5057 FORMAT (1H0,10X,28HTHE LAST EVENT WAS FROM T = ,F11.8,8H TO T = ,F
              118.8,9H MINUTES.)
     337   5058 FORMAT (1H0,6X,26HOUTPUT NODE VALUES FOLLOW.)
     340   5059 FORMAT(1H1,10X,21HTIME VALUES FOR NODE ,I2,4H ARE)
     341   5060 FORMAT (1H0,16X,F18.8)
```

A-97

```
    ISN       SOURCE STATEMENT

    342  5061 FORMAT (1H0,20X,53HSYSTEM HAS RUN THRU 35 DATA POINTS WITH NO SWIT
             1CHING.)
    343  5062 FORMAT (1H0,5X,32HFOUR (4) SWITCHES HAVE OCCURRED.)
    344  5063 FORMAT (1H0,5X,24HSYSTEM ABSOLUTE TIME IS ,F18.8,9H MINUTES.)
    345  5064 FORMAT (1H0,20X,8HNODE IS ,I2,10H TLEVEL = ,F18.8,10H NODVAL = ,F1
             18.8)
    346  5065 FORMAT (1H0,10X,64HTHREE TIME SPANS HAVE ELAPSED WITH NO SWITCHING
             1. JOB TERMINATED.)
    347  5066 FORMAT (1H0,6HSTATE(,I2,1H,,I2,4H) = ,F10.4,5X,6HSTART(,I2,1H,,I2,
             14H) = ,F10.4)
    350  5067 FORMAT (1H0,5X,20HS.20C EXEC.,TIMSYS = ,F18.8)
    351  5068 FORMAT (1H1,35X,22HTHE THERMOSTAT PROBLEM)
    352  5071 FORMAT (1H0,10X,42HNO. OF POINTS TAKEN DURING LAST EVENT WAS ,I2)
    353       STOP
    354       END
```

ADDENDUM A1

TRANSITION MATRIX REDUCTION ALGORITHM

GENERAL

The transition matrix T_{ss} ($\equiv T_0$) exhibited by Equation A-2, is a matrix representation of a linear network, or a linear signal flow graph, References 2 and 3. The linear signal flow graph is a topographical representation of a set of simultaneous linear equations used to describe the dynamic behavior of the physical system. The equation variables may be designated as nodes and the variable coefficients, which are transfer functions, as edges.

Thus, to process a signal through a network in a manner which indicates the nodal behavior, it is necessary to reduce the over-all network transition matrix T_{ss} to the form exhibited by Equation A-4 i.e., from T_0 to T_{n-1}. This operation provides the mechanism to relate all nodal behavior to a single input node.

Consider a subsystem of n nodes. Presume that the connection from node i to node j is described by the transfer function t^o_{ij}. Then if node No. 1 is the source node and all other nodes dependent nodes, the general transfer relationship is

$$
\begin{aligned}
x_1 &= x_1 \\
x_2 &= t^o_{12} x_1 + t^o_{22} x_2 + t^o_{32} x_3 + \cdots + t^o_{n2} x_n \\
x_3 &= t^o_{13} x_1 + t^o_{23} x_2 + t^o_{33} x_3 + \cdots + t^o_{n3} x_n \\
&\vdots \\
x_n &= t^o_{1n} x_1 + t^o_{2n} x_2 + t^o_{3n} x_3 + \cdots + t^o_{nn} x_n .
\end{aligned}
$$

or

$$x_j = \sum_{i=1}^{n} t_{ij} x_j \qquad (j = 2, \ldots, n)$$

It is desired to reduce the latter to the form

$$x_1 = x_1$$

$$x_2 = t_{12}^{n-1} x_1$$

$$x_3 = t_{13}^{n-1} x_1$$

$$x_4 = t_{14}^{n-1} x_1$$

$$\vdots$$

$$x_{n1} = t_{1n}^{n-1}$$

or

$$x_j = t_{ij}^{n-1} x_1$$

That is, determine an equivalent transfer function from the source node to each dependent node of the subsystem. Equivalently, in terms of the transfer matrix reduce

$$T_o = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ t_{12}^{o} & t_{22}^{o} & \cdots & t_{n2}^{o} \\ t_{13}^{o} & t_{23}^{o} & \cdots & t_{n3}^{o} \\ \vdots & \vdots & & \vdots \\ t_{1n}^{o} & t_{2n}^{o} & \cdots & t_{nn}^{o} \end{bmatrix}$$

to the form

$$
T_{n-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ t_{12}^{n-1} & 0 & \cdots & 0 \\ t_{13}^{n-1} & 0 & \cdots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ t_{1n}^{n-1} & 0 & \cdots & 0 \end{bmatrix} .
$$

A reduction algorithm to compute the elements of T_{k+1} from the elements of T_{k1} may be described as follows:

$$
T_{k+1} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 \\ t_{12}^{k+1} & t_{22}^{k+1} & \cdots & t_{[n-(k+1)]2}^{k+1} & 0 & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ t_{1n}^{k+1} & t_{2n}^{k+1} & \cdots & t_{[n-(k+1)]n}^{k+1} & 0 & 0 \end{bmatrix}
$$

where for k = 0, 1, 2 . . . . n-2

$$
t_{i(n-k)}^{k+1} = \frac{t_{i(n-k)}^{k}}{1-t_{(n-k)(n-k)}^{k}} \text{ for } 1 \leq i < n-k \;\&\; t_{(n-k)(n-k)}^{k} \neq 1
$$

$$
t_{ij}^{k+1} = t_{ij}^{k} + t_{i(n-k)}^{k+1}\, t_{(n-k)j}^{k} \text{ for } 1 < j \leq n \quad j \neq n-k
$$

$$
= 0 \text{ for } n-k \leq i \leq n \;\&\; 1 \leq j \leq n
$$

$$
= 0 \quad j = 1 \;\&\; 1 < i \leq n
$$

$$
t_{11}^{k+1} = 1
$$

For the event $t^{k}_{(n-k)(n-k)} = 1$, a multiple source node is present and the above procedure must be modified to recognize this situation. It may be stated, however, that this situation usually does not arise in practice, or can easily be avoided.

MULTIPLE SOURCE NODES

The reduction algorithm from T_k to T_{k+1} is valid only for the case $t^{k}_{(n-k)(n-k)} \neq 1$. The event of $t^{k}_{(n-k)(n-k)} = 1$ complicates the situation and it is probably best to consider an example before writing a general algorithm. To this end, suppose the transfer matrix

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ t^1_{12} & t^1_{22} & t^1_{32} & t^1_{42} & 0 \\ t^1_{13} & t^1_{23} & t^1_{33} & t^1_{43} & 0 \\ t^1_{14} & t^1_{24} & t^1_{34} & 1 & 0 \\ t^1_{15} & t^1_{25} & t^1_{35} & t^1_{45} & 0 \end{bmatrix}$$

The corresponding transform equations are

$$\begin{aligned} x_1 &= x_1 \\ x_2 &= t^1_{12}x_1 + t^1_{22}x_2 + t^1_{32}x_3 + t^1_{42}x_4 \\ x_3 &= t^1_{13}x_1 + t^1_{23}x_2 + t^1_{33}x_3 + t^1_{43}x_4 \\ x_4 &= t^1_{14}x_1 + t^1_{24}x_2 + t^1_{34}x_3 + x_4 \\ x_5 &= t^1_{15}x_1 + t^1_{25}x_2 + t^1_{35}x_3 + t^1_{45}x_4 \;. \end{aligned}$$

By the fourth equation

$$x_3 = -\frac{t^1_{14}}{t^1_{34}}x_1 - \frac{t^1_{24}}{t^1_{34}}x_2 \;.$$

Thus

$$x_1 = x_1$$

$$x_2 = \left(t^1_{12} - \frac{t^1_{14}t^1_{32}}{t^1_{34}}\right)x_1 + \left(t^1_{22} - \frac{t^1_{24}t^1_{32}}{t^1_{34}}\right)x_2 + t^1_{42}x_4$$

$$x_3 = \left(t^1_{13} - \frac{t^1_{14}t^1_{33}}{t^1_{34}}\right)x_1 + \left(t^1_{23} - \frac{t^1_{24}t^1_{33}}{t^1_{34}}\right)x_2 + t^1_{43}x_4$$

$$x_4 = x_4$$

$$x_5 = \left(t^1_{15} - \frac{t^1_{14}t^1_{35}}{t^1_{34}}\right)x_1 + \left(t^1_{25} - \frac{t^1_{24}t^1_{35}}{t^1_{34}}\right)x_2 + t^1_{45}x_4$$

and the transfer matrix becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \left(t^1_{12} - \frac{t^1_{14}t^1_{32}}{t^1_{34}}\right) & \left(t^1_{22} - \frac{t^1_{24}t^1_{32}}{t^1_{34}}\right) & 0 & t^1_{42} & 0 \\ \left(t^1_{13} - \frac{t^1_{14}t^1_{33}}{t^1_{34}}\right) & \left(t^1_{23} - \frac{t^1_{24}t^1_{33}}{t^1_{34}}\right) & 0 & t^1_{43} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \left(t^1_{15} - \frac{t^1_{14}t^1_{35}}{t^1_{34}}\right) & \left(t^1_{25} - \frac{t^1_{24}t^1_{35}}{t^1_{34}}\right) & 0 & t^1_{45} & 0 \end{bmatrix}$$

Thus, node 4 is also a source node and independent of node 1. That is, a stimulus at node 1 results in zero response at node 4 and a stimulus at node 4 results in zero response at node 1. The response caused by stimuli at nodes 1 and 4 is the superposition of the response caused by the stimulus at node 1 alone and the response caused by the stimulus at node 4 alone. Hence consider two systems: (1) the equivalent system if $x_4 \equiv 0$ and (2) the equivalent system if $x_1 \equiv 0$.

$$\tilde{x}_1 = x_1$$

$$\tilde{x}_2 = \left(t^1_{14} - \frac{t^1_{14}\, t^1_{32}}{t^1_{34}}\right)\tilde{x}_1 + \left(t^1_{22} - \frac{t^1_{24}\, t^1_{32}}{t^1_{34}}\right)\tilde{x}_2$$

$$\tilde{x}_3 = \left(t^1_{13} - \frac{t^1_{14}\, t^1_{33}}{t^1_{34}}\right)\tilde{x}_1 + \left(t^1_{23} - \frac{t^1_{24}\, t^1_{33}}{t^1_{32}}\right)\tilde{x}_2$$

$$\tilde{x}_5 = \left(t^1_{15} - \frac{t^1_{14}\, t^1_{35}}{t^1_{34}}\right)\tilde{x}_1 + \left(t^1_{25} - \frac{t^1_{24}\, t^1_{35}}{t^1_{34}}\right)\tilde{x}_2$$

and

$$\hat{x}_4 = x_4$$

$$\hat{x}_2 = t^1_{42}\hat{x}_4 + \left(t^1_{44} - \frac{t^1_{24}\, t^1_{32}}{t^1_{34}}\right)\hat{x}_2$$

$$\hat{x}_3 = t^1_{43}\hat{x}_4 + \left(t^1_{23} - \frac{t^1_{24}\, t^1_{33}}{t^1_{34}}\right)\hat{x}_2$$

$$\hat{x}_5 = t^1_{45}\hat{x}_4 + \left(t^1_{25} - \frac{t^1_{24}\, t^1_{35}}{t^1_{34}}\right)\hat{x}_2 .$$

The respective transfer matrices are

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \left(t_{12}^{1} - \dfrac{t_{14}^{1} t_{32}^{1}}{t_{34}^{1}}\right) & \left(t_{22}^{1} - \dfrac{t_{24}^{1} t_{32}^{1}}{t_{34}^{1}}\right) & 0 & 0 \\ \left(t_{13}^{1} - \dfrac{t_{14}^{1} t_{33}^{1}}{t_{34}^{1}}\right) & \left(t_{23}^{1} - \dfrac{t_{24}^{1} t_{33}^{1}}{t_{34}^{1}}\right) & 0 & 0 \\ \left(t_{15}^{1} - \dfrac{t_{14} t_{35}}{t_{34}^{1}}\right) & \left(t_{25}^{1} - \dfrac{t_{24}^{1} t_{35}^{1}}{t_{34}^{1}}\right) & 0 & 0 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ t_{42}^{1} & \left(t_{22}^{1} - \dfrac{t_{24}^{1} t_{32}^{1}}{t_{34}^{1}}\right) & 0 & 0 \\ t_{43}^{1} & \left(t_{23}^{1} - \dfrac{t_{24}^{1} t_{33}^{1}}{t_{34}^{1}}\right) & 0 & 0 \\ t_{45}^{1} & \left(t_{25}^{1} - \dfrac{t_{24}^{1} t_{35}^{1}}{t_{34}^{1}}\right) & 0 & 0 \end{bmatrix}$$

Application of the reduction algorithm to each of these transfer matrices then yields the transfer matrices to the dependent nodes from the respective source nodes of the form:

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ t^2_{12} & 0 & 0 & 0 \\ t^2_{13} & 0 & 0 & 0 \\ t^2_{15} & 0 & 0 & 0 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ t^2_{42} & 0 & 0 & 0 \\ t^2_{43} & 0 & 0 & 0 \\ t^2_{45} & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{x}_1 = x_1 \qquad \hat{x}_4 = x_4$$

$$\tilde{x}_2 = t^2_{12}x_1 \qquad \hat{x}_2 = t^2_{42}x_4$$

$$\tilde{x}_3 = t^2_{13}x_1 \qquad \hat{x}_3 = t^2_{43}x_4$$

$$\tilde{x}_5 = t^2_{15}x_1 \qquad \hat{x}_5 = t^2_{45}x_4 \;.$$

Superposition leads to

$$x_1 = x_1$$

$$x_2 = t^2_{12}x_1 + t^2_{42}x_4$$

$$x_3 = t^2_{13}x_1 + t^2_{43}x_4$$

$$x_4 = x_4$$

$$x_5 = t^2_{15}x_1 + t^2_{45}x_4 \;.$$

For $t^k_{(n-k)(n-k)} = 1$, the general form of the algorithm to calculate the elements of the two transfer matrices say L_{k+1} and H_{k+1} from T_k is

$$\ell^{k+1}_{ij} = t^k_{ij} - \frac{t^k_{i(n-k)}\, t^k_{[n-(k+1)]j}}{t^k_{[n-(k+1)](n-k)}}, \qquad \begin{array}{l} 1 \leq i < [n-(k+1)] \\ 1 < j < n-k \end{array}$$

$$\ell^{k+1}_{ij} = t^k_{i(j+1)} - \frac{t^k_{i(n-k)}\, t^k_{[n-(k+1)](j+1)}}{t^k_{[n-(k+1)](n-k)}}, \qquad \begin{array}{l} 1 \leq i < [n-(k+1)] \\ n-k \leq j < n \end{array}$$

$$\ell^{k+1}_{11} = 1, \quad \ell^{k+1}_{i1} = 0, \quad 1 < i \leq n-1$$

$$\ell^{k+1}_{ij} = 0, \quad n-(k+1) \leq i \leq n-1, \quad 1 < j \leq n-1$$

$$h^{k+1}_{11} = 1$$

$$h^{k+1} = t^k_{(n-k)j}, \quad 1 < j < n-k$$

$$h^{k+1}_{1j} = t^k_{(n-k)(j+1)}, \quad n-k \leq j < n$$

$$h^{k+1}_{ij} = \ell^{k+1}_{ij}, \quad 1 < i \leq n-1, \quad 1 \leq j \leq n-1.$$

The equivalent systems represented by L_{k+1} and H_{k+1} must be handled separately by the reduction algorithm until the final response functions are determined. Superposition is then carried out in a manner like the example above.

ADDENDUM A2

SERIAL NUMBER

The replacement of the transfer functions, or edge factors t_{ij}'s, in the transition matrices, T_s and T_{ss}, by "equivalent" serial numbers requires some justification. The process is best explained by recapitulating some well-known linear system theory. Consider an arbitrary edge, $G_{ij}(s)$, connecting node x_i to node x_j in an oriented graph:

$$x_i \xrightarrow{\quad G_{ij}(s) \quad} x_j$$

The edge factor $G_{ij}(s)$ is also known as a weighting function, a system function, transfer function, etc.

If a unit impulse function, $\delta(t)$, is applied to $G_{ij}(s)$ the output response is designated as the impulse response, $h(t)$; i.e.:

$$L[h(t)] = G_{ij}(s)L[\delta(t)] \tag{A-76}$$

whence, since $L[\delta(t)] = 1$:

$$h(s) = G_{ij}(s) \tag{A-77}$$

Thus for any input x_i:

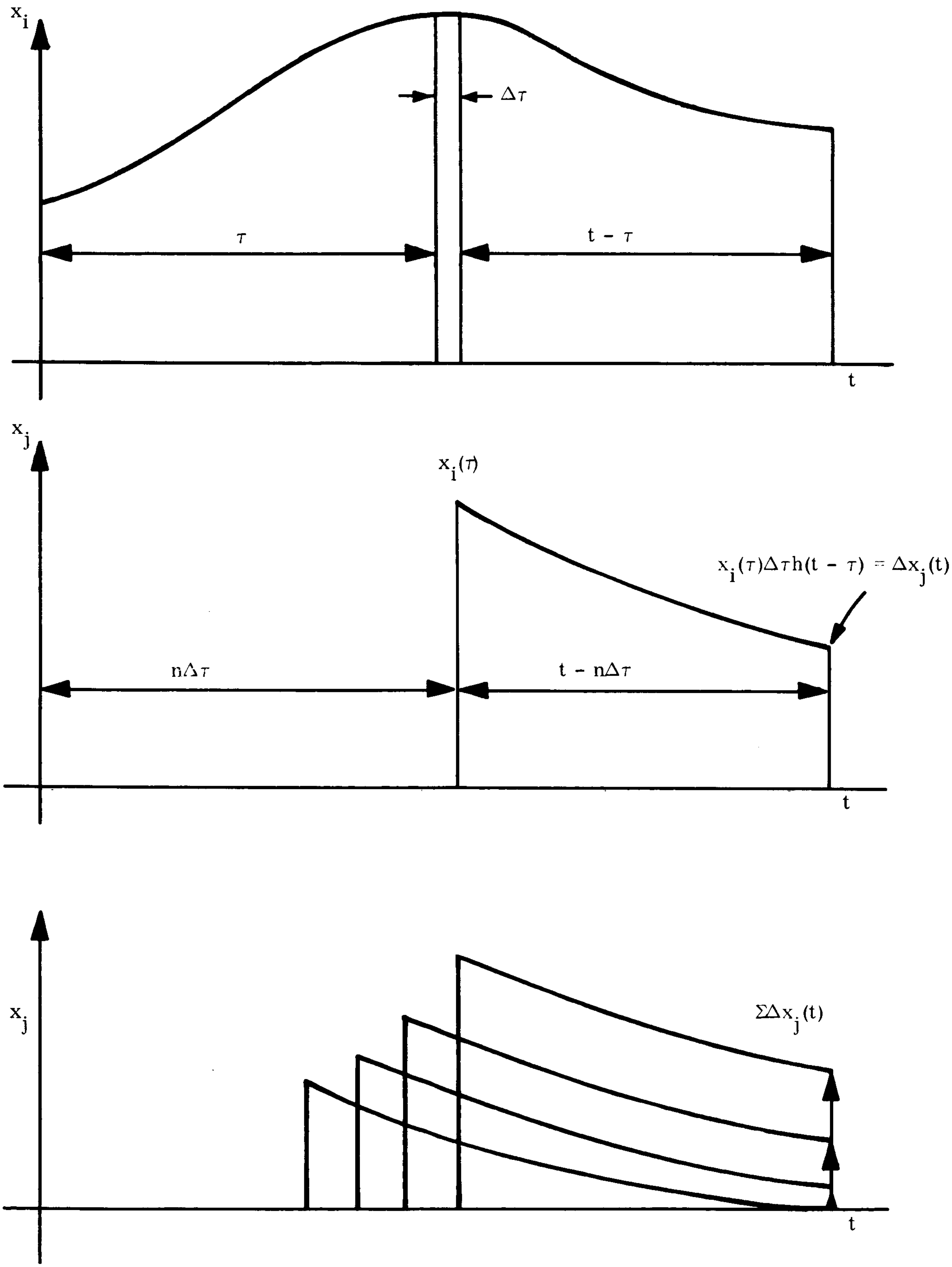
$$x_j(s) = G_{ij}(s)x_i(s) \tag{A-78}$$

$$x_j(t) = L^{-1}[G_{ij}(s)x_i(s)] \tag{A-79}$$

$$= \int_0^t h(\tau)x_i(t-\tau)\,d\tau \tag{A-80}$$

$$= \int_0^t h(t-\tau)x_i(\tau)\,d\tau \tag{A-81}$$

which is the standard convolution integral and has been derived for the case in which the input is represented as a series of pulses attenuated by the system response delay factors. This is illustrated in the following curves:



x_i
$\Delta\tau$
τ
$t - \tau$
t
x_j
$x_i(\tau)$
$x_i(\tau)\Delta\tau h(t - \tau) = \Delta x_j(t)$
$n\Delta\tau$
$t - n\Delta\tau$
t
x_j
$\Sigma\Delta x_j(t)$
t

That is, $x_j(t)$ represents the area under the preceding curves i.e., the sum of all the impulses from $t = 0$ to t, or from $\tau = 0$ to $n\Delta\tau$ if $\Delta\tau$ is the impulse width, and t is divided into n strips. In essence this constitutes numerical integration.

Let the input impulse strips, at $n\Delta\tau$, be represented by the series:

$$x_i(n\Delta\tau):\ x_o, x_1, x_2, x_3, \ldots, x_k, \ldots x_n \tag{A-82}$$

and the system impulsive response, or memory function, at $t - n\Delta\tau$, by

$$h(t = n\Delta\tau):\ g_n, g_{n-1}, g_{n-2} \ldots g_o \tag{A-83}$$

Whence Equation A-81 can be written

$$x_j(t) = \sum_{k=0}^{n} (x_k \Delta\tau g_{n-k})_i = \Delta\tau_i \sum_{k=0}^{n} (x_k g_{n-k})_i \tag{A-84}$$

where the subscript i is used to identify the input x_i and the edge function $G_i(s)$ operating on it.

Expanding Equation A-84:

$$x_j(t) = \Delta\tau_i(x_o g_n + x_1 g_{n-1} + \ldots + x_n g_o)_i. \tag{A-85}$$

However, the area under a curve can be obtained by other integration schemes, i.e., by Simpson's Rule, the Trapezoidal Rule, Weddle's Rule, etc. If the Trapezoidal Rule is used Equation A-85 would be modified only slightly:

$$x_j(t) = \Delta\tau_i\left(\frac{x_o g_n}{2} + x_1 g_{n-1} + \ldots + \frac{x_n g_o}{2}\right)_i \tag{A-86}$$

or

$$x_j(0) = 0$$

$$x_j(t) = \begin{cases} \Delta\tau_i\left(\dfrac{x_o g_1}{2} + \dfrac{x_1 g_o}{2}\right)_i \\ \Delta\tau_i\left[\dfrac{x_o g_n}{2} + \dfrac{x_n g_o}{2} + \displaystyle\sum_{k=1}^{n-1} x_k g_{n-k}\right]_i & 2 \le n \end{cases} \tag{A-87}$$

Equation A-87 can be evaluated by cross multiplication:

$$
\begin{array}{cccccc}
\frac{x_0}{2} & x_1 & x_2 & x_3 & x_4 & \cdots \\[2ex]
\frac{g_0}{2} & g_1 & g_2 & g_3 & g_4 & \cdots \\[1ex]
\hline
\frac{x_0 g_0}{4}^{*} & \frac{x_0 g_1}{2} & \frac{x_0 g_2}{2} & \frac{x_0 g_3}{2} & \frac{x_0 g_4}{2} & \cdots \\[2ex]
 & \frac{x_1 g_0}{2} & x_1 g_1 & x_1 g_2 & x_1 g_3 & \cdots \\[2ex]
 & & \frac{x_2 g_0}{2} & x_2 g_1 & x_2 g_2 & \cdots \\[2ex]
 & & & \frac{x_3 g_0}{2} & x_3 g_1 & \cdots \\[2ex]
 & & & & \frac{x_4 g_0}{2} & \cdots \\
\hline
\end{array}
$$

or:

$$x_j(0) = 0$$

$$x_j(1\Delta t) = \frac{x_0 g_1}{2} + \frac{x_1 g_0}{2}$$

$$x_j(2\Delta t) = \frac{x_0 g_2}{2} + x_1 g_1 + \frac{x_2 g_0}{2}$$

$$x_j(3\Delta t) = \frac{x_0 g_3}{2} + x_1 g_2 + x_2 g_1 + \frac{x_3 g_0}{2}$$

The problem can now be extended to two edges in tandem:

$$x_1 \xrightarrow{G_{12}(s)} x_2 \xrightarrow{G_{23}(s)} x_3$$

*By definition from Equation A-81 x_j (t=0) = 0, so this product is not used.

From Equation A-84

$$x_2(t) = \Delta\tau_1 \sum_{k=0}^{n} (x_k g_{n-k})_1 \qquad \text{(A-88)}$$

$$x_3(t) = \Delta\tau_2 \sum_{k=0}^{n} (x_k g_{n-k})_2 \, , \qquad \text{(A-89)}$$

where $\Delta\tau_2$ may be taken as identical to $\Delta\tau_1$.

If a unit impulse is admitted to x_1, Equation A-88 reduces to:

$$x_2(t) = \Delta\tau_1 \sum_{k=0}^{n} (g_k)_1 \qquad \text{(A-90)}$$

Substituting the summation in Equation A-90 into Equation A-89

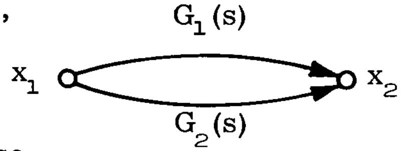
$$x_3(t) = \Delta\tau_1 \sum_{k=0}^{n} (g_k)_1 (g_{n-k})_2 \qquad \text{(A-91)}$$

The accuracy of Equation A-91 may be increased if the trapezoidal rule is used and it is expanded as in Equation A-87

$$x_3(t) = \begin{cases} \Delta\tau_1 \left(\dfrac{g_0^{(1)} g_1^{(2)}}{2} + \dfrac{g_1^{(1)} g_0^{(2)}}{2} \right) \\ \Delta\tau_1 \left(\dfrac{g_0^{(1)} g_n^{(2)}}{2} + \dfrac{g_n^{(1)} g_0^{(2)}}{2} + \displaystyle\sum_{k=1}^{n-1} g_k^{(1)} g_{n-k}^{(2)} \right) & n \geq 2 \end{cases} \qquad \text{(A-92)}$$

where the bracketed superscript is used to indicate which memory function is being considered.

Repetition of Equation A-92 carries us through a tandem system of any order. The case for two edges in parallel,



can be reduced to the first case.

$$x_2(s) = [G_1(s) + G_2(s)]\, x_1(s) \tag{A-93}$$

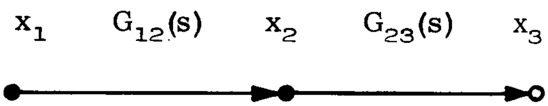
By applying the unit impulse function:

$$x_2(t) = \Delta\tau_1 \sum_{k=0}^{n} (g_k)_1 + \Delta\tau_2 \sum_{k=0}^{n} (g_k)_2 \tag{A-94}$$

if $\Delta\tau_2 = \Delta\tau_1$

$$x_2(t) = \Delta\tau_1 \sum_{k=0}^{n} [(g_k)_1 + (g_k)_2] \tag{A-95}$$

As an example, consider a tandem system, in the time domain:



Let $\Delta\tau = 0.2$ Sec.

$$G_{12}(s) = G_{23}(s) = \frac{1}{1 + s}$$

$$G_{12}(t) = G_{23}(t) = e^{-t}$$

$G_{12}(t) = e^{-t}$ is plotted in Figure A-38. Its cross product muptiplication with $G_{23}(t)(=e^{-t})$ is indicated in the attached table. The total response is indicated as the sum

$$k_j = .2\Sigma(g_k^{(1)} g_{n-k}^{(2)})$$

at the bottom of the columns.

If

$$G(s) = \frac{1}{1 + s} \times \frac{1}{1 + s} = \frac{1}{(1 + s)^2}$$

is the combined transfer function from x_1 to x_2, then

$$G(t) = te^{-t}$$

CROSS MULTIPLICATION: $g_k^{(1)} g_k^{(2)}$

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$g_k^{(1)}$	1/2	819	670	549	449	368	301	.247	202	.165	135	111	.091	.074	.061	050	.041	.033	.027	.022	.018
$g_k^{(2)}$	1/2	819	670	549	449	.368	301	247	202	165	135	111	091	.074	061	.050	.041	.033	.027	.022	.018
	250	410	.335	275	225	134	.151	.124	101	.083	068	056	046	.037	.031	.025	.021	.017	.014	.011	.009
		409	671	549	450	368	301	247	202	.165	135	111	091	075	061	.050	.041	.034	.027	.022	.018
			.335	"	449	"	"	"	"	"	"	"	.090	.074	"	"	"	"	"	"	"
				274	450	"	"	"	"	"	.136	"	091	"	"	"	"	.033	"	.023	"
					224	"	"	"	"	"	.135	"	"	"	"	"	"	"	"	.022	"
						.134	"	"	"	"	"	"	"	"	"	"	"	"	"	"	"
							150	"	"	"	"	"	"	"	"	"	"	"	"	"	"
								.123	"	"	.136	"	"	"	"	"	"	"	"	"	"
									.101	"	135	"	"	"	"	"	"	"	"	"	"
										.082	"	"	"	"	"	"	"	"	"	"	"
											067	"	.090	"	"	"	"	"	"	"	"
												.055	.091	"	"	"	"	"	"	"	"
													045	.075	"	"	"	"	"	"	"
														.037	"	"	"	"	"	"	"
															030	"	"	"	"	"	"
																.025	"	.034	"	"	"
																	.020	"	"	.023	"
																		.016	"	.022	"
																			.013	"	"
																				.011	"
																					.009
	0	.819	1 341	1 647	1 798	1 840	1 806	1 729	1 616	1 485	1 352	1 221	1 090	964	.854	.750	.656	.565	.486	.420	.360
$x_j = .2 \sum =$		.164	268	329	360	368	361	346	323	.297	270	244	213	193	171	.150	.131	.113	.097	.084	.072
e^{-t}	1	819	670	549	.449	368	.301	247	202	.165	135	111	091	.074	.061	.050	.041	.033	.027	.022	.018
t	0	.2	.4	.6	.8	1.0	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3.0	3.2	3.4	3.6	3.8	4.0
te^{-t}	0	.164	.268	329	.359	368	.361	.346	.323	.297	.270	.244	.218	.192	.171	.150	.131	.112	.097	.084	.072

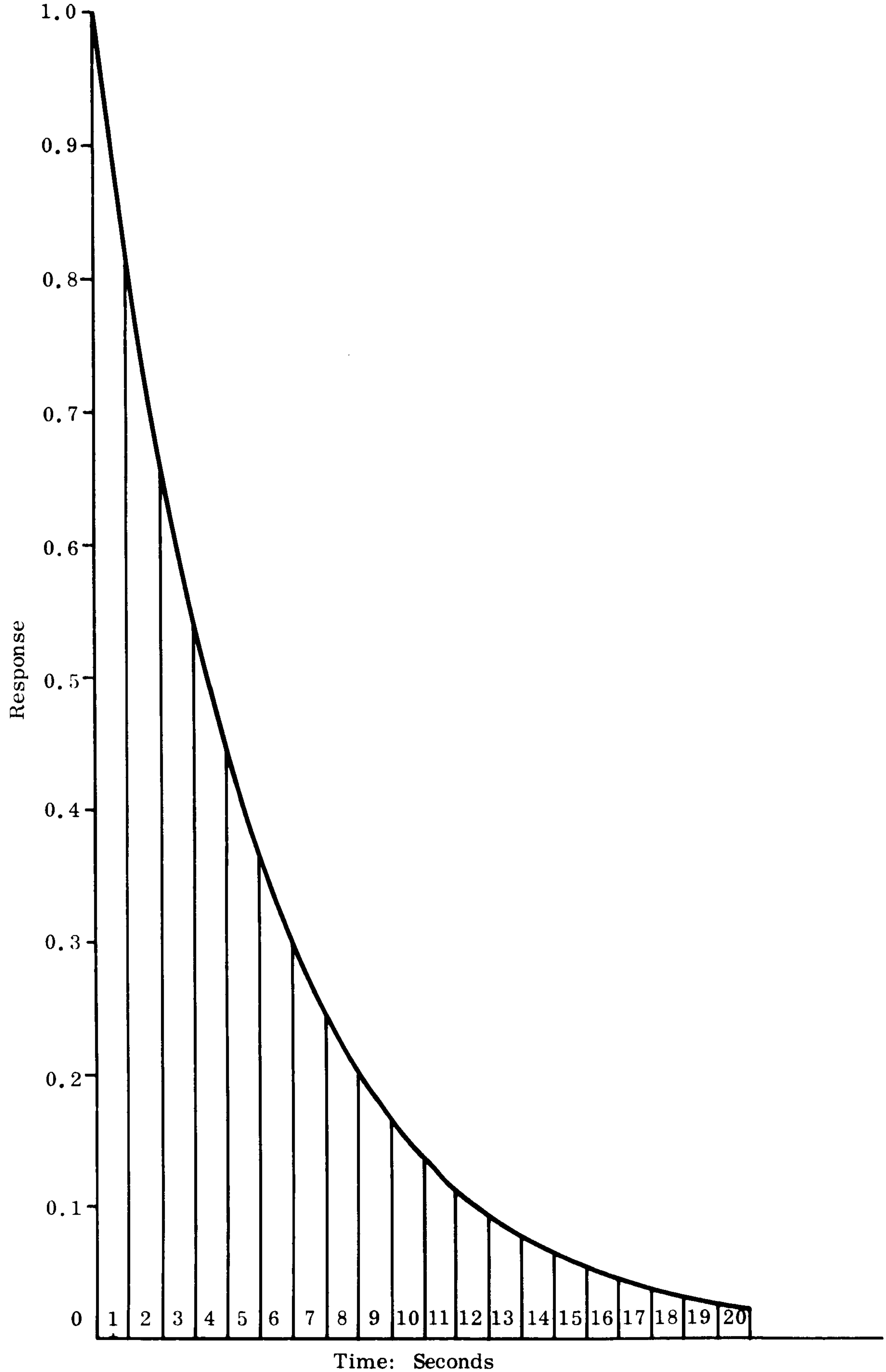


Figure A-38. Impulse Response: e^{-t}

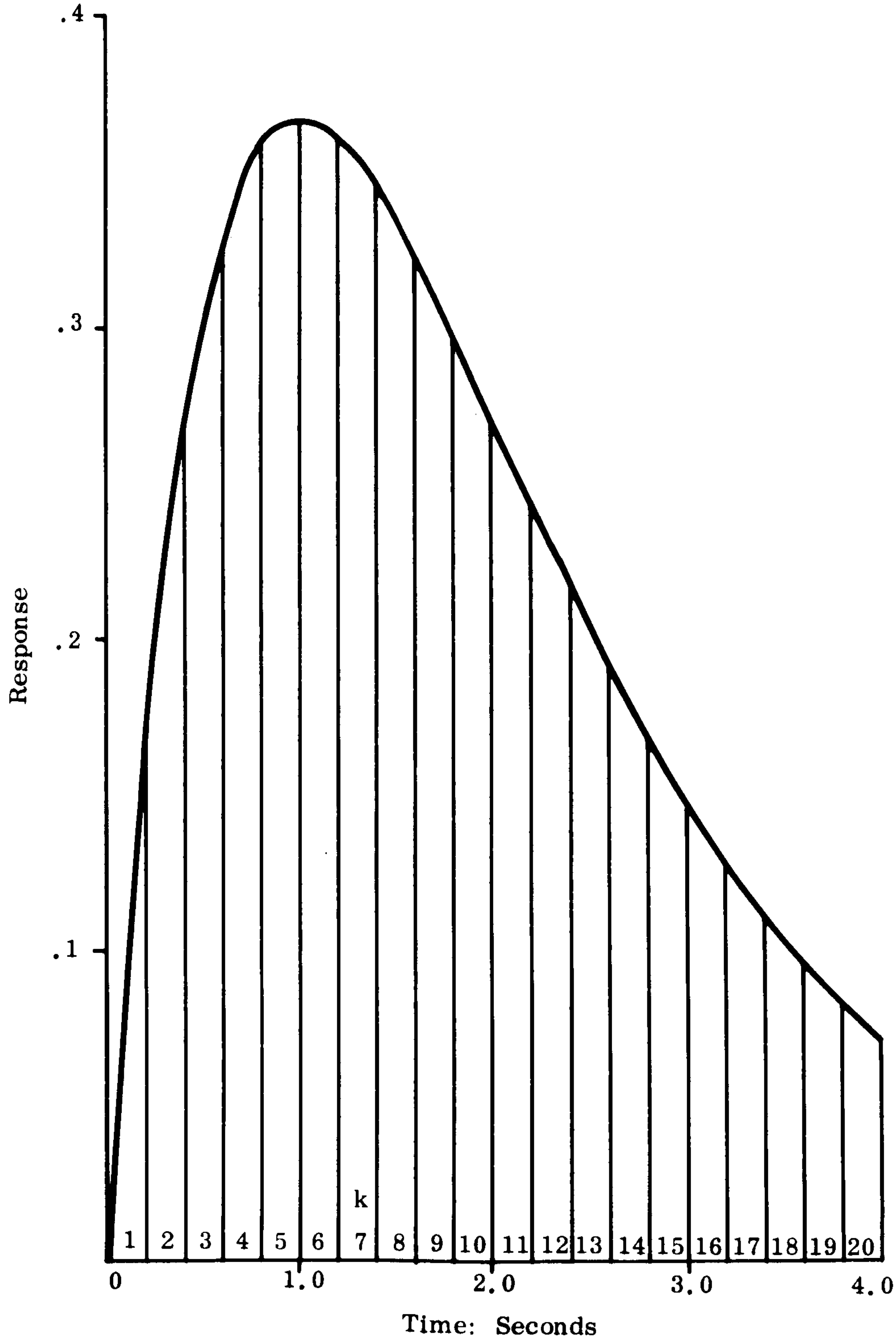


Figure A-39. Impulse Response: te^{-t}

This is calculated at the bottom of the attached table. It agrees almost identically with the above procedure.

The response function is plotted in Figure A-39.

The above derivation along with the illustrative example serves to demonstrate the fundamental nature of number series and the manner in which they combine. The series Equation A-83

$$g_0, g_1, g_2 \ . \ . \ . \ g_n$$

constitutes the serial number for the transfer function G(s). Equation A-90 illustrates the serial number response of G(s) to a unit impluse input. Equations A-91 and A-92 indicate that serial numbers are cross multiplied for cascaded edges and Equation A-95 indicates that serial numbers are added for tandem edges. These two properties demonstrate that serial numbers can be used to replace transfer functions in the transition matrix and indicate the rules which must be used in reducing the matrix.

APPENDIX B

DISCRETE SIMULATION

B1 GENERAL

A number of applications of the Saturn V/ESE discrete simulation are being developed or have been developed and successfully utilized. Further, all the applications that have been developed were successfully utilized on the S-IB simulation effort, and based on this experience, the following paragraphs describe the discrete simulation applications as employed on the LVCLS effort. The available methods of discrete simulation plus a description of the different computer output programs are presented in subsequent sections of this appendix and Appendix D.

a. Failure effects - various components or combinations of components are failed and the resulting sequence of events is compared (by the computer) to the normal sequence of events to identify the effect of the failure. This permits the user to evaluate the over-all system impact of failures and to determine if monitoring devices detected the failure and if so, how long between time of failure and malfunction detection.

b. Test data analysis - discrete event recordings (i.e., prelaunch simulated flight data; post-launch data; etc.,) are compared to the computed events in the simulation and all marginal or abnormal sequence events are identified. The user then may analyze to determine if component tolerances should be altered, marginal components replaced, switching sequences changed, etc.

c. Power profiles - the voltage bus current required as a function of launch countdown time (or checkout test steps) is computed and presented in a manner that permits the user to determine at what time peak loading occurs; total current required for each voltage bus, voltage loads created by failures or abnormal conditions, etc.

d. RFI - Data are compiled (similar to power profiles) and graphically (or numerically) displayed to show total switching versus time, thereby, identifing for the user the critical periods of high noise levels. The simulation data is such that the source of the noise can be identified, sequence of events altered, and simulation iterations performed until the analysis has optimized the noise level.

e. Design evaluation - the nature of the simulation is such that it proves the operability of the equipment design both on a unit level as well as an integrated system level. This eliminates last minute incompatibilities at equipment interfaces. Different configurations (i.e., 501, 502, etc.) as well as combinations of configurations can easily be accommodated.

f. Design change evaluation - proposed design changes may be incorporated into the simulation by simply changing the equation for the components affected. Then the design change simulation can be performed, compared with the prechange operation and all differences identified. The user may then determine if the differences comply with the intent of the design change. This method thoroughly assesses the over-all systems impact of a proposed design change.

g. Test procedure verification - with a simulation model of the test configuration, the operational steps of a test may be used as input to the computer and the output will be in the form of a test procedure that indicates which monitor points (lamps, etc.) are activated (or de-activated) as a result of each operation. This produces a highly accurate test procedure as well as permitting a user to view test results (simulated) before using the procedure on hardware. Further, verification is made, by the computer, that each component has been tested plus the simulation provides a convenient media for changing or up-dating test procedures.

h. Reliability assessment - the simulation program computes and prints out the number of times each component is activated during testing or launch count-down. This data may be obtained on a per test or time period basis and should be valuable in an operation and maintenance program. Also, it is possible to compute reliability numbers for each component, equipment rack/panel or entire system.

i. Computer software verification - the launch complex computer can be connected to the simulation computer via special input/output devices. This would permit preliminary debugging of the launch complex computer test programs without involving additional vehicle/ESE hardware.

j. Safing circuit verification - the operability of safing circuits, such as cutoff, can be verified by simulating malfunctions the circuits are designed to detect. Actually, simulation is the only way some safing circuits can be verified on a practical basis (i.e., engine fire, rough combustion, etc.). One simulation analysis of the cutoff circuits discovered circuits that did not operate as desired - cutoff should have, but didn't occur - plus a condition whereby cutoff can occur after commit. It is difficult to assign a dollar value to this type of analysis.

k. Special purpose - data listings are provided that present only data of special interest (i.e., DEE channel status, EA pens, etc.). This can be utilized as status tables in checkout or serve as back-up information during launch countdown (i.e., verify EDS test matrices).

l. Redundant components - the stimulations' logic equations (which actually represent the circuits) are investigated and all redundant component utilization is identified. The user may then determine if the redundancy was intentional. This is a tremendous aid in circuit design optimization. The investigation also identifies logical inconsistencies (i.e., a relay is required to be on and off simultaneously) for the user to evaluate and correct as necessary.

B2 BOOLEAN EQUATION METHOD OF DISCRETE SIMULATION AT THE COMPONENT LEVEL

B2.1 GENERAL

This digital computer simulation scheme for the Saturn V is simple and straightforward. As has been demonstrated on the ESE simulation, logic equations are developed from Saturn V vehicle and GSE schematics and punched on EDP cards for input to the computer. These equations describe, on a component level, all the necessary operational requirements of each component in the system.

The initial goals of the ESE simulation were, for each component, through the use of the logic equations, to:

a. Identify the type (relay, lamp, switch, etc.).
b. Identify the physical location (rack, panel, etc.).
c. Provide a functional description (i.e., ignition command).
d. Define its operational requirements (what it takes to make it work).
e. Identify its system utilization (once it operates, where it is used).
f. Identify the operational sequence (when in countdown it operates).

The present simulation accomplishes this and, through various output programs, provides other very useful information as described in subsequent sections of Appendix D. To aid in understanding how the logic equations are written and how the computer processes them, the following example is developed.

Figure B-1 is a schematic diagram that shows the circuits of relays K382 and K401. These circuits involve two different pieces of hardware, namely a networks panel and a relay rack. The relay K382 is energized when the voltage +A1D111 is on and the switch S2 is on (closed). This can be expressed in a logic equation as follows:

$$RR382 = A1D111 * NPS02 \qquad \text{(B-1)}$$

where,

A1D111 represents the voltage +A1D111
NP designates networks panel location
RR designates relay rack location
NPS02 represents the switch, S2
RR382 represents the relay K382

---
*Means "and" (corresponds to series circuit)

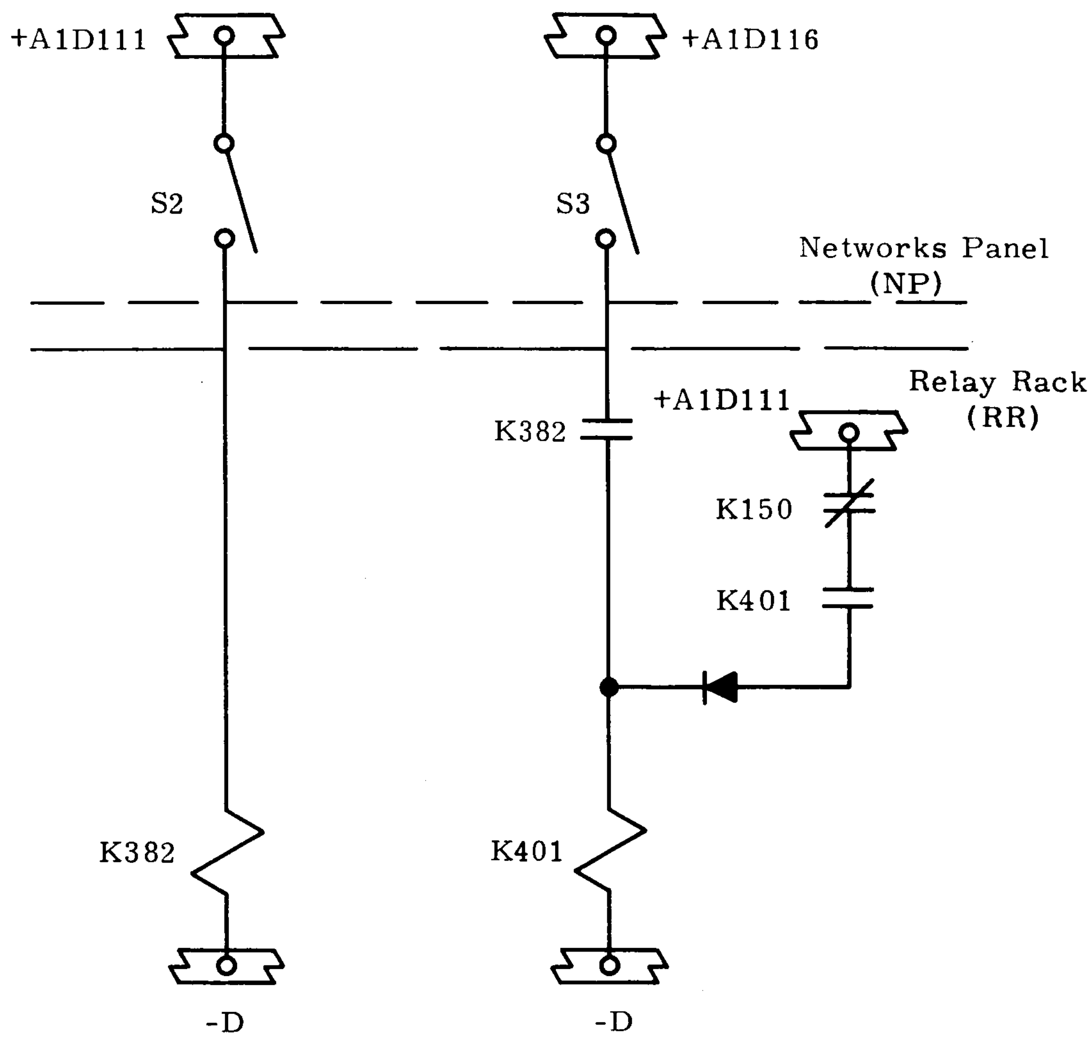


Figure B-1. Relays K382 and K401, Schematic Diagrams

The circuit for relay K401 is a little more complicated than K382; however, using the same approach and definitions as used in developing Equation B-1 an equation can be developed for K401 as follows:

RR401 = (A1D116*NPS03*RR382) + (A1D111* - RR150*RR401) (B-2)

where,

"+" means "or" (corresponds to parallel circuit)

- means "not" (corresponds to de-energized state)

Equation B-2 may be read as RR401 is energized when either the voltage A1D116 is on <u>and</u> the switch NPS03 is closed (on) <u>and</u> relay RR382 is energized (on) <u>or</u> the voltage A1D111 is on <u>and</u> relay RR150 is de-energized (off) <u>and</u> relay RR401 is energized (on). Naturally, RR401 will also be energized (on) if all these conditions are met; that is, both parallel paths are active.

Equations such as B-1 and B-2 are punched on EDP cards, input to the computer, compiled and stored in memory locations. Then, if at a particular time period of interest, the input stimuli are such that A1D111 and NPS02 are on and RR382 is not failed off, the computer program will recognize that the requirements for energizing RR382 have been met (refer to Equation B-1) and the output will state "RR382 ON," assuming that RR382 was previously off.

The computer also notes (reference Equation B-2) that one of the requirements for RR401 has been met, namely RR382. Assume that the voltage A1D116 is on. Then, any time an input statement turns NPS03 on (RR382 is still on), the computer program will recognize that all the requirements are present for RR401 to be energized and the output program will list "RR401 ON."

This method of simulation permits the user to manipulate the input stimuli in any desired arrangement and the computer will determine the status of the associated components as a result of the input stimuli. It should be noted that the scheme described here is used throughout the simulation; therefore the terms in each equation identify particular types of components and their locations. The ground rules for accomplishing this are simple. No more than six characters are used to identify a component. The first two (and sometimes three) characters identify a unique location. The type of component was identified in the ESE Simulation by the third or fourth character as follows: all components are relays unless the third or fourth character is one of the following letters which designates:

S = switch
L = lamp
D = voltage
$ = squib

The remaining characters in the term represent the components number as assigned on the schematic. Referring to Equation B-1 the term NPS02 represents a component on the networks panel, NP, that is switch S number two, 02, on the schematic Figure B-1.

B2.2 SIMULATION DATA BASE

The logic equation developed for each component includes all paths, and the components that comprise these paths, which are capable of activating that component. Accompanying each logic equation is a brief functional description of the component. This information is placed on EDP cards and processed into the computer for error checking and compiling, after which it becomes the data base for the simulation. The fact that this data base consists of a logic equation representing or containing each active component in the hardware guarantees the adequacy of the simulation on a one-to-one basis. A listing of the logic equations and functional descriptions is often a welcome replacement for schematics.

To add realism and depth of the simulation, certain additional data are included to describe each component's operational characteristics. This data consists of time, ranging from minutes required for preoperational warm up to milliseconds associated with a relay's pickup or dropout. The inclusion of this additional information makes a meaningful simulation of sequential operations possible.

Basically, the simulation consists of identifying a set of stimuli (i.e., switch closures, voltages on or off), coding and arranging them in a desired order (preferably with time-of-occurrence signatures), and inputting these to the computer. The computer simulation program then applies the stimuli to the equations and computes which components change their binary state. Also, the computed changes in state of these components are applied to the equations and subsequent changes in state noted in the output, along with the time (countdown time) associated with the change.

Given the data base of an operable design, the simulation program will produce a sequence-of-events output for some given set of inputs, which represents the normal sequence of events. This normal output consists of the total changes in status of all active components during the period of interest and is referred to as the normal discrete history output program. Additional output programs (described in detail in paragraph B2.3) have been developed and are operating. These extract from the history only those items of interest to a particular user. For example, if a test procedure were to be verified, the switch closures in the procedure would serve as inputs to the computer and an output program would identify only those lamps that change state as a result of the input.

B2.3 COMPUTER PROGRAMS

There have been two versions (I and II) of computer programs developed for the ESE simulation that accept the logic equations and perform the simulation described in paragraph B2.1.

Basically the Version II program accomplishes the same functions as the Version I, with certain improvements. The most significant difference is the increase in the number of components that can be simulated. While the Version I is limited to 3500 components, the Version II can accommodate more than 20,000 components on the IBM-7044 digital computer. Other notable improvements include a preprocessing editor that checks for errors at the earliest possible point to reduce computer time in case of error; a file maintenance program that permits changes to be conveniently made to the data base with minimum computer time required to up-date; several tab reports and various arrangements of the data can be made by the computer directly from the data base thus eliminating the need to sort data cards.

The Version I programs have been operating for the past several months and Version II programs became available at the end of October 1965. Using either version there are several listings that may be obtained from the data base itself, and several types of output listings from the simulation runs. A brief description of these listings, by source, is given in the following:

a. Outputs Generated from Data Base (no simulation required)
   (1) A listing of all the different equipment rack and panels involved in a particular operation.
   (2) A functional description of all components in a rack, panel, or entire system that also includes a reference to the schematic number and page number where the component and its circuit is located.
   (3) A listing of logic equations, by rack, panel or system, that defines the operational requirements for each active component involved. The listing also provides, on a per component basis, and indication of how many (if any) engineering change orders (EO's) have affected that component.
   (4) A components' system utilization cross-reference listing which lists all active components and identifies which circuit (if any) each component is used in and indicates whether the component is used in the normally open (on) or normally closed (off) state. Components such as lamps are excluded from this listing.
   (5) A listing of all components which have no functional requirements. Presumably this list would consist of switches and batteries; however, if a relay appeared in this list it would mean that contacts

of the relay appeared in a circuit but the coil of the relay had not been shown on the schematics.

(6) A combined (merged) listing of any or all of the preceding listings.

b. Outputs Generated from Simulation Data

(1) Discrete History (normal) - lists each component change in status (discrete event) which may result from some input stimuli (switch ON/OFF), initial condition and operating characteristics, and change in status of some other component. These discrete events are grouped by time of occurrence intervals and this time may represent actual countdown time or correspond to a step in a test procedure. Because it represents changes in status for all components this output is often referred to as the simulation run.

(2) Discrete History (EA Pen) - a listing consisting of only those status changes that involve Easterline Angus (EA) event recorder functions (pen signals). Since this data is extracted from the normal discrete history output, events are grouped by time or test procedure step.

(3) Discrete History (DEE) - a listing consisting of only Discrete Event Evaluator (DEE) status changes, grouped by time of occurrence (or test procedure step).

(4) Status Changes by Component Location - a listing of all components grouped by location and component schematic reference number with an indication for each component as to how many times it changed state and when (time or test step number). This is often used to verify that a component was tested (activated) during a particular test.

(5) Simulation Test versus Normal Comparison - an output listing, similar to a normal discrete history, except this listing presents status changes with a failed component condition or trial circuit change inserted in the data base. Included with the status changes is an indication whenever the test run differs from a normal run.

(6) Time Line Diagrams - this output is a graphic representation of the status changes of selected functions as computed in a simulation run. These status changes are presented in a format akin to the strip chart recorders.

(7) Test Procedures Report - developed specifically to support the ESE test unit in Huntsville and is identical in format to their test procedure. Briefly, there is an operations column, switch

closures, etc., are listed and under an adjacent Indications column, the corresponding lamp, monitor points, etc., status changes are listed.

B3 MATRIX METHOD OF DISCRETE SIMULATION AT THE COMPONENT LEVEL

B3.1 GENERAL

The simulation of a network interlaced with relays can be treated in a different manner from that indicated in the section on Dynamic Sequencing. That is to say, the network can be represented on a discrete basis to yield a set of Boolean logic equations defining all possible system connections to satisfy some arbitrary system condition.

Consider some arbitrary collection of connected relays, α_{ij}'s, or of an equivalent Boolean variable (see Figure B-2):

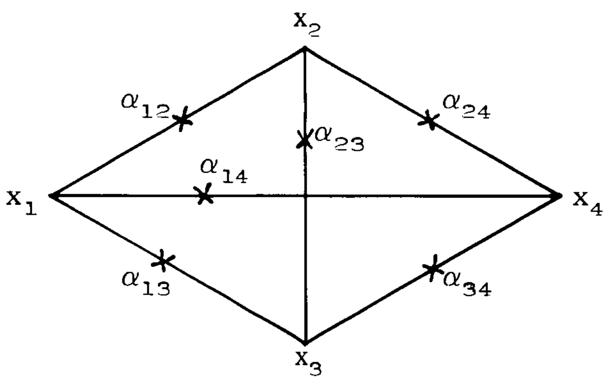


Figure B-2. Generic Boolean Network

α_{ij} represents the relay, or Boolean variable, and has the value 1 for a closed contact and zero for an open contact or the equivalent Boolean function. The nodes x_1, $x_2, \ldots, x_4$, of the nonoriented graph, are arbitrary points in the network at which the network components are joined or at which logic information is desired.

Figure B-2 yields a Boolean connection matrix of the network:

$$M = \begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 1 & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ x_2 & \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} \\ x_3 & \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} \\ x_4 & \alpha_{14} & \alpha_{24} & \alpha_{34} & 1 \end{array} \qquad \text{(B-3)}$$

The Boolean path product, $\pi(x_i, x_j)$, or the Boolean functional statement, for the network requirement between any pair of nodes x_i and x_j may now be calculated from M. For example, consider the condition between x_1 and x_4. M is rearranged to exhibit x_1 and x_4 in the upper lefthand corner of the matrix:

$$\pi(x_1, x_4) = \begin{array}{c|cccc} & x_1 & x_4 & x_2 & x_3 \\ \hline x_1 & 1 & \alpha_{14} & \alpha_{12} & \alpha_{13} \\ x_4 & \alpha_{14} & 1 & \alpha_{24} & \alpha_{34} \\ x_2 & \alpha_{12} & \alpha_{24} & 1 & \alpha_{23} \\ x_3 & \alpha_{13} & \alpha_{34} & \alpha_{23} & 1 \end{array} \tag{B-4}$$

and reduced to a 2x2 matrix by eliminating the internal nodes x_2 and x_3:

$$\pi(x_1, x_4) = \begin{array}{c|ccc} & x_1 & x_4 & x_2 \\ \hline x_1 & 1 & \alpha_{14} + \alpha_{13}\alpha_{34} & \alpha_{12} + \alpha_{13}\alpha_{23} \\ x_4 & \alpha_{14} + \alpha_{13}\alpha_{34} & 1 & \alpha_{24} + \alpha_{23}\alpha_{34} \\ x_2 & \alpha_{12} + \alpha_{13}\alpha_{23} & \alpha_{24} + \alpha_{23}\alpha_{34} & 1 \end{array} \tag{B-5}$$

$$= \begin{array}{c|c} & x_1 \\ \hline x_1 & 1 \\ x_4 & \alpha_{14} + \alpha_{12}(\alpha_{24} + \alpha_{23}\alpha_{34}) + \alpha_{13}(\alpha_{34} + \alpha_{23}\alpha_{24}) \end{array} \tag{B-6}$$

where the x_4 column is dropped in the last reduction because the x_1x_4 intersection is identical to the x_4x_1 intersection, or to the path product $\pi(x_1, x_4)$.

$$\pi(x_1, x_4) = \alpha_{14} + \alpha_{12}(\alpha_{24} + \alpha_{23}\alpha_{34}) + \alpha_{13}(\alpha_{34} + \alpha_{23}\alpha_{24}) \tag{B-7}$$

The Boolean function for any other combination of nodes can be obtained in the same manner, i.e., the network path products are obtained by iterating node combinations.

In the illustrative reduction from Equation B-6 to Equation B-7 it is observed that the final term (x_1, x_4), represents the simplified equivalent Boolean function. This simplification is carried out at each stage of the reduction. The expression of Table B-1, Reference 8, may be used.

Table B-1

Boolean Algebra

1.	$0 \cdot X = 0$	$1 + X = 1$
2.	$1 \cdot X = X$	$0 + X = X$
3.	$XX = X$	$X + X = X$
4.	$X\overline{X} = 0$	$X + \overline{X} = 1$
5.	$XY = YX$	$X + Y = Y + X$
6.	$XYZ = (XY)Z = X(YZ)$	$X + Y + Z = (X + Y) + Z = X + (Y + Z)$
7.	$\overline{XY \ldots Z} = \overline{X} + \overline{Y} + \ldots + \overline{Z}$	$\overline{X + Y + \ldots + Z} = \overline{X}\overline{Y} \ldots \overline{Z}$
8.	$\overline{f}(X, Y, \ldots, Z, \cdot, +) = f(\overline{X}, \overline{Y}, \ldots, \overline{Z}, +, \cdot)$	
9.	$XY + XZ = X(Y + Z)$	$(X + Y)(X + Z) = X + YZ$
10.	$XY + X\overline{Y} = X$	$(X + Y)(X + \overline{Y}) = X$
11.	$X + XY = X$	$X(X + Y) = X$
12.	$X + \overline{X}Y = X + Y$	$X(\overline{X} + Y) = XY$
	$ZX + Z\overline{X}Y = ZX + ZY$	
	$(Z + X)(Z + \overline{X} + Y) = (Z + X)(Z + Y)$	
13.	$XY + \overline{X}Z + YZ = XY + \overline{X}Z$	
	$(X + Y)(\overline{X} + Z)(Y + Z) = (X + Y)(\overline{X} + Z)$	
14.	$XY + \overline{X}Z = (X + Z)(\overline{X} + Y)$	
	$(X + Y)(\overline{X} + Z) = XZ + \overline{X}Y$	
15.	$X \cdot f(X, \overline{X}, Y, \ldots, Z) = X \cdot f(1, 0, Y, \ldots, Z)$	
	$X + f(X, \overline{X}, Y, \ldots, Z) = X \cdot f(0, 1, Y, \ldots, Z)$	
16.	$f(X, \overline{X}, Y, \ldots, Z) = X \cdot f(1, 0, Y, \ldots, Z) + \overline{X} \cdot f(0, 1, Y, \ldots, Z)$	
	$f(X, \overline{X}, Y, \ldots, Z) = [X + f(0, 1, Y, \ldots, Z)]\,[\overline{X} + f(1, 0, Y, \ldots, Z)]$	

B3.2 DISCRETE SIMULATION WITH TIME DELAY

The procedure outlined above is also applicable to non-ideal relay circuits. That is, to relays which require a finite time to make and break contacts. The application to a typical relay circuit might be considered. For example, consider the following relay circuit (see Figure B-3):

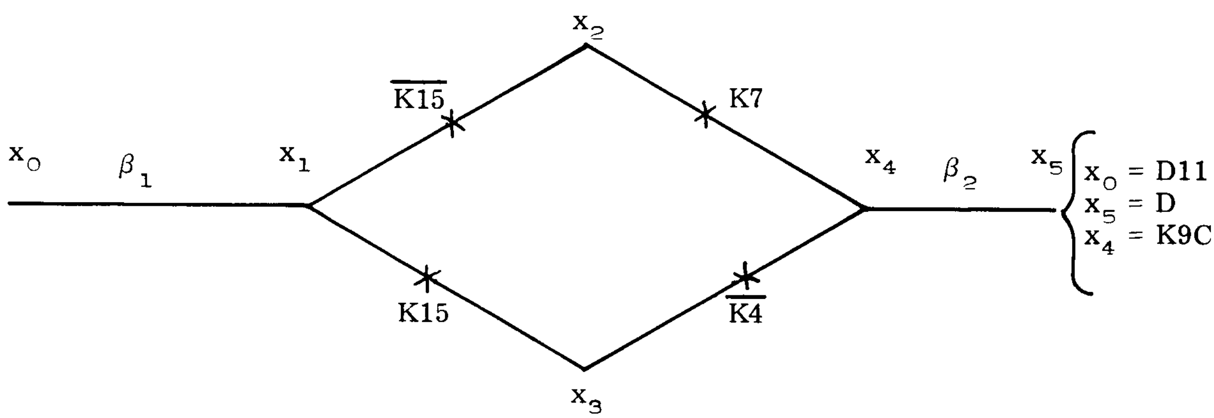


Figure B-3. Typical Relay Circuit

Here the problem is to define the Boolean circuit equation for the K-C contact elements in terms of the K9C coil energization with make and break time delays. The K9C coil is represented by node x_4 in Figure B-3.

In order to incorporate the time delay, the graph in Figure B-3 must be rearranged to indicate the dual states of K9C as represented by node x_4, i.e., x_4 has the value x_4 and $\overline{x}_4$:

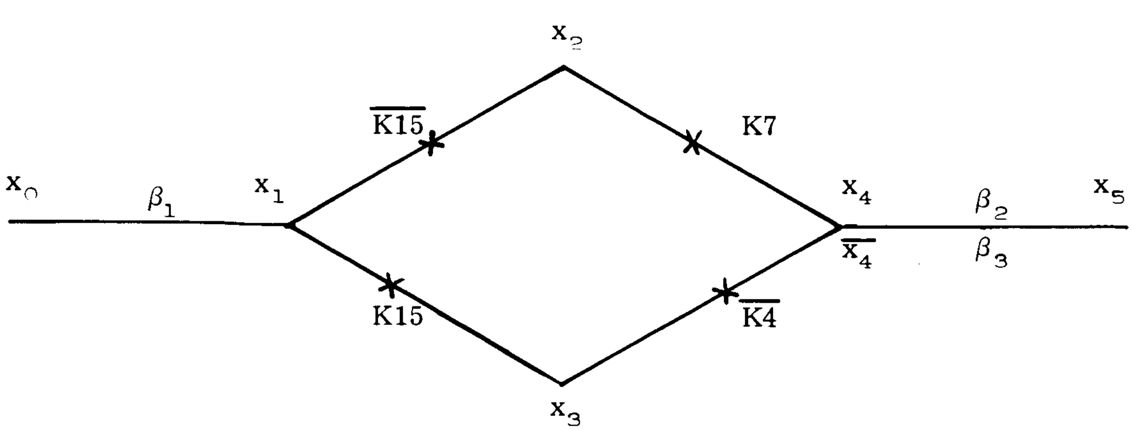


Figure B-4. Relay Circuit With Delay

If the make delay factor is e^{-ms} and the break delay factor is $1 - e^{-bs}$ the quantities $A = \beta_2 e^{-ms}$ and $B = \beta_3 (1 - e^{-bs})$ can be defined where β_2 and β_3 simply indicate the node connection conditions.

Figure B-4 yields the Boolean connection matrix:

$$
M = \begin{array}{c|ccccccc}
 & x_0 & x_5 & x_1 & x_2 & x_3 & x_4 & \overline{x}_4 \\
\hline
x_0 & 1 & & \beta_1 & & & & \\
x_5 & & 1 & & & & A & B \\
x_1 & \beta_1 & & 1 & \overline{K15} & K15 & & \\
x_2 & & & \overline{K15} & 1 & & K7 & \overline{K7} \\
x_3 & & & K15 & & 1 & \overline{K4} & K4 \\
x_4 & & A & & K7 & \overline{K4} & 1 & \\
\overline{x}_4 & & B & & \overline{K7} & K4 & & 1
\end{array} \qquad \text{(B-8)}
$$

which on reduction yields the path product

$$
\pi(x_0, x_5) = \beta_1 [A(K15\,\overline{K4} + K7\,\overline{K15}) + B(K15\,K4 + \overline{K7}\,\overline{K15})] \qquad \text{(B-9)}
$$

or equivalently

$$
\pi(x_0, x_5) = \beta_1 \left[\beta_2 \; e^{-ms} (K15\,\overline{K4} + K7\,\overline{K15}) + \beta_3 (1 - e^{-bs})(K15K4 + \overline{K7}\;\overline{K15})\right] \qquad \text{(B-10)}
$$

and is the Boolean equation of the circuit with time delay.

The Boolean connection matrix, M, can also be used in conjunction with the system transition matrix T_s, to relate the Boolean functions with system dynamics.

In the dynamic system, as illustrated by Figure A-5, the transition matrix T (Equation A-1), is a subdiagonal matrix with 1 as the first element of the main diagonal and 0 in the remaining elements. If these zeros are converted to ones the T matrix is transformed to a Boolean connection matrix and can be reduced in the Boolean manner.

Thus

$$
M_T = \begin{array}{c|cccc}
 & x_1 & x_2 & x_3 & x_4 \\
\hline
x_1 & 1 & & & \\
x_2 & t_{12} & 1 & & \\
x_3 & t_{13} & t_{23} & 1 & \\
x_4 & t_{14} & t_{24} & t_{34} & 1
\end{array} \tag{B-11}
$$

and the corresponding frequency sensitive path from node x_1 to x_4 may be obtained as follows:

$$
\pi(x_1, x_4) = \begin{array}{c|cccc}
 & x_1 & x_4 & x_2 & x_3 \\
\hline
x_1 & 1 & & & \\
x_4 & t_{14} & 1 & t_{24} & t_{34} \\
x_2 & t_{12} & & 1 & \\
x_3 & t_{13} & & t_{23} & 1
\end{array} \tag{B-12}
$$

$$
= \begin{array}{c|ccc}
 & x_1 & x_4 & x_2 \\
\hline
x_1 & 1 & & \\
x_4 & t_{14} + t_{13}t_{34} & 1 & t_{24} + t_{23}t_{34} \\
x_2 & t_{12} & & 1
\end{array} \tag{B-13}
$$

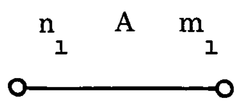
$$
P_{14} = (t_{14} + t_{13}t_{34}) + t_{12}(t_{24} + t_{23}t_{34}) \tag{B-14}
$$

Equation B-14 can be verified by inspection of Figure A-5 for the condition in which all the contacts are closed, i.e., $\alpha_{12} = \alpha_{13} = \alpha_{14} = \alpha_{23} = \alpha_{24} = \alpha_{34} = 1$.

B3.3 ILLUSTRATIVE EXAMPLES OF BOOLEAN CIRCUIT REDUCTION

In this development input nodes will be designated by n_1, n_2, ..., etc., output nodes by m_1, m_2, ..., etc., internal nodes by numerals, and the Boolean variable by A, B, C, ..., etc. Blank spaces in the connection matrix represent zeros.

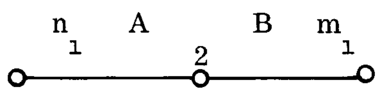
1.



$$\begin{array}{c|cc} & n_1 & m_1 \\ \hline n_1 & 1 & A \\ m_1 & A & 1 \end{array}$$

$\pi(n_1, m_1) = A$

2.

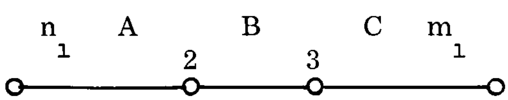


$$\begin{array}{c|ccc} & n_1 & 2 & m_1 \\ \hline n_1 & 1 & A & \\ 2 & A & 1 & B \\ m_1 & & B & 1 \end{array}$$

$$\begin{array}{c|ccc} & n_1 & m_1 & 2 \\ \hline n_1 & 1 & & A \\ m_1 & & 1 & B \\ 2 & A & B & 1 \end{array} \longrightarrow \begin{array}{c|c} & n_1 \\ \hline n_1 & 1 \\ m_1 & AB \end{array}$$

$\pi(n_1, m_1) = AB$

3.



$$\begin{array}{c|cccc} & n_1 & 2 & 3 & m_1 \\ \hline n_1 & 1 & A & & \\ 2 & A & 1 & B & \\ 3 & & B & 1 & C \\ m_1 & & & C & 1 \end{array}$$

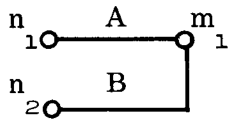
$$\begin{array}{c|cccc} & n_1 & m_1 & 2 & 3 \\ \hline n_1 & 1 & & A & \\ m_1 & & 1 & & C \\ 2 & A & & 1 & B \\ 3 & & C & B & 1 \end{array}$$

$$\begin{array}{c|ccc} & n_1 & m_1 & 2 \\ \hline n_1 & 1 & & A \\ m_1 & & 1 & BC \\ 2 & A & BC & 1 \end{array}$$

$$\begin{array}{c|c} & n_1 \\ \hline n_1 & 1 \\ m_1 & ABC \end{array}$$

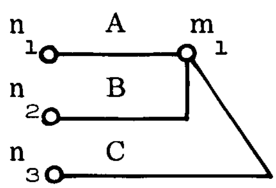
$\pi(n_1, m_1) = ABC$

4.



$$\begin{array}{c|ccc} & n_1 & m_1 & n_2 \\ \hline n_1 & 1 & A & \\ m_1 & A & 1 & B \\ n_2 & & B & 1 \end{array} \longrightarrow \begin{array}{c|c} & n_1 \\ \hline n_1 & 1 \\ m_1 & A \end{array}$$

$$\begin{array}{c|ccc} & n_2 & m_1 & n_1 \\ \hline n_2 & 1 & B & \\ m_1 & B & 1 & A \\ n_1 & & A & 1 \end{array} \longrightarrow \begin{array}{c|c} & n_2 \\ \hline n_2 & 1 \\ m_1 & B \end{array}$$

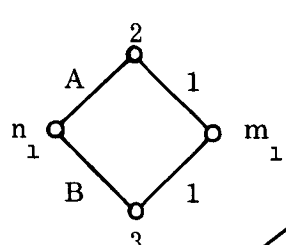


$$\begin{array}{c|cccc} & n_1 & m_1 & n_2 & n_3 \\ \hline n_1 & 1 & A & & \\ m_1 & A & 1 & B & C \\ n_2 & & B & 1 & \\ n_3 & & C & & 1 \end{array}$$

$$\begin{array}{c|ccc} & n_1 & m_1 & n_2 \\ \hline n_1 & 1 & A & \\ m_1 & A & 1 & B \\ n_2 & & B & 1 \end{array} \longrightarrow \begin{array}{c|c} & n_1 \\ \hline n_1 & 1 \\ m_1 & A \end{array}$$

Similar results are obtained for remaining paths.

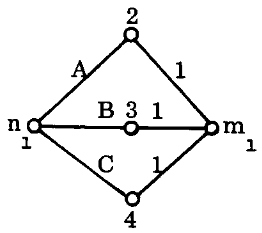
5.



$$\begin{array}{c|cccc} & n_1 & m_1 & 2 & 3 \\ \hline n_1 & 1 & & A & B \\ m_1 & & 1 & 1 & 1 \\ 2 & A & 1 & 1 & \\ 3 & B & 1 & & 1 \end{array}$$

$$\begin{array}{c|ccc} & n_1 & m_1 & 2 \\ \hline n_1 & 1 & B & A \\ m_1 & B & 1 & 1 \\ 2 & A & 1 & 1 \end{array} \longrightarrow \begin{array}{c|c} & n_1 \\ \hline n_1 & 1 \\ m_1 & B + A \end{array}$$

5. Continued



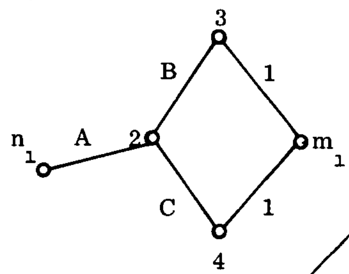
	n_1	m_1	2	3	4
n_1	1		A	B	C
m_1		1	1	1	1
2	A	1	1		
3	B	1		1	
4	C	1			1

	n_1	m_1	2	3
n_1	1	C	A	B
m_1	C	1	1	1
2	A	1	1	
3	B	1		1

	n_1	m_1	2
n_1	1	C + B	A
m_1	C + B	1	1
2	A	1	1

	n_1
n_1	1
m_1	A + B + C

6.1



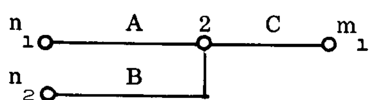
	n_1	m_1	2	3	4
n_1	1		A		
m_1		1		1	1
2	A		1	B	C
3		1	B	1	
4		1	C		1

	n_1	m_1	2	3
n_1	1		A	
m_1		1	C	1
2	A	C	1	B
3		1	B	1

	n_1	m_1	2
n_1	1		A
m_1		1	B + C
2	A	B + C	1

	n_1
n_1	1
m_1	A(B + C)

6.2



	n_1	m_1	n_2	2
n_1	1			A
m_1		1		C
n_2			1	B
2	A	C	B	1

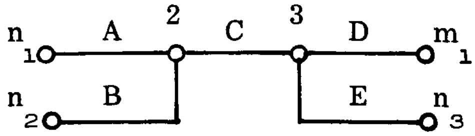
	n_1	m_1	n_2
n_1	1	AC	AB
m_1	AC	1	BC
n_2	AB	BC	1

	n_1
n_1	1
m_1	AC

$\pi(n_1, m_1) = AC$

Same procedure for $\pi(n_2, m_1)$.

6.3



	n_1	m_1	n_2	n_3	2	3
n_1	1				A	
m_1		1				D
n_2			1		B	
n_3				1		E
2	A		B		1	C
3		D		E	C	1

	n_1	m_1	n_2	n_3	2
n_1	1				A
m_1		1		DE	CD
n_2			1		B
n_3		DE		1	CE
2	A	CD	B	CE	1

	n_1	m_1	n_2	n_3
n_1	1	ACD	AB	ACE
m_1	ACD	1	BCD	DE
n_2	AB	BCD	1	BCE
n_3	ACE	DE	BCE	1

	n_1	m_1	n_2
n_1	1	ACD	AB
m_1	ACD	1	BCD
n_2	AB	BCD	1

	n_1
n_1	1
m_1	ACD

	n_2	m_1	n_1	n_3	2	3
n_2	1				B	
m_1		1				D
n_1			1		A	
n_3				1		E
2	B		A		1	C
3		D		E	C	1

	n_2	m_1	n_1	n_3	2
n_2	1				B
m_1		1		DE	DC
n_1			1		A
n_3		DE		1	CE
2	B	DC	A	CE	1

	n_2	m_1	n_1	n_3
n_2	1	BDC	AB	BCE
m_1	BDC	1	ADC	DE
n_1	AB	ADC	1	ACE
n_3	BCE	DE	ACE	1

	n_2	m_1	n_1
n_2	1	BDC	AB
m_1	BDC	1	ADC
n_1	AB	ADC	1

	n_2
n_2	1
m_1	BDC

6.3 Continued

$$
\begin{array}{c|cccccc}
 & n_3 & m_1 & n_1 & n_2 & 2 & 3 \\
\hline
n_3 & 1 & & & & & E \\
m_1 & & 1 & & & & D \\
n_1 & & & 1 & & A & \\
n_2 & & & & 1 & B & \\
2 & & & A & B & 1 & C \\
3 & E & D & & & C & 1
\end{array}
\longrightarrow
\begin{array}{c|ccccc}
 & n_3 & m_1 & n_1 & n_2 & 2 \\
\hline
n_3 & 1 & DE & & & CE \\
m_1 & DE & 1 & & & CD \\
n_1 & & & 1 & & A \\
n_2 & & & & 1 & B \\
2 & CE & CD & A & B & 1
\end{array}
$$

$$
\longrightarrow
\begin{array}{c|cccc}
 & n_3 & m_1 & n_1 & n_2 \\
\hline
n_3 & 1 & DE & ACE & BCE \\
m_1 & DE & 1 & ADC & BCD \\
n_1 & ACE & ADC & 1 & AB \\
n_2 & BCE & BCD & AB & 1
\end{array}
\longrightarrow
\begin{array}{c|ccc}
 & n_3 & m_1 & n_1 \\
\hline
n_3 & 1 & DE & ACE \\
m_1 & DE & 1 & ADC \\
n_1 & ACE & ADC & 1
\end{array}
\longrightarrow
\begin{array}{c|c}
 & n_3 \\
\hline
n_3 & 1 \\
m_1 & DE
\end{array}
$$

Total condition at m_1 = sum of all paths (s stands for source nodes)

$$\pi(s, m_1) = DE + BDC + ACD$$

which can be checked by inspection. Hence by the same principles all source n's can be treated in a repeated manner with respect to the internal nodes and the output node.

Whence:

$$
\begin{array}{c|cccc}
 & n_1 & m_1 & 2 & 3 \\
\hline
n_1 & 1 & & A & \\
m_1 & & 1 & & D \\
2 & A & & 1 & C \\
3 & & D & C & 1
\end{array}
\longrightarrow
\begin{array}{c|ccc}
 & n_1 & m_1 & 2 \\
\hline
n_1 & 1 & & A \\
m_1 & & 1 & CD \\
2 & A & CD & 1
\end{array}
\longrightarrow
\begin{array}{c|c}
 & n_1 \\
\hline
n_1 & 1 \\
m_1 & ACD
\end{array}
$$

6.3 Continued

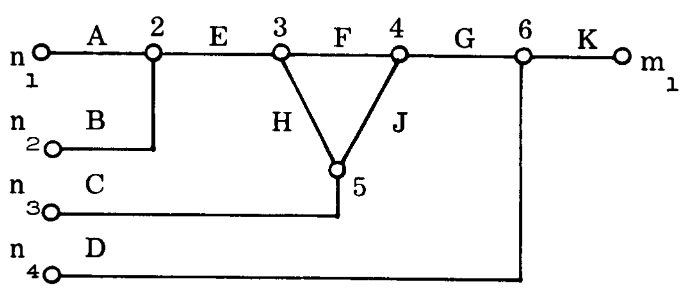
$$
\begin{array}{c|cccc}
 & n_2 & m_1 & 2 & 3 \\
\hline
n_2 & 1 & & B & \\
m_1 & & 1 & & D \\
2 & B & & 1 & C \\
3 & & D & C & 1
\end{array}
\longrightarrow
\begin{array}{c|ccc}
 & n_2 & m_1 & 2 \\
\hline
n_2 & 1 & & B \\
m_1 & & 1 & CD \\
2 & B & CD & 1
\end{array}
\longrightarrow
\begin{array}{c|c}
 & n_2 \\
\hline
n_2 & 1 \\
m_1 & BCD
\end{array}
$$

$$
\begin{array}{c|cccc}
 & n_3 & m_1 & 2 & 3 \\
\hline
n_3 & 1 & & & E \\
m_1 & & 1 & & D \\
2 & & & 1 & C \\
3 & E & D & C & 1
\end{array}
\longrightarrow
\begin{array}{c|ccc}
 & n_3 & m_1 & 2 \\
\hline
n_3 & 1 & DE & CE \\
m_1 & DE & 1 & CD \\
2 & CE & CD & 1
\end{array}
\longrightarrow
\begin{array}{c|c}
 & n_3 \\
\hline
n_3 & 1 \\
m_1 & DE
\end{array}
$$

whence $\pi(s, m_1) \;=\; DE + BCD + ACD$

which agrees with the above reduction.

6.4



	n_1	n_2	n_3	n_4	m_1	2	3	4	5	6
n_1	1					A				
n_2		1				B				
n_3			1						C	
n_4				1						D
m_1					1					K
2	A	B				1	E			
3						E	1	F	H	
4							F	1	J	G
5			C				H	J	1	
6				D	K			G		1

This example will be carried out by iteration of the n's.

6.4 Continued

a.

	n_1	m_1	2	3	4	5	6
n_1	1		A				
m_1		1					K
2	A		1	E			
3			E	1	F	H	
4				F	1	J	G
5				H	J	1	
6		K			G		1

	n_1	m_1	2	3	4	5
n_1	1		A			
m_1		1			KG	
2	A		1	E		
3			E	1	F	H
4		KG		F	1	J
5				H	J	1

	n_1	m_1	2	3	4
n_1	1		A		
m_1		1			KG
2	A		1	E	
3			E	1	F + HJ
4		KG		F + HJ	1

	n_1	m_1	2	3
n_1	1		A	
m_1		1		KG(F + HJ)
2	A		1	E
3		KG(F + HJ)	E	1

	n_1	m_1	2
n_1	1		A
m_1		1	KG(F + HJ)E
2	A	KG(F + HJ)E	1

$$\pi(n_1, m_1) = KG(F + HJ)AE$$

6.4 Continued

b.

	n_2	m_1	2	3	4	5	6
n_2	1		B				
m_1		1					K
2	B		1	E			
3			E	1	F	H	
4				F	1	J	G
5				H	J	1	
6		K			G		1

	n_2	m_1	2	3	4	5
n_2	1		B			
m_1		1			KG	
2	B		1	E		
3			E	1	F	H
4		KG		F	1	J
5				H	J	1

	n_2	m_1	2	3	4
n_2	1		B		
m_1		1			KG
2	B		1	E	
3			E	1	F + HJ
4		KG		F + HJ	1

	n_2	m_1	2	3
n_2	1		B	
m_1		1		KG(F + HJ)
2	B		1	E
3		KG(F + HJ)	E	1

	n_2	m_1	2
n_2	1		B
m_1		1	KG(F + HJ)E
2	B	KG(F + HJ)E	1

	n_2
n_2	1
m_1	KG(F + HJ)EB

Whence:

$$\pi(n_2, m_1) = KG(F + HJ)BE.$$

6.4 Continued

c.

$$
\begin{array}{c|ccccccc}
 & n_3 & m_1 & 2 & 3 & 4 & 5 & 6 \\
\hline
n_3 & 1 & & & & & C & \\
m_1 & & 1 & & & & & K \\
2 & & & 1 & E & & & \\
3 & & & E & 1 & F & H & \\
4 & & & & F & 1 & J & G \\
5 & C & & & H & J & 1 & \\
6 & & K & & & G & & 1
\end{array}
$$

$$
\begin{array}{c|cccccc}
 & n_3 & m_1 & 2 & 3 & 4 & 5 \\
\hline
n_3 & 1 & & & & & C \\
m_1 & & 1 & & & KG & \\
2 & & & 1 & E & & \\
3 & & & E & 1 & F & H \\
4 & & KG & & F & 1 & J \\
5 & C & & & H & J & 1
\end{array}
$$

$$
\begin{array}{c|ccccc}
 & n_3 & m_1 & 2 & 3 & 4 \\
\hline
n_3 & 1 & & & CH & CJ \\
m_1 & & 1 & & & KG \\
2 & & & 1 & E & \\
3 & CH & & E & 1 & F + HJ \\
4 & CJ & KG & & F + HJ & 1
\end{array}
$$

$$
\begin{array}{c|cccc}
 & n_3 & m_1 & 2 & 3 \\
\hline
n_3 & 1 & KGCJ & & CH + CJ(F + HJ) \\
m_1 & KGCJ & 1 & & KG(F + HJ) \\
2 & & & 1 & E \\
3 & CH + CJ(F + HJ) & KG(F + HJ) & E & 1
\end{array}
$$

$$
\begin{array}{c|ccc}
 & n_3 & m_1 & 2 \\
\hline
n_3 & 1 & KG[CJ + CH(F + HJ)] & E[CH + CJ(F + HJ)] \\
m_1 & KG[CJ + CH(F + HJ)] & 1 & KG(F + HJ)E \\
2 & E[CH + CJ(F + HJ)] & KG(F + HJ)E & 1
\end{array}
$$

whence:

$$
\begin{aligned}
\pi(n_3, m_1) &= KG\{CJ + CH(F + HJ) + ECH(F + HJ) + ECJ(F + HJ)\} \\
&= KG\{CJ + CH(F + JH)\} \\
&= KG(CJ + CHF)
\end{aligned}
$$

6.4 Continued

d.

	n_4	m_1	2	3	4	5	6
n_4	1						D
m_1		1					K
2			1	E			
3			E	1	F	H	
4				F	1	J	G
5				H	J	1	
6	D	K			G		1

	n_4	m_1	2	3	4	5
n_4	1	KD			GD	
m_1	KD	1			GK	
2			1	E		
3			E	1	F	H
4	GD	GK		F	1	J
5				H	J	1

	n_4	m_1	2	3	4
n_4	1	KD			GD
m_1	KD	1			GK
2			1	E	
3			E	1	F + HJ
4	GD	GK		F + HJ	1

	n_4	m_1	2	3
n_4	1	KD		GD(F + HJ)
m_1	KD	1		GK(F + HJ)
2			1	E
3	GD(F + HJ)	GK(F + HJ)	E	1

	n_4	m_1	2
n_4	1	KD	GD(F + HJ)E
m_1	KD	1	GK(F + HJ)E
2	GD(F + HJ)E	GK(F + HJ)E	1

whence:

$$\pi(n_4, m_1) = KD$$

and all paths have been correctly calculated.

B4 PATH TRACING VIA ADJACENT NODES

B4.1 INTRODUCTION

During this study, a new method for handling a large data base of discrete variables has been devised. The method and analysis are presented in this section.

The very nature of the problem of simulating a system having a data base consisting of many elements calls for efficient methods of manipulating the data base to achieve the various objectives. The method suggested here considers the problem of discrete elements (relays, switches, lamps, recording pens, etc.) primarily, but its concepts should apply equally well to the more general case in which dynamic response is also an issue. Paragraph B4.7 touches briefly on this aspect.

Experience with the ESE simulation program has shown that it is feasible to handle a large discrete data base by iterative techniques. The system is given an initial status, then input stimuli are introduced, and their effects are determined by evaluating the complete set of logical equations over and over until no changes occur on successive iterations. Simulated countdown proceeds automatically to liftoff by specifying the time at which each input is to become effective, and allowing for time delays in the pickup and dropout of the relays.

A cross-reference program permits the listing of all equations in which each element appears. No distinction in nomenclature is made between a relay coil and the contacts of that relay.

The iterated-equation approach is the first of three methods that will be compared in this section. Its distinct advantage over the other two methods is that it is an operating technique, already programmed and tested.

A second method, discussed in Appendix A, proposes to handle the data base in the Launch Vehicle Component Level Simulation program by matrix methods. A connection matrix will contain the discrete elements, and a related transition matrix the dynamic transfer functions. To process the dynamic effects of input stimuli, it will first be necessary to determine the proper paths through the connection matrix. Such path analysis can theoretically be performed by matrix reduction techniques regardless of matrix size. Once the path is determined, over-all dynamic response can be found by use of a number series to represent the responses of the individual elements - provided that input and output nodes can be suitably identified.

The new method - path tracing via adjacent nodes - is in a sense a matrix approach, although the data base will not require two entries of each element, α_{ij} and α_{ji}, as in the connection matrix of Appendix A. Its outstanding feature is the fact that no explicit matrix operations will ever be performed. Rather, all status and connection information will reside at all times within the data base, and programs designed to establish paths, change status, etc., will operate specifically only upon the circuit elements involved. Paths will be followed by referring to all nodes adjacent to either upper or lower bounds of each involved element.

The subsequent paragraphs of this section solve a specific discrete response example by each of the three methods just described. The objective is to estimate as accurately as possible the amount of computer activity required to program each method. It is concluded that method three, the new method, has pronounced advantages over the other two for some applications. Its consideration in the methodologies of both ESE Simulation and Launch Vehicle Component Level Simulation is therefore recommended.

B4.2 COMPARISON CIRCUIT

The discrete response of the circuit which will be compared is shown in Figure B-5. It is typical of the circuits found in the electrical support equipment of launch vehicles simulated to date. Elements K_A, K_B, ..., K_P represent relay coils. Contacts of these relays are identified by the subscript letter alone (i.e., contacts of K_A are designated A if normally open, $\overline{A}$ if normally closed). Switches K, L, M, N, O, J, X, and H represent input stimuli (i.e., independent variables). Node 00 is the power supply node, and node 99 the common ground.

The specific objective of the comparison will be to estimate the number of computer instructions necessary to cause the simulated circuit to complete its response to the following sequence of input stimuli:

a. J = 1.

b. K = 1.

c. X = 1.

d. H = 1.

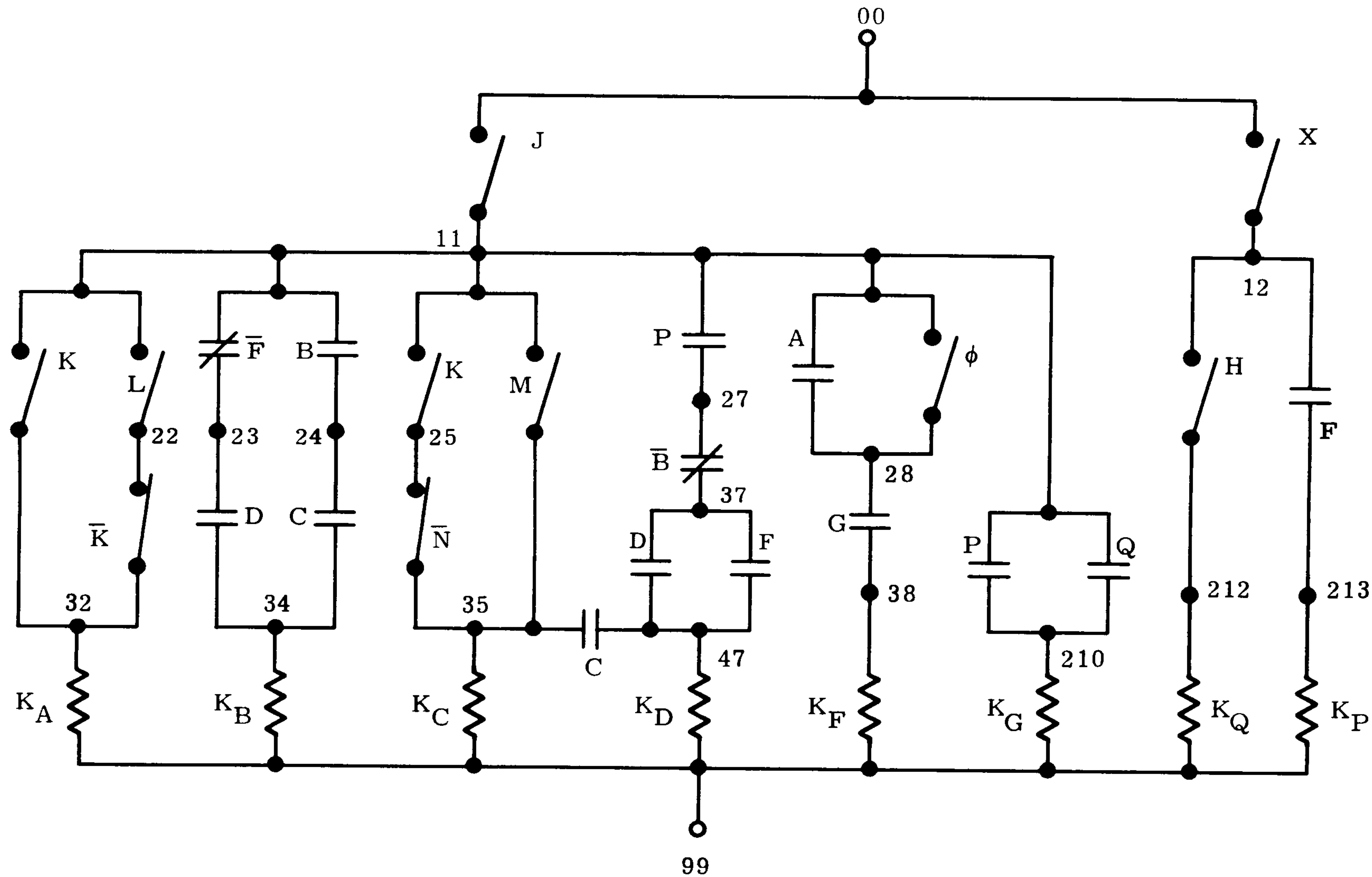


Figure B-5. Comparison Circuit

e. $N = 1$.

f. $M = 1$.

Initially, the circuit is assumed to have all relays de-energized (logical state 0) and all switches open (logical status 0). It is further assumed that the circuit has adequate time to stabilize between inputs. That is, the chain of events initiated by an input will complete itself before the next input is applied.

Figure B-6 is a timing chart on which the status changes resulting from each input can be identified. Note, for example, that when K becomes 1, K_A is energized. This causes K_C to energize, which energizes K_D. Finally, K_B is energized by closure of D contact, to complete the response to the K = 1 stimulus. The circuit is now stabilized and nothing else will happen until another input change occurs.

B4.3 ITERATED-EQUATION METHOD

The logical equations for the eight relays of Figure B-5 appear below, written almost as they would appear in the existing ESE simulations. Three minor differences are the use of different symbols for a relay coil and its contacts (e.g., $-K_A$ for the coil, A for the contact), the use of single letters for Boolean variables (rather than the six-character ESE code), and omission of the explicit AND operator symbol (which appears as * in ESE simulations).

$$K_A = J(K + \overline{K}L)$$

$$K_B = J(\overline{F}D + BC)$$

$$K_D = J[P\overline{B}(D + F) + C(K\overline{N} + M)]$$

$$K_C = J[(K\overline{N} + M) + C(D + F)P\overline{B}]$$

$$K_G = J(P + Q)$$

$$K_F = J(A + \phi)G$$

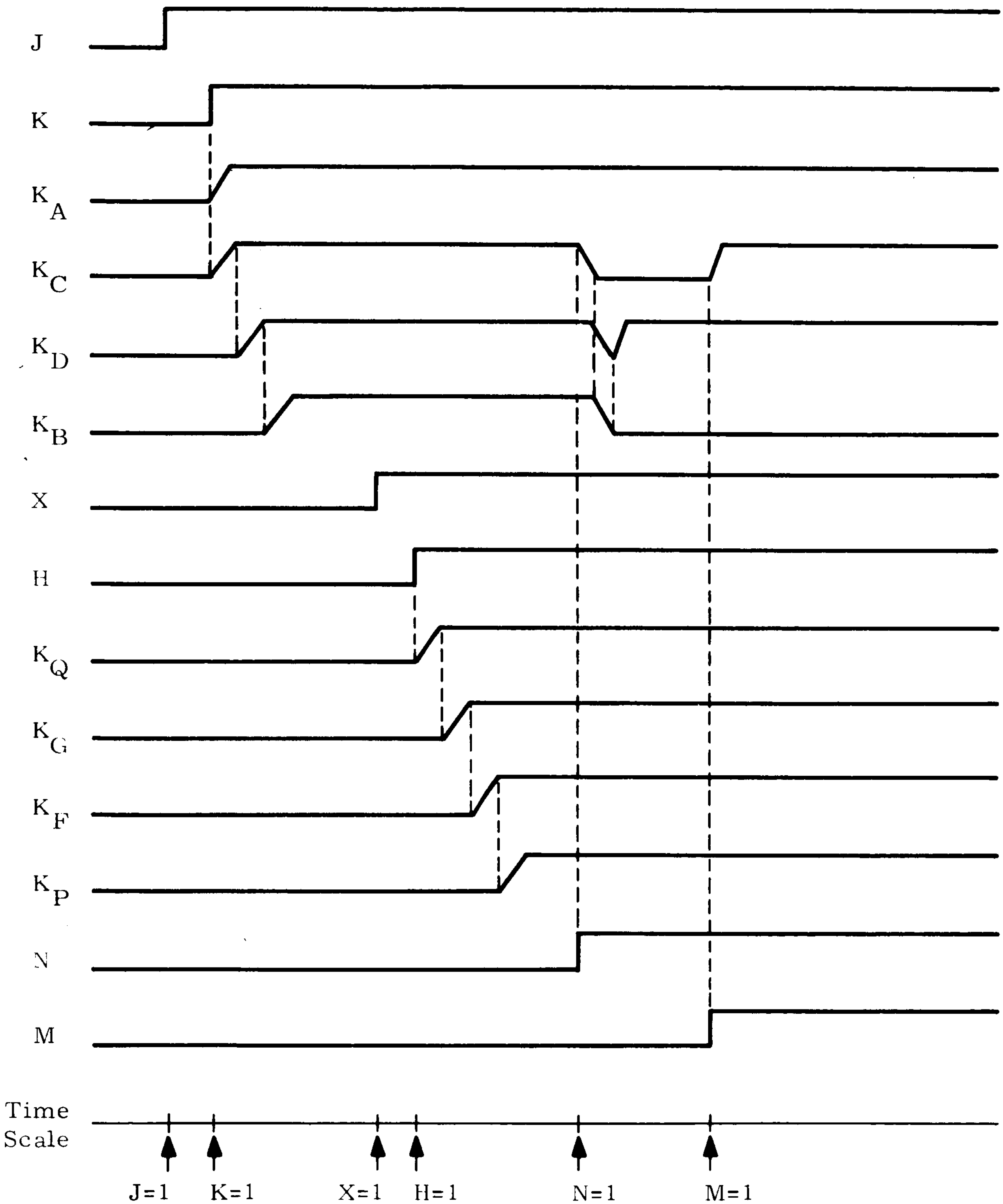


Figure B-6. Timing Chart

$K_P = XF$

$K_Q = XH$

According to ESE simulation procedures, these logical equations would be compiled by a line-scanning compiler to produce a machine language program which, when initiated by insertion of the input stimuli, would update the state of the equation set. It is estimated that the eight equations in question would produce a total of 68 instructions when compiled. Hence one iteration through the complete set would require execution of 76 instructions - 68 to evaluate the relay states, and eight comparisons to determine whether or not state changes occurred.

Figure B-7 shows the sequential conditions of state for the eight relays, and the eight switches of Figure B-5 as the input stimuli are applied in the aforementioned order. Circled 1's and 0's indicate states that have just changed. Any column containing one or more element other than stimuli that have circled states will require another iteration (i.e., another column) to ascertain whether or not stabilization has been reached. Thus, the number of columns corresponds directly to the number of iterations. If an input stimulus change does not cause any other element to change, a second iteration is not required. Note the implicit assumption that the equations are evaluated in the order in which they are listed. As a consequence of this, even though closure of switch K energizes both K_C and K_D, an extra iteration is required to find this out because K_C happens to come after K_D in the listing. If the order of K_C and K_D were reversed, the energizing of both would be recorded on the first pass following $K = 1$, and an iteration would be saved. By the same token, if K_P were listed before K_F, the $H = 1$ stimulus would require an additional iteration. The thought injected here is simply that the computer time required by the iterated-equation approach might be optimized by appropriate equation arrangement. To date, no attempt has been made to implement this possibility within the ESE framework.

Figure B-7 shows that 15 iterations are required to stabilize this eight-equation system subject to the indicated input sequence. At 76 instructions per iteration, this predicts a total number of executed instructions equal to 15 x 76 = 1140.

ITERATION NUMBER

Elements	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
J	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
K	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
M	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
X	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
H	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
ϕ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K_A	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
K_B	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
K_D	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1
K_C	0	1	1	1	1	1	1	1	1	1	0	0	0	1	1
K_G	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
K_F	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
K_P	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
K_Q	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	J=1	K=1				X=1	H=1				N=1			M=1	

Point at which input stimulus is introduced

Figure B-7. State Table Changes

B4.4 MATRIX REDUCTION METHOD

In this approach, the discrete data base is organized as a connection matrix. Both columns and rows of the matrix are labeled with the nodes of the circuit to be processed. Each Boolean variable appears twice: as α_{ij} and also as α_{ji}. When a stimulus is introduced, the general methodology of Appendix A calls for a reduction of the matrix to determine the logical path between two nodes designated respectively as input and output. Immediately a difficulty arises, for the nature of the circuit of Figure B-5 is such that while inputs are uniquely defined (the 8 stimuli), there are many outputs (the intermediate relay responses leading to circuit stabilization). Another way of stating the problem is that many logical paths may be involved in the response to a single input, and path i will not be known until path j has reacted.

To enable method comparison, the interpretation used here will be that the final circuit status occasioned by applying a stimulus can be obtained by reducing the matrix in turn to each of the logical paths involved - both those that contain the stimulus explicitly and those activated by secondary switchings. Applicable paths must be traced from power supply (node 00, Figure B-5) to an element the lower bound of which is node 99. In Figure B-5, this means one of the 8 relays.

To illustrate, consider the events triggered by closure of switch K, assuming that J was previously closed. First it will be necessary to retrieve (by matrix reduction) the paths for K_A and K_C, the relays directly dependent upon K. Since both K_A and K_C are energized by closure of K, the next step will be to retrieve the paths of all relays the operation of which is dependent upon contacts of either K_A or K_C and investigate their response. In this case, K_B and K_D are involved. K_D will energize. Further retrievals then will reveal the fact that K_B is picked up by a contact of K_D. Stabilization due to this input will require six separate matrix reductions.

Since this method has not as yet been reduced to practice for a circuit of this type, the discussion will not be carried further here, except to point out that certain problems of path identification seem to be inherent in it. That is, the answer to the question of what paths to retrieve next may require stored information in excess of element name and bounding nodes.

Figure B-8 shows the connection matrix for the circuit of Figure B-4. It is an 18 by 18 array, mostly vacuous. The main diagonal is of course all 1's. Note the omission of ground node 99. It is implicit that all relay coils terminate there, and the practical

	00	47	11	12	22	23	24	25	27	28	210	212	213	32	34	35	37	38
00	1		J	X														
47		1														C	D+F	
11	J		1		L	$\bar{F}$	B	K	P	A+ϕ	P+Q			K		M		
12	X			1								H	F					
22			L		1									$\bar{K}$				
23			$\bar{F}$			1									D			
24			B				1								C			
25			K					1								$\bar{N}$		
27			P						1								$\bar{P}$	
28			A+ϕ							1								G
210			P+Q								1							
212				H								1						
213				F									1					
32			K		$\bar{K}$									1				
34						D	C								1			
35		C	M					$\bar{N}$								1		
37		D+F							$\bar{B}$								1	
38										G								1

Figure B-8. Connection Matrix

question to be answered is: "Is the path complete from power supply (node 00) to the ungrounded end of a relay coil?" If so, the relay will energize.

To find the logical path between any pair of nodes, the following steps are employed:

a. Rearrange rows and columns so that the nodes the connecting path of which is to be investigated occupy positions 1 and 2. (Note that Figure B-8 is set up to investigate the path between nodes 00 and 47.)

b. Eliminate the other 16 nodes to form a 2 x 2 matrix containing only those nodes the connecting path of which is to be investigated. Nodes are eliminated one at a time by repeated use of the Boolean relationship:

$$E_{jk} = C_{jk} + C_{ji}C_{ik},$$

where

E_{jk} is the element of the reduced matrix between nodes j and k.

C_{jk} is the corresponding element prior to reduction.

C_{ji} is the element of the unreduced matrix between nodes j and i (i is the node being eliminated).

C_{ik} is the element of the unreduced matrix between nodes i and k.

Each evaluation of the E_{jk} formula would involve five computer instructions: a FETCH, an AND, an OR, and two STORE's. To reduce an 18 x 18 matrix to 2 x 2 would require 816* applications of this formula, assuming a straightforward use of the matrix reduction algorithm that makes no attempt to capitalize on the vacuous nature of the array. The total instruction count would therefore be 5 x 816 = 4080.

A much more efficient reduction algorithm could undoubtedly be devised. It might proceed as follows:

a. Search the column of the node to be eliminated for nonzero entries (this would take two instructions: a FETCH and a TEST - per zero entry).

b. Apply the E_{jk} formula to all possible pairs of nonzero entries in the column.

Using this approach, evaluation of the 00-47 path in Figure B-5 would require about 371* instructions instead of 4080.

*See Addendum B1 of this appendix for derivation of this number.

The foregoing instruction count estimates assume that all that is required from the matrix is path status evaluation. If, as in the treatment of Reference 1, the actual Boolean equations must be generated in order to enable association of discrete paths with dynamic transfer functions, the storage and data handling problems become more complicated because of the need for manipulating the alphanumeric symbols for all circuit elements and logical operators. Figure B-9 shows the result of reducing the matrix of Figure B-8 to obtain the Boolean equation for the path from node 00 to node 47. The equation is:

$$T_{00-47} = K_D = J[CM + P\overline{B}(D + F) + C\overline{N}(K + M\overline{N})] .$$

This equation is logically redundant - as, in general, all such equations will be that are derived directly from matrix reduction The simplest form to which it can be reduced explicitly is:

$$T_{00-47} = K_D = J[C(M + \overline{N}K) + P\overline{B}(D + F)] .$$

	00	47
00	1	$J[CM + P\overline{B}(D + F) + C\overline{N}(K + M\overline{N})]$
47	$J[CM + P\overline{B}(D + F) + C\overline{N}(K + M\overline{N})]$	1

Figure B-9. Reduced Matrix (with reduction steps listed)

To determine path from 00 to 47:

a. Eliminate node 38: no effect on remainder of matrix.
b. Eliminate node 37: $E_{27-47} = C_{27-47} + C_{27-47}C_{37-47} = \overline{B}(D + F)$.
c. Eliminate node 35: $E_{47-11} = CM$, $E_{47-25} = C\overline{N}$, $E_{11-25} = K + M\overline{N}$.
d. Eliminate node 34: $E_{23-24} = DC$.
e. Eliminate node 32: $E_{11-22} = L + K\overline{K}$.
f. Eliminate node 213: No effect on remainder of matrix.
g. Eliminate node 212: No effect on remainder of matrix.

h. Eliminate node 210: No effect on remainder of matrix.

i. Eliminate node 28: No effect on remainder of matrix.

j. Eliminate node 27: $E_{47-11} = CM + P\overline{B}(D + F)$.

k. Eliminate node 25: $E_{47-11} = CM + P\overline{B}(D + F) + C\overline{N}(K + M\overline{N})$.

l. Eliminate node 24: $E_{11-23} = \overline{F} + BCD$.

m. Eliminate node 23: No effect on remainder of matrix.

n. Eliminate node 22: No effect on remainder of matrix.

o. Eliminate node 12: No effect on remainder of matrix.

p. Eliminate node 11: $E_{00-47} = J[CM + P\overline{B}(D + F) + C\overline{N}(K + M\overline{N})]$.

Further, it is evident from Figure B-5 that if the use of relays is allowed on the right-hand side of logical equations, the logic of the T_{00-47} path can be expressed by the following:

$$T_{00-47} = K_D = K_C + JP\overline{B}(D + F).$$

The simplification algorithm is a complicated thing for a data base of many variables. See Reference 5 for a detailed discussion of the subject. Suffice it to say that Boolean simplification must occur either at each application of the E_{jk} formula, or else as a final step, operating on the 2 x 2 matrix, to avoid logical redundancy in the path equations. Such logical redundancy, if present, would complicate the dynamic problem considerably.

Reference to Figure B-5 reveals the fact that the sequence of stimuli under consideration calls for a total of 21* relay path retrievals - hence 21 reductions of 18 x 18 arrays to 2 x 2 arrays. Not including the problem of how to identify the nodes between which matrix reduction should take place, or the mechanical step of rearranging rows and columns to get the subject nodes in positions 1 and 2, it is estimated that the matrix reductions themselves - assuming use of the nonzero search algorithm - would require 21 x 371 = 7791 instructions.

B4.5 PATH TRACING VIA ADJACENT NODES

In the third method to be considered (the new method), the data base will be organized in tabular form. Each element will occupy a symbolic address, or slot, which contains

*See Addendum B1 of this appendix for derivation of this number.

the following information:

1. Numbers of bounding nodes.
2. Name of element.
3. Logical state of element (1 or 0).
4. Is this slot an OR slot (1 for yes, 0 for no).
5. Logical state of second element (applicable only if item 4 is 1).
6. Is this element a relay coil (1 for yes, 0 for no).
7. Numbers of node pairs containing contacts (applicable only if item 6 is 1).
8. Numbers of all lower bound lower adjacent nodes (LBLAN).
9. Numbers of all lower bound higher adjacent nodes (LBHAN).
10. Numbers of all upper bound lower adjacent nodes (UBLAN).
11. Numbers of all upper bound higher adjacent nodes (UBHAN). (Items 10 and 11 are applicable only if item 6 is 0.)

For example, the information initially associated with the element C that connects nodes 35 and 47 in Figure B-5 would be:

1. 35-47 (C connects nodes 35 and 47).
2. C (this element is named C).
3. 0 (initial logical state of C = 0).
4. 0 (this is not an OR slot).
5. Not applicable (item 4 is not 1).
6. 0 (C is not a relay coil).
7. Not applicable (item 6 is not 1).
8. 25, 11 (nodes 25 and 11 are connected to 35 - C's lower bound - by elements of the network).
9. 99 (node 99 is connected to 35 by a network element).
10. 37 (node 37 is connected to 47 - C's upper bound - by a network element).
11. 99 (node 99 is connected to 47 by a network element).

Similarly, the stored description of relay coil K_C would be as follows:

1. 35-99 (K_C connects nodes 35 and 99).
2. K_C (this element is named K_C).
3. 0 (initial logical state of K_C is 0).
4. 0 (this is not an OR slot).
5. Not applicable (item 4 is not 1).
6. 1 (this element is a relay coil).

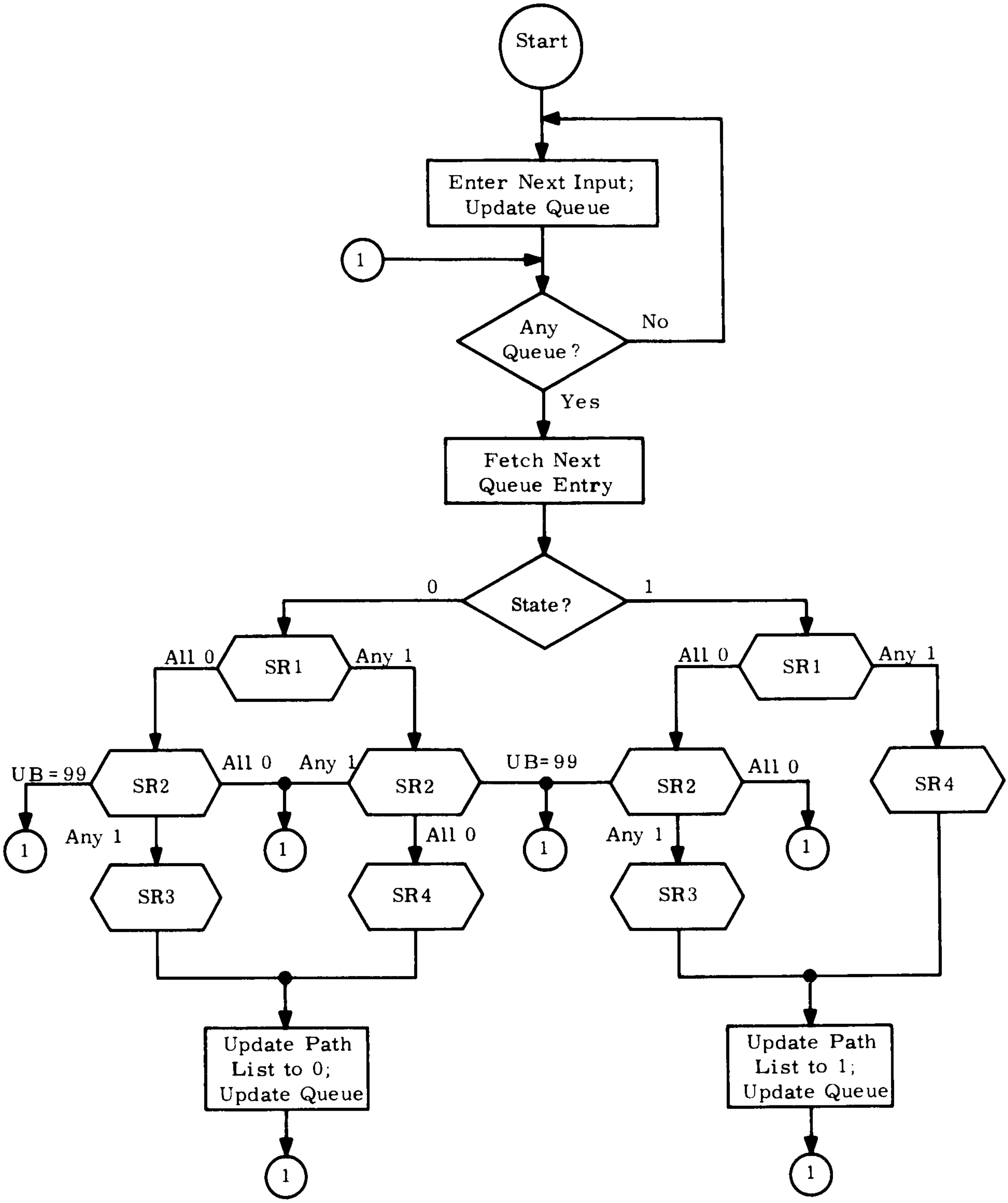


Figure B-10. Over-all Flow Chart for Path Tracing

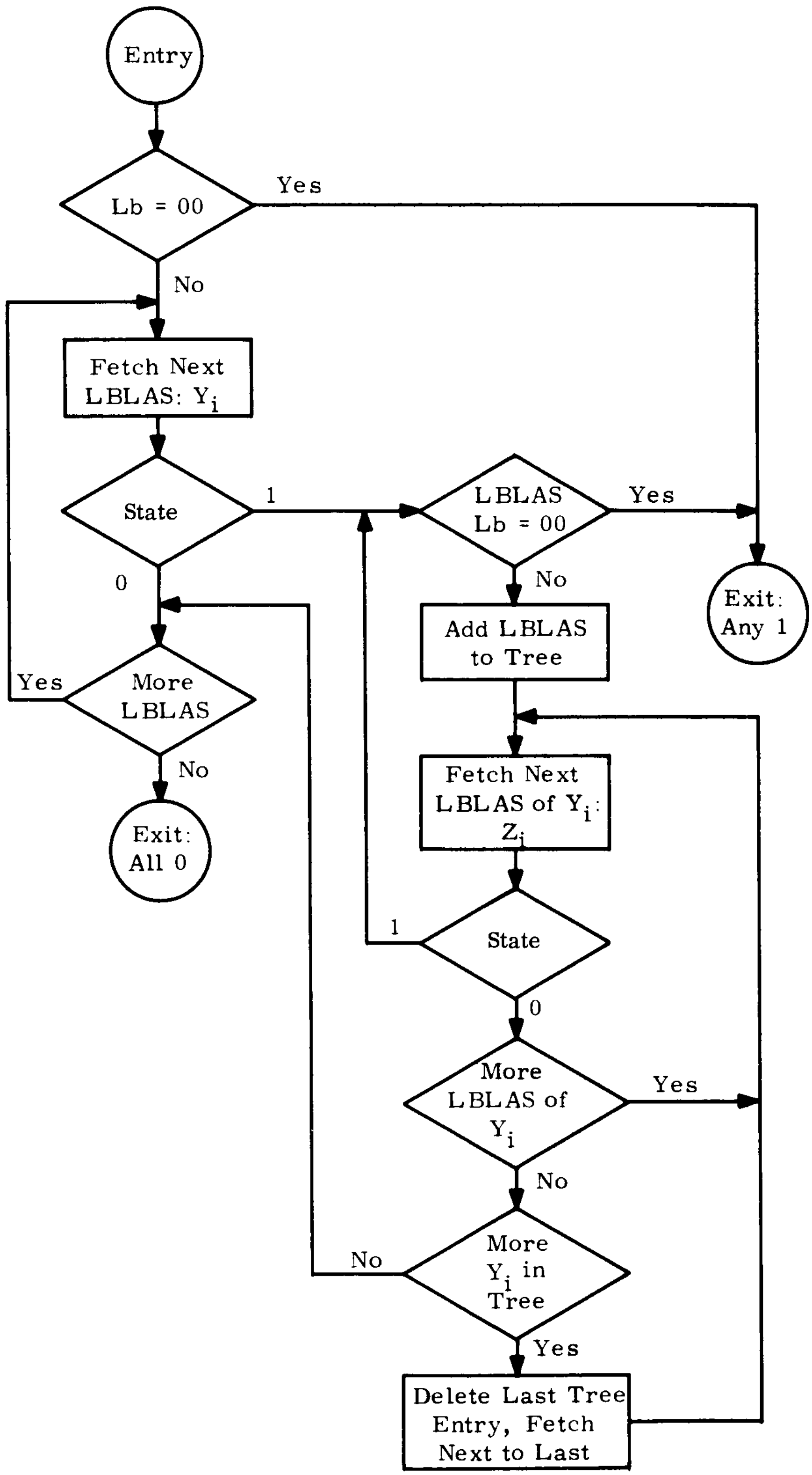


Figure B-11. SR1 Flow Chart

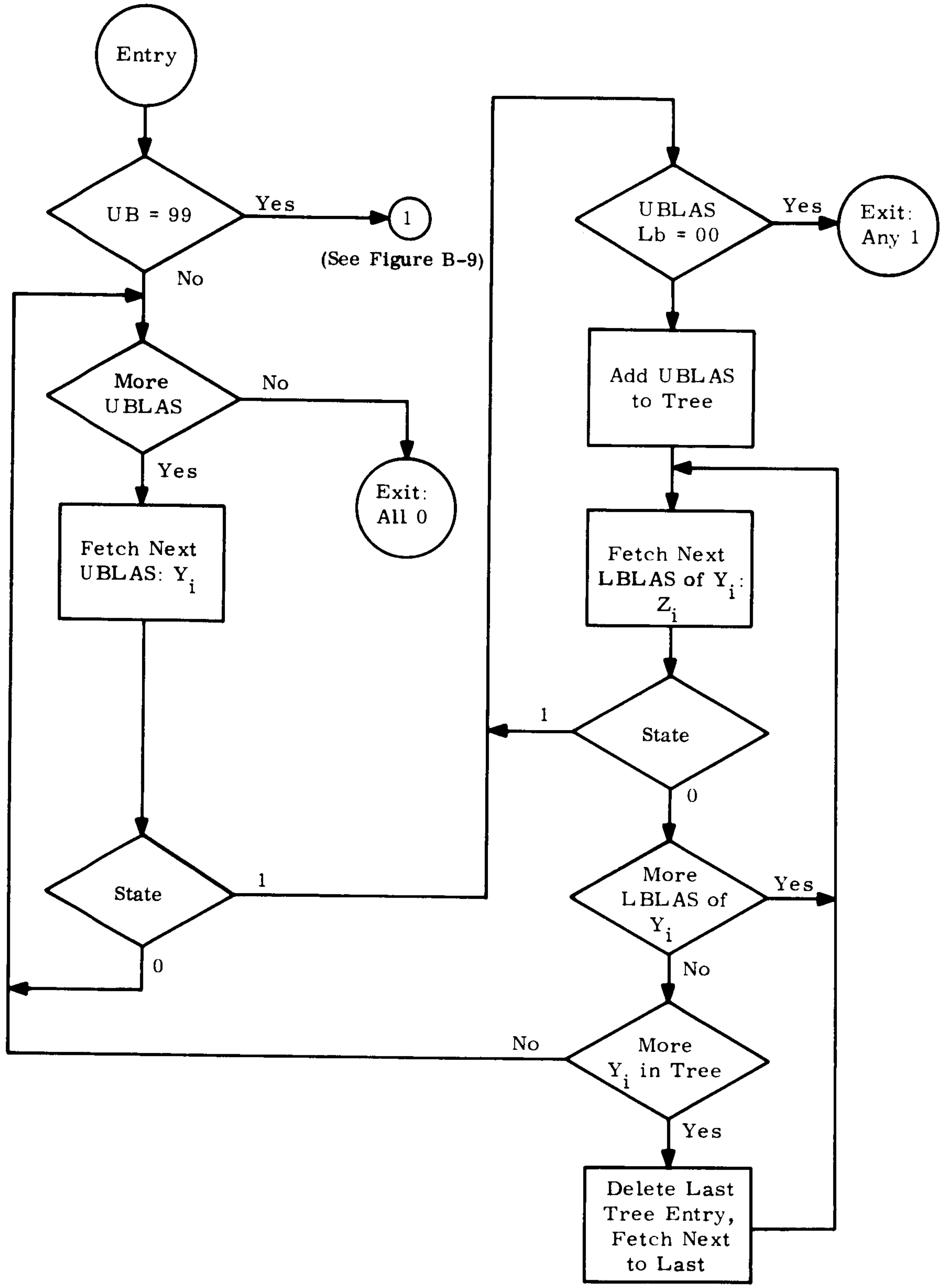


Figure B-12. SR2 Flow Chart

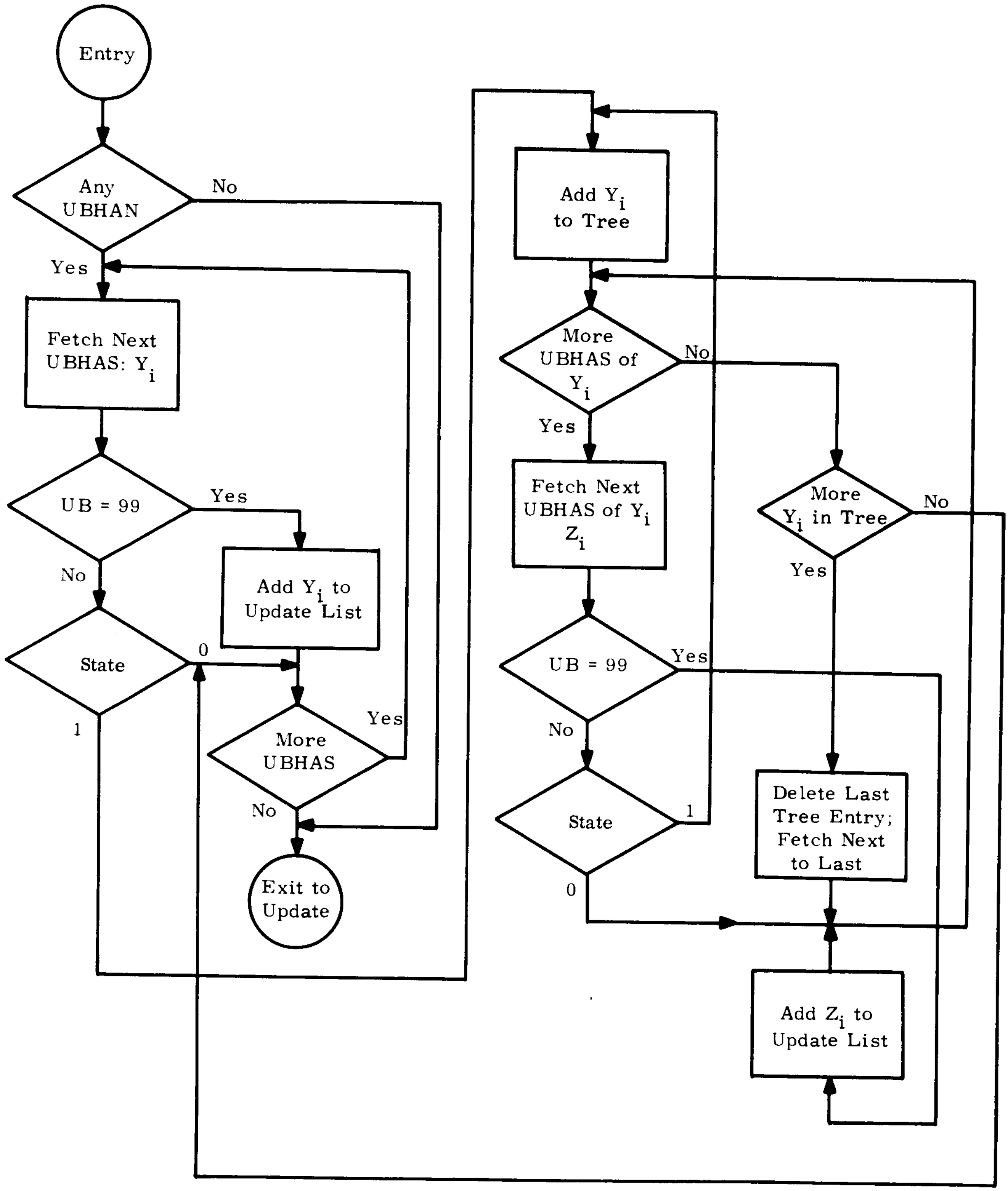


Figure B-13. SR3 Flow Chart

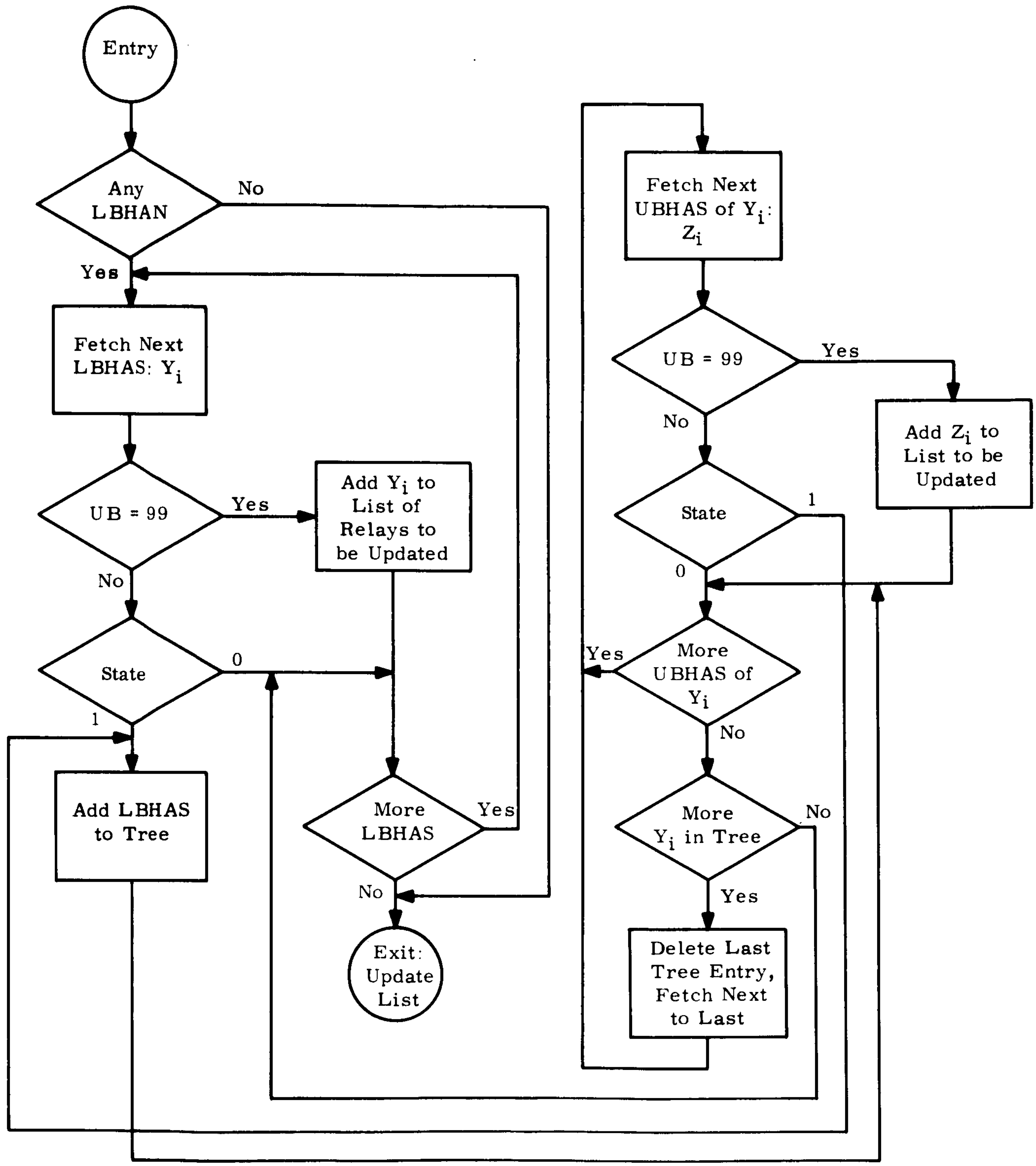


Figure B-14. SR4 Flow Chart

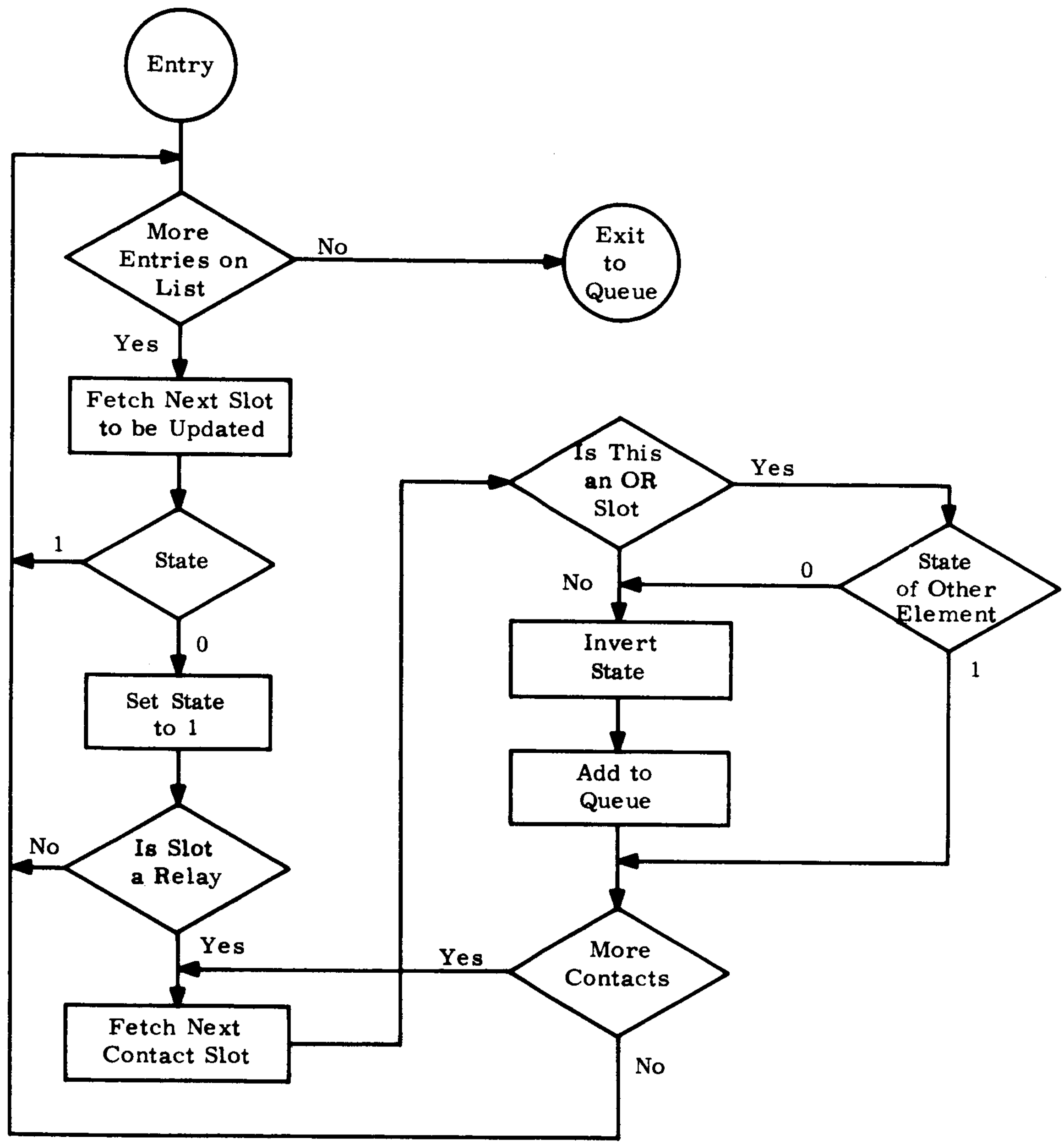


Figure B-15. Update to 1 Flow Chart

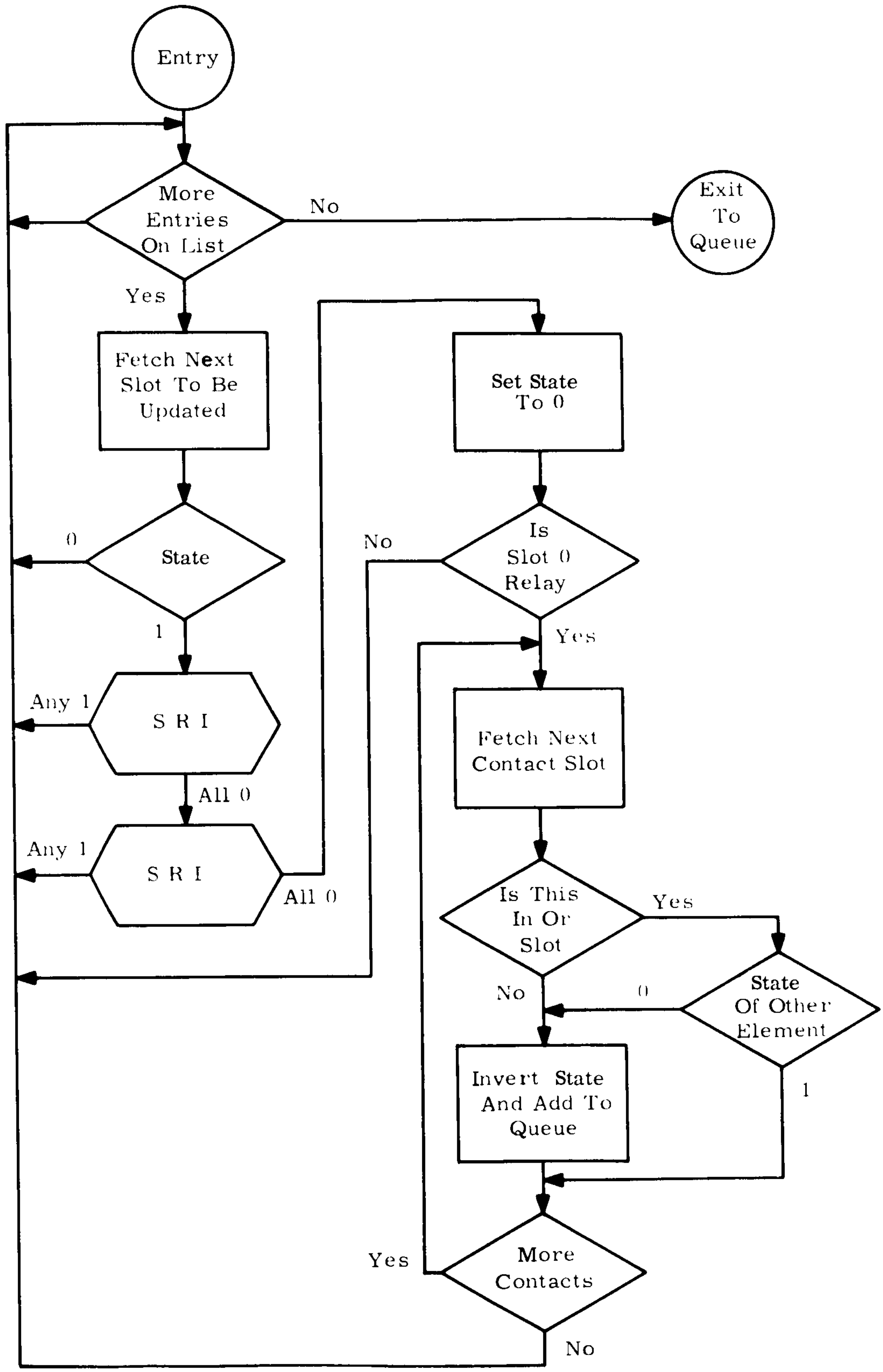


Figure B-16. Update to 0 Flow Chart

7. 35-47, 24-34 (contacts of relay coil K_C will be found connecting nodes 35 and 47, and nodes 24 and 34).
8. 25, 11 (nodes 25 and 11 are connected to 35 - K_C's lower bound - by elements of the network).
9. 47 (node 47 is connected to 35 by a network element).
10. Not applicable (item 6 is not 0).
11. Not applicable (item 6 is not 0).

Table B-2 shows the complete listing for all the elements of Figure B-5. The procedure for path tracing via adjacent nodes then is as follows:

a. Enter input stimulus, including new state and the numbers of all node pairs connected by the stimulus.
b. Form a queue of the node pairs identified by step a.
c. Process first queue entry to determine whether or not the stimulus change caused any relays to change their states. Figure B-10 is a flow chart of the over-all processing algorithm.
d. If relay states are changed by step c., update all slots containing their contacts and add these slots to the queue.
e. The system has stabilized when the queue has no more entries.

The gross flow chart of Figure B-10 is expanded by Figures B-11 through B-16, each of which is a detailed flow chart of one of the subroutines of the Figure B-10 chart. Specifically, these figures consider the following aspects:

Figure B-11: SR1 searches for one closed path from lower bound (LB) to node 00 via LBLAN's. If such a path is found, the slots comprising it are stored in a tree and SR1 exits via its "any 1" output. If there are no closed paths from LB to 00, SR1 exits via its "all 0" output.

Figure B-12: SR2 searches for one closed path from upper bound (UB) to node 00 via UBLAN's. Otherwise identical to SR1.

Figure B-13: SR3 finds all the closed paths from LB to the LB's of slots the UB of which is 99 via LBHAN's, and generates a list of such slots (i.e., relays) for processing by the UPDATE routines. A tree similar to that described under SR1 is developed in the process.

Figure B-14: SR4 finds all closed paths from UB to the LB's of slots whose UB is 99 via UBHAN's. Otherwise identical to SR3.

Figure B-15: UPDATE TO 1 processes the list of relays developed by SR3 or SR4. Relays, their associated contacts, and the queue are all updated if a state change has occurred.

Figure B-16: UPDATE TO 0 processes the same sort of list as UPDATE TO 1. Now, however, an additional test is made for all relays on the list, to be sure there are no alternate paths that would prevent a state change to 0. If said change is indicated, update action is taken as in the UPDATE TO 1 case. (Note that Figure 13 includes an additional subroutine labeled SR1'. This is similar to SR1, except that paths to 00 are sought via LBHAN's. Figure B-14 shows the flow chart.)

See Addendum B2 for a discussion of the development of the algorithm implemented by these flow charts.

Using the flow charts of Figures B-10 through B-17, the number of computer instructions necessary to step the circuit of Figure B-5 through its response to the first four of the input stimuli listed in paragraph B4.2 was estimated. Addendum III contains a sample of the detailed reasoning employed. It was concluded that approximately 413 instructions would be needed to update the simulation from its initial state to the stabilized condition following introduction of the H = 1 stimulus.

B4.6 NUMERICAL COMPARISON OF THE THREE METHODS

The comparison interval covers the complete processing of the first four input stimuli: J = 1, K = 1, X = 1, and H = 1. The results are as follows:

a. Iterated-Equation approach - 10 iterations are required to process the first four stimuli (see Figure B-7). Hence an estimate of 760 computer instructions is obtained.

b. Matrix Reduction method - 15 matrices must be reduced to process the first four stimuli. The estimated computer activity is, therefore, 15 x 371 or 5565 instructions.

c. Path Tracing Via Adjacent Nodes - As indicated in paragraph B4.4, 413 instructions are estimated.

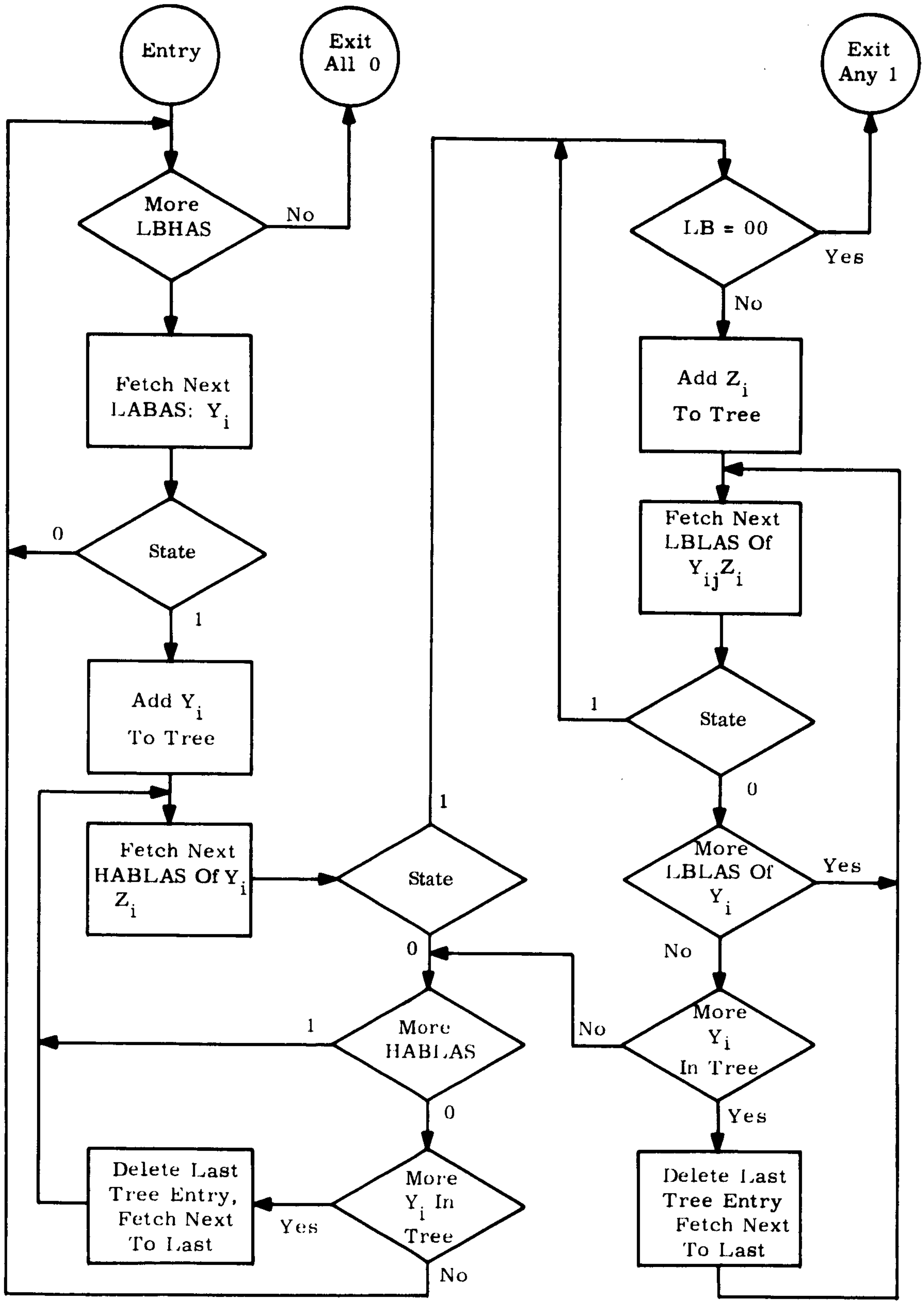


Figure B-17. SRI' Flow Chart

The results favor method c somewhat. The difference between 760 and 413 is probably not too significant, particularly since the comparison is between a proven method and an untried concept. The order of magnitude difference between b and c is more impressive, but it should be borne in mind that in method b no effort has yet been made to simplify the reduction scheme by taking advantage of zero entries in the matrix. However, the point of greatest interest is a consideration of what happens when the comparison circuit is extrapolated to one of much larger size. If it is assumed that Figure B-5 represents a subset of a larger system, but that the logical content of Figure B-5 is complete as shown (i.e., no additional relays or switches), then regardless of system size the 413 figure remains unchanged, whereas both the 760 and the 5565 increase rapidly.

The computer activity required by method a expands directly as the number of equations, since an iteration is an iteration whether it includes eight equations or 8000. The matrix reduction approach expands in greater than direct proportion, since each added node to be eliminated from the matrix at reduction time has more entries than any previous node because of the larger matrix size. This leads to a conclusion that is implicit in Appendix A - namely, that if an attempt should be made to deal with large data bases by matrix techniques, it will be essential to have some way of isolating small portions of the system prior to actual manipulation of the matrices involved. In view of the interwoven nature of the functional system, however, it is not clear yet how the isolation can be achieved. Apparently, the foregoing demonstration shows that even under ideal conditions, specifically, a small system, method b seems to require a good deal more computer activity than method c, although significant differences have not always been found in applications.

B4.7 PROJECTION TO THE DYNAMIC SIMULATION PROBLEM

The outstanding new functional aspect of the Launch Vehicle Component Level Simulation study is its inclusion of dynamic system response, in addition to the discrete analysis that the ESE equations now handle. This section consists of a few remarks about the relative suitability of methods b and c in the dynamic area.

The key discrete operation in the dynamic simulation is the establishment of paths. At any moment, the state of the discrete elements determines which of the dynamic elements are currently responding to the existing stimuli (initial conditions). Thus, it is appropriate to compare Path Tracing Via Adjacent Nodes and Matrix Reduction with this objective in mind.

The matrix reduction method leads directly to the desired path from one node to one other node. Its applicability to a situation in which many output nodes respond dynamically to a single input stimulus is not so clear-cut. The general approach appears to require as many matrix reductions as there are outputs, followed by individual dynamic solutions for each. There is implicit here the problem of how to handle the case in which relay A, responding dynamically to input X, causes a switching to take place in the circuit affecting relay B, which is also responding dynamically to X. Multiple inputs can be handled by superposition or artificially separated in time.

The Path Tracing method can provide logical equations; i.e., the discrete paths in symbolic form, by the same series of steps outlined in paragraph B4.5. The "tree" referred to in the flow charts defines exactly the equation in question, and if that information is explicitly needed to establish the dynamic transfer function, it is available. Every relay switching that occurs during dynamic response can be used to refer directly to the data base for identification of all other (possibly physically remote) functional paths that are affected. Inherently, then, it seems that method (c) offers the advantage of path identification compared to method b.

The problem of dynamic interrelation (i.e., relay A's contact changing relay B's dynamic circuit) is still very much at issue. In fact, therein lies the principal challenge in any dynamic simulation of this type of system. It appears, however, that the path-tracing approach has advantages in the area of path establishment that are not duplicated by conventional matrix techniques without additional features.

B4.8 PROJECTION TO RELATED DISCRETE AREAS

B4.8.1 Failure Effects Analysis, and Fault Location

In the related areas of failure effects analysis and fault location, the suggested approach has been to rely on the former (FEA) to provide a catalog of fault indicator patterns (i.e., lists of discrete indications whose status is not what it would be in a trouble-free system), which can be compared with on-line patterns to produce a considerable degree of fault isolation. This general method is discussed in detail in References 18 through 20. It depends for its effectiveness on the availability of appropriate test points but practical application of the algorithm has been so far content to use the discrete indicators that are available.

If the discrete data base were handled as described in paragraph B4.5, discrete indicators (all of them elements having a UB of 99) could be designated by a single bit in the slot coding and scanned to determine the fault indicator pattern after each discrete stabilization. This could serve either as a generator of patterns resulting from faults intentionally introduced into the simulation, or as a comparison with stored patterns, in on-line applications. If patterns not found in the library are encountered, algorithm of Reference 20 (which is now in the process of being programmed for the ESE simulation) will isolate the trouble to a maintenance unit. There is much still to be done here, particularly in the areas of test-point location and maintenance unit definition, but the feature of interest for this report is that the proposed method for data base handling seems ideally suited to this effort.

B4.8.2 Evaluation of Engineering Changes

The evaluation of engineering changes has much in common with failure effects analysis and fault patterns. The simulator can be used to analyze the effect on the countdown events of changes, which are readily introduced into the data base by storing the new configuration in the slots to be altered. In the iterated equation approach (present ESE), changes require new equation cards and consequent recompilation to update the computer program. With the path-tracing scheme, changes in the data base do not require any changes in the program, which is general and can process whatever is in the slots. Again, it is important to point out that tested and working software are being compared with an untried concept. However, the latter would appear to be somewhat more flexible in this case.

Once changes are proven satisfactory, the simulation data base is permanently updated and records are kept so that it will be possible to determine the point in time at which a change first appeared. No special problems are anticipated in this bookkeeping phase, regardless of data base handling methods.

B4.8.3 Engineering Drawing Printouts

This is essentially a path-tracing operation, hence should be readily compatible with the path-tracing method data base. Node numberings - heretofore discussed as if they were arbitrary - could contain information relative to panels, connector pins, etc., that would be necessary for the printout.

As suggested in Reference 19, such a printout would be a logical output from the on-line fault isolation algorithm.

Table B-2

Data Base for Path Tracing

1	2	3	4	6	7	8		9	10	11	12
Slot	Name	State	OR Slot?	Second Element Status	Relay?	Contact	Slots	LBLAN	LBHAN	UBLAN	UBHAN
00-11	J	0	0	-	0	-		-	12	-	32, 22, 23, 24, 25, 35, 27, 28, 210 212, 213
00-12	X	0	0	-	0	-		-	11	-	
11-32	K	0	0	-	0	-		00	22, 23, 24, 25, 35, 27, 28, 210	22	99
11-22	L	0	0	-	0	-		00	32, 23, 24, 25, 35, 27, 28, 210	-	32
11-23	$\overline{F}$	1	0	-	0	-		00	32, 22, 24, 25, 35, 27, 28, 210	-	34
11-24	B	0	0	-	0	-		00	32, 22, 23, 25, 35, 27, 28, 210	-	34
11-25	K	0	0	-	0	-		00	32, 22, 23, 24, 35, 27, 28, 210	-	35
11-35	M	0	0	-	0	-		00	32, 22, 23, 24, 25, 27, 28, 210	25	99
11-27	P	0	0	-	0	-		00	32, 22, 23, 24, 25, 35, 28, 210	-	37
11-28	A+ϕ	0	1	0	0	-		00	32, 22, 23, 24, 25, 35, 27, 210	-	38
11-210	P+Q	0	1	0	0	-		00	32, 22, 23, 24, 25, 35, 27, 28	-	99
12-212	H	0	0	-	0	-		00	213	-	99
12-213	$\overline{F}$	0	0	-	0	-		00	212	-	99
22-32	$\overline{K}$	1	0	-	0	-		11	-	11	99
23-34	D	0	0	-	0	-		11	-	24	99
24-34	$\overline{C}$	0	0	-	0	-		11	-	23	99
25-35	$\overline{N}$	1	0	-	0	-		11	-	11	99,47
27-37	$\overline{B}$	1	0	-	0	-		11	-	-	47
28-38	G	0	0	-	0	-		11	-	-	99
210-99	K_G	0	0	-	1	28-38		11	-	-	-
212-99	K_Q	0	0	-	1	11-210'		12	-	-	-
213-99	K_P	0	0	-	1	11-210	11-27	12	-	-	-
32-99	K_A	0	0	-	1	11-28		11,22	-	-	-
34-99	K_B	0	0	-	1	11-24	27-37	23,24	-	-	-
35-47	C	0	0	-	0	-		25,11	99	37	99
35-99	K_C	0	0	-	1	35-47	24-34	25,11	47	-	-
37-47	D+F	0	1	-	0	-		27	-	35	99
38-99	K_F	0	0	0	1	11-23, 37-47'	12-213	18	-	-	-
47-99	K_D	0	0	-	1	23-34	37-47	35,37	-	-	-

ADDENDUM B1

Estimation of number of computer instructions required to reduce an 18 x 18 matrix to 2 x 2 size by successive elimination of nodes (i.e., columns and rows).

BRUTE-FORCE APPROACH

Figure B-8 shows the matrix to be reduced. If no cognizance is taken of the fact that the array is mostly vacuous, the reduction will proceed by applying the Boolean relationship:

$$E_{jk} = C_{jk} + C_{ji}C_{ik}, \tag{B-15}$$

to each pair of entries in the column corresponding to the node to be eliminated. Thus, to eliminate the 18th column will require

$${}_{17}C_2 = 136^*, \tag{B-16}$$

applications of Equation B-15.

Continuing, elimination of column 17 requires ${}_{16}C_2 = 120$. Cumulative elimination of columns 18 through 3 thus results in

$$\sum_{n=17}^{3} {}_nC_2 = 816$$

applications of Equation B-15. Since each application requires five instructions, a total of 5 x 816 = 4080 is indicated.

SIMPLIFIED APPROACH

This approach takes advantage of the vacuous nature of the matrix of Figure B-8. The procedure will be to search the column corresponding to the node being eliminated and form a list of all non-zero entries. Then the items on the list will be compared two at a time, and the appropriate entries in the adjacent column will be updated.

---

* ${}_{17}C_2$ stands for "number of combinations of 17 things taken 2 at a time."

For example, in Figure B-8 FETCH and TEST each of the 17 off-diagonal entries in column 38. Since only one is non-zero, no updates result from elimination of node 38, and the required number of computer instructions is 34 instead of $5\,{}_{17}C_2 = 680$!

To eliminate node 37, generate a list of two non-zero nodes. Parenthetically, note that it will be necessary on the list to identify which entries are non-zero, not just "how many." Ignoring this practical problem, there will be 32 testing instructions, two "add to list" instructions, and one application of Equation B-15 - a total of 39 instructions to eliminate node 37.

Table B-3 shows the estimated instruction count for each node elimination. Summation of these figures gives a total of 371 to reduce the matrix to 2 x 2. While it is recognized that different reductions will call for matrix rearrangements that will vary this total somewhat, it will be regarded as typical in calculations involving this method.

Table B-3

Estimated Instruction Count for Each Node Elimination

Node to be Eliminated	"Test for 0" Instructions	"Add to List" Instructions	Reduction Algorithm	Total for This Node
38	34	-	-	34
37	32	2	5	39
35	30	3	15	48
34	28	2	5	35
32	26	2	5	33
213	29	-	-	29
212	22	-	-	22
210	20	-	-	20
28	18	-	-	18
27	16	2	5	23
25	14	2	5	21
24	12	2	5	19
23	10	-	-	10
22	8	-	-	8
12	6	-	-	6
11	4	2	5	11
				371

Reference to Figure B-5 shows that at least 21 matrices would have to be reduced, because:

J = 1 affects paths from 00 to K_A, K_B, K_C, K_D, K_F, and K_G

K = 1 affects paths from 00 to K_A, K_C, K_F, K_B, and K_D

X = 1 affects paths from 00 to K_Q and K_P

H = 1 affects paths from 00 to K_Q and K_G

N = 1 affects paths from 00 to K_C, K_D, and K_B

M = 1 affects paths from 00 to K_C, K_D, and K_B

This count of 21 continues to assume some means of identifying paths (i.e., identifying the nodes between which paths are desired). This optimistic attitude accounts for the "at-least" phrase.

ADDENDUM B2

PATH-TRACING ALGORITHM

To develop an algorithm for path-tracing, a general network model was employed. It is shown in Figure B-18.

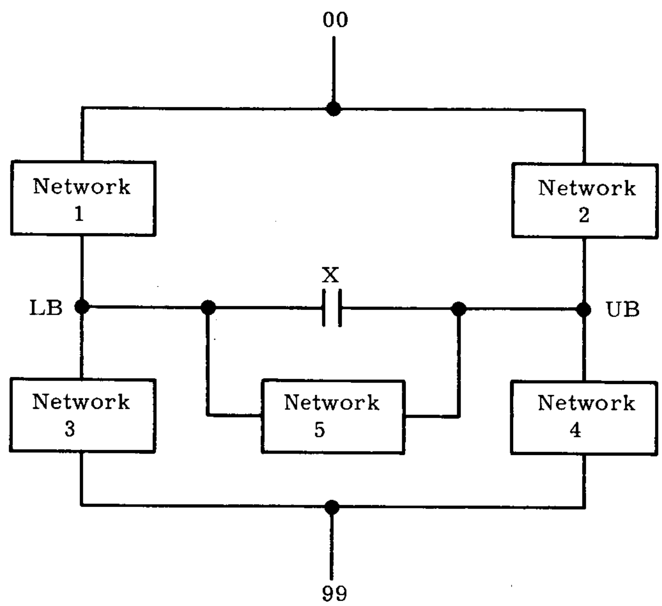


Figure B-18. Model for Algorithm

X is the slot being processed (because its state changed). It has lower bound LB and upper UB. Networks 1, 2, and 5 contain contacts and switches. Networks 3 and 4 contain contacts, switches, and relays (or lamps, pens, etc.). The problem is to locate all complete paths from 00 to slots having 99 as upper bound that include X. Note that whether or not the state of such slots depends on X will be determined by whether X opened or closed (i.e., by the direction of its state change).

Network 5 has the effect of shorting out slot X. That is, if network 5 is closed, state changes of X can have no effect on any relays. It shows up in the algorithm in the following fashion:

> Paths are traced with no special consideration of network 5. Relays on the UPDATE TO 1 list are processed as usual - if they are already 1, no action is necessary. Relays on the UPDATE TO 0 list should not be set to 0 if network 5

is closed, however. Hence all paths to 00 for such relays are checked before the state change is made. (There is probably a more efficient way to do this. Further investigation of the algorithm is indicated.)

The outline of the entire algorithm may now be given. It is as follows:

a. FETCH next queue entry for processing (X).

b. Identify state of X.

c. In all cases, enter SR1 to search for any closed path from LB to 00 via LBLAS (i.e., search network 1). If closed slots are found, form a tree in temporary storage. Each time a dead end is reached, return to most recent branch point. Exit from SR1 with one or the other of the following possibilities:

   (1) All paths 0 from LB to 00

   (2) A 1 path was found from LB to 00

d. If state of X is 1 (meaning that it just changed from 0 to 1) and SR1 exits on case b ("any 1") enter SR4 to search for all closed paths in network 4 from UB via UBHAN to nodes whose UB is 99 (such nodes will be called "relay nodes" hereafter). If such paths are found, add the slots involved to the UPDATE to 1 list. (Note that network 3 need not be searched, since network 1 is closed. Any paths now closed in network 3 were closed before.)

e. If state of X is 1 and SR1 exits on case a ("all 0"), enter SR2 to search for a closed path from UB to 00 via UBLAN (i.e., search network 2). If all are 0, exit to queue. If any is 1, exit to SR3 to search for closed paths in network 3 from LB to "relay nodes" via LBHAN. If any are found, add the corresponding slots to the UPDATE TO 1 list.

f. If state of X is 0, enter SR2 for all cases. If SR1 exited "any 1" and SR2 exists "all 0," enter SR4 to search for closed paths to "relay nodes" in network 4. If any are found, add them to the UPDATE TO 0 list.

   If SR1 exited "all 0," and SR2 also finds "all 0," exit to queue. If SR2 exits on "any 1," enter SR3 to search for closed paths to "relay nodes" in network 3. If any are found, add them to the UPDATE TO 0 list.

g. In the UPDATE TO 1 subroutine, we need merely set the state of all relays on the list to 1. If any were previously 0, their contact slots must be added to the queue and reversed in state.

h. In the UPDATE TO 0 subroutine, it will be necessary to check each relay so designated to be sure there are no closed paths to its LB via network 5.

This is done by checking all paths from LB of the relay to 00. If any one is closed, the relay should not be set to 0. If all are open, the indicated updatings of relay, contacts and queue are made.

The flow charts of Figures B-10 through B-17 implement the foregoing logic in detail. It is felt that they cover all possible configurations without ambiguity. If they do have a loophole or loopholes repair should be possible once the problem is recognized. The basic concept appears to be sound.

ADDENDUM B3

Estimation of number of computer steps required to cause the circuit of Figure B-5 to respond to the input stimulus sequence of paragraph B4.2.

This estimation was made by following the flow charts of Figures B-10 through B-17, as appropriate, to identify each step in the logic of the operation. Following is a step-by-step analysis of the portion starting with the introduction of the stimulus H = 1. The system had previously stabilized (as detected by absence of queue entries) following processing of the X = 1 input.

Operation	Comment	Estimated No. of Instructions
Set H = 1	An input operation	1
Add 12-212 to queue	It affects only slot 12-212	1
Fetch 12-212		1
State = 1	go to SR1	1
(SR1) LB ≠ 00	LB of subject slot is 12	1
Fetch 00-12	00-12 = 1st LBLAS (Y_1)	1
State = 1	Y_1 = 1 (since X = 1)	1
LB = 00	Exit via "any 1" to SR4	1
(SR4) There are UBHAN	12-212 has one or more UBHAN	1
Fetch 212-99	212-99 = 1st UBHAS (Y_1)	1
UB = 99	Y_1 is a relay (K_Q)	1
Add 212-99 to UPDATE TO 1	K_Q should be set to 1 if it isn't already	1
No more UBHAN	Exit to UPDATE TO 1	1
(UPDATE TO 1) there are more entries	K_Q = 212-99 is on the list	1
Fetch 212-99		1

Operation	Comment	Estimated No. of Instructions
State = 0	K_Q was previously de-energized	1
Set state = 1	K_Q is now energized	1
212-99 is a relay	$\therefore$ contacts must be updated	1
Fetch 11-210'	First contact slot	1
Set 11-210' = 1	Update contact status	1
Add 11-210' to queue	Update queue	1
No more contacts	$\therefore$ exit to queue	1

So this little piece of the discrete response - application of input H = 1 and the resulting pickup of relay K_Q - requires a total of 22 computer instructions. In like manner, it is concluded that the complete response of the circuit of Figure B-5 to the first 4 input stimuli (J = K = X = H = 1) requires 413 instructions.

APPENDIX C

CONFIGURATION MANAGEMENT REQUIREMENTS FOR LAUNCH VEHICLE HARDWARE AND SIMULATION SYSTEM

C1 PURPOSE

The purpose of this section is to describe the operating procedures by which simulation system configuration control will be exercised and how the simulation and accounting system can be used by NASA in the execution of its vehicle hardware configuration control.

The measures outlined here are in conformity with the intent of References 15 and 16.

C2 SCOPE

The scope of this section is as follows:

a. Definition and Terminology.
b. Vehicle Requirements.
c. Simulation System Requirements.

C3 DEFINITION AND TERMINOLOGY

C3.1 GENERAL

The information presented in this paragraph is taken from Reference 15 and reorganized in order to present a more definitive picture of the requirements for configuration control of the launch vehicle.

The salient features underlying configuration management and configuration control are as follows:

a. Configuration Management and Control.
b. Configuration Status.
c. Configuration Accounting.
d. Configuration Identification.
e. Configuration Index.
f. Configuration Identification Numbers.

g. Contract End Item (CEI).

h. Contract End Item (CEI) Number.

i. Prime Area of Activity.

j. Uniform Specification.

k. The Engineering Change Proposal.

C3.2 CONFIGURATION MANAGEMENT AND CONTROL

This is the formal set of procedural concepts by which a uniform system of configuration identification, control, and accounting is established and maintained for all NASA systems/equipment and components thereof.

Configuration control is the systematic evaluation, coordination, and approval or disapproval of proposed changes.

Configuration identification and accounting is the business system required to document and report on the as-is configuration and changes thereto.

C3.3 CONFIGURATION STATUS

Configuration Status is the official NASA documented indication of the actual configuration of a serially numbered system or equipment at any given time in relation to an approved configuration.

C3.4 CONFIGURATION ACCOUNTING

This is the act of reporting and documenting changes made to systems, equipment, and components subsequent to the establishment of a baseline configuration in order to establish a configuration status.

C3.5 CONFIGURATION IDENTIFICATION

Configuration Identification is the technical documentation defining the approved configuration under:

a. Design.

b. Development.

c. Test.

C3.6 CONFIGURATION INDEX

This index is a document prepared initially during the design and development period and continued through the acquisition phase. The document is arranged in tabular form and has provisions for inclusion of all changes which result from contractor or program office action. The identification index includes:

a. Section I - End Item of Approved Configuration.
b. Section II - Approved ECP (Change) End Item Index.
c. Section III - End Item Quantitative Requirements Schedule.
d. Section IV - System Configuration Status Accounting - End Item Modification Status.
e. Section V - System Configuration Status Accounting - Spares Status.

C3.7 CONFIGURATION IDENTIFICATION NUMBERS

Configuration Identification Numbers is the relationship of numbers that individually, or in combination, permit accurate selection of the configuration required to perform a given function. These numbers are:

a. Contract End Item (CEI) numbers.
b. Part Numbers.
c. Change Numbers.
d. Manufacturers Code Identification Numbers.

C3.8 CONTRACT END ITEM (CEI)

CEI is an arbitrary designation for the portions of a system/equipment identification as a result of a formal, functional analysis. It is a functional entity, physically related, and selected for the purpose of system development, procurement, and logistics.

The following criteria are applied in the determination of an end item:

a. Procurable by the Government to a single specification.
b. Identified by a single top drawing that has been prepared in conformance with appropriate military specifications.
c. Identified by a separate and distinct part number and serial number.
d. Physical and functional characteristics will be such that its configuration can be controlled and documented economically regardless of the number of changes approved and/or incorporated therein.
e. The location of the distinct/separate parts of an end item should not be remotely located with respect to each other; i.e., black boxes should

be located in the same space system compartment, same maintenance area, etc.

f. By definition, magnetic tapes and card decks used with checkout equipment are classified as end items and subject to change control.

The significance of the CEI lies in the following criteria:

a. There must be a stated requirement for the item based on a completed functional analysis.

b. There must be a specification that completely and accurately describes the item of equipment.

c. There must be an accurate configuration record that documents the approved baseline and changes to the end item.

d. For every change in end-item configuration, there must be a corresponding change to all related documentation affected by the change so that a one-to-one relationship is maintained between the item and its identifying and supporting documentation.

The configuration management structure recognizes the CEI as the lowest level of formal NASA configuration management and that the summation of all CEI's constitutes the entire system.

C3.9 CONTRACT END ITEM (CEI) NUMBER

The CEI number is a permanent number assigned by the contractor to identify a contract end item.

C3.10 PRIME AREA OF ACTIVITY

The material indicates that the prime area of activity for configuration management and control centers around the contract end items (CEI's) and engineering change proposals (ECP's). The program managers manage the acquisition of an end item by use of the CEI specification and Reference 16. Together, these documents establish the basis for configuration control.

Partial control is implemented at the start of the acquisition phase; i.e., when the contractor is authorized to proceed with detail design and development in accordance with approved requirements in the CEI specification. Full configuration control is implemented at the time of acceptance of the first CEI manufactured to the configuration required for a particular series. Thus, programs for contract end items of equipment and facilities are:

a. Defined by detail specification.
b. Phased for progressive configuration control to the requirements specified therein.
c. Controlled to these requirements by the application of Reference 16.

The conclusions to be drawn from the above observations are:

a. The CEI represents the natural interface between NASA and the contractors.
b. There are at least as many NASA/contractor interfaces as there are CEI's.
c. The logical reference line in the total system is the complete set of CEI's.
d. All investigations into the manufacturer's subassemblies must be done with respect to the proper CEI's as well as the manufacturer's identification system.

C3.11 UNIFORM SPECIFICATION

The complete contract end-item technical description used for production release and configuration management is the uniform specification.

It will include:

a. Referenced military and contractor specifications.
b. Documents.
c. Engineering drawings.
d. Production test requirements.
e. Production tests.

These specifications will result from technical data created in the development program.

The concept of the Uniform Specification Program is based on the fact that the system/equipment is not procured by single identifiable systems, but rather by separate end items of contractor-peculiar items and commercial off-the-shelf items. It is recognized that an end-item specification program must be correlated with system procurement programs and methods. Therefore, a basic action of the Uniform Specification Program is the preparation of contract-end-item detail specifications for each provisional end item of the program. The utilization of the contract end-item detail specification thus derived will be as follows:

a. Determination of over-all systems performance for operational use.
b. Rigid control by the Configuration Control Board.
c. Acceptance of end items by NASA.
d. Complete identification of specifications covering all end items required to support the program.
e. Support reprocurement of identical end items where required, or similar end items if exact duplication of performance is not critical.

Formal acceptance of the end item is accomplished at the First Article Configuration Inspection (FACI) conducted at the contractor's facility. At or before FACI, the complete end-item specification will be approved, which establishes the Product Configuration Baseline.

C3.12 THE ENGINEERING CHANGE PROPOSAL (ECP)

The ECP is the primary document by which a change in system, equipment, and components is recommended. The ECP is submitted by the contractor to the Configuration Control Board along with required:

a. Original data.
b. ECP data.
c. Incorporation data.

The Configuration Control Board must evaluate each proposed change from all aspects:

a. Technical.
b. Interface.
c. Logistics.
d. Schedule.
e. Cost.
f. Technical data.
g. Contractual efforts.

In order to establish the total program impact, it must coordinate, with the appropriate Apollo coordination panel, those changes with inter-center interface impact prior to approval and issuance of a Configuration Control Board Directive.

All decisions will be formalized by issuing a Configuration Control Board Directive. This directive may act on the ECP in any one of the following ways:

a. Approve as written.
b. Disapprove with clearly stated reason.
c. Approve with specific changes clearly stated.
d. Defer decision for investigation with responsibility for resolution assigned to a specific person.
e. Refer decision to higher headquarters if judged to be beyond the authority of the center CCB.

The CCB directive is the responsibility of the Program Manager or his appointed representative. He will formalize his decisions by issuing a Configuration Control Board Directive (CCBD). Each CCB member will formalize his official position relative to the decision of the chairman by indicating either a concurrence or non-concurrence on the CCBD. Backup sheets explaining these positions, where required, will be made a part of the official file. The CCBD will be directive on all NASA organizations.

Obviously, it is not the intent of the CCB to arbitrarily dominate the project but to control the configuration through an evaluation of the potential gains and losses in each change proposal.

The CCBD authorization will constitute the sole authority for the contractor to implement the change.

An affirmative CCBD will establish the requirement for concurrent action with respect to development, production, retrofit requirements, and method of accomplishing changes for systems and equipment, including:

a. Training items.
b. AGE.
c. Peculiar tooling.
d. Spares.
e. Spare parts.

f. Revisions to technical manuals.

g. Engineering and technical data.

h. Software end items.

C4 VEHICLE HARDWARE CONFIGURATION MANAGEMENT REQUIREMENTS

C4.1 GENERAL

The vehicle hardware configuration requirements will be satisfied in two ways by the component level simulation:

a. Assistance in the area of engineering change evaluation of the ECP.

b. Up-to-date design file and configuration status index.

C4.2 ENGINEERING CHANGE EVALUATION

The areas where component level simulation will serve as an engineering evaluation tool are (from Reference 16) as follows:

a. Developmental requirements.

b. Alternative solutions.

c. Safety.

d. Reliability.

e. Service life.

f. Operating and test procedure.

g. Checkout.

h. Performance.

i. Weight, balance, and stability.

j. Training.

k. Training installations.

l. Interface effects.

m. Physical constraints.

n. Operational computer programs.

Although some of the above items are not specifically brought out in the usage algorithms, the evaluation can be deduced through data reduction. For example, evaluation of a change that affects checkout equipment would readily reveal interface effects, developmental requirements, and alternative solutions, if necessary.

C4.3 DESIGN FILE

The design file maintains the as-is configuration data and all change data on a parts list, assembly, and usage record basis. The complete file record forms will be maintained in accordance with Reference 15. These are listed in the following exhibits:

a. Exhibit VII

(1) Specification Change Notice.
(2) Specification Change Log.
(3) Specification Identification Index.
(4) Configuration Chart.

b. Exhibit VIII

(1) Configuration Control Board Directive.
(2) ANA Bulletin 445.

c. Exhibit XI

(1) Drawing Application Block.
(2) System Allocation Drawing.
(3) CEI Name Plate.
(4) Technical Manual Configuration Chart.

d. Exhibit XV

(1) End-Item Approved Configuration Index.
(2) Approved Engineering Change Proposal End-Item Index.
(3) End-Item Quantitative Requirements Schedule.
(4) System Configuration Status Accounting: Modification Status.
(5) System Configuration Status Accounting: Spares Status.

Table C-1 lists the exhibits, forms, and specific data elements that will be maintained in the design file. The five forms listed in Exhibit XV reflect the configuration status and will be updated and submitted to NASA at periodic intervals.

C5 SIMULATION SYSTEM CONFIGURATION CONTROL REQUIREMENTS

Configuration control of the simulation system will have to be carried out at a very high technical level. It is essential that the simulation at all times does in fact meet the objectives of vehicle system mathematical definition and user requirements.

To this end, it is essential that configuration control be imposed to:

a. Limit data base information to that which is of specific interest.

b. Assure that the information in the data base can be converted quickly to the mathematical definitions required for the simulations.

c. Assure that the data base is routinely updated on the basis of design change.

d. Assure a building block approach to facilitate future growth of each simulation objective.

e. Provide for check out of each module of the simulation against specified performance and test requirements.

f. Assure that numerical methods do not introduce computational inaccuracies and instabilities.

g. Assure that changes made after mathematical definition, machine programming, debugging and verification will be limited to those required by design ECPS of the space system and those required by User needs.

Table C-1

Major Data Elements Required for Configuration Management Reporting Systems

Types of Input Data Required	Ex-VII Specification Maintenance Sample Formats				Ex-VIII Preparation of Specification Change Figures		Ex-XI Identification and Acceptance Figures				Ex-XV Accounting Reports Requirements Sections					Remarks
	A	B	C	D	1	2	2	3	4	5	I	II	III	IV	V	
Contractor	x	x	x		x	x			x		x	x	x	x	x	Exhibit-VII
Report Date	x	x				x					x	x	x	x	x	Format
Specification Number	x	x	x	x	x	x			x		x					A-Specification
End Item Number	x				x	x	x	x	x		x		x	x	x	Change Notice
Nomenclature			x		x			x	x		x					B-Specification
Part Number			x		x		x	x	x		x		x	x	x	Change Log
New Part Number											x			x	x	C-Specification
ECP/FCR Number	x	x	x	x	x	x				x	x	x		x	x	Identification
ECP/FCR Title					x	x					x					Index
CAT (Sequence of ECP Incorporation)											x					D-Configuration
Effectivity	x			x	x					x	x					Chart
CCN Number											x					
Interface Directive											x					Exhibit-VIII
CCO/TCTO Number (AGE/SV Compatibility)										x	x					Figure
Spares (Effect of ECP on Spares Yes/No)											x					1-Configuration
Affected End Items												x				Control Board
Serial Number	x						x	x	x				x	x		Directive
Allocation													x			2-ANA-Bulletin
Location								x						x	x	Number-445
Type (of ECP Incorporation)														x	x	
Incorporation Dates														x	x	
KIT Identification Number														x	x	
Internal Control Number														x	x	
TRN (Transaction)														x	x	

Table C-1

Major Data Elements Required for Configuration Management Reporting Systems (Cont.)

Types of Input Data Required	Ex-VII Specification Maintenance Sample Formats				Ex-VIII Preparation of Specification Change Figures		Ex-XI Identification and Acceptance Figures				Ex-XV Accounting Reports Requirements Sections					Remarks
	A	B	C	D	1	2	2	3	4	5	I	II	III	IV	V	
Sp/CHS E. I. Serial Number															x	
Sequence (Total Quantity Sp/Chas. Effect)															x	
Next Assembly							x									Exhibit-XI
Required Per N/A							x									Figure
System Employment and Configuration								x								2-Drawing Application
Specification Reference				x				x								Block
NASA Center (Name)	x				x			x								3-System Allocation
NASA Center (Drawing Number)								x								Drawing
Design Activity Code From H4-1									x							4-CEI Name Plate
Federal Stock Number									x							5-Technical
Manufacturing Contract Number	x								x							Manual Configuration
Manufacturers Code From H4-1									x							Chart
Government Ownership Symbol										x						
T. O. Issue										x						
Category					x											
C. C. B. Directive Number					x											
Page of	x	x			x											
Supersedes (Date) Issue of (Date)	x	x			x											
Supersedes ECP No. , Date					x											
TCTR Type					x											
TCTR Number					x											
Part Number Change (Yes/No)					x											
Spares Affected (Yes/No)					x											
Interface Requirements					x											

Table C-1

Major Data Elements Required for Configuration Management Reporting Systems (Cont.)

Types of Input Data Required	Ex-VII Specification Maintenance Sample Formats				Ex-VIII Preparation of Specification Change Figures		Ex-XI Identification and Acceptance Figures				Ex-XV Accounting Reports Requirements Sections					Remarks
	A	B	C	D	1	2	2	3	4	5	I	II	III	IV	V	
Design Deficiency					x											
ECP Action					x											
Procurement Action Required					x											
Program Managers Signatures					x											
System Designation						x										Exhibit-XV
Contractors Recommended Priority						x										<u>Section</u>
Description of Change						x										I-End Item Approved
Justification for Change						x										Configuration
Development Requirements						x										Index
Alternative Solutions						x										II-Approved
Effect on Operational Employment	x	x				x										Engineering
Effect on Specification Requirements	x	x				x										Change Proposal
Effect on Logistic Support	x	x				x										End Item Index
Effect on Logistic Support Materials	x	x				x										III-End Item Quan-
Other Considerations	x	x				x										titative Requirements
Target Completion Date						x										Schedule
Estimated Total Program Costs						x										IV-System
SCN Number	x	x	x													Configuration
Specification Page Number	x			x												Status Accounting
EID			x													Modification
Detailed E.I. Spec. Completion Date			x													Status
Slippage Date			x													V-System
Approval Date			x													Configuration
Title of Change			x													Status Accounting

Table C-1

Major Data Elements Required for Configuration Management Reporting Systems (Cont.)

Types of Input Data Required	Ex-VII Specification Maintenance Sample Formats				Ex-VIII Preparation of Specification Change Figures		Ex-XI Identification and Acceptance Figures				Ex-XV Accounting Reports Requirements Sections					Remarks
	A	B	C	D	1	2	2	3	4	5	I	II	III	IV	V	
Other Specifications Affected			x													Spares Status
Related SCN			x													
Change P/N			x													

APPENDIX D

SIMULATION SYSTEM DESCRIPTION

D1 INTRODUCTION

Appendix D serves as a basis for description of the CLS system at a functional level. The contents of the functional charts are described below:

a. Figures D-1 through D-4 - Information flow at a functional level. This is a diagram relating the potentially desirable outputs to the required functions and required types of data.

b. Tables D-1 and D-2 - A dictionary relating the functions to their representative codes.

c. Table D-3 - A dictionary relating the potentially desirable outputs to their codes.

d. Table D-4 - A dictionary relating the required data to their codes.

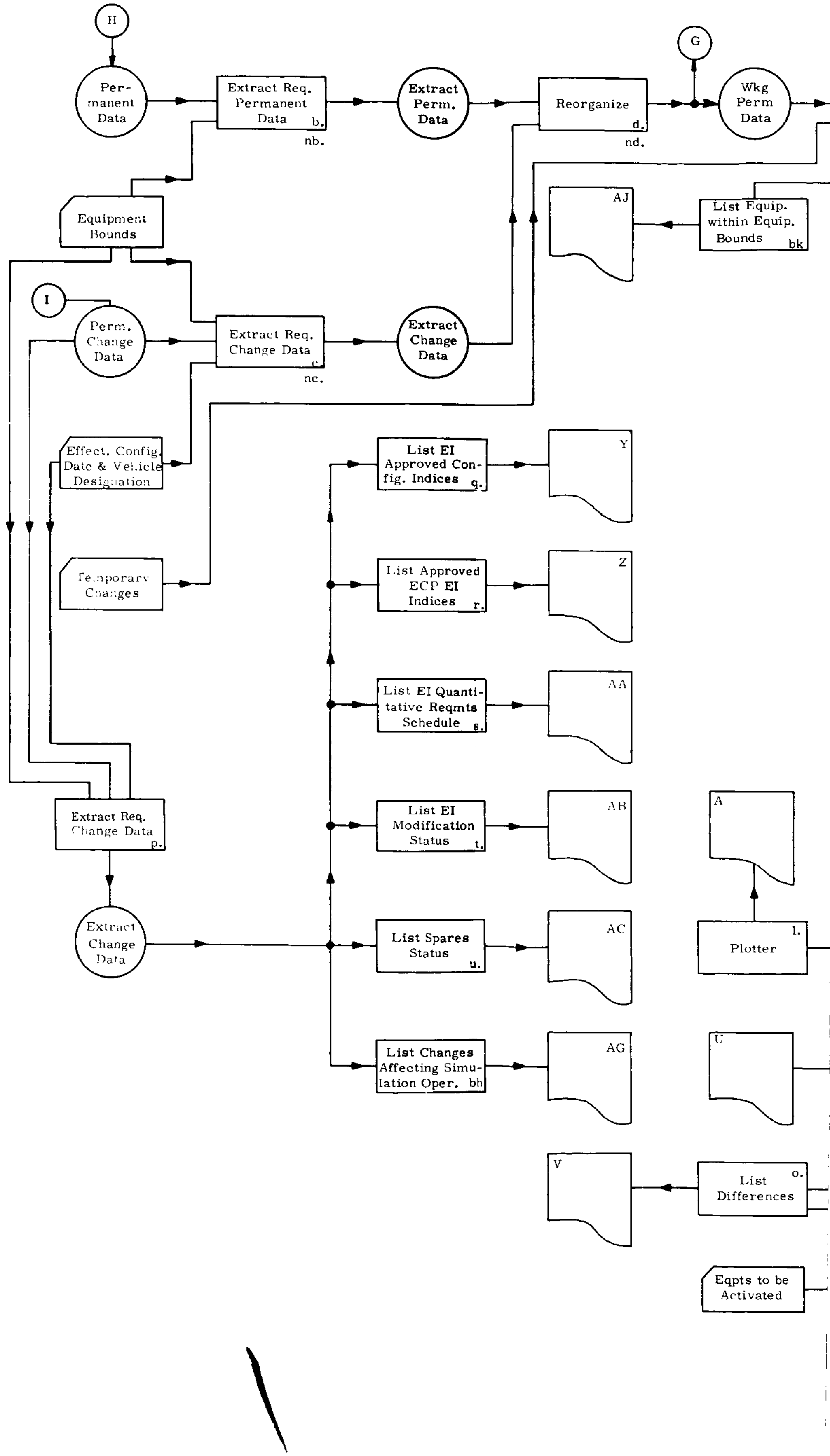
e. Tables D-5 through D-11 - Charts, organized by potential usage, relating required functions to potentially desirable outputs. Section 6 gives an example of applicability of the information on these sheets.

f. Tables D-12 through D-18 - Charts, organized by potential usage, relating required data to potentially desirable outputs. For an example of applicability of the information contained on these sheets, refer to Section 6.

g. Table D-19 - A chart indicating, in coded form, the interrelationship existing between the functions. These codes are identified in the dictionaries mentioned above.

h. Table D-20 - A chart indicating, in coded form, not only the interrelationships existing between the functions but also the recommended order of implementation.

In addition, there follows a set of 20 write-ups, one for each of the potential uses.



H
Permanent Data
Extract Req. Permanent Data b.
nb.
Extract Perm. Data
Reorganize d.
nd.
G
Wkg Perm Data
Equipment Bounds
AJ
List Equip. within Equip. Bounds bk
I
Perm. Change Data
Extract Req. Change Data c.
nc.
Extract Change Data
Effect. Config. Date & Vehicle Designation
List EI Approved Config. Indices q.
Y
Temporary Changes
List Approved ECP EI Indices r.
Z
List EI Quantitative Reqmts Schedule s.
AA
Extract Req. Change Data p.
List EI Modification Status t.
AB
A
Extract Change Data
List Spares Status u.
AC
Plotter l.
List Changes Affecting Simulation Oper. bh
AG
U
V
List Differences o.
Eqpts to be Activated

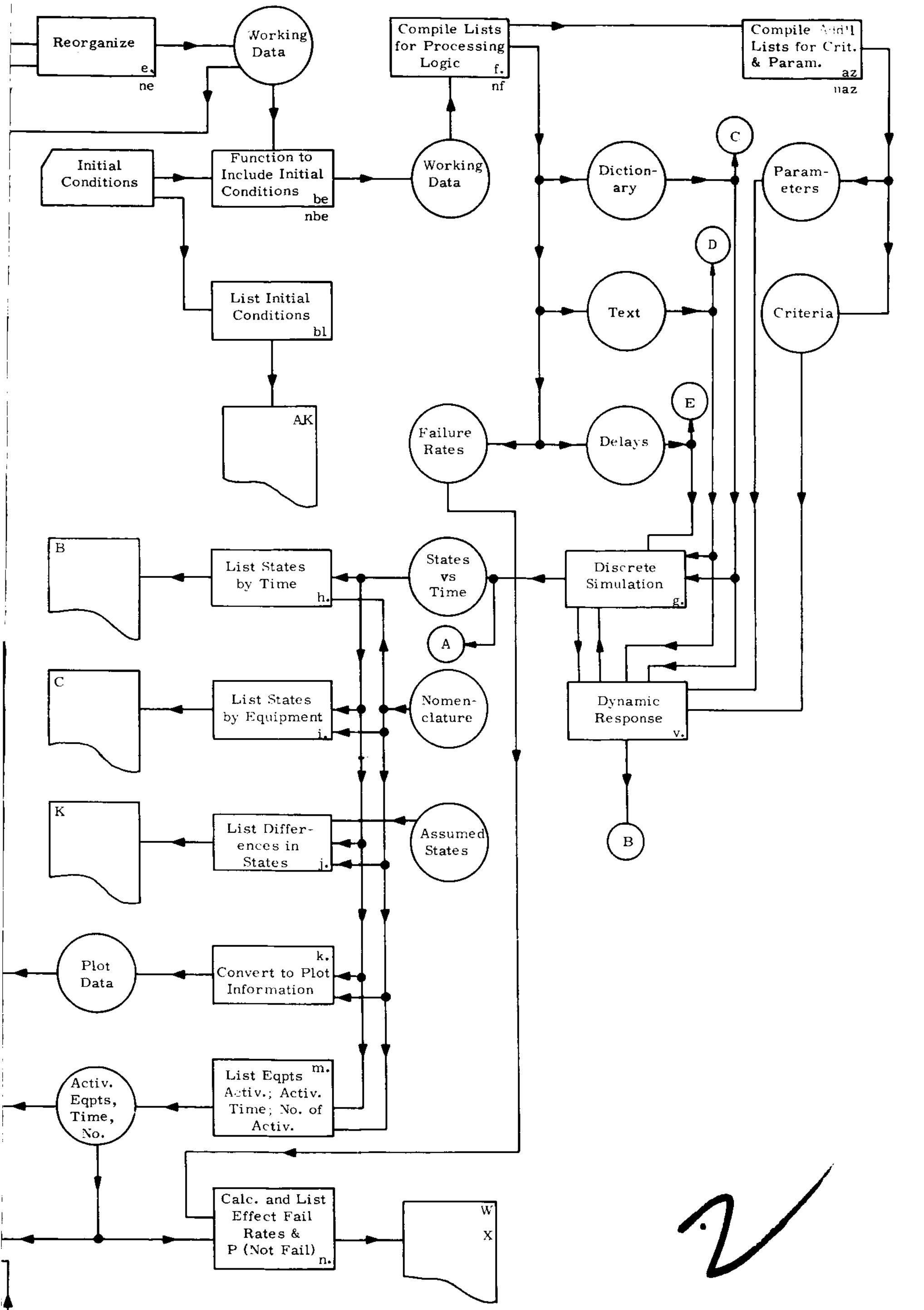
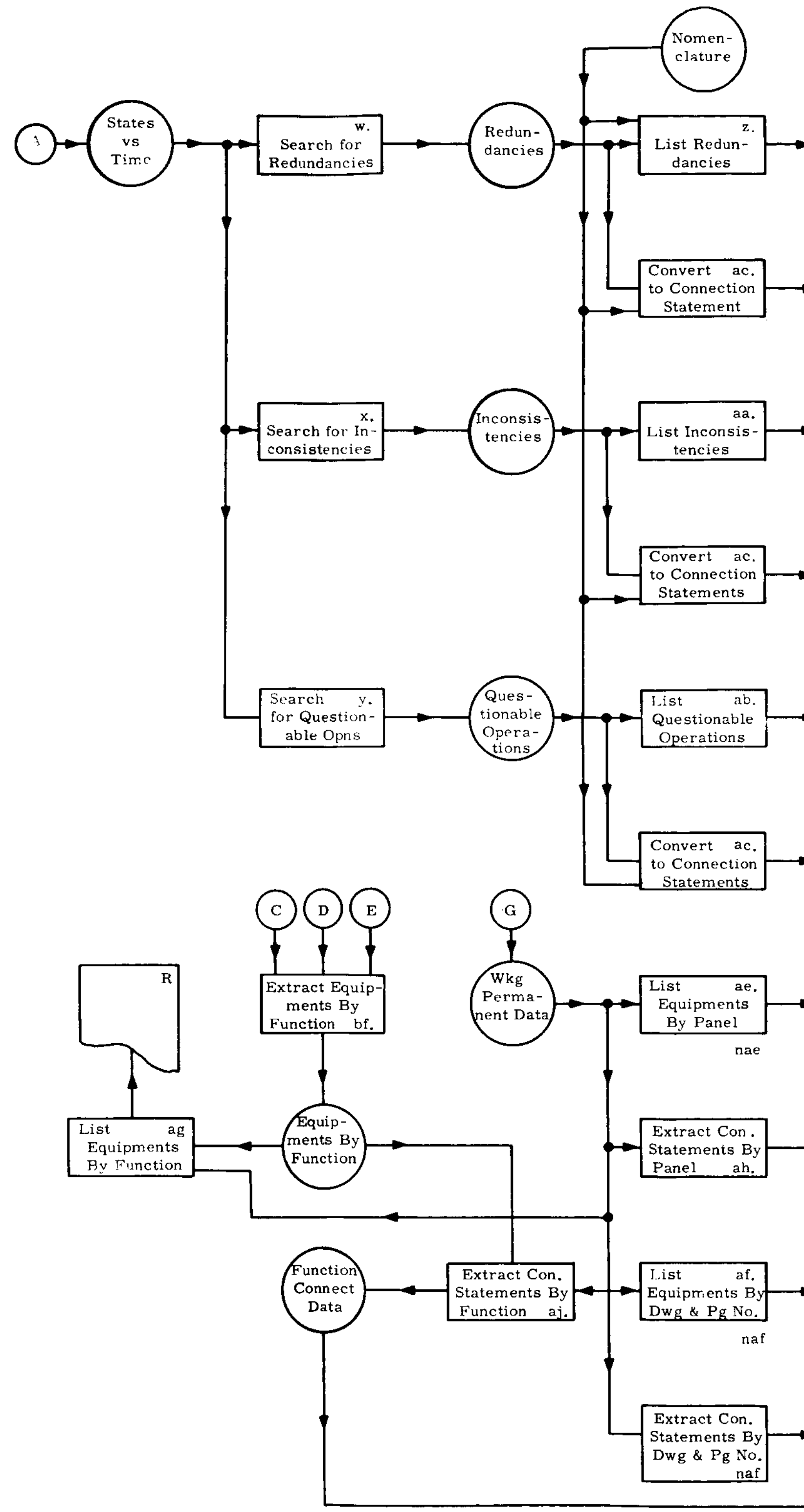


Figure D-1. Information Flow at a Functional Level


States vs Time
w. Search for Redundancies
Redundancies
Nomenclature
z. List Redundancies
Convert ac. to Connection Statement
x. Search for Inconsistencies
Inconsistencies
aa. List Inconsistencies
Convert ac. to Connection Statements
Search y. for Questionable Opns
Questionable Operations
List ab. Questionable Operations
Convert ac. to Connection Statements
C
D
E
G
R
Extract Equipments By Function bf.
Wkg Permanent Data
List ae. Equipments By Panel
nae
List ag Equipments By Function
Equipments By Function
Extract Con. Statements By Panel ah.
Function Connect Data
Extract Con. Statements By Function aj.
List af. Equipments By Dwg & Pg No.
naf
Extract Con. Statements By Dwg & Pg No. naf

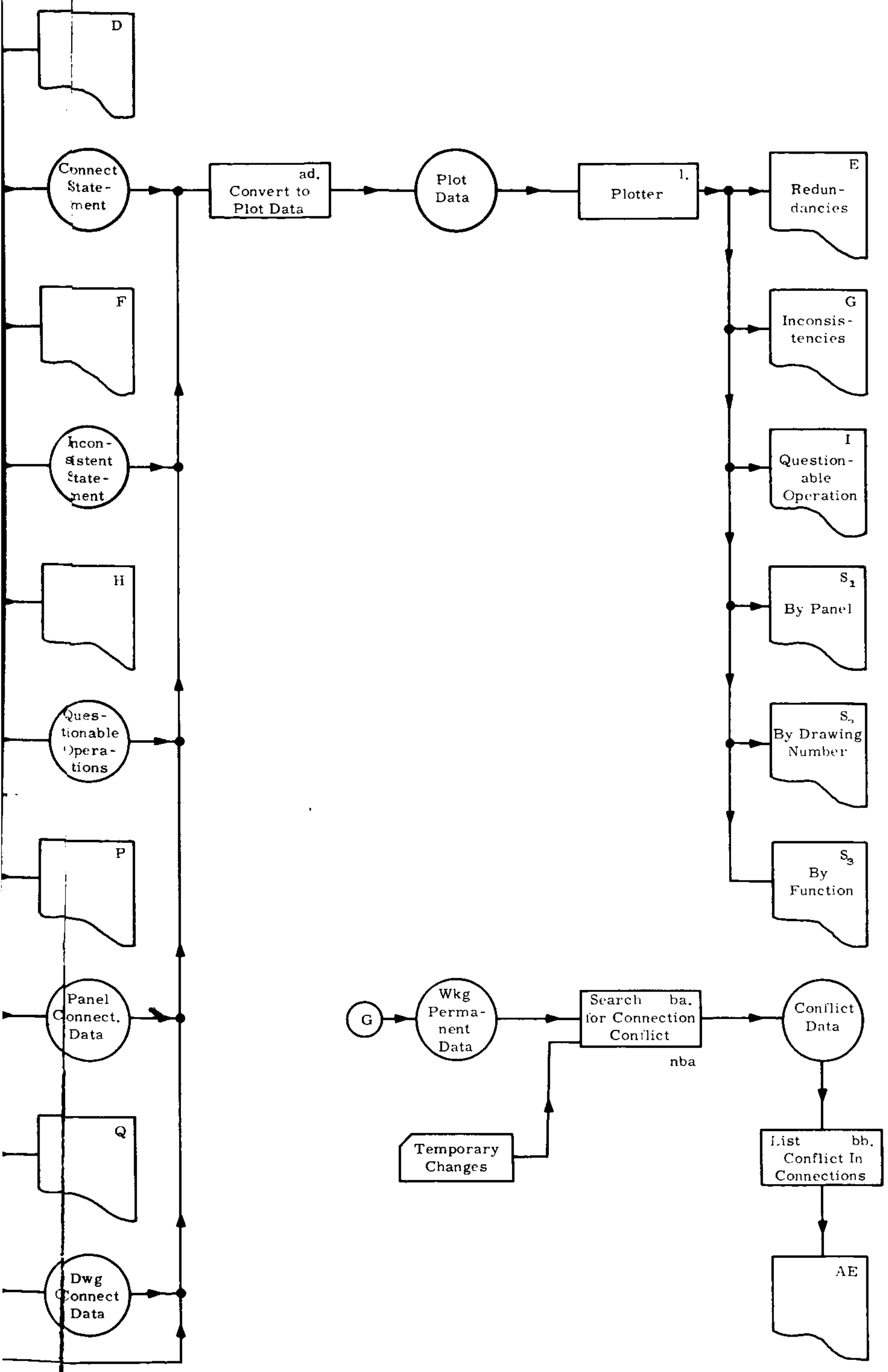
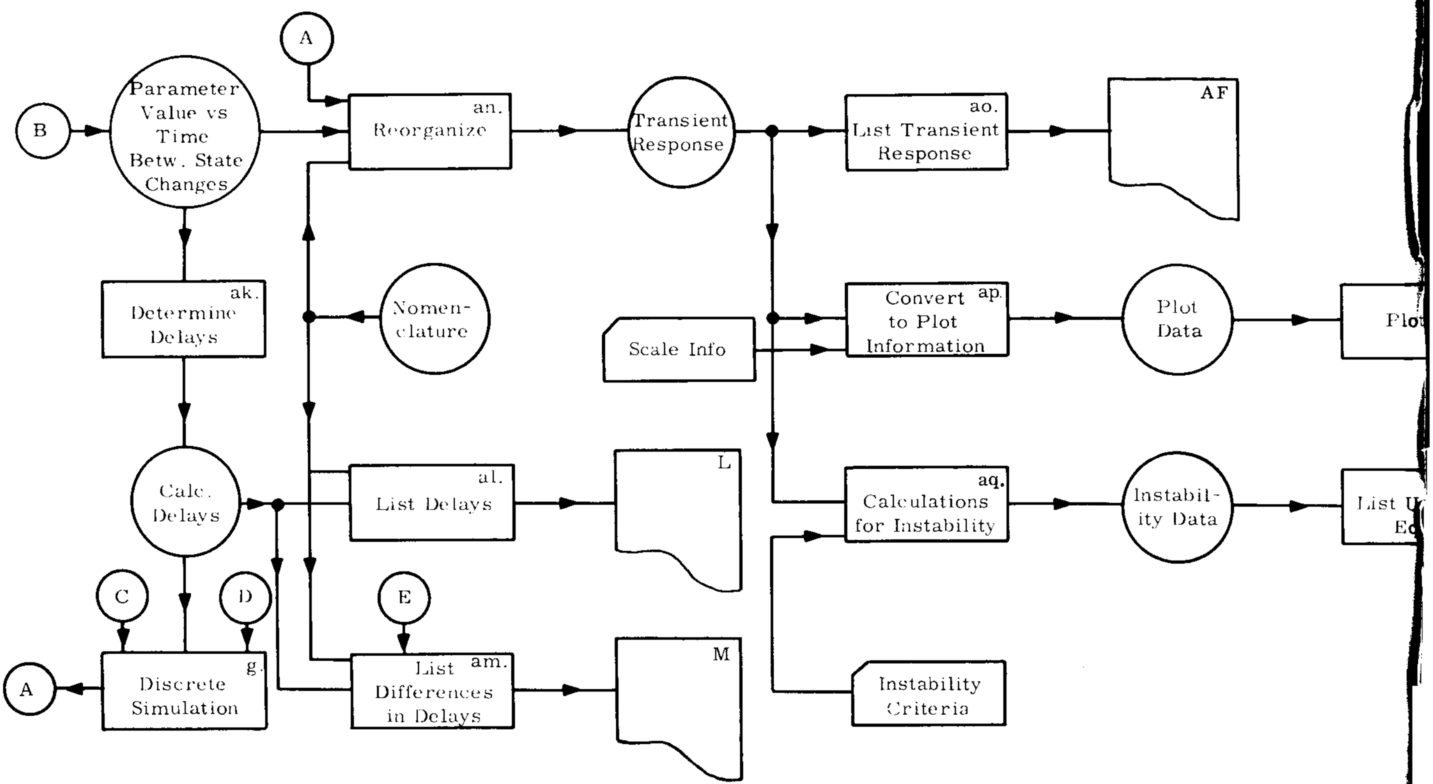


Figure D-2. Information Flow at a Functional Level



A
B
Parameter Value vs Time Betw. State Changes
an.
Reorganize
Transient Response
ao.
List Transient Response
AF
ak.
Determine Delays
Nomen-clature
Scale Info
Convert to Plot Information
ap.
Plot Data
Plot
Calc. Delays
al.
List Delays
L
aq.
Calculations for Instability
Instabil-ity Data
List U
C
D
E
A
g.
Discrete Simulation
List Differences in Delays
am.
M
Instability Criteria

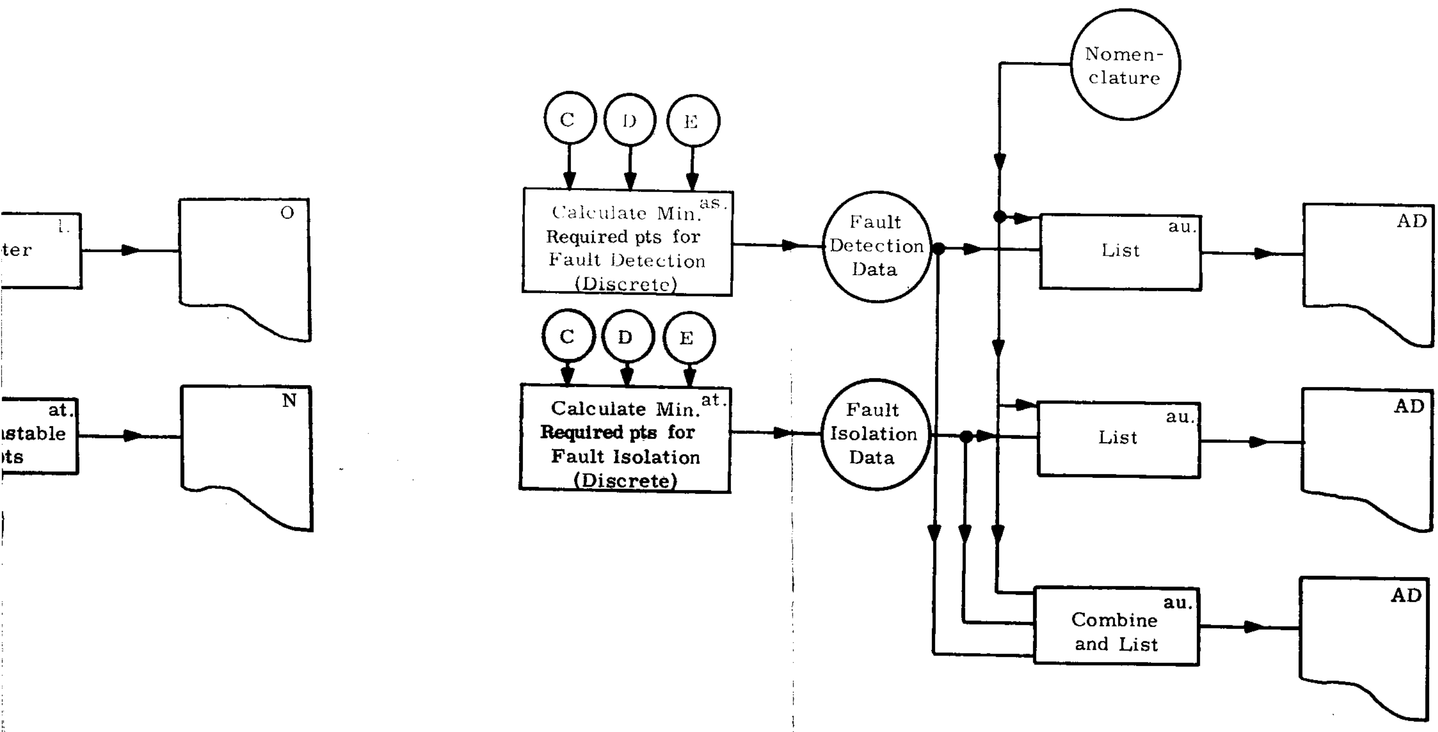
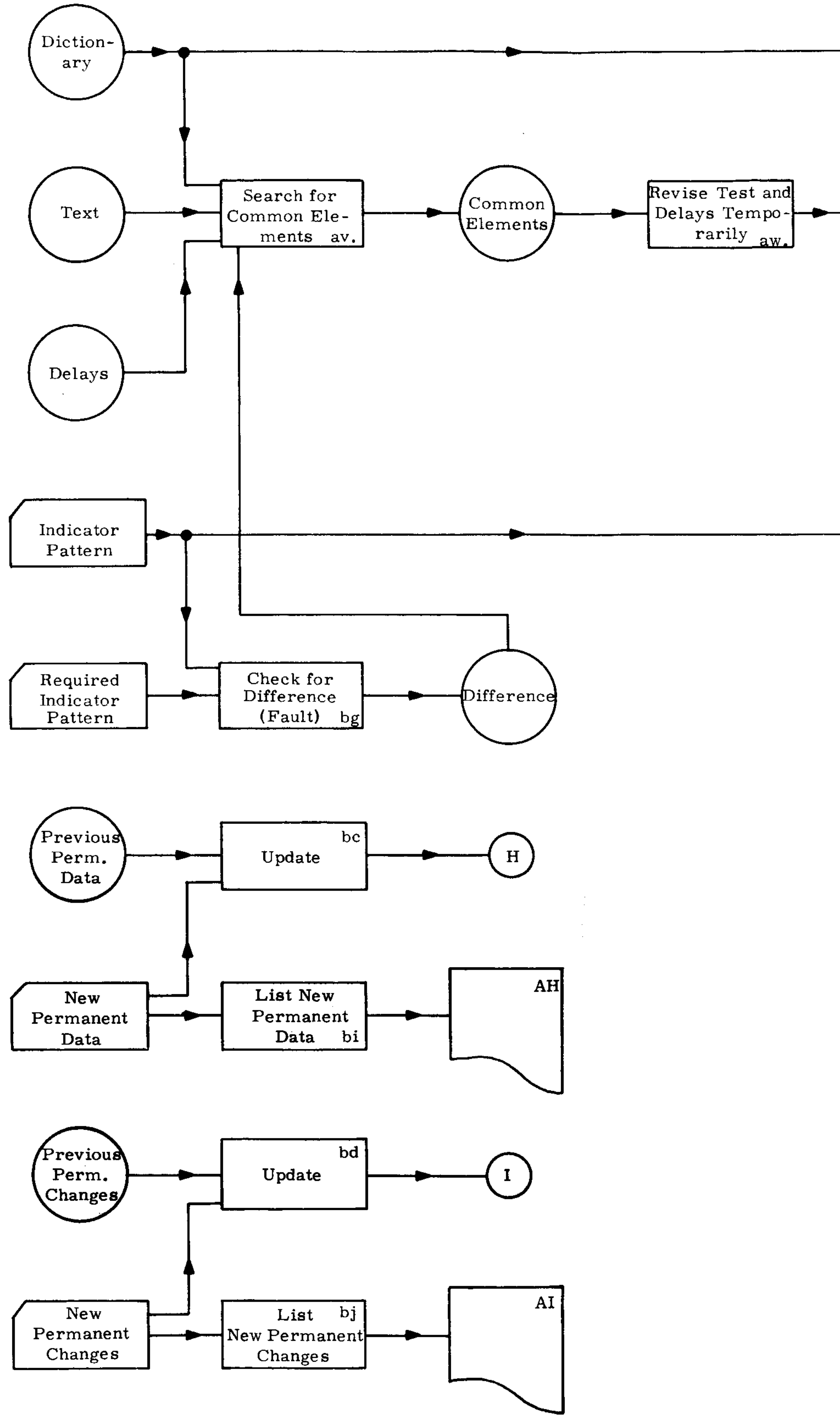
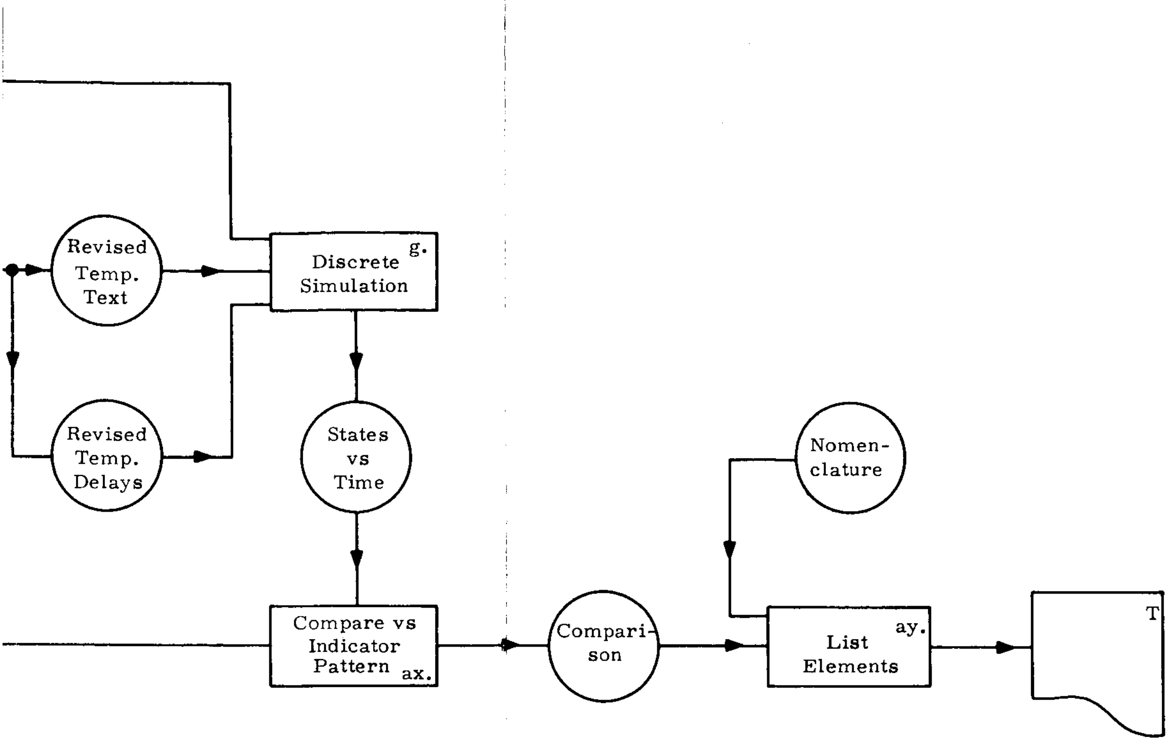


Figure D-3. Information Flow at a Functional Level


Diction-ary
Text
Delays
Search for Common Ele-ments av.
Common Elements
Revise Test and Delays Tempo-rarily aw.
Indicator Pattern
Required Indicator Pattern
Check for Difference (Fault) bg
Difference
Previous Perm. Data
Update bc
H
New Permanent Data
List New Permanent Data bi
AH
Previous Perm. Changes
Update bd
I
New Permanent Changes
List bj New Permanent Changes
AI



V

Figure D-4. Information Flow at a Functional Level

Table D-1

Dictionary Relating Functions to Representative Codes

FUNCTIONS

a. Input Function and Decoder to Define Usage

b. Function to Extract Original Permanent Data to Use

c. Function to Extract Permanent Change Data to Use

d. Function to Reorganize the Extracted Data to Get Permanent Working Data

e. Function to Reorganize Temporary Change Data With Permanent Working Data to Get Working Data

f. Function to Compile Working Data into Lists for Discrete Simulation

g. Discrete Simulation

h. Function to List Equipment States by Time

i. Function to List Equipment States by Equipment

j. Function to List Differences in Equipment States

k. Function to Convert Status Information to Plot Information for Sequence Plot

l. Function to Drive Plotter

m. Function to List Equipments Activated and Times or Number of Activations

n. Function to Calculate Effective Failure Rates and P (Not Failing)

o. Function to List Differences Between Equipments Actually and Assumed to be Activated

p. Function to Extract Change Data for Configuration Management Indices

q. Function to List EI Approved Configuration Indices

r. Function to List Approved ECP EI Indices

s. Function to List EI Quantitative Requirements Schedule

t. Function to List EI Modification Status

u. Function to List Spares Status

v. Function to Calculate Dynamic Response

w. Function to Search for Redundancies

x. Function to Search for Inconsistencies

y. Function to Search for Questionable Operations

z. Function to List Redundancies

aa. Function to List Inconsistencies

ab. Function to List Questionable Operations

ac. Function to Convert Redundancies, Inconsistencies, and Questionable Operations into Connection Statements

ad. Function to Convert Connection Statements into Plot Data

ae. Function to List Equipments by Panel

af. Function to List Equipments by Drawing and Page Number

ag. Function to List Equipments by Function

ah. Function to Extract Connection Statements by Panel

ai. Function to Extract Connection Statements by Drawing and Page Number

aj. Function to Extract Connection Statements by Function

ak. Function to Develop Delay Times

al. Function to List Delay Times

am. Function to Compare and List Differences Between Calculated and Assumed Delay Times

an. Function to Reorganize Discrete Activities and Transient Response

ao. Function to List Transient Response

ap. Function to Convert Transient Response and Scaling Data to Plot Data

aq. Function to Determine Instability Conditions

ar. Function to List Unstable Equipments

as. Function to Determine Least Number of Points Required to Detect Malfunction

at. Function to Determine Least Number of Points Required to Isolate Malfunction

au. Function to List Least Number of Points Required to Detect and/or Isolate Malfunction

av. Function to Search and Identify Common Components Leading to a Given Set of Statuses

aw. Function to Revise Discrete Simulation Data Temporarily

ax. Function to Compare Statuses versus Indicator Pattern

ay. Function to List Elements Possibly Leading to Fault Conditions

az. Function to Compile Parameters and Criteria for Dynamic Response from Working Data

Table D-2

Dictionary Relating Functions to Representative Codes

FUNCTIONS

ba. Function to Search for Conflicts in Connections

bb. Function to List Conflicts in Connections

bc. Function to Update Permanent Data

bd. Function to Update Permanent Changes

be. Function to Include Initial Conditions

bf. Function to Extract Equipments by Function

bg. Function to Check Indicator With Required Indicator Pattern to Determine Differences

bh. Function to List Changes Affecting Simulation Operation

bi. Function to List New Approved Permanent Data Being Entered

bj. Function to List New Approved Permanent Changes Being Entered

bk. Function to List Equipments Involved Within Specified Bounds

bl. Function to List Initial Conditions

NOTE:

If First Letter of Function Code Consisting of 2 or More Letters is n, This Signifies That No Connection Statements are Inputted but Logical Statements are.

Table D-3

Dictionary Relating Outputs to Codes

OUTPUTS

A. Function Sequence Chart

B. Listing of Sequence of Operations by Time

C. Listing of Component Status Change

D. Listing of Redundancies

E. Schematic of System Portion Containing Redundancies

F. Listing of Inconsistencies

G. Schematic of System Portion Containing Inconsistencies

H. Listing of Components Contributing to Questionable Operation

I. Schematic of System Portion Contributing To Questionable Operation

K. Listing For Comparison Run

L. Listing of Delay Titles For Selected System Portion

M. Listing For Comparison of Delay Times

N. Listing of Equipments Unstable in Operation

O. Plot of Transient Response

P. List of Equipments by Panel

Q. List of Equipments by Drawing Numbers

R. Listing of Equipments by Function

S. Schematic of Equipments by Panel, Drawing No., or Function

T. Listing of Equipments and Fault Condition Leading to Fault Indicator Symptoms

U. Listing of Equipments Activated with Time or Number of Activations

V. Listing of Differences of Equipments Expected to be Activated and Those Actually Activated

W. Listing of Effective Failure Rates For Selected System Portions

X. Listing of Probabilities of Not Failing For Selected System Portions

Y. End Item Approved Configuration Indices

Z. Approved ECP End Item Indices

AA End Item Quantitative Requirements Schedule

AB End Item Modification Status

AC Spares Status

AD Listing of Points Required For Detection and Isolation of Faults

AE Listing of Conflicts in Connections

AF Listing of Transient Response

AG Listing of Changes Which Affect Simulation Operation

AH Listing of New Approved Permanent Data Being Entered

AI Listing of New Approved Permanent Changes Being Entered

AJ Listing of Equipments Involved Within Specified Bounds

AK Listing of Initial Conditions

Table D-4

Dictionary Relating Required Data to Codes

PERMANENT DATA		PERMANENT CHANGE DATA		TEMPORARY CHANGE DATA	
1.	Location	1C	Changes In Location	1T	Temporary Changes In Location
2.	Nomenclature	2C	Changes In Nomenclature		
3.	Connection Statements	3C	Changes In Connection Statements	3T	Temporary Changes In Connection Statements
4.	Logical Statements	4C	Changes In Logical Statements	4T	Temporary Changes In Logical Statements
5.	Time Delays	5C	Changes In Time Delays	5T	Temporary Changes In Time Delays
6.	Element Parameters	6C	Changes In Element Parameters	6T	Temporary Changes In Element Parameters
7.	Criteria for Equipment Operations	7C	Changes In Criteria for Equipment Operations	7T	Temporary Changes In Criteria for Equipment Operation
8.	Failure Rates	8C	Changes In Failure Rates	8T	Temporary Changes In Failure Rates
				9T	Equipment Bounds
				10T	Effective Configuration Date and Vehicle Designation
				11T	Initial Conditions
				12T	Output Scale - Abscissa and Ordinate
				13T	Assumed Sequence of Operation
				14T	Indicator Pattern
				15T	Equipments Assumed to be Activated During Run
		16C	End Item Approved Configuration Indices		
		17C	Approved ECP End Item Indices		
		18C	End Item Quantitative Requirements Schedule		
		19C	End Item Modification Status		
		20C	Spares Status		
				21T	Instability Criteria
				22T	Required Indicator Pattern

Table D-5

Potential Usage Relating Functions to Outputs

Potential Use	Outputs	Required Functions: I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
I Define the effect of a proposed change on operation of a selected portion of the launch vehicle and ground support systems.		bc, bd, a, b. c, d	bc, bd, a, b, c, d	nbc, nbd, na, nb, nc, nd	nbc, nbd, na, nb, nc, nd
	P	ae	ae	nae	nae
	Q	af	af	naf	naf
	R	ag, bf, be, f, e	ag, bf, be, f, e	ag, bf, nbe, nf, ne	ag, bf, nbe, nf, ne
	S	ah, ad, 1	ah, ad, 1		
	S	ai, ad, 1	ai, ad, 1		
	S	aj, bf, be, f, ad, 1	aj, bf, be, f, ad, 1		
	AE	ba, bb	ba, bb	nba, bb	nba, bb
		bc, bd, a, b, c, d, e, be, f, az, v, ak, g	bc, bd, a, b, c, d, e, be, f, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g
	B	h	h	h	h
	C	i	i	i	i
	K	j	j	j	j
	A	k, 1	k, 1	k, 1	k, 1
	U	m	m	m	m
	V	m, o	m, o	m, o	m, o
	W, X	m, n	m, n	m, n	m, n
	D	w, z	w, z	w, z	w, z
	E	w, ac, ad, 1	w, ac, ad, 1		
	F	x, aa	x, aa	x, aa	x, aa
	G	x, ac, ad, 1	x, ac, ad, 1		
	H	y, ab	y, ab	y, ab	y, ab
	I	y, ac, ad, 1	y, ac, ad, 1		
	AD	as, au	as, au	as, au	as, au
	AD	at, au	at, au	at, au	at, au
	T	av, aw, ax, ay, bg	av, aw, ax, ay, bg	av, aw, ax, ay, bg	av, aw, ax, ay, bg
	L	al			al
	M	am			am
	N	an, aq, ar			an, aq, ar
	O	an, ap, 1			an, ap, 1
	AF	an, ao			an, ao
II Keep track of approved change orders, drawing changes, and hardware changes made in the simulation data file and the resultant configurations.	AG	a, bc, bd, p, bh	a, bc, bd, p, bh	a, bc, bd, p, bh	a, bc, bd, p, bh

Table D-6

Potential Usage Relating Functions to Outputs

Potential Use		Outputs	Required Functions			
			I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
III	Insert approved changes into central data file.		a	a	na	na
		AH	be, bi	bc, bi	nbc, bi	nbc, bi
		AI	bd, bj	bd, bj	nbd, bj	nbd, bj
IV	Change data temporarily to simulate a fault condition and follow its effect through a selected portion of the system.		bc, bd, a, b, c, d	bc, bd, a, b, c, d	nbc, nbd, na, nb, nc, nd	nbc, nbd, na, nb, nc, nd
		P	ae	ae	nae	nae
		Q	af	af	naf	naf
		R	ag, bf, be, f, e	ag, bf, be, f, e	ag, bf, nbe, bf, ne	ag, bf, nbe, bf, ne
		S	ah, ad, l	ah, ad, l		
		S	ai, ad, l	ai, ad, l		
		S	aj, bf, be, f, ad, l	aj, bf, be, f, ad, l		
		AE	ba, bb	ba, bb	nba, bb	nba, bb
			bc, bd, a, b, c, d, e, be, f, az, v, ak, g	bc, bd, a, b, c, d, e, be, f, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g
		B	h	h	h	h
		C	i	i	i	i
		K	j	j	j	j
		A	k, l	k, l	k, l	k, l
		U	m	m	m	m
		V	m, o	m, o	m, o	m, o
		W, X	m, n	m, n	m, n	m, n
		D	w, z	w, z	w, z	w, z
		E	w, ac, ad, l	w, ac, ad, l		
		F	x, aa	x, aa	x, aa	x, aa
		G	x, ac, ad, l	x, ac, ad, l		
		H	y. ab	y, ab	y, ab	y, ab
		I	y, ac, ad, l	y, ac, ad, l		
		AD	as, au	as, au	as, au	as, au
		AD	at, au	at, au	at, au	at, au
		T	av, aw, ax, ay, bg	av, aw, ax, ay, bg	av, aw, ax, ay, bg	av, aw, ax, ay, bg
		L	al			al
		M	am			am
		N	an, aq. ar			an, aq, ar
		O	an. ap, l			an, ap, l
		AF	an, ao			an, ao

D-12

Table D-7

Potential Usage Relating Functions to Outputs

Potential Use	Outputs	Required Functions: I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
V Calculate expected times for events of the sequential operation of a selected portion of the launch vehicle and ground support systems.		bc, bd, a, b, c, d, e, be, f, az, v, ak, g			nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g
	L	al			al
	M	am			am
VI Perform transient analysis of a selected portion of the launch vehicle and ground support systems.		bc, bd, a, b, c, d, e be, f, az, v, ak, g			nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g
	N	an, aq, ar			an, aq, ar
	O	an, ap, l			an, ap, l
	AF	an, ao			an, ao
VII Follow signals through a selected portion of the launch vehicle on a discrete basis.		bc, bd, a, b, c, d, e be, f, az, v, ak, g	bc, bd, a, b, c, d, e, be, f, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g
	B	h	h	h	h
	C	i	i	i	i
	K	j	j	j	j
	A	k, l	k, l	k, l	k, l
VIII Relate the simulation to the racks, equipment numbers, etc. as given on panel schematics, interconnection diagrams, and advanced system schematics.		bc, bd, a, b, c, d	bc, bd, a, b, c, d	nbc, nbd, na, nb, nc, nd	nbc, nbd, na, nb, nc, nd
	P	ae	ae	nae	nae
	Q	af	af	naf	naf
	R	ag, bf, be, f, e	ag, bf, be, f, e	ag, bf, nbe, nf, ne	ag, bf, nbe, nf, ne
	S	ah, ad, l	ah, ad, l		
	S	ai, ad, l	ai, ad, l		
	S	aj, bf, be, f, ad, l	aj, bf, be, f, ad, l		

Table D-8

Potential Usage Relating Functions to Outputs

Potential Use	Outputs	Required Functions: I Dynamic Simulation. Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
IX Search out closely timed operations and identify the equipments involved to eliminate areas of questionable operation where change plays a significant role in the operation of a system.		bc, bd, a, b, c, d, e, be, f, az, v, ak, g, y	bc, bd, a, b, c, d, e, be, f, g, y	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, g, y	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g, y
	H	ab	ab	ab	ab
	I	ac, ad, l	ac, ad, l		
X Check for inconsistencies such as conflicting signals and component operations which lead to inconsistent functions.		bc, bd, a, b, c, d, e, be, f, az, v, ak, g, x	bc, bd, a, b, c, d, e, be, f, g, x	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, g, x	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g, x
	F	aa	aa	aa	aa
	G	ac, ad, l	ac, ad. l		
XI Check for redundancies to detect unintentional multiple methods of obtaining individual signals or modes of operation and also to verify the presence of intended redundant signals or modes included to improve reliability.		bc, bd, a, b, c, d, e be, f, az, v, ak, g, w	bc, bd, a, b, c, d, e, be, f, g, w	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, g, w	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g, w
	D	z	z	z	z
	E	ac, ad, l	ac, ad. l		
XII Define areas of possible malfunctions given a set of symptoms.		bc, bd, a, b, c, d, e, f	bc, bd, a, b, c, d, e, f	nbc, nbd, na, nb, nc, nd, ne	nbc, nbd, na, nb, nc, nd, ne
	T	az, v, ak, g, ax, ay, be, bg, av, aw	be, g, ax, ay, bg, av, aw	nbe, f, g, ax, ay, bg, av, aw	nbe, nf, naz, v, ak, g, ax, ay, bg, av, aw

D-14

Table D-9

Potential Usage Relating Functions to Outputs

Potential Use	Outputs	Required Functions			
		I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation Logic Statements
XIII Allow a user to set up conditions which identify a portion of a proposed or actual checkout or countdown sequence.		bc, bd, a, b, c, d, e, be	bc, bd, a, b, c, d, e, be	nbc, nbd, na, nb, nc, nd, ne, nbe	nbc, nbd, na, nb, nc, nd, ne, nbe
	AJ	bk	bk	bk	bk
XIV Allow a set of simulated fault conditions to be superimposed on a list of conditions defining a planned checkout or count-down sequence.		bc, bd, a, b, c, d, e, be	bc, bd, a, b, c, d, e, be	nbc, nbd, na, nb, nc, nd, ne, nbe	nbc, nbd, na, nb, nc, nd, ne, nbe
	AK	bl	bl	bl	bl
XV Define and keep track of equipments which have been activated and maintain a record for output.		bc, bd, a, b, c, d, e	bc, bd, a, b, c, d, e	nbc, nbd, na, nb, nc, nd, ne	nbc, nbd, na, nb, nc, nd, ne
	U	be, f, az, v, ak, g, m	be, f, g, m	nbe, nf, g, m	nbe, nf, naz, v, ak, g, m
XVI Define equipments which have not been activated.		bc, bd, a, b, c, d, e, be, f, az, v, ak, g	bc, bd, a, b, c, d, e, be, f, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g
	V	m, o	m, o	m, o	m, o

Table D-10

Potential Usage Relating Functions to Outputs

Potential Use	Outputs	Required Functions			
		I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
XVII Compare resulting sequences with desired ones.		bc, bd, a, b, c, d, e	bc, bd, a, b, c, d, e	nbc, nbd, na, nb, nc, nd, ne	nbc, nbd, na, nb, nc, nd, ne
	K	be, f, az, v, ak, g, j	be, f, g, j	nbe, nf, g, j	nbe, nf, naz, v, ak, g, j
XVIII Determine expected reliability factors for a selected portion of the system.		bc, bd, a, b, c, d, e	bc, bd, a, b, c, d, e	nbc, nbd, na, nb, nc, nd, ne	nbc, nbd, na, nb, nc, nd, ne
	U	be, f, az, v, ak, g, m	be, f, g, m	nbe, nf, g, m	nbe, nf, naz, v, ak, g, m
	V	o	o	o	o
	W, X	n	n	n	n
XIX Configuration management documentation data central and control.		a, bc, bd, p	a, bc, bp, p	a, nbc, nbd, p	a, nbc, nbd, p
	Y	q	q	q	q
	Z	r	r	r	r
	AA	s	s	s	s
	AB	t	t	t	t
	AC	u	u	u	u

Table D-11

Potential Usage Relating Functions to Outputs

Potential Use	Outputs	Required Functions: I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
XX Development of checkout and countdown procedures.		bc, bd, a, b, c, d	bc, bd, a, b, c, d	nbc, nbd, na, nb, nc, nd	nbc, nbd, na, nb, nc, nd
	P	ae	ae	nae	nae
	Q	af	af	naf	naf
	R	ag, bf, be, f, e	ag, bf, be, f, e	ag, bf, nbe, f, ne	ag, bf, nbe, f, ne
	S	ah, ad, l	ah, ad, l		
	S	ai, ad, l	ai, ad, l		
	S	aj, be, bf, f, ad, l	aj, be, bf, f, ad, l		
	AE	ba, bb	ba, bb	nba, bb	nba, bb
		bc, bd, a, b, c, d, e, be, f, az, v, ak, g	bc, bd, a, b, c, d, e, be, f, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, g	nbc, nbd, na, nb, nc, nd, ne, nbe, nf, naz, v, ak, g
	B	h	h	h	h
	C	i	i	i	i
	K	j	j	j	j
	A	k, l	k, l	k, l	k, l
	U	m	m	m	m
	V	m, o	m, o	m, o	m, o
	W, X	m, n	m, n	m, n	m, n
	D	w, z	w, z	w, z	w, z
	E	w, ac, ad, l	w, ac, ad, l		
	F	x, aa	x, aa	x, aa	x, aa
	G	x, ac, ad, l	x, ac, ad, l		
	H	y, ab	y, ab	y, ab	y, ab
	I	y, ac, ad, l	y, ac, ad, l		
	AD	as, au	as, au	as, au	as, au
	AD	at, au	at, au	at, au	at, au
	T	av, aw, ax, ay, bg	av, aw, ax, ay, bg	av, aw, ax, ay, bg	av, aw, ax, ay, bg
	L	al			al
	M	am			am
	N	an, aq, ar			an, aq, ar
	O	an, ap, l			an, ap, l
	AF	an, ao			an, ao

Table D-12

Potential Usage Relating Input Data to Outputs

Potential Use	Outputs	Inputs: I Dynamic Simulation, Connection Statements	Inputs: II No Dynamic Simulation, Connection Statements	Inputs: III No Dynamic Simulation, Logic Statements	Inputs: IV Dynamic Simulation, Logic Statements
I Define the effect of a proposed change on operation of a selected portion of the launch vehicle and ground support systems.		1, 2, 3, 1C, 2C, 3C, 1T, 3T, 9T	1, 2, 3, 1C, 2C, 3C, 1T, 3T, 9T	1, 2, 4, 1C, 2C, 4C, 1T, 4T, 9T	1, 2, 4, 1C, 2C, 4C, 1T, 4T, 9T
	P	10T	10T	10T	10T
	Q	10T	10T	10T	10T
	R	6, 7, 6C, 7C, 6T, 7T, 10T, 11T	5, 5C, 5T, 10T, 11T	5, 5C, 5T, 10T, 11T	6, 7, 6C, 7C, 6T, 7T, 10T, 11T
	S	10T	10T		
	S	10T	10T		
	S	6, 7, 6C, 7C, 6T, 7T, 10T, 11T	5, 5C, 5T, 10T, 11T		
	AE	10T	10T	10T	10T
		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
	B	11T	11T	11T	11T
	C	11T	11T	11T	11T
	K	11T, 13T	11T, 13T	11T, 13T	11T, 13T
	A	11T	11T	11T	11T
	U	11T	11T	11T	11T
	V	11T, 15T	11T, 15T	11T, 15T	11T, 15T
	W, X	11T, 8, 8C, 8T	11T, 8, 8C, 8T	11T, 8, 8C, 8T	11T, 8, 8C, 8T
	D	11T	11T	11T	11T
	E	11T	11T		
	F	11T	11T	11T	11T
	G	11T	11T		
	H	11T	11T	11T	11T
	I	11T	11T		
	AD	11T	11T	11T	11T
	AD	11T	11T	11T	11T
	T	11T, 14T, 22T	11T, 14T, 22T	11T, 14T, 22T	11T, 14T, 22T
	L	11T			11T
	M	11T, 5, 5C, 5T			11T, 5, 5C, 5T
	N	11T, 21T			11T, 21T
	O	11T, 12T			11T, 12T
	AF	11T			11T
II Keep track of approved change orders, drawing changes, and hardware changes made in the simulation data file and the resultant configurations.	AG	1C, 2C, 3C, 6C, 7C, 8C	1C, 2C, 3C, 4C, 5C, 7C, 8C	1C, 2C, 4C, 5C, 7C, 8C	1C, 2C, 4C, 6C, 7C, 8C

Table D-13

Potential Usage Relating Input Data to Outputs

Potential Use	Outputs	Inputs: I Dynamic Simulation, Connection Statements	Inputs: II No Dynamic Simulation, Connection Statements	Inputs: III No Dynamic Simulation, Logic Statements	Inputs: IV Dynamic Simulation, Logic Statements
III Insert approved changes into central data file.		1, 2, 3, 5, 6, 7, 8, 1C, 2C, 3C, 5C, 6C, 7C, 8C	1, 2, 3, 4, 5, 8, 1C, 2C, 3C, 4C, 5C, 8C	1, 2, 4, 5, 8, 1C, 2C, 4C, 5C, 8C	1, 2, 4, 5, 6, 7, 8, 1C, 2C, 4C, 5C, 6C, 7C, 8C
	AH, AI	16C, 17C, 18C, 19C, 20C	16C, 17C, 18C, 19C, 20C	16C, 17C, 18C, 19C, 20C	16C, 17C, 18C, 19C, 20C
IV Change data temporarily to simulate a fault condition and follow its effect through a selected portion of the system.		1, 2, 3, 1C, 2C, 1T, 3T, 9T	1, 2, 3, 1C, 2C, 3C, 1T, 3T, 9T	1, 2, 4, 1C, 2C, 4C, 1T, 4T, 9T	1, 2, 4, 1C, 2C, 4C, 1T, 4T, 9T
	P	10T	10T	10T	10T
	Q	10T	10T	10T	10T
	R	6, 7, 6C, 7C, 6T, 7T, 10T, 11T	5, 5C, 5T, 10T, 11T	5, 5C, 5T, 10T, 11T	6, 7, 6C, 7C, 6T, 7T, 10T, 11T
	S	10T	10T		
	S	10T	10T		
	S	6, 7, 6C, 7C, 6T, 7T, 10T, 11T	5, 5C, 5T, 10T, 11T		
	AE	10T	10T	10T	10T
		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
	B	11T	11T	11T	11T
	C	11T	11T	11T	11T
	K	11T, 13T	11T, 13T	11T, 13T	11T, 13T
	A	11T	11T	11T	11T
	U	11T	11T	11T	11T
	V	11T, 15T	11T, 15T	11T, 15T	11T, 15T
	W, X	11T, 8, 8C, 8T	11T, 8, 8C, 8T	11T, 8, 8C, 8T	11T, 8, 8C, 8T
	D	11T	11T	11T	11T
	E	11T	11T		
	F	11T	11T	11T	11T
	G	11T	11T		
	H	11T	11T	11T	11T
	I	11T	11T		
	AD	11T	11T	11T	11T
	AD	11T	11T	11T	11T
	T	11T, 14T, 22T	11T, 14T, 22T	11T, 14T, 22T	11T, 14T, 22T
	L	11T			11T
	M	11T, 5, 5C, 5T			11T, 5, 5C, 5T
	N	11T, 21T			11T, 21T
	O	11T, 12T			11T, 12T
	AF	11T			11T

Table D-14

Potential Usage Relating Input Data to Outputs

Potential Use	Outputs	Inputs			
		I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
V Calculate expected times for events of the sequential operation of a selected portion of the launch vehicle and ground support systems.		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T			1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
	L	11T			11T
	M	11T, 5, 5C, 5T			11T, 5, 5C, 5T
VI Perform transient analysis of a selected portion of the launch vehicle and ground support systems.		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T			1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
	N	11T, 21T			11T, 21T
	O	11T, 12T			11T, 12T
	AF	11T			11T
VII Follow signals through a selected portion of the launch vehicle on a discrete basis.		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
	B	11T	11T	11T	11T
	C	11T	11T	11T	11T
	K	11T, 13T	11T, 13T	11T, 13T	11T, 13T
	A	11T	11T	11T	11T
VIII Relate the simulation to the racks, equipment numbers, etc. as given on panel schematics, interconnection diagrams, and advanced system schematics.		1, 2, 3, 1C, 2C, 3C, 1T, 3T, 9T	1, 2, 3, 1C, 2C, 3C, 1T, 3T, 9T	1, 2, 4, 1C, 2C, 4C, 1T, 4T, 9T	1, 2, 4, 1C, 2C, 4C, 1T, 4T, 9T
	P	10T	10T	10T	10T
	Q	10T	10T	10T	10T
	R	6, 7, 6C, 7C, 6T, 7T, 10T, 11T	5, 5C, 5T, 10T, 11T	5, 5C, 5T, 10T, 11T	6, 7, 6C, 7C, 6T, 7T, 10T, 11T
	S	10T	10T		
	S	10T	10T		
	S	6, 7, 6C, 7C, 6T, 7T, 10T, 11T	5, 5C, 5T, 10T, 11T		

Table D-15

Potential Usage Relating Input Data to Outputs

Potential Use		Outputs	Inputs			
			I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
IX	Search out closely timed operations and identify the equipments involved to eliminate areas of questionable operation where chance plays a significant role in the operation of a system.		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
		H	11T	11T	11T	11T
		I	11T	11T		
X	Check for inconsistencies such as conflicting signals and component operations which lead to inconsistent functions.		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
		F	11T	11T	11T	11T
		G	11T	11T		
XI	Check for redundancies to detect unintentional multiple methods of obtaining individual signals or modes of operation and also to verify the presence of intended redundant signals or modes included to improve reliability.		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
		D	11T	11T	11T	11T
		E	11T	11T		
XII	Define areas of possible malfunctions given a set of symptoms.	T	1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T, 11T, 14T, 22T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T, 11T, 14T, 22T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T, 11T, 14T, 22T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T, 11T, 14T, 22T

Table D-16

Potential Usage Relating Input Data to Outputs

Potential Use	Outputs	Inputs			
		I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
XIII Allow a user to set up conditions which identify a portion of a proposed or actual checkout or countdown sequence.	AJ	1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T, 11T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T, 11T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T, 11T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T, 11T
XIV Allow a set of simulated fault conditions to be superimposed on a list of conditions defining a planned checkout or countdown sequence.	AK	1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T, 11T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T, 11T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T, 11T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T, 11T
XV Define and keep track of equipments which have been activated and maintain a record for output.	U	1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T, 11T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T, 11T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T, 11T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T, 11T
XVI Define equipments which have not been activated.		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
	V	11T	11T	11T	11T

Table D-17

Potential Usage Relating Input Data to Outputs

Potential Use	Outputs	Inputs			
		I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
XVII Compare resulting sequences with desired ones.	K	1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T, 11T, 13T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T, 11T, 13T	1, 2, 4. 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T, 11T, 13T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T, 11T, 13T
XVIII Determine expected reliability factors for a selected portion of the system.		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
	U	11T	11T	11T	11T
	V	11T, 15T	11T, 15T	11T, 15T	11T, 15T
	W, X	11T, 8, 8C, 8T	11T, 8, 8C, 8T	11T, 8, 8C, 8T	11T, 8, 8C, 8T
XIX Configuration management documentation data central and control.	Y	16C	16C	16C	16C
	Z	17C	17C	17C	17C
	AA	18C	18C	18C	18C
	AB	19C	19C	19C	19C
	AC	20C	20C	20C	20C

Table D-18

Potential Usage Relating Input Data to Outputs

Potential Use	Outputs	Inputs: I Dynamic Simulation, Connection Statements	II No Dynamic Simulation, Connection Statements	III No Dynamic Simulation, Logic Statements	IV Dynamic Simulation, Logic Statements
XX Development of checkout and count-down procedures.		1, 2, 3, 1C, 2C, 3C, 1T, 3T, 9T	1, 2, 3, 1C, 2C, 3C, 1T, 3T, 9T	1, 2, 4, 1C, 2C, 4C, 1T, 4T, 9T	1, 2, 4, 1C, 2C, 4C, 1T, 4T, 9T
	P	10T	10T	10T	10T
	Q	10T	10T	10T	10T
	R	6, 7, 6C, 7C, 6T, 7T, 10T, 11T	5, 5C, 5T, 10T, 11T	5, 5C, 5T, 10T, 11T	6, 7, 6C, 7C, 6T, 7T, 10T, 11T
	S	10T	10T		
	S	10T	10T		
	S	6, 7, 6C, 7C, 6T, 7T, 10T, 11T	5, 5C, 5T, 10T, 11T		
	AE	10T	10T		
		1, 2, 3, 6, 7, 1C, 2C, 3C, 6C, 7C, 1T, 3T, 6T, 7T, 9T, 10T	1, 2, 3, 5, 1C, 2C, 3C, 5C, 1T, 3T, 5T, 9T, 10T	1, 2, 4, 5, 1C, 2C, 4C, 5C, 1T, 4T, 5T, 9T, 10T	1, 2, 4, 6, 7, 1C, 2C, 4C, 6C, 7C, 1T, 4T, 6T, 7T, 9T, 10T
	B	11T	11T	11T	11T
	C	11T	11T	11T	11T
	K	11T, 13T	11T, 13T	11T, 13T	11T, 13T
	A	11T	11T	11T	11T
	U	11T	11T	11T	11T
	V	11T, 15T	11T, 15T	11T, 15T	11T, 15T
	W, X	11T, 8, 8C, 8T	11T, 8, 8C, 8T	11T, 8, 8C, 8T	11T, 8, 8C, 8T
	D	11T	11T	11T	11T
	E	11T	11T		
	F	11T	11T	11T	11T
	G	11T	11T		
	H	11T	11T	11T	11T
	I	11T	11T		
	AD	11T	11T	11T	11T
	AD	11T	11T	11T	11T
	T	11T, 14T, 22T	11T, 14T, 22T	11T, 14T, 22T	11T, 14T, 22T
	L	11T			11T
	M	11T, 5, 5C, 5T			11T, 5, 5C, 5T
	N	11T, 21T			11T, 21T
	O	11T, 12T			11T, 12T
	AF	11T			11T

Table D-19

Interrelationships Between Functions

Function																Output	Usage
na		nbc	-	-	-	-	-	-	-	-	-	-	-	-	bi	AH	III
	nbd	-	-	-	-	-	-	-	-	-	-	-	-	-	bj	AI	III
		nbc	p	-	-	-	-	-	-	-	-	-	-	-	bh	AG	II
				-	-	-	-	-	-	-	-	-	-	-	q	Y	XIX
				-	-	-	-	-	-	-	-	-	-	-	r	Z	XIX
				-	-	-	-	-	-	-	-	-	-	-	s	AA	XIX
				-	-	-	-	-	-	-	-	-	-	-	t	AB	XIX
				-	-	-	-	-	-	-	-	-	-	-	u	AC	XIX
			nb	nc	nd	-	-	-	-	-	-	-	-	-	nae	P	I, IV, VIII, XX
						-	-	-	-	-	-	-	-	-	naf	Q	I, IV, VIII, XX
						nba	-	-	-	-	-	-	-	-	bb	AE	I, IV, XX
						nbe	ne	nf	bf	-	-	-	-	-	ag	R	I, IV, VIII, XX
								-	-	-	-	-	-	-	bk	AJ	XIII
								-	-	-	-	-	-	-	bl	AK	XIV
								nf	g	-	-	-	-	-	h	B	I, IV, VII, XX
										-	-	-	-	-	i	C	I, IV, VII, XX
										-	-	-	-	-	j	K	I, IV, VII, XVII, XX
										m	-	-	-	-	-	U	I, IV, XV, XVIII, XX
										m	-	-	-	-	o	V	I, IV, XVI, XVIII, XX
										m	-	-	-	-	n	W, X	I, IV, XVIII, XX
										x	-	-	-	-	aa	F	I, IV, X, XX
										w	-	-	-	-	z	D	I, IV, XI, XX
										y	-	-	-	-	ab	H	I, IV, IX, XX
										k	-	-	-	-	l	A	I, IV, VII, XX
										as	-	-	-	-	au	AD	I, IV, XX
										at	-	-	-	-	au	AD	I, IV, XX
										av	aw	ax	bg	-	ay	T	I, IV, XII, XX
										naz	nak	nv	an	-	ao	AF	I, IV, VI, XX
														-	al	L	I, IV, V, XX
														-	am	M	I, IV, V, XX
														ap	l	O	I, IV, VI, XX
														aq	ar	N	I, IV, VI, XX

Table D-20

Interrelationships Between Functions and Order of Implementation

Function																Output	Usage
na		nbc	-	-	-	-	-	-	-	-	-	-	-	-	bi	AH	III
	nbd	-	-	-	-	-	-	-	-	-	-	-	-	-	bj	AI	III
		nbc	p	-	-	-	-	-	-	-	-	-	-	-	bh	AG	II
			nb	nc	nd	-	-	-	-	-	-	-	-	-	nae	P	I, IV, VIII
						-	-	-	-	-	-	-	-	-	naf	Q	I, IV, VIII, XX
						nba	-	-	-	-	-	-	-	-	bb	AE	I, IV, XX
						nbe	bf	nf	-	-	-	-	-	-	ag	R	I, IV, VIII, XX
							ne	-	-	-	-	-	-	-	bk	AJ	XIII
								-	-	-	-	-	-	-	bl	AK	XIV
								nf	g	-	-	-	-	-	h	B	I, IV, VII, XX
										-	-	-	-	-	i	C	I, IV, VII, XX
										-	-	-	-	-	j	K	I, IV, VII, XVII, XX
										m	-	-	-	-	-	U	I, IV, XV, XVIII, XX
										m	-	-	-	-	o	V	I, IV, XVI, XVIII, XX
										naz	nak	nv	an	-	ao	AF	I, IV, VI, XX
													-	-	al	L	I, IV, V, XX
													-	-	am	M	I, IV, V, XX
										-	-	-	k	-	l	A	I, IV, VII, XX
										naz	nak	nv	an	ap	l	O	I, IV, VI, XX
										naz	nak	nv	an	aq	ar	N	I, IV, VI, XX
										m	-	-	-	-	n	W, X	I, IV, XVIII, XX
										x	-	-	-	-	aa	F	I, IV, X, XX
										w	-	-	-	-	z	D	I, IV, XI, XX
										av	aw	ax	bg	-	ay	T	I, IV, XII, XX
										as	-	-	-	-	au	AD	I, IV, XX
										at	-	-	-	-	au	AD	I, IV, XX
										y	-	-	-	-	ab	H	I, IV, IX, XX
			p	-	-	-	-	-	-	-	-	-	-	-	q	Y	XIX
				-	-	-	-	-	-	-	-	-	-	-	r	Z	XIX
				-	-	-	-	-	-	-	-	-	-	-	s	AA	XIX
				-	-	-	-	-	-	-	-	-	-	-	t	AB	XIX
				-	-	-	-	-	-	-	-	-	-	-	u	AC	XIX

D2 DESCRIPTION OF INDIVIDUAL UTILIZATIONS

In arriving at the desirable types of outputs, data, and necessary functions, recourse was taken to defining a series of potential uses to which a CLS might be applied. The result is the compilation which follows. A total of twenty were compiled and each is broken down into the following:

a. Purpose of the potential usage.
b. Usage to which this potential usage might be put.
c. Type and form of output required for the potential usage to be useful.
d. Variations on the simulation system which appear to be desirable.
e. Types of data required.
f. Functions required for implementation.
g. Tie-in of required functions.
h. Source of required data.

It is from such a compilation that the desirable outputs, data, and functions have been determined.

It is from these lists that selection is expected to be made for the CLS portions to be implemented.

I. DEFINE THE EFFECT OF A PROPOSED CHANGE ON OPERATION OF A SELECTED PORTION OF THE LAUNCH VEHICLE AND GROUND SUPPORT SYSTEMS

Potential Usage I applies to possible changes which might be made to equipment being simulated by CLS. Thus, it would cut across all functions which make up the simulation.

A. Purpose

1. To check the effect of a circuit parameter change on delays associated with operation of elements in the system portion under consideration.
2. To check the effect on operation sequence of a change in:
    a. Delays.
    b. Addition or deletion of elements.
    c. Addition or deletion of signals.
    d. Rerouting of signals.
3. To check for conflict in physical location of connections between elements of the system portion under consideration.
4. Changes in the data base are not made under this feature. This feature is a check only and if a permanent change is to be made, it will be handled under a different operation.
5. To check for possible redundancies, inconsistencies, or questionable operation of equipment making up the system portion under consideration using the possible change data.
6. To perform transient analysis for the system portion under consideration and using the possible change data.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for This Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-5 and/or D-12. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions of the system to be considered:
    a. Smallest part - a signal or a function.
    b. Intermediate portion - number of such signals or functions.
    c. Largest portion - total system including both the vehicle and ground systems.
2. One or more of the outputs as indicated to the simulation.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-12. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-5. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Reliability data.
5. Vendor information on equipment parameters.
6. Quality and reliability assurance test and analysis data for the equipment.

II. KEEP TRACK OF APPROVED CHANGE ORDERS, DRAWING CHANGES, AND HARDWARE CHANGES MADE IN THE SIMULATION DATA FILE AND THE RESULTANT CONFIGURATIONS

A. Purpose

To keep a record of the changes that have been made such that administrative and technical control can be kept to prevent the invalidation of some or all of the system data. This potential usage provides a method of keeping track of changes made in the data used by CLS in simulating equipment operations. It does not check the effect of such a change or the equipment being simulated and thus it does not duplicate Potential Usage I. Further, since Potential Usage II pertains to changes in data used by CLS in simulating equipment operations, it does not duplicate Potential Usage XIX.

B. Usage

Usages will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.
4. Simulation administrative personnel having responsibility for management of the simulation status.

C. Type and Form of Output Required for This Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-5 and/or D-12. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variation on the Simulation System

Portions of the system to be considered:

1. Smallest portion - a signal, a function, or an element in which a change has been made.
2. Intermediate portion - a number of such signals, functions, or elements.
3. The number of such changes could extend to the total system although such an extension would not be expected since completely new data would be read in for such a case.

Changes resulting in the data control from such other changes as those fed in will be made automatically. This results in the entries in the data control being consistent. Records of such changes will be kept and will be part of the output.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-12. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-5. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Reliability data.
5. Vendor information on equipment parameters.
6. Quality and reliability assurance test and analysis data for the equipment.

III. INSERT APPROVED CHANGES INTO THE CENTRAL DATA FILE

A. Purpose

To update the data used by CLS such that it reflects the system configuration in its latest form. This updating consists of inserting new information for all items of data associated with the following:

1. Connection statements.
2. Logical statements.
3. Advanced schematics.
4. Panel schematics.
5. Delays.
6. Element parameters.

These changes will be made only under carefully controlled procedures and a record will be kept.

B. Usage

Since the use of this feature results in updated information being in the data central, the results of its use are available in all areas requiring:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for This Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-6 and/or D-13. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

Portions of the system to be considered:

1. Smallest portion - a signal, a function, or an element in which a change has been made as indicated under the above section - Purpose.
2. Intermediate portion - a number of such signals, functions, or elements.
3. The number of such changes could extend to the total system although such an extension would not be expected since completely new data would be read in for such a case.

Changes resulting in the data central from such other changes as those fed in by this feature will be made automatically. This results in the entries in the data central being consistent.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-13. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-6. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Reliability data.
5. Vendor information on equipment parameters.
6. Quality and reliability assurance test and analysis data for the equipment.

IV. CHANGE THE DATA TEMPORARILY TO SIMULATE A FAULT CONDITION AND FOLLOW ITS EFFECT THROUGH A SELECTED PORTION OF THE SYSTEM

A. Purpose

To allow one to make such temporary changes in the data that a fault condition or a number of fault conditions may be simulated. Through the use of all of the features of the simulation, the effect of such fault conditions may be traced through the vehicle and/or ground system or a selected portion of it. Changes in the data may consist of the following, singly or in consistent combinations:

1. Changes in connection statements.
2. Changes in delays.
3. Changes in element parameters.
4. Changes in logical statements.
5. Elimination of elements or signals.

This feature closely parallels the one whereby the effect of a proposed change may be traced.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.
8. Knowledge of the effect of abnormal conditions on system operations.
9. Knowledge of maintenance activities required.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.

3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment. This includes those training, those trained, and those doing the training.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-6 and/or D-13. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions of the system to be considered:
   a. Smallest part - change in a signal, element, or a function.
   b. Intermediate part - change in a number of such signals, elements, or functions.
   c. Largest part - theoretically the whole vehicle and/or ground system. However, it is inconceivable that this many faults will be simulated.
   d. The effects of these changes will be observed on a specified portion of the system up to the whole.
2. One or more of the outputs may be obtained as desired.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-13. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-6. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Reliability data.
5. Vendor information on equipment parameters.
6. Quality and reliability assurance test and analysis data for the equipment.

V. CALCULATE EXPECTED TIMES FOR EVENTS OF THE SEQUENTIAL OPERATION OF A SELECTED PORTION OF THE LAUNCH VEHICLE AND GROUND SUPPORT SYSTEMS

A. Purpose

To determine the times at which discrete operations may occur but based on delays calculated from parameters of the elements rather than assumed delays. This pertains to electrical, mechanical, hydraulic, and pneumatic elements in the portion of the vehicle and/or ground system under consideration.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Logical connections.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-7 and/or D-14. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variation on the Simulation System

1. Portion of the system to be considered:
   a. Smallest part - a signal or function.
   b. Intermediate part - a number of such signals or functions.
   c. Largest part - total system including both the vehicle and ground system.
2. Outputs from this to be used with portions of the simulation for discrete events.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-14. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-7. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Vendor information on equipment parameters.
5. Quality and reliability assurance test and analysis data for the equipment.

VI. PERFORM TRANSIENT ANALYSIS OF A SELECTED PORTION OF THE LAUNCH VEHICLE AND GROUND SUPPORT SYSTEMS

A. Purpose

To perform transient analysis of a selected portion of the vehicle and ground system, the objective being:

1. To determine the time of operation of the elements based on their parameters.
2. To determine stability characteristics for the portion of the system under consideration.

The portion of the system under consideration includes electrical, mechanical, hydraulic, and pneumatic elements. Further, discrete as well as continuous operation will also enter into the analysis.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-7 and/or D-14. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portion of the system to be considered:
   a. Smallest part - a signal or function.
   b. Largest part - total system including both the vehicle and ground system.
2. One or more of the outputs as desired and indicated to the simulation.
3. Outputs from this to be used with portions of the simulation for discrete events.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-14. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-7. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Vendor information on equipment parameters.
5. Quality and reliability assurance test and analysis data for the equipment.

VII. FOLLOW SIGNALS THROUGH A SELECTED PORTION OF THE LAUNCH VEHICLE ON A DISCRETE BASIS

A. Purpose

The object of Potential Usage VII is to allow the user to follow the sequence of operations, on a discrete basis, through a selected portion of the vehicle and/or ground system. The system involved includes electrical, mechanical, hydraulic, and pneumatic equipment. The time intervals either have predefined values or are the result of Potential Usage V - Calculate Expected Times for Events of the Sequential Operation of a Selected Portion of the Launch Vehicle and Ground Support Equipment.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground systems with changes included.
2. Evaluation of alternate schemes for the sequential operation of the vehicle and/or ground system.
3. Logical connections.
4. Panel schematics.
5. Advanced schematics.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-7 and/or D-14. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions of the system to be considered:
   a. Smallest part - a signal or a function.
   b. Largest portion - total system including both the vehicle and/or ground system.
   c. Intermediate portion - a number of signals or functions.

2. One or more of the outputs as indicated to the simulation.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-14. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-7. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

Tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Vendor information on equipment parameters.
5. Quality and reliability assurance test and analysis data for the equipment.

VIII. RELATE THE SIMULATION TO THE RACKS, EQUIPMENT NUMBERS, ETC., AS GIVEN ON PANEL SCHEMATICS, INTERCONNECTION DIAGRAMS, AND ADVANCED SYSTEM SCHEMATICS

A. Purpose

The purpose of Potential Usage VIII is to provide, for the user, the ability of relating the equipment used in the vehicle and/or ground equipment back to the drawings. This can appear in more than one form such as:

1. Coding of equipment designations such that they reflect type and location.
2. Tying the equipment back to panel and rack.
3. Tying equipment back to drawing and page number

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Logical connections.
2. Panel and rack equipment.
3. Panel schematics.
4. Advanced schematics.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-7 and/or D-14. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. One or more of the outputs as indicated to the simulations.
2. Portion of the system as desired and as indicated to the simulation.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-14. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-7. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.

IX. SEARCH OUT CLOSELY TIMED OPERATIONS AND IDENTIFY THE EQUIPMENT INVOLVED TO ELIMINATE AREAS OF QUESTIONABLE OPERATION WHERE CHANCE PLAYS A SIGNIFICANT ROLE IN THE OPERATION OF A SYSTEM

A. Purpose

Closely timed functions (e.g., relay races) may sometimes appear inadvertently in a system and so this potential usage is being considered as a means for calling attention to such a condition, if it exists. Changes in the data which may result from such a disclosure is not automatic - they would require evaluation and then use of procedures necessary for updating of data. Thus, this potential usage has, as its objective, the following:

1. To search out and define closely timed operations which can result in questionable operation in the vehicle and ground system.
2. To define the equipment involved in such closely timed operations in the ground and vehicle system that they result in highly restrictive or questionable operation of the system.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-8 and/or D-15. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions of the system to be considered:
   a. Smallest part - a signal or function.
   b. Largest part - total system including both the vehicle and ground systems.
   c. Intermediate portion - a number of such signals or functions.
2. One or more of the outputs as desired and indicated to the simulation.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-15. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-8. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Vendor information on equipment parameters.
5. Quality and reliability assurance test and analysis data for the equipment.

X. CHECK FOR INCONSISTENCIES SUCH AS CONFLICTING SIGNALS AND COMPONENT OPERATIONS WHICH LEAD TO INCONSISTENT FUNCTIONS

A. Purpose

Inconsistent functions are not intentionally designed into a system and so this potential usage is being considered as a means for calling attention to such a consideration, if it exists. Changes in the data which can accompany such a disclosure are not automatic, they would require evaluation and then use of procedures necessary for updating of data. This potential usage has as its objective, the following:

1. Check for signals which conflict.
2. Check for components, the operation of which lead to inconsistent functions.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Logical connections.
4. Panel and rack equipments.
5. Panel schematics.
6. Advanced schematics.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-8 and/or D-15. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions of the system to be considered
   a. Smallest part - a signal or a function.
   b. Intermediate portion - a number of such signals or functions.
   c. Largest part - the total system including both the vehicle and ground systems.
2. One or more of the outputs as indicated to the simulation.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-15. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions in coded forms, required for this potential usage are given on Table D-8. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Vendor information on equipment parameters.
5. Quality and reliability assurance test and analysis data for the equipment.

XI. CHECK FOR REDUNDANCIES TO DETECT UNINTENTIONAL MULTIPLE METHODS OF OBTAINING INDIVIDUAL SIGNALS OR MODES OF OPERATION AND ALSO TO VERIFY THE PRESENCE OF INTENDED REDUNDANT SIGNALS OR MODES INCLUDED TO IMPROVE RELIABILITY

A. Purpose

Redundancies in a system may appear inadvertently or may have been included intentionally (e.g., to increase reliability). This potential usage is being considered as a means for calling attention to the fact that a redundancy may exist. Changes which may result in the data will not be automatic - they would require evaluation and then use of procedures necessary for the updating of the data. Thus, this potential usage has, as its objective, the following:

1. Check for multiple methods of obtaining individual signals.
2. Check for components, the operation of which lead to duplicate functions.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-8 and/or D-15. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions on the system to be considered:
   a. Smallest part - a signal or function.
   b. Intermediate portion - a number of such signals or functions.
   c. Largest part - total system including both the vehicle and ground systems.
2. One or more of the outputs as indicated to the simulation.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-15. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-8. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:
1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Vendor information on equipment parameters.
5. Quality and reliability assurance test and analysis data for the equipment.

XII. DEFINE AREAS OF POSSIBLE MALFUNCTIONS GIVEN A SET OF SYMPTOMS

A. Purpose

This feature is essentially the reverse of Potential Usage XIV - change the data temporarily to simulate a fault condition and follow its effect through a selected portion of the system. It has as its purpose to define possible malfunctions which can give rise to the given set of symptoms. It would be restricted to the case where the symptoms could result from single rather than multiple simultaneous equipment malfunctions.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.
8. Effect of abnormal conditions on vehicle and/or ground systems operations.
9. Diagnosis of fault conditions which can occur during checkout or countdown activities.
10. Maintenance activities which might be required.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment. Specifically this covers checkout or countdown operating personnel who will direct testing and maintenance activities particularly where fault diagnosis is required. It includes those in training, those trained, and those doing the training.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-8 and/or D-15. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System (Portions of the System to be Considered)

a. Smallest portion - a symptom consisting of one erroneous signal.

b. Normal sized part - a symptom consisting of a number of erroneous signals.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-15. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-8. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Reliability data.
5. Vendor information on equipment parameters.
6. Quality and reliability assurance test and analysis data for the equipment.

XIII. ALLOW A USER TO SET UP CONDITIONS WHICH IDENTIFY A PORTION OF A PROPOSED OR ACTUAL CHECKOUT OR COUNTDOWN SEQUENCE

A. Purpose

To allow the user to set up the conditions which spell out a portion of a proposed or actual checkout or countdown. Since this potential usage ties in with the two listed below, its function is not to duplicate them. Instead, it provides a means for specifying the time interval or equipment bounds of the system portion to be investigated along with the necessary initial conditions. The two potential usages with which it ties in are:

1. Define the effect of a proposed change in the operation of a selected portion of the vehicle and/or ground system.
2. Change the data temporarily to simulate a fault condition and follow its effect through a selected portion of the system.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.
8. Checkout and countdown procedures, either existing ones or ones in development.
9. Effect of test procedures on operation of the system.
10. Maintenance activities required.
11. Effect of abnormal conditions on system operations.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment. Specifically, this covers checkout or countdown operating personnel who will direct testing and maintenance

activities particularly where fault diagnosis is required. It includes those in training, those trained, and those doing the training.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-9 and/or D-16. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions of the system to be considered:
   a. Smallest part - input conditions or output conditions concerned with a single signal, element or logical statement.
   b. Intermediate portion - input conditions or output conditions concerned with a group of signals, elements of logical statements.
   c. Largest portion - a test sequence concerned with the whole system.
2. One or more of the outputs as indicated to the simulation by the user.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-16. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-9. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Quality and reliability assurance test specifications.

XIV. ALLOW A SET OF SIMULATED FAULT CONDITIONS TO BE SUPERIMPOSED ON A LIST OF CONDITIONS DEFINING A PLANNED CHECKOUT OR COUNTDOWN SEQUENCE

A. Purpose

To allow the user to simulate a system fault during a simulated checkout or countdown. Since this potential usage ties in with the three listed below, its function is not to duplicate them. Instead, it provides a means for specifying simulated faults in a simulated checkout or countdown sequence. The three potential uses with which it ties in are:

1. Change the data temporarily to simulate a fault condition and follow its effect through a selected portion of the system.
2. Define the effect of a proposed change on the operation of a selected portion of the system.
3. Allow an operator to set up initial conditions according to a predefined plan on the vehicle and checkout equipment.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.
8. Checkout and countdown procedures either existing or in development.
9. Effect of test procedures on system operation.
10. Maintenance activities required.
11. Effect of abnormal conditions on system operations.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment. Specifically, this covers checkout or

countdown operating personnel who will direct testing and maintenance activities particularly where fault diagnosis is required. It includes those in training, those trained, and those doing the training.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-9 and/or D-16. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions of the system to be considered:
   a. Smallest part - input conditions or output conditions concerned with a single signal, element or logical statement.
   b. Intermediate portion - input or output conditions concerned with a group of signals, elements, or logical statements.
   c. Largest portion - a test sequence concerned with the whole system. However, simulated faults would only appear in limited numbers.
2. One or more of the listed outputs as indicated by the user to the simulation.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-16. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given in Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-9. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Quality and reliability assurance test specifications.

XV. DEFINE AND KEEP TRACK OF EQUIPMENT WHICH HAS BEEN ACTIVATED AND MAINTAIN A RECORD FOR OUTPUT

A. Purpose

To help the user to verify the efficiency of an existing or proposed test procedure. This potential usage is intended to identify the equipment which has been exercised and the time of activation. This is particularly applicable during a simulated checkout or countdown.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Equipment and system reliability.
2. Checkout or countdown procedures either existing or in development.
3. Effect of test procedures on system operation.
4. Maintenance activities required.
5. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-9 and/or D-16. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

Portions of the system to be considered:

1. Smallest part - a signal or function.
2. Intermediate portion - a number of such signals or functions.
3. Largest part - total system including both the vehicle and ground systems.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-16. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alpha-numeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-9. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. These are information flow diagrams relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Quality and reliability assurance test sequences.

XVI. DEFINE EQUIPMENTS WHICH HAVE NOT BEEN ACTIVATED

A. Purpose

To help the user to verify the efficiency of an existing or a proposed test procedure. This potential usage is intended to identify equipment which was not exercised even though it was intended to be activated during some portion of the test procedure under investigation. It is essentially the converse of Potential Usage XV - Define and Keep Track of Equipment which has been Exercised and Maintain a Record for Output.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Equipment and system reliability.
2. Checkout or countdown procedures either existing or in development.
3. Effect of test procedures on system operation.
4. Maintenance activities required.
5. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-9 and/or D-16. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

Portions of the system to be considered:

1. Smallest part - a signal or function.
2. Intermediate portion - a number of such signals or functions.

3. Largest part - total system including both the vehicle and ground systems.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-16. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-9. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. These are information flow diagrams relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Quality and reliability assurance test sequences.

XVII. COMPARE RESULTING SEQUENCES WITH DESIRED ONES

A. Purpose

The purpose of this potential usage is to permit a user to compare equipment states resulting from a simulation run with expected states. It is not intended that this potential usage duplicate other usages, such as determining the equipment states. Instead, it uses such information and compares it with anticipated ones.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.
8. Checkout or countdown procedures either existing or in development.
9. Effect of test procedures on system operation.
10. Maintenance activities required.
11. Effect of abnormal conditions on system operation.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment. Specifically, this covers checkout or countdown operating personnel who will direct testing and maintenance activities particularly where fault diagnosis is required. It includes those in training, those trained, and those doing the training.

C. <u>Type and Form of Output for this Feature to be Useful</u>

The types of output associated with this potential usage are listed on Tables D-10 and/or D-17. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. <u>Variations on the Simulation System</u>

Portions of the system to be considered:

a. Smallest part - a signal or function.

b. Intermediate portion - a number of such signals or functions.

c. Largest part - total system including both the vehicle and ground systems.

E. <u>Types of Data Required</u>

The types of data required by this potential usage are listed on Table D-17. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. <u>Particular Function Required</u>

The functions, in coded forms, required for this potential usage are given on Table D-10. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. <u>Tie-In of Functions</u>

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. These are information flow diagrams relating functions with data and potential outputs.

H. <u>Source of Data</u>

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.
3. Connection diagrams.
4. Vendor information on equipment parameters.
5. Quality and reliability assurance test sequences.

XVIII. DETERMINE THE EXPECTED RELIABILITY FACTORS FOR A SELECTED PORTION OF THE SYSTEM

A. Purpose

1. To determine the effective failure rate for a selected portion of the system based on checkout or countdown usage. In making reliability predictions, the normal assumption is that the failure rate (λ) is constant which, in turn, assumes a Poisson distribution for failure prediction. The determination of failure rate for an equipment that is made up of many elements becomes quite involved. A reasonable estimate of the equipment failure rate may be obtained from the sum of weighted failure rates of the elements - weighted by the ratio of time of activation (number of activations) to the total activation time (total number of activations) of the equipment. Since these times (activations) are normally ball-park guesses, keeping track of them during a simulated test procedure can improve precision in the equipment failure rate.
2. To allow the user to predict the probability of a failure (or its converse, not failing) in an equipment during a selected portion of a test sequence.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Equipment and system reliability.
2. Checkout or countdown procedures either existing or in development.
3. Effect of test procedures on system operation.
4. Maintenance activities required.
5. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipment.

C. Type and Form of Output Required for this Feature to be Useful

The types of output associated with this potential usage are listed on Tables D-10 and/or D-17. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. Variations on the Simulation System

1. Portions of the system to be considered:
   a. Smallest part - a signal or a function.
   b. Intermediate portion - a number of such signals or functions.
   c. Largest part - total system including both the vehicle and ground systems.
2. One or more of the outputs as indicated to the simulation.

E. Types of Data Required

The types of data required by this potential usage are listed on Table D-17. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. Particular Function Required

The functions, in coded forms, required for this potential usage are given on Table D-10. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. Tie-In of Functions

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. These are information flow diagrams relating functions with data and potential outputs.

H. Source of Data

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.

3. Connection diagrams.
4. Reliability data.
5. Vendor information on equipment parameters.
6. Quality and reliability assurance test specifications and analysis data for the equipment.

XIX. CONFIGURATION MANAGEMENT DOCUMENTATION DATA CENTER AND CONTROL

A. Purpose

1. To allow the user to follow the procedural concepts required for identification, control, and accounting for all systems, equipment, and components of the Saturn V launch vehicle. These are the procedures outlined by NPC 500-1.
2. Specifically this is accounting information directed toward keeping track of the following information:
    a. Specifications for contract end items.
    b. Changes to and maintenance of the specifications.
    c. Engineering documentation required for:
        (1) Design releases.
        (2) Design changes.
        (3) Design reviews.
        (4) Test acceptance and reviews.
3. This does not entail storage of the documents themselves into the computer bulk memory, but, rather, the document identification (indices) containing the required information on changes to the design, when completely cleared through the configuration management system.
4. Reference to engineering changes in layout, structural, etc., drawings will be from the indices listed under outputs.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. End item approved configuration indices.
2. Approved ECP end item indices.
3. End-item quantitative requirements schedule.
4. End-item modification status.
5. Spares status.

C. <u>Type and Form of Output Required for this Feature to be Useful</u>

The types of output associated with this potential usage are listed on Tables D-10 and/or D-17. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes is given on Table D-3.

D. <u>Variations on the Simulation System</u>

One or more of the outputs as indicated to the simulation.

E. <u>Types of Data Required</u>

The types of data required by this potential usage are listed on Table D-17. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. <u>Particular Function Required</u>

The functions, in coded forms, required for this potential usage are given on Table D-10. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. <u>Tie-In of Functions</u>

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. These are information flow diagrams relating functions with data and potential outputs.

H. <u>Source of Data</u>

1. End item approved configuration indices.
2. Approved ECP end item indices.
3. End item quantitative requirements schedule.
4. End item modification status.
5. Spares status.

XX. DEVELOPMENT OF CHECKOUT AND COUNTDOWN PROCEDURES

A. Purpose

The intent of this potential usage is to act as an aid in preparing test procedures for checkout and countdown. It is not intended that this usage write the procedures, but, by simulating the consequences of a procedural step, it acts as a tool for individuals responsible for writing such procedures.

B. Usage

Usage will be in areas requiring knowledge of the following:

1. Sequential operation of the vehicle and/or ground system with changes included.
2. Dynamic operation of the vehicle and/or ground system with changes included.
3. Evaluation of alternate schemes for the sequential and/or dynamic operation of the vehicle and/or ground system.
4. Logical connections.
5. Panel and rack equipment.
6. Panel schematics.
7. Advanced schematics.
8. Checkout or countdown procedures either existing or in development.
9. Effect of test procedures on system operation.
10. Maintenance activities required.
11. Effect of abnormal conditions on system operation.

In particular this covers the following areas:

1. Design Engineering.
2. Quality and Reliability Assurance.
3. Checkout and countdown personnel requiring knowledge of logical and dynamic operation of equipments. Specifically, this covers personnel responsible for developing and writing up test procedures to be used in checkout and countdown.

C. <u>Type and Form of Output Required for this Feature to be Useful</u>

The types of output associated with this potential usage are listed on Tables D-11 and/or D-18. These output types are coded (capital alphabetic characters) for conciseness. The dictionary for these codes if given on Table D-3.

D. <u>Variations on the Simulation System</u>

1. Portions of the system to be considered:
   a. Smallest part - a signal or a function.
   b. Intermediate portion - a number of such signals or functions.
   c. Largest part - total system including both the vehicle and ground systems.
2. One or more of the outputs as indicated to the simulation.

E. <u>Types of Data Required</u>

The types of data required by this potential usage are listed on Table D-18. This sheet relates the required types of data to the potential outputs and all entries are in coded form. The dictionary for the required data codes (alphanumeric) is given on Table D-4.

F. <u>Particular Function Required</u>

The functions, in coded forms, required for this potential usage are given on Table D-11. This sheet relates required functions to potential outputs and the dictionary for these coded functions is given on Tables D-1 and D-2.

G. <u>Tie-In of Functions</u>

The tie-in of the required functions for this potential usage is shown on Figures D-1 through D-4. This is an information flow diagram relating functions with data and potential outputs.

H. <u>Source of Data</u>

The sources of the required data for this potential usage are as follows:

1. Advanced system schematics.
2. Panel schematics.

3. Connection diagrams.
4. Reliability data.
5. Vendor information on equipment parameters.
6. Quality and reliability assurance test sequences and analysis data for the equipment.

Sample Sequence of Operations by Time - Output B

SATURN 1B SA201 PRELIMINARY S IVB SIMULATION 06/10/65 PAGE 15

COMPONENT	TYPE	NOMENCLATURE	STATUS	COMPONENT	TYPE	NOMENCLATURE	STATUS
TIME = -93.000				TIME = -93.000			
TLA093	I	LS PRESS PROPELLANTS	ON				
AA022	S	REQ LOX PRESS FOR DRAIN	ON	AA167	S	PRESS LOX DRAIN OR LAUNC	ON
AA175	S	PRESS FUEL FOR DRAIN/LAU	ON	AA242	S	LOX FILL/DRAIN CLOSE	OFF
AA347	S	L.S. PRESS PROPELLANTS	ON	AAC168	S	LH2 VENT BOOST 5 SEC.TIM	ON
AAC190	S	LOX VENT BOOST 5 SEC.TIM	ON	AAC241	S	LOX F+D BOOST 5 SEC.TIME	OFF
AAC315	S	LOX F+D BOOST 2 SEC.DELA	OFF	AAC348	S	LOX VENT BOOST 2 SEC.DEL	ON
AAC349	S	LH2 VENT BOOST 2 SEC.DEL	ON				
TIME = -92.000				TIME = -92.000			
TLA093	I	LS PRESS PROPELLANTS	OFF				
TIME = -77.000				TIME = -77.000			
TLA077	I	LS PRESS PROPS +16 SEC	ON				
DT343	S	LAU SEQ PRESS PROP +16SE	ON				
TIME = -76.000				TIME = -76.000			
TLA077	I	LS PRESS PROPS +16 SEC	OFF				
TIME = -28.000				TIME = -28.000			
TLA028	I	LS PT SIGNAL	ON				
AA005	S	P.T.COMMAND INTERNAL	ON	AA047	S	SEQUENCER PWR TRANSFER	ON
AGL14	S	S IVB PWR.EXT.	OFF	AI0135	S	NOMENCLATURE UNKNOWN	OFF
DEGL1	S	PT EXTERNAL	OFF	DEGPK1	S	PT EXTERNAL	OFF
DEHL1	S	PT INTERNAL	ON	DEHPK1	S	PT INTERNAL	ON
L00135	S	NOMENCLATURE UNKNOWN	OFF				
TIME = -27.965				TIME = -27.965			
KRKS1B	D	FWD POWER TRANSFER SWITC	ON	KZAS1B	D	AFT BUS 2 ON INTERNAL	ON
KZCS1B	D	AFT BUS 1 ON INTERNAL	ON				
AA006	S	AFT BUS NO.2 ON INTERNAL	ON	AA007	S	AFT BUS NO.1 ON INTERNAL	ON
AA008	S	FWD BUS ON INTERNAL	ON				
TIME = -27.960				TIME = -27.960			
KRKS1A	D	FWD POWER TRANSFER SWITC	OFF	KZAS1A	D	AFT BUS 2 ON EXTERNAL	OFF
KZCS1A	D	AFT BUS 1 ON EXTERNAL	OFF				
TIME = -27.865				TIME = -27.865			
AAC166	D	PT TIMER 100 MILLISECOND	ON				
A4D111	S	GRD PWR TO AFT BUS 1 28V	OFF	A4D121	S	GRD PWR TO FWD BUS 2 28V	OFF
A4D131	S	GRD PWR TO FWD BUS 1 28V	OFF	A4D141	S	GRD PWR TO AFT BUS 2 56V	OFF
AA010	S	S4B POWER INTERNAL	ON	AA019	S	POWER TO STAGE	OFF
AA331	S	+4D141 SUPERVISION ON	OFF	AGL07	S	S IVB PWR.INT.	ON
AGL50	S	READY FOR +4D141 PWR.	OFF	AI0137	S	NOMENCLATURE UNKNOWN	ON

Sample Listing of Component Status Changes - Output C

SATURN 1B   SA201   PRELIMINARY S IVB SIMULATION   06/10/65   PAGE 1

STATUS CHANGES GROUPED ACCORDING TO COMPONENTS

COMPONENT	NOMENCLATURE	ON	OFF	ON	OFF	COUNT
20D100	LCC POWER SUPPLY	-1024.000				
20D110	LCC BUS	-1024.000				
A4D10	S4B STAGE AFT BATT 1 28V	-1023.900				
A4D100	POWER SUPPLY NO 1 28V	-1023.900				
A4D11	S4B STAGE AFT BUS 1 28V	-1023.750				
A4D111	GRD PWR TO AFT BUS 1 28V	-1023.750	-27.865			
A4D110	ESE NETWORKS BUS NO 1 28	-1023.900				
A4D113	COMMIT BUS	-1023.750				
A4D115	PRE LAUNCH BUS	-1023.750	-153.000			
A4D119	S4B STAGE INDICATION BUS	-1023.850				
A4D12	APS 1 POWER BUS	-1023.200				
A4D121	GRD PWR TO FWD BUS 2 28V	-1023.750	-27.865			
A4D13	APS 2 POWER BUS	-1023.200				
A4D131	GRD PWR TO FWD BUS 1 28V	-1023.750	-27.865			
A4D141	GRD PWR TO AFT BUS 2 56V	-1023.550	-27.865			
A4D15	SEQUENSER POWER BUS	-1022.600				
A4D20	S4B STAGE FWD BATT 2 28V	-1023.900				
A4D200	POWER SUPPLY NO 2 28V	-1023.900				
A4D21	S4B STAGE FWD BUS 2 28V	-1023.750				
A4D210	ESE NETWORKS BUS NO 4 16	-1023.900				
A4D30	S4B STAGE FWD BATT 1 28V	-1023.900				
A4D300	POWER SUPPLY NO 3 28V	-1023.900				
A4D31	S4B STAGE FWD BUS 1 28V	-1023.750				
A4D310	ESE NETWORKS BUS NO 3 28	-1023.900				
A4D40	S4B STAGE AFT BATT 2 56V	-1023.900				
A4D400	POWER SUPPLY NO 4 56V	-1023.900				
A4D41	S4B STAGE AFT BUS 2 56V	-1023.550				
A4D410	ESE NETWORKS BUS NO 2 6	-1023.900				
A4D42	AUX.HYD.MOTOR POWER BUS	-1021.800				
A4D43	AUX.HYD.MOTOR POWER BUS	-1021.800				
A4D810	GROUND MEAS POWER (A4D80					
AADCT	COMPONENT TEST POWER					
AADECT	ENGINE COMPONENT TEST PW					
AADPSP	PLUG SUPERVISION POWER	-1023.900				
AAJUMP	APS SERVCRS 28V TALKBACK					
AA001	LAUNCH BUS ENERGIZED	-153.000				
AA002	COMMIT					
AA003	+4D113 ON	-1023.750				
AA005	P.T.COMMAND INTERNAL	-28.000				
AA006	AFT BUS NO.2 ON INTERNAL	-27.965				
AA007	AFT BUS NO.1 ON INTERNAL	-27.965				
AA008	FWD BUS ON INTERNAL	-27.965				
AA010	S4B POWER INTERNAL	-27.865				
AA016	CKOUT MEAS GROUP COMM					
AA019	POWER TO STAGE	-1023.750	-27.865			
AA020	STAGE CONT.ME S/O CLOSE					
AA022	REQ LOX PRESS FOR DRAIN	-93.000				
AA027	FIRING COMMAND	-153.000				
AA031	NETWORKS MALFUNCTION	-1023.900	-1023.550			
AA034	+4D10 ON	-1023.850				

Sample Listing of Comparison Run - Output K

SATURN 1B SA201 PRELIMINARY S IVB SIMULATION 06/16/65 PAGE 20

AN ATTEMPT TO DETERMINE A FALT INDICATIOR PATTERN

COMPONENT	TYPE	NOMENCLATURE	STATUS	DIFF.	COMPONENT	TYPE	NOMENCLATURE	STATUS	DIFF.
TIME = -3.000					TIME = -3.000				
AKS61A	S	COLD HE SUP,VENT CLOSED	OFF		AKS618	S	COLD HE SUP,VENT OPEN	ON	
AML01	S	BOT SUP VENT OPENED	ON		AML07	S	START TK SUP VENT OPENED	ON	
AML09	S	BOT SUP VENT CLOSED	OFF		AML15	S	START TK SUP VENT CLOSED	OFF	
DT253	S	SIB IGNITION COMM.	ON		DT306	S	START TANK SUPPLY VENT	ON	
L00155	S	NOMENCLATURE UNKNOWN	OFF		L00156	S	NOMENCLATURE UNKNOWN	ON	
L00162	S	NOMENCLATURE UNKNOWN	OFF		L00163	S	NOMENCLATURE UNKNOWN	ON	
L00171	S	NOMENCLATURE UNKNOWN	ON		L00172	S	NOMENCLATURE UNKNOWN	OFF	
TIME = -0.000					TIME = -0.000				
THIS TIME INTERVAL APPEARS IN BOTH RUNS									
TLA000	I	LS COMMIT SIGNAL	ON						
A4D111	S	GRD PWR TO AFT BUS 1 28V	OFF	TS	A4D113	S	COMMIT BUS	OFF	
A4D119	S	S4B STAGE INDICATION BUS	OFF		A4D121	S	GRD PWR TO FWD BUS 2 28V	OFF	TS
A4D131	S	GRD PWR TO FWD BUS 1 28V	OFF	TS	A4D141	S	GRD PWR TO AFT BUS 2 56V	OFF	TS
AA002	S	COMMIT	ON		AA003	S	*4D113 ON	OFF	
AA005	S	R.T.COMMAND INTERNAL	OFF		AA006	S	AFT BUS NO.2 ON INTERNAL	OFF	
AA007	S	AFT BUS NO.1 ON INTERNAL	OFF		AA008	S	FWD BUS ON INTERNAL	OFF	
AA010	F	S4B POWER INTERNAL	OFF	NS	AA019	S	POWER TO STAGE	OFF	TS
AA031	S	NETWORKS MALFUNCTION	ON		AA034	S	*4D10 ON	OFF	
AA043	S	RECIRCULATION OK	OFF		AA048	S	AUX HYD COAST MODE RESET	OFF	
AA054	S	S IVB STAGE POWER ON	OFF		AA061	S	A4D20 ON	OFF	
AA067	S	*4D30 ON	OFF		AA068	S	ENGINE READY	OFF	
AA070	S	A4D40 ON	OFF		AA075	S	LH DIR VENT IN FLIGHT PO	OFF	
AA080	S	LH F AND D CLOSED IND	OFF		AA085	S	LOX FILL AND DRAIN CLOSE	OFF	
AA093	S	STG CONT HE MIN L/O OK	OFF		AA096	S	LOX TANK LIFTOFF PRESS.O	OFF	
AA100	S	COLD HE MIN LIFTOFF PRES	OFF		AA114	S	LH PREVALVE CLOSED	OFF	
AA115	S	LOX PREVALVE CLOSED	OFF		AA153	S	SEQ BUS PWR ON IND	OFF	
AA226	S	LH EMER SHTOFF VALVE OPE	OFF		AA227	S	LOX EMER SHTOFF VALVE OP	OFF	
AA231	S	LOX RECIRC POWER ON	OFF		AA232	S	LH2 RECIRC POWER ON	OFF	
AA259	S	APS BUS PWR ON	OFF		AA331	S	*4D141 SUPERVISION ON	OFF	TS
AA332	S	AUX HYD PUMP SWITCH ON	OFF		AA345	S	ENG START RELAY RESET	OFF	
AA355	S	STG PRES RDY FIRING COMM	OFF		AA370	S	ULL ROCKETS RELAYS RESET	OFF	
AAC166	S	RT TIMER 100 MILLISECOND	OFF		AB001	S	*4D30 LOAD ENABLE OFF	OFF	
AB002	S	*4D40 LOAD ENABLE OFF	OFF		AC018	S	INFLIGHT RELAYS ON	OFF	
AC023	S	FLIGHT MULTIPLEXER ON	OFF		AC028	S	FLT SCO GRP ON	OFF	
AC030	S	RC MEAS IND	OFF		AC111	S	HYDRAULIC SYSTEMS READY	OFF	
AC118	S	RSCR NO1 ON IND	OFF		AC127	S	S AND A UNIT ARMED	OFF	
AC129	S	RSCR NO2 ON IND	OFF		AC148	S	*4D31 NETWORK 1 OK	OFF	
AC149	S	*4D21 NETWORK 2 OK	OFF		AC150	S	*4D11 NETWORK 3 OK	OFF	
AC151	S	*4D41 NETWORK 4 OK	OFF		AC216	S	PU POWER ON INDICATION	OFF	
AC217	S	PU READY	OFF		AC295	S	APS 1 HE MIN L/O PRES IN	OFF	
AC300	S	APS 2 HE MIN L/O PRES IN	OFF		ADICGP	S	NOMENCLATURE UNKNOWN	OFF	
ADICHZ	S	NOMENCLATURE UNKNOWN	OFF		ADICIB	S	NOMENCLATURE UNKNOWN	OFF	
ADICID	S	NOMENCLATURE UNKNOWN	OFF		ADICIF	S	NOMENCLATURE UNKNOWN	OFF	
ADICIH	S	NOMENCLATURE UNKNOWN	OFF		ADICIJ	S	NOMENCLATURE UNKNOWN	OFF	
AEVMA	S	VOLT MON NETWRK 1-4D31	OFF		AEVMB	S	VOLT MON NETWRK 2-4D21	OFF	
AEVMC	S	VOLT MON NETWRK 3-4D11	OFF		AEVMD	S	VOLT MON NETWRK 4-4D41	OFF	
AFL05	S	LOX TANK MIN L/O PRES OK	OFF		AFL07	S	LH2 TANK MIN L/O PRESS O	OFF	

Sample List of Equipments by Panel Drawing - Output P

IGNITION SEQUENCER PANEL 1278A2		DRAWING NUMBER	PAGE	VEHICLE
ISBIN1	BINARY 1 TIME PULSE 2.5 MILLISECONDS	40M03480	012	201
ISBIN2	BINARY 2 TIME PULSE 5 MILLISECONDS	"	012	"
ISBIN3	BINARY 3 TIME PULSE 10 MILLISECONDS	"	013	"
ISBIN4	BINARY 4 TIME PULSE 20 MILLISECONDS	"	013	"
ISBIN5	BINARY 5 TIME PULSE 40 MILLISECONDS	"	013	"
ISBIN6	BINARY 6 TIME PULSE 80 MILLISECONDS	"	013	"
ISBIN7	BINARY 7 TIME PULSE 160 MILLISECONDS	"	013	"
ISCLK	IGNITION SEQUENCER CLOCK 400 PPS	"	012	"
ISDREG	REGULATED 28VDC IGNITION SEQUENCER	"	012	"
IS020V	REGULATED 20VDC IGNITION SEQUENCER	"	012	"
ISG1	CHANNEL 1 GATE	"	013	"
ISG2	CHANNEL 2 GATE	"	013	"
ISG3	CHANNEL 3 GATE	"	013	"
ISG4	CHANNEL 4 GATE	"	013	"
ISG5	CHANNEL 5 GATE	"	013	"
ISG9	CHANNEL 9 GATE	"	013	"
ISR1	CHANNEL 1 RELAY DRIVER	"	013	"
ISR2	CHANNEL 2 RELAY DRIVER	"	013	"
ISR3	CHANNEL 3 RELAY DRIVER	"	013	"
ISR4	CHANNEL 4 RELAY DRIVER	"	013	"
ISR5	CHANNEL 5 RELAY DRIVER	"	013	"
ISS01	IGNITION SEQUENCER SAFE-ARM SWITCH	"	011	"
ISS02	IGNITION SEQUENCER RESET SWITCH	"	011	"
ISS03	IGNITION SEQUENCER START SWITCH	"	011	"
ISS04	IGNITION SEQUENCER STOP SWITCH	"	011	"
IS001	IGNITION SEQUENCER ARM ENABLE	"	012	"
IS002	ALL BINARIES ZERO	"	012	"
IS003	IGNITION SEQUENCER PANEL RELAY K3	"	013	"
IS004	STOP ENABLE	"	011	"
IS005	IGNITION SEQUENCER START	"	011	"
IS005R	START RESET	"	011	"
IS006	IGNITION SEQUENCER CUTOFF INHIBIT	"	011	"
IS006R	CUTOFF INHIBIT RESET	"	011	"
IS007	IGNITION SEQUENCER STOP	"	011	"
IS007R	STOP RESET	"	011	"
IS008	ENGINES 5 & 7 IGNITION COMMAND	"	013	"
IS008R	IS008 RESET	"	013	"
IS009	ENGINES 6 & 8 IGNITION COMMAND	"	013	"
IS009R	IS009 RESET	"	013	"
IS010	ENGINES 2 & 4 IGNITION COMMAND	"	013	"
IS010R	IS010 RESET	"	013	"
IS011	ENGINES 1 & 3 IGNITION COMMAND	"	013	"
IS011R	IS011 RESET	"	013	"
IS012	CONT 5 OUTPUT RELAY SET POS K12	"	013	"
IS013	CONT 6 OUTPUT RELAY SET POS K13	"	013	"
IS014	CONT 7 OUTPUT RELAY SET POS K14	"	013	"
IS015	CONT 8 OUTPUT RELAY SET POS K15	"	013	"
IS016	IGNITION SEQUENCER DISARM ENABLE	"	013	"
IS016R	IS016 RESET	"	013	"

Sample List of Equipments by Drawing - Output Q

IGNITION SEQUENCER PANEL 1278A2			DRAWING NUMBER	PAGE	VEHICLE
ISBIN3	BINARY 3 TIME PULSE	10 MILLISECONDS	40M03480	013	201
ISBIN4	BINARY 4 TIME PULSE	20 MILLISECONDS	"	013	"
ISBIN5	BINARY 5 TIME PULSE	40 MILLISECONDS	"	013	"
ISBIN6	BINARY 6 TIME PULSE	80 MILLISECONDS	"	013	"
ISBIN7	BINARY 7 TIME PULSE	160 MILLISECONDS	"	013	"
ISG1	CHANNEL 1 GATE		"	013	"
ISG2	CHANNEL 2 GATE		"	013	"
ISG3	CHANNEL 3 GATE		"	013	"
ISG4	CHANNEL 4 GATE		"	013	"
ISG5	CHANNEL 5 GATE		"	013	"
ISG9	CHANNEL 9 GATE		"	013	"
ISR1	CHANNEL 1 RELAY DRIVER		"	013	"
ISR2	CHANNEL 2 RELAY DRIVER		"	013	"
ISR3	CHANNEL 3 RELAY DRIVER		"	013	"
ISR4	CHANNEL 4 RELAY DRIVER		"	013	"
ISR5	CHANNEL 5 RELAY DRIVER		"	013	"
IS003	IGNITION SEQUENCER PANEL RELAY K3		"	013	"
IS008	ENGINES 5 & 7 IGNITION COMMAND		"	013	"
IS008R	IS008 RESET		"	013	"
IS009	ENGINES 6 & 8 IGNITION COMMAND		"	013	"
IS009R	IS009 RESET		"	013	"
IS010	ENGINES 2 & 4 IGNITION COMMAND		"	013	"
IS010R	IS010 RESET		"	013	"
IS011	ENGINES 1 & 3 IGNITION COMMAND		"	013	"
IS011R	IS011 RESET		"	013	"
IS012	CONT 5 OUTPUT RELAY SET POS K12		"	013	"
IS013	CONT 6 OUTPUT RELAY SET POS K13		"	013	"
IS014	CONT 7 OUTPUT RELAY SET POS K14		"	013	"
IS015	CONT 8 OUTPUT RELAY SET POS K15		"	013	"
IS016	IGNITION SEQUENCER DISARM ENABLE		"	013	"
IS016R	IS016 RESET		"	013	"
IS017	CONT 1 OUTPUT RELAY SET POS K17		"	"	
IS018	IGNITION SEQUENCER 4-4 CUTOFF		"	"	
IS018R	IS018 RESET		"	"	
IS019	ENGINES 2 & 4 IG COMMAND K19		"	"	
IS020	ENGINES 1 & 3 IG COMMAND K20		"	"	
IS021	CONT 5 OUTPUT RELAY SET POS K21		"	"	
IS022	CONT 6 OUTPUT RELAY SET POS K22		"	"	
IS023	CONT 7 OUTPUT RELAY SET POS K23		"	"	
IS024	CONT 8 OUTPUT RELAY SET POS K24		"	"	
IS025	CONT 9 OUTPUT RELAY SET POS K25		"	"	

EXHIBIT XV

Sample End Item Approved Configuration Index - Output Y

SECTION I
NOMENCLATURE: TEST SET F/C SYSTEM | CONTRACTOR: NORTH AMERICAN | SPEC. NO: TS-1050A

ECP/FCR NUMBER	ECP/FCR TITLE	CAT	EFFECTIVITY RETROFIT FIRST	EFFECTIVITY RETROFIT LAST	EFFECTIVITY PRODUCTION FIRST	EFFECTIVITY PRODUCTION LAST	CCN NO.	INTER. DIR.	CCO/TCTO NO. *AGE/SV COMP.	SPARE	NEW PART NUMBER
NA-101	VERIFICATION OF SWITCHOVER AT LIFTOFF	C	001	003	N/R	N/R	43	214	MIK - EO 1936 *FROM SA-201 THRU SA-206	Y	424-2600-129

SEQUENCE OF ECP INCORPORATION SUCH AS:
A-PRIOR TO MATING AT LC
B-AFTER MATING AT LC
C-PRIOR TO NEXT SYSTEM TEST
ETC.

SERIAL NUMBER OF END ITEM

INTERFACE DIRECTIVE
NUMERIC CODE FOR PANEL AGREEMENT OR OTHER INTERFACE AUTHORITY TO INDICATE COORDINATION AND APPROVAL. NO ENTRY IF ECP HAS NO INTERFACE AFFECTS

CONTRACTOR RETROFIT CHANGE NUMBER
NO ENTRY IF ECP AFFECTS PRODUCTION ONLY
*INDICATES THAT MODIFIED END-ITEM WILL BE USED FOR SA 201 THRU 206

EFFECT OF ECP ON SPARES
Y-YES
N-NO

REPORT DATE: 30 MARCH 1964 | END ITEM: TS-2600 | PART NO. 424-2600-019

Sample Approved ECP (Change) End Item Index - Output Z

SECTION II
CONTRACTOR: NORTH AMERICAN

ECP/FCR NUMBER	AFFECTED END ITEMS							
NA- 95	CM-1053							
NA-101	CM-3625							
NA-101	TS -2600	GE-2310	GE-4250					
NA-101	TS -9601							

ECP 95 AFFECTS ONLY ONE END ITEM. ECP 101 AFFECTS THREE NAA END ITEMS AND INTERFACES WITH TWO GE END ITEMS. INTERFACE DIRECTIVE 214 DENOTED COORDINATION COMPLETED.

AS A RESULT OF ECP 101 TWO GE ITEMS WILL REQUIRE MODIFICATION.

GE SUBMISSION WOULD INCLUDE FOLLOWING ENTRY IN THIS SECTION:

GE-14	GE-2310	NA TS-2600						
GE-14	GE-4250	NA TS-2600						

G.E. ECP REQUIRED TO MODIFY END ITEMS 2310 AND 4250 IN ACCORDANCE WITH INTERFACE AGREEMENT ESTABLISHED BY INTERFACE DIRECTIVE 214

EXHIBIT XV

REPORT DATE: 30 MARCH 1964

EXHIBIT XV

Sample End Item Quantitative Requirements Schedule - Output AA

SECTION III
CONTRACTOR: NORTH AMERICAN

END ITEM NUMBER		PART NUMBER	SERIAL NUMBER	1 KSC	2 MICH.	3 MTF	4 MSC	5 MSFC	6 FACTORY/ DEPOT	7 IN TRANSIT.
TS-2600		424-2600-019	001						x	
TS-2600		424-2600-019	002	x						
TS-2600		424-2600-019	003	x						

SERIAL #001 OF END ITEM TS-2600 IS ALLOCATED FOR PLANT USE AT NAA WHILE SERIAL NO.'S 002 AND 003 ARE ALLOCATED TO KSC

REPORT DATE: 30 MARCH 1964

Sample End Item Modification Status - Output AB

SECTION IV
CONTRACTOR: NORTH AMERICAN

END ITEM IDENT.NO.	END ITEM PART NUMBER	SERIAL NUMBER	LOC'N.	ECP/FCR NUMBER	NEW END ITEM PART NUMBER	TYP	INCORP. DATES SCD.	ACT.	KIT IDENT.	INTERNAL CONT.NUMBER	TRN
TS-2600	424-2600-019	001	6	NA-101	424-2600-029	S	13123	11014	A	EO-1936-035	B
TS-2600	424-2600-019	002	1	NA-101	424-2600-029	S	19014	07024	A	EO-1936-035	B
TS-2600	424-2600-019	003	1	NA-101	424-2600-029	S	19014	14024	A	EO-1936-035	B

LOCATION WHERE ECP WILL BE INCORPORATED:
1-KSC
2-MICHOUD
3-MTF
ETC.

UPON INCORPORATION OF ECP NA-101 THE-019 CONFIGURATION OF THE END ITEM WILL BECOME THE -029

SCHEDULED AND ACTUAL INCORPORATION DATES OF ECP

TYPE OF ECP INCORPORATION
P-PRODUCTION
S-SERVICE (RETROFIT)

THIS IDENTIFIER DEFINES THE ACTUAL MODIFICATION KIT REQUIRED TO INCORPORATE ECP NA-101 INTO END ITEM TS-2600

EXHIBIT XV

REPORT DATE: 30 MARCH 1964

Sample Spares Status - Output AC

SECTION V
CONTRACTOR: NORTH AMERICAN

END ITEM IDENT. NO.	ECP/FCR NUMBER	SP/CHS E.I. SERIAL NO.	SEQ	OLD PART NUMBER	NEW PART NUMBER	TYP	INTERNAL CONTROL NO.	KIT IDENT	INCORP. DATES SCD	INCORP. DATES ACT.	LOC	TRN
TS-2600	NA-101	001	AA	424-2640-009	424-2640-019	S	EO-1936-036	B	19014	07024	6	B
TS-2600	NA-101	002	AB	424-2650-009	424-2650-019	S	EO-1936-037	C	13123	12014	1	B
TS-2600	NA-101	003	AC	424-2640-009	424-2640-019	S	EO-1936-036	B	19014		1	B

SERIAL NUMBER OF SPARE AFFECTED

SEQUENCE (SEQ) DEFINES THE TOTAL QUANTITY OF SPARES/ CHASSIS EFFECT.

IDENTIFIES THE CONFIGURATION OF THE SPARE TO BE MODIFIED

NEW PART NUMBER OF SPARE SUBSEQUENT TO MODIFICATION

IDENTIFICATION NO. OF MOD. KIT REQUIRED FOR INCORPORATION OF RETROFIT

FILE TRANSACTION FOR THIS SPECIFIC RECORD
A-ADDITION (INITIAL)
Z-DELETE (ENTIRE RECORD)
B THRU Y-CHANGES TO SPECIFIC FIELDS.

REPORT DATE: 30 MARCH 1964

EXHIBIT XV

APPENDIX E

COMPUTATIONAL ERRORS

E1 GENERAL

This appendix illustrates one method of verifying the accuracy of numerical results, such as the cross-multiplication of number series or the solutions of differential equations, by comparing the solutions of a difference equation and a differential equation to which it is equivalent. Two examples are given, an exponential and a sinusoid, as the solutions would be obtained on a Litton Digital Differential Analyzer. They are based on work done at the Hanford Laboratories, General Electric Company, Richland, Washington, in January 1960.

E2 INTRODUCTION

The Litton DDA uses the trapezoidal rule for integration. Formulas for determining the errors made in evaluating definite integrals exist and are not difficult to apply. But the solution of a differential equation involves a feedback, and it is not simple to infer the error over a period of time from that computed for a definite integral. The error can be determined, however, by setting up and solving the difference equations the DDA uses to approximate differential equations for a few simple cases.

E3 ERROR IN AN EXPONENTIAL

One of the simplest differential equations is that which determines an exponential:

$$\frac{dx}{dt} = \mu x \cdot \tag{E-1}$$

The solution of Equation E-1 is:

$$x(t) = x(0)e^{\mu t}, \tag{E-2}$$

where x(0) is the initial value of x(t).

The DDA solves Equation E-1 by integrating both sides:

$$x = c + \mu \int x \, dt \cdot \tag{E-3}$$

Let t increase by an amount h and express Equation E-3 as a definite integral:

$$x(t + h) - x(t) = \mu \int_{t}^{t+h} x(t)\, dt \, . \tag{E-4}$$

Now approximate the right side of Equation E-4 by using the trapezoidal rule to evaluate the definite integral:

$$x(t + h) - x(t) = \mu \left\{ \left(\frac{h}{2}\right) \left[x(t + h) + x(t) \right] \right\} . \tag{E-5}$$

Equation E-5 is that used in the DDA to approximate the solution of Equation E-1. It may be written in the form of a first-order linear difference equation, as follows:

$$(2 - h\mu)\, x(t + h) - (2 + h\mu)\, x(t) = 0 \, . \tag{E-6}$$

For a solution of Equation E-6, try

$$x(t) = c\rho^{t} \, . \tag{E-7}$$

Substituting Equation E-7 in Equation E-6:

$$(2 - h\mu)\, c\rho^{t+h} = (2 + h\mu)\, c\rho^{t} = 0 \, . \tag{E-8}$$

Dividing through by $c\rho^{t}$:

$$(2 - h\mu)\rho^{h} - (2\rho + h\mu) = 0 \tag{E-9}$$

$$\rho = \left(\frac{2 + h\mu}{2 - h\mu}\right)^{1/h} . \tag{E-10}$$

With Equation E-10 in Equation E-7:

$$x(t) = c\left(\frac{2 + h\mu}{2 - h\mu}\right)^{t/h} . \tag{E-11}$$

One point of difference between differential and difference equations is that the latter are defined only for discrete values of t, in this case, values which differ from each other by multiples of h. Hence in Equation E-11 the constant of integration, c, may have the value c_1 for $t = 0, h, 2h, 3h, \ldots$, but it may have the value c_2 for $t = h/2, 3h/2, 5h/2, \ldots$ and Equation E-11 still be a solution of Equation E-6. More generally, c is a periodic function of time with a period h. In most cases the periodicity of the constant of integration is of no practical significance, especially when h is small, as in DDA solutions (in an analysis of reactor kinetic equations, μh was equal to 2^{-15} units). In this appendix, c will be treated as a true constant.

Putting t = 0 in Equation E-11 gives

$$x(0) \quad = \quad c \tag{E-12}$$

and

$$x(t) \quad = \quad x(0)\left(\frac{2 \; + \; h\mu}{2 \; - \; h\mu}\right)^{t/h} . \tag{E-13}$$

To put Equation E-13 in a form easier to compare with Equation E-2, make the substitution

$$\frac{h\mu}{2} \quad = \quad \tanh\left(\frac{h\mu'}{2}\right) . \tag{E-14}$$

Using Equation E-14:

$$\begin{aligned} \frac{2 \; + \; h\mu}{2 \; - \; h\mu} \quad &= \quad \frac{1 \; + \; \left(\frac{h\mu}{2}\right)}{1 \; - \; \left(\frac{h\mu}{2}\right)} \\ &= \quad \frac{1 \; + \; \tanh\left(\frac{h\mu'}{2}\right)}{1 \; - \; \tanh\left(\frac{h\mu'}{2}\right)} \\ &= \quad \frac{\cosh\left(\frac{h\mu'}{2}\right) \; + \; \sinh\left(\frac{h\mu'}{2}\right)}{\cosh\left(\frac{h\mu'}{2}\right) \; - \; \sinh\left(\frac{h\mu'}{2}\right)} \\ &= \quad \frac{e^{\left(\frac{h\mu'}{2}\right)}}{e^{-\left(\frac{h\mu'}{2}\right)}} \quad = \quad e^{h\mu'} \end{aligned} \tag{E-15}$$

in which the properties and definitions of hyperbolic functions have been used. Finally, with Equation E-15 in Equation E-13:

$$x(t) = x(0)\, e^{\mu' t} . \tag{E-16}$$

Thus Equation E-16, the solution of the difference Equation E-6, is the same as Equation E-2, the solution of the corresponding differential equation, but with the growth constant, μ, replaced by μ'. From Equation E-14:

$$\mu' = \left(\frac{2}{h}\right) \tanh^{-1} \left(\frac{h\mu}{2}\right) . \tag{E-17}$$

Approximately:

$$\begin{aligned} \mu' &= \left(\frac{2}{h}\right)\left[\left(\frac{h\mu}{2}\right) + \frac{\left(\frac{h\mu}{2}\right)^3}{3} + \ldots\right] \\ &= \mu\left(1 + \frac{h^2\mu^2}{12} + \ldots\right) \end{aligned} \tag{E-18}$$

or

$$\mu' - \mu = \frac{h^2\mu^3}{12} + \ldots \tag{E-19}$$

The difference is negligible when h is small.

The analogy between Equation E-2 and Equation E-16 may be interpreted in two ways. First, write Equation E-16 as:

$$x(t) = x(0)\, e^{(\mu' - \mu)t}\, e^{\mu t} . \tag{E-20}$$

At any given time the magnitude of Equation E-20 is greater than that of Equation E-2 by a factor $e^{(\mu' - \mu)t}$, where $\mu' - \mu$ is given by Equation E-19, and is small with h small.

Second, write Equation E-16 as:

$$x(t) = x(0)\, e^{\mu t'} , \tag{E-21}$$

where, from Equation E-18,

$$
\begin{aligned}
t' &= \frac{\mu' t}{\mu} \\
&= t\left(1 + \frac{h^2\mu^2}{12} + \dots\right).
\end{aligned}
\qquad \text{(E-22)}
$$

With this interpretation, Equations E-2 and E-21 have the same magnitudes at different times, t and t', respectively. This is equivalent to plotting Equation E-2 and stretching the paper uniformly by an amount $1 + h^2\mu^2/12$ in the horizontal direction.

E4 ERROR IN A SINUSOID

The differential equation, the solution of which is a pure sine or cosine, can be written as a single second-order equation:

$$
\frac{d^2x}{dt^2} + \omega^2 x = 0 \qquad \text{(E-23a)}
$$

or as a pair of first-order equations:

$$
\begin{aligned}
\frac{dx}{dt} &= \omega z \\
\frac{dz}{dt} &= -\omega x
\end{aligned}
\qquad \text{(E-23b)}
$$

the solution of which is:

$$
\begin{aligned}
x &= c_1 \sin \omega t + c_2 \cos \omega t \\
z &= c_1 \cos \omega t - c_2 \sin \omega t \ .
\end{aligned}
\qquad \text{(E-24)}
$$

The constants of integration, c_1 and c_2, may be found by setting $t = 0$ in Equation E-24.

$$
\begin{aligned}
x(0) &= c_2 \\
z(0) &= c_1 \ .
\end{aligned}
\qquad \text{(E-25)}
$$

Putting Equation E-25 into E-24:

$$
\begin{aligned}
x(t) &= z(0) \sin \omega t + x(0) \cos \omega t \\
z(t) &= z(0) \cos \omega t - x(0) \sin \omega t .
\end{aligned}
\tag{E-26}
$$

To get the comparable DDA solution, write Equation E-23b in the form of integrals:

$$
\begin{aligned}
x(t + h) - x(t) &= \omega \int_{t}^{t+h} z(t)\, dt \\
z(t + h) - z(t) &= - \omega \int_{t}^{t+h} x(t)\, dt .
\end{aligned}
\tag{E-27}
$$

With the trapezoidal rule approximation to the right sides, Equation E-27 becomes:

$$
\begin{aligned}
x(t + h) - x(t) &= \left(\frac{h\omega}{2}\right) [z(t + h) + z(t)] \\
z(t + h) - z(t) &= -\left(\frac{h\omega}{2}\right) [x(t + h) + x(t)] .
\end{aligned}
\tag{E-28}
$$

To convert Equation E-28 into a single second-order difference equation, write a second set of equations for the interval $t + h$ to $t + 2h$:

$$
\begin{aligned}
x(t + 2h) - x(t + h) &= \left(\frac{h\omega}{2}\right) [z(t + 2h) + z(t + h)] \\
z(t + 2h) - z(t + h) &= -\left(\frac{h\omega}{2}\right) [x(t + 2h) + x(t + h)] .
\end{aligned}
\tag{E-29}
$$

Equations E-28 and E-29 constitute a set of four simultaneous equations in six unknowns: $x(t)$, $x(t + h)$, $x(t + 2h)$, $z(t)$, $z(t + h)$, and $z(t + 2h)$. From these any three of the unknowns may be eliminated to give a relation between the other three. Thus the z's may be eliminated to give the desired difference equation.

Rewriting Equations E-28 and E-29:

$$
\begin{aligned}
\left(\frac{h\omega}{2}\right)z(t) + \left(\frac{h\omega}{2}\right)z(t+h) \qquad\qquad & + [x(t) - x(t+h)] = 0 \\
-z(t) + z(t+h) \qquad\qquad & + \left(\frac{h\omega}{2}\right)[x(t) + x(t+h)] = 0 \\
\left(\frac{h\omega}{2}\right)z(t+h) + \left(\frac{h\omega}{2}\right)z(t+2h) & + [x(t+h) - x(t+2h)] = 0 \\
-z(t+h) + z(t+2h) & + \left(\frac{h\omega}{2}\right)[x(t+h) + x(t+2h)] = 0 \ .
\end{aligned}
\tag{E-30}
$$

To eliminate the z's, Equation E-30 may be treated as a set of four equations in four variables, $z(t)$, $z(t+h)$, $z(t+2h)$, and the coefficient of the last term on each left side. For the set to have a solution different from zero, the determinant of the coefficients must be zero. Thus:

$$
\begin{vmatrix}
\frac{h\omega}{2} & \frac{h\omega}{2} & 0 & [x(t) - x(t+h)] \\
-1 & 1 & 0 & \left(\frac{h\omega}{2}\right)[x(t) + x(t+h)] \\
0 & \frac{h\omega}{2} & \frac{h\omega}{2} & [x(t+h) - x(t+2h)] \\
0 & -1 & 1 & \left(\frac{h\omega}{2}\right)[x(t+h) + x(t+2h)]
\end{vmatrix} = 0 \ . \tag{E-31}
$$

To evaluate Equation E-31, first subtract the first and third columns from the second:

$$
\begin{vmatrix}
\frac{h\omega}{2} & 0 & 0 & [x(t) - x(t+h)] \\
-1 & 2 & 0 & \left(\frac{h\omega}{2}\right)[x(t) + x(t+h)] \\
0 & 0 & \frac{h\omega}{2} & [x(t+h) - x(t+2h)] \\
0 & -2 & 1 & \left(\frac{h\omega}{2}\right)[x(t+h) + x(t+2h)]
\end{vmatrix} = 0 \ . \tag{E-32}
$$

Next, add the second row of Equation E-32 to the fourth:

$$\begin{vmatrix} \frac{h\omega}{2} & 0 & 0 & [x(t) - x(t+h)] \\ -1 & 2 & 0 & \left(\frac{h\omega}{2}\right)[x(t) + x(t+h)] \\ 0 & 0 & \frac{h\omega}{2} & [x(t+h) - x(t+2h)] \\ -1 & 0 & 1 & \left(\frac{h\omega}{2}\right)[x(t) + 2x(t+h) + x(t+2h)] \end{vmatrix} = 0 . \tag{E-33}$$

Determinant E-32 is reduced to third order by expansion by minors of the second column. The resulting third-order determinant is evaluated by the usual rules.

$$2\begin{vmatrix} \frac{h\omega}{2} & 0 & [x(t) - x(t+h)] \\ 0 & \frac{h\omega}{2} & [x(t+h) - x(t+2h)] \\ -1 & 1 & \left(\frac{h\omega}{2}\right)[x(t) + 2x(t+h) + x(t+2h)] \end{vmatrix} = 0 . \tag{E-34}$$

$$2\left\{\left(\frac{h\omega}{2}\right)^3 [x(t) + 2x(t+h) + x(t+2h)] + \left(\frac{h\omega}{2}\right)[x(t) - x(t+h)] - \left(\frac{h\omega}{2}\right)[x(t+h) - x(t+2h)]\right\} = 0 . \tag{E-35}$$

$$(h\omega)\left\{\left(\frac{h\omega}{2}\right)^2 [x(t) + 2x(t+h) + x(t+2h)] + [x(t) - 2x(t+h) + x(t+2h)]\right\} = 0 . \tag{E-36}$$

$$\therefore \quad (4 + h^2\omega^2)\, x(t+2h) - 2(4 - h^2\omega^2)\, x(t+h) + (4 + h^2\omega^2)\, x(t) = 0. \tag{E-37}$$

Equation E-37 is the difference equation by which the DDA approximates Equation E-23a or Equation E-23b. For its solution try

$$x(t) = c\rho^t . \tag{E-38}$$

By substituting Equation E-38 in E-37, and following the same procedure as in solving Equation E-6, it is seen that

$$(4 + h^2\omega^2)\rho^{2h} - 2(4 - h^2\omega^2)\rho^h + (4 + h^2\omega^2) = 0 . \tag{E-39}$$

The solution of Equation E-39 gives two values of ρ:

$$\begin{aligned}
\rho^h &= \frac{(4 - h^2\omega^2) \pm \sqrt{(4 - h^2\omega^2)^2 - (4 + h^2\omega^2)^2}}{4 + h^2\omega^2} \\
&= \frac{4 - h^2\omega^2 \pm \sqrt{- 16h^2\omega^2}}{4 + h^2\omega^2} \\
&= \frac{4 - h^2\omega^2 \pm 4ih\omega}{4 + h^2\omega^2} \\
&= \frac{(2 \pm ih\omega)^2}{(2 + ih\omega)(2 - ih\omega)} ,
\end{aligned} \tag{E-40}$$

where, as usual, $i = \sqrt{-1}$. From Equation E-40 the two values of ρ are:

$$\rho_1 = \left(\frac{2 + ih\omega}{2 - ih\omega}\right)^{1/h} \tag{E-41}$$

$$\rho_2 = \left(\frac{2 - ih\omega}{2 + ih\omega}\right)^{1/h} . \tag{E-42}$$

Hence, with Equations E-41 and E-42 in Equation E-38:

$$x(t) = c_1'\left(\frac{2 + ih\omega}{2 - ih\omega}\right)^{t/h} + c_2'\left(\frac{2 - ih\omega}{2 + ih\omega}\right)^{t/h} . \tag{E-43}$$

z(t) may be found by solving Equation E-28 and substituting Equation E-43, as follows:

$$\left(\frac{h\omega}{2}\right)[z(t+h) + z(t)] = [x(t+h) - x(t)]$$

$$\left(\frac{h\omega}{2}\right)[z(t+h) - z(t)] = -\left(\frac{h\omega}{2}\right)^2 [x(t+h) + x(t)] \qquad \text{(E-28b)}$$

$$h\omega z(t) = \left[1 + \left(\frac{h\omega}{2}\right)^2\right] x(t+h) - \left[1 - \left(\frac{h\omega}{2}\right)^2\right] x(t) \; .$$

$$\begin{aligned} 4h\omega z(t) &= (4 + h^2\omega^2)\left[c_1'\left(\frac{2+ih\omega}{2-ih\omega}\right)^{(t+h)/h} + c_2'\left(\frac{2-ih\omega}{2+ih\omega}\right)^{(t+h)/h}\right] \\ &\quad - (4 - h^2\omega^2)\left[c_1'\left(\frac{2+ih\omega}{2-ih\omega}\right)^{t/h} + c_2'\left(\frac{2-ih\omega}{2+ih\omega}\right)^{t/h}\right] \\ &= (2+ih\omega)^2\, c_1'\left(\frac{2+ih\omega}{2-ih\omega}\right)^{t/h} + (2-ih\omega)^2\, c_2'\left(\frac{2-ih\omega}{2+ih\omega}\right)^{t/h} \\ &\quad - (4 - h^2\omega^2)\left[c_1'\left(\frac{2+ih\omega}{2-ih\omega}\right)^{t/h} + c_2'\left(\frac{2-ih\omega}{2+ih\omega}\right)^{t/h}\right] \; . \end{aligned}$$

$$\therefore \quad z(t) = i\left[c_1'\left(\frac{2+ih\omega}{2-ih\omega}\right)^{t/h} - c_2'\left(\frac{2-ih\omega}{2+ih\omega}\right)^{t/h}\right]. \qquad \text{(E-44)}$$

To put Equations E-43 and E-44 into a form more suitable for comparison with Equation E-24, let

$$\frac{h\omega}{2} = \tan\left(\frac{h\omega'}{2}\right) . \qquad \text{(E-45)}$$

Then:

$$\begin{aligned} \frac{2+ih\omega}{2-ih\omega} &= \frac{1 + i\tan\left(\frac{h\omega'}{2}\right)}{1 - i\tan\left(\frac{h\omega'}{2}\right)} = \frac{\cos\left(\frac{h\omega'}{2}\right) + i\sin\left(\frac{h\omega'}{2}\right)}{\cos\left(\frac{h\omega'}{2}\right) - i\sin\left(\frac{h\omega'}{2}\right)} \\ &= \frac{e^{\left(\frac{ih\omega'}{2}\right)}}{e^{-\left(\frac{ih\omega'}{2}\right)}} = e^{ih\omega'} \; . \end{aligned} \qquad \text{(E-46)}$$

With Equation E-46 in Equations E-43 and E-44:

$$
\begin{aligned}
x(t) &= c_1' e^{i\omega't} + c_2' e^{-i\omega't} \\
z(t) &= ic_1' e^{i\omega't} - ic_2' e^{-i\omega't} .
\end{aligned}
\tag{E-47}
$$

Or:

$$
\begin{aligned}
x(t) &= c_1'(\cos \omega t + i \sin \omega't) + c_2'(\cos \omega't - i \sin \omega't) \\
z(t) &= ic_1'(\cos \omega t + i \sin \omega't) - ic_2'(\cos \omega't - i \sin \omega't) .
\end{aligned}
\tag{E-48}
$$

In Equation E-48, put

$$
\begin{aligned}
c_1 &= i(c_1' - c_2') \\
c_2 &= c_1' + c_2' .
\end{aligned}
\tag{E-49}
$$

$$
\begin{aligned}
\therefore \quad x(t) &= c_1 \sin \omega't + c_2 \cos \omega't \\
z(t) &= c_1 \cos \omega't - c_2 \sin \omega't .
\end{aligned}
\tag{E-50}
$$

This is the same form as Equation E-24. Putting initial conditions into Equation E-50:

$$
\begin{aligned}
x(0) &= c_2 \\
z(0) &= c_1 .
\end{aligned}
\tag{E-51}
$$

Finally:

$$
\begin{aligned}
x(t) &= z(0) \sin \omega't + x(0) \cos \omega't \\
z(t) &= z(0) \cos \omega't - x(0) \sin \omega't .
\end{aligned}
\tag{E-52}
$$

Equations E-52 are the solutions of the difference Equation E-37. This is the output of the DDA as an approximate solution of Equation E-23. It is seen that the approximate solution (Equation E-52) is of the same form as the true solution (Equation E-24);

for the same initial conditions it is a sine wave of the same amplitude and phase, but with a slightly smaller frequency (ω' instead of ω). From Equation E-45:

$$\begin{aligned} \omega' &= \left(\frac{2}{h}\right)\tan^{-1}\left(\frac{h\omega}{2}\right) \\ &= \left(\frac{2}{h}\right)\left[\left(\frac{h\omega}{2}\right) - \frac{\left(\frac{h\omega}{2}\right)^3}{3} + \dots\right] \\ &= \omega\left[1 - \frac{h^2\omega^2}{12} + \dots\right]. \end{aligned} \tag{E-53}$$

From this relation it is evident that ω' differs from ω by a negligible amount when h is small, as it usually is.

In a manner similar to the case of the exponential, Equation E-52 can be made identical to Equation E-24 if it is plotted and the paper then shrunk uniformly by an amount $1 - h^2\omega^2/12$ in the horizontal direction.

E6 ERROR IN OTHER SOLUTIONS

The form of solutions of other linear differential equations with constant coefficients and their error can be found by the methods employed herein. For practical problems, the errors can undoubtedly be estimated closely enough from the equations of this appendix, since solutions of this type of equation are expressed in terms of exponential and sine functions.

The errors in solutions of other types of differential equations are not as easy to calculate, in general. It is reasonable to believe, however, that they can be estimated in most cases from the known errors in sinusoids and exponentials.

REFERENCES

1. C.B. Clarkson, The Inverse Laplace Transforms of a Rational Function, General Electric Company Report R62DSD2, Syracuse, New York, 5 January 1962.

2. J.G. Truxal, Automatic Feedback Control System Synthesis, McGraw-Hill Book Co., New York, 1955.

3. S.J. Mason and H.J. Zimmerman, Electronic Circuits, Signals, and Systems, John Wiley and Sons, New York, 1960.

4. E. Whittaker and G. Robinson, The Calculus of Observations, Blackie and Son Ltd., London and Glasgow, 1958.

5. C.L. McDearmont, B.H. Henson, and C.M. Webster, Dynasar Simulation of the Saturn V, F-1 Engine Propulsion, (65 SIMAR 82/8), Study report prepared for MSFC Reliability and Quality Assurance Laboratory.

6. H.H. Koelle, Handbook of Astronautical Engineering, McGraw-Hill Book Co., New York, 1961.

7. B. Lewis, R.N. Pease, H.S. Taylor, Combustion Processes, High Speed Aerodynamics, and Jet Propulsion, Volume 2, Princeton University Press, Princeton, N.J., 1956.

8. M.P. Marcus, Switching Circuits for Engineers, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1962.

9. A. Tustin, "A Method of Analyzing the Behavior of Linear Systems in Terms of Time Series," Journal of the Institution of Electrical Engineers, Vol. 94, Part II-A, No. 1, 1947, pp 130-142.

10. A. Madwed, Number Series Method of Solving Linear and Nonlinear Differential Equations, Instrumentation Laboratory Report No. 6445-T-26, 1950, Massachusetts Institute of Technology, Cambridge, Mass.

11. C.H. Weaver, The Analysis and Design of Linear Closed-Loop Control Systems by Means of Number Series, University of Wisconsin Ph.D. Thesis, 1955, Madison, Wisc.

12. H.M. Powell, A Number Series Method for Digital Computer Solution of Differential Equations, University of Tennessee M.S. Thesis, Dec. 1958, Knoxville, Tenn.

13. Saturn V, Composite Mechanical Schematic, George C. Marshall Space Flight Center Report 10M30531, 25 April 1965, Huntsville, Ala.

14. E.L. Ince, Ordinary Differential Equations, Dover Publications, Inc., New York, 1956.

15. Apollo Configuration Manual, NASA Document NPC-500-1, 18 May 1964, Washington, D.C.

16. Engineering Changes to Weapons Systems, Engineering and Facilities, Air Force-Navy Aeronautical (ANA) Bulletin 445, 12 July 1963, Washington, D.C.

17. T.R. Hoffman, Simplification Algorithm for SA-9 ESE Simulation Equations, General Electric Company Technical Report TR-MSFC/DB-3.18, Daytona Beach, Florida, 25 June 1965.

18. T.R. Hoffman, General Aspects of the Problem of Machine Assistance in Troubleshooting, Unpublished report to R. Habermann, Jr., 25 August 1965, General Electric Co., Daytona Beach, Florida.

19. E.E. Kivari, Performance Requirements for Application of the ESE Simulation to Fault Isolation Study, General Electric Company Technical Memorandum TR-MSFC/DB-3.8, 29 January 1965, Daytona Beach, Florida.

20. E.E. Kivari, Application of Fault Isolation Algorithm to SA-9 Post-Flight Data Analysis, General Electric Company Memorandum for File MF-MSFC/DB-3.30, 26 March 1965, Daytona Beach, Florida.

21. Final Reports AFSC-TR-65-1, 2, 3, 4, 5, 6, Weapon System Effectiveness Advisory Committee, January 1965, United States Air Force, Washington, D.C.