RESEARCH NOTE
RN 79-6

LEVEL II

A SAINT MODEL OF THE AN/TSQ-73 MISSILE MINDER: USER'S GUIDE

Pritsker & Associates, Inc.

Consultants in Systems Engineering

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RN 79-6

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): A SAINT Model for the AN/TSQ-73 Missile Minder: User's Guide

5. TYPE OF REPORT & PERIOD COVERED: Final Report 12/77 - 8/78

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): David B. Wortman, Alonzo F. Hixson, III

8. CONTRACT OR GRANT NUMBER(s): DAHC19-77-M-0031

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Pritsker & Associates, Inc., West Lafayette, Indiana 47906

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2Q762722A765

11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Research Institute, 5001 Eisenhower Avenue, Alexandria, VA 22333

12. REPORT DATE: August 1978

13. NUMBER OF PAGES: 188

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Final rept. Dec 77-Aug 78

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for open release; distribution unlimited.

12. 200 p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. ARI 19. RN-79-6

18. SUPPLEMENTARY NOTES

Research monitored technically by C.C. Jorgensen, ARI Field Unit at Ft Bliss, Texas.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

SAINT	man-machine systems
model	network modeling
air defense systems	simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SAINT, Systems Analysis of Integrated Networks of Tasks, is a network modeling and simulation technique developed to assist in the design and analysis of complex, man-machine systems. SAINT provides the concepts necessary to model systems that consist of tasks (discrete elements), state variables (continuous elements), and interactions between them. It facilitates the assessment of the contribution that system components make to overall system performance.

391 382.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

RESEARCH NOTE
RN 79-6

AD A068998

DDC FILE COPY

A SAINT MODEL OF THE AN/TSQ-73 MISSILE MINDER: USER'S GUIDE

Prepared by

David B. Wortman
Alonzo F. Hixson, III

Pritsker & Associates, Inc.
P.O. Box 2413
West Lafayette, Indiana 47906

for

U.S. Army Research Institute
for the Behavioral and Social Sciences
ARI Field Unit-Ft. Bliss
P.O. Box 6057
Ft. Bliss, Texas 79916

August 1978

OVERVIEW

This programmer guide consists of five major sections:

1. Introduction
2. Description of SAINT Tasks
3. Documentation of User Written Subprograms
4. Data Input Prodedures
5. Sample Simulation Outputs

The introduction describes the model purpose and background. A more detailed treatment of the issues and potential areas of application can be found in "A SAINT Model of the AN/TSQ-73 Guided Missile ARI Defense System," D. Wortman, A. Hixon, and C. Jorgensen. ARI Research Memorandum (in press) 1979.

Section two consists of a detailed description of the 88 tasks which represent the full range of system interactions required by a 25L system operator/repairman. The tasks can be broken down into six groups:

1. General operating procedures such as assigning a track
2. Automatic processing, e.g., system updating
3. Flight path processing such as track initiation
4. System clock functions
5. Fire unit procedures such as hold fire
6. Hooking procedures, e.g., moving a tab

Each of these groups is reproduced in a network drawing which presents source tasks, sink tasks, and information flows. The representation of each task is also given on page 125 in SAINT code. Support documentation describing the SAINT language can be found in the reference section. The networks which follow this introduction should be consulted as the model is studied.

Section three deals with user subprograms. In order to perform specialized data collection and generation, a series of FORTRAN routines were written. These include verbal descriptions of subprograms, functions, subroutines, user functions, and moderator functions. Each function is numbered and its purpose and variable requirements are presented. For example, on page 111, moderator function 8 is discussed. It is used to obtain operator trace printouts at the beginning of a task. If a programmer desired to understand where in the model it was used, he could proceed to a description of the tasks with moderator function 8 such as task 11 on page 12 (Press interrogate). He could then explore the next referenced task or, by examining the task network, follow each sequence of operator events through the model.

Section four describes the data input procedures. Input is broken into two parts, (1) the model code and (2) the mission requirements against which the model is to be evaluated. It is possible to quickly vary the operator modes, the type of hooking procedure, the number of fire units, the number of missiles, the kill probabilities, location of batteries, flight paths, speeds, and track status data.

Section five, the model output, includes three sets of information:

1. A SAINT model echo check
2. A mission input echo check
3. A mission output (trace) and statistical summary

A complete output for a sample run is presented which includes all input parameters, data cards, task descriptions, echo checks, and outputs. Included in the listings are task by task events and times, means and standard deviations and frequency and cumulative histograms on selected task variables.

To use this manual in the most effective manner, it is recommended that the programmer first study the SAINT support documentation. After familiarity with SAINT has been achieved, the next step should be to examine the network drawing of the model and the sample input and output (page 125). When an overview is firmly in mind, details on individual tasks can be obtained by reading each task description, the associated user functions, moderator functions, and resource descriptions.

This manual has been designed to permit a potential user to trace any task sequence of interest at any depth desired. To illustrate the procedure, an example will be given. Suppose the network drawing was examined for hooking procedures. The first network SAINT symbol shows a label of "type hook" task number 35. From the network drawing it is seen that the task uses two moderator functions number 8 and 10. Nine tasks lead into task 35. They are 10, 15, 18, 22, 26, 28, 29, 31, and 33. Three tasks follow task 35. They are numbers 36, 39, and 42. Suppose we wanted to know details about task 35. On page 25 of the manual a detailed verbal description is given in which system attribute 1 is mentioned. If we want to learn more about it we can turn to table III, where we find attribute 1 deals with operator branching status. In a similar manner, moderator function 10 (page 112) can be used to collect user statistics by calling subroutine UTMST. Since UTMST is not found in the user written subroutines, it must be a SAINT language routine. Details about UTMST could then be found in the documentation for the SAINT Simulation Program.

By using the cross referencing procedures described above, it is possible to study the AN/TSQ/73 model and rapidly identify the exact task or parameters within a task that might require modification. In terms of

its machine requirements, the model currently runs on an IBM 360-65 FORTRAN G compiler in about 330K non-overlayed. Considerable reductions in core usage are possible if overlaying is performed. Reference (6) is recommended for this task.

(NETWORK DRAWINGS go here - immediately after the Diversion)

Replace with OVERVIEW

SUMMARY

With the development of increasingly complex weapons systems, the Army community has become sensitive to the need for accurate assessment of the human performance components of such systems. Because the human operators are integrally tied to a dynamically changing weapons environment, it is becoming increasingly difficult to assess overall system performance without considering the human element. As a result, a computer simulation model of the AN/TSQ-73 Guided Missile Air Defense System operator/repairman was developed to demonstrate the capability to estimate human and other system performance measures using digital simulation. SAINT, a combined discrete/continuous network simulation language, was used as the vehicle for developing the model.

The SAINT model represents an efficient and effective approach to modeling and analyzing the performance of the AN/TSQ-73 Guided Missile Air Defense System operator/repairman. An expanded model can be used as a vehicle for evaluating the performance of the entire system, including all human elements.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

SECTION I

INTRODUCTION

AN/TSQ-73 System

Guided Missile Air Defense System AN/TSQ-73 is a lightweight mobile automatic data processing command and coordination system for Nike-Hercules and Hawk Army Air Defense units [1,2,3]. The AN/TSQ-73 integrates radar and identification friend or foe (IFF) data from local and remote radars for console display. Through programming of the automatic data processing equipment, alphanumerics, track and site symbols, map symbols, coordinates, and lines are generated. This data is integrated with radar and IFF data to provide the operator with a display of aircraft and missile targets identified by track symbols and alphanumerics, as well as site and map symbols identified by alphanumerics. Target data, fire unit profile data, and defended point characteristics are processed and analyzed automatically for primary and secondary fire unit selection and type of weapon assignment.

Tactical operation of the system is accomplished by one tactical directions officer and two operator/repairmen. They may operate the system in a number of manual or automatic modes. Some of the possible modes are:

1. Air Track Identification
    a. Automatic and sector scan
    b. Manual

2. Tracking
    a. Automatically initiated automatic tracking
    b. Manually initiated automatic tracking
3. Fire Unit Selection and Weapon Assignment
    a. Automatic, by computer-generated commands
    b. Semi-automatic, by operator acceptance of computer recommendations
    c. Manual, by operator

SAINT Simulation Language

SAINT, Systems Analysis of Integrated Networks of Tasks, is a network modeling and simulation technique developed to assist in the design and analysis of complex, man-machine systems. SAINT provides the concepts necessary to model systems that consist of tasks (discrete elements), state variables (continuous elements), and interactions between them [4,5,6]. It facilitates the assessment of the contribution that system components make to overall system performance.

SAINT has been designed, developed, and used for modeling and analyzing systems in which resources (men and machines) perform tasks subject to physical and environmental constraints. It satisfies the need for a graphical approach to the modeling and analysis of systems which contain procedural, risk and random elements. For engineers and human factors specialists, SAINT provides modeling capabilities similar to those provided by circuit diagrams for electrical engineers, signal flow graphs and block diagrams for systems analysts, and PERT/CPM networks

for project managers. Further, it provides automatic model analysis capabilities via the SAINT simulation program.

The SAINT philosophy is to separate the modeling process from the analysis process. A graphical approach to modeling is taken in which the system to be analyzed is represented by a network model. The network model facilitates communication regarding the characteristics of the system and also serves as the basis for subsequent system analysis.

A SAINT network model is developed using symbols contained in the SAINT symbol set. The fundamental elements of SAINT networks are tasks, resources (personnel and/or equipment) required to perform tasks, relationships among tasks, and system status variables referred to as state variables. System performance is related to which tasks are performed, the manner in which they are initiated, utilization of the system resources, and the extent to which certain states of the system are achieved or maintained.

AN/TSQ-73 Model

The AN/TSQ-73 model is designed to simulate the activities of an operator/repairman involved in monitoring and operating the display console during a simulated mission. Missions can be simulated in any of the operating modes identified in the first section. In addition, if the system becomes overloaded and the automatic procedures are delayed, the model has the capability of simulating the operator manually updating data even though the system is operating in an automatic mode.

The model of the AN/TSQ-73 system is divided into four submodels. The first submodel represents the operator/repairman's response to a given visual stimulus that is presented on the display console and the physical actions that result. Those procedures used by the operator to "hook" a track or site (i.e., focus the attention of the system on one particular item on the display) were given special attention. The second submodel controls the location and identification status of all aircraft involved in a mission. This submodel utilizes the continuous simulation capabilities of SAINT to maintain a current position on all aircraft. The third submodel represents the activities performed by the fire units once they have been assigned a target. The fourth submodel initiates the automatic engagement, reengagement, and cease fire commands that are generated by the system computer or the fire units themselves. The four submodels interact with each other on a continuing basis throughout the simulation. As a result, tasks are scheduled, interrupted, and rescheduled as a function of system status.

Input data for the model is divided into two categories. The SAINT model input cards are used to define the structure and parameters of the model described above. The second category of input data is used to configure the model for a particular mission scenario. This data incorporates the selection of automatic or manual operating modes, the number of aircraft and fire units to be simulated, and the movement and status of the aircraft over time. All model inputs are designed to allow the

user to evaluate a wide range of operating procedures and mission scenarios in a relatively simple and straightforward manner.

The first output provided by the simulation model is a complete echo check of all input data. This is followed by a detailed account of the operator's task-by-task performance and corresponding system status over simulated time. Finally, summary performance measures related to the operator, fire units, and the overall system are provided.

The model can be used to establish both operation and training procedures in a timely and cost-effective manner. In addition, the techniques and methods employed in this model can serve as the basis for the development of models of other operator-controlled systems with a minimal amount of time and effort.

SECTION II

DESCRIPTION OF TASKS OF THE SAINT MODEL

This section presents a detailed discussion of the activities and operating conditions represented by the SAINT model of the AN/TSQ-73 system. Each task will be fully defined. The discussion of the task characteristics are presented in the following order (as needed):

1. Represented activity or purpose of the task.
2. Conditions causing release of the task and its resource assignment.
3. Task performance time.
4. Moderator functions called.
5. Attribute values set.
6. Conditional branching decisions.
7. Task statistics collected.
8. User functions called.
9. Possible branching directions.

In additions, Tables I, II, and III appearing at the end of this section provide model references for and definitions of the distribution sets, values assoicated with visual symbols, and definitions of SAINT attributes used in these task descriptions. Further descriptions of the user-written subprograms referenced in this discussions are presented in Section III.

Task 1: Searching (Search)

This task represents the operator visually scanning the radar screen to determine on which track or site he will focus his attention next. This task is released at time zero. It may also be released upon the completion of other operator tasks.

Moderator function 1 is called by this task to determine what the next symbol to be processed will be. The function first assigns values to all the sites and tracks on the scope by calling subroutine SETV. These values reflect the level of visual stimulation that they present, and their relative importance on the scope. Using these values, the function randomly selects the next site or track to be processed. If none are selected, the operator will be idle for the next period of time. The task performance time for task 1 is also computed in this function. It is randomly based on the number of symbols that appear on the scope. The more objects on the scope, the shorter the task performance time will generally be. In addition, the function sets the value of system attribute 1 to the type of symbol or site that the operator will process and, through a call to subroutine SETTR, initializes the value of information attribute 1 to the track number if the object is a track or information attribute 2 to the fire unit number if the object is a fire unit.

Moderator function 8 is called to trace the actions of the operator. Moderator function 10 is called to collect statistics.

Branching from task 1 is based on system attribute 1 and is dependent on the type of object that will be processed. The possible branches from task 1 are:

system attribute 1 = 0 - idle time - task 2
system attribute 1 = 1 - video data - task 3
system attribute 1 = 2 - unknown track - task 9
system attribute 1 = 3 - friendly track - task 21
system attribute 1 $\geq$ 4 - other - task 24

Task 2: Idle Time (IDLETIME)

This task represents the time spent by the operator when no attention is being given to the radar system. It may be released upon the completion of task 1. The task performance time is determined by user function 40.

Moderator function 8 is called to trace the actions of the operator.

The task statistic, BET, STA is collected.

Upon completion of this task, the operator returns to task 1.

Task 3: Observing Video (OBSVDEO)

This task represents the observance and recognition of video data. It may be released upon the completion of task 1 or task 4 and requires resource 1. The parameters for the task performance time are stored in distribution set 4.

Moderator function 8 is called to trace the actions of the operator. Moderator function 10 is called to collect statistics.

The branching probabilities are set in system attributes 1, 2, and 3 by user function 2. The branching conditions for this task are based on the track initialization mode and the number of symbols on the display.

The task statistic BET, STA is collected.

The probabilistic branching may direct the operator to task 1, 4 or 5.

Task 4: Wait 1 Sweep (WAITONE)

This task represents the operator observing the radar screen for a period of time to determine if the video is indeed a valid track. It may be released upon the completion of task 3 and requires resource 1. The parameters for the task performance time are stored in distribution set 5.

Moderator function 9 is called to trace the actions of the operator. Upon completion of this task, the operator returns to task 3.

Task 5: Auto/Manual Initiation (AUTOMAN)

This task represents the decision made by the operator to manually initiate the track. It may be released upon the completion of task 3 and requires resource 1. The task performance time is 0.

The probabilities for branching are stored in system attributes 1, 2, and 3 by user function 2. The branching conditions for this task are based on the track initialization mode and the total number of tracks currently in this system.

Probabilistic branching directs the operator to task 1, 6, or 7.

Task 6: Watching Video (WATCHVID)

This task represents the operator watching the radar scope to determine if the video data is in fact a track. It may be released upon the completion of task 5 and requires resource 1. Task performance time is determined by user function 3. This time is characterized by the sweep time of the specific radar system used.

Moderator function 9 is called to trace the actions of the operator.

The operator checks to see if the video has in fact disappeared. If it has, user function 41 sets system attribute 1 to a value greater than 1. If the symbol is still video data, the function sets system attribute 1 to the value 1. These values are then used in the conditional branching.

The conditional branching directs the operator to task 1 if system attribute 1 is greater than 1 and to task 7 if system attribute 1 is equal to 1.

Task 7: Position Tab (POSTAB)

This task represents the operator positioning the tab on the data to be hooked. It may be released upon the completion of task 5 or 6 and requires resource 1. The parameters for the task performance time are stored in distribution set 6.

Moderator function 8 is called to trace the actions of the operator.

Upon completion of this task, the operator is directed to task 8.

Task 8: Press Initiate (PINIT)

This task represents the operator pressing the TASK FUNCTION - INITIATE button. It is released upon the completion of task 7 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator. Moderator function 9 is called for operator statistics.

User function 4 is called to record the status of the track.

Upon completion of this task, the operator is directed to task 1.

Task 9: Observing Unknown Track (OBSUNK)

This task represents the operator's observance and recognition of an unknown track. It may be released upon the completion of task 1 and requires resource 1. The parameters for the task performance time are stored in distribution set 4.

Moderator function 8 is called to trace the actions of the operator. Moderator function 10 is called to collect statistics.

The probabilities for branching are stored in system attributes 1 and 2 by user function 5. These probabilities depend on the auto/manual interrogation mode of the system and the range that the target is from the center of the system. The distance factor which is incorporated into the probabilities is stored in user-defined task characteristics.

The task statistic BET, STA is collected.

Upon completion of this task, probabilistic branching directs the operator to task 1 or 10.

Task 10: Press IDENT IFF (PIDIFF)

This task represents the operator pressing the TASK SELECTION - IDENT IFF button. It may be released upon the completion of task 9 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

System attribute 4 is set to the value 1, the return address when the operator has completed the hooking procedures which follow. System attribute 5 is set to the value 0, indicating to the hooking procedure that a track is being hooked.

Upon completion of this task, the operator is directed to task 35.

Task 11: Press Interrogate (PINTERRO)

This task represents the operator pressing the TASK FUNCTIONS - INTERROGATE button. It may be released upon the completion of task 45 when the hooking procedures were initiated from task 10 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

Upon completion of the task, the operator is directed to task 12.

Task 12: Read Message (READMSG)

This task represents the operator reading the results of the interrogation which appear in the hooked track data field. It is released upon the completion of task 11 and requires resource 1. The parameters for the task performance time are stored in distribution set 8.

Moderator function 8 is called to trace the actions of the operator.

The probabilities for branching are stored in system attributes 1, 2, and 3 by user function 6. These probabilities are based on the number of tracks in the system and the distance this track is from the center of the system. The distance factor, which is incorporated into the probabilities, is stored in user-defined task characteristics.

Upon completion of this task, probabilistic branching directs the operator to task 1, 13, or 14.

Task 13: Press Friend/Hostile (PFH)

This task represents the operator pressing the TASK FUNCTION - FRIEND/HOST button. It may be released upon completion of task 12 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator. Moderator function 9 is called to collect operator statistics.

Upon completion of this task, the operator is directed to task 1.

Task 14: Tight/Free Status (TIGHFREE)

This task represents the evaluation of the current operating policy as tight or free. It may be released upon the completion of task 12 and requires resource 1. The task performance time is 0.

Moderator function 8 is called to trace the actions of the operator. Moderator function 9 is called to collect operator statistics.

The probabilities for branching are stored in system attribute 1 and 2 by user function 7. These probabilities are based on the number of tracks currently in the system and whether the system is in the automatic engagement mode.

User function 7 also assigns to system attribute 7 the value of 0 if the system is free or the value 1 if the system is tight. These values will be used later for branching purposes.

Upon completion of this task, probabilistic branching directs the operator to task 1 or task 15.

Task 15: Press Assign (PASSIGN)

This task represents the operator pressing the TASK SELECTION - ASSIGN button. It may be released upon the completion of task 14 or task 25 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator. Moderator function 10 is called to collect operator statistics.

System attribute 4 is set to the value 2, the return address when the operator has completed the hooking procedures which follow. System attribute 5 is set to the value 0, indicating to the hooking procedure that a track is being hooked.

Branching from this task is conditional and based on the value stored in system attribute 7 which was set at task 14 or task 25. If the value of system attribute 7 is less than -.5, the branch is to task 35 for hooking purposes. If the value of system attribute 7 is less than 5., the branch is to task 18, indicating that the track was hooked following task 10.

Task 18: Branching Task (BRANCH)

The purpose of this task is to determine the method by which an assignment will be made. The possible methods are manual assignment of the track, semi-automatic assignment, or no assignment. The choice not to use the automatic assignment was made at task 12 or task 25. This task may be released upon the completion of task 15 or 45 and requires resource 1. The task performance time is 0.

The probabilities for branching are stored in system attributes 1, 2, and 3 by user function 10. These reflect operating procedures. Information attribute 2 is set by user function 42. This represents the fire unit number that is being assigned to the track. A value of 0 would indicate that no fire unit was assigned to the track. In addition, system attribute 4 is set to the value 3, the return address from the hooking procedure. System attribute 5 is set to the

value 1 indicating to the hooking procedure that a fire unit is being hooked.

Upon completion of this task, probabilistic branching directs the operator to task 35, 19 or 1.

Task 19: Press Engage/Accept (PENGACC)

This task represents the operator pressing the TASK FUNCTION - ACCEPT RECMD ASSIGN button or pressing the TASK FUNCTION - ENGAGE button. It may be released by task 18 or by task 45 if hooking procedures were used. This task requires resource 1.

Moderator function 8 is called to trace the actions of the operator.

Information attribute 3 is set to 2 which will be used by the fire unit section to identify the information as an engagement message.

Upon completion of this task, a conditional branching accomplishes two actions. First, if the value of system attribute 7 is less than .5, a branch is made to task 1 indicating that the operator returns to search the scope. If the value of system attribute 7 is greater than .5, a branch to task 20 is made indicating that the operator must press the hold fire button. Second, if the value of information attribute 2 is greater than 0, a branch to task 46 is made. This information is a message to the fire units that a track assignment is being made.

Task 20: Press Hold Fire (PHODF)

This task represents the operator pressing the SYSTEM MODE - HOLD FIRE button. It may be released upon the completion of task 19 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

Information attribute 3 is set to the value 3 to indicate to the fire unit section that the information is a hold fire message.

Upon completion, the operator is directed to task 1 and a message is sent to the fire unit section that will contain the hold fire status.

Task 21: Observing Friendly Track (OBSFREND)

This task represents the operator's observance and recognition of a friendly track. It may be released upon the completion of task 1 and requires resource 1. The parameters for the task performance time are stored in distribution set 4.

Moderator function 8 is called to trace the actions of the operator. Moderator function 10 is called to collect operator statistics.

The probabilities for branching are stored in system attribute 1 and 2 by user function 11. The possibility that this track was formerly classified as a hostile track is the key factor in determining these probabilities.

The task statistic BET, STA is collected.

Upon completion of this task, probabilistic branching directs the operator to task 1 or 22.

Task 22: Check for Fire Unit (CKFU)

This task represents the operator checking the display for a possible ongoing engagement. It may be released upon the completion of task 21 and requires resource 1. The parameters for the task performance time are stored in distribution set 4.

Moderator function 8 is called to trace the actions of the operator.

The probabilities for branching are stored in system attribute 1 and 2 by user function 12. The fact that a fire unit has been assigned to the track is the factor in determining these probabilities. In addition, system attribute 4 is set to the value 7, the return address from the hooking procedures, and system attribute 5 is set to 0, indicating to the hooking procedure that a track is being hooked.

Upon completion of this task, probabilistic branching directs the operator to task 1 or 35.

Task 23: Press Cease Fire (PCFIRE)

This task represents the operator pressing the SYSTEM MODE - CEASE FIRE button. It may be released upon the completion of task 45 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

Information attribute 3 is set to the value 4 indicating to the fire unit section that the information is a cease fire message.

Upon completion of this task, the operator is directed to task 1 and a message is sent to task 46.

Task 24: Search Branch B (SEARCHB)

The purpose of this task is to continue the branching originating from task 1. It may be released upon the completion of task 1. The task performance time is 0.

The possible branchings are:

system attribute 1 = 4 - task 25
system attribute 1 = 5 - task 28

Task 25: Observe Hostile Track (OBSHOST)

This task represents the operator's observance and recognition of a hostile track. It may be released upon the completion of task 24 and requires resource 1. The parameters for the task performance time are stored in distribution set 9.

Moderator function 8 is called to trace the actions of the operator. Moderator function 10 is called to collect statistics.

The probabilities for branching are stored in system attributes 1, 2, and 3 by user function 13. The distance the track is from the center of the system and the fact that the system is in the automatic engagement mode are used to determine these probabilities. The data used in determining the effect

of the range factor is stored in the user-defined task characteristics. In addition, system attribute 7 is set to the value -1. This is used by task 15 and task 19 to determine the branching of the operator at that time.

The task statistic BET, STA is collected.

Upon completion of this task, probabilistic branching directs the operator to task 1, 15 or 26.

Task 26: Press Assign (PASSIGN)

This task represents the operator pressing the TASK SELECTION - ASSIGN button. It may be released upon the completion of task 25 or 27 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

System attribute 4 is set to the value 5, the return address from the hooking procedures. System attribute 5 is set to the value 0, indicating to the hooking procedure that a track is being hooked.

Upon completion of this task, the operator is directed to task 35.

Task 27: Clear Hold Fire (CLEARHF)

This task represents the operator pressing the TASK FUNCTION - CLEAR HOLD FIRE button. It may be released upon the completion of task 45 and requires resource 1. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

User function 14 is called to record the associated fire unit number. The task also assigns to information attribute 3 the value 5, which indicates to the fire unit section that the information is a clear hold fire message.

Upon completion of this task, the operator is directed to task 1 or 26 and a message is sent to task 46.

Task 28: Observing Fire Unit (OBFU)

This task represents the operator's observance and recognition of a fire unit symbol. It may be released upon the completion of task 24. The parameters for the task performance time are stored in distribution set 4.

Moderator function 8 is called to trace the actions of the operator. Moderator function 10 is called to collect statistics.

The probabilities for branching are stored in system attributes 1, 2, and 3 by user function 15. This function checks the fire unit symbol for a blinking condition. The task also assigns to system attribute 4 the value 6, the return address from the hooking procedures and assigns to system attribute 5 the value 1, indicating that the hooking procedures will hook a fire unit.

The task statistic BET, STA is collected.

Upon completion of this task, probabilistic branching directs the operator to task 1, 35 or 33.

Task 29: Read Fire Unit AN Block (READOOAC)

This task represents the operator reading information from the fire unit AN block. It may be released upon the completion of task 45. The parameters for the task performance time are stored in distribution set 9.

Moderator function 8 is called to trace the actions of the operator.

User function 16 is called to set the value of system attribute 1. This value is then used for the conditional branching. The assignment is based on whether the fire unit has tracks currently attached. A value of 4 is assigned to information attribute 3; this will be used by the fire unit section to indicate that this track should be dropped. System attribute 5 is set to the value 0, indicating to the hooking procedures that a track is being hooked. System attribute 4 is set to the value 4, the return address when the operator has completed the hooking procedures.

Upon completion of this task, conditional branching directs the operator to task 30 if system attribute is equal to 0. This indicates that no tracks were assigned to the fire unit. The operator is directed to task 35 if the value of system attribute 1 is equal to 1. This indicates that there was only a primary track assigned to the fire unit. The operator is directed to task 31 if the value of system attribute 1 is equal to 2. This indicates that there was both a primary and secondary assignment made to this fire unit.

Task 30: Drop A Site (DROPSITE)

This task represents the operator pressing the TASK FUNCTION - DROP button. It may be released upon the completion of task 29. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

The task calls user function 17 to update the status of the tracks.

Upon completion of the task, the operator is directed to task 1.

Task 31: Clear Secondary Assignment (C2ASSIGN)

This task represents the operator recognizing a secondary assignment and initiating action to clear the secondary assignment. It may be released upon the completion of task 29.

Moderator function 8 is called to trace the actions of the operator.

System attribute 3 is set to the value 4, indicating to the fire unit section to update their status.

Upon completion of this task, the operator is directed to task 35 and a message is sent to task 46.

Task 32: Clear Primary Assignment (C1ASSIGN)

This task represents the operator recognizing and initiating steps to clear a primary assignment. It may be released upon the completion of task 45. The parameters for the task performance time is stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

User function 19 is called to update the status arrays. Information attribute 3 is set to the value 4, indicating to the fire unit section to update their arrays and check for a secondary assignment.

Upon completion of this task, the operator is directed to task 30 and a message is sent to task 46.

Task 33: Observing DDG (OBSDDG)

This task represents the operator observing and evaluating the data contained on the DDG. It may be released upon the completion of task 28 or 34. The parameters for the task performance time are stored in distribution set 9.

Moderator function 8 is called to trace the actions of the operator.

The probabilities for branching are stored in system attributes 1 and 2 by user function 8. These probabilities are based on the status of all fire units. If there is an effective status showing for a fire unit, the probabilities will exhibit a tendency toward directing the operator to clear the status. System attribute 4 is set to the value 8, the return address from the hooking procedures. System attribute 5 is set to the value 1, indicating to the hooking procedure that a fire unit is being hooked.

Upon completion of this task, the operator is directed to task 35 or 1.

Task 34: Clear Effective Status (PCLEFF)

This task represents the operator pressing the TASK FUNCTION - CLEAR EFFECT button. It may be released upon the completion of task 45. The parameters for the task performance time are stored in distribution set 7.

Moderator function 8 is called to trace the actions of the operator.

User function 9 is called to initialize the fire unit message. A value of 1 is assigned to information attribute 3 to indicate that this message is a clear effective message.

Upon completion of this task, the operator is directed to task 33 and a message is sent to task 46.

Task 35: Type of Hook (TYPEHOOK)

The purpose of this task is to determine what type of hooking procedures will be used. It may be released upon the completion of task 10, 22, 15, 18, 26, 28, 29, 31 or 33. The task performance time is 0.

Moderator function 8 is called to trace the actions of the operator. Moderator function 10 is called to collect operator statistics.

The value of system attribute 1 is set by user function 20. This function determines if a sequence hook, a tab hook, or a location/position hook will be used. This decision is based on the type of symbol being hooked and the hooking policy specified at input.

A task statistical MARK is set.

Upon completion of this task, the operator is directed to task 36, 39 or 42.

Task 36: Type of Sequence (TYPESEQ)

This task represents the decision whether or not a new special category must be entered through the keyboard. It may be released upon the completion of task 35 and requires resource 2. The paramters for the task performance time are stored in distribution set 7.

The value of system attribute 1 is assigned by user function 21. This function determines whether the category matches the type of symbol being hooked.

Upon completion of this task, the operator is directed to task 37 if the value of system attribute 1 is equal to 0, indicating that a new category needs to be entered or the operator is directed to task 38 if the value of system attribute 1 equals 1, indicating that the category does not need to be changed.

Task 37: Enter Category (ENTCATSQ)

This task represents the operator entering a new hooking category on the keyboard. It may be released upon the completion of task 36 and requires resource 2. The parameters for the task performance time are stored in distribution set 11.

User function 22 is called to update the status of the category.

Upon completion of this task, the operator is directed to task 38.

Task 38: Press Sequence Hook (PSEQHOOK)

This task represents the operator pressing the TASK FUNCTION - SEQ HOOK button. It may be released upon the completion of task 36, 37, or 38 and requires resource 2. The parameters for the task performance time are stored in distribution set 7.

A value is assigned to system attribute 1 by user function 23. This function determines if the item that was hooked by pressing the sequence hook button is the desired item.

Upon completion of this task, the operator may be directed to task 38 if the value of system attribute 1 equals 0, indicating that the item is not the desired item; to task 1 if the value of system attribute 1 is equal to 1, indicating that there are no items of the given type; or the operator may be directed to task 45 if the value of system attribute 1 is equal to 2, indicating that the desired item has been found.

Task 39: Enter Number/Position (ENTNUM)

This task represents the operator entering the track number, fire unit or site address through the AN keyboard, or the operator entering the GEOREF coordinates of the symbols through the AN keyboard. It may be released upon the completion of task 35 or 41 and requires resource 2. The parameters for the task performance time are stored in distribution set 11.

Upon completion of this task, the operator is directed to task 40.

Task 40: Press Number Hook (PNUMHOOK)

This task represents the operator pressing the TASK FUNCTION - NUMBER HOOK button or the operator pressing the TASK FUNCTION - POSN ENTRY button. It may be released upon the completion of task 39 and requires resource 2. The parameters for the task performance time are stored in distribution set 7.

Upon completion of this task, probabilistic branching directs the operator to task 41, indicating that an error has been made in the entry or to task 45 to return to the completion of his tasks.

Task 41: Press Dehook (PDEHOOK)

This task represents the operator pressing the TASK FUNCTION - DE HOOK button. It may be released upon the completion of task 40 and requires resource 2. The parameters for the task performance time are stored in distribution set 7.

Upon completion of this task, the operator is directed to task 39.

Task 42: Removing the Tab (MOVETAB)

This task represents the positioning of the tab on the symbol to be hooked. It may be released upon the completion of task 35 or 44 and requires resource 2. The parameters for the task performance time are stored in distribution set 6.

Upon completion of this task, the operator is directed to task 43.

Task 43: Position Hook (PSNHOOK)

This task represents the operator pressing the TASK FUNCTION - POSN HOOK button. It may be released upon the completion of task 42 and requires resource 2. The parameters for the task performance time are stored in distribution set 7.

Upon completion of this task, the probabilistic branching directs the operator to task 44, indicating that an error has been made in position hook or to task 45 directing the operator to the completion of his tasks.

Task 44: Press Dehook (PDEHOOK)

This task simulates the operator pressing the TASK FUNCTION - DE HOOK button. It may be released upon the completion of task 43 and requires resource 2. The parameters for the task performance time are stored in distribution set 7.

Upon completion of this task, the operator is directed to task 42.

Task 45: Return from Hook (RETHOOK)

This task directs the operator to the sequence of events he left to enter the hooking procedures section. It may be released upon the completion of task 38, 40 or 43. The task performance time is 0.

Moderator function 10 is called to collect statistics.

Branching from this task is based on system attribute 4 which was previously set before branching to the hooking procedures. The possible branchings are:

system attribute 4 = 1 - task 11
system attribute 4 = 2 - task 18
system attribute 4 = 3 - task 19
system attribute 4 = 4 - task 32
system attribute 4 > 4 - task 70

Task 46: Fire Unit Router (FUROUTER)

The purpose of this task is to send the fire unit message to the appropriate task for execution. It may be released upon the completion of task 19, 20, 23, 27, 31, 32, 34, 63 or 64. The task performance time is 0.

Moderator function 11 is called to collect fire unit statistics.

Branching from this task is based on information attribute 3, which was set in the operator task section. The possible branches are:

information attribute 3 = 1 - clear effective status - task 53
information attribute 3 = 2 - engage track - task 47
information attribute 3 = 3 - hold fire - task 54
information attribute 3 = 4 - cease fire - task 57
information attribute 3 > 4 - other - task 59

Task 47: Attaching Fire Unit (ATTACH)

This task represents the fire unit receiving an engagement message and the initial steps taken by the fire unit to attach the track to their fire unit. It may be released upon the completion of task 46. The parameters for the task performance time are stored in distribution set 12.

Moderator function 3 is called to initialize these activities.

Branching is based on system attribute 8 which is set by user function 24. The value is based on whether or not a cease fire message has been received.

Upon completion of this task, the fire unit is directed to task 48 if the value of system attribute 8 is equal to 1, indicating that the engagement should continue or the fire unit

is directed to task 83 if the value of system attribute 8 is equal to 0, indicating that the engagement should be terminated.

Task 48: Engagement Part A (ENGAGEA)

This task represents the initial tracking activities of the fire unit on the assigned track. It may be released upon the completion of task 47. The parameters for the task performance time are stored in distribution set 13.

The conditional branching from this task is governed by system attribute 8 which is set by user function 25. This function checks to see if the fire unit received a cease fire message, if the track is a primary or secondary assignment, or if the track is within the firing range.

The possible branchings upon completion of the task are:

system attribute 8 > 0 - continue engagement - task 49
system attribute 8 > -1 - out of range - task 84
system attribute 8 > -2 - secondary track - task 85
system attribute 8 > -3 - cease engagement - task 83

Task 49: Engagement Part B (ENGAGEB)

This task represents the final portion of the tracking process. It may be released upon the completion of task 48, 51, 53, 55 or 58. The parameters for the task performance time are stored in distribution set 1.

Moderator function 4 is called to update fire unit status. Moderator function 11 is called to collect statistics.

Branching is based on system attribute 8 which is set by user function 26. This function checks to see if the fire unit has received a cease fire or hold fire message. Upon completion

of this task, the possible branches are:

system attribute 8 > 0 - fire - task 50
system attribute 8 > -1 - hold fire - task 86
system attribute 8 > -2 - cease fire - task 83

Task 50: Fire (FIRE)

This task represents the fire unit firing a missile at the target. It may be released upon the completion of task 49. The parameters for the task performance time are stored in distribution set 14.

Moderator function 2 is called to collect statistics.

Upon completion of this task, the fire unit proceeds to task 51.

Task 51: Evaluation of Firing (EVALFIRE)

This task controls the evaluation and branching that results from the evaluation of a firing of a missile at a target. It may be released upon the completion of task 50. The task performance time is 0.

Moderator function 12 is called to collect statistics.

Branching is based on system attribute 8 which is set by user function 27. This function checks to see if the missile destroyed the target. If it did not, it further checks to see if the target is still within the range and if the fire unit still has weapons. The conditional branching for this task directs the fire unit to:

system attribute 8 > 1 - effective, ineffective out of range: check for secondary track - task 53
system attribute 8 > 0 - ineffective within range - task 49
system attribute 8 >-1 - out of missiles - task 74

Task 53: Check for Secondary Track (CKFOR2)

This task represents the fire unit checking for a secondary track to continue with an engagement. It may be released upon the completion of task 46, 51 or 57. The task performance time is 0.

Branching is based on system attribute 8 which is set by user function 28. This function updates both the fire unit and track status. It also checks for a secondary assignment.

Upon completion of this task, the fire unit is directed to task 49 if system attribute 8 is greater than 0, indicating there is a secondary assignment to engage, or the fire unit is directed to task 88 if system attribute 8 is greater than -1, indicating that the fire unit should return to an unused status.

Task 54: Hold Fire (HOLDFIRE)

This task represents the fire unit receiving a hold fire message. It may be released upon the completion of task 46. The task performance time is 0.

User function 29 is called to update the fire unit status and acknowledge the receipt of the hold fire message.

There is no branching from this task.

Task 55: Clear Hold Fire (CLEARHF)

This task represents the fire unit receiving a clear hold fire message and its resulting activity. It may be released upon the completion of task 59. The task performance time is 0.

The resulting action may involve the reengagement of a track if a hold fire message had been received. If the fire unit is not currently holding fire, no action is taken.

Branching is based on system attribute 8 which is set by user function 30. This function checks to see if a clear hold fire message has been received.

Upon completion of this task, the fire unit is directed to task 50 if system attribute 8 is greater than 0, indicating there had been a hold fire message or to task 87 if system attribute 8 is greater than -1, indicating no action was necessary.

Task 57: Cease Fire Message (CEASEF)

This task represents the fire unit receiving a cease fire message. It may be released upon the completion of task 46. The task performance time is 0.

Three possible actions may result from a cease fire message. First, the fire unit may simply cease fire on a secondary track. Second, the fire unit may cease fire on a primary track, then proceed to check for a secondary track to engage. Third, the fire unit may not be engaging any track in which case no action is taken.

Branching is based on system attribute 8 which is set by user function 31. This function checks for the appropriate conditions and sets the appropriate value to system attribute 8.

Upon completion of this task, the fire unit is directed to task 53 if system attribute 8 is equal to 0, indicating that a secondary target exists and an engagement should be made on that track or to task 87 if system attribute 8 is equal to 1, indicating that no further action is necessary.

Task 58: In Range (INRANGE)

This task represents the fire unit's determining that a target that was out of range now is in range; at this time the engagement will continue. It may be released upon the completion of task 59. The task performance time is 0.

Upon completion of this task, the message to reengage the track is forwarded to task 49.

Task 59: Fire Unit Router B (FUROUTB)

This task continues the branching originated in task 46. It may be released upon the completion of task 46. The task performance time is 0.

The possibilities for branching on the completion of this task are:

information attribute 3 = 5 - clear hold fire - task 55
information attribute 3 = 6 - in range - task 58

Task 61: Update Auto (UDAUTO)

This task is a system task and is used to initiate automatic updating of the track and fire unit status. This is a source task and is released at time 0. It may also be released upon the completion of task 63. The task performance time is 0.

Moderator function 10 is called for fire unit statistics.

User function 32 is called to initiate the counter used by task 63.

Upon completion of this task, the system is directed to task 62.

Task 62: Range Timer (RANGETIM)

This task is used for system control to regulate the frequency with which the automatic updates are made. It may be released upon the completion of task 61.

Upon completion of this task, the system is directed to task 63.

Task 63: Automatic Update (AUTOUD)

This task is used for system control to update the system status in four areas. It may be released upon the completion of task 62 or 63. The task performance time is 0.

First, it checks to see if a friendly track has been engaged. If it has, it sends a cease fire message. Second, it checks for tracks that are being held by a fire unit because they are out of firing range. If it is within firing range, it reinitiates the engagement. Third, it checks on those engaged tracks that are under a hold fire order. If the track has changed to a hostile target, it reengages the track. If the unknown track has come within a specified range and the system is under free attack policy, the target is reengaged. Fourth, if the target is not engaged and it has come within range and it is either unknown or hostile, it may be engaged. These checks are made for one track every time this task is executed.

The branching for this task is determined by system attribute 6 which is set by user function 33. This function checks for each of the conditions described above and sets the branching and status variables accordingly.

Each time a change is found, a message is sent to the fire unit section and a signal is sent to reexecute this task. This is continued until no changes have been made at which tiem a signal is sent to task 61 and the automatic cycle continues. In addition, a signal is sent to task 75 if an unassigned track has been engaged by a fire unit. Also, if the track was an unknown track and it was engaged under the tight engagement condition, a hold fire message will automatically be sent to the fire unit by task 64. The possible branches from this task are:

system attribute 6 $\leq$ 1 - task 63
system attribute 6 $\leq$ 1 - task 46
system attribute 6 $\leq$ 0 - task 64
system attribute 6 > 1 - task 61
system attribute 10 > 0 - task 75

Task 64: Automatic Hold Fire (AUTOHF)

This task is used for system control to send a hold fire message to the fire units when an unknown track has been engaged with the tight engagement policy in effect. It may be released upon the completion of task 63. The task performance time is 0.

The value of information attribute 3 is set to 3 to indicate to the fire unit that this is a hold fire message.

Upon completion of this task, the message is sent to task 46.

Task 65: Start Tracks (STTRACK)

This task is used for system control and generates the information packet used to control the aircraft flight. The task is released once for each track that will be flown during

the mission. This is a source task and is released at time 0. It may also be released upon the completion of itself. The task performance time is 0.

Branching is based on system attribute 9 which is set by user function 34. This function counts the number of tracks that are to be used and initiates the information packet associated with each aircraft. It sets the value of system attribute 9 to 0 if there are more planes to be generated. After all planes have been generated, the value of system attribute 9 is set to 1.

Upon the completion of this task, the system is directed to task 66 or back to itself.

Task 66: Initiate Tracks (INITRAK)

This task is used for system control to initiate the tracks at the time they are scheduled to appear on the scope. It may be released upon the completion of task 65. The task performance time is set by moderator function 5.

User function 35 is called at the completion of this task to initiate the SS variables and the track status.

Upon completion of this task, the system is directed to task 67 and 68.

Task 67: Route Update (ROUTUD)

This task is used for system control to update the flight path of the aircraft. It is released at each turning point for all flight paths and upon the comletion of task 66 and itself.

The task performance time is set to the time of the next turning point by user function 38.

The status of the flight is updated by a call to user function 36.

Upon completion of this task, the system is directed to itself, task 67.

Task 68: Status Update (STATUD)

This task is used for system control to update the identification status of the aircraft. It may be released upon the completion of task 66 and itself. The task performance time is set by user function 39 so that the task is completed each time a status update is necessary.

The status of the tracks is updated by a call to user function 37.

Moderator function 10 is called to collect statistics on this update.

Upon completion of this task, the system is directed to task 68. It may also be directed to task 75, if the change produced by this task is reflected on the scope. This is accomplished by branching to task 68 if the value of system attribute 10 is less than or equal to 1 and by branching to task 75 if the value of system attribute 10 is greater than 0.

Task 70: Return from Hooking B (RHOOKB)

This task is used to return the operator to the standard sequence of events. It may be released upon the completion of task 45. The task performance time is 0.

The branching from this task is controlled by system attribute 4. The possible branchings are:

system attribute 4 = 5 - task 27
system attribute 4 = 6 - task 29
system attribute 4 = 7 - task 23
system attribute 4 = 8 - task 34

Task 71: Timer (TIMER)

This system timer is used to control the length of the simulation. This task is a source task and is released at time 0. The task performance is the length of the simulation and should be set by the user.

Upon completion of this task, the system is directed to task 72.

Task 72: Sink (SINK)

This task is used to indicate to the system that the simulation has ended. It may be released upon the completion of task 71. The task performance time is 0.

This is a sink task and the simulation is terminated upon completion of this task.

Task 75: Branch for Clearing (BRCEARA)

This system task is used to determine if the operator could be processing (not including hooking) a track whose identification status has concurrently been altered. It is assumed that the operator would recognize the change and begin processing the symbol under its new form. This task may be released upon the completion of task 63 or 68. The task performance time is 0.

Branching from this task is based on system attribute 5 which is set by user function 45. This function checks to see if the operator is currently processing the track symbol that has just been updated. If it is, an appropriate branch is made to clear the resource. This clearing represents the operator interrupting his processing of the track symbol. If the operator might currently be in the process of hooking the track, the system is directed to task 79.

The possible branches from this task are:

system attribute 5 = 2 - to unknown - task 76
system attribute 5 = 3 - to friendly - task 77
system attribute 5 = 4 - to hostile - task 78
system attribute 5 = 5 - other - task 79

Task 76: Clear to Unknown (CLUNKA)

This task is used for system control to terminate the operator's processing of a track under a status different from unknown and to initiate processing of the track under the unknown status. It may be released upon the completion of task 75. The task performance time is 0.

Resource number 1 is cleared and a signal is sent to task 9.

Task 77: Clear to Friendly (CLFRNA)

This task is used for system control to terminate the operator's processing of a track under a status different from friendly and to initiate his processing of the track under the friendly status. It may be released upon the completion of task 75. The task performance time is 0.

Resource 1 is cleared and a signal is sent to task 21.

Task 78: Clear to Hostile (CLHOSA)

This task is used for system control to terminate the operator's processing of a track with a status different from hostile and to initiate his processing of the track under the hostile status. It may be released upon the completion of task 75. The task performance time is 0.

Resource 1 is cleared and a signal is sent to task 25.

Task 79: Branch for Clearing (BRCLEARB)

This system task is used to determine if the operator could be processing (including hooking) a track whose identification status has concurrently been altered. It is assumed that the operator would recognize the change and begin processing the symbol under its new form. This task may be released upon the completion of task 75. The task performance time is 0.

The value of system attribute 5 is used for this branching and is set by user function 46.

The possible branchings from this task are:

system attribute 5 = 2 - task 80
system attribute 5 = 3 - task 81
system attribute 5 = 4 - task 82

Task 80: Clear to Unknown (CLUNKB)

This task is used for system control and performs the same function as task 76. It may be released upon the completion of task 79.

Resource 1 is cleared and a signal is sent to task 9. Resource 2 is cleared and a signal is sent to task 9.

Task 81: Clear to Friendly (CLFRNB)

This task is used for system control and has the same function as task 77. It may be released upon the completion of task 79. The task performance time is 0.

Resource 1 is cleared and a signal is sent to task 21. Resource 2 is cleared and a signal is sent to task 21.

Task 82: Clear to Hostile (CLHOSB)

This task is used for system control and has the same function as task 78. It may be released upon the completion of task 79. The task performance time is 0.

Resource 1 is cleared and a signal is sent to task 25. Resource 2 is cleared and a signal sent to task 25.

Task 83: Cease Fire Trap (CSTRAP)

This task represents the conclusion of the processing of a cease fire message. It may be released upon the completion of tasks 47, 48, or 49. The task performance time is 0.

There is no branching from this task.

Task 84: Out of Range Trap (ORANTRAP)

This task is called when a target has finished the first part of the engagement but is still out of range for missile firing and represents the holding action of a fire unit during this process. It may be released upon the completion of task 48. The task performance time is 0.

There is no branching from this task.

Task 85: Hold Secondary Assignment (HLD2TRAP)

This task represents the delaying of the engagement when the secondary target has been processed to a point before firing. It may be released upon the completion of task 48. The task performance time is 0.

No branching is taken from this task.

Task 86: Hold Fire Trap (HFTRAP)

This task represents the halting of the engagement due to a hold fire message. It may be released upon the completion of task 49. The task performance time is 0.

There is no branching from this task.

Task 87: Message Trap (MSGTRAP)

This task absorbs the messages to the fire units that require no further action. It may be released upon the completion of task 55 or 57. The task performance time is 0.

There is no branching from this task.

Task 88: Fire Unit Trap (FUTRAP)

This task represents the fire unit returning to the unused state. It may be released upon the completion of task 53. The task performance time is 0.

There is no branching from this task.

Table I

DISTRIBUTION SETS

The distribution sets included in the SAINT model are used as the basis for generating the operator performance times, the fire unit performance times, and fire unit engagement ranges. In this table, M-n refers to moderator function n, U-n refers to user function n and T-n refers to task n.

Distribution Set Number	Location of Use	Definition
1	M-1,BUZY,NHOOK	Random number generator
2	M-1(2)	Search, scanning time
3		Not used
4	T-3,T-9,T-21, T-22,T-28	Observing and recognizing display data
5	U-3,T-4	Radar sweep rate
6	T-7,T-42	Position tap
7	U-18,T-8,T-10, T-11,T-13,T-15, T-19,T-20,T-23, T-26,T-27,T-30, T-32,T-34,T-35, T-40,T-41,T-43, T-44	Pressing single button
8	T-12	Read ARO message
9	T-25,T-27,T-33	Read DDG message
10	T-49	B engagement time for FU
11	T-37,T-39	Enter data via keyboard
12	T-47	Attachment time for FU
13	T-48	A engagement time for FU

Table I (continued)

Distribution Set Number	Location	Definition
14	T-50	Firing time for FU
15	U-33(2)	Auto engagement range for hostile targets
16	U-33(2),U-28	Auto engagement range for unknown targets

Table II

VISUAL VALUES

The visual values given here are used by moderator function 1 to pick the next object for the operator to process. They represent a significance value (VAL) and a stimulation value (STI) that each type of symbol exhibits.

			VAL	STI
		RAW/PROCESS VIDEO		
*	1	Auto Init.	1	2
*	2	Man Init.	6	2
		TRACKS		
		Unknown		
*	3	New	4	3
*	4	Old > 60	2	3
*	5	Old < 60	7	3
		Friendly		
		New		
	6	H → F	9	3
*	7	U → F	4	3
*	8	Old	1	3
		Hostile		
		New		
*	9	U → H	5	4
	10	F → H	7	4
*	11	Old > 60	3	4
*	12	Old < 60	8	4
		Special		
	13	Blinking	9	6
	14	N-B > 60	7	5
	15	N-B < 60	5	5
		FIRE UNITS		
*	16	Blinking	9	6
		Non-Blinking		
*	17	No Host.	1	1
	18	Host. No Eng.	3	3
	19	Engagements	6	9

Table III

SAINT ATTRIBUTES

Information Attributes

1: Track number

2: Fire unit number

3: Fire unit message indicator

1 = clear effective status

2 = engage fire unit to track

3 = hold fire

4 = cease fire

5 = clear hold fire

6 = in range

System Attributes

1-3: Operator branching

4: Hooking procedures - return address

5: Hooking procedures - tape symbol

6: Auto system branching

7: Operator assignment branching

8: Fire unit branching

9: Flight system branching

10-11: Interrupt system branching

SECTION III

DOCUMENTATION OF USER-WRITTEN SUBPROGRAMS

This section presents the program documentation for the user-written subprograms of the SAINT model of the AN/TSQ-73 system. A functional breakdown of the subprograms is given in Table IV. The categories included in the table are:

1. Echo Check (EC) - listing mission input.
2. Operator Processing (OP) - controls simulated actions of operator/repairman.
3. System Processing (SP) - controls simulation of the system's computer operation and the operation of the fire units.
4. Aircraft Processing (AP) - maintains the simulated tracks.
5. Model Operation (MO) - maintains the actual operation of the model, i.e., initializes and resets variables.

Figures 1, 2, and 3 provide a complete listing of all user-written support subprograms, function USERF, and subroutine MODRF, respectively. Figure 1 also contains a listing of the BLOCK DATA subprogram. This subprogram is used to initialize selected variables that appear in the labeled COMMON blocks. In addition, Tables V and VI, appearing at the end of this section, provide definitions of variables used in the user-written subprograms.

Table IV

SUBPROGRAM FUNCTIONAL AREAS

	EC	OP	SP	AP	MO
AHEAD	x				
ASSIG			x		
BUZY		x			
CLOTR			x		
CONT		x			
ENDIT					x
ENG			x		
INTLC					x
LOC	x				
NEWTR		x			
NHOOK			x		
RANGF		x			
RSTART					x
SETTR		x			
SETV		x			
STATE				x	
STORP		x			
UECHO	x				
UIN					x
UPTR		x			
USERF		x	x	x	
MODRF		x	x	x	

Function AHEAD (X,Y)

This function returns the heading in degrees of an aircraft where the aircraft's velocity is given as x and y. This function is called from subroutine UECHO. This function makes use of the standard functions for Arcsine and Arccosine.

Function ASSIG (TR)

This function is used to assign a fire unit to a track for an engagement. It may be called from user function 33 to assign a track in the automatic mode or from user function 42 to assign a track in the manual mode.

The function checks all available fire units eliminating those that are out of commission and those for which the track will come no closer than 25 miles. It also eliminates the possibility of a secondary assignment if the target is not hostile. It assigns a penalty of 40 miles for those fire units that have a primary assignment and then picks the fire unit that is closest to the track at the present time. If no suitable units are available, a value of 0 is returned.

Function BUZY (B,T)

The purpose of this function is to assign a value that has a lower limit of B and an upper limit of T. This value will depend on the total number of tracks and types of tracks that are currently being observed on the scope (i.e., the total sum of values as figured in moderator function 1). This function value will be close to B if the system is busy and close to T if the system is not busy.

This function is called numerous times by different user functions.

Subroutine CLOTR (TR,IIFU,CLV,MIND,TMIN,DIS)

This subroutine returns the closing velocity, CLV, the minimum distance ever obtained, MIND, the time to this minimum distance, TMIN, and the current distance, DIS, of a track, TR, and a fire unit, IIFU. It is called by the function ASSIGN and user functions 30 and 33.

The subroutine first figures the distance between the fire unit and the track. Notice that this is not the value of the SS variable. For an unassigned track the SS variable represents the distance to the center of the system and not the distance to the fire unit. The current distance to the fire unit and the current speed of the aircraft are then computed. If the speed is 0 (or near 0), a special case is assumed and special values are returned. The same approach is used if the distance is 0 (or near 0), i.e., special values are returned. The closing velocity of the track to the site is then computed using the dot product. Finally, the minimum distance and time to this minimum distance are figured.

There are three special case messages returned by this subroutine via special values for the return variables. If the aircraft is flying away from the site, the minimum distance, MIND, is set to 5,000. Also TMIN is set to -1. For a track that is over the site the distance is set to 0, the closing velocity is set to 0 and the minimum distance and time to

minimum are set as if the aircraft were flying away from the site. If the target is not moving the distance to the site, DIS, is set to -1. The closing velocity, CLV, is set to 0 and the minimum distance and time to minimum are set as above.

Subroutine CONT (ITN)

This function is used to set a flag so that the search task will continue on with this target under a new classification that is the result of the current task. This subroutine is called by user functions 6, 18, and 19.

Subroutine ENDIT (I)

This SAINT subroutine is used to print the statistics for each run and reset the variables needed for the next run. This is accomplished by calls to SAINT subroutines UCLCT, UHIST and UTMST as well as a call to the user subroutine RSTART.

Function ENG (ID)

This function is used to check if an engagement has been cancelled. It is a logical value function and returns the value of false if the track and fire units are no longer engaged. It returns the value of true otherwise. It also returns the value for ID equal to 0 if the track is a primary track and the value for ID equal to 1 if a secondary track.

This function is called from several locations in the fire unit section of the program.

Subroutine INTLC

This SAINT subroutine is called by the user subroutine UIN for the purpose of inputting user data. In addition, it initializes the SS variables to a value well outside the range of the radar screen.

Subroutine LOC (TA,TB,X,Y,VX,VY)

This subroutine is used to update the position of the aircraft for the purpose of the echo check. This subroutine is called by subroutine UECHO. The value of TA is the last time of update; the value of TB is the current time. The variables X and Y are input as the old location and output as the current location. Velocity is input through VX and VY.

Function NEWTR (PRN)

This function checks to see if the identification status of the track is the same as the last time the track was observed. It is a logical value function that returns a value true if they are different and false if they are the same. This function is called from the user functions associated with tasks 3, 9, 21, and 25.

Function NHOOK (IT,GTRN)

This function is used to find the next track or fire unit number that would be hooked by pressing the sequence hook button. This function returns the value of 1 if the fire unit or track is the same as that requested by the variable GTRN. It returns the value of 0 if the track or fire unit is different than that requested. It returns the value of -1 if no track or fire unit of the given type is

located. This function is called by user function 23 during the processing of a sequence hook.

The function proceeds through all symbols, both fire units and tracks, until one of the correct type is found. At that time a check is made to see if it is the track requested and a value for the function is assigned accordingly. The possible types for which a function can look are tracks, fire units, and high threat tracks.

Function RANGF (TR,TK)

This function is used to return a value of a piecewise linear function whose characteristics are stored in the user-generaced task characteristics for task TK, values 3-6. This function is called by the user functions associated with tasks 9, 12, and 25.

The value of the function is the value of task characteristic 5 if the range of the track is less than task characteristic 3. The value is equal to task characteristic 6 if the range of the track is greater than the task characteristic 4. If the range is between the values of task characteristics 3 and 4, a value is computed that lies on the linear function between the two end points.

Subroutine RSTART

This subroutine is used to initialize values before each run. It is called by subroutine UIN before run 1 and by subroutine ENDIT before each subsequent run.

Subroutine SETTR (TFU)

This subroutine initializes the information packet and the system attribute that will be required upon completion of the search task.

This subroutine is called by moderator function 1.

Subroutine SETV

This subroutine is called by moderator function 1. It is used to assign observation values to each track. These values are based on three factors. The first depends on time. Its value increases the longer the fire unit or track is not processed by the operator. The second factor is a value that characterizes the type of symbol and therefore its importance. The third factor is a value that reflects the eye-catching ability of the particular type of symbol. For example, a hostile track will have a larger second value than a friendly track and a flashing fire unit will have a larger third value than a friendly track. In order to assign these values, the subroutine checks if the symbol is a track or fire unit, whether it is blinking or not, the type of track it is, whether the system is in an automatic or manual mode, whether the track is close to the center of the system and if the operator has previously processed that particular track under its current identification. (See Table II.)

Subroutine STATE

The SAINT subroutine STATE is used to maintain a continuous monitor on all track locations. Three SS variables are used

for each track. They represent:

SS(I) = location: X-coordinate
SS(I+1) = location: Y-coordinate
SS(I+2) = range to paired fire unit or to center of system

The location is defined by a linear difference equation. Turning points require only an update of the pointer PTR to change the direction of flight. The range for all tracks is initially defined to the center of the system. When a track is engaged by a fire unit, the variable PAIR is updated and the value of the range changes to the distance the track is from the specific fire unit.

Function STORP (B,C,NT)

The purpose of this function is to record the probabilities used for branching in system attributes 1, 2, and 3. This function is called several times by numerous user functions.

The function first checks to see if the variable NT is equal to 0. If it is not equal to 0, the user defined task characteristics for task NT are used to modify the results. If it is equal to 0, no task characteristics are used. The values of B and C are then stored in system attributes 2 and 3, respectively and the function is given the value for system attribute 1, i.e., 1 - (B+C).

Subroutine UECHO

This subroutine is used to print the data that is read by subroutine UIN. In addition, it prints out the meaning of

the symbols used and the operator job trace. It is called by subroutine UIN and subroutine ENDIT.

The logical variables that control the system procedures are converted to alphanumerics for printing. This data is then printed. Fire unit data is then printed with one line for each fire unit. Track information is then printed in the following manner. First, the track number, the initialization time and characteristics are printed. Then the remaining heading changes and status changes are sorted and printed in the proper chronological order without a track number. Since the turning points are determined by time, the location at each time is calculated by a call to LOC. Also, the speed in miles per hour and the heading in degrees are calculated and printed along with the information that was read as data.

Subroutine UIN

This subroutine is used to handle the user input data. It is called from subroutine INTLC. It can be divided into four sections. Section 1 reads the policy information; section 2 reads the fire unit information; section 3 reads track routing information; and section 4 reads the track identification information. Upon completion of the reading process, a call to RSTART is made to initialize values. Then a call is made to UECHO to print the user echo check of all the data read.

Function UPTR (TRN)

This function updates the status of a track as a result of an interrogation process by the operator. It is a logical value function that returns the value FALSE if an update was made and a value of TRUE if no change occurred. This function is called from the user functions associated with tasks 8 and 12.

```
      FUNCTION AHEAD(Y,X)                                                AHEAD    1
      LOGICAL LS,LC                                                      AHEAD    2
                                                                         AHEAD    3
      R = (X**2 + Y**2)**.5                                              AHEAD    4
      C = X/R                                                            AHEAD    5
      LC = C .GT. 0.                                                     AHEAD    6
      S = Y/R                                                            AHEAD    7
      LS = S .GT. 0.                                                     AHEAD    8
      S = ASIN(S)                                                        AHEAD    9
      C = ACOS(C)                                                        AHEAD   10
      IF(LS .AND. LC) AHEAD = IFIX((S*57.29577) + .5)                    AHEAD   11
      IF(LS .AND. .NOT. LC  .OR. .NOT. LS .AND. .NOT. LC)                AHEAD   12
     *          AHEAD = IFIX(180.1 - (S * 57.29577))                    ERR2     1
      IF((.NOT. LS) .AND. LC)                                            AHEAD   14
     *          AHEAD = IFIX(360.1 - (C * 57.29577))                    ERR2     2
      RETURN                                                             AHEAD   16
      END                                                                AHEAD   17
```

```
      FUNCTION ASSIG(TR)                                                 ASSIG    1
      LOGICAL LP                                                         ASSIG    2
      INTEGER TR,BFU                                                     ASSIG    3
                                                                         ASSIG    4
      REAL TRCLA(33,5),FUCLA(11,9)                                       UCOM1    1
      COMMON /UCOM1/ TRCLA,FUCLA                                         UCOM1    2
                                                                         UCOM1    3
      INTEGER NFU,NTRFU,NTRK                                             UCOM7    1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                      UCOM7    2
                                                                         UCOM7    3
                                                                         UCOM7    4
                                                                         UCOM7    5
                                                                         ASSIG    7
C     CHECK FOR POSSIBLE FU                                              ASSIG    8
      BV = 10000000.                                                     ASSIG    9
      BFU = 0                                                            ASSIG   10
      DO 10 NF = 1,NFU                                                   ASSIG   11
         IF((FUCLA(NF,1) .EQ. 7.)  .OR.                                  ASSIG   12
     *      (FUCLA(NF,1) .EQ. 0.)  .OR.                                  ASSIG   13
     *      (FUCLA(NF,3) .NE. 0.))  GO TO 10                             ASSIG   14
                                                                         ASSIG   15
C     CHECK DISTANCES                                                    ASSIG   16
         CALL CLOTR(TR,NF,CV,DMIN,TMIN,DIS)                              ASSIG   17
         IF(DMIN .GT. 25.)  GO TO 10                                     ASSIG   18
            LP = FUCLA(NF,2) .NE. 0.                                     ASSIG   19
            IF(LP .AND. (TRCLA(TR,1) .NE. 4.))  GO TO 10                 ASSIG   20
            IF(LP)  DIS = DIS + 40.                                      ASSIG   21
            IF(DIS .GT. BV)  GO TO 10                                    ASSIG   22
                                                                         ASSIG   23
C     RECORD BEST VALUES                                                 ASSIG   24
               BV = DIS                                                  ASSIG   25
               BFU = NF                                                  ASSIG   26
   10 CONTINUE                                                           ASSIG   27
                                                                         ASSIG   28
C     RECORD SELECTION  IF 0  NO POSSIBLE                                ASSIG   29
      ASSIG = FLOAT(BFU)                                                 ASSIG   30
      RETURN                                                             ASSIG   31
      END                                                                ASSIG   32
```

Figure 1(1). Program Listing: Support Programs

```
      FUNCTION BUZY(B,T)                                                BUZY      1
      REAL VALUE(20),STI(20),STOT                                       UCOM5     1
      COMMON /UCOM5/ VALUE,STI,STOT                                     UCOM5     2
                                                                        UCOM5     3
                                                                        BUZY      3
      BUZY = (T - B) * (1. - (AMIN1(1.,(STOT / 300.)) *                 BUZY      4
     *         UNFRM(1))) + B                                           BUZY      5
      RETURN                                                            BUZY      6
      END                                                               BUZY      7



      SUBROUTINE CLOTR(TR,IIFU,CLV,MIND,TMIN,DIS)                       CLOTR     1
      INTEGER TR,FU                                                     CLOTR     2
      REAL CLV,MIND,TMIN                                                CLOTR     3
                                                                        CLOTR     4
      INTEGER PAIR(33),PTR(33),PTT(33),RSTAT(33)                        CLOTR     5
      COMMON /UCOM3/ PTR,PTT,RSTAT,PAIR                                 CLOTR     6
                                                                        CLOTR     7
      REAL TRCLA(33,5),FUCLA(11,9)                                      UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                        UCOM1     2
                                                                        UCOM1     3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)             UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                            UCOM2     2
                                                                        UCOM2     3
                                                                        CLOTR    10
      COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)     COM17     1
                                                                        CLOTR    12
C     FIGURE VECTOR BETWEEN TR AND FU                                   CLOTR    13
      FU = IIFU                                                         CLOTR    14
      IF(FU .EQ. 0)  FU = 11                                            CLOTR    15
      ITR = TR * 3 - 2                                                  CLOTR    16
      FX = FUCLA(FU,4) - SS(ITR)                                        CLOTR    17
      FY = FUCLA(FU,5) - SS(ITR + 1)                                    CLOTR    18
                                                                        CLOTR    19
C     FIGURE DIST                                                       CLOTR    20
      DIS = (FX**2 + FY**2) **.5                                        CLOTR    21
                                                                        CLOTR    22
C     FIGURE SPEED                                                      CLOTR    23
      VX = TRROU(PTR(TR),2)                                             CLOTR    24
      VY = TRROU(PTR(TR),3)                                             CLOTR    25
      SP = (VX**2 + VY**2) **.5                                         CLOTR    26
      IF(ABS(SP) .LT. .005)  GO TO 30                                   CLOTR    27
                                                                        CLOTR    28
C     FIGURE CLOSSING VELOCITY IF NOT AT SITE                           CLOTR    29
      IF(ABS(DIS) .LT. .1)  GO TO 20                                    CLOTR    30
         CLV = (FX * VX  +  FY * VY) / DIS                              CLOTR    31
                                                                        CLOTR    32
C     FIGURE MIND                                                       CLOTR    33
      IF(CLV .LT. 0.)  GO TO 10                                         CLOTR    34
         CO = CLV / SP                                                  CLOTR    35
      SI = (1. - CO**2)**.5                                             CLOTR    36
      MIND = DIS * SI                                                   CLOTR    37
      TMIN = CO * DIS / SP                                              CLOTR    38
         RETURN                                                         CLOTR    39
                                                                        CLOTR    40
C     GOING AWAY FROM SITE                                              CLOTR    41
 10      MIND = 5000.                                                   CLOTR    42
         TMIN = -1                                                      CLOTR    43
         RETURN                                                         CLOTR    44
                                                                        CLOTR    45
C     TR OVER SITE                                                      CLOTR    46
 20      DIS = 0.                                                       CLOTR    47
         CLV = 0.                                                       CLOTR    48
         GO TO 10                                                       CLOTR    49
                                                                        CLOTR    50
C     TR NOT MOVING                                                     CLOTR    51
 30      DIS = -1                                                       CLOTR    52
         CLV = 0.                                                       CLOTR    53
         GO TO 10                                                       CLOTR    54
      END                                                               CLOTR    55
```

Figure 1(2). Program Listing: Support Programs

```
      SUBROUTINE CONT(ITN)                                                  CONT      1
      INTEGER NFU,NTRFU,NTRK                                                UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                         UCOM7     2
                                                                            UCOM7     3
                                                                            UCOM7     4
                                                                            UCOM7     5
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)                 UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                                UCOM2     2
                                                                            UCOM2     3
                                                                            CONT      4
                                                                            CONT      5
      DO 10 I = 1,NTRFU                                                     CONT      6
         IF(IFIX(TRSTA(I,1)) .EQ. ITN) TRSTA(I,3) = -1                      CONT      7
   10    CONTINUE                                                           CONT      8
      RETURN                                                                CONT      9
      END                                                                   CONT     10
```

```
      SUBROUTINE ENDIT(I)                                                   ENDIT     1
      CALL RSTART                                                           ENDIT     2
      CALL UCLCT(1.,0)                                                      ENDIT     3
      CALL UHIST(1.,0)                                                      ENDIT     4
      CALL UTMST(1.,A,0)                                                    ENDIT     5
      RETURN                                                                ENDIT     6
      END                                                                   ENDIT     7
```

```
      FUNCTION ENG(ID)                                                      ENG       1
      LOGICAL ENG                                                           ENG       2
                                                                            ENG       3
      REAL TRCLA(33,5),FUCLA(11,9)                                          UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                            UCOM1     2
                                                                            UCOM1     3
                                                                            ENG       5
                                                                            ENG       6
C     CHECK IF ENGAGEMENT HAS BEEN CANCLED                                  ENG       7
      CALL GETIA(1,TRN)                                                     ENG       8
      CALL GETIA(2,FN)                                                      ENG       9
      ITRN = TRN                                                            ENG      10
      IFUN = FN                                                             ENG      11
      ENG = .FALSE.                                                         ENG      12
      IF(((FUCLA(IFUN,2) .EQ. TRN)  .AND.                                   ENG      13
     *         (TRCLA(ITRN,4) .EQ. FN))  .OR.                               ENG      14
     *         ((FUCLA(IFUN,3) .EQ. TRN)  .AND.                             ENG      15
     *         (TRCLA(ITRN,4) .EQ. FN)))  ENG = .TRUE.                      ENG      16
      ID = 0                                                                ENG      17
      IF(FUCLA(IFUN,2) .NE. TRN)  ID = 1                                    ENG      18
      RETURN                                                                ENG      19
      END                                                                   ENG      20
```

Figure 1(3). Program Listing: Support Programs

```
      SUBROUTINE INTLC                                                    INTLC     1
      INTEGER NFU,NTRFU,NTRK                                              UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                       UCOM7     2
                                                                          UCOM7     3
                                                                          UCOM7     4
                                                                          UCOM7     5
                                                                          INTLC     3
      COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)       COM17     1
                                                                          INTLC     5
                                                                          INTLC     6
      CALL UIN                                                            INTLC     7
      K = 3 * NTRK                                                        INTLC     8
      DO 10 I = 1,K                                                       INTLC     9
         SS(I) = 5000.                                                    INTLC    10
   10    CONTINUE                                                         INTLC    11
      RETURN                                                              INTLC    12
      END                                                                 INTLC    13


      SUBROUTINE LOC(TA,TB,X,Y,VX,VY)                                     LOC       1
      T = TB - TA                                                         LOC       2
      X = X + VX * T                                                      LOC       3
      Y = Y + VY * T                                                      LOC       4
      RETURN                                                              LOC       5
      END                                                                 LOC       6


      FUNCTION NEWTR(TRN)                                                 NEWTR     1
      INTEGER TRN                                                         NEWTR     2
      LOGICAL NEWTR                                                       NEWTR     3
                                                                          NEWTR     4
      REAL TRCLA(33,5),FUCLA(11,9)                                        UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                          UCOM1     2
                                                                          UCOM1     3
                                                                          NEWTR     6
                                                                          NEWTR     7
C     CHECK IF OBSERVED .EQ. LAST                                         NEWTR     8
      IF(TRCLA(TRN,1) .NE. TRCLA(TRN,3))  GO TO 10                        NEWTR     9
                                                                          NEWTR    10
C     THEY ARE THE SAME                                                   NEWTR    11
         NEWTR = .FALSE.                                                  NEWTR    12
         RETURN                                                           NEWTR    13
                                                                          NEWTR    14
C     THEY ARE DIFFERENT                                                  NEWTR    15
   10 NEWTR = .TRUE.                                                      NEWTR    16
      TRCLA(TRN,3) = TRCLA(TRN,1)                                         NEWTR    17
      RETURN                                                              NEWTR    18
      END                                                                 NEWTR    19
```

Figure 1(4). Program Listing: Support Programs

```
      FUNCTION NHOOK(IT,GTRN)                                          NHOOK     1
      INTEGER SHPT,SHBPT                                               NHOOK     2
                                                                       NHOOK     3
                                                                       NHOOK     4
      COMMON /COM17/ SS(100),DDDX(500)                                 NHOOK     5
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)            UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                           UCOM2     2
                                                                       UCOM2     3
      INTEGER NFU,NTRFU,NTRK                                           UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                    UCOM7     2
                                                                       UCOM7     3
                                                                       UCOM7     4
                                                                       UCOM7     5
                                                                       NHOOK     8
      DATA SHPT/1/                                                     NHOOK     9
                                                                       NHOOK    10
C     STORE BEGINNING POINTER                                          NHOOK    11
      SHBPT = SHPT                                                     NHOOK    12
      SHPT = SHPT + 1                                                  NHOOK    13
                                                                       NHOOK    14
C     LOOK FOR NEXT PPOSSIBLE SITE                                     NHOOK    15
   10 IF(SHPT .GT. NTRFU)  SHPT = 1                                    NHOOK    16
       IF(GTRN .NE. 0.)  GO TO 11                                      NHOOK    17
         IF(UNFRM(1) .GT. .6)  GO TO 15                                NHOOK    18
         GO TO 30                                                      NHOOK    19
   11  CONTINUE                                                        NHOOK    20
         IF(IT .GT. 0)  GO TO 50                                       NHOOK    21
                                                                       NHOOK    22
C     LOOKING FOR TRACK                                                NHOOK    23
            IF(TRSTA(SHPT,1) .LE. 0.)  GO TO 20                        NHOOK    24
C     FOUND TRACK                                                      NHOOK    25
               IF(TRSTA(SHPT,1) .NE. GTRN)  GO TO 30                   NHOOK    26
                                                                       NHOOK    27
C     FOUND CORRECT TRACK                                              NHOOK    28
   15             NHOOK = 1                                            NHOOK    29
                  RETURN                                               NHOOK    30
                                                                       NHOOK    31
C     NOT A TRACK OR INCORRECT TRACK                                   NHOOK    32
   20       IF(SHBPT .EQ. SHPT)  GO TO 40                             NHOOK    33
            SHPT = SHPT + 1                                            NHOOK    34
            GO TO 10                                                   NHOOK    35
                                                                       NHOOK    36
C      FOUND INCORRECT TRACK                                           NHOOK    37
   30       NHOOK = 0                                                 NHOOK    38
            RETURN                                                     NHOOK    39
                                                                       NHOOK    40
C     FOUND NO TRACK                                                   NHOOK    41
   40       NHOOK = -1                                                NHOOK    42
            RETURN                                                     NHOOK    43
   50     IF(IT .GT. 1)  GO TO 60                                     NHOOK    44
                                                                       NHOOK    45
C     LOOKING FOR FIRE UNIT                                            NHOOK    46
            IF(TRSTA(SHPT,1) .GE. 0.)  GO TO 20                        NHOOK    47
                                                                       NHOOK    48
C     FOUND FIRE UNIT                                                  NHOOK    49
            IF(TRSTA(SHPT,1) .EQ. GTRN)  GO TO 15                      NHOOK    50
            GO TO 30                                                   NHOOK    51
                                                                       NHOOK    52
C     LOOKING FOR HT                                                   NHOOK    53
   60       IF(TRSTA(SHPT,1) .LE. 0.)  GO TO 20                       NHOOK    54
                                                                       NHOOK    55
C     FOUND TRACK                                                      NHOOK    56
            IR = TRSTA(SHPT,1) * 3.                                    NHOOK    57
            IF(SS(IR) .GT. 60.)  GO TO 20                              NHOOK    58
                                                                       NHOOK    59
C     FOUND HOSTIL TRACK                                               NHOOK    60
            IF(TRSTA(SHPT,1) .EQ. GTRN)  GO TO 15                      NHOOK    61
            GO TO 30                                                   NHOOK    62
      END                                                              NHOOK    63
```

Figure 1(5). Program Listing: Support Programs

```
      FUNCTION RANGF(TR,TK)                                                  RANGF     1
      INTEGER TR,TK                                                          RANGF     2
                                                                             RANGF     3
      COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)          COM17     1
                                                                             RANGF     5
                                                                             RANGF     6
C     GET RANGES AND PROPABILITIES                                           RANGF     7
      R = SS(3 * TR)                                                         RANGF     8
      CALL GETTC(TK,3,R1)                                                    RANGF     9
      CALL GETTC(TK,4,R2)                                                    RANGF    10
      CALL GETTC(TK,5,P1)                                                    RANGF    11
      CALL GETTC(TK,6,P2)                                                    RANGF    12
                                                                             RANGF    13
C     CHECK RANGE SIZE                                                       RANGF    14
      IF(R .GE. R2)  GO TO 10                                                RANGF    15
      IF(R .LE. R1)  GO TO 20                                                RANGF    16
                                                                             RANGF    17
C     RANGE IN MIDDLE SECTION                                                RANGF    18
      RANGF = ((R - R1)/(R2 - R1)) * (P2 - P1) + P1                          RANGF    19
         RETURN                                                              RANGF    20
                                                                             RANGF    21
C     RANGF IN UPPER SECTION                                                 RANGF    22
   10    RANGF = P2                                                          RANGF    23
         RETURN                                                              RANGF    24
                                                                             RANGF    25
C     RANGF IN LOWER SECTION                                                 RANGF    26
   20    RANGF = P1                                                          RANGF    27
         RETURN                                                              RANGF    28
      END                                                                    RANGF    29
```

Figure 1(6). Program Listing: Support Programs

```
      SUBROUTINE RSTART                                                  RSTART    1
      REAL TRCLA(33,5),FUCLA(11,9)                                       UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                         UCOM1     2
                                                                         UCOM1     3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)              UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                             UCOM2     2
                                                                         UCOM2     3
      INTEGER PAIR(33),PTR(33),PTT(33),RSTAT(33)                         UCOM3     1
      COMMON /UCOM3/ PTR,PTT,RSTAT,PAIR                                  UCOM3     2
                                                                         UCOM3     3
      LOGICAL AUTOI,AUTOR,AUTOE,TIGH                                     UCOM4     1
      COMMON /UCOM4/ AUTOI,AUTOR,AUTOE,TIGH                              UCOM4     2
                                                                         UCOM4     3
      REAL VALUE(20),STI(20),STOT                                        UCOM5     1
      COMMON /UCOM5/ VALUE,STI,STOT                                      UCOM5     2
                                                                         UCOM5     3
      INTEGER TYHOOK,SEQT,PSEQ                                           UCOM6     1
      COMMON /UCOM6/ TYHOOK,SEQT,PSEQ                                    UCOM6     2
                                                                         UCOM6     3
      INTEGER NFU,NTRFU,NTRK                                             UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                      UCOM7     2
                                                                         UCOM7     3
                                                                         UCOM7     4
                                                                         UCOM7     5
      REAL CX(33),CY(33)                                                 UCOM8     1
      INTEGER IPTR(33),IPTT(33)                                          UCOM8     2
      COMMON /UCOM8/ CX,CY,IPTR,IPTT,IPC                                 UCOM8     3
                                                                         UCOM8     4
      LOGICAL TRCH                                                       UCOM9     1
      REAL TRMOD(33),TOTRT(33),TMARK,TMARE                               UCOM9     2
      INTEGER NOLDTY,LPAGE                                               UCOM9     3
      COMMON /UCOM9/ TRCH,TRMOD,TOTRT,TMARK,TMARE,LPAGE,NOLDTY           UCOM9     4
      COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)      COM17     1
                                                                         RSTART    4
      REAL TFUN(10)                                                      RSTART    5
      COMMON /UCOM0/ TFUN                                                RSTART    6
      IPC = 0                                                            RSTART    7
      LPAGE = 0                                                          RSTART    8
      TMARK = 0.                                                         RSTART    9
      TMARE = 0.                                                         RSTART   10
      NOLDTY = 1                                                         RSTART   11
      TRCH = .FALSE.                                                     RSTART   12
      DO 10 I = 1,33                                                     RSTART   13
         TRMOD(I) = 1.                                                   RSTART   14
10       TOTRT(I) = 0.                                                   RSTART   15
      DO 20 I = 1,NTRK                                                   RSTART   16
         RSTAT(I) = 4.                                                   RSTART   17
         TRMOD(I) = 0.                                                   RSTART   18
         PAIR(I) = 11                                                    RSTART   19
         PTR(I) = IPTR(I)                                                RSTART   20
         PTT(I) = IPTT(I)                                                RSTART   21
         TRROU(PTR(I),2) = CX(I)                                         RSTART   22
20       TRROU(PTR(I),3) = CY(I)                                         RSTART   23
      DO 30 I = 1,NTRFU                                                  RSTART   24
30       TRSTA(I,2) = 0.                                                 RSTART   25
      DO 40 I = 1,NFU                                                    RSTART   26
         FUCLA(I,8) = TFUN(I)                                            RSTART   27
      FUCLA(I,1) = 1.                                                    ERR2      1
      TRSTA(I,1) = -I                                                    ERR2      2
         FUCLA(I,2) = 0.                                                 RSTART   28
         FUCLA(I,3) = 0.                                                 RSTART   29
         FUCLA(I,6) = 0.                                                 RSTART   30
40       FUCLA(I,7) = 0.                                                 RSTART   31
      DO 50 I = 1,NTRK                                                   RSTART   32
      DO 50 J = 1,5                                                      RSTART   33
50       TRCLA(I,J) = 0.                                                 RSTART   34
      J = NTRK * 3                                                       RSTART   35
      DO 60 I = 1,J                                                      RSTART   36
         SS(I) = 5000.                                                   RSTART   37
60    CONTINUE                                                           RSTART   38
      RETURN                                                             RSTART   39
      END                                                                RSTART   40
```

Figure 1(7). Program Listing: Support Programs

```
      SUBROUTINE SETTR(TFU)                                              SETTR    1
      REAL TRCLA(33,5),FUCLA(11,9)                                       UCOM1    1
      COMMON /UCOM1/ TRCLA,FUCLA                                         UCOM1    2
                                                                         UCOM1    3
                                                                         SETTR    3
                                                                         SETTR    4
C     CHECK FOR FU OR TRACK                                              SETTR    5
      IF(TFU .GT. 0.)  GO TO 10                                          SETTR    6
                                                                         SETTR    7
C     IT IS A FU NUMBER                                                  SETTR    8
         CALL PUTIA(2,-TFU)                                              SETTR    9
         CALL PUTSA(1,5.)                                                SETTR   10
         RETURN                                                          SETTR   11
                                                                         SETTR   12
C     IT IS A TRACK NUMBER                                               SETTR   13
   10    CALL PUTIA(1,TFU)                                               SETTR   14
         CALL PUTSA(1,TRCLA(IFIX(TFU),1))                                SETTR   15
         RETURN                                                          SETTR   16
      END                                                                SETTR   17
```

```
      SUBROUTINE SETV                                                    SETV     1
      LOGICAL LDIS,LOLD                                                  SETV     2
                                                                         SETV     3
      COMMON /COM17/ SS(100),DDDX(500)                                   SETV     4
      REAL TRCLA(33,5),FUCLA(11,9)                                       UCOM1    1
      COMMON /UCOM1/ TRCLA,FUCLA                                         UCOM1    2
                                                                         UCOM1    3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)              UCOM2    1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                             UCOM2    2
                                                                         UCOM2    3
      LOGICAL AUTOI,AUTOR,AUTOE,TIGH                                     UCOM4    1
      COMMON /UCOM4/ AUTOI,AUTOR,AUTOE,TIGH                              UCOM4    2
                                                                         UCOM4    3
      REAL VALUE(20),STI(20),STOT                                        UCOM5    1
      COMMON /UCOM5/ VALUE,STI,STOT                                      UCOM5    2
                                                                         UCOM5    3
      INTEGER NFU,NTRFU,NTRK                                             UCOM7    1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                      UCOM7    2
                                                                         UCOM7    3
                                                                         UCOM7    4
                                                                         UCOM7    5
                                                                         SETV    10
      COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NPUNCH,      COM06    1
     *               NRNIT,NRENT,MNDC,NDC,NDTN,NNTC                      COM06    2
                                                                         SETV    12
                                                                         SETV    13
                                                                         SETV    14
C     FOR ALL TR AND FU                                                  SETV    15
      DO 80 I = 1,NTRFU                                                  SETV    16
         TRSTA(I,3) = 0.                                                 SETV    17
         NTF = TRSTA(I,1)                                                SETV    18
         TVAL = (TNOW - TRSTA(I,2)) / 50.                                SETV    19
                                                                         SETV    20
C     DECIDE IF TR OR FU                                                 SETV    21
         IF(NTF .EQ. 0)  GO TO 80                                        SETV    22
         IF(NTF .GT. 0)  GO TO 20                                        SETV    23
                                                                         SETV    24
C     IT IS A FU                                                         SETV    25
         NTF = -NTF                                                      SETV    26
         IF(FUCLA(NTF,1) .NE. 10.)  GO TO 10                             SETV    27
                                                                         SETV    28
```

Figure 1(8). Program Listing: Support Programs

```
C     BLINKING FU                                                    SETU    29
            J = 13                                                   SETU    30
            GO TO 70                                                 SETU    31
                                                                     SETU    32
C     NON -- BLINKING FU                                             SETU    33
 10         J = 18                                                   SETU    34
            GO TO 70                                                 SETU    35
                                                                     SETU    36
C     IT IS A TRACK                                                  SETU    37
 20       IR = NTF * 3                                              SETU    38
          LDIS = SS(IR) .GT. 60                                      SETU    39
          LOLD = TRCLA(NTF,1) .EQ. TRCLA(NTF,3)                      SETU    40
          ITY = TRCLA(NTF,1)                                         SETU    41
          IF(ITY .EQ. 0)  GO TO 80                                   SETU    42
          GO TO (30,40,50,60), ITY                                   SETU    43
                                                                     SETU    44
C     IT IS RAW VIDEO                                                SETU    45
 30         J = 2                                                    SETU    46
            IF(AUTOI)  J = 1                                         SETU    47
            GO TO 70                                                 SETU    48
                                                                     SETU    49
C     IT IS UNKNOWN                                                  SETU    50
 40         J = 3                                                    SETU    51
            IF(LOLD)  J = 4                                          SETU    52
            IF(LOLD .AND. LDIS)  J = 5                               SETU    53
            GO TO 70                                                 SETU    54
                                                                     SETU    55
C     IT IS FRIENDLY                                                 SETU    56
 50         J = 7                                                    SETU    57
            IF(LOLD)  J = 8                                          SETU    58
            GO TO 70                                                 SETU    59
                                                                     SETU    60
C     IT IS HOSTILE                                                  SETU    61
 60         J = 9                                                    SETU    62
            IF(LOLD)  J = 11                                         SETU    63
            IF(LOLD .AND. LDIS)  J = 12                              SETU    64
                                                                     SETU    65
 70       TRSTA(I,3) = (TVAL + .1) * (VALUE(J) + STI(J))            SETU    66
                                                                     SETU    67
 80   CONTINUE                                                       SETU    68
      RETURN                                                         SETU    69
      END                                                            SETU    70
```

Figure 1(9). Program Listing: Support Programs

```
      SUBROUTINE STATE                                                    STATE    1
      REAL TRCLA(33,5),FUCLA(11,3)                                        UCOM1    1
      COMMON /UCOM1/ TRCLA,FUCLA                                          UCOM1    2
                                                                          UCOM1    3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)               UCOM2    1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                              UCOM2    2
                                                                          UCOM2    3
      INTEGER PAIR(33),PTR(33),PTT(33),RSTAT(33)                          UCOM3    1
      COMMON /UCOM3/ PTR,PTT,RSTAT,PAIR                                   UCOM3    2
                                                                          UCOM3    3
      INTEGER NFU,NTRFU,NTRK                                              UCOM7    1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                       UCOM7    2
                                                                          UCOM7    3
                                                                          UCOM7    4
                                                                          UCOM7    5
                                                                          STATE    6
      COMMON /COM16/ AAERR,DTMAX,DTMIN,DTSAV,IITES,LLERR,RRERR,TTLAS,     COM16    1
     *               TTSAV,DTSUG,DTFUL,DTNOW,ISEES,RESLS,DTACC,LLSAV,     COM16    2
     *               LSAVE                                                COM16    3
      COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)       COM17    1
                                                                          STATE    9
                                                                          STATE   10
C     FOR EACH TRACK                                                      STATE   11
      K = ((3 * NTRK) - 2)                                                STATE   12
      J = 0                                                               STATE   13
      DO 10 I = 1,K,3                                                     STATE   14
         J = J + 1                                                        STATE   15
                                                                          STATE   16
C     X - POS   Y - POS   DISTANCE FORM SITE                              STATE   17
         SS(I) = SSL(I) + TRROU(PTR(J),2) * DTNOW                         STATE   18
         SS(I + 1) = SSL(I + 1) + TRROU(PTR(J),3) * DTNOW                 STATE   19
         SS(I + 2) = ((SS(I) - FUCLA(PAIR(J),4)) **2 +                    STATE   20
     *         (SS(I + 1) - FUCLA(PAIR(J),5)) **2) **.5                   STATE   21
   10    CONTINUE                                                         STATE   22
      RETURN                                                              STATE   23
      END                                                                 STATE   24
```

```
      FUNCTION STORP(B,C,NT)                                              STORP    1
                                                                          STORP    2
C     CHECK IF TASK CHAR ARE USED                                         STORP    3
      IF(NT .NE. 0)  GO TO 10                                             STORP    4
         BB = B                                                           STORP    5
         BC = C                                                           STORP    6
         GO TO 20                                                         STORP    7
                                                                          STORP    8
   10 CONTINUE                                                            STORP    9
         CALL GETTC(NT,1,AB)                                              STORP   10
         CALL GETTC(NT,2,AC)                                              STORP   11
         BB = B * AB                                                      STORP   12
         BC = C * AC                                                      STORP   13
                                                                          STORP   14
   20 STORP = 1. - (BB + BC)                                              STORP   15
      CALL PUTSA(2,BB)                                                    STORP   16
      CALL PUTSA(3,BC)                                                    STORP   17
      RETURN                                                              STORP   18
      END                                                                 STORP   19
```

Figure 1(10). Porgram Listing: Support Programs

```
      SUBROUTINE UECHO                                                  UECHO     1
                                                                        UECHO     2
      REAL TRCLA(33,5),FUCLA(11,9)                                      UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                        UCOM1     2
                                                                        UCOM1     3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)             UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                            UCOM2     2
                                                                        UCOM2     3
      INTEGER PAIR(33),PTR(33),PTT(33),RSTAT(33)                        UCOM3     1
      COMMON /UCOM3/ PTR,PTT,RSTAT,PAIR                                 UCOM3     2
                                                                        UCOM3     3
      LOGICAL AUTOI,AUTOR,AUTOE,TIGH                                    UCOM4     1
      COMMON /UCOM4/ AUTOI,AUTOR,AUTOE,TIGH                             UCOM4     2
                                                                        UCOM4     3
      REAL VALUE(20),STI(20),STOT                                       UCOM5     1
      COMMON /UCOM5/ VALUE,STI,STOT                                     UCOM5     2
                                                                        UCOM5     3
      INTEGER TYHOOK,SEQT,PSEQ                                          UCOM6     1
      COMMON /UCOM6/ TYHOOK,SEQT,PSEQ                                   UCOM6     2
                                                                        UCOM6     3
      INTEGER NFU,NTRFU,NTRK                                            UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                     UCOM7     2
                                                                        UCOM7     3
                                                                        UCOM7     4
                                                                        UCOM7     5
      REAL CX(33),CY(33)                                                UCOM8     1
      INTEGER IPTR(33),IPTT(33)                                         UCOM8     2
      COMMON /UCOM8/ CX,CY,IPTR,IPTT,IPC                                UCOM8     3
                                                                        UCOM8     4
      LOGICAL TRCH                                                      UCOM9     1
      REAL TRMOD(33),TOTRT(33),TMARK,TMARE                              UCOM9     2
      INTEGER NOLDTY,LPAGE                                              UCOM9     3
      COMMON /UCOM9/ TRCH,TRMOD,TOTRT,TMARK,TMARE,LPAGE,NOLDTY          UCOM9     4
                                                                        UECHO     4
1000  FORMAT(1H1///49X,31HS A I N T   S I M U L A T I O N/49X,          FORMAT    1
     *       9(1H-),3X,19(1H-)//59X,11HO F   T H E/59X,3H---,3X,       FORMAT    2
     *       5H-----//56X,17HA N / T S Q - 7 3/56X,17(1H-)//           FORMAT    3
     *      32X,36HG U I D E D   M I S S I L E   A I R ,               FORMAT    4
     *      29H  D E F E N S E   S Y S T E M/                          FORMAT    5
     *      32X,11(1H-),3X,13(1H-),3X,5H-----,3X,13(1H-),3X,11(1H-)//  FORMAT    6
     *)                                                                FORMAT    7
                                                                       FORMAT    8
1001  FORMAT(///27X,77(1H-)/27X,77(1H-)//49X,                          FORMAT    9
     *      31HO P E R A T I O N A L   D A T A/49X,21(1H-),3X,7(1H-)   FORMAT   10
     *      //49X,34HINITIAL OPERATIONAL MODES/POLICIES//              FORMAT   11
     *      49X,20HAUTO/MANUAL INITIATE,6X,2A4/                       FORMAT   12
     *      49X,23HAUTO/MANUAL INTERROGATE,3X,2A4/                    FORMAT   13
     *      49X,22HAUTO/MANUAL ENGAGEMENT,4X,2A4/                     FORMAT   14
     *      49X,21HTIGHT/FREE ENGAGEMENT,5X,2A4/                      FORMAT   15
     *      49X,14HHOOKING POLICY,12X,2A4/                            FORMAT   16
     *      78X,2A4)                                                  FORMAT   17
1002  FORMAT(//27X,77(1H-)/27X,77(1H-)//                               FORMAT   18
     *      32X,30HA S S O C I A T E D   F I R E ,                    FORMAT   19
     *      33H  U N I T   I N F O R M A T I O N/                     FORMAT   20
     *      32X,19(1H-),3X,7(1H-),3X,7(1H-),3X,21(1H-)/               FORMAT   21
     *      54X,33HL O C A T I O N  QUANTITY  EFFECT/                 FORMAT   22
     *      48X,39HNO    X-CORD   Y-CORD  WEAPONS    RATIO/)          FORMAT   23
                                                                       FORMAT   24
1003  FORMAT(48X,I2,F10.2,F9.2,I7,F10.3)                               FORMAT   25
                                                                       FORMAT   26
1004  FORMAT(1H1////49X,33HT R A C K   I N F O R M A T I O N/          FORMAT   27
     *      49X,9(1H-),3X,21(1H-)/                                    FORMAT   28
     *      50X,32HL O C A T I O N  V E L O C I T Y/                  FORMAT   29
     *      23X,2HNO,6X,4HTIME,7X,2HID,6X,                            FORMAT   30
     *         31HX-CORD   Y-CORD   X-VEL   Y-VEL,5X,                 FORMAT   31
     *         20HS P E E D    HEADING/                               FORMAT   32
     *      68X,29H(MILES / SEC)  (MILES / HOUR)/)                    FORMAT   33
```

Figure 1(11). Program Listing: Support Programs

```
1005  FORMAT(23X,I2,F11.2,3X,2A4,2F9.2,2F8.3,5X,F8.3,7X,I3)                  FORMAT      34
1105  FORMAT(25X,F11.2,3X,2A4,2F9.2,2F8.3,5X,F8.3,7X,I3)                    FORMAT      35
                                                                            FORMAT      36
1006  FORMAT(1H )                                                           FORMAT      37
1007  FORMAT(1H1//)                                                         FORMAT      38
1008  FORMAT(1H1/////40X,26HM I S S I O N   T R A C E ,                     FORMAT      39
     *      23H  I N F O R M A T I O N/40X,13(1H-),3X,9(1H-),3X,21(1H-)     FORMAT      40
     *      //41X,41HFIELDS    SYMBOL    U S E / M E A N I N G//           FORMAT      41
     *      43X,3H1,2,15X,27HTIME IN MINUTES AND SECONDS//                  FORMAT      42
     *      44X,1H3,16X,25HCURRENT OPERATOR JOB AREA)                       FORMAT      43
1009  FORMAT(53X,3HSER,8X,12HSEARCH SCOPE/                                  FORMAT      44
     *      53X,3HIDL,8X,9HIDLE TIME/                                       FORMAT      45
     *      53X,3HOBR,8X,21HOBSERVE/PROCESS VIDEO/                          FORMAT      46
     *      53X,3HOBU,8X,29HOBSERVE/PROCESS UNKNOWN TRACK/                  FORMAT      47
     *      53X,3HOBF,8X,30HOBSERVE/PROCESS FRIENDLY TRACK/                 FORMAT      48
     *      53X,3HOBH,8X,29HOBSERVE/PROCESS HOSTILE TRACK/                  FORMAT      49
     *      53X,3HASS,8X,25HASSIGN FIRE UNIT TO TRACK/                      FORMAT      50
     *      53X,3HOFU,8X,25HOBSERVE/PROCESS FIRE UNIT/                      FORMAT      51
     *      56X,1H*,7X,23HHOOKING A SITE OR TRACK)                          FORMAT      52
                                                                            FORMAT      53
1010  FORMAT(/44X,1H4,16X,17HSAINT TASK NUMBER//                            FORMAT      54
     *      44X,1H5,8X,2HTR,6X,35HTRACK NUMBER ASSOCIATED WITH ACTION/      FORMAT      55
     *      44X,9X,2HFU,6X,35HFIRE UNIT NO ASSOCIATED WITH ACTION//         FORMAT      56
     *      44X,1H6,16X,15HSTATUS OF TRACK/                                 FORMAT      57
     *      54X,1HR,9X,5HVIDEO/                                             FORMAT      58
     *      54X,1HU,9X,13HUNKNOWN TRACK)                                    FORMAT      59
1011  FORMAT(54X,1HF,9X,14HFRIENDLY TRACK/                                  FORMAT      60
     *      54X,1HH,9X,13HHOSTILE TRACK/                                    FORMAT      61
     *      54X,1HS,9X,14HSPECIAL SYMBOL//                                  FORMAT      62
     *      61X,19HSTATUS OF FIRE UNIT/                                     FORMAT      63
     *      54X,1HU,9X,6HUNUSED/                                            FORMAT      64
     *      54X,1HA,9X,8HACCESSED/                                          FORMAT      65
     *      54X,1HX,9X,7HENGAGED)                                           FORMAT      66
1012  FORMAT(54X,1HF,9X,6HFIRING/                                           FORMAT      67
     *      54X,1HE,9X,9HEFFECTIVE/                                         FORMAT      68
     *      54X,1HI,9X,11HINEFFECTIVE/                                      FORMAT      69
     *      54X,1HZ,9X,15HNOT OPERATIONAL/                                  FORMAT      70
     *      54X,1HD,9X,9HDISENGAGE/                                         FORMAT      71
     *      54X,1HC,9X,10HCEASE FIRE/                                       FORMAT      72
     *      54X,1H*,9X,24HBLINKING (OUT OF ACTION))                         FORMAT      73
1013  FORMAT(/44X,1H7,16X,16HTRACK - DISTANCE/                              FORMAT      74
     *      61X,30HFIRE UNIT - PRIMARY ASSIGNMENT//                         FORMAT      75
     *      44X,1H8,16X,26HTRACK - ATTACHED FIRE UNIT/                      FORMAT      76
     *      61X,32HFIRE UNIT - SECONDARY ASSIGNMENT//                       FORMAT      77
     *      44X,1H9,7X,26H(SEE 6)  ALL TRACKS STATUS//                      FORMAT      78
     *      43X,2H10,7X,30H(SEE 6)  ALL FIRE UNITS STATUS)                  FORMAT      79
                                                                            FORMAT      80
      REAL DAT(5)                                                           UECHO        6
      INTEGER IDC(8),ID(2),ALO(12),ALP(26),IT(6)                            UECHO        7
                                                                            UECHO        8
      DATA IDC/4H VID,4HEO  ,4HUNKN,4HOWN ,4HFRIE,4HNDLY,                   UECHO        9
     *      4HHOST,4HILE /                                                  UECHO       10
      DATA ALP/4HAUTO,4H    ,4HMANU,4HAL  ,4HTIGH,4HT   ,4HFREE,4H    ,    UECHO       11
     *      4HSEQU,4HENCE,4HPOSI,4HTION,4HTAB ,4H    ,4HFU  ,4H    ,        UECHO       12
     *      4HTR  ,4H    ,4HHT  ,4H    ,4HTR -,4H FU ,4HHT -,4H FU ,        UECHO       13
     *      4H    ,4H    /                                                  UECHO       14
                                                                            UECHO       15
      IT(1) = 3                                                             UECHO       16
      IF(AUTOI) IT(1) = 1                                                   UECHO       17
      IT(2) = 3                                                             UECHO       18
      IF(AUTOR) IT(2) = 1                                                   UECHO       19
      IT(3) = 3                                                             UECHO       20
      IF(AUTOE)  IT(3) = 1                                                  UECHO       21
      IT(4) = 7                                                             UECHO       22
      IF(TIGH)  IT(4) = 5                                                   UECHO       23
      IT(5) = 13                                                            UECHO       24
      IF(TYHOOK .EQ. 0)  IT(5) = 9                                          UECHO       25
      IF(TYHOOK .EQ. 1)  IT(5) = 11                                         UECHO       26
      IT(6) = 17                                                            UECHO       27
      IF(SEQT .EQ. 0)  IT(6) = 21                                           UECHO       28
      IF(SEQT .EQ. 2)  IT(6) = 15                                           UECHO       29
      IF(SEQT .EQ. 3)  IT(6) = 23                                           UECHO       30
      IF(SEQT .EQ. 4)  IT(6) = 19                                           UECHO       31
      IF(TYHOOK .NE. 0)  IT(6) = 25                                         UECHO       32
```

Figure 1(12). Program Listing: Support Programs

```
                                                                    UECHO      34
      DO 5 I = 1,6                                                  UECHO      35
         J = I + I - 1                                              UECHO      36
         ALO(J) = ALP(IT(I))                                        UECHO      37
         ALO(J + 1) = ALP(IT(I) + 1)                                UECHO      38
 5    CONTINUE                                                      UECHO      39
      WRITE(6,1000)                                                 UECHO      40
      WRITE(6,1001) ALO                                             UECHO      41
      WRITE(6,1002)                                                 UECHO      42
      DO 10 I = 1,NFU                                               UECHO      43
         IA = FUCLA(I,8)                                            UECHO      44
10       WRITE(6,1003) I,FUCLA(I,4),FUCLA(I,5),IA,FUCLA(I,9)       UECHO      45
      JLEG = 1                                                      UECHO      46
      WRITE(6,1004)                                                 UECHO      47
      LPAGE = 5                                                     UECHO      48
                                                                    UECHO      49
      DO 31 I = 1,NTRK                                              UECHO      50
         IF(JLEG .NE. 1)  GO TO 20                                  UECHO      51
            WRITE(6,1006)                                           UECHO      52
            IF(LPAGE .GT. 50)  LPAGE = 0                            UECHO      53
            IF(LPAGE .EQ.0)  WRITE(6,1007)                          UECHO      54
            ITPR = PTR(I)                                           UECHO      55
            ITPT = PTT(I)                                           UECHO      56
            T = TRROU(ITPR,1)                                       UECHO      57
            ID(1) = IDC(1)                                          UECHO      58
            ID(2) = IDC(2)                                          UECHO      59
            DAT(1) = TRROU(ITPR,2)                                  UECHO      60
            DAT(2) = TRROU(ITPR,3)                                  UECHO      61
            ITPR = TRROU(ITPR,4)                                    UECHO      62
            DAT(3) = TRROU(ITPR,2)                                  UECHO      63
            DAT(4) = TRROU(ITPR,3)                                  UECHO      64
            DAT(5) = ((DAT(3)**2 + DAT(4)**2)**.5) * 3600.          UECHO      65
            IA = AHEAD(DAT(3),DAT(4))                               UECHO      66
            JLEG = 2                                                UECHO      67
            ST = TRROU(ITPR,1)                                      UECHO      68
      ITT = 0                                                       UECHO      69
      IF(LPAGE.EQ. 0)  LPAGE = 2                                    UECHO      70
            GO TO 29                                                UECHO      71
20       CONTINUE                                                   UECHO      72
         IF(ITT .EQ. 1)  GO TO 23                                   UECHO      73
               IF(ST .LE. TRTYP(ITPT,1))  GO TO 22                  UECHO      74
21                TL = T                                            UECHO      75
                  T = TRTYP(ITPT,1)                                 UECHO      76
                  IF(T .GT. 4000.)  GO TO 31                        UECHO      77
                  IK = (TRTYP(ITPT,2) * 2.) - 1.                    UECHO      78
                  ID(1) = IDC(IK)                                   UECHO      79
                  ID(2) = IDC(IK + 1)                               UECHO      80
                  CALL LOC(TL,T,DAT(1),DAT(2),DAT(3),DAT(4))        UECHO      81
                  ITPT = TRTYP(ITPT,3)                              UECHO      82
                  ST = TRTYP(ITPT,1)                                UECHO      83
               ITT = 1                                              UECHO      84
                  GO TO 30                                          UECHO      85
22             CONTINUE                                           UECHO      86
                  TL = T                                            UECHO      87
                  T = TRROU(ITPR,1)                                 UECHO      88
                  IF(T .GT. 4000.)  GO TO 31                        UECHO      89
                  ITPR = TRROU(ITPR,4)                              UECHO      90
      CALL LOC(TL,T,DAT(1),DAT(2),DAT(3),DAT(4))                    ERR2        1
                  DAT(3) = TRROU(ITPR,2)                            UECHO      91
                  DAT(4) = TRROU(ITPR,3)                            UECHO      92
                  DAT(5) = ((DAT(3)**2 + DAT(4)**2)**.5) * 3600.    UECHO      94
                  IA = AHEAD(DAT(3),DAT(4))                         UECHO      95
                  ST = TRROU(ITPR,1)                                UECHO      96
               ITT = 0                                              UECHO      97
                  GO TO 30                                          UECHO      98
                                                                    UECHO      99
```

Figure 1(13). Program Listing: Support Programs

```
23        CONTINUE                                                    UECHO     100
             IF(ST .LT. TRROU(ITPR,1))  GO TO 21                      UECHO     101
             GO TO 22                                                 UECHO     102
                                                                      UECHO     103
29        WRITE(6,1005)I,T,ID,DAT,IA                                  UECHO     104
          LPAGE = LPAGE + 1                                           UECHO     105
          GO TO 20                                                    UECHO     106
30        WRITE(6,1105) T,ID,DAT,IA                                   UECHO     107
          LPAGE = LPAGE + 1                                           UECHO     108
          GO TO 20                                                    UECHO     109
31   JLEG = 1                                                         UECHO     110
     LPAGE = 0                                                        UECHO     111
     WRITE(6,1008)                                                    UECHO     112
     WRITE(6,1009)                                                    UECHO     113
     WRITE(6,1010)                                                    UECHO     114
     WRITE(6,1011)                                                    UECHO     115
     WRITE(6,1012)                                                    UECHO     116
     WRITE(6,1013)                                                    UECHO     117
     RETURN                                                           UECHO     118
     END                                                              UECHO     119
```

```
      SUBROUTINE UIN                                                  UINPT     1
      REAL TRCLA(33,5),FUCLA(11,9)                                    UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                      UCOM1     2
                                                                      UCOM1     3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)           UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                          UCOM2     2
                                                                      UCOM2     3
      INTEGER PAIR(33),PTR(33),PTT(33),RSTAT(33)                      UCOM3     1
      COMMON /UCOM3/ PTR,PTT,RSTAT,PAIR                               UCOM3     2
                                                                      UCOM3     3
      LOGICAL AUTOI,AUTOR,AUTOE,TIGH                                  UCOM4     1
      COMMON /UCOM4/ AUTOI,AUTOR,AUTOE,TIGH                           UCOM4     2
                                                                      UCOM4     3
      REAL VALUE(20),STI(20),STOT                                     UCOM5     1
      COMMON /UCOM5/ VALUE,STI,STOT                                   UCOM5     2
                                                                      UCOM5     3
      INTEGER TYHOOK,SEQT,PSEQ                                        UCOM6     1
      COMMON /UCOM6/ TYHOOK,SEQT,PSEQ                                 UCOM6     2
                                                                      UCOM6     3
      INTEGER NFU,NTRFU,NTRK                                          UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                   UCOM7     2
                                                                      UCOM7     3
                                                                      UCOM7     4
                                                                      UCOM7     5
      REAL CX(33),CY(33)                                              UCOM8     1
      INTEGER IPTR(33),IPTT(33)                                       UCOM8     2
      COMMON /UCOM8/ CX,CY,IPTR,IPTT,IPC                              UCOM8     3
                                                                      UCOM8     4
      LOGICAL TRCH                                                    UCOM9     1
      REAL TRMOD(33),TOTRT(33),TMARK,TMARE                            UCOM9     2
      INTEGER NOLDTY,LPAGE                                            UCOM9     3
      COMMON /UCOM9/ TRCH,TRMOD,TOTRT,TMARK,TMARE,LPAGE,NOLDTY        UCOM9     4
      REAL TFUN(10)                                                   UINPT     3
      COMMON /UCOM0/ TFUN                                             UINPT     4
                                                                      UINPT     5
      DATA IFI/0/                                                     UINPT     6
                                                                      UINPT     7
```

Figure 1(14). Program Listing: Support Programs

```
      PSEQ = 0                                                          UINPT      8
      IF(IFI .GT. 0)  GO TO 60                                          UINPT      9
         IFI = 1                                                        UINPT     10
         READ(5,1000) AUTOI,AUTOR,AUTOE,TIGH                            UINPT     11
         READ(5,1001) TYHOOK,SEQT                                       UINPT     12
         READ(5,1001) NFU,NTRK                                          UINPT     13
         NTRFU = NFU + NTRK                                             UINPT     14
                                                                        UINPT     15
         DO 10 I = 1,NFU                                                UINPT     16
         READ(5,1002) FUCLA(I,4),FUCLA(I,5),FUCLA(I,8),FUCLA(I,9)       UINPT     17
         TFUN(I) = FUCLA(I,8)                                           UINPT     18
            FUCLA(I,1) = 1.                                             UINPT     19
   10       TRSTA(I,1) = -I                                             UINPT     20
                                                                        UINPT     21
         K = 0                                                          UINPT     22
         I = 0                                                          UINPT     23
   30    K = K + 1                                                      UINPT     24
         READ(5,1003) J,TRROU(K,1),TRROU(K,2),TRROU(K,3)                UINPT     25
         TRROU(K,4) = K + 1                                             UINPT     26
         IF(J .EQ. I)  GO TO 30                                         UINPT     27
            IF(J .GT. NTRK)  GO TO 40                                   UINPT     28
            I = J                                                       UINPT     29
            IPTR(I) = K                                                 UINPT     30
            CX(I) = TRROU(K,2)                                          UINPT     31
            CY(I) = TRROU(K,3)                                          UINPT     32
            GO TO 30                                                    UINPT     33
                                                                        UINPT     34
   40    K = 0                                                          UINPT     35
         I = 0                                                          UINPT     36
   50    K = K + 1                                                      UINPT     37
         READ(5,1003) J,TRTYP(K,1),TRTYP(K,2)                           UINPT     38
            TRTYP(K,3) = K + 1                                          UINPT     39
            IF(J .EQ.I)  GO TO 50                                       UINPT     40
            IF(J .GT. NTRK)  GO TO 60                                   UINPT     41
            I = J                                                       UINPT     42
            IPTT(I) = K                                                 UINPT     43
            GO TO 50                                                    UINPT     44
                                                                        UINPT     45
   60 CONTINUE                                                          UINPT     46
      CALL RSTART                                                       UINPT     47
      CALL UECHO                                                        UINPT     48
      RETURN                                                            UINPT     49
                                                                        UINPT     50
 1000 FORMAT(4L1)                                                       UINPT     51
 1001 FORMAT(2I5)                                                       UINPT     52
 1002 FORMAT(4F10.0)                                                    UINPT     53
 1003 FORMAT(I2,3F10.0)                                                 UINPT     54
                                                                        UINPT     55
      END                                                               UINPT     56
      SUBROUTINE UINPT                                                  UINPT     57
      RETURN                                                            UINPT     58
      END                                                               UINPT     59
```

Figure 1(15). Program Listing: Support Programs

```
      FUNCTION UPTR(TRN)                                              UPTR      1
      INTEGER TRN                                                     UPTR      2
      LOGICAL UPTR                                                    UPTR      3
                                                                      UPTR      4
      REAL TRCLA(33,5),FUCLA(11,9)                                    UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                      UCOM1     2
                                                                      UCOM1     3
                                                                      UPTR      6
                                                                      UPTR      7
C     CHECK IF OBSERVED .EQ. REAL                                     UPTR      8
      UPTR = .FALSE.                                                  UPTR      9
      IF(TRCLA(TRN,1) .NE. TRCLA(TRN,2))  GO TO 10                    UPTR     10
                                                                      UPTR     11
C     THEY ARE THE SAME                                               UPTR     12
      UPTR = .TRUE.                                                   UPTR     13
         RETURN                                                       UPTR     14
                                                                      UPTR     15
C     THEY ARE DIFFERENT                                              UPTR     16
   10 TRCLA(TRN,1) = TRCLA(TRN,2)                                     UPTR     17
      CALL CONT(TRN)                                                  UPTR     18
      RETURN                                                          UPTR     19
      END                                                             UPTR     20
```

Figure 1(16). Program Listing: Support Programs

```
      BLOCK DATA                                                        BLOCK     1
                                                                        BLOCK     2
      REAL TRCLA(33,5),FUCLA(11,9)                                      UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                        UCOM1     2
                                                                        UCOM1     3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)             UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                            UCOM2     2
                                                                        UCOM2     3
      INTEGER PAIR(33),PTR(33),PTT(33),RSTAT(33)                        UCOM3     1
      COMMON /UCOM3/ PTR,PTT,RSTAT,PAIR                                 UCOM3     2
                                                                        UCOM3     3
      LOGICAL AUTOI,AUTOR,AUTOE,TIGH                                    UCOM4     1
      COMMON /UCOM4/ AUTOI,AUTOR,AUTOE,TIGH                             UCOM4     2
                                                                        UCOM4     3
      REAL VALUE(20),STI(20),STOT                                       UCOM5     1
      COMMON /UCOM5/ VALUE,STI,STOT                                     UCOM5     2
                                                                        UCOM5     3
      INTEGER TYHOOK,SEQT,PSEQ                                          UCOM6     1
      COMMON /UCOM6/ TYHOOK,SEQT,PSEQ                                   UCOM6     2
                                                                        UCOM6     3
      INTEGER NFU,NTRFU,NTRK                                            UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                     UCOM7     2
                                                                        UCOM7     3
                                                                        UCOM7     4
                                                                        UCOM7     5
      REAL CX(33),CY(33)                                                UCOM8     1
      INTEGER IPTR(33),IPTT(33)                                         UCOM8     2
      COMMON /UCOM8/ CX,CY,IPTR,IPTT,IPC                                UCOM8     3
                                                                        UCOM8     4
      LOGICAL TRCH                                                      UCOM9     1
      REAL TRMOD(33),TOTRT(33),TMARK,TMARE                              UCOM9     2
      INTEGER NOLDTY,LPAGE                                              UCOM9     3
      COMMON /UCOM9/ TRCH,TRMOD,TOTRT,TMARK,TMARE,LPAGE,NOLDTY          UCOM9     4
                                                                        BLOCK     4
      DATA VALUE/1.,6.,4.,2.,7.,9.,4.,1.,5.,7.,3.,                      BLOCK     5
     *         8.,9.,7.,5.,9.,1.,3.,6.,0./                             BLOCK     6
                                                                        BLOCK     7
       DATA STI/2.,2.,6*3.,4*4.,6.,5.,                                  BLOCK     8
     *         5.,6.,1.,3.,9.,0./                                      BLOCK     9
                                                                        BLOCK    10
      DATA TRCLA/165*0./                                                BLOCK    11
                                                                        BLOCK    12
      DATA FUCLA/99*0./                                                 BLOCK    13
                                                                        BLOCK    14
      DATA TRSTA/132*0./                                                BLOCK    15
                                                                        BLOCK    16
      DATA PAIR/33*11/                                                  BLOCK    17
                                                                        BLOCK    18
      DATA RSTAT/33*4/                                                  BLOCK    19
                                                                        BLOCK    20
      DATA TYHOOK,SEQT,PSEQ/2,0,0/                                      BLOCK    21
                                                                        BLOCK    22
      DATA AUTOI,AUTOR,AUTOE,TIGH/4*.TRUE./                             BLOCK    23
                                                                        BLOCK    24
      END                                                               BLOCK    25
```

Figure 1(17). Program Listing: Support Programs

User Function 1

User function 1 is called at the completion of task 3. It is used to assign the probabilities used in branching from task 3. If the video data has been observed at a previous time, one set of probabilities is figured. A second set of probabilities is figured if this is the first time this data has been observed. This is determined by a call to function NEWTR. The function makes use of the function BUZY to vary these probabilities. It also uses function STORP to store the set of three probabilities. One value is assigned to the function value, while the other two are stored in system attributes 2 and 3.

User Function 2

User function 2 is called at the completion of task 5 to assign the branching probabilities to the system attributes 1, 2, and 3. There are two sets of probabilities that may be assigned. If the system is operating in the auto-initialization mode, one set of probabilities is figured. A second set of probabilities is assigned if the system is in the manual mode. The probabilities are varied through a call to function BUZY and stored in system attributes 2 and 3, and the function value, through a call to function STORP.

User Function 3

User function 3 is called at the release of task 6. It calculates the performance time of this task. The value

returned is uniformly distributed and has a maximum value of two radar sweeps.

User Function 4

User function 4 is called at the completion of task 8. It is used to update the status of the track on the scope. It calls function UPTR which checks if the observed status is equal to the actual status. If there is a difference, the status is updated. In addition, the value of TRCH is set to true so that the proper statistics can be collected when the operator returns to task 1.

User Function 5

User function 5 is called at the completion of task 9. It is used to assign the branching probabilities to system attributes 1 and 2. One set of probabilities is used if the system is not in automatic identification mode. These probabilities are varied by the function BUZY and by the function RANGF; combining these functions gives probabilities for branching to task 10 that range from small, if the system is very busy and the range is greater than 60 miles, to a value near 1.0, if the system is not busy and the range is less than 40 miles. A second set of probabilities is used if the system is in the automatic identification mode. Here the range of values is from extremely small, if the system is very busy and the range is greater than 60 miles, to a moderate value, if the system is not busy and the range is less than 40 miles.

User Function 6

User function 6 is called at the completion of task 12 to assign the branching probabilities to system attributes 1, 2, and 3. The first set of probabilities reflects no identification change, that is, the track is still an unknown target. This is determined by a call to function UPTR. This first set of probabilities will cause the operator to proceed to either task 14 or task 1. There is a 0 probability of proceeding to task 13. The probabilities are varied through a call to function BUZY and a call to function RANGF. The current engagement mode is also taken into account in calculating the probabilities. A second set of probabilities is used when a change has been made in the classification of the track. In this case, the operator will proceed to task 13 100 percent of the time. In addition, a call to subroutine CONT is made to insure that the operator will continue processing this track under its new classification. The variable TRCH is set to TRUE so that statistics can be collected upon returning to task 1.

User Function 7

User function 7 is called at the completion of task 14 to assign the branching probabilities to system attributes 1 and 2. There is only one set of probabilities that may be assigned. They are varied by a call to BUZY and a factor that is dependent on the current engagement mode status of the system. In addition, the tight/free engagement policy is checked.

System attribute 7 is set to 0 if that policy is free and it is set to 1 if that policy is tight.

User Function 8

User function 8 is called at the completion of task 33 to assign the branching probabilities to system attributes 1 and 2. This function checks all fire units to find one that has an effective status showing on the DDG. If no effective status is found, the probability of returning to task 1 is very high. This is set by a call to STORP. If a fire unit is found with an effective status, the value that is assigned to system attribute 2 is very small. Therefore, the chance of returning to task 1 is very small and the operator is more likely to proceed to tasks 35 and 34 to clear the effective status of the fire unit.

User Function 9

User function 9 is called at the completion of task 34. It has two functions. First, it clears the effective status from the fire unit status table. Second, it stores the fire unit number in information attribute 2. This is used by the fire unit section to clear effective status and possibly initiate a secondary engagement.

User Function 10

User function 10 is called at the completion of task 18 to assign the branching probabilities to system attributes 1, 2, and 3. These values are set by a call to function STORP.

User Function 11

User function 11 is called at the completion of task 21 to assign branching probabilities to system attributes 1 and 2. The function checks to see if the track was last classified as a hostile track. If it was, the probability is high that the operator will proceed with tasks 22 and 23 to determine if the track was assigned to a fire unit. If the track was not previously a hostile track, the operator will return to task 1 by assigning the value of 1 to system attribute 1. In either case, the last observed status is updated to the current status.

User Function 12

User function 12 is called at the completion of task 22 to assign branching probabilities to system attributes 1 and 2. This function checks to see if a fire unit has been assigned to this friendly track. If it has, a high probability is set in system attribute 2 so that the operator will proceed and clear the engagement. If no fire unit has been assigned, a low probability is set in system attribute 2 so that the operator will return to task 1.

User Function 13

User function 13 is called at the completion of task 25 to assign branching probabilities to system attributes 1, 2, and 3. This function updates the last observed status of this track. It then checks to see if the track is currently assigned to a fire unit or if the track is currently under a hold fire order. The hold fire order would have originated if

the track was engaged as an unknown while the system was in the tight engagement mode. Since the track is now hostile, the operator should clear this status.

There are three sets of probabilities that may be assigned. The first would reflect the fact that the target is not engaged. The probability is varied by a call to function RANGF and by multiplication by an automatic-engaged factor. The probability that the operator will proceed with an assignment may vary from 0 if the target is outside 50 miles to relative certainty if the target is within 50 miles. The second set of probabilities is assigned if the system determines that the track has received a hold fire order. In this case, the operator will be sent to task 26 with a very high probability. This will insure that the operator clears the hold fire status. The third set of probabilities is when the track is already engaged. In this case, the probability of 1.0 is given to system attribute 1 insuring that the operator will return to task 1.

User Function 14

User function 14 is called at the completion of task 27. This function is used to store the fire unit number in information attribute 2. This will then be used by the fire unit section to process the clear hold fire message that is sent from this task.

User Function 15

User function 15 is called at the completion of task 28 to assign branching probabilities to system attributes 1, 2, and 3.

This function first checks to see if the fire unit is operational. If it is, it will direct the operator back to task 1. Second, it checks to see if the fire unit is blinking. If it is, it will direct the operator to task 35 and then to task 29 to reassign the tracks that are currently engaged by this fire unit. In addition, it will change the condition from blinking to not operational. If the site is not blinking, it will assign probabilities so that there is a large possibility of proceeding to task 33 to observe the DDG and clear any effective status.

User Function 16

User function 16 is called at the completion of task 29. It is used to assign values to system attribute 1, which is used for the conditional branching from this task. The value of system attribute 1 will be 0 if there are no tracks assigned to this fire unit, 1 if there is a primary site only assigned to this fire unit, and 2 if there are both primary and secondary tracks assigned to this fire unit.

User Function 17

User function 17 is called at the completion of task 30. This function clears the fire unit from the fire unit status array.

User Function 18

User function 18 is called at the release of task 31 to assign a task performance time. This time is uniformly distri-

buted so that the maximum value is the time of 2 radar sweeps. In addition, the track is flagged so that the operator will continue to process the unassigned track after completing the current task of dropping the fire unit. This is to insure that the track will be reengaged if necessary. The function also assigns the track and fire unit numbers to information attributes 1 and 2 so that the disengagement message can be processed by the fire unit section.

User Function 19

User function 19 is called at the completion of task 32. It is used to flag the primary track for continued processing by the operator. It also assigns the track and fire unit numbers to information attributes 1 and 2 so that the fire unit section can process the cease engagement message correctly.

User Function 20

User function 20 is called at the completion of task 35 to assign a value to system attribute 1. This value is then used in the conditional branching from this task. The function sets a value of 2 if the operator will use the tab hook method. It sets a value of 1 if the method is the number or position hook. If a sequence hook is to be used, it checks to see if the unit to be hooked is of the proper type. That is, it may be requesting a sequence hook for tracks but the site being hooked is indeed a fire unit. In this case, the method defaults to tab hook. If the type of sequence hook matches

the object being hooked, a value of 0 is set to route the operator to the sequence hook procedures.

User Function 21

User function 21 is called at the completion of task 36. It is used to assign values to system attribute 1 which is used for the conditional branching. The function returns a value of 1 if the object being hooked matches the category currently being used by this system. If the category does not match, then a value of 0 is assigned.

User Function 22

User function 22 is called at the completion of task 37. This function records the category required for the sequence hook.

User Function 23

User function 23 is called at the completion of task 38. It assigns a value to system attribute 1 which is used for the conditional branching from this task. By a call to function NHOOK the action of pressing TASK FUNCTION - SEQHOOK is simulated. If no symbols of the category requested are found, the function is set to 1. If a symbol of the correct type is found but it is not the track or fire unit desired, the function is set to 0. If the correct track is found, the function is set to 2.

User Function 24

User function 24 is called at the completion of task 47. It assigns a value to system attribute 8 which is used for the conditional branching from this task. The function first checks to see if there was actually a fire unit assigned. If information attribute 2 is equal to 0, then there were no fire units available at the time of assignment and nothing further is done. If there was a fire unit assigned, the fire unit and track are paired. This causes the range of the SS variables to change from giving the distance of the track to the center of the system to giving the distance of the track to the fire unit. Next, the function checks to see if the fire unit is still assigned to the track. If it is not, the function value is set to 0 and variable RSTAT is set to 4, indicating that the track is not attached. If the fire unit is still attached, the value of the function is set to 1 and the fire unit status is set to 3, engaged.

User Function 25

User function 25 is called at the completion of task 48. It assigns a value to system attribute 8 which is used in the conditional branching from this task. This function checks to see if the track is still engaged to the fire unit. If it is not, the value of the function is set to -2 and the value of variable RSTAT is set to 4, indicating that the track is no longer engaged. If the track is engaged, a check is made to see if it is a secondary or primary track. If it is a primary track, a further check is made to see if it is within

firing range. If the track is within firing range, the function is set to 1. If it is a secondary track, the fire unit status is updated to indicate the holding of a track and the function is set to -1. If the track is primary but not within firing range, variable RSTAT is set to 2, indicating that the track is being held for distance reasons and the function is set to 0.

User Function 26

User function 26 is called at the completion of task 49. It assigns a value to system attribute 8 which is used for the conditional branching from this task. The function first checks to see if the fire unit and track are still assigned. If they are not, then the function is set to -1 and variable RSTAT is set to 4. If the fire unit is still engaged to the track, then the fire unit is checked for a hold fire message. If there is a hold fire message, the fire unit status is updated and variable RSTAT is set to 3. In addition, the function is set to 0. If the engagement is to continue, the fire unit status is set to 4, indicating the firing process, and the function is set to 1.

User Function 27

User function 27 is called at the completion of task 51. It assigns a value to system attribute 8 which is used for the conditional branching from this task. This function first checks to see if a cease fire message was received. If no cease fire message was received, it makes a random check to

see if the firing of the missile was effective. This is done by comparing the effectiveness ratio with a uniformly distributed random variable. When the firing was effective an effective status is set in the fire unit status array and the track status is set to 0. If the firing was not effective, a check is made to see if the minimum distance is greater than 35. This check will determine if the target is out of range due to a change in course or flying past the firing unit. If the missile is still within range and the firing was ineffective, a second firing is initiated. This is accomplished by setting the function value to 1. If the track is no longer in range or a cease fire message was received, the value of the function is set to 2; the track status and the fire unit status are updated and the value of variable RSTAT is set to 4 to indicate that the track is no longer engaged. After all of these checks have been made, the fire unit is checked to see if it has any missiles remaining. If there are none, the status of the fire unit is changed to blinking. If this is done, the value of the function is reset to 0. This evaluation overrides any of the previous evaluations but does not change the status of the track as it was previously set.

User Function 28

User function 28 is called at the completion of task 53. It assigns a value to system attribute 8 which is used for the conditional branching from this task. The function first reclassifies the primary track as not assigned. This may not be necessary if the primary track was effectively shot down in

task 51, however, if this task is the result of a clear cease fire message, this will not have been done. Next, the function checks to see if there is a secondary track that is being held by the firing unit. If there is, the fire unit status is updated by changing the secondary assignment to the primary assignment and reengaging the track by setting the function value to 1. If there is no secondary assignment, the primary assignment is cleared and the function value is set to 0.

User Function 29

User function 29 is called at the completion of task 54. This function sets a flag in the fire unit status array. This flag will later indicate that a hold fire message has been received.

User Function 30

User function 30 is called at the completion of task 55. It assigns a value to system attribute 8 which is used for the conditional branching from this task. This function first checks to see if the fire unit status array indicates that a target is currently being held under a hold fire order. If nothing is being held, the hold fire flag is cleared and the function value is set to 0. If a track is being held, information attribute 1 is given the track number, the value of subroutine RSTAT is set to 1, indicating that the track is engaged and the value of the function is set to 1 to route a message to task 49 for the reengagement.

User Function 31

User function 31 is called at the completion of task 57. It assigns a value to system attribute 8 which is used for the conditional branching from this task. The function clears the track status array and then checks to see if the track is the primary or secondary assignment for the fire unit. If it is the primary assignment, a further check is made to see if the fire unit is currently firing at the target. If it is not firing at the target, the value of the function is set to 0 which will send a message to task 53 for the possible engagement of the secondary assignment. If it is firing at the target, a flag is set in the fire unit status array and the function value is set to 1. This will cause task 54, the evaluation of the firing, to make that evaluation under the restrictions of the cease fire message. If the target is the secondary track, the fire unit status array is cleared, the value of variable RSTAT is set to 4 indicating that the target is no longer assigned and the value of the function is set to 1.

User Function 32

User function 32 is called at the completion of task 61. This function is used to initialize the value of variable LTRN which is a counter used in task 63. It represents the track numbers to this system processing area.

User Function 33

User function 33 is called at the completion of task 63. It assigns a value to system attribute 6 which is used for the conditional branching from this task. This function represents four automatic procedures that the system or fire unit may be using. The first area checks all engaged tracks. If the engaged track is friendly, a situation that would result from a change of identification, a cease fire message is initiated to the fire unit. The second section represents a fire unit action. If the fire unit is holding fire because the target is currently not within its firing range, then a check is made to see if the minimum distance is greater than 30 miles. This is to see if the track has changed course and will no longer pass within range of the fire unit or that the track has flown past the fire unit. Next, the check is made to see if the target is within range. This is a random check that is governed by distribution set 16. If these conditions are satisfied, a reengagement message is sent. This is accomplished by assigning to information attribute 1 the track number, information attribute 2 the fire unit number and information attribute 3 the value 6. If the track will not come within range, a message is sent to the fire units to engage a possible secondary target. This is accomplished by setting information attribute 3 to the value 1. The third section is used to check and possibly update the status of an engaged track that is currently under a hold fire order. If the status is now free, a cancel hold fire message is sent. If the identification of the target had changed from unknown

to hostile and the target is within range, a clear hold fire message is sent. This message is indicated by assigning the value 5 to information attribute 3. The fourth section of this function checks nonengaged tracks for possible engagement. This would be true if they are hostile and within the range defined by distribution set 15. It would also be true if they are an unknown track and within the range defined by distribution set 16. If no action was taken, a value of 2 is returned. This sends the system back to task 61. If any action was taken, a value of 1 or 0 is returned. This will send a message to the fire unit section and it will also route the system back to this task, task 63. In addition, if a hold fire message is needed, that is, the target is an unknown target and the system is tight, the value will be set to 0 which will route the system to task 64 to send a hold fire message. In addition, the value of system attribute 10 will be set to 1 if an engagement message of any type is sent to the fire units. This will direct the system to task 75.

User Function 34

User function 34 is called at the completion of task 65. It assigns a value to system attribute 9 which is used for the conditional branching from this task. This task assigns the aircraft number to information attribute 1 for each aircraft. It does this by branching to itself if the count on the aircraft is less than the total number of aircraft required. This is accomplished by assigning 0 to the value of the function.

Once the total number of aircraft has been processed, the value of 1 is assigned to the function to terminate this activity.

User Function 35

User function 35 is called at the completion of task 66. This function assigns the initial SS values for each aircraft, that is, the location where the aircraft first appears on the radar screen. It also updates the pointer for this track which will cause the track to begin movement and it assigns a video status to each track.

User Function 36

User function 36 is called at the completion of task 67. This function updates the pointer so that the velocity vectors for the next leg will be used by subroutine STATE.

User Function 37

User function 37 is called at the completion of task 68. It assigns a value to system attribute 9 which is used for the conditional branching from this task. This function updates the true identification of the track as well as updating the status on the screen if the proper automatic modes are in effect. In addition, if the change is reflected on the radar screen, a value of 1 is given to system attribute 10. This will direct the system to task 75.

User Function 38

User function 38 is called at the release of task 67 to assign the task performance time. This task performance time is the difference in time until the next route update is required.

User Function 39

User function 39 is called at the release of task 68 to assign the task performance time. This task performance time is the difference in time until the next identification status update is required.

User Function 40

User function 40 is called at the release of task 2 to assign the task performance time. At present, the task performance time is uniformly distributed 0 to 10.

User Function 41

User function 41 is called at the completion of task 6. It assigns a value to system attribute 1 which is used for the conditional branching from this task. This function returns the current track identification.

User Function 42

User function 42 is called at the completion of task 18. It is used to assign a track to a fire unit by returning the fire unit number to be attached. This is accomplished by

calling subroutine ASSIG. A value of 0 would indicate no fire unit was available.

User Function 43

User function 43 is not used.

User Function 44

User function 44 is not used.

User Function 45

User function 45 is called at the completion of task 75. It assigns a value to system attribute 5 which is used for the conditional branching from this task. This function determines if the current track being processed by the operator is the same as the one that was automatically updated by tasks 63 or 68. It also checks if the hooking procedures were used. If the first condition was satisfied, the function returns the value 2, 3 or 4 depending on the updated identification status of the aircraft. If both conditions were satisfied, the function returns the value 5, if neither condition was satisfied, the function returns the value 6.

User Function 46

User function 46 is called at the completion of task 79. This function makes use of user function 45 to make the same check on the tracks being processed and returns the same values if the first condition is satisfied. If this condition is not satisfied, the function returns the value 5.

```
      FUNCTION USERF(JJ)                                                    USERF     1
      REAL TRCLA(33,5),FUCLA(11,9)                                          UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                            UCOM1     2
                                                                            UCOM1     3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)                 UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                                UCOM2     2
                                                                            UCOM2     3
      INTEGER PAIR(33),PTR(33),PTT(33),RSTAT(33)                            UCOM3     1
      COMMON /UCOM3/ PTR,PTT,RSTAT,PAIR                                     UCOM3     2
                                                                            UCOM3     3
      LOGICAL AUTOI,AUTOR,AUTOE,TIGH                                        UCOM4     1
      COMMON /UCOM4/ AUTOI,AUTOR,AUTOE,TIGH                                 UCOM4     2
                                                                            UCOM4     3
      REAL VALUE(20),STI(20),STOT                                           UCOM5     1
      COMMON /UCOM5/ VALUE,STI,STOT                                         UCOM5     2
                                                                            UCOM5     3
      INTEGER TYHOOK,SEQT,PSEQ                                              UCOM6     1
      COMMON /UCOM6/ TYHOOK,SEQT,PSEQ                                       UCOM6     2
                                                                            UCOM6     3
      INTEGER NFU,NTRFU,NTRK                                                UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                         UCOM7     2
                                                                            UCOM7     3
                                                                            UCOM7     4
                                                                            UCOM7     5
      REAL CX(33),CY(33)                                                    UCOM8     1
      INTEGER IPTR(33),IPTT(33)                                             UCOM8     2
      COMMON /UCOM8/ CX,CY,IPTR,IPTT,IPC                                    UCOM8     3
                                                                            UCOM8     4
      LOGICAL TRCH                                                          UCOM9     1
      REAL TRMOD(33),TOTRT(33),TMARK,TMARE                                  UCOM9     2
      INTEGER NOLDTY,LPAGE                                                  UCOM9     3
      COMMON /UCOM9/ TRCH,TRMOD,TOTRT,TMARK,TMARE,LPAGE,NOLDTY              UCOM9     4
                                                                            USERF     3
      COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NPUNCH,         COM06     1
     *               NRNIT,NRENT,MNDC,NDC,NDTN,NNTC                         COM06     2
      COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)         COM17     1
      COMMON /COM22/ TTIME,PFIRB                                            COM22     1
                                                                            USERF     7
      LOGICAL DUML,UPTR,ENG,NEWTR                                           USERF     8
                                                                            USERF     9
      GO TO (100,200,300,400,500,600,700,800,900,1000,                      USERF    10
     *       1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,             USERF    11
     *       2100,2200,2300,2400,2500,2600,2700,2800,2900,3000,             USERF    12
     *       3100,3200,3300,3400,3500,3600,3700,3800,3900,4000,             USERF    13
     *       4100,4200,4300,4400,4500,4600),JJ                              USERF    14
                                                                            USERF    15
                                                                            USERF    16
                                                                            USERF    17
                                                                            USERF    18
C     USER FUNCTION 1                                                       USERF    19
                                                                            USERF    20
C     CHECK IF NEW VIDEO                                                    USERF    21
 100  CALL GETIA(1,TRN)                                                     USERF    22
      IF(NEWTR(IFIX(TRN))) GO TO 110                                        USERF    23
                                                                            USERF    24
C     NOT NEW                                                               USERF    25
      USERF = STORP(0.,BUZY(.75,1.),0)                                      USERF    26
      RETURN                                                                USERF    27
                                                                            USERF    28
C     NEW                                                                   USERF    29
 110  USERF = STORP(BUZY(.3,.8),0.,0)                                       USERF    30
      RETURN                                                                USERF    31
                                                                            USERF    32
                                                                            USERF    33
C     USER FUNCTION 2                                                       USERF    34
                                                                            USERF    35
C     CHECK IF AUTO INITIALIZE                                              USERF    36
 200  IF(AUTOI) GO TO 210                                                   USERF    37
                                                                            USERF    38
C     NOT AUTO MODE                                                         USERF    39
        USERF = STORP(0.,BUZY(.8,1.),0)                                     USERF    40
        RETURN                                                              USERF    41
 210  CONTINUE                                                              USERF    42
                                                                            USERF    43
```

Figure 2(1). Program Listing: USERF(JJ)

```
C     AUTO MODE                                                          USERF    44
         USERF = STORP(BUZY(.25,.75),0.,0)                               USERF    45
         RETURN                                                          USERF    46
                                                                         USERF    47
                                                                         USERF    48
C     USER FUNCTION 3                                                    USERF    49
                                                                         USERF    50
C     TWO SWEEP ROTATIONS                                                USERF    51
 300  USERF = 2. * UNFRM(5)                                              USERF    52
      RETURN                                                             USERF    53
                                                                         USERF    54
                                                                         USERF    55
C     USER FUNCTION 4                                                    USERF    56
                                                                         USERF    57
C     UPDATE SYMBOL STATUS                                               USERF    58
 400  CALL GETIA(1,TRN)                                                  USERF    59
      DUML = UPTR(IFIX(TRN))                                             USERF    60
      USERF = 0.                                                         USERF    61
      TRCH = .TRUE.                                                      USERF    62
      RETURN                                                             USERF    63
                                                                         USERF    64
                                                                         USERF    65
C     USER FUNCTION 5                                                    USERF    66
                                                                         USERF    67
                                                                         USERF    68
C      CHECK FOR AUTO INTERROGATE                                        ERR2      1
 500  CALL GETIA(1,TRN)                                                  USERF    70
      ITRN = TRN                                                         USERF    71
      TRCLA(ITRN,3) = TRCLA(ITRN,1)                                      USERF    72
      IF(AUTOR)  GO TO 510                                               ERR2      2
                                                                         USERF    74
C     NOT AUTO ID                                                        USERF    75
         USERF = STORP((BUZY(.9,1.) * RANGF(ITRN,9)),0.,0)               USERF    76
         RETURN                                                          USERF    77
                                                                         USERF    78
C     AUTO ID                                                            USERF    79
 510     USERF = STORP((BUZY(.1,.5) * RANGF(ITRN,9)),0.,0)               USERF    80
         RETURN                                                          USERF    81
                                                                         USERF    82
                                                                         USERF    83
C     USER FUNCTION 6                                                    USERF    84
                                                                         USERF    85
C     CHECK FOR ID CHANGE                                                USERF    86
 600  CALL GETIA(1,TRN)                                                  USERF    87
      ITRN = TRN                                                         USERF    88
      IF(.NOT. UPTR(ITRN))  GO TO 610                                    USERF    89
                                                                         USERF    90
C     NO ID CHANGE                                                       USERF    91
                                                                         USERF    92
C     AUTO ENGAGE FACTOR                                                 USERF    93
         AF = 1.                                                         USERF    94
         IF(AUTOE)  AF = .8                                              USERF    95
                                                                         USERF    96
         USERF = STORP(0.,BUZY(.9,1.) * AF * RANGF(ITRN,12),0)           USERF    97
         RETURN                                                          USERF    98
                                                                         USERF    99
C     ID CHANGE                                                          USERF   100
 610  CALL CONT(IFIX(TRN))                                               USERF   101
      TRCH = .TRUE.                                                      USERF   102
      USERF = STORP(1.,0.,0)                                             USERF   103
      RETURN                                                             USERF   104
                                                                         USERF   105
                                                                         USERF   106
C     USER FUNCTION 7                                                    USERF   107
                                                                         USERF   108
C     SET AUTO ENGAGE FACTOR                                             USERF   109
 700  AF = 1.                                                            USERF   110
      IF(AUTOE)  AF = .9                                                 USERF   111
                                                                         USERF   112
C     CHECK IF TIGHT OF FREE STATUS                                      USERF   113
      IF(TIGH)  GO TO 710                                                USERF   114
                                                                         USERF   115
```

Figure 2(2). Program Listing: USERF(JJ)

```
C     FREE                                                          USERF   116
      CALL PUTSA(7,0.)                                              USERF   117
      USERF = STORP((BUZY(.9,1.) * AF),0.,0)                        USERF   118
      RETURN                                                        USERF   119
                                                                    USERF   120
C     TIGHT                                                         USERF   121
 710  CALL PUTSA(7,1.)                                              USERF   122
      USERF = STORP((BUZY(.9,1.) * AF),0.,0)                        USERF   123
      RETURN                                                        USERF   124
                                                                    USERF   125
                                                                    USERF   126
C     USER FUNCTION 8                                               USERF   127
                                                                    USERF   128
C     CHECK FOR EFFECTIVE STATUS                                    USERF   129
 800  DO 810 I = 1,NFU                                              USERF   130
         IF(FUCLA(I,1) .EQ. 5.)  GO TO 820                          USERF   131
 810  CONTINUE                                                      USERF   132
      GO TO 830                                                     USERF   133
                                                                    USERF   134
C     THERE IS AN EFFECTIVE STATUS                                  USERF   135
 820  USERF = STORP(.1,0.,0)                                        USERF   136
      RETURN                                                        USERF   137
                                                                    USERF   138
C     THERE IS NO EFF STATUS                                        USERF   139
 830  USERF = STORP(1.,0.,0)                                        USERF   140
      RETURN                                                        USERF   141
                                                                    USERF   142
                                                                    USERF   143
C     USER FUNCTION 9                                               USERF   144
                                                                    USERF   145
C     CLEAR EFFECTIVE STATUS                                        USERF   146
 900   DO 910 I = 1,NFU                                             USERF   147
         IF(FUCLA(I,1) .NE. 5.)  GO TO 910                          USERF   148
            FUCLA(I,1) = 1.                                         USERF   149
            GO TO 920                                               USERF   150
 910  CONTINUE                                                      USERF   151
 920  CALL PUTIA(2,FLOAT(I))                                        USERF   152
      CALL PUTIA(1,FUCLA(I,2))                                      ERR2      3
      RETURN                                                        USERF   153
                                                                    USERF   154
                                                                    USERF   155
C     USER FUNCTION 10                                              USERF   156
                                                                    USERF   157
C     STORE BRANCH PROBABILITY                                      USERF   158
1000  USERF = STORP(.8,.15,0)                                       USERF   159
      RETURN                                                        USERF   160
                                                                    USERF   161
                                                                    USERF   162
C     USER FUNCTION 11                                              USERF   163
                                                                    USERF   164
C     CHECK IF LAST HOSTIL                                          USERF   165
1100  CALL GETIA(1,TRN)                                             USERF   166
      ITRN = TRN                                                    USERF   167
      IF(TRCLA(IFIX(TRN),3) .NE. 4.)  GO TO 1110                    USERF   168
                                                                    USERF   169
C     HOSTILE                                                       USERF   170
         USERF = STORP(.95,0.,0)                                    USERF   171
         TRCLA(IFIX(TRN),3) = TRCLA(IFIX(TRN),1)                    USERF   172
      RETURN                                                        USERF   173
                                                                    USERF   174
C     NON HOSTILE                                                   USERF   175
1110     USERF = STORP(0.,0.,0)                                     USERF   176
      TRCLA(ITRN,3) = TRCLA(ITRN,1)                                 USERF   177
      RETURN                                                        USERF   178
                                                                    USERF   179
                                                                    USERF   180
C     USER FUNCTION 12                                              USERF   181
                                                                    USERF   182
C     CHECK FOR ASSIGNED FU                                         USERF   183
1200  FU = 0.                                                       USERF   184
      CALL GETIA(1,TRN)                                             USERF   185
      IF(TRCLA(IFIX(TRN),4) .EQ. 0.)  GO TO 1210                    USERF   186
                                                                    USERF   187
```

Figure 2(3). Program Listing: USERF(JJ)

```
C     FIRE UNIT ASSIGNED                                                    USERF    188
         USERF = STORP(1.,0.,0)                                             USERF    189
         TRCLA(IFIX(TRN),4) = 0.                                            USERF    190
         RETURN                                                             USERF    191
                                                                            USERF    192
C     NO FU ASSIGNED                                                        USERF    193
1210     USERF = STORP(0.,0.,0)                                             USERF    194
         RETURN                                                             USERF    195
                                                                            USERF    196
                                                                            USERF    197
C     USER FUNCTION 13                                                      USERF    198
                                                                            USERF    199
C     SET RANGF FACTOR AND AUTO EXCHANGE FACTOR                             USERF    200
1300  AF = 1.                                                               USERF    201
      CALL GETIA(1,TRN)                                                     USERF    202
      ITRN = TRN                                                            USERF    203
      TRCLA(ITRN,3) = TRCLA(ITRN,1)                                         USERF    204
      IF(TRCLA(IFIX(TRN),4) .NE. 0.)  GO TO 1320                            USERF    205
      IF(AUTOE)  AF = .2                                                    USERF    206
                                                                            USERF    207
C     CHECK FOR HF                                                          USERF    208
      IFUNP = IFIX(TRCLA(IFIX(TRN),4))                                      USERF    209
      IH = 0                                                                USERF    210
      IF(FUCLA(IFUNP,1) .EQ. 9.)  IH = IH + 2                               USERF    211
      IF(IH .GT. 0)  GO TO 1310                                             USERF    212
                                                                            USERF    213
C     NO HOLD FIRE                                                          USERF    214
         USERF = STORP(AF * RANGF(IFIX(TRN),25),0.,0)                       USERF    215
         RETURN                                                             USERF    216
                                                                            USERF    217
C     CLEAR HOLD FIRE                                                       USERF    218
1310     USERF = STORP(0.,1.,0)                                             USERF    219
         RETURN                                                             USERF    220
                                                                            USERF    221
C     ATTACHED RETURN TO SEARCH                                             USERF    222
1320  USERF = STORP(0.,0.,0)                                                USERF    223
      RETURN                                                                USERF    224
                                                                            USERF    225
                                                                            USERF    226
C     USER FUNCTION 14                                                      USERF    227
                                                                            USERF    228
1400  CONTINUE                                                              USERF    229
C     PRIMARY ONLY                                                          USERF    230
         CALL PUTIA(2,FLOAT(IFUNP))                                         USERF    231
         USERF = 0.                                                         USERF    232
         RETURN                                                             USERF    233
                                                                            USERF    234
                                                                            USERF    235
                                                                            USERF    236
                                                                            USERF    237
C     USER FUNCTION 15                                                      USERF    238
                                                                            USERF    239
C     CHECK FU STATUS                                                       USERF    240
C     CHECK IF OPERATIONAL                                                  USERF    241
1500  CALL GETIA(2,FN)                                                      USERF    242
      IFUN = FN                                                             USERF    243
      IF(FUCLA(IFUN,1) .NE. 7.)  GO TO 1510                                 USERF    244
         USERF = STORP(0.,0.,0)                                             USERF    245
         RETURN                                                             USERF    246
                                                                            USERF    247
C     SOTRE TR NO                                                           USERF    248
1510     CALL PUTIA(1,FUCLA(IFUN,2))                                        USERF    249
                                                                            USERF    250
C     CHECK IF BLINKING  SET TO NOT OPERATIONAL                             USERF    251
      IF(FUCLA(IFUN,1) .NE. 10.)  GO TO 1520                                USERF    252
         USERF = STORP(1.,0.,0)                                             USERF    253
         FUCLA(IFUN,1) = 7.                                                 USERF    254
         RETURN                                                             USERF    255
                                                                            USERF    256
C     ALL OTHERS                                                            USERF    257
1520     USERF = STORP(0.,.7,0)                                             USERF    258
         RETURN                                                             USERF    259
                                                                            USERF    260
                                                                            USERF    261
                                                                            USERF    262
```

Figure 2(4). Program Listing: USERF(JJ)

```
C     USER FUNCTION 16                                                  USERF     263
                                                                        USERF     264
C     SET BRANCH IF P OR S ASSIGN EXISTS                                USERF     265
1600  USERF = 0.                                                        USERF     266
      IF(FUCLA(IFUN,2) .NE. 0.)  USERF = 1.                             USERF     267
      IF(FUCLA(IFUN,3) .NE. 0.)  USERF = 2.                             USERF     268
      RETURN                                                            USERF     269
                                                                        USERF     270
                                                                        USERF     271
                                                                        USERF     272
C     USER FUNCTION 17                                                  USERF     273
                                                                        USERF     274
C      SET FIRE UNIT STATUS                                             USERF     275
1700  FUCLA(IFUN,1) = 0.                                                USERF     276
       TRSTA(IFUN,1) = 0.                                               USERF     277
      RETURN                                                            USERF     278
                                                                        USERF     279
                                                                        USERF     280
                                                                        USERF     281
C     USER FUNCTION 18                                                  USERF     282
                                                                        USERF     283
C     SET TIME AT TWO SWEEPS                                            USERF     284
1800  USERF = UNFRM(7) * 2.                                             USERF     285
                                                                        USERF     286
C     CLEAR SECONDARY ASSIGN                                            USERF     287
      ITRN = FUCLA(IFUN,3)                                              USERF     288
      TRCLA(ITRN,4) = 0.                                                USERF     289
      IF(TRCLA(ITRN,1) .NE. 0.)  CALL CONT(ITRN)                        ERR2        4
                                                                        USERF     291
C     SET MESSAGE                                                       USERF     292
      CALL PUTIA(1,FLOAT(ITRN))                                         USERF     293
      CALL PUTIA(2,FN)                                                  USERF     294
      RETURN                                                            USERF     295
                                                                        USERF     296
                                                                        USERF     297
                                                                        USERF     298
C     USER FUNCTION 19                                                  USERF     299
                                                                        USERF     300
C     CLEAR FU                                                          USERF     301
1900  ITRN = FUCLA(IFUN,2)                                              USERF     302
      TRCLA(ITRN,4) = 0                                                 USERF     303
      IF(TRCLA(ITRN,1) .NE. 0.)  CALL CONT(ITRN)                        ERR2        5
                                                                        USERF     305
C     SET MESSAGE                                                       USERF     306
      CALL PUTIA(1,FLOAT(ITRN))                                         USERF     307
      CALL PUTIA(2,FN)                                                  USERF     308
      RETURN                                                            USERF     309
                                                                        USERF     310
                                                                        USERF     311
                                                                        USERF     312
C     USER FUNCTION 20                                                  USERF     313
                                                                        USERF     314
C     CHECK IF TAB OR NUM HOOK IS ALWAYS USED                           USERF     315
2000  IF(TYHOOK .EQ. 0)  GO TO 2010                                     USERF     316
                                                                        USERF     317
C     NUMBER OR SEQUENCE HOOK                                           USERF     318
      USERF = TYHOOK                                                    USERF     319
      RETURN                                                            USERF     320
                                                                        USERF     321
C     SEQUENCE HOOK                                                     USERF     322
C     CHECK IF FOR REQUIRED TYPE                                        USERF     323
2010  CALL GETSA(5,RESQT)                                               USERF     324
      USERF = 2.                                                        USERF     325
                                                                        USERF     326
C     CHECK TRACK                                                       USERF     327
      IF((SEQT .LE. 1) .AND. (RESQT .EQ. 0.))  USERF = 0.               ERR2        6
                                                                        USERF     329
C     CHECK FU                                                          USERF     330
      IF((SEQT .EQ. 0  .OR.  SEQT .EQ. 2  .OR.  SEQT .EQ. 3)           USERF     331
     *      .AND. (RESQT .EQ. 1.))  USERF = 0.                         USERF     332
                                                                        USERF     333
```

Figure 2(5). Program Listing: USERF(JJ)

```
C     CCECK HT                                                    USERF    334
      IF((SEQT .GE. 3) .AND. (RESQT .EQ. 0))  USERF = 0.          USERF    335
                                                                  USERF    336
      RETURN                                                      USERF    337
                                                                  USERF    338
                                                                  USERF    339
                                                                  USERF    340
C     USER FUNCTION 21                                            USERF    341
                                                                  USERF    342
C     CHECK IF CORRECT TYPE                                       USERF    343
2100  USERF = 1                                                   USERF    344
      IF(SEQT .GE. 3)  GO TO 2110                                 USERF    345
                                                                  USERF    346
C     NON HOSTIL TARGET                                           USERF    347
      IF((PSEQ .EQ. 0) .AND. (RESQT .GT. 0.))  USERF = 0.         USERF    348
      IF((PSEQ .GT. 0) .AND. (RESQT .LT. 1.))  USERF = 0.         USERF    349
      RETURN                                                      USERF    350
                                                                  USERF    351
C     HOSTIL TARGET                                               USERF    352
2110  IF((PSEQ .EQ. 2) .AND. (RESQT .GT. 0.))  USERF = 0.         USERF    353
      IF((PSEQ .EQ. 1) .AND. (RESQT .LT. 1.))  USERF = 0.         USERF    354
      RETURN                                                      USERF    355
                                                                  USERF    356
                                                                  USERF    357
                                                                  USERF    358
C     USER FUNCTION 22                                            USERF    359
                                                                  USERF    360
C     UPDATE THE SEQUENCE TYPE                                    USERF    361
2200  USERF = 0.                                                  USERF    362
      PSEQ = IFIX(RESQT)                                          USERF    363
      IF((SEQT .GE.3) .AND. (PSEQ .EQ. 0))  PSEQ = 2              USERF    364
      RETURN                                                      USERF    365
                                                                  USERF    366
                                                                  USERF    367
                                                                  USERF    368
C     USER FUNCTION 23                                            USERF    369
                                                                  USERF    370
C     GET THE NEXT HOOKED ITEM                                    USERF    371
2300  CALL GETIA(1,TRN)                                           USERF    372
      CALL GETIA(2,FNF)                                           USERF    373
      IF(PSEQ .EQ. 1)  TRN = -FNF                                 USERF    374
      IF(NHOOK(PSEQ,TRN)) 2310,2320,2330                          USERF    375
                                                                  USERF    376
C     NOTHING OF THAT TYPE                                        USERF    377
2310  USERF = 1.                                                  USERF    378
      RETURN                                                      USERF    379
                                                                  USERF    380
C     NOT THE CORRECT TRACK                                       USERF    381
2320  USERF = 0.                                                  USERF    382
      RETURN                                                      USERF    383
                                                                  USERF    384
C     CORRECT TRACK                                               USERF    385
2330  USERF = 2.                                                  USERF    386
      RETURN                                                      USERF    387
                                                                  USERF    388
                                                                  USERF    389
                                                                  USERF    390
C     USER FUNCTION 24                                            USERF    391
                                                                  USERF    392
C     CHECK IF FU AVALIABLE                                       USERF    393
2400  CALL GETIA(2,FNF)                                           USERF    394
      IF(FNF .EQ. 0.)  RETURN                                     USERF    395
      CALL GETIA(1,TRNF)                                          USERF    396
      PAIR(IFIX(TRNF)) = IFIX(FNF)                                USERF    397
                                                                  USERF    398
C     CHECK IF TRACK IS STILL ENGAGED                             USERF    399
      USERF = 0.                                                  USERF    400
      IF(ENG(IPS))  GO TO 2410                                    USERF    401
      RSTAT(IFIX(TRNF)) = 4                                       USERF    402
      RETURN                                                      USERF    403
                                                                  USERF    404
```

Figure 2(6). Program Listing: USERF(JJ)

```
C     STILL ACTIVE UPDATE STATUS IF PRIMARY                          USERF    405
2410  USERF = 1.                                                     USERF    406
      IF(IPS .GT. 0)  GO TO 2420                                     USERF    407
      FUCLA(IFIX(FNF),1) = 3.                                        USERF    408
2420  RETURN                                                         USERF    409
                                                                     USERF    410
                                                                     USERF    411
                                                                     USERF    412
C     USER FUNCTION 25                                               USERF    413
                                                                     USERF    414
C     CHECK IF TRACK IS STILL ENGAGED                                USERF    415
2500  USERF = 0.                                                     USERF    416
      IF(ENG(IPS))  GO TO 2510                                       USERF    417
      CALL GETIA(1,TRNF)                                             USERF    418
      RSTAT(IFIX(TRNF)) = 4                                          USERF    419
      USERF = -2.                                                    USERF    420
      RETURN                                                         USERF    421
                                                                     USERF    422
C     STILL ACTIVE UPDATE STATUS IF PRIMARY HOLD SECONDARY           USERF    423
2510  IF(IPS .GT. 0)  GO TO 2520                                     USERF    424
      CALL GETIA(1,TRNF)                                             USERF    425
       IR = TRNF * 3.                                                USERF    426
       IF(SS(IR) .GT. 35.)  GO TO 2530                               USERF    427
      USERF = 1.                                                     USERF    428
      RETURN                                                         USERF    429
                                                                     USERF    430
C     HOLD SECONDARY                                                 USERF    431
2520  CALL GETIA(2,FNF)                                              USERF    432
      FUCLA(IFIX(FNF),3) = -FUCLA(IFIX(FNF),3)                       USERF    433
      USERF = -1.                                                    USERF    434
      RETURN                                                         USERF    435
                                                                     USERF    436
2530  CALL GETIA(2,FNF)                                              USERF    437
      CALL GETIA(1,TRNF)                                             USERF    438
      RSTAT(IFIX(TRNF)) = 2                                          USERF    439
      RETURN                                                         USERF    440
                                                                     USERF    441
                                                                     USERF    442
                                                                     USERF    443
C     USER FUNCTION 26                                               USERF    444
                                                                     USERF    445
C     CHECK FOR CANCEL                                               USERF    446
2600  USERF = 0.                                                     USERF    447
      IF(ENG(IPS))  GO TO 2605                                       USERF    448
      CALL GETIA(1,TRNF)                                             USERF    449
      RSTAT(IFIX(TRNF)) = 4                                          USERF    450
      USERF = -1.                                                    USERF    451
      RETURN                                                         USERF    452
                                                                     USERF    453
C     STILL ACTIVE  CHECK FOR HOLD FIRE                              USERF    454
2605  CALL GETIA(2,FNF)                                              USERF    455
      IF(FUCLA(IFIX(FNF),7) .EQ. 0)  GO TO 2610                      USERF    456
                                                                     USERF    457
C     IS HOLD FIRE  HOLD FU                                          USERF    458
      FUCLA(IFIX(FNF),7) = -1.                                       USERF    459
      CALL GETIA(1,TRNF)                                             USERF    460
      RSTAT(IFIX(TRNF)) = 3                                          USERF    461
      RETURN                                                         USERF    462
                                                                     USERF    463
C     PROCEED UPDATE STATUS                                          USERF    464
2610  FUCLA(IFIX(FNF),1) = 4.                                        USERF    465
      USERF = 1.                                                     USERF    466
      RETURN                                                         USERF    467
                                                                     USERF    468
                                                                     USERF    469
                                                                     USERF    470
C     USER FUNCTION 27                                               USERF    471
                                                                     USERF    472
C     CHECK IF EFFECTIVE                                             USERF    473
2700  CALL GETIA(2,FNF)                                              USERF    474
      CALL GETIA(1,TRNF)                                             USERF    475
      PAIR(IFIX(TRNF)) = 11                                          ERR2       7
      IF(FUCLA(IFIX(FNF),6) .EQ. 1.)  GO TO 2720                     USERF    476
                                                                     USERF    477
```

Figure 2(7). Program Listing: USERF(JJ)

```
      IF(UNFRM(1) .GT. FUCLA(IFIX(FNF),9))  GO TO 2710
C     STATUS EFFECTIVE
      FUCLA(IFIX(FNF),1) = 5.
      USERF = 0.
      TRCLA(IFIX(TRNF),2) = 0.
      TRCLA(IFIX(TRNF),1) = 0.
      TRCLA(IFIX(TRNF),4) = -1.
      GO TO 2730
C     CHECK RANGF AND CEASE FIRE STATUS
2710  CALL GETIA(1,TRNF)
       IR = TRNF * 3.
      IFUNF = FNF
      ITRN = TRNF
      CALL CLOTR(ITRN,IFUNF,CLU,DMIN,TMIN,DIS)
      IF(DMIN .GT. 35)  GO TO 2720
C     STILL IN RANGF CONTINUE TO FIRE
      USERF = 1.
      GO TO 2730
C     NOT IN RANGE OR CEASE FIRE
2720  USERF = 2.
      TRCLA(IFIX(TRNF),4) = 0.
      FUCLA(IFIX(FNF),1) = 1.
      RSTAT(ITRN) = 4
      FUCLA(IFIX(FNF),6) = 0.
C     CHECK IF OUT OF MISSLES
2730  FUCLA(IFIX(FNF),8) = FUCLA(IFIX(FNF),8) - 1.
      IF(FUCLA(IFIX(FNF),8) .GE. 1.)  GO TO 2740
         FUCLA(IFIX(FNF),1) = 10.
         USERF = 0.
2740  RETURN


C     USER FUNCTION 28
C     CHECK FOR SECONDARY
2800  CALL GETIA(2,FNF)
      CALL GETIA(1,TRNF)
      RSTAT(IFIX(TRNF)) = 4
      IFUNF = FNF
      IF(FUCLA(IFUNF,3) .GE. 0)  GO TO 2810
C     THERE IS A SEC ASSIGN
C     SET NEW FU STATUS
         FUCLA(IFUNF,2) = -FUCLA(IFUNF,3)
         FUCLA(IFUNF,3) = 0.
         CALL PUTIA(1,FUCLA(IFUNF,2))
         USERF = 1.
      ITRNF = FUCLA(IFUNF,2)
      PAIR(ITRNF) = IFUNF
      CALL CLOTR(ITRNF,IFUNF,DA,DB,DC,DIS)
      IF(DIS .LT. UNFRM(16))  RETURN
C     SECONDARY OUT OF RANGE
      RSTAT(ITRNF) = 2.
      USERF = 0.
      RETURN
         RETURN
C     NO SEC STORED
2810     FUCLA(IFUNF,2) = FUCLA(IFUNF,3)
      FUCLA(IFUNF,3) = 0.
      PAIR(IFIX(TRNF)) = 11
         USERF = 0.
         RETURN
```

```
USERF   478
USERF   479
USERF   480
USERF   481
USERF   482
USERF   483
USERF   484
USERF   485
USERF   486
USERF   487
USERF   488
USERF   489
USERF   490
USERF   491
USERF   492
USERF   493
USERF   494
USERF   495
USERF   496
USERF   497
USERF   498
USERF   499
USERF   500
USERF   501
USERF   502
USERF   503
USERF   504
USERF   505
USERF   506
USERF   507
USERF   508
USERF   509
USERF   510
USERF   511
USERF   512
USERF   513
USERF   514
USERF   515
USERF   516
USERF   517
USERF   518
USERF   519
USERF   520
USERF   521
USERF   522
USERF   523
USERF   524
USERF   525
USERF   526
USERF   527
USERF   528
USERF   529
USERF   530
USERF   531
ERR2      8
ERR2      9
ERR2     10
ERR2     11
ERR2     12
ERR2     13
ERR2     14
ERR2     15
USERF   532
USERF   533
USERF   534
USERF   535
USERF   536
ERR2     16
USERF   537
USERF   538
USERF   539
USERF   540
USERF   541
```

Figure 2(8). Program Listing: USERF(JJ)

```
C     USER FUNCTION 29                                                  USERF     542
                                                                        USERF     543
C     SET HOLD FIRE MESSAGE                                             USERF     544
2900  CALL GETIA(2,FNF)                                                 USERF     545
      IF(FNF .EQ. 0.)  RETURN                                           USERF     546
      FUCLA(IFIX(FNF),7) = 1.                                           USERF     547
      RETURN                                                            USERF     548
                                                                        USERF     549
                                                                        USERF     550
                                                                        USERF     551
C     USER FUNCTION 30                                                  USERF     552
                                                                        USERF     553
C     CLEAR HF MESSAGE                                                  USERF     554
3000  CALL GETIA(2,FNF)                                                 USERF     555
      IF(FUCLA(IFIX(FNF),7) .EQ. -1.)  GO TO 3010                       USERF     556
                                                                        USERF     557
C     NOTHING HAS BEEN HELD  CLEAR                                      USERF     558
      FUCLA(IFIX(FNF),7) = 0.                                           USERF     559
      USERF = 0                                                         USERF     560
      RETURN                                                            USERF     561
                                                                        USERF     562
C     RESTART HELD TRACK                                                USERF     563
3010  CALL PUTIA(1,FUCLA(IFIX(FNF),2))                                  USERF     564
      CALL GETIA(1,TRNF)                                                USERF     565
      RSTAT(IFIX(TRNF)) = 1                                             USERF     566
      USERF = 1.                                                        USERF     567
      RETURN                                                            USERF     568
                                                                        USERF     569
                                                                        USERF     570
                                                                        USERF     571
C     USER FUCNTION 31                                                  USERF     572
                                                                        USERF     573
C     PROCESS CEASE FIRE/ENGAGEMENT                                     USERF     574
3100  CALL GETIA(1,TRNF)                                                USERF     575
      CALL GETIA(2,FNF)                                                 USERF     576
      ITRNF = TRNF                                                      USERF     577
      IFUNF = FNF                                                       USERF     578
      PAIR(ITRNF) = 11                                                  ERR2       17
      TRCLA(ITRNF,4) = 0.                                               USERF     579
                                                                        USERF     580
C     PRIMARY OR SECONDARY TARGET                                       USERF     581
      IF(FUCLA(IFUNF,2) .EQ. TRNF)  GO TO 3110                          USERF     582
      IF(FUCLA(IFUNF,3) .EQ. TRNF)  GO TO 3130                          USERF     583
                                                                        USERF     584
C     NEITHER DISREGARD                                                 USERF     585
      USERF = 1.                                                        USERF     586
      RETURN                                                            USERF     587
                                                                        USERF     588
C     PRIMARY TARGET                                                    USERF     589
3110  IF(FUCLA(IFUNF,1) .GT. 3)  GO TO 3120                             USERF     590
C     NOT YET FIRED  CHANNGE SED TO PRI                                 USERF     591
      USERF = 0.                                                        USERF     592
      RETURN                                                            USERF     593
                                                                        USERF     594
C     FIRE  CEASE FIRE                                                  USERF     595
3120  FUCLA(IFUNF,6) = 1.                                               USERF     596
      USERF = 1.                                                        USERF     597
      RETURN                                                            USERF     598
                                                                        USERF     599
C     SECONDARY TARGET  CLEAR                                           USERF     600
3130  FUCLA(IFUNF,3) = 0.                                               USERF     601
      RSTAT(ITRNF) = 4.                                                 USERF     602
      USERF = 1.                                                        USERF     603
      RETURN                                                            USERF     604
                                                                        USERF     605
                                                                        USERF     606
                                                                        USERF     607
C     USER FUNCTION 32                                                  USERF     608
                                                                        USERF     609
C     BRANCH ON AUTO ENGAGE                                             USERF     610
3200  USERF = 1.                                                        USERF     611
      LTRN = 0                                                          USERF     612
      RETURN                                                            USERF     613
                                                                        USERF     614
```

Figure 2(9). Program Listing: USERF(JJ)

```
                                                                      USERF     615
                                                                      USERF     616
C     USER FUNCTION 33                                                USERF     617
                                                                      USERF     618
C     CHECK ALL TRACKS                                                USERF     619
3300     USERF = 1.                                                   USERF     620
         LTRN = LTRN + 1                                              USERF     621
         IF(LTRN .GT. NTRK)  GO TO 3309                               USERF     622
            IF(RSTAT(LTRN) .GE. 3 .AND..NOT.AUTOE)  GO TO 3300        USERF     623
            GO TO(3301,3302,3303,3306), RSTAT(LTRN)                   USERF     624
                                                                      USERF     625
C        ENGAGED  CHECK IF FRIENDLY                                   USERF     626
3301        IF(TRCLA(LTRN,1) .NE. 3)  GO TO 3300                      USERF     627
                                                                      USERF     628
C        SEND CEASE FIRE                                              USERF     629
            CALL PUTIA(1,FLOAT(LTRN))                                 USERF     630
            CALL PUTIA(2,TRCLA(LTRN,4))                               USERF     631
            CALL PUTIA(3,4.)                                          USERF     632
            GO TO 3312                                                USERF     633
                                                                      USERF     634
C        ENG/OUT OF RANGE  CHECK IF IN RANGE                          USERF     635
3302        IR = LTRN * 3                                             USERF     636
         CALL CLOTR(LTRN,IFIX(TRCLA(LTRN,4)),CLM,DMIN,TMIN,DIS)       USERF     637
         IF(DMIN .GT. 30.)  GO TO 3310                                USERF     638
            IF(SS(IR) .GT. UNFRM(16)  .OR.                            USERF     639
     *           IFIX(FUCLA(IFIX(TRCLA(LTRN,4)),2)) .NE. LTRN)        USERF     640
     *              GO TO 3300                                       USERF     641
C        SEND IN RANGE                                                USERF     642
               CALL PUTIA(1,FLOAT(LTRN))                              USERF     643
               CALL PUTIA(2,TRCLA(LTRN,4))                            USERF     644
               CALL PUTIA(3,6.)                                       USERF     645
               GO TO 3312                                             USERF     646
                                                                      USERF     647
C        ENG/HF  CHECK IF UNK OR HOST                                 USERF     648
3303        IF(TRCLA(LTRN,1) .NE. 2)  GO TO 3305                      USERF     649
                                                                      USERF     650
C        UNKNOWN TARGET                                               USERF     651
               IF(TIGH)  GO TO 3300                                   USERF     652
                                                                      USERF     653
C        FREE STATUS  CANCEL HF                                       USERF     654
3304           CALL PUTIA(1,FLOAT(LTRN))                              USERF     655
               CALL PUTIA(2,TRCLA(LTRN,4))                            USERF     656
               CALL PUTIA(3,5.)                                       USERF     657
               GO TO 3312                                             USERF     658
                                                                      USERF     659
C        POSSIBLE HOSTILE TARGET                                      USERF     660
3305           IR = LTRN * 3                                          USERF     661
               IF((TRCLA(LTRN,1) .NE. 4)  .OR.                        USERF     662
     *            (SS(IR) .GT. UNFRM(15)))  GO TO 3300                USERF     663
               GO TO 3304                                             USERF     664
                                                                      USERF     665
C        NOT ENGAGED  CHECK IF NOT HOSTIL                             USERF     666
3306        IF(TRCLA(LTRN,1) .NE. 4)  GO TO 3308                      USERF     667
               IR = LTRN * 3                                          USERF     668
               IF(SS(IR) .GT. UNFRM(15))  GO TO 3300                  USERF     669
                                                                      USERF     670
C        WITHIN RANGE                                                 USERF     671
3307           CALL PUTIA(1,FLOAT(LTRN))                              USERF     672
               A = ASSIG(LTRN)                                        USERF     673
               IF(A .EQ. 0.)  GO TO 3300                              USERF     674
            CALL PUTIA(2,A)                                           USERF     675
               CALL PUTIA(3,2.)                                       USERF     676
            CALL PUTSA(10,1.)                                         USERF     677
               GO TO 3312                                             USERF     678
                                                                      USERF     679
C        CHECK IF NOT UNK OR NOT IN RANGE                             USERF     680
3308           IR = LTRN * 3                                          USERF     681
               IF((SS(IR) .GT. UNFRM(16))  .OR.                       USERF     682
     *            (TRCLA(LTRN,1) .NE. 2.))  GO TO 3300                USERF     683
               USERF = 0.                                             USERF     684
               GO TO 3307                                             USERF     685
```

Figure 2(10). Program Listing: USERF(JJ)

```
3310  TRCLA(LTRN,4) = 0.                                                    USERF    686
      RSTAT(LTRN) = 4                                                       USERF    687
      CALL PUTIA(1,FLOAT(LTRN))                                             USERF    688
      CALL PUTIA(2,TRCLA(LTRN,4))                                           USERF    689
      CALL PUTIA(3,1.)                                                      USERF    690
      GO TO 3312                                                            USERF    691
3309  USERF = 2.                                                            USERF    692
3312     RETURN                                                             USERF    693
                                                                            USERF    694
                                                                            USERF    695
                                                                            USERF    696
C     USER FUNCTION 34                                                      USERF    697
                                                                            USERF    698
C     ADVANCE TRACK COUNT RETURN FOR ALL TRACKS                             USERF    699
3400  IPC = IPC + 1                                                         USERF    700
      USERF = 0.                                                            USERF    701
      IF(IPC .GE. NTRK)  USERF = 1.                                         USERF    702
                                                                            USERF    703
C     STORE TRACK NO                                                        USERF    704
      CALL PUTIA(1,FLOAT(IPC))                                              USERF    705
      RETURN                                                                USERF    706
                                                                            USERF    707
                                                                            USERF    708
                                                                            USERF    709
C     USER FUNCTION 35                                                      USERF    710
                                                                            USERF    711
C     INITIALIZE STATE VARIABLES                                            USERF    712
3500  CALL GETIA(1,TRNK)                                                    USERF    713
      ITRNK = TRNK                                                          USERF    714
      K = (ITRNK * 3) - 2                                                   USERF    715
                                                                            USERF    716
      SS(K) = INROU(ITRNK,1)                                                USERF    717
      SS(K + 1) = INROU(ITRNK,2)                                            USERF    718
                                                                            USERF    719
C     UPDATE POINTER                                                        USERF    720
      PTR(ITRNK) = TRROU(PTR(ITRNK),4)                                      USERF    721
      USERF = 0.                                                            USERF    722
                                                                            USERF    723
C     START UNKNOWN STATUS                                                  USERF    724
      TRCLA(ITRNK,2) = 2.                                                   USERF    725
      TRCLA(ITRNK,3) = -1.                                                  USERF    726
      TRCLA(ITRNK,1) = 1.                                                   USERF    727
      RETURN                                                                USERF    728
                                                                            USERF    729
                                                                            USERF    730
                                                                            USERF    731
C     USER FUNCTION 36                                                      USERF    732
                                                                            USERF    733
C     ADVANCE TRACK ROUTE AT TIME                                           USERF    734
3600  CALL GETIA(1,TRNK)                                                    USERF    735
      PTR(IFIX(TRNK)) = TRROU(PTR(IFIX(TRNK)),4)                            USERF    736
      RETURN                                                                USERF    737
                                                                            USERF    738
                                                                            USERF    739
                                                                            USERF    740
C     USER FUNCTION 37                                                      USERF    741
                                                                            USERF    742
C     UPDATE STATUS IF NOT IN AUTO MODES                                    USERF    743
3700  CALL GETIA(1,TRNK)                                                    USERF    744
      ITRNK = TRNK                                                          USERF    745
      IF(TRCLA(ITRNK,4) .EQ. -1.)  RETURN                                   USERF    746
      TYP = TRTYP(PTT(ITRNK),2)                                             USERF    747
      TRCLA(ITRNK,2) = TYP                                                  USERF    748
      IF((TYP .EQ. 2.  .AND.  AUTOI)  .OR.                                  USERF    749
     *      (TYP .EQ. 3.  .AND.  AUTOR)  .OR.                               USERF    750
     *      (TYP .EQ. 4.  .AND. AUTOR))  TRCLA(ITRNK,1) = TYP               USERF    751
                                                                            USERF    752
      IF(TYP .NE. 0.)  GO TO 3710                                           USERF    753
         TRCLA(ITRNK,1) = 0.                                                USERF    754
         TRCLA(ITRNK,2) = 0.                                                USERF    755
         TRCLA(ITRNK,3) = 0.                                                USERF    756
         TRCLA(ITRNK,4) = 0.                                                USERF    757
         TRMOD(ITRNK) = 0.                                                  USERF    758
```

Figure 2(11). Program Listing: USERF(JJ)

```
3710  CONTINUE                                                          USERF     759
      IF(TYP .EQ. TRCLA(ITRNK,1))  CALL PUTSA(10,1.)                    USERF     760
      PTT(ITRNK) = TRTYP(PTT(ITRNK),3)                                  USERF     761
      RETURN                                                            USERF     762
                                                                        USERF     763
                                                                        USERF     764
                                                                        USERF     765
C     USER FUNCTION 38                                                  USERF     766
                                                                        USERF     767
C     GET NEXT TIME FROM STORAGE                                        USERF     768
3800  CALL GETIA(1,TRNI)                                                USERF     769
      USERF = TRROU(PTR(IFIX(TRNI)),1) - TNOW                           USERF     770
      RETURN                                                            USERF     771
                                                                        USERF     772
                                                                        USERF     773
                                                                        USERF     774
C     USER FUNCTION 39                                                  USERF     775
                                                                        USERF     776
C     GET NEXT TIME FROM STORAGE                                        USERF     777
3900  CALL GETIA(1,TRNI)                                                USERF     778
      USERF = TRTYP(PTT(IFIX(TRNI)),1) - TNOW                           USERF     779
      RETURN                                                            USERF     780
                                                                        USERF     781
                                                                        USERF     782
                                                                        USERF     783
C     USER FUNCTION 40                                                  USERF     784
                                                                        USERF     785
C     IDLE TIME                                                         USERF     786
4000  USERF = 10 * UNFRM(1)                                             USERF     787
      RETURN                                                            USERF     788
                                                                        USERF     789
                                                                        USERF     790
                                                                        USERF     791
C     USER FUNCTION 41                                                  USERF     792
                                                                        USERF     793
C     S1 = TRK TYPE  RETURN IF NOT RAW DATA                             USERF     794
4100  USERF = TRCLA(IFIX(TRN),1)                                        USERF     795
      RETURN                                                            USERF     796
                                                                        USERF     797
                                                                        USERF     798
                                                                        USERF     799
C     USER FUNCTION 42                                                  USERF     800
                                                                        USERF     801
C     ASSIGN FU TO TRACK                                                USERF     802
4200  USERF = ASSIG(IFIX(TRN))                                          USERF     803
      RETURN                                                            USERF     804
                                                                        USERF     805
                                                                        USERF     806
                                                                        USERF     807
C     USER FUNCTION 43                                                  USERF     808
                                                                        USERF     809
4300  CALL GETSA(1,TR)                                                  USERF     810
      IF((TR .EQ. 5.).OR.(TR .EQ. 0.)) GO TO 4310                       ERR2       18
      CALL GETIA(1,TRN)                                                 USERF     812
      USERF = TRCLA(IFIX(TRN),1)                                        USERF     813
      RETURN                                                            USERF     814
4310  USERF = TR                                                        ERR2       19
      RETURN                                                            USERF     816
                                                                        USERF     817
                                                                        USERF     818
                                                                        USERF     819
C     USER FUNCTION 44                                                  USERF     820
                                                                        USERF     821
4400  RETURN                                                            USERF     822
                                                                        USERF     823
                                                                        USERF     824
                                                                        USERF     825
```

Figure 2(12). Program Listing: USERF(JJ)

```
C     USER FUCNTION 45                                                          USERF    826
                                                                                USERF    827
C     CHECK IF POSSIBLE HOOK CLEARING                                           USERF    828
4500  USERF = 6.                                                                USERF    829
      CALL PUTSA(10,0.)                                                         USERF    830
      CALL GETSA(4,CFU)                                                         USERF    831
      ICFU = CFU                                                                USERF    832
         IF(ICFU .EQ. 4  .OR.                                                   USERF    833
     *   ICFU .EQ. 6  .OR.                                                      USERF    834
     *   ICFU .EQ. 8)  GO TO 4510                                               USERF    835
         USERF = 5.                                                             USERF    836
         RETURN                                                                 USERF    837
                                                                                USERF    838
C     CHECK FOR SAME TRACK                                                      USERF    839
4510  CALL GETIA(1,CTR)                                                         USERF    840
      ICTR = CTR                                                                USERF    841
      IF(CTR .NE. TRN)  RETURN                                                  USERF    842
         IF(TRCLA(ICTR,1) .EQ. 2.  .OR.                                         USERF    843
     *      TRCLA(ICTR,1) .EQ. 3.  .OR.                                         USERF    844
     *      TRCLA(ICTR,1) .EQ. 4.)  USERF = TRCLA(ICTR,1)                       USERF    845
      RETURN                                                                    USERF    846
                                                                                USERF    847
                                                                                USERF    848
                                                                                USERF    849
C     USER FUNCTION 46                                                          USERF    850
                                                                                USERF    851
C     CHECK FOR SAME TRACK                                                      USERF    852
4600  USERF = 5.                                                                USERF    853
      GO TO 4510                                                                USERF    854
      END                                                                       USERF    855
```

Figure 2(13). Program Listing: USERF(JJ)

Moderator Function 1

Moderator function 1 is called by task 1. It calculates the task performance time and determines which symbol the operator will process next. The function first checks to see if there are any continuation tracks, that is, those that the operator was processing and caused the ID to change and so will continue processing under the new ID. If there are any tracks of this type, the task performance time is set to 0 and the subroutine SETTR is called to store the choice. If there are no continuation jobs, subroutine SETV is called. This assigns a value to each symbol. These values are then summed and stored in the variable STOP. An additional value is added to reflect not selecting any symbol. The partial sums of the values are compared to a random number in order to choose the specific symbol to process. If no symbol is found, the moderator function chooses the idle time task. Task performance time is then computed and is based on the total value which reflects the number and importance of the symbols appearing on the scope.

Moderator Function 2

Moderator function 2 is called by task 50. This function records the firing of missiles.

Moderator Function 3

Moderator function 3 is called by task 47. This function attaches the track to a fire unit. It updates the fire unit and track status arrays. To do this, it first determines whether the track is a primary or secondary assignment for the fire unit and updates the corresponding cells in the array.

Moderator Function 4

Moderator function 4 is called by task 49. It updates the status of the fire unit engaged. In addition, it resets the value of RSTAT to 1 indicating that the track is engaged. This function would not be necessary except for the reengagement or restarted engagements that are processed by the firing unit. For example, tracks that are being held because of the distance factor will be sent to this task to be reengaged when they are within range. This function then updates their status so that the system immediately knows that they are being processed.

Moderator Function 5

Moderator function 5 is called by task 66. This function is used to partially initialize the tracks. It is done at time 0 and sets the task performance time so that task 66 is finished when the track is scheduled to appear on the scope. It adjusts the SS variables so that they are well out of range of the scope and saves the initial location where the track will appear. on the scope. It sets the velocities to 0 so that the target will remain stationary until it appears.

Moderator Function 6

Moderator function 6 is called by all tasks. It is used to keep a running account of the operator's procedures. The function first checks to see if the task is indeed an operator task. If it is not, the function returns. If it is an operator task, it is classified into eight possible categories. These categories are saved as well as the task number that caused the action;

if a track is involved, the track number is recorded; or if a fire unit is involved, the fire unit number is recorded. The function then returns control to the calling task.

Moderator Function 7

Moderator function 7 may be called by task 73. It is used when an operator trace output is desired at regular intervals. The function first checks all tracks and records their status. It then does the same for all fire units. The time is changed from seconds to minutes and seconds. It then decides if the operator is currently looking at a track or a fire unit, since there is a different output format for each. Once these are printed, the function returns control to the calling task.

Moderator Function 8

Moderator function 8 may be called by any task to cause the operator trace to be printed out at the beginning of each task. This function branches to moderator function 7, so the output is identical. At present, moderator function 8 is called by tasks 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 15, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, and 35.

Moderator Function 9

Moderator function 9 is called by tasks 8 and 13. It sets the value of TRCH to true, which indicates to moderator function 10 that there is a possible change in the status of the track.

Moderator Function 10

Moderator function 10 is called by tasks 1, 3, 9, 15, 21, 25, 28, 35, 45, 61, and 68. It is used to collect the user statistics. First, a branching is made according to the calling task. Then, the time since the last event is figured and recorded by calls to subroutines UCLCT and UHIST. The new time is set and the new calssification is stored. If there is no possible change in track status, the function returns to the calling task. If there was a possible change, the function checks all tracks for an actual change. If there was a change in status, statistics are collected by a call to subroutine UCLCT recording how much time has elapsed from the time the target originally appeared as video data. A call is made to subroutine UTMST to record operator efficiency. In addition, this moderator function is used by tasks 61 and 68 for initializing the track statistics.

Moderator Function 11

Moderator function 11 is called by tasks 46 and 49. It is used to update the status of the fire unit to record their overall usage. This is done by a call to UTMST.

Moderator Function 12

Moderator function 12 is called by tasks 51 and 53. It is used to record the amount of time a fire unit is used. This is done by a call to subroutine UTMST.

Moderator Function 13

Moderator function 13 is called by task 74. It is used to record the number of effective firings that a fire unit has on all targets assigned to it during the run. This is accomplished by a call to subroutine UHIST.

Moderator Function 14

Moderator function 14 is called by task 74. It is used to record the time to the effective shooting of the track. It uses that portion of moderator function 10 that dealt with track changes.

```
      SUBROUTINE MODRF(MFN,NNODE)                                        MODRF     1
      REAL TRCLA(33,5),FUCLA(11,9)                                       UCOM1     1
      COMMON /UCOM1/ TRCLA,FUCLA                                         UCOM1     2
                                                                         UCOM1     3
      REAL TRSTA(44,3),TRROU(155,4),INROU(33,2),TRTYP(33,3)              UCOM2     1
      COMMON /UCOM2/ TRSTA,TRROU,INROU,TRTYP                             UCOM2     2
                                                                         UCOM2     3
      INTEGER PAIR(33),PTR(33),PTT(33),RSTAT(33)                         UCOM3     1
      COMMON /UCOM3/ PTR,PTT,RSTAT,PAIR                                  UCOM3     2
                                                                         UCOM3     3
      LOGICAL AUTOI,AUTOR,AUTOE,TIGH                                     UCOM4     1
      COMMON /UCOM4/ AUTOI,AUTOR,AUTOE,TIGH                              UCOM4     2
                                                                         UCOM4     3
      REAL VALUE(20),STI(20),STOT                                        UCOM5     1
      COMMON /UCOM5/ VALUE,STI,STOT                                      UCOM5     2
                                                                         UCOM5     3
      INTEGER TYHOOK,SEQT,PSEQ                                           UCOM6     1
      COMMON /UCOM6/ TYHOOK,SEQT,PSEQ                                    UCOM6     2
                                                                         UCOM6     3
      INTEGER NFU,NTRFU,NTRK                                             UCOM7     1
      COMMON /UCOM7/ NFU,NTRFU,NTRK                                      UCOM7     2
                                                                         UCOM7     3
                                                                         UCOM7     4
                                                                         UCOM7     5
      REAL CX(33),CY(33)                                                 UCOM8     1
      INTEGER IPTR(33),IPTT(33)                                          UCOM8     2
      COMMON /UCOM8/ CX,CY,IPTR,IPTT,IPC                                 UCOM8     3
                                                                         UCOM8     4
      LOGICAL TRCH                                                       UCOM9     1
      REAL TRMOD(33),TOTRT(33),TMARK,TMARE                               UCOM9     2
      INTEGER NOLDTY,LPAGE                                               UCOM9     3
      COMMON /UCOM9/ TRCH,TRMOD,TOTRT,TMARK,TMARE,LPAGE,NOLDTY           UCOM9     4
                                                                         MODRF     3
      COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NPUNCH,      COM06     1
     *               NRNIT,NRENT,MNDC,NDC,NDTN,NNTC                      COM06     2
      COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)      COM17     1
      COMMON /COM22/ TTIME,PFIRB                                         COM22     1
                                                                         MODRF     7
      INTEGER LJFUL(11),LJTRL(6),LJTRK(33),LJFU(10)                      MODRF     8
      DATA LJFUL/1H ,1HU,1HA,1HX,1HF,1HE,1HI,1HZ,1HD,                    MODRF     9
     *          1HC,1H*/                                                MODRF    10
      DATA LJTRL/1H ,1HR,1HU,1HF,1HH,1HS/                                MODRF    11
      DATA LJTRK/33*1H /                                                 MODRF    12
      DATA LJFU/10*1H /                                                  MODRF    13
                                                                         MODRF    14
      INTEGER MROUT(100),IRTY(5)                                         MODRF    15
      DATA MROUT(1),MROUT(2),MROUT(3),MROUT(9),MROUT(21),MROUT(25),      MODRF    16
     *     MROUT(28),MROUT(15),MROUT(35),MROUT(45),MROUT(61),            MODRF    17
     *     MROUT(68)/1,2,3,4,5,6,7,8,9,10,11,12/                         MODRF    18
      DATA IRTY/13,99,10,11,12/                                          MODRF    19
                                                                         MODRF    20
                                                                         MODRF    21
      GO TO (100,200,300,400,500,600,700,800,900,                        MODRF    22
     *       1000,1100,1200,1300,1400), MFN                              MODRF    23
                                                                         MODRF    24
                                                                         MODRF    25
                                                                         MODRF    26
C     MODERATOR FUNCTION 1                                               MODRF    27
C     FIND CONTINUE TRACKS                                               MODRF    28
 100  DO 110 I = 1,NTRFU                                                 MODRF    29
         TRN = TRSTA(I,1)                                                MODRF    30
 110     IF(TRSTA(I,3) .LT. 0.)  140                                     MODRF    31
                                                                         MODRF    32
C     NO CONTINUE JOBS  FIND TOTALS                                      MODRF    33
      CALL SETV                                                          MODRF    34
                                                                         MODRF    35
C     SUM TOTALS                                                         MODRF    36
      STOT = 0.                                                          MODRF    37
      DO 120 I = 1,NTRFU                                                 MODRF    38
 120     STOT = TRSTA(I,3) + STOT                                        MODRF    39
```

Figure 3(1). Program Listing: MODRF(MFN,NNODE)

```
C     ADD IDLE TIME                                                    MODRF   40
      STOT = STOT + 10.                                                MODRF   41
                                                                       MODRF   42
C     FIND NEXT TRN                                                    MODRF   43
      DIS = UNFRM(1) * STOT                                            MODRF   44
      VAL = 0.                                                         MODRF   45
      DO 130 I = 1,NTRFU                                               MODRF   46
         VAL = VAL + TRSTA(I,3)                                        MODRF   47
         IF(VAL .LT. DIS)  GO TO 130                                   MODRF   48
                                                                       MODRF   49
C     STORE TRACK AND ROUTE                                            MODRF   50
         CALL SETTR(TRSTA(I,1))                                        MODRF   51
         TTIME = UNFRM(2) * (100. / STOT)                              MODRF   52
         TRSTA(I,2) = TNOW                                             MODRF   53
         RETURN                                                        MODRF   54
                                                                       MODRF   55
 130     CONTINUE                                                      MODRF   56
                                                                       MODRF   57
C     DEFALTS TO IDLE TIME                                             MODRF   58
      CALL PUTSA(1,0.)                                                 MODRF   59
      TTIME = UNFRM(2) * (100. / STOT)                                 MODRF   60
      RETURN                                                           MODRF   61
                                                                       MODRF   62
C     CONTINUATION JOB                                                 MODRF   63
 140  J = IFIX(TRN) + NFU                                              MODRF   64
      TRSTA(J,3) = 0.                                                  MODRF   65
      TTIME = 0.                                                       MODRF   66
      CALL SETTR(TRN)                                                  MODRF   67
      RETURN                                                           MODRF   68
                                                                       MODRF   69
                                                                       MODRF   70
                                                                       MODRF   71
C     MODERATOR FUNCTION 2                                             MODRF   72
                                                                       MODRF   73
 200  CALL GETIA(2,FN)                                                 MODRF   74
      CALL UHIST(FN,2)                                                 MODRF   75
      RETURN                                                           MODRF   76
      RETURN                                                           MODRF   77
                                                                       MODRF   78
                                                                       MODRF   79
                                                                       MODRF   80
C     MODERATOR FUNCTION 3                                             MODRF   81
                                                                       MODRF   82
C     DETERMIN IF PRIMARY OR SECONDARY TRACK                           MODRF   83
 300  CALL GETIA(1,TRN)                                                MODRF   84
      CALL GETIA(2,FN)                                                 MODRF   85
      ITRN = TRN                                                       MODRF   86
      IFUN = FN                                                        MODRF   87
      IF(IFUN .EQ. 0)  GO TO 320                                       MODRF   88
      PAIR(ITRN) = IFUN                                                ERR2     1
      RSTAT(ITRN) = 1                                                  ERR2     2
                                                                       MODRF   89
      IF(FUCLA(IFUN,2) .NE. 0.)  GO TO 310                             MODRF   90
                                                                       MODRF   91
C     THIS IS PRIMARY  SET STATUS RECORD TR AND FU                     MODRF   92
      FUCLA(IFUN,1) = 2.                                               MODRF   93
      FUCLA(IFUN,2) = TRN                                              MODRF   94
      TRCLA(ITRN,4) = FN                                               MODRF   95
      RETURN                                                           MODRF   96
                                                                       MODRF   97
C     SECONDARY TRACK  RECORD TR AND FU                                MODRF   98
 310  FUCLA(IFUN,3) = TRN                                              MODRF   99
      TRCLA(ITRN,4) = FN                                               MODRF  100
      RETURN                                                           MODRF  101
C     NO FU WAS AVAILABLE                                              MODRF  102
 320   TTIME = 0.                                                      ERR2     3
      RETURN                                                           ERR2     4
                                                                       ERR2     5
                                                                       MODRF  103
                                                                       MODRF  104
                                                                       MODRF  105
```

Figure 3(2). Program Listing: MODRF(NFN,NNODE)

```
C     MODERATOR FUNCTION 4                                              MODRF   106
                                                                        MODRF   107
C     SET FU STATUS                                                     MODRF   108
 400  CALL GETIA(2,FN)                                                  MODRF   109
      FUCLA(IFIX(FN),1) = 3.                                            MODRF   110
      CALL GETIA(1,TRN)                                                 MODRF   111
      RSTAT(IFIX(TRN)) = 1                                              MODRF   112
      RETURN                                                            MODRF   113
                                                                        MODRF   114
                                                                        MODRF   115
                                                                        MODRF   116
C     MODERATOR FUNCTION 5                                              MODRF   117
                                                                        MODRF   118
C     SET TIME FOR TRACK TO APPPEAR                                     MODRF   119
 500  CALL GETIA(1,TRN)                                                 MODRF   120
      ITRN = TRN                                                        MODRF   121
      TTIME = TRROU(PTR(IFIX(TRN)),1)                                   MODRF   122
                                                                        MODRF   123
C     SET STATE VARIABLES                                               MODRF   124
      K = (ITRN * 3) - 2                                                MODRF   125
      SS(K) = 1000.                                                     MODRF   126
      SS(K + 1) = 1000.                                                 MODRF   127
      SS(K + 2) = 1000000.                                              MODRF   128
                                                                        MODRF   129
C     SET TRACK STATUS                                                  MODRF   130
      K = NFU + ITRN                                                    MODRF   131
      TRSTA(K,1) = TRN                                                  MODRF   132
      TRSTA(K,2) = TTIME                                                MODRF   133
                                                                        MODRF   134
C     SET TRACK CLASIFICATION                                           MODRF   135
      DO 510 I = 1,5                                                    MODRF   136
 510     TRCLA(ITRN,I) = 0.                                             MODRF   137
                                                                        MODRF   138
C     SET INITIAL COORD                                                 MODRF   139
      INROU(ITRN,1) = TRROU(PTR(ITRN),2)                                MODRF   140
      INROU(ITRN,2) = TRROU(PTR(ITRN),3)                                MODRF   141
      TRROU(PTR(ITRN),2) = 0.                                           MODRF   142
      TRROU(PTR(ITRN),3) = 0.                                           MODRF   143
      RETURN                                                            MODRF   144
                                                                        MODRF   145
                                                                        MODRF   146
                                                                        MODRF   147
C     MODERATOR FUNCTION 6                                              MODRF   148
                                                                        MODRF   149
 600  CONTINUE                                                          MODRF   150
      LJHOK = 1H                                                        MODRF   151
      IF(NNODE .GT. 34)  GO TO 610                                      MODRF   152
      GO TO (601,602,603,603,603,603,603,603,604,604,604,604,604,604,   MODRF   153
     *       605,605,605,605,605,605,606,606,606,606,607,607,607,       MODRF   154
     *       608,608,608,608,608,608,608), NNODE                        MODRF   155
                                                                        MODRF   156
 601  LJTYP = 3HSER                                                     MODRF   157
      LJTRN = 0                                                         MODRF   158
      LJTSK = NNODE                                                     MODRF   159
      RETURN                                                            MODRF   160
                                                                        MODRF   161
 602  LJTYP = 3HIDL                                                     MODRF   162
      LJTRN = 0                                                         MODRF   163
      LJTSK = NNODE                                                     MODRF   164
      RETURN                                                            MODRF   165
                                                                        MODRF   166
 603  LJTYP = 3HOBR                                                     MODRF   167
      CALL GETIA(1,A)                                                   MODRF   168
      LJTSK = NNODE                                                     MODRF   169
      LJTRN = A                                                         MODRF   170
      RETURN                                                            MODRF   171
                                                                        MODRF   172
 604  LJTYP = 3HOBU                                                     MODRF   173
      LJTSK = NNODE                                                     MODRF   174
      CALL GETIA(1,A)                                                   MODRF   175
      LJTRN = A                                                         MODRF   176
      RETURN                                                            MODRF   177
```

Figure 3(3). Program Listing: MODRF(MFN,NNODE)

```
 605  LJTYP = 3HASS                                                  MODRF  178
      CALL GETIA(1,A)                                                MODRF  179
      LJTRN = A                                                      MODRF  180
      LJTSK = NNODE                                                  MODRF  181
      RETURN                                                         MODRF  182
                                                                     MODRF  183
 606  LJTYP = 3HOBF                                                  MODRF  184
      CALL GETIA(1,A)                                                MODRF  185
      LJTRN = A                                                      MODRF  186
      LJTSK = NNODE                                                  MODRF  187
      RETURN                                                         MODRF  188
                                                                     MODRF  189
607   LJTYP = 3HOBH                                                  MODRF  190
      CALL GETIA(1,A)                                                MODRF  191
      LJTRN = A                                                      MODRF  192
      LJTSK = NNODE                                                  MODRF  193
      RETURN                                                         MODRF  194
                                                                     MODRF  195
608   LJTYP = 3HOFU                                                  MODRF  196
      CALL GETIA(2,A)                                                MODRF  197
      LJTRN = -A                                                     MODRF  198
      IF(A .EQ. 0.)  LJTRN = -11                                     MODRF  199
      LJTSK = NNODE                                                  MODRF  200
      RETURN                                                         MODRF  201
                                                                     MODRF  202
 610  IF(NNODE .LT. 46)  LJHOK = 1H*                                 MODRF  203
      RETURN                                                         MODRF  204
                                                                     MODRF  205
                                                                     MODRF  206
                                                                     MODRF  207
C     MODERATOR FUNCTION 7                                           MODRF  208
                                                                     MODRF  209
 700  DO 710 I = 1,NTRK                                              MODRF  210
 710     LJTRK(I) = LJTRL(IFIX(TRCLA(I,1)) + 1)                      MODRF  211
                                                                     MODRF  212
      DO 720 I = 1,NFU                                               MODRF  213
 720     LJFU(I) = LJFUL(IFIX(FUCLA(I,1)) + 1)                       MODRF  214
                                                                     MODRF  215
      ALJTA = AMOD(TNOW,60.)                                         MODRF  216
      LJTB = IFIX(TNOW) / 60                                         MODRF  217
      IF(LPAGE .GT. 55)  LPAGE = 0                                   MODRF  218
      IF(LPAGE .EQ. 0)  WRITE (6,5002)                               MODRF  219
      LPAGE = LPAGE + 1                                              MODRF  220
      IF(LJTRN .LT. 0)  GO TO 730                                    MODRF  221
         LJA = LJTRN                                                 MODRF  222
         LJB = LJTRL(IFIX(TRCLA(LJA,1)) + 1)                         MODRF  223
         LJC = SS(LJA * 3)                                           MODRF  224
         LJD = TRCLA(LJA,4)                                          MODRF  225
      WRITE(6,5000) LJTB,ALJTA,LJTYP,LJHOK,LJTSK,                    MODRF  226
     *           LJA,LJB,LJC,LJD,LJTRK,LJFU                          MODRF  227
      RETURN                                                         MODRF  228
                                                                     MODRF  229
 730  CONTINUE                                                       MODRF  230
         LJA = -LJTRN                                                MODRF  231
      IF(LJTRN .EQ. -11)  LJA = 0.                                   MODRF  232
         LJB = LJFUL(IFIX(FUCLA(LJA,1)) + 1)                         MODRF  233
         LJC = FUCLA(LJA,2)                                          MODRF  234
         LJD = FUCLA(LJA,3)                                          MODRF  235
      WRITE(6,5001) LJTB,ALJTA,LJTYP,LJHOK,LJTSK,                    MODRF  236
     *           LJA,LJB,LJC,LJD,LJTRK,LJFU                          MODRF  237
      RETURN                                                         MODRF  238
5000  FORMAT(1H ,I4,F6.2,3X,A3,A1,I4,5H  TR-,I2,2X,A1,4H  D-,        MODRF  239
     *       I3,6H  AFU-,I2,7X,33A1,5X,10A1)                         MODRF  240
5001  FORMAT(1H ,I4,F6.2,3X,A3,A1,I4,5H  FU-,I2,2X,A1,4H  P-,        MODRF  241
     *       I2,5H   S-,I2,9X,33A1,5X,10A1)                         MODRF  242
                                                                     MODRF  243
                                                                     MODRF  244
                                                                     MODRF  245
C     MODERATOR FUNCTION 8                                           MODRF  246
                                                                     MODRF  247
C     PRINT OUT TASK STARTS                                          MODRF  248
 800  GO TO 700                                                      MODRF  249
                                                                     MODRF  250
```

Figure 3(4). Program Listing: MODRF(MFN,NNODE)

```
                                                                      MODRF      251
                                                                      MODRF      252
                                                                      MODRF      253
C     MODERATOR FUNCTION 9                                            MODRF      254
                                                                      MODRF      255
C     SET BRANCH FOR POSSIBLE TRACK TYPE CHANGE                       MODRF      256
 900  TRCH = .TRUE.                                                   MODRF      257
      RETURN                                                          MODRF      258
                                                                      MODRF      259
                                                                      MODRF      260
                                                                      MODRF      261
C     MODERATOR FUNCTION 10                                           MODRF      262
                                                                      MODRF      263
C      BRANCH FOR CORRECT NODE                                        MODRF      264
1000  GO TO(1001,1002,1003,1004,1005,1006,1007,1008,1009,             MODRF      265
     *      1010,1030,1040), MROUT(NNODE)                            MODRF      266
                                                                      MODRF      267
1001  NNEWTY = 1                                                      MODRF      268
      GO TO 1020                                                      MODRF      269
1002  NNEWTY = 2                                                      MODRF      270
      GO TO 1020                                                      MODRF      271
1003  NNEWTY = 3                                                      MODRF      272
      GO TO 1020                                                      MODRF      273
1004  NNEWTY = 4                                                      MODRF      274
      GO TO 1020                                                      MODRF      275
1005  NNEWTY = 5                                                      MODRF      276
      GO TO 1020                                                      MODRF      277
1006  NNEWTY = 6                                                      MODRF      278
      GO TO 1020                                                      MODRF      279
1007  NNEWTY = 7                                                      MODRF      280
      GO TO 1020                                                      MODRF      281
1008  NNEWTY = 8                                                      MODRF      282
      GO TO 1020                                                      MODRF      283
1009  TMARH = TNOW                                                    MODRF      284
      RETURN                                                          MODRF      285
                                                                      MODRF      286
1010  TMARH = TNOW - TMARH                                            MODRF      287
      CALL UCLCT(TMARH,9)                                             MODRF      288
      CALL UHIST(9.,1)                                                MODRF      289
      RETURN                                                          MODRF      290
                                                                      MODRF      291
1020  TMARK = TNOW - TMARK                                            MODRF      292
      CALL UCLCT(TMARK,NOLDTY)                                        MODRF      293
      CALL UHIST(FLOAT(NOLDTY),1)                                     MODRF      294
      TMARK = TNOW                                                    MODRF      295
      NOLDTY = NNEWTY                                                 MODRF      296
      IF(TRCH)  GO TO 1030                                            MODRF      297
      RETURN                                                          MODRF      298
                                                                      MODRF      299
1030  TRCH = .FALSE.                                                  MODRF      300
      T = 0.                                                          MODRF      301
      DO 1034 I = 1,NTRK                                              MODRF      302
         K = 3 * I                                                    MODRF      303
         IF(SS(K) .GT. 500)  GO TO 1034                               MODRF      304
         IF(TRMOD(I) .EQ. TRCLA(I,1))  GO TO 1033                     MODRF      305
         TMARE = TNOW - TOTRT(I)                                      MODRF      306
         TRMOD(I) = TRCLA(I,1)                                        MODRF      307
         DO 1031 J = 1,5                                              MODRF      308
            RJ = J - 1                                                MODRF      309
            ITY = IRTY(J)                                             MODRF      310
            IF(TRMOD(I) .EQ. RJ)  GO TO 1032                          MODRF      311
1031        CONTINUE                                                  MODRF      312
1032     CALL UCLCT(TMARE,ITY)                                        MODRF      313
1033     IF(TRCLA(I,1) .NE. TRCLA(I,2))  T = 1.                       MODRF      314
1034  CONTINUE                                                        MODRF      315
      CALL UTMST(T,TNOW,1)                                            MODRF      316
      RETURN                                                          MODRF      317
                                                                      MODRF      318
1040  CALL GETIA(1,TN)                                                MODRF      319
      ITN = TN                                                        MODRF      320
      IF(TRCLA(ITN,3) .NE. -1.)  GO TO 1030                           MODRF      321
         TRCLA(ITN,3) = 0.                                            MODRF      322
         CALL UTMST(1.,TNOW,1)                                        MODRF      323
```

Figure 3(5). Program Listing: MODRF(MFN,NNODE)

```
1041      TRMOD(ITN) = 1.                                                MODRF   324
          TOTRT(ITN) = TNOW                                              MODRF   325
          RETURN                                                         MODRF   326
                                                                         MODRF   327
                                                                         MODRF   328
                                                                         MODRF   329
                                                                         MODRF   330
C     MODERATOR FUNCTION 11                                              MODRF   331
                                                                         MODRF   332
C     START ALL FIRE UNITS                                               MODRF   333
1100  CALL GETIA(2,FN)                                                   MODRF   334
      IFN = FN                                                           MODRF   335
      IF(IFN .EQ. 0)  RETURN                                             MODRF   336
      CALL UTMST(1.,TNOW,(IFN + 1))                                      MODRF   337
      RETURN                                                             MODRF   338
                                                                         MODRF   339
                                                                         MODRF   340
                                                                         MODRF   341
C     MODERATOR FUNCTION 12                                              MODRF   342
                                                                         MODRF   343
C     STOP ALL FIRE UNITS                                                MODRF   344
1200  CALL GETIA(2,FN)                                                   MODRF   345
      IFN = FN                                                           MODRF   346
      IF(IFN .EQ. 0)  RETURN                                             MODRF   347
      CALL UTMST(0.,TNOW,(IFN + 1))                                      MODRF   348
      RETURN                                                             MODRF   349
                                                                         MODRF   350
                                                                         MODRF   351
                                                                         MODRF   352
C     MODERATOR FUNCTION 13                                              MODRF   353
                                                                         MODRF   354
C     RECORD EFFECTIVE FIRE UNITS                                        MODRF   355
1300  CALL GETIA(2,FN)                                                   MODRF   356
      IFN = FN                                                           MODRF   357
      IF(IFN .EQ. 0) RETURN                                              MODRF   358
      CALL UHIST(FN,3)                                                   MODRF   359
      RETURN                                                             MODRF   360
                                                                         MODRF   361
                                                                         MODRF   362
                                                                         MODRF   363
C     MODERATOR FUNCTION 14                                              MODRF   364
                                                                         MODRF   365
C     RECORD AUTO TRACK CHANGES                                          MODRF   366
1400  GO TO 1030                                                         MODRF   367
5002  FORMAT(1H1/19X,4HTASK,40X,10(1H1),10(1H2),4H3333,14X,1H1/          MODRF   368
     *       4X,18HT I M E   JOB   NO,32X,3(10H1234567890),3H123,5X,     MODRF   369
     *       10H1234567890/)                                             MODRF   370
      END                                                                MODRF   371
```

Figure 3(6). Program Listing: MODRF(MFN,NNODE)

Table V

GLOBAL USER VARIABLES

Variable Name	User Common Block	Definition
AUTOE	4	The auto/manual track engagement indicator.
AUTOI	4	The auto/manual track initiate indicator.
AUTOR	4	The auto/manual track interrogate indicator.
CX(I) I=1,33	8	Stores the initial X-coordinate of track I for multiple runs.
CY(I) I=1,33	8	Stores the initial Y-coordinate of track I for multiple runs.
FUCLA(I,J) I=1,11 J=1,9	1	The status array for fire unit I, I=1,10. J=1: fire unit status 1:U = unused 2:A = accessed 3:X = engaged 4:F = firing 5:E = effective 6:I = ineffective 7:Z = not operational 8:D = disengage 9:C = hold fire 10:* = blinking J=2: primary track number J=3: secondary track number J=4: location of X-coordinate J=5: location of Y-coordinate J=6: cease fire flag J=7: hold fire flag J=8: initial weapons count J=9: effectiveness ratio
INROU(I,J) I=1,33 J=1,2	2	The initial location of the track. J=1: X-coordinate J=2: Y-coordinate
IPC	8	Used by task 61 to count tracks initialized.
IPTR(I) I=1,33	8	Stores the initial route pointer for track I for multiple runs.

Table V (continued)

Variable Name	Common Block	Definition
IPTT(I) I=1,33	8	Stores the initial status pointer for track I for multiple runs.
LPAGE	9	Used to count page lines.
NFU	7	The total number of fire units.
NOLDTY	9	Stores operator task classification to collect statistics.
NTRFU	7	The total number of fire units (NFU) plus the total number of tracks (NTRK).
NTRK	7	The total number of tracks.
PAIR(I) I=1,33	3	The fire unit number that is attached with track I. (11 - not attached)
PSEQ	6	The current sequence category for the system.
PTR(I) I=1,33	3	The current pointer to the routing *information* (TRROU) for track I.
PTT(I) I=1,33	3	The current pointer to the identification status update information (TRTYP) for track I.
RSTAT(I) I=1,33	3	The automatic status used by task 63. 1 = engaged 2 = range-hold fire 3 = hold fire message 4 = other
SEQT	6	The sequence hook category. 0 = track and fire unit 1 = track 2 = fire unit 3 = hostile track and fire unit 4 = hostile track
SS(I)	COM17	I = 1 mod 3: location (X-coordinate) I = 2 mod 3: location (Y-coordinate) I = 0 mod 3: range (center/fire unit)
STI(I) I=1,20	5	The visual stimulation provided by each type of symbol (see "Visual Value" page).
STOT	5	The total value figured in moderator function 1.

Table V (continued)

Variable Name	Common Block	Definition
TIGH	4	The tight/free policy indicator.
TMARE	9	Interval marker for collecting statistics on track status.
TMARK	9	Interval marker for collecting statistics on operator tasks.
TOTRT(I) I=1,33	9	Stores the time the track appeared on the scope to collect statistics.
TRCH	9	Indicator for possible track update.
TRCLA(I,J) I=1,33 J=1,5	1	The status array for track I. J=1: observed classification J=2: real classification J=3: last observed classification J=4: assigned fire unit number
TRMOD(I) I=1,33	9	Stores track identification status to collect statistics.
TRROU(I,J) I=1,155 J=1,4	2	The routing information for all tracks. J=1: next turn time J=2: current velocity (X-coordinate) J=3: current velocity (Y-coordinate) J=4: pointer to next line
TRSTA(I,J) I=1,44 J=1,3	2	The current value of each symbol. J=1: track/fire unit number J=2: time last observed J=3: value of symbol
TRTYP(I,J) I=1,33 J=1,3	2	The identification status update information. J=1: next change time J=2: next type J=3: pointer to next line
TYHOOK	6	The hooking policy used during a simulation run. 0 = sequence 1 = position/number 2 = tab
VALUE(I) I=1,20	5	The significant value associated with each type of symbol (see "Visual Value" page).

Table VI

LOCAL USER VARIABLES

Variable Name	Subroutine	Definition
BFU	ASSIG	The fire unit with the smallest distance to the given track.
BV	ASSIG	The smallest distance from the given track to a usable fire unit.
CLV	CLOTR	The closing velocity between the given track and fire unit.
DIS	CLOTR	The current distance between the given track and fire unit.
FN	MODRF	Fire unit number being processed. (Real)
FN	USERF	Operator's fire unit number. (Real)
FNF	USERF	Fire unit's fire unit number. (Real)
GTRN	NHOOK	Track or fire unit number of the desired symbol. (Real)
IFUN	MODRF	Fire unit number being processed. (Integer)
IFUN	USERF	Operator's fire unit number. (Integer)
IFUNF	USERF	Fire unit's fire unit number. (Integer)
ITRN	MODRF	Track number being processed. (Integer)
ITRN	USERF	Fire unit/operator track number.(Integer)
ITRNK	USERF	System's track number. (Integer)
LDIS	SETV	An occurrence flag used to calculate the symbol value. TRUE = long range
LJFU(I) I=1,10	MODRF	Used to print status of all ten possible fire units in trace output.
LJFUL(I) I=1,11	MODRF	Contains the possible fire unit status symbols used in trace output.
LJTRN	MODRF	Stores track number operator is processing for trace output.

Table VI (continued)

Variable Name	Subroutine	Definition
LJTSK	MODRF	Stores the operator task number for trace output.
LJTRIL(I) I=1,33	MODRF	Used to print status of all 33 possible tracks in trace output.
LJTRL(I) I=1,6	MODRF	Contains the possible track status symbols used in trace output.
LJTYP	MODRF	Stores type job operator is starting for trace output.
LOLD	SETV	An occurrence flag used to calculate the symbol value. TRUE = has been observed in this status before
LP	ASSIG	Flag - TRUE if the fire unit already has a primary assignment.
MIND	CLOTR	The minimum distance achieved by the given track and fire unit (same as DMIN as used in ASSIG and USERF). (Real)
SHBPT	NHOOK	Starting symbol for a sequence hook.
SHPT	NHOOK	Symbol pointer for sequence hook.
TMIN	CLOTR	The time until the minimum distance achieved by the given track and fire unit (MIND) occurs.
TRN	MODRF	Track number being processed. (Real)
TRN	USERF	Operator's track number. (Real)
TRNF	USERF	Fire unit's track number. (Real)
TRNK	USERF	System's track number (also TRNI). (Real)
TVAL	SETV	The time factor used in task 1 (SEARCH).

SECTION IV

DATA INPUT PROCEDURES

This section describes the input data required for the SAINT model. The data requirements are divided into two categories: SAINT model input and AN/TSQ-73 mission input.

SAINT Model Input

The SAINT model input provides the SAINT simulation program with a description of the model described in Section II. A detailed description of the SAINT model input requirements is found in The SAINT User's Manual [5]. A complete listing of the SAINT model input appears in Figure 4.

AN/TSQ-73 Mission Input

The AN/TSQ-73 mission input data describes the specifics of the mission under study. This data defines the system operating modes, the characteristics of all fire units, and the flight paths and identification of all tracks. A sample listing of mission input data appears in Figure 5. A summary of mission input requirements appears in Table VII.

The following is an explanation of the mission input data shown in Figure 5. The data was used to generate the output discussed in Section V. The first three lines (cards) are used for general mission information.

Line 1:
- T - Automatic initiate mode (F - manual)
- T - Automatic interrogate mode (F - manual)
- T - Automatic engagement mode (F - manual)
- T - Tight engagement policy (F - free)

Line 2:
- 1 - Position or number hooking by the operator (0,2 - other: see Table VII)
-] - N/A (used only for sequence hooking)

Line 3: 2 - Two fire units are specified for this mission.
10 - Ten tracks (both hostile and friendly) are specified for this mission.

The next two lines (4,5) define the characteristics of the two fire units used in the mission. In general, there will be one line for each fire unit specified on line 3.

Line 4: 10 - The x-coordinate of the location of fire unit one is 10.0 miles from the origin. (The origin should be selected to reflect the "center" of the system.)

10. - The y-coordinate of the location of fire unit one is 10.0 miles from the origin.

4. - The fire unit starts the simulation with 4 missiles.

.99 - 99% of the missiles fired result in an effective engagement.

Line 5: The location (10.,-10), number of missiles (4) and the site's effectiveness (99%) is given for fire unit two.

The next 22 lines define the flight paths for all tracks. Each track requires at least <u>two</u> lines to represent a flight path. A single line must be added for each additional leg of the track (lines 8, 9, and 10 provide an example of a track with two legs). Note that the meaning of the variables in the first flight path line for each track is different from the remaining. If the data contains more than two lines, all those lines after the first have the same meaning.

Line 6: 1 - Flight path for track 1.
50. - Track 1 will appear at time 50 seconds.
80. - The x-coordinate where track 1 will first appear (at time 50).
0 - The y-coordinate where track 1 will first appear.

Line 7: 1 - Flight path for track 1.
5000: - The time of the next turn (5000 is too large to occur, therefore there is no turn).
-1. - The x-velocity for track 1 in miles per second until time 5000.
-03. - The y-velocity for track 1 in miles per second until time 5000.

Line 8: Track 2 will appear at time 50 at location (80.,0.).

Line 9: Until time 500., the velocity vector for track 2 will be x-velocity = -.1 and y-velocity = -.03.

Line 10: Until time 5000. (beginning at the 500. by line 9), the velocity vector for track 2 will be x-velocity = -.075 and y-velocity = -.075.

Line 11, 12: Flight path information for track 3.

Line 13 - 26: Flight path information for tracks 4-10.

Line 27: 99 - Stops processing of flight paths. (Note this must only be a number larger than the total number of tracks specified on line 3.)

The next 23 lines define the track identification information. Each track requires at least two lines and both have the same meaning. This information reflects the highest level of identification for the system. If the appropriate automatic mode of operation is initialized, these updates will be made automatically. However, if a manual mode is chosen, the data will be stored for future use by the operator. The tracks defined in lines 6-27 will first 'appear' as video data. This may be updated at any time manually to a track, but will not be initialized automatically until the time specified in the input data (see line 28).

Line 28: 1 - Identification information for track 1.
75. - The time (75 seconds) for the information update given on this line for track 1.
2. - The status may now be given as an unknown track. (This is used only in the automatic initiate track mode.)

Line 29: 1 - Identification information for track 1.
250. - The time (250 seconds) for the information update.
4. - The status of the track may now (250 seconds) be given as hostile.

Line 30: 1 - Identification information for track 1.
5000. - The time (never reached) for the information update.
- - May be left blank since it is not used.

Line 31 - 33: The identification information for track 2.
At time 100 - unknown track.
At time 200 - hostile track (for the duration).

Line 34 - 49: The identification information for tracks 3 - 10 (note tracks 9 and 10 are identified as friendly).

Line 50: 99 - Stops processing of track identification information (see line 27).

```
GEN,ARI,3,1,1978,1,2,(11)N*
SGE,0,39,1.,1000.*
POP,2,0,3,11,14*
OUT,0,(6)0,0,0,0,0,0,0,0,N,Y,Y,Y*
DIS,1,UN,,0.,1.*
DIS,2,UN,,1.,5.*
DIS,3,UN,,1.,10.*
DIS,4,UN,,.5,1.5*
DIS,5,UN,,2.,06.*
DIS,6,UN,,1.,05.*
DIS,7,UN,,2.,04.*
DIS,8,UN,,2.,05.*
DIS,9,UN,,04.,08.*
DIS,10,UN,,10.,20.*
DIS,11,UN,,04.,09.*
DIS,12,UN,,10.,20.*
DIS,13,UN,,10.,20.*
DIS,14,UN,,20.,40.*
DIS,15,UN,,45.,65.*
DIS,16,UN,,30.,45.*
UBO,1,SEARCHT,
    2,IDLET,
    3,VIDEOT,
    4,UNKT,
    5,FRIENDT,
    6,HOSTILET,
    7,FIREUT,
    8,ASSIGNT,
    9,HOOKINGT,
   10,TIMETRAK,
   11,TIMEFRND,
   12,TIMEHOST,
   13,KILLT*
UTI,1,OBEFF,,
    2,FU1,,
    3,FU2,,
    4,FU3,,
    5,FU4,,
    6,FU5,,
    7,FU6,,
    8,FU7,,
    9,FU8,,
   10,FU9,,
   11,FU10*
UHI,1,OPERATOR,12,0.,1.,
    2,FUOPERAT,12,0.,1.,
    3,FUEFFECT,12,0.,1.*
IMO,6,A*
TAS,1,SEARCH,0,1,SC,0,(10)SD*
MOD,1,1,A,,
      8,A,,
     10,A*
ATA,1,COM,SA,0,1,UF,43*
CFI,1,2,ALV,0.,1,SA,,
       3,ALV,1.,1,SA,,
       9,ALV,2.,1,SA,,
      21,ALV,3.,1,SA,,
      24,ALV,9.,1,SA*
TAS,2,IDLETIME,1,1,UF,40*
MOD,2,8,A,,
      10,A*
STA,2,(5)BET,STA,10,0.,30.*
DET,2,1*
TAS,3,OBSVIDEO,1,1,DS,4,(16)1*
MOD,3,8,A,,
     10,A*
STA,3,(5)BET,STA,10,0.,30.*
ATA,3,COM,SA,0,1,UF,1*
PRO,3,SA,0,1,1,
           4,2,
           5,3*
```

Figure 4(1). SAINT Model Input

```
TAS,4,WAITONE,1,1,DS,5,
      (16)1*
MOD,4,8,A*
DET,4,3*
TAS,5,AUTOMANN,1,1,SC,0,
      (16)1*
ATA,5,COM,SA,0,1,UF,2*
PRO,5,SA,0,1,1,
            6,2,
            7,3*
TAS,6,WATCHVID,1,1,UF,3,
      (16)1*
MOD,6,8,A*
ATA,6,COM,SA,0,1,UF,41*
CFI,6,1,AGV,1.,1,SA,,
      7,ALV,1.,1,SA*
TAS,7,POSTAB,1,1,DS,6,
      (16)1*
MOD,7,8,A*
DET,7,8*
TAS,8,PINDICAT,1,1,DS,7,
      (16)1*
MOD,8,8,A,,
      9,A*
ATA,8,COM,SA,0,1,UF,4*
DET,8,1*
TAS,9,OBSUNK,1,1,DS,4,
      (16)1*
MOD,9,8,A,,
     10,A*
STA,9,(5)BET,STA,10,0.,30.*
UTC,9,,,40.,60.,1.,.2*
ATA,9,COM,SA,0,1,UF,5*
PRO,9,SA,0,1,1,
            10,2*
TAS,10,PIDIFF,1,1,DS,7,
      (16)1*
MOD,10,8,A*
ATA,10,COM,SA,0,4,SC,1,
            SA,0,5,SC,0*
DET,10,35*
TAS,11,PINTERRO,1,1,DS,7,
      (16)1*
MOD,11,8,A*
DET,11,12*
TAS,12,READMSG,1,1,DS,8,
      (16)1*
MOD,12,8,A*
UTC,12,,,50.,50.,1.,.1*
ATA,12,COM,SA,0,1,UF,6*
PRO,12,SA,0,1,1,
            13,2,
            14,3*
TAS,13,PFH,1,1,DS,7,
      (16)1*
MOD,13,8,A,,
       9,A*
DET,13,1*
TAS,14,TIGHGREE,1,1,SC,0,
      (16)1*
ATA,14,COM,SA,0,1,UF,7*
PRO,14,SA,0,1,1,
            15,2*
TAS,15,PASSIGN,1,1,DS,7,
      (16)1*
MOD,15,8,A,,
      10,A*
ATA,15,COM,SA,0,4,SC,2,
            SA,0,5,SC,0*
CFI,15,35,ALV,-.5,7,SA,,
      18,ALV,5.,7,SA*
```

Figure 4(2). SAINT Model Input

```
TAS,18,BRANCH,1,1,SC,0,
      (16)1*
ATA,18,COM,SA,0,1,UF,10,
           IA,0,2,UF,42,
           SA,0,4,SC,3,
           SA,0,5,SC,1*
PRO,18,SA,0,35,1,
           19,2,
           1,3*
TAS,19,PENGACC,1,1,DS,7,
      (16)1*
MOD,19,8,A*
ATA,19,COM,IA,0,3,SC,2*
CAL,19,1,ALV,.5,7,SA,,
      20,AGV,.5,7,SA,,
      46,AGV,0.,2,IA*
TAS,20,PHOLDF,1,1,DS,7*
MOD,20,8,A*
ATA,20,COM,IA,0,3,SC,3*
DET,20,1,46*
TAS,21,OBSFREND,1,1,DS,4,
      (16)1*
MOD,21,8,A,,
      10,A*
STA,21,(5)BET,STA,10,0.,30.*
ATA,21,COM,SA,0,1,UF,11*
PRO,21,SA,0,1,1,
           22,2*
TAS,22,CKFU,1,1,DS,4,
      (16)1*
MOD,22,8,A*
ATA,22,COM,SA,0,1,UF,12,
           SA,0,4,SC,7,
           SA,0,5,SC,0*
PRO,22,SA,0,1,1,
           35,2*
TAS,23,PCFIRE,1,1,DS,7,
      (16)1*
MOD,23,8,A*
ATA,23,COM,IA,0,3,SC,4*
DET,23,1,46*
TAS,24,SEARCHB,1,1,SC,0*
CFI,24,25,ALV,4.,1,SA,,
      28,ALV,5.,1,SA,,
      1,ALV,99.,1,SA*
TAS,25,OBSHOST,1,1,DS,9,
      (16)1*
MOD,25,8,A,,
      10,A*
STA,25,(5)BET,STA,10,0.,30.*
UTC,25,,,50.,50.,.8,0.*
ATA,25,COM,SA,0,1,UF,13,
           SA,0,7,SC,-1*
PRO,25,SA,0,1,1,
           15,2,
           26,3*
TAS,26,PASSIGN,1,1,DS,7,
      (16)1*
MOD,26,8,A*
ATA,26,COM,SA,0,4,SC,5,
           SA,0,5,SC,0*
DET,26,35*
TAS,27,CLEARHF,1,1,DS,7,
      (16)1*
MOD,27,8,A*
ATA,27,COM,SA,0,1,UF,14,
           IA,0,3,SC,5*
CAL,27,1,ALV,0.,1,SA,,
      26,AGV,0.,1,SA,,
      46,ALV,2.,1,SA*
```

Figure 4(3). SAINT Model Input

```
TAS,28,OBFU,1,1,DS,4*
MOD,28,8,A,,
      10,A*
STA,28,(5)BET,STA,10,0.,30.*
ATA,28,COM,SA,0,1,UF,15,
           SA,0,4,SC,6,
           SA,0,5,SC,1*
PRO,28,SA,0,1,1,
           35,2,
           33,3*
TAS,29,READOOAC,1,1,DS,9*
MOD,29,8,A*
ATA,29,COM,SA,0,1,UF,16,
           SA,0,4,SC,4,
           IA,0,3,SC,4,
           SA,0,5,SC,0*
CFI,29,30,ALU,0.,1,SA,,
      35,ALU,1.,1,SA,,
      31,ALU,2.,1,SA*
TAS,30,DROPSITE,1,1,DS,7*
MOD,30,8,A*
ATA,30,COM,SA,0,1,UF,17*
DET,30,1*
TAS,31,C2ASSIGN,1,1,UF,18*
MOD,31,8,A*
ATA,31,COM,IA,0,3,SC,4*
DET,31,35,46*
TAS,32,C1ASSIGN,1,1,DS,7*
MOD,32,8,A*
ATA,32,COM,SA,0,1,UF,19,
           IA,0,3,SC,4*
DET,32,46,30*
TAS,33,OBSDDG,1,1,DS,9*
MOD,33,8,A*
ATA,33,COM,SA,0,1,UF,8,
           SA,0,4,SC,8,
           SA,0,5,SC,1*
PRO,33,SA,0,35,1,
           1,2*
TAS,34,PCLENG,1,1,DS,7*
MOD,34,8,A*
ATA,34,COM,SA,0,1,UF,9,
           IA,0,3,SC,1*
DET,34,46,33*
TAS,35,TYPEHOOK,1,1,SC,0*
MOD,35,8,A,,
      10,A*
STA,35,M*
ATA,35,COM,SA,0,1,UF,20*
CFI,35,36,ALU,0.,1,SA,,
      39,ALU,1.,1,SA,,
      42,ALU,2.,1,SA*
TAS,36,TYPESEQ,1,1,DS,7,
      (16)2*
ATA,36,COM,SA,0,1,UF,21*
CFI,36,37,ALU,0.,1,SA,,
      38,ALU,1.,1,SA*
TAS,37,ENTCATSQ,1,1,DS,11,
      (16)2*
ATA,37,COM,SA,0,1,UF,22*
DET,37,38*
TAS,38,PSEQHOOK,1,1,DS,7,
      (16)2*
ATA,38,COM,SA,0,1,UF,23*
CFI,38,38,ALU,0.,1,SA,,
       1,ALU,1.,1,SA,,
      45,ALU,2.,1,SA*
TAS,39,ENTNUM,1,1,DS,11,
      (16)2*
DET,39,40*
```

Figure 4(4). SAINT Model Input

```
TAS,40,PNUMHOOK,1,1,DS,7,
      (16)2*
PRO,40,NO,0,41,.1,
           45,.9*
TAS,41,PDEHOOK,1,1,DS,7,
      (16)2*
DET,41,39*
TAS,42,MOVETAB,1,1,DS,6,
      (16)2*
DET,42,43*
TAS,43,PSNHOOK,1,1,DS,7,
      (16)2*
PRO,43,NO,0,44,.1,
           45,.9*
TAS,44,PDEHOOK,1,1,DS,7,
      (16)2*
DET,44,42*
TAS,45,RETHOOK,1,1,SC,0*
MOD,45,10,A*
STA,45,(5)INT,STA,10,0.,15.*
CFI,45,11,ALV,1.,4,SA,,
      18,ALV,2.,4,SA,,
      19,ALV,3.,4,SA,,
      32,ALV,4.,4,SA,,
      70,ALV,20.,4,SA*
TAS,70,RHOOKB,1,1,SC,0*
CFI,70,27,ALV,5.,4,SA,,
      29,ALV,6.,4,SA,,
      23,ALV,7.,4,SA,,
      34,ALV,8.,4,SA*
TAS,46,FUROUTER,1,1,SC,0*
MOD,46,11,A*
CFI,46,53,ALV,1.,3,IA,,
       47,ALV,2.,3,IA,,
       54,ALV,3.,3,IA,,
       57,ALV,4.,3,IA,,
       59,ALV,20.,3,IA*
TAS,59,FUROUTB,1,1,SC,0*
CFI,59,55,ALV,5.,3,IA,,
       58,ALV,6.,3,IA*
TAS,47,ATTACH,1,1,DS,12*
ATA,47,COM,SA,0,8,UF,24*
MOD,47,3,A*
CFI,47,48,AGV,0.,8,SA,,
       83,AGV,-1.,8,SA*
TAS,48,ENGAGEA,1,1,DS,13*
ATA,48,COM,SA,0,8,UF,25*
CFI,48,49,AGV,0.,8,SA,,
       84,AGV,-1.,8,SA,,
       85,AGV,-2.,8,SA,,
       83,AGV,-3.,8,SA*
TAS,49,ENGAGEB,1,1,DS,1*
MOD,49,4,A,,
     11,A*
ATA,49,COM,SA,0,8,UF,26*
CFI,49,50,AGV,0.,8,SA,,
       86,AGV,-1.,8,SA,,
       83,AGV,-2.,8,SA*
TAS,50,FIRE,1,1,DS,14*
MOD,50,2,A*
DET,50,51*
TAS,51,EVALFIRE,1,1,SC,0*
MOD,51,12,A*
ATA,51,COM,SA,0,8,UF,27*
CFI,51,53,AGV,1.,8,SA,,
       49,AGV,0.,8,SA,,
       74,AGV,-1.,8,SA*
TAS,53,CKFOR2,1,1,SC,0*
MOD,53,12,A*
ATA,53,COM,SA,0,8,UF,28*
CFI,53,49,AGV,0.,8,SA,,
       88,AGV,-1.,8,SA*
```

Figure 4(5). SAINT Model Input

```
TAS,54,HOLDFIRE,1,1,SC,0*
ATA,54,COM,SA,0,8,UF,29*
TAS,55,CLEARHF,1,1,SC,0*
ATA,55,COM,SA,0,8,UF,30*
CFI,55,50,AGV,0.,8,SA,,
       87,AGV,-1.,8,SA*
TAS,57,CEASEF,1,1,SC,0*
ATA,57,COM,SA,0,8,UF,31*
CFI,57,53,ALV,0.,8,SA,,
       87,ALV,1.,8,SA*
TAS,58,INRANGE,1,1,SC,0*
DET,58,49*
TAS,61,UDAUTO,0,1,SC,0,(10)SO*
MOD,61,10,A*
ATA,61,COM,SA,0,6,UF,32*
CFI,61,62,AGV,0.,6,SA*
TAS,62,RANGETIM,1,1,SC,10*
DET,62,63*
TAS,63,AUTOUD,1,1,SC,0*
ATA,63,COM,SA,0,6,UF,33*
CAL,63,63,ALV,1.,6,SA,,
       46,ALV,1.,6,SA,,
       64,ALV,0.,6,SA,,
       61,AGV,1.,6,SA,,
       75,AGV,0.,10,SA*
TAS,64,AUTOHF,1,1,SC,0*
ATA,64,COM,IA,0,3,SC,3*
DET,64,46*
TAS,65,STTRACK,0,1,SC,0,(9)2.,SO*
ATA,65,COM,SA,0,9,UF,34*
CAL,65,65,ALV,0.,9,SA,,
       66,ALV,1.,9,SA*
TAS,66,INITTRAK,1,1,SC,0,(9)3.*
ATA,66,COM,SA,0,9,UF,35*
MOD,66,5,A*
DET,66,67,68*
TAS,67,ROUTUD,1,1,UF,38*
ATA,67,COM,SA,0,9,UF,36*
DET,67,67*
TAS,68,STATUD,1,1,UF,39*
MOD,68,10,A*
ATA,68,COM,SA,0,9,UF,37*
CAL,68,68,ALV,1.,10,SA,,
       75,AGV,0.,10,SA*
TAS,73,OUTPUT,0,1,SC,800,(10)SO*
DET,73,73*
TAS,71,TIMER,0,1,SC,800,(10)SO*
DET,71,72*
TAS,72,SINK,1,1,(10)SI*
TAS,74,RECEFFEC,1,1,SC,0*
MOD,74,13,A,,
       14,A*
TAS,75,BRCEARA,1,1,SC,0*
ATA,75,COM,SA,0,11,UF,45*
CFI,75,76,ALV,2.,11,SA,,
       77,ALV,3.,11,SA,,
       78,ALV,4.,11,SA,,
       79,ALV,5.,11,SA*
TAS,76,CLUNKA,1,1,SC,0*
RCL,76,1,9*
TAS,77,CLFRNA,1,1,SC,0*
RCL,77,1,21*
TAS,78,CLHOSA,1,1,SC,0*
RCL,78,1,25*
TAS,79,BRCLEARB,1,1,SC,0*
ATA,79,COM,SA,0,11,UF,46*
CFI,79,80,ALV,2.,11,SA,,
       81,ALV,3.,11,SA,,
       82,ALV,4.,11,SA*
TAS,80,CLUNKB,1,1,SC,0*
RCL,80,1,9,2,9*
```

Figure 4(6). SAINT Model Input

```
TAS,81,CLFRNB,1,1,SC,0*
RCL,81,1,21,2,21*
TAS,82,CLHOSB,1,1,SC,0*
RCL,82,1,25,2,25*
TAS,83,CFTRAP,1,1,SC,0*
TAS,84,ORANTRAP,1,1,SC,0*
TAS,85,HLD2TRAP,1,1,SC,0*
TAS,86,HFTRAP,1,1,SC,0*
TAS,87,MSGTRAP,1,1,SC,0*
TAS,88,FUTRAP,1,1,SC,0*
FIN*
```

Figure 4(7). SAINT Model Input

```
TTTT
    1    1
    2   10
  10.          10.          4.          .99
  10.          -10.         4.          .99
   1 50.        80.          0.
   1 5000.      -.1          -.03
   2 50.        80.          0.
   2 500.       -.1          .03
   2 5000.      -.075        -.075
   3 0.         40.          0.
   3 5000.      -.1          0.
   4 60.        40.          0.
   4 5000.      -.1          0.
   5 120.       40.          0.
   5 5000.      -.1          0.
   6 180.       40.          0.
   6 5000.      -.1          0.
   7 240.       40.          0.
   7 5000.      -.1          0.
   8 300.       40.          0.
   8 5000.      -.1          0.
   9 360.       0.           0.
   9 5000.      .075         .075
  10 420.       0.           0.
  10 5000.      .075         -.075
  99
   1 75.        2.
   1 250.       4.
   1 5000.
   2 100.       2.
   2 200.       4.
   2 5000.
   3 10.        4.
   3 5000.
   4 70.        4.
   4 5000.
   5 130.       4.
   5 5000.
   6 190.       4.
   6 5000.
   7 250.       4.
   7 5000.
   8 310.       4.
   8 5000.
   9 370.       3.
   9 5000.
  10 430.       3.
  10 5000.
  99
```

Figure 5. Mission Input Data

Table VII

MISSION INPUT DATA

Card	Columns	Format	Definition
1	1	L1	Auto/manual initiate
	2	L1	Auto/manual interrogate
	3	L1	Auto/manual engagement
	4	L1	Tight/free policy
2	1-5	I5	Hooking policy 0 = sequence hook 1 = position/number hook 2 = tab hook
	6-10	I5	Form of sequence hook 0 = track and fire unit 1 = track 2 = fire unit 3 = hostile track and fire unit 4 = hostile track
3	1-5	I5	Number of fire units in mission
	6-10	I5	Number of tracks in mission
F1-FN	1-10	F10.0	Location of fire unit N (x coordinate in miles)
	11-20	F10.0	Location of fire unit N (y coordinate in miles)
	21-30	F10.0	Number of missiles at fire unit N
	31-40	F10.0	Effectiveness of fire unit N ($0 < E \leq 1$)
TR1/1	1-2	I2	Track number (1)
	3-12	F10.0	Time track 1 appears on screen (in seconds)
	13-22	F10.0	Initial location of track 1 (x coordinate in miles)
	23-32	F10.0	Initial location of track 1 (y coordinate in miles)
TR1/2-TR1/m	1-2	I2	Track number (1)
	3-12	F10.0	Time track 1 makes its (m-1) turn (in seconds)

Table VIII (cont.)

Card	Columns	Format	Definition
	13-22	F10.0	X velocity before the time in field 1 (miles/second)
	23-32	F10.0	Y velocity before the time in field 1 (miles/second)
(TR2/1-TR2/m) - (TRn/1-TRm/m)			Same as track 1
S1	1-2	I2	Any value larger than the number of tracks (used to signal the end of TR cards)
TS1/1	1-2	I2	Track number (1)
	3-12	F10.0	Time of first ID change for track 1
	13-22	F10.0	New ID for track 1 (0 = track disappeared) 1 = video data 2 = unknown track 3 = friendly track 4 = hostile track
TS1/2-TS1/m	2-3	I2	Track number (1)
	3-12	F10.0	Time for m^{th} ID change for track 1
	13-22	F10.0	New ID for track 1
(TS2/1-TS2/m) - (TSn/1-TSn/m)			Same as track 1
S2	1-2	I2	Any value larger than the total number of tracks (used to signal the end of TS cards)

SECTION V

EXAMPLE OF SAINT SIMULATION OUTPUT

This section presents examples of output generated by the SAINT simulation of the AN/TSQ-73 system. There are three categories of output. The first, presented in Figure 5, is an echo check of the SAINT model input. The second, shown in Figure 7, is an echo check of the AN/TSQ-73 mission input. The information contained in these two output categories was discussed in the previous sections. The third category of output, shown in Figure 8, is mission-related output generated by the simulation. Both a detailed mission output (trace) and a statistical summary output are provided.

Mission Trace Output

The mission trace provides a step-by-step account of the simulation as it progresses. A small section of the mission trace is shown here for purposes of explanation:

```
:  4.10   OBH   25  TR- 3  H  D- 25  AFU- 2       RRHR                    UF
:  9.57   SER    1  TR- 0     D-  0  AFU- 0       RR R                    UE
: 15.17   OFU   28  FU- 1  A  P- 4   S- 0        UR H                    AE
: 15.68   OFU   33  FU- 1  A  P- 4   S- 0        UR H                    AE
: 21.33   OFU*  33  FU- 1  A  P- 4   S- 0        UR H                    AE
: 31.60   OFU   34  FU- 1  X  P- 4   S- 0        UR H                    XE
: 33.85   OFU   33  FU- 2  U  P- 3   S- 0        UR H                    XU
: 40.20   SER    1  TR- 0     D-  0  AFU- 0       UU H                    FU
```

The first line above represents the simulation at time 1 minute 4.10 seconds. The current job (task) of the operator is OBH (observing hostile track), task 25. The track number associated with this target is 3; it is classified as H (hostile) and the distance to the assigned fire unit two is 25 miles. The letters, RRHR, indicate that targets one, two and four are not yet assigned as tracks, whereas target three is classified as a hostile track. The letters, UF, at the far right, indicate that fire unit one is unassigned

(U) and fire unit two is in the process of firing (F) a missile (at track 3).

At time 1 minute 21.33 seconds, the operator is OFU* (hooking an observed fire unit). The asterisk (*) is used to indicate the hooking process. This is task number 33. As there is a delay in the output at this time, the fire unit information on the current line does not correspond to the fire unit being hooked. The operator is hooking FU-2 (fire unit 2) in order to release the effective status. The letters at the far right, AE, indicate that fire unit one is attached and fire unit two is showing an effective status. This effective status is also reflected by the condition of track three. It has been removed as a track, indicating its elimination. The remainder of the line gives the information that FU-1 (fire unit one) is A (attached) with P-4 (primary assignment to track 4) and S-0 (no second assignment). Track one is U (unknown), track two is R (video) and track four is H (hostile).

As the mission proceeds (see Figure 8), the fire units eliminate all hostile tracks, but in doing so expend all their missiles. This is represented by an * (blinking fire unit) followed by a Z (out of action). At this point, the operator drops the fire unit from the scope. These events occur at times 6 minutes 11.55 seconds, 6 minutes 19.21 seconds and 6 minutes 50.85 seconds, respectively.

Statistical Summary Output

The mission trace output is followed by a series of statistical summaries that represent a variety of system performance

measures. At the present time, there are seven SAINT-generated task statistics collected, thirteen user-generated statistics based on observation, three user-generated histograms, and eleven user-generated statistics for time-persistent variables. These statistics are representative of the types that can be collected, but are by no means all inclusive. The statistics currently collected are:

1. SAINT task statistics.
    a. Task 2: The time between occurrences when the operator enters an idle period.
    b. Task 3: The time between occurrences when the operator processes video data.
    c. Task 9: The time between occurrences when the operator processes unknown tracks.
    d. Task 21: The time between occurrences when the operator processes friendly tracks.
    e. Task 25: The time between occurrences when the operator processes hostile tracks.
    f. Task 28: The time between occurrences when the operator processes fire unit sites.
    g. Task 45: The interval of time that has occurred since task 35. It represents the amount of time spent in the hooking procedures.
2. User-generated statistics for variables based on observation.
    a. SEARCHT: The amount of time spent scanning the scope in the search task.

b. IDLET: The time spent as idle time.

c. VIDEOT: The time spent processing video data.

d. UNKT: The time spent processing unknown tracks.

e. FRIENDT: The time spent processing friendly tracks.

f. HOSTILET: The time spent processing hostile tracks.

g. FIREUT: The time the operator is working with the fire units.

h. ASSIGNT: The time the operator spent assigning fire units to tracks.

i. HOOKINGT: The time spent by the operator in the hooking procedures.

j. TIMETRK: The time the system or operator spent in assigning a track to video data.

k. TIMEFRND: The time the system or operator spent in identifying a track as friendly.

l. TIMEHOST: The time the system or operator spent in identifying a track as hostile.

m. KILLT: The time from initial appearance to effective kill.

3. User-generated histograms.

a. Histogram 1: The number of occurrences that have been recorded by the operator in each of the nine possible categories.

The categories are:

(i) 1 = Search
(ii) 2 = Idle Time
(iii) 3 = Processing Video
(iv) 4 = Processing Unknown Track

(v) 5 = Processing Friendly Track
(vi) 6 = Processing Hostile Track
(vii) 7 = Observing Fire Unit
(viii) 8 = Hooking Procedures

b. Histogram 2: The number of missiles fired by each unit. Each line represents a single fire unit, where the upper cell limit is the fire unit number.

c. Histogram 3: The number of effective kills each fire unit has had. Each line represents a single fire unit, where the upper cell limit is the fire unit number.

4. User-generated statistics for time-persistent variables.

a. OBEFF: The effectiveness of the system and operator. The value gives the percentage of time that the system or operator was not up-to-date in its identification. Therefore, the smaller the value, the more efficient the system or operator was.

b. FU1-FU10: The percentage of time that the fire units were active.

SAINT SIMULATION PROJECT 1 BY ARI
DATE 3/ 1/ 1978

RUN PARAMETERS

PARAMETER	VALUE
NUMBER OF ITERATIONS	1
NUMBER OF SINK TASKS TO END ITERATION	1
INTEGER RANDOM NUMBER SEED	71268659
SCALE FACTOR FOR FUNCTION SC	1.000

PROGRAM OPTIONS

OPTION	CODE
NUMBER OF RESOURCES	2
NUMBER OF RESOURCE ATTRIBUTES PER RESOURCE	0
NUMBER OF INFORMATION ATTRIBUTES	3
NUMBER OF SYSTEM ATTRIBUTES	11
NUMBER OF MODERATOR FUNCTIONS	14
NETWORK MODIFICATION	
DISTRIBUTION SET MODIFICATION	
RANKING OF TASKS AWAITING SCHEDULING	3

OUTPUT OPTIONS

OPTION	CODE
DETAILED ITERATION OUTPUT (BEGIN)	0
DETAILED ITERATION OUTPUT (END)	0
RESOURCE UTILIZATION SUMMARY (BEGIN)	0
RESOURCE UTILIZATION SUMMARY (END)	0
STATISTICS TASK SUMMARY (BEGIN)	0
STATISTICS TASK SUMMARY (END)	0
INITIAL/FINAL STATE VARIABLE VALUES (BEGIN)	0
INITIAL/FINAL STATE VARIABLE VALUES (END)	0
STATE VARIABLE STATISTICS (BEGIN)	0
STATE VARIABLE STATISTICS (END)	0
STATE VARIABLE PLOTS/TABLES (BEGIN)	0
STATE VARIABLE PLOTS/TABLES (END)	0
RESOURCE UTILIZATION SUMMARY REPORT	NO
STATISTICS TASK SUMMARY REPORT	YES
HISTOGRAM OUTPUT FOR STATISTICS TASKS	
SUMMARY FOR ITERATION 1	YES
SUMMARY REPORT	YES

Figure 6(1). SAINT Echo Check

USER-GENERATED STATISTICS FOR VARIABLES BASED ON OBSERVATION

VARIABLE NUMBER	VARIABLE LABEL
1	SEARCHT
2	IDLET
3	VIDEOT
4	UNKT
5	FRIENDT
6	HOSTILET
7	FIREUT
8	ASSIGNT
9	HOOKINGT
10	TIMETRAK
11	TIMEFRND
12	TIMEHOST
13	KILLT

Figure 6(2). SAINT Echo Check

USER-GENERATED STATISTICS FOR TIME-PERSISTENT VARIABLES

VARIABLE NUMBER	VARIABLE LABEL	INITIAL VALUE
1	OBEFF	0
2	FU1	0
3	FU2	0
4	FU3	0
5	FU4	0
6	FU5	0
7	FU6	0
8	FU7	0
9	FU8	0
10	FU9	0
11	FU10	0

Figure 6(3). SAINT Echo Check

USER-GENERATED HISTOGRAMS

VARIABLE NUMBER	VARIABLE LABEL	NUMBER OF CELLS	UPPER LIMIT OF FIRST CELL	CELL WIDTH
1	OPERATOR	12	0	1.0000E+00
2	FUOPERAT	12	0	1.0000E+00
3	FUEFFECT	12	0	1.0000E+00

Figure 6(4). SAINT Echo Check

INITIAL MODERATOR FUNCTION STATUS

MODERATOR FUNCTION	INITIAL STATUS
1	INAC
2	INAC
3	INAC
4	INAC
5	INAC
6	ACT
7	INAC
8	INAC
9	INAC
10	INAC
11	INAC
12	INAC
13	INAC
14	INAC

Figure 6(5). SAINT Echo Check

DISTRIBUTION SETS

SET NUMBER	DISTRIBUTION TYPE	PARAMETERS 1	2	3	4
1	UN	0	0	1.0000	0
2	UN	0	1.0000	5.0000	0
3	UN	0	1.0000	10.0000	0
4	UN	0	.5000	1.5000	0
5	UN	0	2.0000	6.0000	0
6	UN	0	1.0000	5.0000	0
7	UN	0	2.0000	4.0000	0
8	UN	0	2.0000	5.0000	0
9	UN	0	4.0000	8.0000	0
10	UN	0	10.0000	20.0000	0
11	UN	0	4.0000	9.0000	0
12	UN	0	10.0000	20.0000	0
13	UN	0	10.0000	20.0000	0
14	UN	0	20.0000	40.0000	0
15	UN	0	45.0000	65.0000	0
16	UN	0	30.0000	45.0000	0

Figure 6(6). SAINT Echo Check

RESOURCE DESCRIPTIONS

RESOURCE NUMBER	RESOURCE LABEL	ATTRIBUTE NUMBER	ATTRIBUTE VALUE
1	----		
2	----		

Figure 6(7). SAINT Echo Check

INITIAL SYSTEM ATTRIBUTE VALUES

ATTRIBUTE NUMBER	ATTRIBUTE VALUE
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0

Figure 6(8). SAINT Echo Check

TASK DEFINITIONS

TASK NUMBER	TASK LABEL	SPEC CHAR	PREDECESSOR FIRST	PREDECESSOR SUBS	REQTS DIFF	PERFORMANCE FUNC	PERFORMANCE PMTR	TIME RESR	TASK PRTY	INFO CODE	CHOICE ATRIB	COMP PREC	RESR CODE	RESOURCES ASSOCIATED WITH THIS TASK
1	SEARCH	SOU	0	1		SC	0		0	LAS		0	AND	
2	IDLETIME		1	1		UF	40		0	LAS		0	AND	
3	OBSVIDEO		1	1		DS	4		0	LAS		0	AND	1
4	WAITONE		1	1		DS	5		0	LAS		0	AND	1
5	AUTOMANN		1	1		SC	0		0	LAS		0	AND	1
6	WATCHVID		1	1		UF	3		0	LAS		0	AND	1
7	POSTAB		1	1		DS	6		0	LAS		0	AND	1
8	PINDICAT		1	1		DS	7		0	LAS		0	AND	1
9	OBSUNK		1	1		DS	4		0	LAS		0	AND	1
10	PIDIFF		1	1		DS	7		0	LAS		0	AND	1
11	PINTERRO		1	1		DS	7		0	LAS		0	AND	1
12	READMSG		1	1		DS	8		0	LAS		0	AND	1
13	PFH		1	1		DS	7		0	LAS		0	AND	1
14	TIGHGREE		1	1		SC	0		0	LAS		0	AND	1
15	PASSIGN		1	1		DS	7		0	LAS		0	AND	1
18	BRANCH		1	1		SC	0		0	LAS		0	AND	1
19	PENGACC		1	1		DS	7		0	LAS		0	AND	1
20	PHOLDF		1	1		DS	7		0	LAS		0	AND	
21	OBSFREND		1	1		DS	4		0	LAS		0	AND	1
22	CKFU		1	1		DS	4		0	LAS		0	AND	1
23	PCFIRE		1	1		DS	7		0	LAS		0	AND	1
24	SEARCHB		1	1		SC	0		0	LAS		0	AND	
25	OBSHOST		1	1		DS	9		0	LAS		0	AND	1
26	PASSIGN		1	1		DS	7		0	LAS		0	AND	1
27	CLEARHF		1	1		DS	7		0	LAS		0	AND	1
28	OBFU		1	1		DS	4		0	LAS		0	AND	
29	READOOAC		1	1		DS	9		0	LAS		0	AND	
30	DROPSITE		1	1		DS	7		0	LAS		0	AND	
31	C2ASSIGN		1	1		UF	18		0	LAS		0	AND	
32	C1ASSIGN		1	1		DS	7		0	LAS		0	AND	
33	OBSDDG		1	1		DS	9		0	LAS		0	AND	
34	PCLENG		1	1		DS	7		0	LAS		0	AND	
35	TYPEHOOK		1	1		SC	0		0	LAS		0	AND	
36	TYPESEQ		1	1		DS	7		0	LAS		0	AND	2
37	ENTCATSQ		1	1		DS	11		0	LAS		0	AND	2
38	PSEQHOOK		1	1		DS	7		0	LAS		0	AND	2
39	ENTNUM		1	1		DS	11		0	LAS		0	AND	2
40	PNUMHOOK		1	1		DS	7		0	LAS		0	AND	2
41	PDEHOOK		1	1		DS	7		0	LAS		0	AND	2
42	MOVETAB		1	1		DS	6		0	LAS		0	AND	2
43	PSNHOOK		1	1		DS	7		0	LAS		0	AND	2
44	PDEHOOK		1	1		DS	7		0	LAS		0	AND	2
45	RETHOOK		1	1		SC	0		0	LAS		0	AND	
46	FUROUTER		1	1		SC	0		0	LAS		0	AND	
47	ATTACH		1	1		DS	12		0	LAS		0	AND	
48	ENGAGEA		1	1		DS	13		0	LAS		0	AND	
49	ENGAGEB		1	1		DS	1		0	LAS		0	AND	
50	FIRE		1	1		DS	14		0	LAS		0	AND	

Figure 6(9). SAINT Echo Check

51	EVALFIRE		1	1	SC	0	0	LAS	0	AND
53	CKFOR2		1	1	SC	0	0	LAS	0	AND
54	HOLDFIRE		1	1	SC	0	0	LAS	0	AND
55	CLEARHF		1	1	SC	0	0	LAS	0	AND
57	CEASEF		1	1	SC	0	0	LAS	0	AND
58	INRANGE		1	1	SC	0	0	LAS	0	AND
59	FUROUTB		1	1	SC	0	0	LAS	0	AND
61	UDAUTO	SOU	0	1	SC	0	0	LAS	0	AND
62	RANGETIM		1	1	SC	10	0	LAS	0	AND
63	AUTOUD		1	1	SC	0	0	LAS	0	AND
64	AUTOHF		1	1	SC	0	0	LAS	0	AND
65	STTRACK	SOU	0	1	SC	0	2.00	LAS	0	AND
66	INITTRAK		1	1	SC	0	3.00	LAS	0	AND
67	ROUTUD		1	1	UF	38	0	LAS	0	AND
68	STATUD		1	1	UF	39	0	LAS	0	AND
70	RHOOKB		1	1	SC	0	0	LAS	0	AND
71	TIMER	SOU	0	1	SC	0	0	LAS	0	AND
72	SINK	SIN	1	1	SC	0	0	LAS	0	AND
73	OUTPUT	SOU	0	1	SC	800	0	LAS	0	AND
74	RECEFFEC		1	1	SC	0	0	LAS	0	AND
75	BRCEARA		1	1	SC	0	0	LAS	0	AND
76	CLUNKA		1	1	SC	0	0	LAS	0	AND
77	CLFRNA		1	1	SC	0	0	LAS	0	AND
78	CLHOSA		1	1	SC	0	0	LAS	0	AND
79	BRCLEARB		1	1	SC	0	0	LAS	0	AND
80	CLUNKB		1	1	SC	0	0	LAS	0	AND
81	CLFRNB		1	1	SC	0	0	LAS	0	AND
82	CLHOSB		1	1	SC	0	0	LAS	0	AND
83	CFTRAP		1	1	SC	0	0	LAS	0	AND
84	ORANTRAP		1	1	SC	0	0	LAS	0	AND
85	HLD2TRAP		1	1	SC	0	0	LAS	0	AND
86	HFTRAP		1	1	SC	0	0	LAS	0	AND
87	MSGTRAP		1	1	SC	0	0	LAS	0	AND
88	FUTRAP		1	1	SC	0	0	LAS	0	AND

Figure 6(10). SAINT Echo Check

MARK AND STATISTICS TASKS

TASK NUMBER	MARK POINT	STATISTICS TYPE	COLLECTION POINT	HISTOGRAM		
				NO. CELLS	UPPER LIMIT	CELL WIDTH
2		BET	STA	10	0	30.00
3		BET	STA	10	0	30.00
9		BET	STA	10	0	30.00
21		BET	STA	10	0	30.00
25		BET	STA	10	0	30.00
28		BET	STA	10	0	30.00
35	COM					
45		INT	STA	10	0	15.00

Figure 6(11). SAINT Echo Check

USER DEFINED TASK CHARACTERISTICS

TASK NUMBER	CHARACTERISTIC NUMBER	CHARACTERISTIC VALUE
9	1	0
	2	0
	3	40.00
	4	60.00
	5	1.00
	6	.20
12	1	0
	2	0
	3	50.00
	4	50.00
	5	1.00
	6	.10
25	1	0
	2	0
	3	50.00
	4	50.00
	5	.80
	6	0

Figure 6(12). SAINT Echo Check

MODERATOR FUNCTION STATUS UPDATES

TASK NUMBER	UPDATE		
	FUNCTION	STATUS	DURATION
1	1	ACT	TASK
	8	ACT	TASK
	10	ACT	TASK
2	8	ACT	TASK
	10	ACT	TASK
3	8	ACT	TASK
	10	ACT	TASK
4	8	ACT	TASK
6	8	ACT	TASK
7	8	ACT	TASK
8	8	ACT	TASK
	9	ACT	TASK
9	8	ACT	TASK
	10	ACT	TASK
10	8	ACT	TASK
11	8	ACT	TASK
12	8	ACT	TASK
13	8	ACT	TASK
	9	ACT	TASK
15	8	ACT	TASK
	10	ACT	TASK
19	8	ACT	TASK
20	8	ACT	TASK
21	8	ACT	TASK
	10	ACT	TASK
22	8	ACT	TASK
23	8	ACT	TASK
25	8	ACT	TASK
	10	ACT	TASK
26	8	ACT	TASK
27	8	ACT	TASK
28	8	ACT	TASK
	10	ACT	TASK

Figure 6(13). SAINT Echo Check

29	8	ACT	TASK
30	8	ACT	TASK
31	8	ACT	TASK
32	8	ACT	TASK
33	8	ACT	TASK
34	8	ACT	TASK
35	8	ACT	TASK
	10	ACT	TASK
45	10	ACT	TASK
46	11	ACT	TASK
47	3	ACT	TASK
49	4	ACT	TASK
	11	ACT	TASK
50	2	ACT	TASK
51	12	ACT	TASK
53	12	ACT	TASK
61	10	ACT	TASK
66	5	ACT	TASK
68	10	ACT	TASK
74	13	ACT	TASK
	14	ACT	TASK

Figure 6(14). SAINT Echo Check

ATTRIBUTE ASSIGNMENT INFORMATION

TASK NUMBER	ASSIGNMENT POINT	ASSIGNMENT TYPE	RESR NUMBER	ATRIB NUMBER	FUNCTION TYPE	PARAMETER SPEC
1	COM	SA		1	UF	43
3	COM	SA		1	UF	1
5	COM	SA		1	UF	2
6	COM	SA		1	UF	41
8	COM	SA		1	UF	4
9	COM	SA		1	UF	5
10	COM	SA		4	SC	1
		SA		5	SC	0
12	COM	SA		1	UF	6
14	COM	SA		1	UF	7
15	COM	SA		4	SC	2
		SA		5	SC	0
18	COM	SA		1	UF	10
		IA		2	UF	42
		SA		4	SC	3
		SA		5	SC	1
19	COM	IA		3	SC	2
20	COM	IA		3	SC	3
21	COM	SA		1	UF	11
22	COM	SA		1	UF	12
		SA		4	SC	7
		SA		5	SC	0
23	COM	IA		3	SC	4
25	COM	SA		1	UF	13
		SA		7	SC	-1
26	COM	SA		4	SC	5
		SA		5	SC	0
27	COM	SA		1	UF	14
		IA		3	SC	5
28	COM	SA		1	UF	15
		SA		4	SC	6
		SA		5	SC	1
29	COM	SA		1	UF	16
		SA		4	SC	4
		IA		3	SC	4

Figure 6(15). SAINT Echo Check

		SA	5	SC	0
30	COM	SA	1	UF	17
31	COM	IA	3	SC	4
32	COM	SA	1	UF	19
		IA	3	SC	4
33	COM	SA	1	UF	8
		SA	4	SC	8
		SA	5	SC	1
34	COM	SA	1	UF	9
		IA	3	SC	1
35	COM	SA	1	UF	20
36	COM	SA	1	UF	21
37	COM	SA	1	UF	22
38	COM	SA	1	UF	23
47	COM	SA	8	UF	24
48	COM	SA	8	UF	25
49	COM	SA	8	UF	26
51	COM	SA	8	UF	27
53	COM	SA	8	UF	28
54	COM	SA	8	UF	29
55	COM	SA	8	UF	30
57	COM	SA	8	UF	31
61	COM	SA	6	UF	32
63	COM	SA	6	UF	33
64	COM	IA	3	SC	3
65	COM	SA	9	UF	34
66	COM	SA	9	UF	35
67	COM	SA	9	UF	36
68	COM	SA	9	UF	37
75	COM	SA	11	UF	45
79	COM	SA	11	UF	46

Figure 6(16). SAINT Echo Check

DETERMINISTIC BRANCHING

TASK NUMBER	SUCCESSOR TASKS	
2	1	
4	3	
7	8	
8	1	
10	35	
11	12	
13	1	
20	1	46
23	1	46
26	35	
30	1	
31	35	46
32	46	30
34	46	33
37	38	
39	40	
41	39	
42	43	
44	42	
50	51	
58	49	
62	63	
64	46	
66	67	68
67	67	
71	72	
73	73	

Figure 6(17). SAINT Echo Check

PROBABILISTIC BRANCHING

TASK NUMBER	SUCC TASK	PROB/ ATRIB	ATRB TYPE	RESR NUMBER
3	1	1	SA	
	4	2	SA	
	5	3	SA	
5	1	1	SA	
	6	2	SA	
	7	3	SA	
9	1	1	SA	
	10	2	SA	
12	1	1	SA	
	13	2	SA	
	14	3	SA	
14	1	1	SA	
	15	2	SA	
18	35	1	SA	
	19	2	SA	
	1	3	SA	
21	1	1	SA	
	22	2	SA	
22	1	1	SA	
	35	2	SA	
25	1	1	SA	
	15	2	SA	
	26	3	SA	
28	1	1	SA	
	35	2	SA	
	33	3	SA	
33	35	1	SA	
	1	2	SA	
40	41	.1000		
	45	.9000		
43	44	.1000		
	45	.9000		

Figure 6(18). SAINT Echo Check

CONDITIONAL BRANCHING

TASK NUMBER	BRANCH TYPE	SUCC TASK	CONDITION CODE	ATRB/ VALUE	ATRB TYPE	RESR NUMBER	COMPARED ATTRIBUTE
1	FIR	2	ALV	0	SA		1
		3	ALV	1.00	SA		1
		9	ALV	2.00	SA		1
		21	ALV	3.00	SA		1
		24	ALV	9.00	SA		1
6	FIR	1	AGV	1.00	SA		1
		7	ALV	1.00	SA		1
15	FIR	35	ALV	-.50	SA		7
		18	ALV	5.00	SA		7
19	ALL	1	ALV	.50	SA		7
		20	AGV	.50	SA		7
		46	AGV	0	IA		2
24	FIR	25	ALV	4.00	SA		1
		28	ALV	5.00	SA		1
		1	ALV	99.00	SA		1
27	ALL	1	ALV	0	SA		1
		26	AGV	0	SA		1
		46	ALV	2.00	SA		1
29	FIR	30	ALV	0	SA		1
		35	ALV	1.00	SA		1
		31	ALV	2.00	SA		1
35	FIR	36	ALV	0	SA		1
		39	ALV	1.00	SA		1
		42	ALV	2.00	SA		1
36	FIR	37	ALV	0	SA		1
		38	ALV	1.00	SA		1
38	FIR	38	ALV	0	SA		1
		1	ALV	1.00	SA		1
		45	ALV	2.00	SA		1
45	FIR	11	ALV	1.00	SA		4
		18	ALV	2.00	SA		4
		19	ALV	3.00	SA		4
		32	ALV	4.00	SA		4
		70	ALV	20.00	SA		4
46	FIR	53	ALV	1.00	IA		3
		47	ALV	2.00	IA		3
		54	ALV	3.00	IA		3
		57	ALV	4.00	IA		3
		59	ALV	20.00	IA		3
47	FIR	48	AGV	0	SA		8
		83	AGV	-1.00	SA		8
48	FIR	49	AGV	0	SA		8

Figure 6(19). SAINT Echo Check

		84	AGV	-1.00	SA	8
		85	AGV	-2.00	SA	8
		83	AGV	-3.00	SA	8
49	FIR	50	AGV	0	SA	8
		86	AGV	-1.00	SA	8
		83	AGV	-2.00	SA	8
51	FIR	53	AGV	1.00	SA	8
		49	AGV	0	SA	8
		74	AGV	-1.00	SA	8
53	FIR	49	AGV	0	SA	8
		88	AGV	-1.00	SA	8
55	FIR	50	AGV	0	SA	8
		87	AGV	-1.00	SA	8
57	FIR	53	ALV	0	SA	8
		87	ALV	1.00	SA	8
59	FIR	55	ALV	5.00	IA	3
		58	ALV	6.00	IA	3
61	FIR	62	AGV	0	SA	6
63	ALL	63	ALV	1.00	SA	6
		46	ALV	1.00	SA	6
		64	ALV	0	SA	6
		61	AGV	1.00	SA	6
		75	AGV	0	SA	10
65	ALL	65	ALV	0	SA	9
		66	ALV	1.00	SA	9
68	ALL	68	ALV	1.00	SA	10
		75	AGV	0	SA	10
70	FIR	27	ALV	5.00	SA	4
		29	ALV	6.00	SA	4
		23	ALV	7.00	SA	4
		34	ALV	8.00	SA	4
75	FIR	76	ALV	2.00	SA	11
		77	ALV	3.00	SA	11
		78	ALV	4.00	SA	11
		79	ALV	5.00	SA	11
79	FIR	80	ALV	2.00	SA	11
		81	ALV	3.00	SA	11
		82	ALV	4.00	SA	11

Figure 6(20). SAINT Echo Check

RESOURCE CLEARING

TASK NUMBER	CLEAR RESR	SIGNAL TASK	CLEAR RESR	SIGNAL TASK	CLEAR RESR	SIGNAL TASK	CLEAR RESR	SIGNAL TASK
76	1	9						
77	1	21						
78	1	25						
80	1	9	2	9				
81	1	21	2	21				
82	1	25	2	25				

Figure 6(21). SAINT Echo Check

STATE VARIABLE GENERAL INFORMATION

NUMBER OF EQUATIONS WRITTEN IN DD	=	0
NUMBER OF EQUATIONS WRITTEN IN SS	=	39
INTEGRATION ERROR OPTION	=	WARN
ABSOLUTE INTEGRATION ERROR ALLOWED	=	1.0000E-05
RELATIVE INTEGRATION ERROR ALLOWED	=	1.0000E-05
MINIMUM STEP SIZE	=	1.0000E+00
MAXIMUM STEP SIZE	=	1.0000E+03
COMMUNICATION INTERVAL	=	1.0000E+20

Figure 6(22). SAINT Echo Check

STATE VARIABLE DESCRIPTIONS

STATE VARIABLE NUMBER	STATE VARIABLE LABEL
1	----
2	----
3	----
4	----
5	----
6	----
7	----
8	----
9	----
10	----
11	----
12	----
13	----
14	----
15	----
16	----
17	----
18	----
19	----
20	----
21	----
22	----
23	----
24	----
25	----
26	----
27	----
28	----
29	----
30	----
31	----
32	----
33	----
34	----
35	----
36	----
37	----
38	----
39	----

Figure 6(23). SAINT Echo Check

SAINT SIMULATION

OF THE

AN/TSQ-73

GUIDED MISSILE AIR DEFENSE SYSTEM

OPERATIONAL DATA

INITIAL OPERATIONAL MODES/POLICIES

AUTO/MANUAL INITIATE	AUTO
AUTO/MANUAL INTERROGATE	AUTO
AUTO/MANUAL ENGAGEMENT	AUTO
TIGHT/FREE ENGAGEMENT	TIGHT
HOOKING POLICY	POSITION

ASSOCIATED FIRE UNIT INFORMATION

NO	LOCATION X-CORD	LOCATION Y-CORD	QUANTITY WEAPONS	EFFECT RATIO
1	10.00	10.00	4	.990
2	10.00	-10.00	4	.990

Figure 7(1). Mission Echo Check

			TRACK INFORMATION					
			LOCATION		VELOCITY			
NO	TIME	ID	X-CORD	Y-CORD	X-VEL (MILES / SEC)	Y-VEL	SPEED (MILES / HOUR)	HEADING
1	50.00	VIDEO	80.00	0	-.100	-.030	375.851	253
	75.00	UNKNOWN	77.50	-.75	-.100	-.030	375.851	253
	250.00	HOSTILE	60.00	-6.00	-.100	-.030	375.851	253
2	50.00	VIDEO	80.00	0	-.100	.030	375.851	286
	100.00	UNKNOWN	75.00	1.50	-.100	.030	375.851	286
	200.00	HOSTILE	65.00	4.50	-.100	.030	375.851	286
	500.00	HOSTILE	35.00	13.50	-.075	-.075	381.838	225
3	0	VIDEO	40.00	0	-.100	0	360.000	270
	10.00	HOSTILE	39.00	0	-.100	0	360.000	270
4	60.00	VIDEO	40.00	0	-.100	0	360.000	270
	70.00	HOSTILE	39.00	0	-.100	0	360.000	270
5	120.00	VIDEO	40.00	0	-.100	0	360.000	270
	130.00	HOSTILE	39.00	0	-.100	0	360.000	270
6	180.00	VIDEO	40.00	0	-.100	0	360.000	270
	190.00	HOSTILE	39.00	0	-.100	0	360.000	270
7	240.00	VIDEO	40.00	0	-.100	0	360.000	270
	250.00	HOSTILE	39.00	0	-.100	0	360.000	270
8	300.00	VIDEO	40.00	0	-.100	0	360.000	270
	310.00	HOSTILE	39.00	0	-.100	0	360.000	270
9	360.00	VIDEO	0	0	.075	.075	381.838	45
	370.00	FRIENDLY	.75	.75	.075	.075	381.838	45
10	420.00	VIDEO	0	0	.075	-.075	381.838	135
	430.00	FRIENDLY	.75	-.75	.075	-.075	381.838	135

Figure 7(2). Mission Echo Check

MISSION TRACE INFORMATION

FIELDS	SYMBOL	USE/MEANING
1,2		TIME IN MINUTES AND SECONDS
3		CURRENT OPERATOR JOB AREA
	SER	SEARCH SCOPE
	IDL	IDLE TIME
	OBR	OBSERVE/PROCESS VIDEO
	OBU	OBSERVE/PROCESS UNKNOWN TRACK
	OBF	OBSERVE/PROCESS FRIENDLY TRACK
	OBH	OBSERVE/PROCESS HOSTILE TRACK
	ASS	ASSIGN FIRE UNIT TO TRACK
	OFU	OBSERVE/PROCESS FIRE UNIT
	*	HOOKING A SITE OR TRACK
4		SAINT TASK NUMBER
5	TR	TRACK NUMBER ASSOCIATED WITH ACTION
	FU	FIRE UNIT NO ASSOCIATED WITH ACTION
6		STATUS OF TRACK
	R	*VIDEO*
	U	UNKNOWN TRACK
	F	FRIENDLY TRACK
	H	HOSTILE TRACK
	S	SPECIAL SYMBOL
		STATUS OF FIRE UNIT
	U	UNUSED
	A	ACCESSED
	X	ENGAGED
	F	FIRING
	E	EFFECTIVE
	I	INEFFECTIVE
	Z	NOT OPERATIONAL
	D	DISENGAGE
	C	CEASE FIRE
	*	BLINKING (OUT OF ACTION)
7		TRACK - DISTANCE FIRE UNIT - PRIMARY ASSIGNMENT
8		TRACK - ATTACHED FIRE UNIT FIRE UNIT - SECONDARY ASSIGNMENT
9	(SEE 6)	ALL TRACKS STATUS
10	(SEE 6)	ALL FIRE UNITS STATUS

Figure 8(1). Mission Output

TIME	JOB	TASK NO							1111111111222222222233333 12345678901234567890123456789012345678901234	1 1234567890
0 0	SER	1	TR- 0		D- 0	AFU- 0				UU
0 36.15	IDL	2	TR- 0		D- 0	AFU- 0			H	UX
0 39.54	SER	1	TR- 0		D- 0	AFU- 0			H	UX
0 51.11	OFU	28	FU- 2	F	P- 3	S- 0			RRH	UF
0 52.30	OFU	33	FU- 2	F	P- 3	S- 0			RRH	UF
0 56.30	SER	1	TR- 0		D- 0	AFU- 0			RRH	UF
1 4.10	OBH	25	TR- 3	H	D- 25	AFU- 2			RRHR	UF
1 9.57	SER	1	TR- 0		D- 0	AFU- 0			RR R	UE
1 15.17	OFU	28	FU- 1	A	P- 4	S- 0			UR H	AE
1 15.68	OFU	33	FU- 1	A	P- 4	S- 0			UR H	AE
1 21.33	OFU*	33	FU- 1	A	P- 4	S- 0			UR H	AE
1 31.69	OFU	34	FU- 1	X	P- 4	S- 0			UR H	XE
1 33.85	OFU	33	FU- 2	U	P- 3	S- 0			UR H	XU
1 40.20	SER	1	TR- 0		D- 0	AFU- 0			UU H	FU
1 49.28	OBH	25	TR- 4	H	D- 26	AFU- 1			UU H	FU
1 53.94	SER	1	TR- 0		D- 0	AFU- 0			UU H	FU
2 1.92	OBU	9	TR- 2	U	D- 72	AFU- 0			UU HR	FU
2 2.49	SER	1	TR- 0		D- 0	AFU- 0			UU HR	FU
2 6.33	OFU	28	FU- 2	U	P- 0	S- 0			UU HR	FU
2 7.02	OFU	33	FU- 2	U	P- 0	S- 0			UU HR	FU
2 12.28	OFU*	33	FU- 2	A	P- 5	S- 0			UU H	EA
2 38.00	OFU	34	FU- 2	F	P- 5	S- 0			UU H	EF
2 41.90	OFU	33	FU- 1	U	P- 4	S- 0			UU H	UF
2 45.99	SER	1	TR- 0		D- 0	AFU- 0			UU H	UF
2 52.23	OBU	9	TR- 1	U	D- 67	AFU- 0			UU H	UF
2 53.71	SER	1	TR- 0		D- 0	AFU- 0			UU H	UF
3 2.10	OFU	28	FU- 1	U	P- 0	S- 0			UU R	UE
3 3.49	OFU	33	FU- 1	U	P- 0	S- 0			UU R	UE
3 10.91	OFU*	33	FU- 1	A	P- 6	S- 0			UU H	AE
3 17.56	OFU	34	FU- 1	A	P- 6	S- 0			UU H	AE
3 20.23	OFU	33	FU- 2	U	P- 5	S- 0			UH H	AU
3 24.49	SER	1	TR- 0		D- 0	AFU- 0			UH H	AU
3 30.86	OBH	25	TR- 2	H	D- 64	AFU- 0			UH H	XU
3 38.15	SER	1	TR- 0		D- 0	AFU- 0			UH H	XU
3 45.14	OBU	9	TR- 1	U	D- 62	AFU- 0			UH H	FU
3 46.09	OBU	10	TR- 1	U	D- 62	AFU- 0			UH H	FU
3 48.39	OBU*	10	TR- 1	U	D- 62	AFU- 0			UH H	FU
3 59.03	OBU	11	TR- 1	U	D- 61	AFU- 0			UH H	FU
4 2.48	OBU	12	TR- 1	U	D- 61	AFU- 0			UH HR	FU
4 6.01	SER	1	TR- 0		D- 0	AFU- 0			UH HR	FU
4 8.59	OBH	25	TR- 2	H	D- 60	AFU- 0			UH HR	FU
4 10.00	OBH	25	TR- 2	H	D- 60	AFU- 1			HH HH	FA
4 15.33	SER	1	TR- 0		D- 0	AFU- 0			HH HH	FA
4 22.77	IDL	2	TR- 0		D- 0	AFU- 0			HH H	EX
4 25.66	SER	1	TR- 0		D- 0	AFU- 0			HH H	EX
4 33.11	OBH	25	TR- 1	H	D- 47	AFU- 2			HH H	EX
4 40.01	SER	1	TR- 0		D- 0	AFU- 0			HH H	EF
4 47.08	OFU	28	FU- 1	E	P- 6	S--2			HH H	EF
4 48.28	OFU	33	FU- 1	E	P- 6	S--2			HH H	EF
4 55.53	OFU*	33	FU- 1	E	P- 6	S--2			HH H	EF
5 7.92	OFU	34	FU- 1	E	P- 6	S--2			HH R	EE
5 10.22	OFU	33	FU- 1	U	P- 6	S--2			HH H	UE
5 18.07	OFU*	33	FU- 1	U	P- 2	S- 0			HH H	UE
5 28.84	OFU	34	FU- 1	U	P- 2	S- 8			HH H	UE
5 32.35	OFU	33	FU- 2	U	P- 7	S--1			HH H	UU
5 38.03	SER	1	TR- 0		D- 0	AFU- 0			HH H	UF

Figure 8(2). Mission Output

TIME	JOB	TASK NO							1111111111222222222233333 123456789012345678901234567890123	1 1234567890
5 42.44	OBH	25	TR- 2	H	D- 40	AFU- 1		HH H		UF
5 49.52	SER	1	TR- 0		D- 0	AFU- 0		HH H		UF
5 55.48	OFU	28	FU- 2	F	P- 1	S- 0		HH H		UF
5 56.45	SER	1	TR- 0		D- 0	AFU- 0		HH H		UF
6 3.84	IDL	2	TR- 0		D- 0	AFU- 0		HH HR		FF
6 11.55	SER	1	TR- 0		D- 0	AFU- 0		H HF		F*
6 18.54	OFU	28	FU- 2	*	P- 1	S- 0		H HF		F*
6 19.21	OFU*	28	FU- 2	Z	P- 1	S- 0		H HF		FZ
6 29.09	OFU	29	FU- 2	Z	P- 1	S- 0		H HF		FZ
6 35.70	OFU*	29	FU- 2	Z	P- 1	S- 0		HF		EZ
6 43.38	OFU	32	FU- 2	Z	P- 1	S- 0		HF		EZ
6 47.23	OFU	30	FU- 2	Z	P- 1	S- 0		HF		EZ
6 50.85	SER	1	TR- 0		D- 0	AFU- 0		HF		E
6 58.31	OFU	28	FU- 1	E	P- 2	S--8		HF		E
6 59.38	OFU	33	FU- 1	E	P- 2	S--8		HF		E
7 4.84	OFU*	33	FU- 1	E	P- 2	S--8		HFR		E
7 26.12	OFU	34	FU- 1	E	P- 2	S--8		HFF		E
7 28.96	OFU	33	FU- 1	U	P- 2	S--8		HFF		U
7 35.47	SER	1	TR- 0		D- 0	AFU- 0		HFF		F
7 39.45	OFU	28	FU- 1	F	P- 8	S- 0		HFF		F
7 40.31	SER	1	TR- 0		D- 0	AFU- 0		HFF		F
7 42.17	IDL	2	TR- 0		D- 0	AFU- 0		HFF		F
7 51.05	SER	1	TR- 0		D- 0	AFU- 0		HFF		F
7 56.51	OBH	25	TR- 8	H	D- 15	AFU- 1		HFF		F
8 2.99	SER	1	TR- 0		D- 0	AFU- 0		FF		*
8 9.53	OBF	21	TR- 9	F	D- 13	AFU- 0		FF		*
8 11.02	SER	1	TR- 0		D- 0	AFU- 0		FF		*
8 25.04	OBF	21	TR-10	F	D- 9	AFU- 0		FF		*
8 26.04	SER	1	TR- 0		D- 0	AFU- 0		FF		*
8 36.11	IDL	2	TR- 0		D- 0	AFU- 0		FF		*
8 44.65	SER	1	TR- 0		D- 0	AFU- 0		FF		*
8 55.65	IDL	2	TR- 0		D- 0	AFU- 0		FF		*
9 2.68	SER	1	TR- 0		D- 0	AFU- 0		FF		*
9 5.64	OFU	28	FU- 1	*	P- 8	S- 0		FF		*
9 6.78	OFU*	28	FU- 1	Z	P- 8	S- 0		FF		Z
9 16.29	OFU	29	FU- 1	Z	P- 8	S- 0		FF		Z
9 23.15	OFU*	29	FU- 1	Z	P- 8	S- 0		FF		Z
9 31.87	OFU	32	FU- 1	Z	P- 8	S- 0		FF		Z
9 34.64	OFU	30	FU- 1	Z	P- 8	S- 0		FF		Z
9 36.96	SER	1	TR- 0		D- 0	AFU- 0		FF		
9 51.46	OBF	21	TR- 9	F	D- 24	AFU- 0		FF		
9 52.10	SER	1	TR- 0		D- 0	AFU- 0		FF		
10 4.49	IDL	2	TR- 0		D- 0	AFU- 0		FF		
10 7.43	SER	1	TR- 0		D- 0	AFU- 0		FF		
10 25.10	IDL	2	TR- 0		D- 0	AFU- 0		FF		
10 30.86	SER	1	TR- 0		D- 0	AFU- 0		FF		
10 39.75	IDL	2	TR- 0		D- 0	AFU- 0		FF		
10 45.52	SER	1	TR- 0		D- 0	AFU- 0		FF		
10 51.98	IDL	2	TR- 0		D- 0	AFU- 0		FF		
10 54.88	SER	1	TR- 0		D- 0	AFU- 0		FF		
11 8.99	IDL	2	TR- 0		D- 0	AFU- 0		FF		
11 15.59	SER	1	TR- 0		D- 0	AFU- 0		FF		
11 23.68	IDL	2	TR- 0		D- 0	AFU- 0		FF		
11 26.39	SER	1	TR- 0		D- 0	AFU- 0		FF		
11 40.23	IDL	2	TR- 0		D- 0	AFU- 0		FF		
11 45.40	SER	1	TR- 0		D- 0	AFU- 0		FF		

Figure 8(3). Mission Output

T I M E	JOB	TASK NO						1111111111222222222233333 12345678901234567890123456789012345678901234567890	1 1234567890
11 53.96	OBF	21	TR-10	F	D- 31	AFU- 0		FF	
11 55.20	SER	1	TR- 0		D- 0	AFU- 0		FF	
12 14.46	OBF	21	TR- 9	F	D- 39	AFU- 0		FF	
12 15.12	SER	1	TR- 0		D- 0	AFU- 0		FF	
12 26.47	IDL	2	TR- 0		D- 0	AFU- 0		FF	
12 29.87	SER	1	TR- 0		D- 0	AFU- 0		FF	
12 38.57	OBF	21	TR-10	F	D- 35	AFU- 0		FF	
12 39.38	SER	1	TR- 0		D- 0	AFU- 0		FF	
13 8.15	IDL	2	TR- 0		D- 0	AFU- 0		FF	
13 12.58	SER	1	TR- 0		D- 0	AFU- 0		FF	

Figure 8(4). Mission Output

HISTOGRAM OF THE FIRST ITERATION VALUES OF THE BET STA STATISTIC FOR TASK 2 (IDLETIME)

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
				+ + + + + + + + + + +
0	0	0	0	+
7	.500	.500	3.0000E+01	+**********
3	.214	.714	6.0000E+01	+********** C
1	.071	.786	9.0000E+01	+***** C
2	.143	.929	1.2000E+02	+******** C
0	0	.929	1.5000E+02	+ C
0	0	.929	1.8000E+02	+ C
0	0	.929	2.1000E+02	+ C
1	.071	1.000	2.4000E+02	+***** C
0	0	1.000	2.7000E+02	+ C
0	0	1.000	3.0000E+02	+ C
0	0	1.000	INF	+ C
---				+ + + + + + + + + + +
14				0 20 40 60 80 100

HISTOGRAM OF THE FIRST ITERATION VALUES OF THE BET STA STATISTIC FOR TASK 3 (OBSVIDEO)

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
				+ + + + + + + + + + +

NO VALUES RECORDED

Figure 8(5). Mission Output

HISTOGRAM OF THE FIRST ITERATION VALUES OF THE BET STA STATISTIC FOR TASK 9 (OBSUNK)

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
0	0	0	0	+
0	0	0	3.0000E+01	+
2	1.000	1.000	6.0000E+01	+**********
0	0	1.000	9.0000E+01	+ C
0	0	1.000	1.2000E+02	+ C
0	0	1.000	1.5000E+02	+ C
0	0	1.000	1.8000E+02	+ C
0	0	1.000	2.1000E+02	+ C
0	0	1.000	2.4000E+02	+ C
0	0	1.000	2.7000E+02	+ C
0	0	1.000	3.0000E+02	+ C
0	0	1.000	INF	+ C
---				+ + + + + + + + + + +
2				0 20 40 60 80 100

HISTOGRAM OF THE FIRST ITERATION VALUES OF THE BET STA STATISTIC FOR TASK 21 (OBSFREND)

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
0	0	0	0	+
3	.600	.600	3.0000E+01	+**********
0	0	.600	6.0000E+01	+ C
1	.200	.800	9.0000E+01	+********** C
0	0	.800	1.2000E+02	+ C
1	.200	1.000	1.5000E+02	+********** C
0	0	1.000	1.8000E+02	+ C
0	0	1.000	2.1000E+02	+ C
0	0	1.000	2.4000E+02	+ C
0	0	1.000	2.7000E+02	+ C
0	0	1.000	3.0000E+02	+ C
0	0	1.000	INF	+ C
---				+ + + + + + + + + + +
5				0 20 40 60 80 100

Figure 8(6). Mission Output

HISTOGRAM OF THE FIRST ITERATION VALUES OF THE BET STA STATISTIC FOR TASK 25 (OBSHOST)

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
				+ + + + + + + + + + +
0	0	0	0	+
2	.286	.286	3.0000E+01	+**********
2	.286	.571	6.0000E+01	+**********C
1	.143	.714	9.0000E+01	+*******C
1	.143	.857	1.2000E+02	+*******C
1	.143	1.000	1.5000E+02	+*******C
0	0	1.000	1.8000E+02	+C
0	0	1.000	2.1000E+02	+C
0	0	1.000	2.4000E+02	+C
0	0	1.000	2.7000E+02	+C
0	0	1.000	3.0000E+02	+C
0	0	1.000	INF	+C
---				+ + + + + + + + + + +
7				0 20 40 60 80 100

HISTOGRAM OF THE FIRST ITERATION VALUES OF THE BET STA STATISTIC FOR TASK 28 (OBFU)

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
				+ + + + + + + + + + +
0	0	0	0	+
2	.222	.222	3.0000E+01	+**********
4	.444	.667	6.0000E+01	+**********C
2	.222	.889	9.0000E+01	+**********C
1	.111	1.000	1.2000E+02	+******C
0	0	1.000	1.5000E+02	+C
0	0	1.000	1.8000E+02	+C
0	0	1.000	2.1000E+02	+C
0	0	1.000	2.4000E+02	+C
0	0	1.000	2.7000E+02	+C
0	0	1.000	3.0000E+02	+C
0	0	1.000	INF	+C
---				+ + + + + + + + + + +
9				0 20 40 60 80 100

Figure 8(7). Mission Output

HISTOGRAM OF THE FIRST ITERATION VALUES OF THE INT STA STATISTIC FOR TASK 45 (RETHOOK)

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT
0	0	0	0
9	.818	.818	1.5000E+01
2	.182	1.000	3.0000E+01
0	0	1.000	4.5000E+01
0	0	1.000	6.0000E+01
0	0	1.000	7.5000E+01
0	0	1.000	9.0000E+01
0	0	1.000	1.0500E+02
0	0	1.000	1.2000E+02
0	0	1.000	1.3500E+02
0	0	1.000	1.5000E+02
0	0	1.000	INF
11			

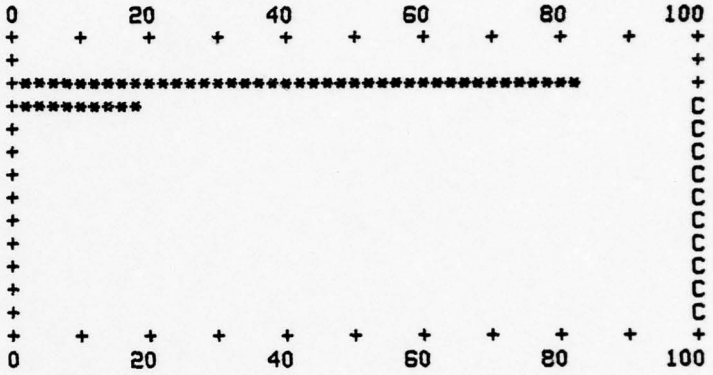


Figure 8(8). Mission Output

USER-GENERATED HISTOGRAM NUMBER 1

OPERATOR

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
0	0	0	0	+
42	.442	.442	1.0000E+00	+**********
15	.158	.600	2.0000E+00	+****C
0	0	.600	3.0000E+00	+C
3	.032	.632	4.0000E+00	+**C
6	.063	.695	5.0000E+00	+***C
8	.084	.779	6.0000E+00	+****C
10	.105	.884	7.0000E+00	+*****C
0	0	.884	8.0000E+00	+C
11	.116	1.000	9.0000E+00	+******C
0	0	1.000	1.0000E+01	+C
0	0	1.000	1.1000E+01	+C
0	0	1.000	1.2000E+01	+C
0	0	1.000	INF	+C
95				0 20 40 60 80 100

USER-GENERATED HISTOGRAM NUMBER 2

FUOPERAT

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
0	0	0	0	+
4	.500	.500	1.0000E+00	+**********
4	.500	1.000	2.0000E+00	+**********C
0	0	1.000	3.0000E+00	+C
0	0	1.000	4.0000E+00	+C
0	0	1.000	5.0000E+00	+C
0	0	1.000	6.0000E+00	+C
0	0	1.000	7.0000E+00	+C
0	0	1.000	8.0000E+00	+C
0	0	1.000	9.0000E+00	+C
0	0	1.000	1.0000E+01	+C
0	0	1.000	1.1000E+01	+C
0	0	1.000	1.2000E+01	+C
0	0	1.000	INF	+C
8				0 20 40 60 80 100

Figure 8(9). Mission Output

USER-GENERATED STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN	STD DEV	SD OF MEAN	CV	MINIMUM	MAXIMUM	OBS
SEARCHT	9.4106E+00	6.6326E+00	1.0234E+00	7.0480E-01	0	3.6149E+01	42
IDLET	5.2069E+00	2.1708E+00	5.6050E-01	4.1692E-01	2.7128E+00	8.8757E+00	15
VIDEOT			NO VALUES RECORDED				
UNKT	7.6395E+00	1.1464E+01	6.6187E+00	1.5006E+00	5.7577E-01	2.0867E+01	3
FRIENDT	9.7459E-01	3.4272E-01	1.3992E-01	3.5166E-01	6.4074E-01	1.4979E+00	6
HOSTILET	5.5773E+00	1.9279E+00	6.8160E-01	3.4566E-01	1.4100E+00	7.2836E+00	8
FIREUT	2.4584E+01	1.7280E+01	5.4645E+00	7.0289E-01	8.6150E-01	5.0952E+01	10
ASSIGNT			NO VALUES RECORDED				
HOOKINGT	1.2148E+01	5.9079E+00	1.7813E+00	4.8633E-01	6.6546E+00	2.5720E+01	11
TIMETRAK	3.7500E+01	1.7678E+01	1.2500E+01	4.7140E-01	2.5000E+01	5.0000E+01	2
TIMEFRND	1.0000E+01	0	0	0	1.0000E+01	1.0000E+01	2
TIMEHOST	5.1250E+01	7.7540E+01	2.7415E+01	1.5130E+00	1.0000E+01	2.0000E+02	8
KILLT	1.4788E+02	1.1955E+02	4.2268E+01	8.0845E-01	6.2057E+01	3.4345E+02	8

Figure 8(10). Mission Output

USER-GENERATED HISTOGRAM NUMBER 3

FUEFFECT

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
				+ + + + + + + + + + +
0	0	0	0	+
4	.500	.500	1.0000E+00	+**********
4	.500	1.000	2.0000E+00	+********** C
0	0	1.000	3.0000E+00	+ C
0	0	1.000	4.0000E+00	+ C
0	0	1.000	5.0000E+00	+ C
0	0	1.000	6.0000E+00	+ C
0	0	1.000	7.0000E+00	+ C
0	0	1.000	8.0000E+00	+ C
0	0	1.000	9.0000E+00	+ C
0	0	1.000	1.0000E+01	+ C
0	0	1.000	1.1000E+01	+ C
0	0	1.000	1.2000E+01	+ C
0	0	1.000	INF	+ C
				+ + + + + + + + + + +
8				0 20 40 60 80 100

Figure 8(11). Mission Output

USER-GENERATED STATSTICS FOR TIME-PERSISTENT VARIABLES

	MEAN	STD DEV	MINIMUM	MAXIMUM	TIME INTERVAL	CUR. VALUE
OBEFF	8.2010E-02	2.7438E-01	0	1.0000E+00	8.0000E+02	0
FU1	5.6999E-01	4.9508E-01	0	1.0000E+00	8.0000E+02	1.0000E+00
FU2	7.4232E-01	4.3736E-01	0	1.0000E+00	8.0000E+02	1.0000E+00
FU3			NO VALUES RECORDED			
FU4			NO VALUES RECORDED			
FU5			NO VALUES RECORDED			
FU6			NO VALUES RECORDED			
FU7			NO VALUES RECORDED			
FU8			NO VALUES RECORDED			
FU9			NO VALUES RECORDED			
FU10			NO VALUES RECORDED			

Figure 8(12). Mission Output

TIME	JOB	TASK NO							1111111111222222222233333 1234567890123456789012345678901234	1 1234567890
0 0	SER	1	TR- 0		D- 0	AFU- 0				UU
0 35.52	IDL	2	TR- 0		D- 0	AFU- 0			H	UF
0 41.18	SER	1	TR- 0		D- 0	AFU- 0			H	UF
0 57.21	OFU	28	FU- 1	U	P- 0	S- 0			RRH	UF
0 58.52	SER	1	TR- 0		D- 0	AFU- 0			RRH	UF
1 6.06	IDL	2	TR- 0		D- 0	AFU- 0			RRHR	UF
1 15.07	SER	1	TR- 0		D- 0	AFU- 0			UR H	AE
1 22.99	IDL	2	TR- 0		D- 0	AFU- 0			UR H	XE
1 23.57	SER	1	TR- 0		D- 0	AFU- 0			UR H	XE
1 34.67	OBU	9	TR- 1	U	D- 75	AFU- 0			UR H	FE
1 36.16	SER	1	TR- 0		D- 0	AFU- 0			UR H	FE
1 44.18	OFU	28	FU- 2	E	P- 3	S- 0			UU H	FE
1 45.02	OFU	33	FU- 2	E	P- 3	S- 0			UU H	FE
1 52.61	OFU*	33	FU- 2	E	P- 3	S- 0			UU H	FE
2 2.78	OFU	34	FU- 2	E	P- 3	S- 0			UU HR	FE
2 6.52	OFU	33	FU- 2	U	P- 3	S- 0			UU HR	FU
2 13.89	OFU*	33	FU- 2	A	P- 5	S- 0			UU H	EA
2 23.10	OFU	34	FU- 2	A	P- 5	S- 0			UU H	EA
2 25.51	OFU	33	FU- 1	U	P- 4	S- 0			UU H	UA
2 33.38	SER	1	TR- 0		D- 0	AFU- 0			UU H	UX
2 36.86	OBU	9	TR- 1	U	D- 69	AFU- 0			UU H	UX
2 38.35	SER	1	TR- 0		D- 0	AFU- 0			UU H	UX
2 46.01	OFU	28	FU- 2	F	P- 5	S- 0			UU H	UF
2 47.27	SER	1	TR- 0		D- 0	AFU- 0			UU H	UF
2 49.63	OBU	9	TR- 2	U	D- 68	AFU- 0			UU H	UF
2 50.52	OBU	10	TR- 2	U	D- 68	AFU- 0			UU H	UF
2 54.35	OBU*	10	TR- 2	U	D- 67	AFU- 0			UU H	UF
3 4.19	OBU	11	TR- 2	U	D- 66	AFU- 0			UU HR	UF
3 7.18	OBU	12	TR- 2	U	D- 66	AFU- 0			UU HR	UF
3 11.11	SER	1	TR- 0		D- 0	AFU- 0			UU HH	AF
3 14.68	OFU	28	FU- 1	A	P- 6	S- 0			UU H	AE
3 16.02	OFU	33	FU- 1	A	P- 6	S- 0			UU H	AE
3 21.97	OFU*	33	FU- 1	A	P- 6	S- 0			UH H	AE
3 32.07	OFU	34	FU- 1	X	P- 6	S- 0			UH H	XE
3 34.86	OFU	33	FU- 2	U	P- 5	S- 0			UH H	XU
3 40.33	SER	1	TR- 0		D- 0	AFU- 0			UH H	XU
3 46.71	OBU	9	TR- 1	U	D- 62	AFU- 0			UH H	FU
3 47.92	SER	1	TR- 0		D- 0	AFU- 0			UH H	FU
3 52.64	IDL	2	TR- 0		D- 0	AFU- 0			UH H	FU
4 2.15	SER	1	TR- 0		D- 0	AFU- 0			UH HR	FU
4 10.36	IDL	2	TR- 0		D- 0	AFU- 0			HH H	EA
4 14.79	SER	1	TR- 0		D- 0	AFU- 0			HH H	EA
4 23.02	OBH	25	TR- 2	H	D- 59	AFU- 0			HH H	EA
4 28.42	SER	1	TR- 0		D- 0	AFU- 0			HH H	EA
4 32.22	IDL	2	TR- 0		D- 0	AFU- 0			HH H	EX
4 37.14	SER	1	TR- 0		D- 0	AFU- 0			HH H	EX
4 41.20	OFU	28	FU- 1	E	P- 6	S- 0			HH H	EF
4 42.22	OFU	33	FU- 1	E	P- 6	S- 0			HH H	EF
4 46.81	OFU*	33	FU- 1	E	P- 6	S- 0			HH H	EF
4 56.99	OFU	34	FU- 1	E	P- 6	S- 0			HH H	EF
5 .53	OFU	33	FU- 1	U	P- 6	S- 0			HH HR	UF
5 8.07	SER	1	TR- 0		D- 0	AFU- 0			HH HR	UF
5 9.68	OBH	25	TR- 7	H	D- 25	AFU- 2			HH HR	UF
5 14.16	SER	1	TR- 0		D- 0	AFU- 0			HH HH	AF
5 17.47	OFU	28	FU- 2	E	P- 7	S--1			HH H	AE
5 18.51	SER	1	TR- 0		D- 0	AFU- 0			HH H	AE

Figure 8(13). Mission Output

T I M E	JOB	TASK NO						1111111111222222222233333 1234567890123456789012345678901234567890123	1234567890
5 22.27	OBH	25	TR- 1	H	D- 42	AFU- 2		HH H	AE
5 28.61	SER	1	TR- 0		D- 0	AFU- 0		HH H	AE
5 33.75	IDL	2	TR- 0		D- 0	AFU- 0		HH H	XE
5 43.53	SER	1	TR- 0		D- 0	AFU- 0		HH H	FE
5 48.23	IDL	2	TR- 0		D- 0	AFU- 0		HH H	FE
5 58.00	SER	1	TR- 0		D- 0	AFU- 0		HH H	FE
6 1.02	OBH	25	TR- 8	H	D- 25	AFU- 1		HH HR	FE
6 8.52	SER	1	TR- 0		D- 0	AFU- 0		HH HR	FE
6 13.37	OBH	25	TR- 2	H	D- 37	AFU- 1		HH HF	FE
6 19.11	SER	1	TR- 0		D- 0	AFU- 0		HH HF	FE
6 24.79	OFU	28	FU- 1	E	P- 8	S--2		HH F	EE
6 26.08	OFU	33	FU- 1	E	P- 8	S--2		HH F	EE
6 34.04	OFU*	33	FU- 1	E	P- 8	S--2		HH F	EE
6 43.32	OFU	34	FU- 1	E	P- 8	S--2		HH F	EE
6 46.76	OFU	33	FU- 1	U	P- 8	S--2		HH F	UE
6 54.27	OFU*	33	FU- 1	F	P- 2	S- 0		HH F	FE
7 4.36	OFU	34	FU- 1	F	P- 2	S- 0		HH FR	FE
7 7.90	OFU	33	FU- 2	U	P- 7	S--1		HH FR	FU
7 15.48	SER	1	TR- 0		D- 0	AFU- 0		HH FF	FF
7 20.00	OFU	28	FU- 1	F	P- 2	S- 0		HH FF	FF
7 21.02	OFU	33	FU- 1	F	P- 2	S- 0		HH FF	FF
7 26.89	SER	1	TR- 0		D- 0	AFU- 0		H FF	*F
7 34.05	IDL	2	TR- 0		D- 0	AFU- 0		H FF	*F
7 42.58	SER	1	TR- 0		D- 0	AFU- 0		H FF	*F
7 44.13	IDL	2	TR- 0		D- 0	AFU- 0		H FF	*F
7 52.07	SER	1	TR- 0		D- 0	AFU- 0		FF	**
7 54.93	OFU	28	FU- 2	*	P- 1	S- 0		FF	**
7 55.98	OFU*	28	FU- 2	Z	P- 1	S- 0		FF	*Z
8 3.64	OFU	29	FU- 2	Z	P- 1	S- 0		FF	*Z
8 8.04	OFU*	29	FU- 2	Z	P- 1	S- 0		FF	*Z
8 20.49	OFU	32	FU- 2	Z	P- 1	S- 0		FF	*Z
8 23.09	OFU	30	FU- 2	Z	P- 1	S- 0		FF	*Z
8 26.39	SER	1	TR- 0		D- 0	AFU- 0		FF	*
8 30.07	OBF	21	TR- 9	F	D- 15	AFU- 0		FF	*
8 30.59	SER	1	TR- 0		D- 0	AFU- 0		FF	*
8 39.53	OFU	28	FU- 1	*	P- 2	S- 0		FF	*
8 40.92	OFU*	28	FU- 1	Z	P- 2	S- 0		FF	Z
8 48.33	OFU	29	FU- 1	Z	P- 2	S- 0		FF	Z
8 55.95	OFU*	29	FU- 1	Z	P- 2	S- 0		FF	Z
9 6.99	OFU	32	FU- 1	Z	P- 2	S- 0		FF	Z
9 10.72	OFU	30	FU- 1	Z	P- 2	S- 0		FF	Z
9 14.25	SER	1	TR- 0		D- 0	AFU- 0		FF	
9 28.04	OBF	21	TR-10	F	D- 15	AFU- 0		FF	
9 28.64	SER	1	TR- 0		D- 0	AFU- 0		FF	
9 48.94	OBF	21	TR- 9	F	D- 24	AFU- 0		FF	
9 49.83	SER	1	TR- 0		D- 0	AFU- 0		FF	
10 1.57	OBF	21	TR- 9	F	D- 25	AFU- 0		FF	
10 2.23	SER	1	TR- 0		D- 0	AFU- 0		FF	
10 33.62	IDL	2	TR- 0		D- 0	AFU- 0		FF	
10 34.80	SER	1	TR- 0		D- 0	AFU- 0		FF	
10 46.21	OBF	21	TR-10	F	D- 23	AFU- 0		FF	
10 47.64	SER	1	TR- 0		D- 0	AFU- 0		FF	
10 56.84	OBF	21	TR- 9	F	D- 31	AFU- 0		FF	
10 57.39	SER	1	TR- 0		D- 0	AFU- 0		FF	
11 24.82	IDL	2	TR- 0		D- 0	AFU- 0		FF	
11 28.62	SER	1	TR- 0		D- 0	AFU- 0		FF	

Figure 8(14). Mission Output

T I M E	JOB	TASK NO					11111111112222222222223333 / 123456789012345678901234567890123	1 / 1234567890
11 53.50	IDL	2	TR- 0		D- 0	AFU- 0	FF	
11 59.11	SER	1	TR- 0		D- 0	AFU- 0	FF	
12 17.61	OBF	21	TR-10	F	D- 33	AFU- 0	FF	
12 18.76	SER	1	TR- 0		D- 0	AFU- 0	FF	
12 28.71	IDL	2	TR- 0		D- 0	AFU- 0	FF	
12 28.73	SER	1	TR- 0		D- 0	AFU- 0	FF	
12 49.17	OBF	21	TR-10	F	D- 37	AFU- 0	FF	
12 50.23	SER	1	TR- 0		D- 0	AFU- 0	FF	
12 56.60	OBF	21	TR- 9	F	D- 44	AFU- 0	FF	
12 57.27	SER	1	TR- 0		D- 0	AFU- 0	FF	

Figure 8(15). Mission Output

USER-GENERATED STATISTICS FOR VARIABLES BASED ON OBSERVATION

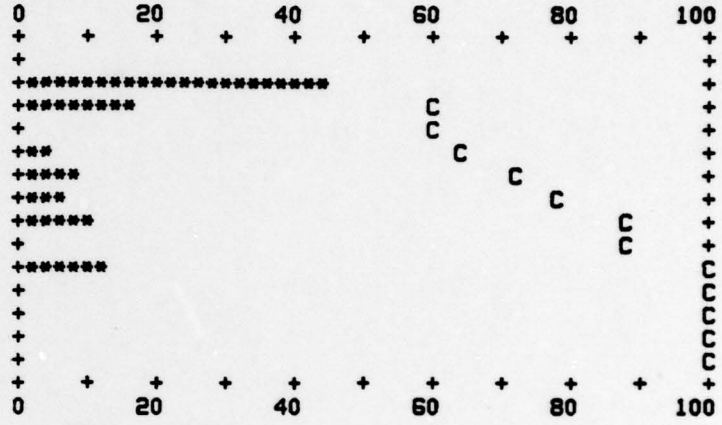
	MEAN	STD DEV	SD OF MEAN	CV	MINIMUM	MAXIMUM	OBS
SEARCHT	9.4118E+00	7.4524E+00	8.0833E-01	7.9181E-01	0	3.6149E+01	85
IDLET	5.4775E+00	2.8334E+00	5.2615E-01	5.1728E-01	1.6342E-02	9.7881E+00	29
VIDEOT			NO VALUES RECORDED				
UNKT	6.9423E+00	9.7308E+00	3.6779E+00	1.4017E+00	5.7577E-01	2.1486E+01	7
FRIENDT	8.9266E-01	3.2348E-01	8.3522E-02	3.6238E-01	5.1529E-01	1.4979E+00	15
HOSTILET	5.6994E+00	1.6167E+00	4.4840E-01	2.8367E-01	1.4100E+00	7.5040E+00	13
FIREUT	2.3746E+01	1.7874E+01	3.9968E+00	7.5273E-01	8.6150E-01	5.0952E+01	20
ASSIGNT			NO VALUES RECORDED				
HOOKINGT	1.0957E+01	4.3653E+00	9.3068E-01	3.9840E-01	6.6546E+00	2.5720E+01	22
TIMETRAK	3.7500E+01	1.4434E+01	7.2169E+00	3.8490E-01	2.5000E+01	5.0000E+01	4
TIMEFRND	1.0000E+01	0	0	0	1.0000E+01	1.0000E+01	4
TIMEHOST	5.1250E+01	7.4911E+01	1.8728E+01	1.4617E+00	1.0000E+01	2.0000E+02	16
KILLT	1.5157E+02	1.3403E+02	3.3508E+01	8.8428E-01	6.2057E+01	4.1719E+02	16

Figure 8(16). Mission Output

USER-GENERATED HISTOGRAM NUMBER 1

OPERATOR

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT
0	0	0	0
85	.445	.445	1.0000E+00
29	.152	.597	2.0000E+00
0	0	.597	3.0000E+00
7	.037	.634	4.0000E+00
15	.079	.712	5.0000E+00
13	.068	.780	6.0000E+00
20	.105	.885	7.0000E+00
0	0	.885	8.0000E+00
22	.115	1.000	9.0000E+00
0	0	1.000	1.0000E+01
0	0	1.000	1.1000E+01
0	0	1.000	1.2000E+01
0	0	1.000	INF
191			



USER-GENERATED HISTOGRAM NUMBER 2

FUOPERAT

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT
0	0	0	0
8	.500	.500	1.0000E+00
8	.500	1.000	2.0000E+00
0	0	1.000	3.0000E+00
0	0	1.000	4.0000E+00
0	0	1.000	5.0000E+00
0	0	1.000	6.0000E+00
0	0	1.000	7.0000E+00
0	0	1.000	8.0000E+00
0	0	1.000	9.0000E+00
0	0	1.000	1.0000E+01
0	0	1.000	1.1000E+01
0	0	1.000	1.2000E+01
0	0	1.000	INF
16			

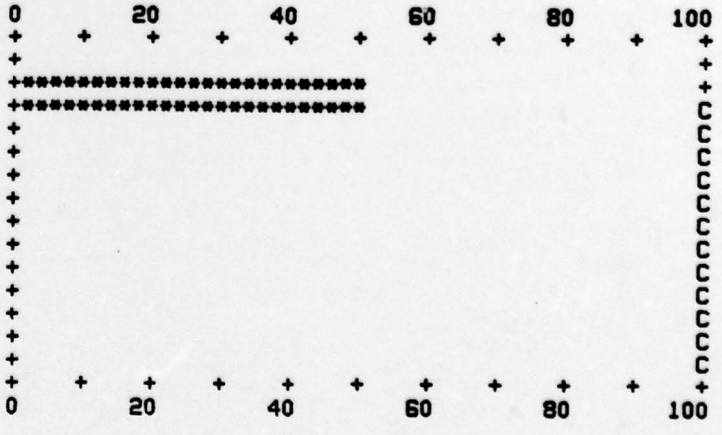


Figure 8(17). Mission Output

USER-GENERATED HISTOGRAM NUMBER 3

FUEFFECT

OBSV FREQ	RELA FREQ	CUML FREQ	UPPER CELL LIMIT	0 20 40 60 80 100
				+ + + + + + + + + + +
0	0	0	0	+
8	.500	.500	1.0000E+00	+**********
8	.500	1.000	2.0000E+00	+**********C
0	0	1.000	3.0000E+00	+C
0	0	1.000	4.0000E+00	+C
0	0	1.000	5.0000E+00	+C
0	0	1.000	6.0000E+00	+C
0	0	1.000	7.0000E+00	+C
0	0	1.000	8.0000E+00	+C
0	0	1.000	9.0000E+00	+C
0	0	1.000	1.0000E+01	+C
0	0	1.000	1.1000E+01	+C
0	0	1.000	1.2000E+01	+C
0	0	1.000	INF	+C
---				+ + + + + + + + + + +
16				0 20 40 60 80 100

Figure 8(18). Mission Output

USER-GENERATED STATSTICS FOR TIME-PERSISTENT VARIABLES

	MEAN	STD DEV	MINIMUM	MAXIMUM	TIME INTERVAL	CUR. VALUE
OBEFF	7.0833E-02	2.5655E-01	0	1.0000E+00	8.0000E+02	0
FU1	6.7716E-01	4.6756E-01	0	1.0000E+00	8.0000E+02	1.0000E+00
FU2	6.6086E-01	4.7342E-01	0	1.0000E+00	8.0000E+02	1.0000E+00
FU3			NO VALUES RECORDED			
FU4			NO VALUES RECORDED			
FU5			NO VALUES RECORDED			
FU6			NO VALUES RECORDED			
FU7			NO VALUES RECORDED			
FU8			NO VALUES RECORDED			
FU9			NO VALUES RECORDED			
FU10			NO VALUES RECORDED			

STATISTICS TASK SUMMARY REPORT

AVERAGES OF THE STATISTICS COLLECTED FOR 2 ITERATIONS

TASK NUMBER	TASK LABEL	STAT TYPE	COLCT POINT	STATISTICS ON THE AVERAGE VALUE PER ITERATION				
				AVERAGE	STD DEV	NO. ITER	MINIMUM	MAXIMUM
2	IDLETIME	BET	STA	5.4288E+01	8.1057E-01	2	5.3715E+01	5.4861E+01
3	OBSVIDEO	BET	STA			0		
9	OBSUNK	BET	STA	4.7815E+01	5.3720E+00	2	4.4016E+01	5.1613E+01
21	OBSFREND	BET	STA	4.3563E+01	1.4490E+01	2	3.3317E+01	5.3809E+01
25	OBSHOST	BET	STA	4.3252E+01	2.2153E+01	2	2.7587E+01	5.8917E+01
28	OBFU	BET	STA	5.3159E+01	2.5304E+00	2	5.1369E+01	5.4948E+01
45	RETHOOK	INT	STA	1.0957E+01	1.6843E+00	2	9.7662E+00	1.2148E+01

Figure 8(19). Mission Output

REFERENCES

1. DTM 9-1425-650-12; Operator's and Organizational Maintenance Manual: Overall System Description (Guided Missile Air Defense System AN/TSQ-73), 3 October 1977.

2. DEP TM 9-1430-652-10-3; Operator's Manual: Chapter 4 - Operating Procedures (Guided Missile Air Defense System AN/TSQ-73), 3 October 1977.

3. ST 44-196-73A; Operation and Maintenance Reference Handbook (AN/TSQ-73), January, 1976.

4. Wortman, D.B., S.D. Duket, R.L. Hann, G.P. Chubb, and D.J. Seifert, Simulation Using SAINT: A User-Oriented Instruction Manual, AMRL-TR-77-61, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

5. Wortman, D.B., S.D. Duket, D.J. Seifert, R.L. Hann, and G.P. Chubb, The SAINT User's Manual, AMRL-TR-77-62, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

6. Duket, S.D., D.B. Wortman, D.J. Seifert, R.L. Hann, and G.P. Chubb, Documentation for the SAINT Simulation Program, AMRL-TR-77-63, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.