

Research Note 84-24

AN ANALYTICAL MODEL OF LEARNING AND PERFORMANCE OF ARMOR PROCEDURES

Paul J. Sticha, Teresa D. Edwards, and John F. Patterson
Decisions and Designs, Inc.

Stephen L. Goldberg, Contracting Officer's Representative

Submitted by

Donald F. Haggard, Jr., Chief
ARI FIELD UNIT AT FORT KNOX, KENTUCKY

and

Harold F. O'Neil, Jr., Director
TRAINING RESEARCH LABORATORY



U. S. Army

Research Institute for the Behavioral and Social Sciences

January 1984

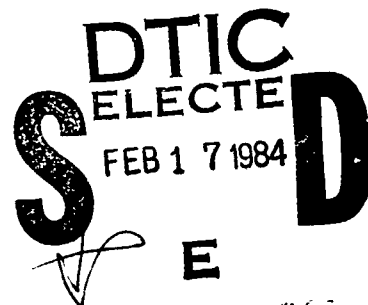
Approved for public release; distribution unlimited.

This report has been cleared for release to the Defense Technical Information Center (DTIC). It has been given no other primary distribution and will be available to requestors only through DTIC or other reference services such as the National Technical Information Service (NTIS). The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

84 02 16 05

AD A138000

DTIC FILE COPY



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Research Note 84-24	2. GOVT ACCESSION NO. ADA138 000	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Analytical Model of Learning and Performance of Armor Procedures		5. TYPE OF REPORT & PERIOD COVERED Interim Report
7. AUTHOR(s) Paul J. Sticha, Teresa D. Edwards, and John F. Patterson		6. PERFORMING ORG. REPORT NUMBER Technical Report DDI/PR 82-15-334
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decisions and Designs, Inc., Suite 600 Westpark Drive, P. O. Box 907 McLean, Virginia 22101		8. CONTRACT OR GRANT NUMBER(s) MDA 903-81-C-0517
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue Alexandria, Virginia 22333		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q26374A794
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE January 1984
		13. NUMBER OF PAGES 234
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Work was monitored technically by Dr. Stephen L. Goldberg, ARI Field Unit at Fort Knox, Kentucky		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Mathematical Models Learning Armor Tasks Military Training Skills Army Training SAINT (Systems Analysis of Integrated Networks of Tasks)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) → The rate at which performance improves during training; and the extent to which information is retained during intervals without practice, is a concern of those who plan and manage military training. This report documents the development and features of a model to investigate issues regarding acquisition and retention of complex, military skills. The modeling effort focused on procedural armor tasks. Learning and retention models were developed for eight tasks performed by the driver, gunner, and loader positions in the M60A1 tank. Sequencing control-		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. (continued)

was modeled using the SAINT (System Analysis of Integrated Networks of Tasks) simulation system. Psychological models described acquisition, retention, retrieval and choice of task information. Models were validated by comparing their predictions to two samples of data, one composed of soldiers in training, and one from soldiers in operational armor units.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SUMMARY

Decisions regarding the management of training can be made most effectively when they are supported by information about the effectiveness of different training options. However, the process of obtaining this information is generally expensive, and fraught with serious methodological and practical problems. Psychological theory, especially when it is expressed in mathematical terms, can have an important role in guiding the conduct and interpretation of research on the acquisition and retention of military skills. This report describes a project to develop and test models of learning and performance in a military setting. The modeling effort has focused on procedural tasks because: (1) these tasks represent the majority of military tasks, (2) there has been little previous research providing an integrative mathematical model of procedural learning and retention, and (3) many of the components of such a model have been developed.

Learning and retention models were developed for eight procedural tasks performed by the driver, gunner, and loader positions in an M60A1 tank. The model integrated theories regarding two aspects of skilled performance in these tasks: (1) control of the sequencing of task elements, and (2) learning and retention processes operating at the task-element level.

Sequencing control was modeled using the SAINT (System Analysis of Integrated Networks of Tasks) simulation system. A general SAINT model was developed for each of the procedural tasks which allows the modeler to simulate performance of several trials on the task. The models were developed through an iterative process involving information from a variety of sources, including behavioral analyses, the soldier's training

manual, the M60A1 technical manual, and discussions with subject-matter experts. The general versions of the models were modified to describe details of the two samples of data used to develop and validate the models.

Psychological models described acquisition, retention, retrieval, and choice. The overall approach to learning and retention is based on the concept of a strength of an association, which is assumed to be a random variable. The probability of correctly retrieving the association is the probability that the strength of the association exceeds a threshold. The acquisition model assumes that strength increases geometrically to an asymptote. The retention model postulates decrease of strength during intervals without practice according to a power function. Finally, the choice model assumes that the choices from a set of correct associations can be represented by assigning each alternative a random variable representing its utility.

The model was validated by comparing its predictions to two samples of data, one from a unit in training, and one from an operational unit. Soldiers in the training unit received five acquisition trials shortly after their formal training on a task, and one retention trial after a one-month delay. Each soldier in the training sample performed two of the eight tasks. Soldiers in the operational unit performed one repetition of each of the eight tasks.

The learning and retention models were able to predict general features of the data from the training sample quite well. These features include the overall go rate, the percentage of task elements performed correctly, and the performance time, each as a function of trial. For more specific features of the data, there were suggestions that a more complex model was

required. Specifically, differences between task elements in the value of learning and retention parameters were found, as were individual differences. It was suggested that these differences be the topic of further study.

The data from the operational unit gave no indication of either decay or improvement in performance as a function of experience in the unit. Thus, the retention models received no support from the data from the operational unit (in contrast to the results from the soldiers in training). These results are not consistent with any simple decay function, and it was suggested that they are caused by differential rates of practice among task elements.

The preliminary analysis described in this report shows considerable success in the effort to develop a model that describes learning and retention of procedural skills. However, there are needs for both further empirical validation and continued model development. The current data should be analyzed in greater detail to obtain a better understanding of task-element and individual differences. Other currently available data should be reanalyzed to expand the scope of the model and to understand task differences better. Finally, additional data should be collected to investigate retention processes in the operational unit.

ACKNOWLEDGMENTS

The authors would like to thank many individuals who helped conduct the research that led to this report. Project monitoring by Stephen L. Goldberg helped ensure efficient performance, especially in arranging for assistance from Army personnel and scheduling of the data-collection effort. C. Mazie Knerr, James H. Harris, and Bridgette K. O'Brien, of Human Resources Research Organization, were in charge of the effort to establish a data base for model development and testing, including data collection and task analyses. Their professional efforts are largely responsible for the high quality of the data. Robert Bachert, of the Air Force Aerospace Medical Research Laboratory, provided the SAINT software.

John Correnti, Margaret Bate, Ed Billheimer, and Phyllis Hutchison, from DDI's Art Department, are responsible for the high quality artwork contained in this report.

CONTENTS

	<u>Page</u>
SUMMARY	iii
ACKNOWLEDGEMENTS	vii
FIGURES	xiv
TABLES	xvi
1.0 INTRODUCTION	1
2.0 SAINT MODELING FRAMEWORK	6
2.1 SAINT Symbolism	6
2.1.1 Task input	7
2.1.2 Task description	7
2.1.3 Task output	7
2.2 The SAINT Network	11
2.2.1 Source task	11
2.2.2 Sink task	11
2.2.3 Nonspecific task	11
2.3 SAINT Attributes and User-Defined Task Characteristics	12
2.3.1 Resource attributes	12
2.3.2 Information attributes	12
2.3.3 System attribute	13
2.3.4 User-defined task characteristics	13
2.3.5 Assigning attribute values	13
2.4 SAINT Subprograms	13
2.4.1 SAINT user-written subprograms	14
2.4.2 SAINT user-support subprograms	15
2.5 SAINT Input Data	15
3.0 ARMOR SKILLS MODELS	18

CONTENTS (Continued)

	<u>Page</u>
3.1 A General Description of the Models	19
3.1.1 General model structure	20
3.1.2 Attribute definitions	22
3.1.3 User-defined functions	25
3.1.4 Common storage	27
3.2 Load on M240 Coax Machinegun (LOADMG)	27
3.2.1 Tasks	29
3.2.2 Precedence relationships	29
3.2.3 Attributes	29
3.2.4 Resource	32
3.2.5 Times	32
3.3 Start the M60A1 Tank Engine	32
3.3.1 Tasks	33
3.3.2 Precedence relationships	38
3.3.3 Attributes	39
3.3.4 Resource	39
3.3.5 Times	39
3.4 Stop the M60A1 Tank Engine (STOPTANK)	40
3.4.1 Tasks	40
3.4.2 Precedence relationships	43
3.4.3 Attributes	43
3.4.4 Resources	43
3.4.5 Times	43
3.5 Perform Gunner's Prepare-to-Fire Checks (GUNNERPF)	43
3.5.1 Tasks	44
3.5.2 Precedence relationships	52
3.5.3 Attributes	52
3.5.4 Resources	53
3.5.5 Times	53
3.6 Perform Loader's Prepare-to-Fire Checks (LOADERPF)	53

CONTENTS (Continued)

	<u>Page</u>
3.6.1 Tasks	54
3.6.2 Precedence relationships	54
3.6.3 Attributes	59
3.6.4 Resources	59
3.6.5 Times	59
3.7 Engage Targets Using Precision Fire Techniques (PRECFIRE)	59
3.7.1 Tasks	60
3.7.2 Precedence relationships	64
3.7.3 Attributes	65
3.7.4 User-defined functions	65
3.7.5 Resources	66
3.7.6 Times	66
3.8 Communicate Over Tactical FM Radio AN/VRC-64 (RADIOMSG)	66
3.8.1 Tasks	69
3.8.2 Precedence relationships	70
3.8.3 Attributes	70
3.8.4 Resources	70
3.8.5 Times	70
3.9 Communicate Using Visual Signalling Techniques (SIGNALS)	70
3.9.1 Tasks	71
3.9.2 Precedence relationships	77
3.9.3 Attributes	77
3.9.4 User-defined functions	77
3.9.5 Resources	79
3.9.6 Times	79
4.0 REPRESENTATION OF LEARNING AND PERFORMANCES PROCESSES	80
4.1 Acquisition	81
4.2 Retention	82
4.3 Retrieval	84

CONTENTS (Continued)

	<u>Page</u>
4.4 Performance Speed	85
4.5 Choice	85
4.6 Summary	86
5.0 MODEL REPRESENTATION OF PSYCHOLOGICAL PROCESSES	87
5.1 Acquisition	87
5.2 Retention	87
5.3 Retrieval	90
5.4 Performance Speed	92
5.5 Choice	92
6.0 PARAMETER ESTIMATION AND MODEL TESTING	95
6.1 Method	95
6.1.1 Operational unit data collection	96
6.1.2 OSUT data collection	97
6.1.3 Differences between the samples	97
6.2 Overview of Analysis	99
6.3 Results for OSUT Data	101
6.3.1 Hierarchy of models	101
6.3.2 Models of accuracy	108
6.3.3 Models of speed and accuracy	115
6.3.4 Application of models to cross-validation sample	117
6.3.5 Analysis of the combined sample	122
6.4 Results for Operational Unit Data	122
6.4.1 Hierarchy of models	125
6.4.2 Models of accuracy	125
6.4.3 Models of speed and accuracy	127
6.4.4 Analysis of the combined sample	127

CONTENTS (Continued)

	<u>Page</u>
6.5 Summary	130
7.0 EXPERIMENTAL SIMULATIONS	132
7.1 OSUT Models	132
7.1.1 Implementing the models	134
7.1.2 Model output	136
7.2 Operational Unit Models	144
7.2.1 The OSUT framework	145
7.2.2 Implementing the framework	147
8.0 DISCUSSION	150
8.1 Procedural Control	150
8.1.1 Task analysis	151
8.1.2 The SAINT representation	152
8.2 Results from OSUT Data	152
8.3 Results from the Operational Unit	152
8.4 Future Prospects	154
8.4.1 Analysis of current data	154
8.4.2 Needs for model development	155
8.4.3 Use of the model	156
REFERENCES	157
APPENDIX A	A-1
APPENDIX B	B-1
APPENDIX C	C-1

FIGURES

<u>Figure</u>		<u>Page</u>
2-1	SAINT TASK SYMBOL	8
2-2	SAINT TASK SYMBOLS FOR AVAILABLE BRANCHING TYPES	9
3-1	FRAMEWORK FOR GENERAL SAINT MODELS	21
3-2	SAINT NETWORK FOR LOADMG	30
3-3	SAINT NETWORK FOR STARTANK	34
3-4	SAINT NETWORK FOR STOPTANK	41
3-5	SAINT NETWORK FOR GUNNERPF	45
3-6	SAINT NETWORK FOR LOADERPF	55
3-7	SAINT NETWORK FOR PRECFIRE	61
3-8	SAINT NETWORK FOR RADIOMSG	67
3-9	SAINT NETWORK FOR SIGNALS	72
5-1	LISTING OF SUBROUTINE LEARN	88
5-2	LISTING OF SUBROUTINE RETEN	89
5-3	LISTING OF MODRF 2	91
5-4	LISTING OF FUNCTION PRIOR	93
6-1	HIERARCHY OF LEARNING MODELS FOR OSUT SAMPLE	102
7-1	FRAMEWORK FOR OSUT SAMPLE SIMULATION	133
7-2	SAINT INPUT DATA FOR OSUT MODEL FRAMEWORK	135
7-3	SIMULATED AND ACTUAL WHOLE-TASK GO RATES	137
7-4	SIMULATED AND ACTUAL TASK-ELEMENT GO RATES	140
7-5	SIMULATED AND ACTUAL PERFORMANCE TIMES	142

FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
7-6	FRAMEWORK FOR OPERATIONAL UNIT MODELS	146
7-7	SAINT INPUT DATA FOR OPERATIONAL UNIT MODEL FRAMEWORK	148

TABLES

<u>Table</u>		<u>Page</u>
2-1	SAINT TASK DESCRIPTION CODES	10
2-2	LIST OF SAINT INPUT DATA TYPES	16
3-1	INPUT TO REDIMEN PROGRAM	28
3-2	PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR LOADMG	31
3-3	PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR STARTANK	36
3-4	PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR STOPTANK	42
3-5	PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR GUNNERPF	48
3-6	PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR LOADERPF	57
3-7	PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR PRECFIRE	63
3-8	PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR RADIOMSG	68
3-9	PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR SIGNALS	75
6-1	DESCRIPTION OF LEARNING MODELS FOR OSUT SAMPLE	103
6-2	NUMBER OF PARAMETERS REQUIRED FOR DIFFERENT VERSIONS OF LEARNING MODELS	109
6-3	PARAMETER ESTIMATES FOR OSUT MODEL OF PERFORMANCE ACCURACY (LEARN 4) FOR MODEL DEVELOPMENT GROUP	111
6-4	TWICE NEGATIVE LOG LIKELIHOOD OF DATA FROM MODEL DEVELOPMENT GROUP - OSUT SAMPLE	113
6-5	TWICE NEGATIVE LOG LIKELIHOOD OF DATA FROM MODEL DEVELOPMENT GROUP - OSUT SAMPLE	116

TABLES (Continued)

<u>Table</u>		<u>Page</u>
6-6	INITIAL AND FINAL BY LIKELIHOOD VALUES FOR CROSS-VALIDATION TESTS	119
6-7	IMPROVEMENT FROM BASELINE TO INITIAL PARAMETER VALUES AS A PERCENTAGE OF IMPROVEMENT TO OPTIMAL PARAMETERS	121
6-8	PARAMETER ESTIMATES FOR OSUT MODEL OF PERFORMANCE ACCURACY (LEARN 4) FOR ALL SUBJECTS	123
6-9	PARAMETER ESTIMATES FOR OSUT MODEL OF PERFORMANCE SPEED AND ACCURACY (LEARN 10) FOR ALL SUBJECTS	124
6-10	DESCRIPTION OF RETENTION MODELS FOR OPERATIONAL UNIT SAMPLE	126
6-11	TESTS OF RETENTION MODELS FOR MODEL DEVELOPMENT GROUP	128
6-12	CORRELATION BETWEEN PERFORMANCE TIME AND SINCE OSUT GRADUATION IN OPERATION UNIT SAMPLE	129
6-13	TESTS OF RETENTION MODELS FOR ALL SUBJECTS IN OPERATIONAL UNIT	131

AN ANALYTIC MODEL OF LEARNING AND PERFORMANCE OF ARMOR PROCEDURES

1.0 INTRODUCTION

Decisions regarding the management of training can be made most effectively when they are supported by information about the effectiveness of different training options. Such information includes the rate at which performance improves as a result of training and practice, the rate at which performance deteriorates during periods without practice, and the amount of practice required to maintain an acceptable level of performance. That is, the information required relates to the acquisition and retention of military skills.

The process of obtaining this information is generally expensive and fraught with serious methodological and practical difficulties. Field studies are difficult to design and carry out, and require a great commitment of resources. In addition, it is often impossible to include proper experimental control in empirical studies, so that the results often cannot be interpreted unambiguously. Consequently, the results of empirical studies often have limited applicability, and cannot be generalized without considerable risk beyond populations having characteristics similar to the uncontrolled characteristics in the sample from which data were obtained.

Psychological theory can have an important role in guiding the conduct and interpretation of research in the environment described above. Proper theory increases the extent to which empirical results may be extrapolated to new situations. In addition, theory may allow the researcher to remove the effects of experimental artifacts to produce clearer

interpretation of empirical results. When a theory has organized a large body of facts from different situations, then predictions for new situations will no longer involve extrapolation from limited data, but will be based on the combined knowledge derived from a large number of situations. Finally, good theory may lead to insights about the way individuals learn to perform Army tasks, and hence ways in which the training system can be improved.

Theories are often most useful when they can be formulated in mathematical terms. Mathematical models offer a precision that is generally not possible with theories that are stated informally. In addition, the process of formulating a mathematical model forces a coherency on the researcher which may not be present in theories that are developed to lesser detail. For these reasons, there is a substantial need for mathematical models describing learning and retention of Army tasks.

Human learning and performance has been a major topic for psychological research throughout the history of psychological investigation of behavior. More recently, mathematical models have been applied to a variety of tasks, including memorization of verbal material (Greeno, 1980), visual perception (Krantz, 1974), choice and decision making (Einhorn and Hogarth, 1981), problem solving (Newell and Simon, 1972), and manual control (Kleinman, Baron, and Levison, 1970). There has been a relative dearth, however, of models directed to the tasks that are most common in many settings, including most military tasks. These tasks, termed procedural tasks following Aagard and Braby (1976), are most easily described as a list of steps. Since there may be several orders in which the steps of a particular task may be performed, the steps are only partially ordered.

Recent reviews of the modeling literature (Pew, Baron, Feeher, and Miller, 1977; Pew and Baron, 1982; Sticha, 1982) have recognized the lack of mathematical models of learning and retention of procedural tasks, but have indicated that many of the components of such models already exist. In particular, Sticha (1982) has reviewed models especially relevant to procedural learning and retention, and has identified some of the most promising candidate models.

The objective of the research described in this report is to develop and validate a model of procedural learning and performance. A general modeling strategy was developed and applied to eight tasks taught in Armor One Station Unit Training (OSUT). These tasks were chosen to vary in their length, complexity, and the amount of practice they received in an operational unit. Models were developed for the following tasks:

1. Load an M240 Coax Machinegun (LOADMG)
2. Start the M60A1 Tank Engine (STARTANK)
3. Stop the M60A1 Tank Engine (STOPTANK)
4. Perform Gunner's Prepare-to-Fire Checks (GUNNERPF)
5. Perform Loader's Prepare-to-Fire Checks (LOADERPF)
6. Engage Targets using Precision Fire Techniques (PRECFIRE)
7. Communicate over Tactical FM Radio AN/VCR-64 (RADIOMSG)
8. Communicate using Visual Signaling Techniques: Ground Guiding (SIGNALS)

The general modeling strategy must consider two aspects of procedural learning and performance. First, the model must represent the structures and processes that control the order in which individual task elements are performed. Sticha (1982) has identified two general frameworks for representing procedural control, network simulation models and production systems.

The models developed for this project represent procedural control by a network using the SAINT (System Analysis of Integrated Networks of Tasks) network simulation language (Wortman, Duket, Seifert, Hann, and Chubb, 1978a,b). General information about the SAINT models is given in section 2.0, and a specific description of the model for each task is given in section 3.0.

The second concern of a learning and performance model is the representation of psychological processes involved in learning, retention, retrieval, and choice. The models developed in this project are based on the strength models developed by Hull (1943, 1952) and extended by Wickelgren (1974a,b) and others. New extensions were made of these models to increase their correspondence to the military environment. The psychological learning and performance models are described in section 4.0. The mechanisms used to represent them within the SAINT framework are described in section 5.0.

The validity of the models was tested by comparing their predictions to two sets of data collected from soldiers in OSUT, and operational units, respectively. This report presents an assessment of the goodness-of-fit of the models; a subsequent report will describe more detailed analyses using the models. Model testing is a process in which best-fitting parameters are estimated for a series of models. The goodness-of-fit of more and less general models are compared to determine whether the more general models are required to account for the data. The processes of parameter estimation and model testing are presented in section 6.0.

After the parameters were estimated for the learning and retention models, their values were incorporated into the SAINT framework. The resulting SAINT models allow for simulation of more complex situations than those represented in the two

data samples. The behavior of the SAINT models is described in section 7.0.

The model validation effort described in this report is preliminary. However, the results described in section 6.0 and 7.0 suggest more detailed analyses be carried out on the current data, as well as tests which require additional data. Section 8.0 gives a general discussion of the results of the model development and testing effort, paying particular attention to further tests suggested by current results.

2.0 SAINT MODELING FRAMEWORK

SAINT is a network-oriented simulation technique that has been used to model human behavior in situations ranging from standard laboratory tasks (Hann & Kuperman, 1975) to tasks involving the control of complex systems (Wortman and Hixon, 1978). A SAINT model represents the individual task elements that constitute a complex procedure, as well as the interactions between them. The intent of this section is to explain the basic modeling framework of SAINT, with emphasis on the characteristics most frequently used in the project. An in-depth discussion of the symbolism and terminology required for modeling with SAINT can be found in Wortman et al. (1978a,b). The overall structure and individual FORTRAN subprograms of SAINT are described in Duket, Wortman, Seifert, Hann, and Chubb (1978b). The use of statistical analysis packages to analyze SAINT output is discussed by Duket, Wortman, Seifert, and Chubb (1978a).

2.1 SAINT Symbolism

The basic element of a SAINT network is a task. The task in a SAINT model corresponds to a single task element of a procedural task, although the correspondence need not be exact. There are four events that change the state of SAINT tasks: release, start, completion, and clearing. When required predecessors for a particular task are completed, the task is released. Performance will begin on the task if the resources required for the task are not being used to perform another task. When a task is started, resources are assigned to it and the time required by the task is determined. Task completion refers to the successful accomplishment of a task. Finally, task clearing terminates the performance of a task.

before its completion.

The symbol used to represent a SAINT task (Figure 2-1) is slightly modified from the notation used by Wortman et al. (1978b). Each task symbol has three components: (1) input side, (2) task description, and (3) output side.

2.1.1 Task input - The number of predecessor completions required for the release of a task is shown on the input side of the task symbol. The number of predecessor completions required for the first release of the task is at the top (PR1), and the number of predecessor completions required for subsequent releases of the task is at the bottom (PR2).

2.1.2 Task description - The center portion of the task symbol is reserved for all task description information. Information such as task name, time characteristics, statistics to be collected, and attribute assignments are specified here. Rows contain distinct pieces of information. Six rows are provided for descriptive information (the original notation provides four) and more rows can be added if necessary. For each row, a four letter SAINT code on the left-hand side identifies the type of information that is given on the right-hand side. In all there are 17 SAINT task description codes (Table 2-1). More detailed explanations for each code can be found in Wortman et al. (1978a,b).

2.1.3 Task output - Each task is uniquely identified by a task number (TSK) shown on the output side of the task symbol. The shape of the output side indicates the type of branching to be performed upon completion. Four types of branching are available in SAINT language (Figure 2-2). In the first type, deterministic branching, all tasks attached to

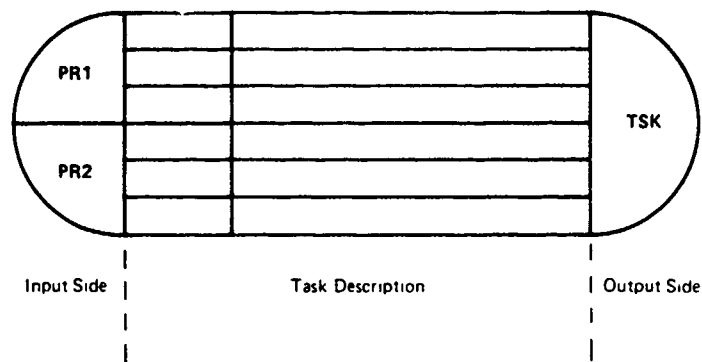


Figure 2-1
SAINT TASK SYMBOL

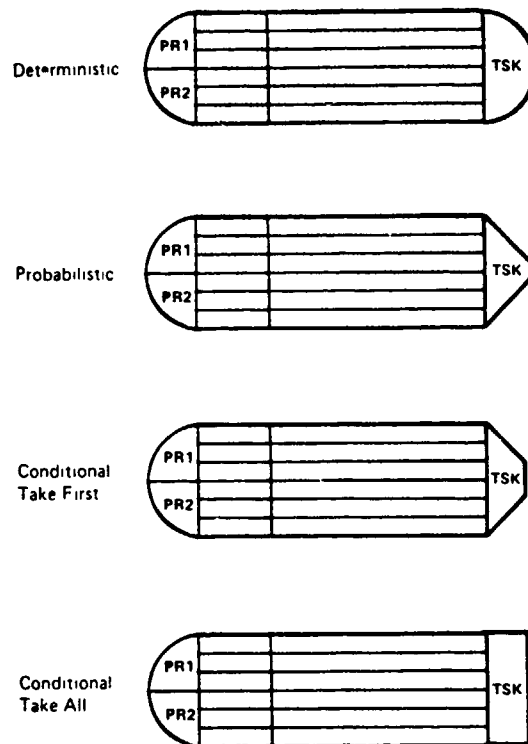


Figure 2-2
SAINT TASK SYMBOLS FOR AVAILABLE BRANCHING TYPES

TABLE 2-1. SAINT TASK DESCRIPTION CODES¹

<u>Description Code</u>	<u>Information Required by Description Code</u>
ATAS	attribute assignments to be performed
DIFF	different predecessor option
DMOD	distribution set modifications to be made
INCM	information choice mechanism decision mode
LABL	label associated with task
MARK	marking information
MODF	moderator functions to be applied to task performance
PREC	completion precedence
PRTY	priority
RCLR	resource clearing to be performed
REGL	regulation to be performed
RESR	resource requirements
STAT	statistics to be collected
SWIT	switching to be performed
TCLR	task clearing to be performed
TIME	performance time characteristics
UTCH	user-defined task characteristics

¹Wortman et al. (1978b).

a given task element are activated upon the completion of the first task. In the second type of branching, probabilistic branching, a single task is chosen from a set of tasks according to a specified probability distribution. The remaining two types of branching cause tasks to be released according to the state of a specified condition. In one type of conditional branching, all tasks satisfying the condition are released; in the other type, only the first task satisfying the condition is released.

2.2 The SAINT Network

A SAINT model consists of a collection of SAINT tasks organized into a network by predecessor relationships. There are three types of tasks in a SAINT model: (1) source tasks, (2) sink tasks, and (3) nonspecific tasks.

2.2.1 Source tasks - Source tasks (SO) are represented with a zigzag arrow pointing to the input side of the task symbol. These tasks occur at the beginning of the simulation and require no predecessor completions for their first release (i.e., PR1 equals 0).

2.2.2 Sink tasks - A sink task (SI) does not connect to another task. The number of sink task completions required for completion of the simulation must be specified by the user. The simulation will end once the number of sink tasks specified for completion has been reached.

2.2.3 Nonspecific tasks - Any task that is neither a source task nor a sink task is referred to as a nonspecific task (NS). These tasks always have at least one predecessor task, and almost always have at least one successor task. A nonspecific task that has no successor is similar to a sink

task except that its completions are not counted in determining when to end the simulation.

2.3 SAINT Attributes and User-Defined Task Characteristics.

SAINT provides a capability for including four types of descriptive information in a model: (1) resource attributes, (2) information attributes, (3) system attributes, and (4) task characteristics.

2.3.1 Resource attributes - A resource is any non-consumable commodity required for the performance of one or more tasks (e.g. operators and equipment). Resource attributes uniquely describe the associated resource. Depending on the type of resource, resource attributes can represent a variety of characteristics. For example, the soldier performing a procedure is a resource for the models developed in this project; associated resource attributes might include the soldier's duty position, aptitude, training, or other information.

2.3.2 Information attributes - Information attributes characterize items of information that flow through the network, and give an indication of which tasks have been performed. Information attributes are organized into packets that are passed through the networks from task to task. At any time in the simulation there may be several information packets traversing the network with different values for the information attributes. In several of our models, information attributes are used as signaling devices. When certain tasks are completed, the value of an information attribute is changed. Then, when a branching choice depends on whether or not that task has been completed, the value of the associated information attribute is checked.

2.3.3 System attribute - Attributes that are not directly applicable to an information- or resource-oriented characterization are called system attributes. For example, system attributes may represent the position of switches on the equipment, or other general information about a procedure, such as the type of ammunition being used in an engagement. The chief difference between system attributes and information attributes is that system attributes have a single value throughout the SAINT network, while information attributes may have different values at different tasks in the network.

2.3.4 User-defined task characteristics - User-defined task characteristics describe information specific to individual tasks in a SAINT network. For example, different tasks in a SAINT network may be learned to a greater or lesser extent. A user-defined task characteristic would be used to represent the degree to which a task is learned.

2.3.5 Assigning attribute values - The number of resource, information, and system attributes included in a model must be specified by the user. Values can be assigned to resource, information and system attributes, and user-defined task characteristics at the start of the simulation; the values can also be assigned or changed at the release, start, or completion of any task in the SAINT model. Attributes may be assigned constant values, values sampled from probability distributions, or values calculated through user-written subroutines .

2.4 SAINT Subprograms

The SAINT computer program was designed in a modular form so that additions, modifications, and deletions could be easily made. This section discusses user-written subprograms and user-support subprograms.

2.4.1 SAINT user-written subprograms - User-written subprograms are those FORTRAN subprograms used to define additional model components, such as summary reports, initializations, and alterations to time. Five of the nine user-written subprograms were used in the models developed for this project. The general functions of these subprograms are described below.

Two subprograms are principally concerned with input and output. These subroutines are named UINPT and UOTPT. UINPT is called at the beginning of a simulation immediately after the processing of all input data. The function of UINPT is to allow for input of additional data, such as model parameters, which cannot be specified in SAINT input data. UOTPT is called at the end of all required iterations of the simulation. This subroutine may be used to produce user-generated summary reports and to print information which is not in the standard SAINT output.

The subroutines MODRF and PRIOR, and the function USERF affect the processing of tasks during the simulation. Many of the psychological models of learning and retention are implemented in the SAINT models using these functions. Subroutine PRIOR is used to modify the priorities of SAINT tasks during a simulation. It is called for all released tasks before they are started. Subroutine MODRF is used to modify the way a task is performed. For example, MODRF may be used to adjust the time required for completion of the task, or its branching probabilities. MODRF is used in the armor skill models to represent learning and retention. It is called at the start of a task for which it is active. Function USERF is principally used to make attribute assignments that cannot be made directly, using standard SAINT options. It is called when the attribute assignment is made.

Of the remaining user-written subprograms, INTLC and STATE are concerned with state variables, which are used to represent continuously varying quantities; UERR is used to create user-generated error reports, and ENDIT is used to create summary reports at the end of a single iteration. These four subprograms were not used in the models of armor skill learning and retention.

2.4.2 SAINT user-support subprograms - SAINT's user-support subprograms allow the user to access frequently required SAINT variables and to collect statistics on any user-defined variables. User-support subprograms relating to: (1) attribute values, (2) user-defined task characteristics, (3) task scheduling, (4) random number generators, and (5) user-generated statistics were used in this project. For descriptions of the subprograms in these categories and their argument definitions refer to Wortman et al. (1978a,b).

2.5 SAINT Input Data

In order to implement a particular network model on the SAINT system, the SAINT network diagram must be translated into the SAINT input language. There are several types of SAINT input data statements corresponding to the task-descriptive and network-control information represented in a SAINT MODEL. These include: (1) general information, (2) user-generated statistics and plots, (3) initial status and labels, (4) task description, (5) task branching, and (6) effects of task completions. Table 2-2 presents a list of the SAINT input data types. A complete description of SAINT input procedures is found in Wortman et al. (1978a).

TABLE 2-2. LIST OF SAINT INPUT DATA TYPES²

General Information

GEN	<u>General</u> Information
SGE	<u>State</u> variable <u>general</u> information
POP	<u>Program</u> <u>options</u>
OUT	<u>Output</u> options
DIS	<u>Distribution</u> set definitions
SIM	Used when multiple networks are to be <u>simulated</u> to mark the end of input for each network except the last
FIN	Signals end (<u>finish</u>) of all SAINT input cards

User-Generated Statistics and Plots

UBO	<u>User</u> -generated statistics for variables <u>based</u> on <u>observation</u>
UTI	<u>User</u> -generated statistics for <u>time</u> -persistent variables
UHI	<u>User</u> -generated <u>histograms</u>
UPL	<u>User</u> -generated <u>plot</u> /table characteristics
UVA	<u>User</u> -generated plot/table <u>variable</u> information

Initial Status and Labels

IMO	<u>Initial</u> <u>moderator</u> function status specification
IRA	<u>Initial</u> <u>resource</u> <u>attribute</u> specification
ISA	<u>Initial</u> <u>system</u> <u>attribute</u> specification
LRE	<u>Labels</u> for <u>resources</u>
LSV	<u>Labels</u> for <u>state</u> <u>variables</u>

Task Description

TAS	<u>Task</u> characteristics
STA	<u>Statistics</u> task information

²ibid

UTC Specification of user-defined task characteristics

MOD Specification of moderator function status

ATA Specification of atttribute assignment information

Task Branching

DET Deterministic branching

PRO Probabilistic branching

CFI Conditional-take first branching

CAL Conditional-take all branching

Effects of Task Completions

NMO Network modifications

DMO Distribution set modifications

TCL Task clearing

RCL Resource clearing

SWI Switching information

REG Regulation information

State Variable Monitors

MON Monitor characteristics

MTA Monitor actuated task signaling

MSW Monitor actuated switching

State Variable Statistics and Plots

SST State variable statistics information

PLO State variable plot/table characteristics

VAR State variable plot/table variable information

3.0 ARMOR SKILLS MODELS

The SAINT models provide a framework for representing the network of task elements for the eight procedural tasks. In addition, the SAINT models provide mechanisms for representing psychological processes involved in learning and retention of these tasks, and for representing specific training schedules corresponding to real or hypothetical training situations. Three different versions of the SAINT models were developed for each task. The first version was designed to represent a general situation in which the tasks were trained over several trials; the number of trials and the spacing between trials could be determined by the user. The other two versions of the models were designed to simulate the training activities of the two groups of soldiers from which data were collected to develop and test the models. Except for differences corresponding to specific data-collection procedures, the tasks themselves are represented the same way in each of the three versions. However, each version of the model contains additional tasks representing original training, duty position assignment, in-unit practice, and so forth, of the study samples. Since the representation of task performance is the same for the three versions, only the general version will be discussed in this section. The results of specific simulations are presented in section 7.0.

In describing the SAINT models, we must distinguish several meanings of the word "task" as it relates to military procedures and model structures. The eight procedures studied in this project are referred to as tasks; components of these tasks are called task-elements, steps, or subtasks. On the other hand, the basic building block of a SAINT model is the task. A task in this context refers to a single operation, or step, in the procedure. Thus, SAINT tasks corresponds to the task

elements of the procedures. Where it is necessary to avoid ambiguity, the tasks within the SAINT models will be referred to as model tasks or SAINT tasks. The eight tasks studied in this project will be termed procedural tasks or procedures.

A general description of the process by which the models were developed and of some of the features common to the models is given in section 3.1. In sections 3.2 through 3.9, each SAINT model will be described, and the SAINT tasks will be defined. For ease of exposition, and to avoid redundancy, the SAINT networks presented in these sections will include only those task elements describing a single repetition of the procedural task. Other task elements, which apply to all tasks, are described in section 3.1.

3.1 A General Description of the Models

A SAINT simulation model was developed for each of the armor procedures. Each model was based on information from a variety of sources, including behavioral analyses, the soldier's training manual, the M60A1 technical manual, and discussion with subject-matter experts. In order to understand the procedures better, a flow chart was developed linking the individual task elements for each task. The flow chart was an interim representation of the procedures from which the SAINT networks were created. Thus, the flow charts were often incomplete, or had different levels of detail for different segments of the procedure. When each procedure was understood, a SAINT network was developed for it. The model was then translated to SAINT input data, and the behavior of the model was examined to determine that the representation was accurate. Finally, psychological models of learning, retention, and choice were incorporated into the SAINT performance models. The models described

in this section, therefore, are the result of several iterations of modeling and validation activities.

3.1.1 General model structure - The SAINT performance models represent several practice trials for a given task in each iteration of the simulation. The general structure for representing these training trials is presented in Figure 3-1. The base tasks of a performance model lie within the dotted lines; two special purpose tasks, Task 50, ENDTRIAL, and Task 51, ENDSIM, handle the processing required at the end of each trial and at the end of each iteration, respectively. Their purpose in these models is to allow for statistics to be gathered over a specified number of trials.

Examination of the simple SAINT model in Figure 3-1 gives an introduction to SAINT nomenclature. The simulation begins with the performance of the SAINT tasks representing the procedure. When the procedure has been performed, control is transferred to Task 50, ENDTRIAL. Task 50 requires a constant time of 60 seconds to perform. The value of this time may be varied to examine effects of spacing of training. Task 50 requires no resources, so it will be started as soon as it is released. At this time, moderator function 3 will be called. This function collects end of trial statistics and resets some of the counters used on the next trial. System attribute 3 is the trial counter. At the completion of Task 50, the counter is incremented by user function 5. Then, if the last trial (stored in system attribute 4) has not occurred, the task is performed again. If the last trial has been completed, control is transferred to Task 51, END SIM. Task 51 requires no time and no resources, and does not perform any activities; it merely serves to end the iteration of the simulation.

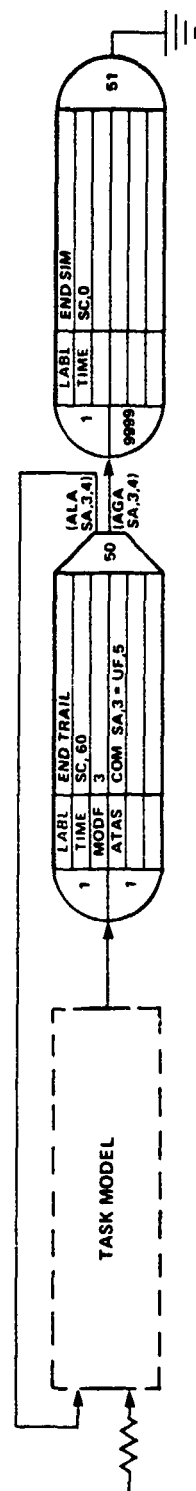


Figure 3-1
FRAMEWORK FOR GENERAL SAINT MODELS

3.1.2 Attribute definitions - One of the goals of this project is to develop a general way to represent learning and retention of Armor procedures. Consequently, learning and performance processes are represented the same way in each of the SAINT models. In addition, we have tried to represent commonly used attributes and functions in a consistent manner to provide savings in the development process. This section defines the roles played by system, information, resource attributes, and user-defined task characteristics. The following section describes the functions performed by user-written subprograms.

System attributes (SA). The first four system attributes have a common definition across all models, and will be described in this section. The remaining attributes represent task-specific information about the state of the system, and are described along with the model in which they are used.

SA(1) is used as a counter for several procedures, and is incremented by USERF 1. A counter is required by GUNNERPF, LOADERPF, RADIOMSG, and SIGNALS.

SA(2) is used to record whether any errors were made in a particular trial for the entire procedure. A value of 1 indicates a "go" for the task, and a value of 0 indicates a "no go." The value of SA(2) is set in MODRF 2; the final value for the trial is used to update the statistical counter NUMGO; and the value of SA(2) is reinitialized in MODRF 3. SA(2) is used by all models.

SA(3) contains the trial number. It is incremented by USERF 5 which is called by Task 50, ENDTRIAL. SA(3) is used by all models.

SA(4) contains the value of the limit on the number of trials (NUMTR), and is set by UINPT at the beginning

of a simulation run. SA(3) and SA(4) are compared at the end of each trial to determine if another learning trial should be simulated. All models use SA(4).

Information attributes (IA). Information attributes are used by some of the models to control branching between tasks. There is no common function served by information attributes for all tasks in the general simulation models.

User-defined task characteristics (UTC). User task characteristics contain information specific to a task in the SAINT network. Within the models of armor procedures, UTCs are used to represent three of the variables in the psychological models, namely, learning strength, priority, and trace fragility. Six user task characteristics are incorporated in the models.

UTC(1) contains an index into the common arrays associated with task elements. Not all SAINT tasks are subject to the learning and retention models discussed later in this report. Some of them are at a greater level of detail than is represented in the learning and retention models. Others represent activities performed by the scorer or equipment, or required for network synthesis. The index designates a SAINT task to which the learning and retention models apply with an entry other than zero; the value of UTC (1) is used to retrieve model parameters and store task-element performance data. A zero entry for UTC(1) indicates that the SAINT task does not correspond to a performance measure identified by a behavioral analysis; the learning and retention models are not applied to those SAINT tasks. UTC(1) is used by all of the models.

UTC(2) contains the mean of the priority distribution for the task. In the subroutine PRIOR, a random deviate

is added to UTC(2) to determine the priority the task. UTC(2) is used by all models.

UTC(3) contains the mean trace strength for a task element. Learning increments are applied by LEARN which is called by MODRF 2. Retention is determined by RETEN which, in turn, is called by PRIOR and MODRF 2. UTC(3) is defined for all task elements, but is only relevant for those elements in which UTC(1) is greater than zero. UTC(3) is used by all models.

UTC(4) contains the trace fragility for a task element. It is adjusted by RETEN which is called by PRIOR and MODRF 2. UTC(4) is defined for all task elements, but is relevant only for those elements in which UTC(1) is greater than zero. It is used in all models.

UTC(5) contains the momentary trace strength which is the sum of the mean trace strength and a random deviate. It is calculated and used in PRIOR to generate the priorities, and used again in MODRF 2 to determine whether the task element is performed correctly. Its relevance is the same as UTC(3) and UTC(4).

UTC(6) contains the time of the most recent strength update. It is reset whenever strength is updated by RETEN. All models use UTC(6).

Resource attributes (RA). Resource attributes are not used by the general version of the models described in this section. They are used extensively in one of the specific models to represent the experience of soldiers in an operational unit. In these models, resource attributes are used to represent the duty position of a soldier as the simulation progresses, the type of training (i.e., whether he was in a gunner/loader or

driver track) the soldier received, the duty position of the soldier at the time of the experiment, the time at which the soldier received his current duty position, and the time since the soldier received his initial training.

3.1.3 User-defined subprograms - Several of the user-defined subprograms, including UINPT and UOTPT, are used by each of the task models. In addition, some of the moderator functions and user functions are used in all models. The functions performed by these subprograms are discussed in this section, with a listing of the subprograms provided in Appendix A. Although they are used in all tasks, subroutines LEARN and PETEN, and function PRIOR, are not described in this section because they directly implement the psychological processes involved in learning, retention, and choice. Those functions will be described in section 5.0.

UINPT. UINPT reads the values of model parameters and other values required for the simulations. The values are stored in arrays in two labeled common blocks, SITVR, which contains variables characterizing the experimental situation, and PARMS, which contains model parameters. This method of organizing data input allows the user easily to specify changes in model parameters because they are all together at the end of the SAINT input data.

UOTPT. UOTPT produces plots of performance by trial for the entire task, and may be used to produce plots of performance for each task element if desired by the user. This function operates in conjunction with two of the moderator functions, MODRF 2 and MODRF 3. The moderator functions collect statistics on performance at each trial. UOTPT, then, prints summaries of the statistics at the end of the simulation. In the version of the programs presented in Appendix A, the statements that produce histograms of individual-trial performance

for each task element have been made inoperative by changing them into comment statements. If the user wishes to print these histograms, the calls to subroutine UHIST in MODRF 2, MODRF 3, and UOTPT should be restored, and these subroutines should be recompiled.

Moderator functions (MODRF). Three of the ten moderator functions, MODRF 2, MODRF 3, and MODRF 4, are used by all of the general models and will be described in this section. MODRF 1, MODRF 5, and MODRF 6 are used by specific task models, and the remaining moderator functions are used by the specific versions of the models.

MODRF 2 is the vehicle by which performance is linked to the learning and retention models. This subroutine is called by all SAINT tasks for which learning and retention models apply. MODRF 2 has four major functions: (1) it determines whether the task is performed correctly, depending on its strength; (2) it sets the performance time for the model task, depending on how well it is learned; (3) it records performance measures for the task; and (4) it adjusts the strength according to the learning and retention models. Details about how MODRF 2 works are presented in section 5.0

MODRF 3 is called at the end of each learning trial (Task 50) to collect statistics and to reset the trial counter.

MODRF 4 resets the value of UTC(5), the momentary task strength, to the value, -11. This action is required so that task priorities will be set properly.

User-written functions (USERF). Two of the eight user-written functions, USERF 1 and USERF 5, are used by all models, and will be described in this section.

USERF 1 provides the general ability to maintain the counter in SA(1). This function adds one to the current value of SA(1).

USERF 5 increments the trial number stored in SA(3) at the end of each learning trial.

3.1.4 Common storage. Much of the communication within the SAINT system is handled through labeled common storage. To accomodate for the large number of user task characteristics and plots used in the models, and to take advantage of the fact that the models do not use state variables, it was necessary to redimension the arrays in common storage. This task was performed with the aid of REDIMEN (Seifert, Koeplinger, and Hoyland, 1980), a program for redimensioning the common variables in the SAINT programs. REDIMEN was run with the input listed in Table 3-1. It should be noted that it was necessary to change some of the output formats of the REDIMEN program in order to accomodate the large number of plots made available in the models.

3.2 Load an M240 Coax Machinegun (LOADMG)

This model describes a soldier's ability to load an M240 coaxial machinegun. This procedure is performed in twelve steps in which the machinegun is first cleared, and then loaded. Clearing the machinegun involves pulling the charger handle back to lock the bolt, placing the safety on S, raising the cover and the feedtray, examining the empty chamber while holding the charger handle, and lowering the feedtray. Loading the machinegun involves placing the ammunition in the feedtray, closing the cover, placing the safety in F, and announcing "up" when the machinegun is loaded.

TABLE 3-1. INPUT TO REDIMEN PROGRAM

VALUE	VARIABLE	COMMENT
100	ID	
100	IMN	
100	IMNA	REDUCED FROM 250
100	MAXDS	REDUCED FROM 1000
3	MDAD	REDUCED FROM 300
6	MDDR	REDUCED FROM 600
4	MDNPT	REDUCED FROM 40
3	MDNSS	REDUCED FROM 60
50	MDOAT	REDUCED FROM 100
2	MDQP	REDUCED FROM 1100
800	MDSTR	
2	MEQT	REDUCED FROM 100
2	MFLAG	REDUCED FROM 20
20	MMDFN	
300	MMSTU	
1000	MNCEL	INCREASED FROM 540
1350	MNCLS	
200	MNCLT	INCREASED FROM 20
2	MNCUP	REDUCED FROM 11000
200	MNHIS	INCREASED FROM 20
600	MNOPA	
10	MNPLT	
20	MNSTP	
2	MNSWA	REDUCED FROM 20
10	MNVAR	
2	MNVPP	REDUCED FROM 10
10	MOPNO	REDUCED FROM 20
100	MPARM	
2	MPLOT	REDUCED FROM 10
2	MSTAT	REDUCED FROM 20
100	MSYAT	
400	MTCHR	INCREASED FROM 200
50	MSXTA	
850	MYABA	

The network of task elements is shown in Figure 3-2; the SAINT input data implementing the network are shown in Appendix B. The SAINT tasks may be compared with the performance measures that resulted from the behavioral analysis shown in Appendix C. This comparison is shown explicitly in Table 3-2. The following subsections describe the task elements, precedence relationships, attributes, and resource and time requirements for LOADMG.

3.2.1 Tasks - The SAINT simulation model for LOADMG consists of 13 base tasks (that is, there are 13 tasks other than Task 50 and Task 51). All of these tasks correspond to performance measures (PM) as shown in Table 3-2. With one exception, there is a one to one correspondence between tasks and PMs. The fifth PM, "Pulls charger handle back and to side of gun, then holds handle against cover," has two tasks associated with it. They are Task 5, HNDLBACK (for pull handle back and to side of gun), and Task 6, HOLDHNDL (for holds handle against cover).

3.2.2 Precedence relationships - As indicated in Figure 3-2, the task network is essentially linear, with one exception. Task 5, HNDLBACK, is followed by two tasks. Task 7, EXAMINE, performs the examination of the chamber while the charger handle is being held. When the examination is completed, the handle is released. That is, upon the completion of Task 7, Task 6 is cleared. Each task completion is followed by deterministic branching, signifying no choice in the task sequencing.

3.2.3 Attributes - LOADMG requires one system attribute in addition to the four that are used for all tasks. This attribute, SA(5), represents the position of the safety. A value of 0 indicates that the safety is in "safe," while a value of 1 indicates that the safety is in "fire." Initially the safety is placed in "fire."

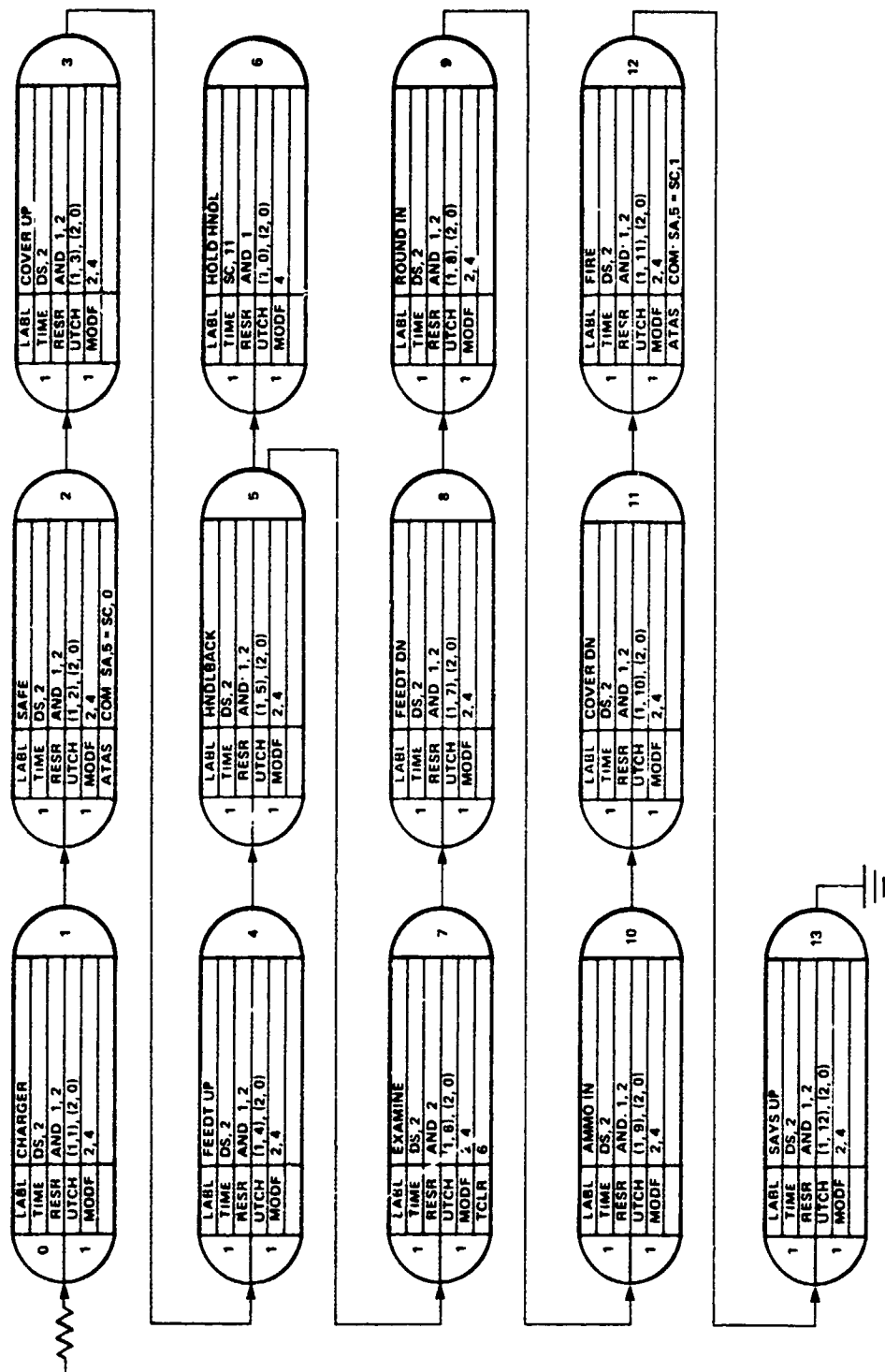


Figure 3-2
SAINT NETWORK FOR LOADMG

TABLE 3-2. PERFORMANCE MEASURES CORRESPONDING TO
SAINT TASKS FOR LOADMG

TASK	ASSOCIATED PERFORMANCE MEASURE
1 CHARGER	PM 1a: Pulls charger handle rearward to lock bolt back.
2 SAFE	PM 1b: Places safety on S.
3 COVER UP	PM 1c: Raises cover.
4 FEEDT UP	PM 1d: Lifts feedtray.
5 HNDLBACK	PM 1e (first part): Pulls charger handle back and to side of gun.
6 HOLDHNDL	PM 1e (second part): Holds handle against cover.
7 EXAMINE	PM 1f: Looks and feels empty chamber.
8 FEEDT DN	PM 1g: Lowers feedtray.
9 ROUND IN	PM 2a: Places first round in feedtray with open side of belt face down.
10 AMMO IN	PM 2b: Pushes ammunition in feedtray until it comes in contact with cartridge stops.
11 COVER DN	PM 2c: Closes cover.
12 FIRE	PM 2d: Places safety in F.
13 SAYS UP	PM 2e: Announces "UP" when machinegun is loaded.

3.2.4 Resources - The two resources involved are the soldiers two hands labeled FIRST and SECOND. The soldier uses both hands (AND:1,2) until he "holds handle against cover" and "looks and feels empty chamber." These PMs correspond to Task 6 HOLDHNDL and Task 7, EXAMINE, respectively. Task 6 requires resource FIRST (AND:1); Task 7 requires resource SECOND (AND:2). Just as the PM "lowers feedtray" follows the PM "looks and feels empty chamber," Task 8, FEEDTDN (for lowers feedtray), follows Task 7, EXAMINE. Task 8, FEEDTDN requires both resources, so it must wait until Task 7 is completed and Task 6 is cleared. All remaining task require both resources.

3.2.5 Times - The time required for Task 6, HOLDHNDL, is set so that this task cannot be completed before Task 7. For this preliminary version in which parameters have notional values, the time for Task 6 was set at a constant 11 seconds (SC,11). The performance time for all other tasks is a random variable sampled from an exponential distribution (DS,2), with mean 1 second, and maximum value 10 seconds.

3.3 Start the M60A1 Tank Engine (STARTANK)

The soldier in this exercise is the driver of an M60A1 tank. His ability to start the tank engine is being described by the model. Starting the M60A1 tank involves thirteen steps roughly organized into three major subtasks. In the first subtask, the soldier ensures that the equipment is in the proper condition for performing the task by checking the parking brake, drain valves, fuel shutoff valve, fuel pump switch, and other electronic equipment. The generator switch may also be checked at this time. In the second subtask, the soldier turns on the master battery switch, checks the fuel level, depresses the accelerator, and presses the starter switch until the engine starts (or waits 15 seconds, whichever comes first). In the third subtask, the soldier asks a crew member to check the

generator blower (the task continues with other checks which were not considered in the models).

The SAINT network of task elements is shown in Figure 3-3, and the SAINT input data are shown in Appendix B. Table 3-3 compares the tasks in the SAINT model with those in the behavioral analysis shown in Appendix C.

3.3.1 Tasks - Although the behavioral analysis of this task identified only thirteen performance measures, the SAINT simulation model for STARTANK consists of twenty-nine base tasks. Most of the factors which complicate the SAINT models, of the eight procedures, are present in this model.

Task 1, BEGIN, is included for network synthesis. The main function of the task is to set up the proper initial conditions at the beginning of each trial. The task also ensures that the six checks which begin the procedure are released at the same time at the beginning of each trial.

The first three performance measures, which set the parking brake, are represented by Task 2 through Task 6 in the SAINT model. Task 2 serves as an organizing task for the two components of these PMs. Tasks 3 and 4 ensure that the transmission is placed in Park (PM 2), by checking the transmission (Task 3) to see if it is already in park, and putting it in park (Task 4) if it is not already there. We let Task 3 represent this PM, since TASK 4 is not always carried out. Task 5, PRESSBRAK, presses the brake pedal (PM 1), and Task 6, RELBRAK, releases the brake when the transmission is in park.

Task 8 through Task 12 are related to closing the drain valves (PM 4). Since this PM is scored only once, Task 8 is used for scoring; the other tasks are included as additional

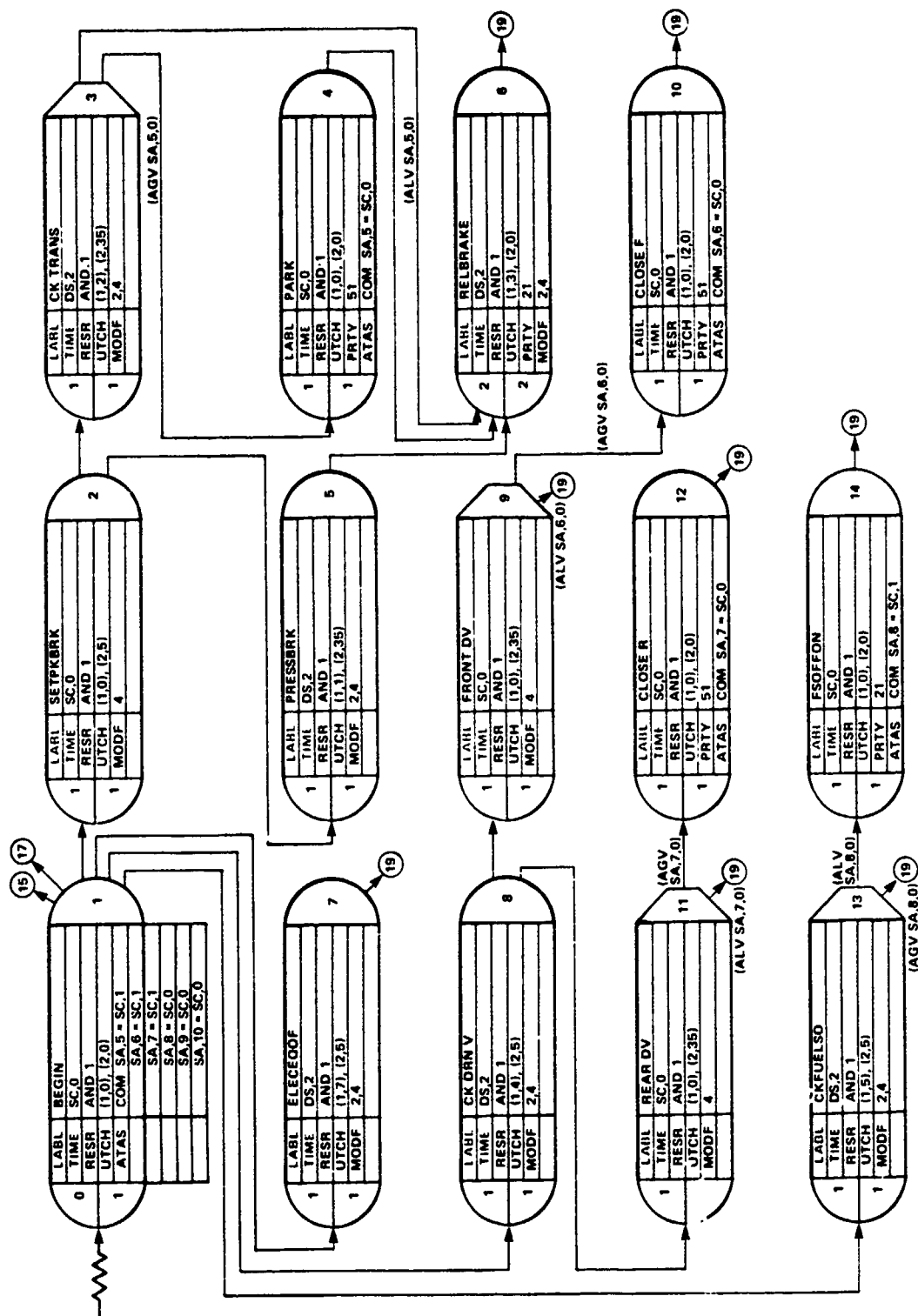


Figure 3-3
SAINT NETWORK FOR STARTANK

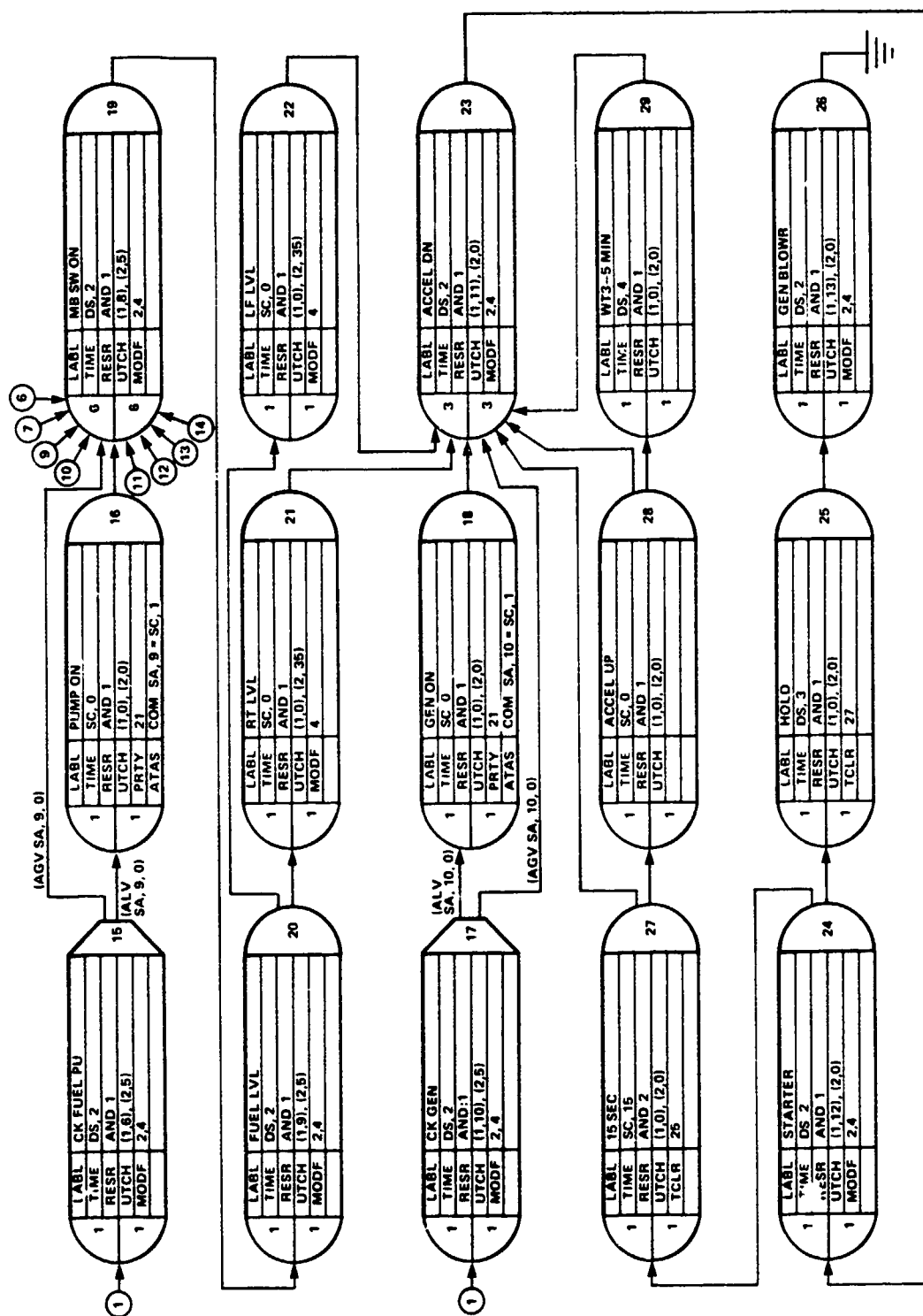


Figure 3-3
SAINT NETWORK FOR STARTANK
(Continued)

TABLE 3-3. PERFORMANCE MEASURES CORRESPONDING
TO SAINT TASKS FOR STARTANK

TASK	ASSOCIATED PERFORMANCE MEASURE
1 BEGIN	Used for network synthesis
2 SETPKBRK	Used for network synthesis
3 CK TRANS	PM 2: Places transmission in park.
4 PARK	Component of PM 2.
5 PRESSBRK	PM 1: Sets parking brake by pushing brake pedal.
6 RELBRAKE	PM 3: Releases brake pedal
7 ELECEQOF	PM 7: Asks crew if their electronic equipment is off.
8 CK DRN V	PM 4: Closes both drain valves.
9 FRONT DV	Component of PM 4.
10 CLOSE F	Component of PM 4.
11 REAR DV	Component of PM 4.
12 CLOSE R	Component of PM 4.
13 CKFUEL SO	PM 5: Places fuel shut-off valve handle in the ON position.
14 FSOFFON	Component of PM 5.
15 CKFUELPU	PM 6: Places fuel pump switch in ON position.
16 PUMP ON	Component of PM 6.
17 CK GEN	PM 10: Turns generator switch ON.
18 GEN ON	Component of PM 10.
19 MB SW ON	PM 8: Turns mastery battery switch on.

TABLE 3-3. PERFORMANCE MEASURES CORRESPONDING
TO SAINT TASKS FOR START-UP (Continued)

TASKS	ASSOCIATED PERFORMANCE MEASURES
20 FUEL LVL	PM 9: Check fuel level in both tanks.
21 RT LVL	Component of PM 9 (PM 9b)
22 LF LVL	Component of PM 9 (PM 9a)
23 ACCEL DN	PM 11: Depress accelerator pedal
24 STARTER	PM 12: Press starter switch until engine starts.
25 HOLD	Component of PM 12.
26 GENBLOWR	PM 13: Asks crew member to check generator blower.
27 15 SEC	Component of PM 12.
28 ACCEL UP	Additional observable task.
29 WT3-5MIN	Additional observable task.

observable tasks. Task 7, ELECEQOF, corresponds to PM 7, "ask crew if electronic equipment is off."

Placing the fuel shut-off valve handle in the ON position (PM 5) involves first checking its current position (Task 13), and putting it in the ON position (Task 14) if it is not already in the ON position. Here, we let TASK 13 correspond to PM 5, and let TASK 14 be an additional observable task, since it is not always carried out. Similarly, we let Task 15 correspond to PM 6, place fuel pump in the ON position, and let Task 16 be an additional observable task. Again, we let Task 17, CK GEN, correspond to PM 10, turn the generator switch ON, and let TASK 18 be an additional observable task.

When preliminary checks have been performed, the soldier turns on the master battery switch (PM 8). TASK 19 represents this PM. The check of the fuel levels (PM 9) is represented by Tasks 20, 21, and 22. Performance on this task is measured in Task 20.

Task 23, ACCEL DN, corresponds to PM 11; Task 24, STARTER, corresponds to PM 12; and Task 26, GENBLOWER, corresponds to PM 13. PM 12 involves pressing the starter switch until the engine starts (or for 15 seconds--whichever comes first). Task 27, 15 SEC, is a 15 second timer that is started when the starter is pressed. If the timer finishes before the tank starts, that is, if Task 27 is completed before Task 25, Task 25 is cleared, and the procedures for when the tank does not start are followed (Task 28 and Task 29). Otherwise, Task 27 is cleared, the tank is started, and the final PM, "check the generator blower," is performed (Task 26).

3.3.2 Precedence relationships - The precedence relationships reflect the intended flow of the network. Notice that when a switch is to be turned to a certain position, the network

assumes that the soldier first checks to see what the current position is. A conditional-take first branching specification is used to signify the possibility of the switch already being in the required position and the changing action not being carried out (see Tasks 3, 9, 11, 13, 15, and 17).

A second aspect about the branching for STARTANK is the fact that several tasks are released simultaneously at several points in the network. Note especially, the six branches at Task 1. These branches represent six checks on the position of switches. Five of these checks must be performed before the master battery switch is turned on, and the sixth must be performed before the accelerator is pressed. The order in which the tasks are performed is determined by function PRIOR, and depends on the value of UTC(2). Priorities of tasks following the first task of a branch are set so that an entire branch will be completed before another one is started. For example, Task 14 has a priority of 21, which ensures that it will follow Task 13, rather than any of the other tasks that were released at the completion of Task 1.

3.3.3 Attributes - STARTANK requires six system attributes to represent the positions of the six controls checked in Tasks 3, 9, 11, 13, 15, and 17. SA(5) through SA(10) represent the transmission (0 = Park), the front and rear drain (0 = off), the fuel pump switch (0 = off), and the generator switch (0 = off), respectively.

3.3.4 Resources - The two resources are the TIMER and the DRIVER. Resource 1, DRIVER, is required by all tasks except Task 27. Resource 2, TIMER, is required only by Task 27, 15 SEC.

3.3.5 Times - In general, tasks that correspond to performance measures require a time sampled from the second distribution set, as was the case in LOADMG. The times specified

at Tasks 25 and 27 are important, since the completion of either task clears the other. The time to perform Task 27 is a constant of 15 seconds, and the time to perform Task 25 is a random sample from a normal distribution. For Task 29, the time required is uniformly distributed between three and five minutes.

3.4 Stop the M60A1 Tank Engine (STOPTANK)

The task described by this model involves stopping the engine of an M60A1 tank. The soldier begins the procedure by setting the parking brake. Then, he idles the engine at 1000-1200 rpm for 5 minutes and at 700-750 rpm for 3 minutes. After checking to ensure that all electronic equipment is off, the soldier holds the engine fuel shutoff switch "up" until the engine stops. Then, he turns the master battery switch off.

The SAINT network for STOPTANK is shown in Figure 3-4, and the SAINT input data are shown in Appendix B. Table 3-4 compares the tasks in the SAINT model with those in the behavioral analysis described in Appendix C. The following subsections describe the task elements, precedence relationships, attributes, resources, and times for STOPTANK.

3.4.1 Tasks - The SAINT simulation model for STOPTANK has twelve base tasks. Most of the task elements correspond directly to individual performance measures. Both Tasks 2 and 3 are related to PM 2, "places transmission in PARK." The model assumes that this performance measure involves checking the transmission (Task 2, CK TRANS) first to see if it is already in park; and moving it into park (Task 3, PARK) if it is not already there. We let Task 2 represent PM 2, since Task 3 is not always carried out, and we treat Task 3 as an additional observable task. The procedure of setting the parking brake is the same as it is in STARTANK.

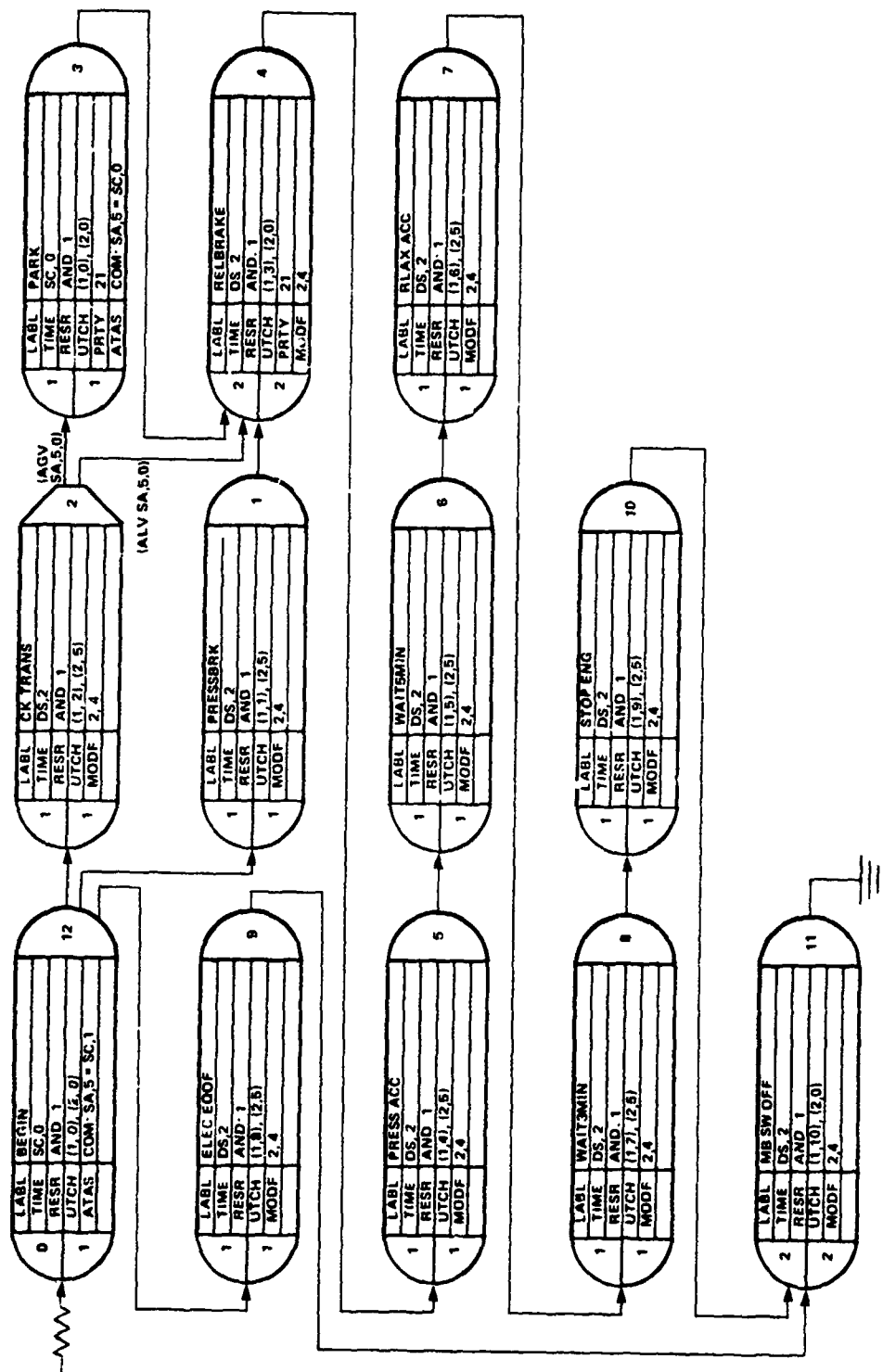


Figure 3-4
SAINT NETWORK FOR STOPTANK

TABLE 3-4. PERFORMANCE MEASURES CORRESPONDING
TO SAINT TASK FOR STOPTANK

TASK	ASSOCIATED PERFORMANCE MEASURE
1 PRESSBRK	PM 1: Sets parking brake by pushing brake pedal.
2 CK TRANS	PM 2: Places transmission in PARK.
3 PARK	Component of PM 2.
4 RELBRAKE	PM 3: Releases brake pedal.
5 PRESSACC	PM 4: Presses accelerator so that engine idles at 1000-1200 rpm.
6 WAIT5MIN	PM 5: Soldier says engine idles at 1000-1200 rpm for 5 minutes.
7 RLAX ACC	PM 6: Releases accelerator and idles engine at 700-750 rpm.
8 WAIT3MIN	PM 7: Soldier says engine idles at 750 rpm for 3 minutes.
9 ELECEQOF	PM 8: Ask Gunner and TC if their equipment is off.
10 STOP ENG	PM 9: Holds engine fuel shut-off switch in "SHUT-OFF" (Up) until engine stops.
11 MBSW OFF	PM 10: Turns master battery OFF, after engine stops.
12 BEGIN	Used for network synthesis.

3.4.2 Precedence relationships - The completion of Task 2 is followed by conditional-take first branching to either Task 3 or Task 4. This allows for the possibility that the soldier may bypass placing the transmission in park, if he sees that it is already in PARK. Each of the other task completions is followed by deterministic branching, signifying no alternative sequence. Task 9, ELECEQOF, represents checking to see that all electrical equipment is turned off, and may be performed at any time before the master battery switch is turned off (Task 11).

3.4.3 Attributes - STOPTANV requires one system attribute in addition to the four used by each task. SA(5) represents the state of the transmission, and has the value of 0 if the transmission is in Park, and 1 if the transmission is at any other position.

3.4.4 Resources - The only resource is the DRIVER; all of the tasks require that resource.

3.4.5 Times - All tasks that correspond to performance measures require a time sampled from the second distribution set; that is, time is exponentially distributed. This is true of the task elements that involve waiting, as well, because in the tests the soldier was merely asked how long he should wait, instead of actually waiting for that period of time. Tasks 3 and 12 do not require any time for performance.

3.5 Perform Gunner's Prepare-to-Fire Checks (GUNNERPF)

This model describes the process that a gunner of a M60A1 tank uses to check the gun controls in performing prepare-to-fire checks. GUNNERPF is a long task with thirty-four performance measures. The task is divided into three components: (1) place the turret into power operation; (2) check the azimuth

indicator for accuracy; and (3) check the azimuth indicator for slippage. Placing the turret into power operation involves holding the power solenoid plunger until zero pressure is indicated on the pressure gauge, checking the hydraulic power pack oil level, unlocking the turret traverse lock, announcing "power," turning the ELEV/TRAV power switch on, and rotating the gunner's power controls to verify that the procedure has been successful. In the accuracy check, the soldier aligns the cross on the gunner's daylight periscope on a target, sets the azimuth indicator to zero, traverses the turret manually through a complete circle until the cross is on the same aiming point, and checks to determine if the azimuth indicator is set to zero. In the slippage checks, a target is identified, and the soldier uses the gunner's control handles to traverse rapidly to the right (left) and stops suddenly while traversing. He then returns manually to the aiming point and checks the azimuth indicator.

The SAINT network for this model is shown in Figure 3-5; SAINT input data derived from this network is given in Appendix B. Table 3-5 shows the relationship between the tasks of the SAINT model and the performance measures determined through behavioral analysis shown in Appendix C.

3.5.1 Tasks - The GUNNERPF model represents the procedure with thirty-nine SAINT tasks, most of which correspond directly to the performance measures. The three components of the task correspond to Tasks 1 through 15, Tasks 16 through 25, and Tasks 26 through 36, respectively. Tasks 37 through 39 relate to all three components.

The first eight performance measures are represented by the first fifteen SAINT tasks. PM 1g, "squeezes magnetic brake switch while rotating gunner's power control handles to left and right," is represented by Task 9, ROTATE. Task 10, ROTLEFT,

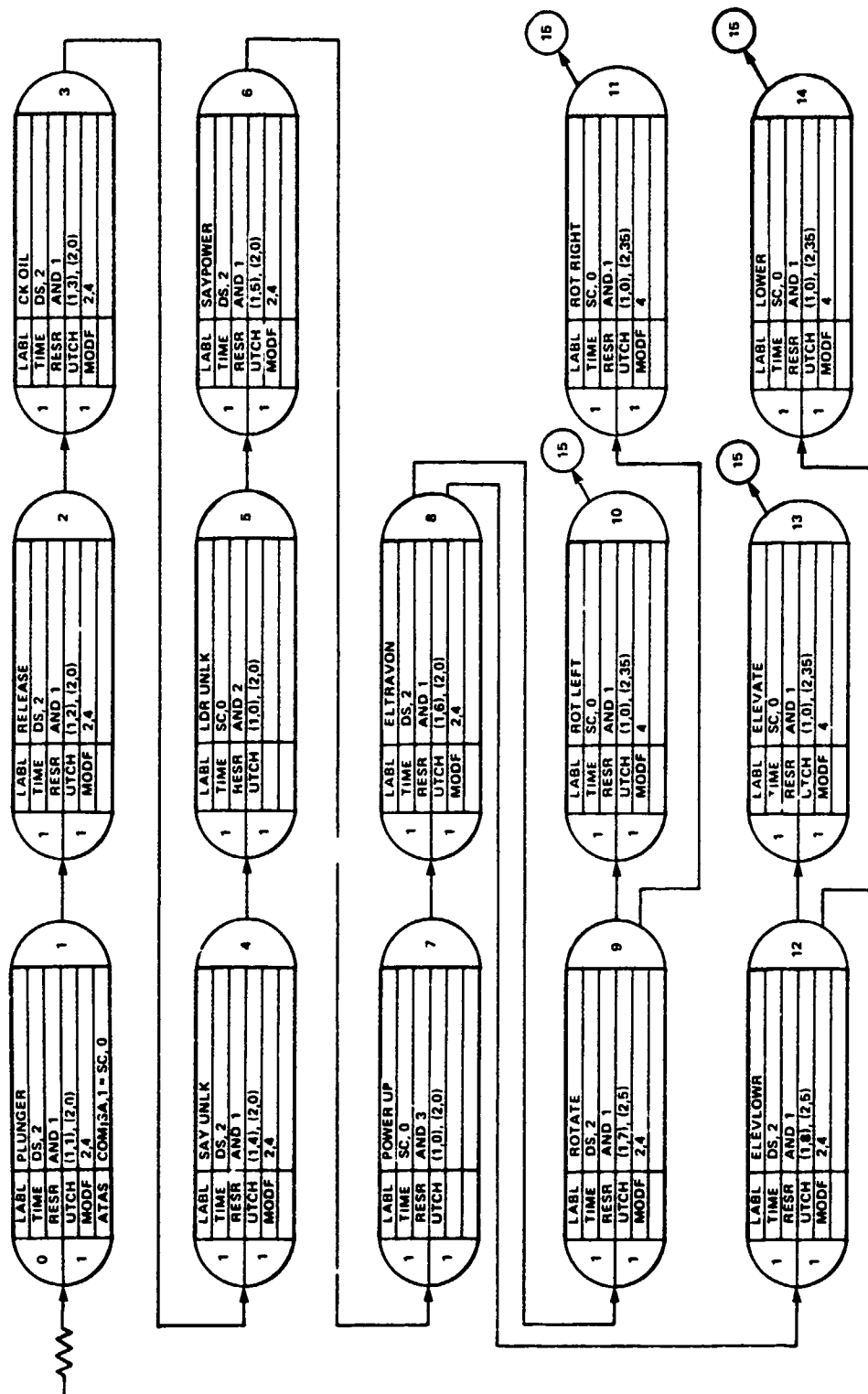


Figure 3-5
SAINT NETWORK FOR GUNNERPF

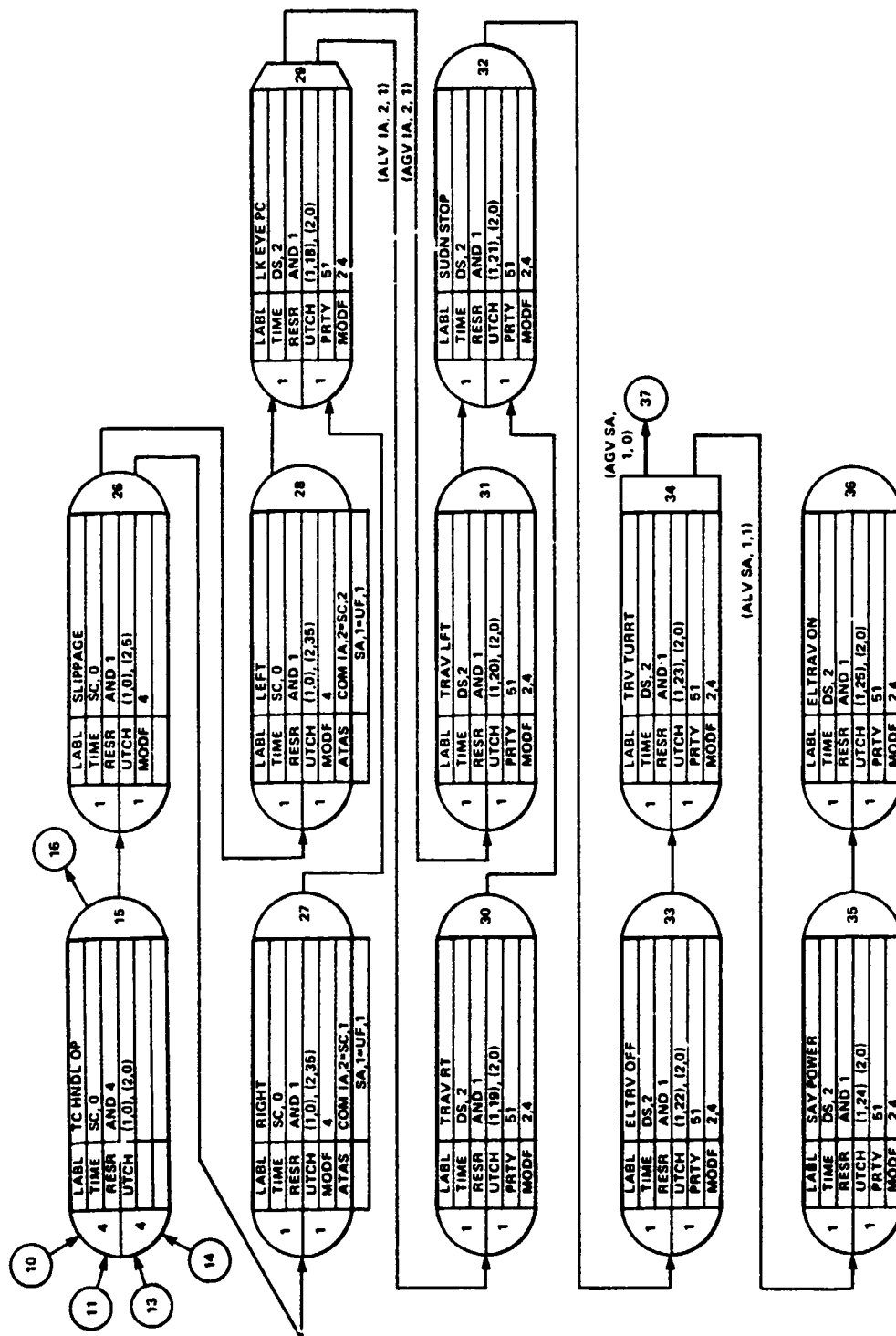


Figure 3-5
SAINT NETWORK FOR GUNNERPF
(Continued)

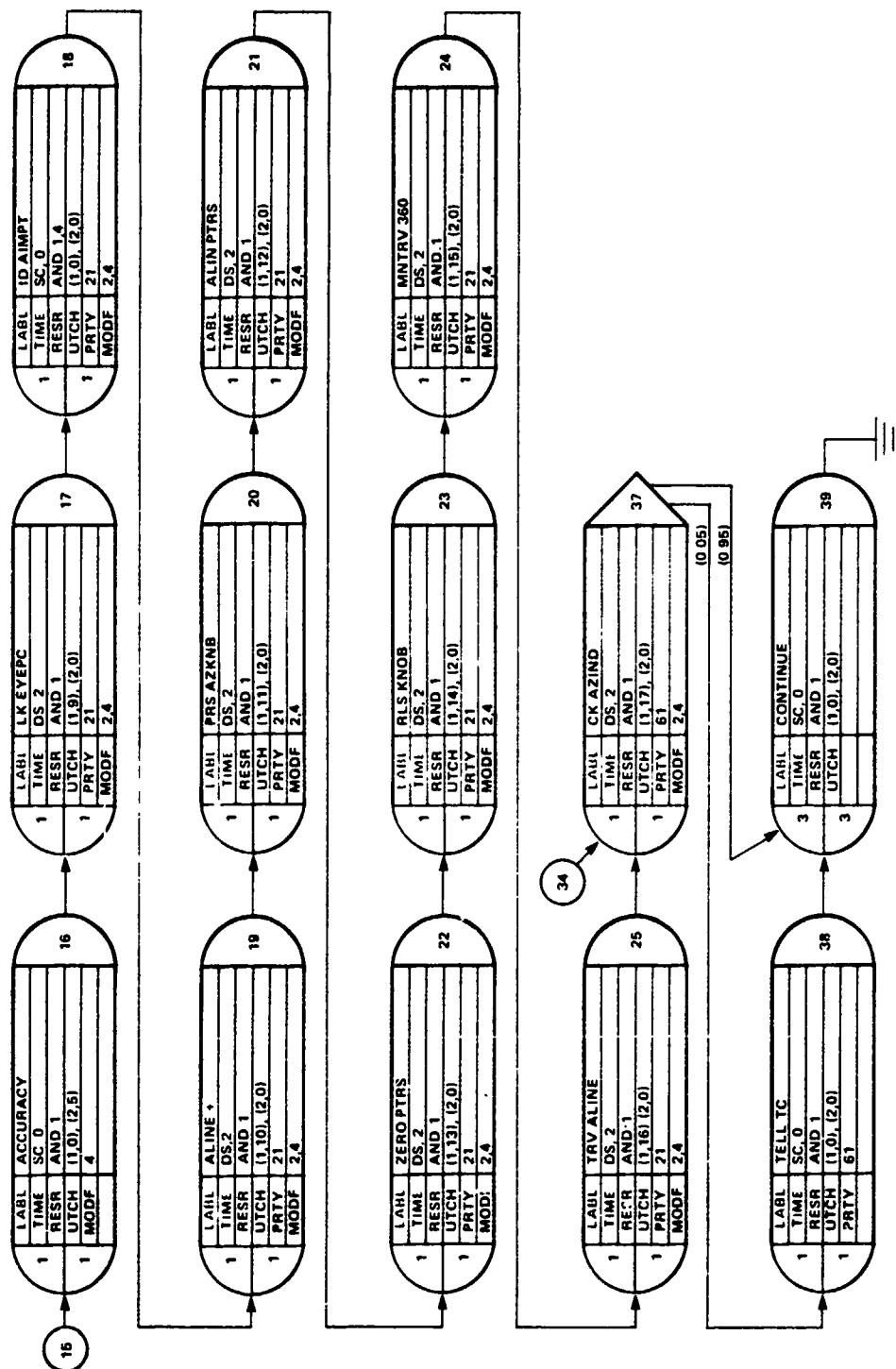


Figure 3-5
SAINT NETWORK FOR GUNNERPF
(Continued)

TABLE 3-5. PERFORMANCE MEASURES CORRESPONDING
TO SAINT TASK FOR GUNNERPF

<u>TASK</u>	<u>ASSOCIATED PERFORMANCE MEASURE</u>
1 PLUNGER	PM 1a: Holds down power solenoid plunger while rotating gunner's control handle either left or right.
2 RELEASE	PM 1b: Holds gunner control handle in position described in (a) until zero pressure is indicated on pressure gage.
3 CK OIL	PM 1c: Checks hydraulic power pack oil level by removing dipstick of oil level gage.
4 SAY UNLK	PM 1d: Tells loader to unlock turret traverse lock.
5 LDR UNLK	Scorer unlocks turret lock.
6 SAYPOWER	PM 1e: Announces "POWER."
7 POWER UP	Scorer announces "POWER UP."
8 ELTRAVON	PM 1f: Turns ELEV/TRAV power switch ON.
9 ROTATE	PM 1g: Squeezes magnetic brake switch while rotating gunner's power control handles to left and right.
10 ROT LEFT	Component of PM 1g.
11 ROTRIGHT	Component of PM 1g.
12 ELEVLOWR	PM 1h: Moves handles rearward to elevate gun; forward to lower gun.
13 ELEVATE	Component of PM 1h.
14 LOWER	Component of PM 1h.
15 TCHNDLOP	Scorer tells soldier TC's power control handles have been operated.

TABLE 3-5. PERFORMANCE MEASURES CORRESPONDING
TO SAINT TASKS FOR GUNNERPF (Continued)

TASK	ASSOCIATED PERFORMANCE MEASURE
16 ACCURACY	Used for network synthesis.
17 LK EYEPC	PM 2a: Looks through eyepiece on gunner's daylight periscope.
18 ID AIMDT	Scorer tells soldier the aiming point.
19 ALINE +	PM 2b: Alines across on aiming point using manual elevating and traversing handles.
20 PRSAZKNB	PM 2c(1): Presses resetter knob.
21 ALINPTRS	PM 2c(2): Turns resetter knob to aline middle scale pointer with inner scale pointer.
22 ZEROPTRS	PM 2c(3): Turns resetter knob moving both pointers to zero.
23 RLS KNOB	PM 2c(4): Releases resetter knob.
24 MNTRV360	PM2d: Traverses turret through complete circle using manual traversing handle.
25 TRVALINE	PM 2e: Brings aiming cross back on same aiming point.
26 SLIPPAGE	Used for network synthesis.
27 RIGHT	Used for network synthesis.
28 LEFT	Used for network synthesis.
29 LKEYEPC	PM 3a and 4a: Looks through eyepiece of gunner's daylight periscope.
30 TRAV RT	PM 3b: Uses gunner's control handles to traverse rapidly to right.
31 TRAV LFT	PM 4b: Uses gunner's contro' handles to traverse rapidly to left.

TABLE 3-5. PERFORMANCE MEASURES CORRESPONDING TO
SAINT TASKS FOR GUNNERPF (Continued)

TASK	ASSOCIATED PERFORMANCE MEASURE
32 SUNDSTOP	PM 3c and 4c: Stops suddenly while traversing.
33 ELTRVOFF	PM 3d and 4d: Turns ELEV/TRAV power switch OFF.
34 TRVTURRT	PM 3e and 4e: Traverse turret left (right) using manual traverse handle until cross is alined with original aiming point.
35 SAYPOWER	PM 3h: Announces POWER.
36 ELTRAVON	PM 3i: Turns ELEV/TRAV power switch ON.
37 CK AZIND	PM 2f, 3f, and 4f: Turns head to check that pointer is within acceptable area.
38 TELL TC	PM 2g(2), 3g(2), and 4g(2): Notifies TC pointer is not within acceptable area. Not scored in model.
39 CONTINUE	PM 2g(1), 3g(1), and 4g(1): Proceeds to next check if pointer is within acceptable area. Not scored in model.

and Task 11, ROTRIGHT are two distinct parts of TASK 9, and are included as additional observable tasks. Similarly, PM 1h, "moves handles rearward to elevate gun, forward to lower gun, while squeezing magnetic brake switch" is represented by Task 12, ELEVLOWR. Task 13, ELEVATE, and Task 14, LOWER, are the distinct parts of Task 12, and are included as additional observable tasks. Three of the first fifteen tasks represent actions performed by the scorer: Task 5, LDR UNLK, Task 7, POWER UP, and Task 15, TCHNDLOP.

The test for accuracy of the azimuth indicator may be performed either before or after the test for slippage. With the exception of Tasks 16 and 18, all model tasks involved with this check relate directly to performance measures. Task 18, ID AIMPT, describes an action performed by the scorer, identifying an aiming point. Task 16, ACCURACY, is used for network synthesis purposes.

The slippage tests are performed to the right and to the left in either order. Task 26, SLIPPAGE, Task 27, RIGHT, and Task 28, LEFT, are used to produce the proper amount of freedom in the order in which the tests are performed. The remaining SAINT tasks all correspond directly to the performance measures involved in the slippage test. Task 30, TRAV RT, and Task 31, TRAV LFT, represent the actual process of traversing the turret to the left or right. All other tasks involved in the slippage test are assumed to be the same for the two times the test is performed.

After each of the tests for accuracy and slippage, the soldier must look at the azimuth indicator to verify that it has returned to its initial setting. These checks are represented by Tasks 37, 38, and 39. Task 39, CONTINUE, corresponds to PM 2g1 and PM 3g1, but since it does not involve any action by the soldier, it was not considered to be affected

by learning and retention.

3.5.2 Precedence relationships - Most of the task completions are followed by deterministic branching. This signifies a set pattern of steps the soldier should follow. Both Task 16, ACCURACY, and Task 26, SLIPPAGE, follow Task 15, TCHNDLOP, indicating that accuracy and slippage tests may be performed in either order. However, the first test chosen will always be completed before the second is begun. Similarly, both Task 27, RIGHT, and Task 28, LEFT, follow Task 26, SLIPPAGE, indicating that the slippage tests to the right and to the left may be performed in any order. The particular order in which these tests are performed at any trial is determined by function PRIOR.

Conditional branching at Task 29 insures that the turret is traversed to the right when the slippage test to the right is being performed, and to the left when the slippage test to the left is being performed. The conditional branching at Task 34, TRVTVRRT, inserts two tasks that must be performed between the first and second slippage tests.

The completion of Task 37 is followed by probabilistic branching. This specification allows for the possibility that the pointer is not within the acceptable area after either the accuracy or slippage test.

3.5.3 Attributes - The GUNNERPF model requires one system attribute and one information attribute, in addition to those required for all models. IA(2) is used to indicate whether the right or left slippage test is being performed so that the turret is traversed in the proper direction. It is set to 1 in Task 27, RIGHT, and to 2 in Task 28, LEFT. The value of IA(2) determines the branching at Task 29. SA(1) is used as a counter so that Task 35 and Task 36 are performed only after the first

slippage test. SA(1) is initialized to 0 at the beginning of a trial, and incremented by USERF 1 in Tasks 27 and 28. The branch from Task 34 to Task 35 is taken only when SA(1) has the value of 1.

3.5.4 Resources - The four resources in the GUNNERPF model and the tasks that require them are as follows:

<u>RESOURCE</u>	<u>TASKS</u>
1 GUNNER	all except 5, 7, 15, and 18
2 LOADER	5
3 DRIVER	7
4 TANK CDR	15 and 18

The scorer performs the duties of loader, driver, and tank commander.

3.5.4 Times - Times are set to be exponentially distributed for all tasks corresponding to performance measures. For other tasks, it is assumed that no time is required.

3.6 Perform Loader's Prepare-to-Fire Checks (LOADERPF)

The particular aspect of the Loader's Prepare-to-Fire Checks that was represented in the SAINT model is the section, "Check Main Gun Firing Switches." This section involves all four tank crew positions, but the model is centered around the activities of the loader. In the procedure used to collect data for model development and validation, all other positions are played by the scorer. In this procedure, the soldier closes the breech of the main gun, inserts a circuit tester into the opening between the gun tube and the breechblock, moves the safety switch to the "fire" position, and announces "up." While the gunner and tank commander press their main

gun triggers, the loader checks to verify that the circuit tester lights when each trigger is pressed. Then the loader moves the safety switch to the "safe" position. The gunner presses two triggers, and the loader verifies that the circuit tester does not light. Finally, the gun is turned off, and the circuit tester is removed.

The SAINT network of task elements for this task is shown in Figure 3-6, and the SAINT input data are given in Appendix B. Table 3-6 compares the tasks in the SAINT model with those in the behavioral analysis shown in Appendix C.

3.6.1 Tasks - The SAINT simulation model for LOADERPF consists of twenty-one base tasks. Ten of these tasks are equivalent to the ten performance measures shown in Appendix C. The remaining tasks represent activities performed by other crew positions. The correspondence between tasks and observable activities of the crew members is relatively direct, as shown in Table 3-6.

3.6.2 Precedence relationships - The majority of the task completions are followed by deterministic branching. This signifies that there is a set pattern of steps which the soldier should follow. There is an option on the order in which gunner's and tank commander's triggers are checked. Whether gunner's or commander's switches are checked first is determined by the choice model in function PRIOR. It is assumed that all of the triggers for the first position chosen will be checked before any from the second position are checked. Tasks 9 and 10, and Tasks 16 and 17 are loops in which triggers are checked sequentially until all are checked. Task 18 has no successors, as it is always cleared by some other task, and hence, is never completed.

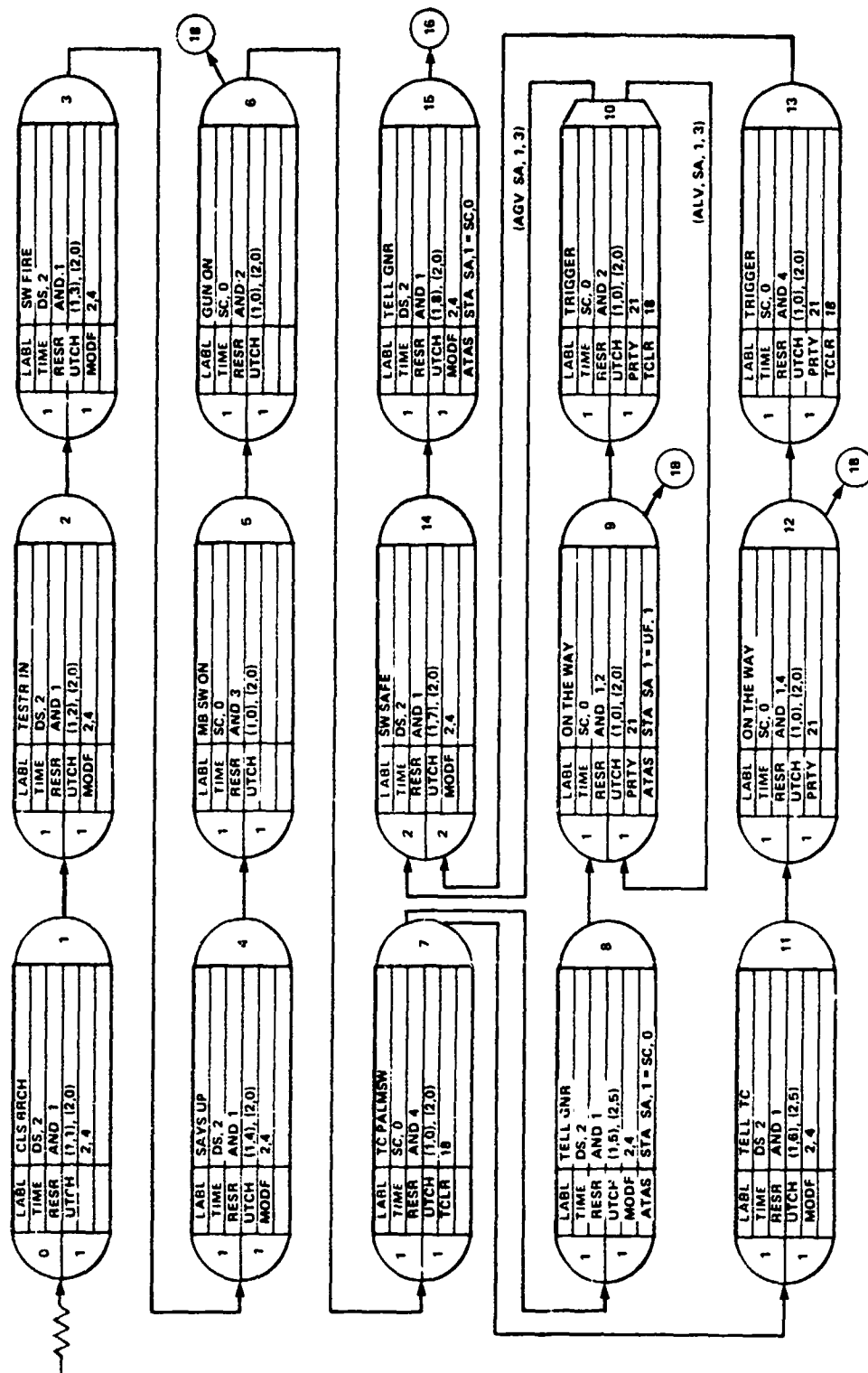


Figure 3-6
SAINT NETWORK FOR LOADERPF

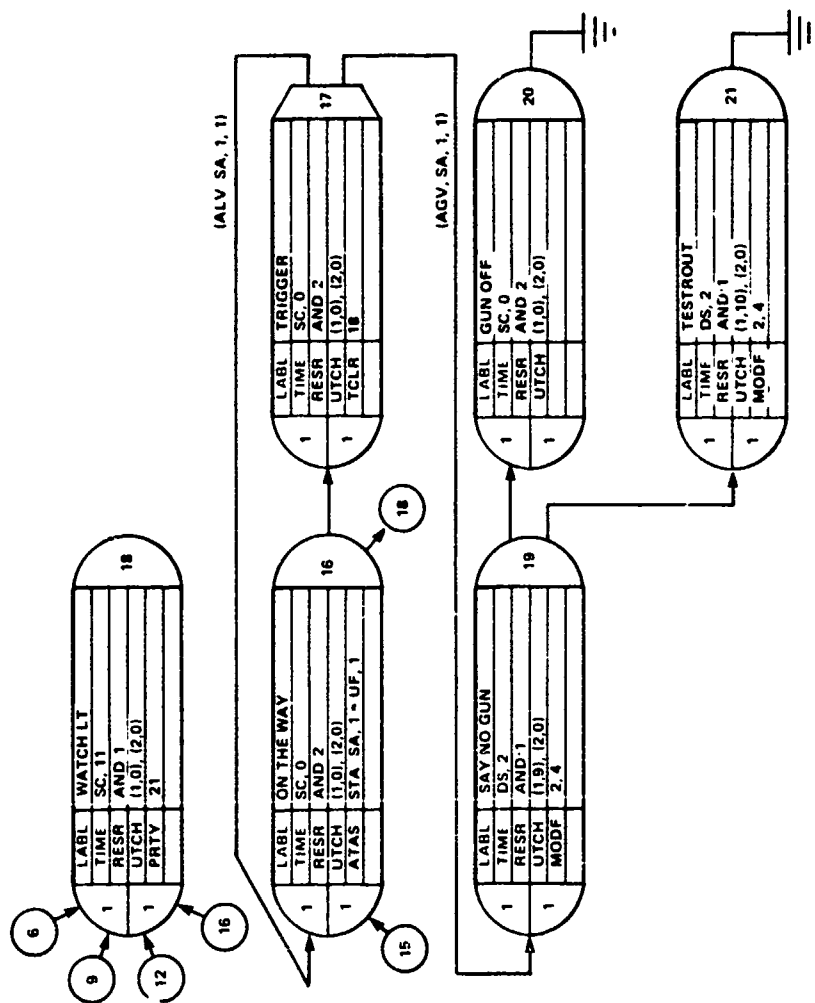


Figure 3-6
SAINT NETWORK FOR LOADERPF
(Continued)

TABLE 3-6. PERFORMANCE MEASURES CORRESPONDING TO SAINT TASKS FOR LOADERPF

TASK	ASSOCIATED PERFORMANCE MEASURE
1 CLS BRCH	PM 1: Closes breech by tripping extractors with block of wood.
2 TESTR IN	PM 2: Inserts circuits tester into opening between rear face of gun tube and front face of breechblock.
3 SW FIRE	PM 3: Moves main gun safety switch to FIRE position.
4 SAYS UP	PM 4: Announces "UP."
5 MB SW ON	Scorer turns master battery switch ON.
6 GUN ON	Scorer turns the main gun switch ON.
7 TCPALMSW	Scorer presses the commander's control handle palm switch.
8 TELL GNR	PM 5: Tells gunner to squeeze main gun triggers.
9 ONTHEWAY	Scorer announces "ON THE WAY."
10 TRIGGER	Scorer squeezes the trigger.
11 TELL TC	PM 6: Tells TC to squeeze main gun trigger.
12 ONTHEWAY	Scorer announces "ON THE WAY."
13 TRIGGER	Scorer squeezes the trigger.
14 SW SAFE	PM 7: Moves main gun safety switch to SAFE.
15 TELL GNR	PM 8: Tells gunner to press trigger on manual firing handle.
16 ONTHEWAY	Scorer announces "ON THE WAY."
17 TRIGGER	Scorer squeezes the trigger.
18 WATCH LT	Soldier watches circuit tester light while triggers are pressed.

TABLE 3-6. PERFORMANCE MEASURES CORRESPONDING TO
SAINT TASKS FOR LOADERPF (Continued)

TASK	ASSOCIATED PERFORMANCE MEASURES
19 SAYNOGUN	PM 9: Tells gunner to turn main gun OFF.
20 GUN OFF	Scorer turns main gun OFF
21 TESTROUT	PM 10: Removes circuit tester from breechblock.

3.6.3 Attributes - LOADERPF uses SA(1) as a counter for the two loops described above. The attribute is initialized to zero in Tasks 8 and 15, and incremented by USERF 1 in Tasks 9 and 16.

3.6.4 Resources - Four resources are required for the LOADERPF model. These resources are the loader, gunner, driver, and tank commander, and are designated by resource numbers one through four, respectively. The assignment of resource to tasks is as follows:

<u>RESOURCE</u>	<u>TASK</u>
1 LOADER	1, 2, 3, 4, 8, 11, 14, 15, 18, 19, 21
2 GUNNER	6, 9, 10, 16, 17, 20
3 DRIVER	5
4 TANK CDR	7, 12, 13

All roles except the loader are played by the scorer.

3.6.5 Times - As in other models, SAINT tasks that correspond to performance measures require a time sampled from the second distribution set, representing exponential distribution of times. Other tasks, with the exception of Task 18, are set to require no time. Task 18 was assigned a time sufficiently long so that it would never be completed, but would always be cleared by another task.

3.7 Engage Targets Using Precision Fire Techniques (PRECFIRE)

This exercise models the way in which a soldier engages moving and stationary targets using precision fire techniques. There are eight steps, seven of which were tested (the soldiers did not actually fire the weapon). When the soldier receives the command, he turns the main gun on and indexes the ammunition

in the computer. Then he looks through the sight (usually the periscope), and identifies the target. He selects the correct reticle (if he is using the telescope) according to the type of ammunition, and lays the crosshair at the center of the target with the correct lead applied (if the target is moving). Finally, the soldier says "on the way," and fires.

The SAINT network for PRECFIRE is shown in Figure 3-7; SAINT input data representing the network are shown in Appendix B. Table 3-7 compares the tasks in the SAINT model with those in the behavioral analysis described in Appendix C.

3.7.1 Tasks - The SAINT simulation model for PRECFIRE consists of twenty-two base tasks. Twelve of the tasks represent performance measures. The relationship between tasks and performance measures is somewhat different for this model than it is for the other models, particularly in the tasks involving conditional branching.

The first three performance measures are represented by Task 2, GUN ON; Task 3, CK COMP; and Task 8, SAYS ID. As is the case on previous models, the performance measure, "indexes ammunition," is associated with the model task that checks the computer, rather than the task that performs the indexing, because the latter is not always performed.

Choosing the correct sight involves two performance measures: looking at the periscope if it is working, and looking through the telescope if the periscope is damaged. These two performance measures are represented in Tasks 6 and 7. In this case, contrary to the usual practice, the performance

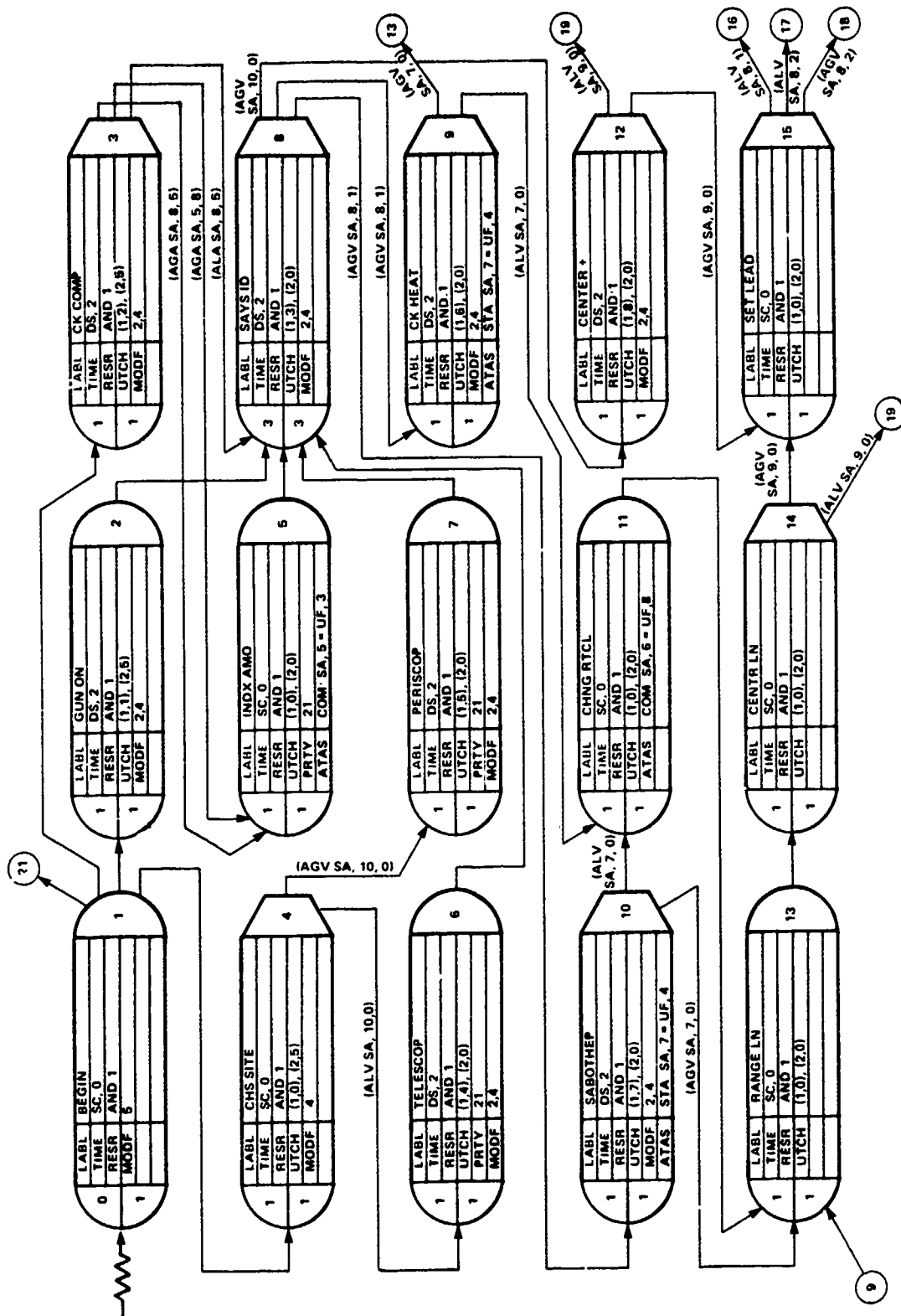


Figure 3-7
SAINT NETWORK FOR PRECFIRE

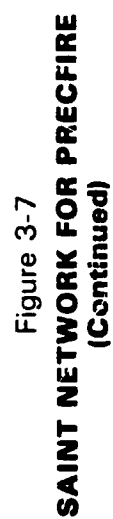


TABLE 3-7. PERFORMANCE MEASURES CORRESPONDING
TO SAINT TASK FOR PRECFIRE

TASK	ASSOCIATED PERFORMANCE MEASURE
1 BEGIN	Used for network synthesis.
2 GUN ON	PM 1: Turns main gun switch ON.
3 CK COMP	PM 2: Indexes ammunition.
4 CHS SITE	Used for network synthesis (PM 5).
5 INDX AMO	Component of PM 2.
6 TELESCOP	PM 5b: Looks through telescope.
7 PERISCOP	PM 5a: Looks through periscope.
8 SAYS ID	PM 4: Announces IDENTIFIED.
9 CK HEAT	PM 6b: Selects HEAT reticle.
10 SABOTHEP	PM 6a: Selects SABOT/HEP reticle.
11 CHNGRTCL	Component of PM 6a and PM 6b.
12 CENTER +	PM 7a: Lays periscope crosshair at center of the target.
13 RANGELN	An additional observable task.
14 CNTR LN	An additional observable task.
15 SET LEAD	Used for network synthesis.
16 5.0 MIL	PM 7d: HEAT 1800M range line 5.0 mil lead
17 2.5 MIL	PM 7b: SABOT 2000M range line, 2.5 mil lead.
18 7.5 MIL	PM 7c: HEP 1000M range, 7.5 mil lead.
19 ONTHEWAY	PM 8: Says ON THE WAY.
20 FIRE	Task element was not tested.
21 WAIT	Used for network synthesis.
22 SAYS UP	Scorer says "UP."

measure is not represented at the initial model task (in this case, Task 4, CHS SITE) even though that task is always chosen. This representation asserts that a single engagement only partially trains the task elements involved in selecting the site, although the process involved in indexing the ammunition is trained by a single engagement.

Similarly, Tasks 9 and 10 are used to represent the performance measures involved with selecting the correct telescope reticle. In addition, Task 12, CENTER +; Task 16, 5.0 MIL; Task 17, 2.5 MIL; and Task 18, 7.5 MIL, represent the performance measures involved with laying the crosshair at the center of the target with the proper lead applied.

Task 1, BEGIN, and Task 21, WAIT, are included for network synthesis purposes. TASK 1 allows Tasks 2, 3, 4, and 19 to be started at the same time for each trial. Task 21 permits time to pass before the scorer (acting as the LOADER) says "Up" (Task 21, SAYS UP).

3.7.2 Precedence relationships - Many of the tasks in PRECFIRE require decision making. Conditional-take first branching is used to represent the choice points involved in the task. These choice points include Task 3, CK COMP, in which the proper setting for the computer is chosen; Task 4, CHS SITE, in which the proper sight is chosen; Task 9, CK HEAT, and Task 10, SABOTHEP, in which the correct reticle is chosen; and Task 15, SET LEAD, in which the proper lead is selected. In other task elements, the required steps are different, depending on the situation. For example, the conditional branching in Task 8, SAYS ID, represents the fact that a reticle must be chosen only if the telescope is being used. Similarly, the branching at Task 12, CENTER +, represents differences between procedures used for moving and stationary targets.

Since all task elements are not trained a single engagement, four engagements composed a single trial. To accommodate multiple engagements within a trial, branching from Task 20 was made conditional-take first. Task 20 branched to Task 1 for the first three engagements, and to Task 50 after the fourth engagement.

3.7.3 Attributes - The conditions upon which branching is based are represented by several system attributes unique to the PRECFIRE model. Of these, SA(5) and SA(6) represent the state of the tank, while SA(8), SA(9), and SA(10) represent information that is communicated to the soldier in the initial command. SA(7) is an index of matching used to select the correct reticle. In addition, SA (1) is used to count the engagements in a trial.

SA(5) represents the current setting on the computer, with HEAT, SABOT, and HEP represented by the values, 1, 2, and 3, respectively. SA(6) represents the telescope reticle currently in place, with the value of 1 indicating the HEAT reticle, and the value of 2 indicating the HEP/SABOT reticle. SA(8) represents the type of ammunition called for in the command using the same scale as SA(5). SA(9) indicates whether the target is moving (the value, 1, indicates a moving target), and SA(10) indicates whether the periscope is working (the value, 1, indicates that it is). The value of SA(7) is calculated by USERF 4, and indicates that the correct reticle is in place with the value, 1.

3.7.4 User-defined functions - The PRECFIRE model requires five user-defined functions in addition to those required by all tasks. USERF 1 increments the engagement counter, SA(1). USERF 3 indexes the ammunition; specifically, it sets SA(5), the setting on the computer, to be equal to SA(8), the type of ammunition required. USERF 4 compares the reticle,

SA(6), with the type of ammunition required, SA(8), to see if they are compatible. It returns the value of 1 if they are, and 0 if they are not. USERF 8 changes the value of SA(6), the reticle currently in place. Finally, MODRF 5 is called by Task 1 to update the values of SA(8), SA(9), and SA (10) to represent the current engagement.

3.7.5 Resources - The two resources required in the PRECFIRE model are a gunner and a loader. The role of the LOADER is played out by the scorer and is required by Tasks 20 and 21. All other tasks require the gunner.

3.7.6 Times - Performance times have been set so that all tasks that represent performance measures require time sampled from an exponential distribution. In addition, time required for Task 21, WAIT, is exponentially distributed.

3.8 Communicate Over Tactical FM Radio AN/VRC-64 (RADIOMSG)

This exercise models the procedure a soldier uses to communicate over a tactical FM radio. In the specific procedure, the soldier communicates three routine messages to a net control station (NCS). Three exchanges with the NCS are required for the communication to occur. During the course of this research, other procedures for performing this task were identified; the alternative procedures were used to test some of the subjects.

The SAINT network for RADIOMSG is shown in Figure 3-8, and the SAINT input data are shown in Appendix B. Table 3-8 compares the tasks in the SAINT model with those in the behavioral analysis described in Appendix C. The following subsections describe the task elements, precedence relationships, attributes, resources, and times for RADIOMSG.

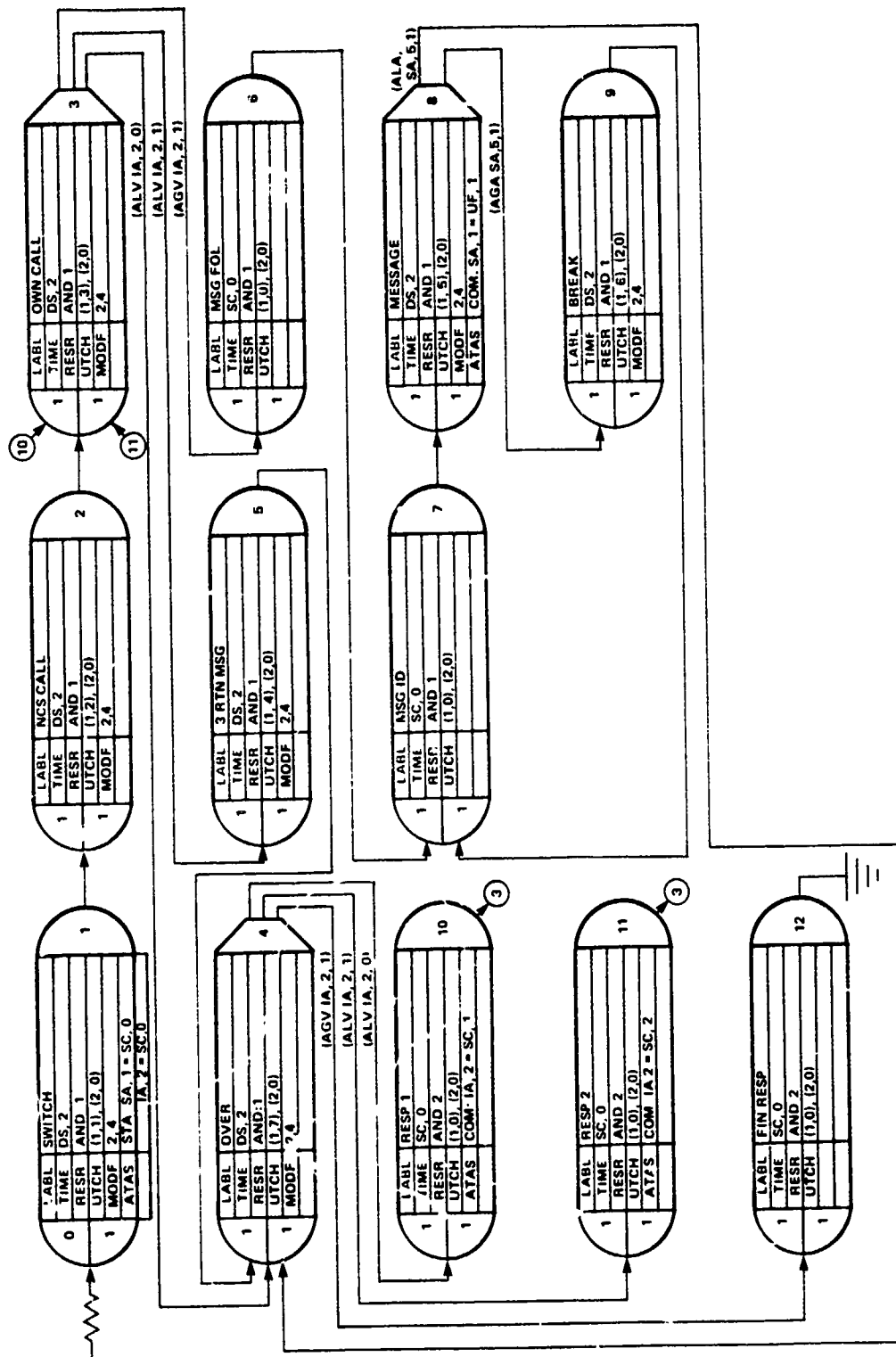


Figure 3-8
SAINT NETWORK FOR RADIOMSG

TABLE 3-8. PERFORMANCE MEASURES CORRESPONDING
TO SAINT TASKS FOR RADIOMSG

<u>TASK</u>	<u>ASSOCIATED PERFORMANCE MEASURES</u>
1 SWITCH	PM 1: Places CVC helmet switch in center position.
2 NCS CALL	PM 2: Calls net control station.
3 OWN CALL	PM 3: Identifies himself before giving the messages.
4 OVER	PM 17: Says OVER after Message #3. Also, additional observable task.
5 3 RTNMSG	PM 4: Tells net control station number of messages. PM 5: Tells net control station precedence of messages.
6 MSG FOL	An additional observable task.
7 MSG ID	An additional observable task.
8 MESSAGE	PM 6, 10, and 14: Transmits Message. PM 7, 11, and 15: Uses phonetic alphabet as required. PM 8, 12, and 16: Pronounces numbers correctly.
9 BREAK	PM 9 and 13: Says BREAK at end of Message #1 and Message #2.
10 RESP1	Response by scorer.
11 RESP2	Response by scorer.
12 FIN RESP	Response by scorer.

3.8.1 Tasks - The SAINT simulation model for RADIOMSG consists of twelve base tasks. The tasks represent performance measures and two additional observable tasks. The entire task may be divided into three exchanges between the soldier and the NCS. The soldier's part of the dialog always terminates with Task 4, OVER, and is followed by one of three responses by the NCS.

In the first exchange, the soldier calls the NCS, identifies himself, and says "over," as shown in Tasks 2, 3, and 4. The NCS then responds in Task 10. In the second exchange, the soldier identifies himself, states the number and priority of messages, and says "over," as shown in Tasks 3, 5, and 4. The NCS responds with Task 11, RESP 2. Finally, in the third exchange, the soldier identifies himself (Task 3); states that messages follow (Task 6); and gives the messages (Tasks 7, 8, and 9), terminating the exchange by saying "over." The NCS's final response in Task 12 ends the task.

The relationship between the tasks in the model and the performance measures identified by the behavioral analysis is not exact. Several performance measures, specifically PMs 7, 8, 11, 12, 15, and 16, do not correspond to tasks in the model. They are more concerned with the performance of an individual task element than with the linking of task elements into a procedure. Incorporating these PMs into the model would require considerably more detail than is present in the current model. In addition, there are some SAINT tasks, namely Tasks 6 and 7, which do not correspond to PMs although they could have been used as such. The RADIOMSG model is the one in which the SAINT representation is the furthest removed from the behavioral analysis.

3.8.2 Precedence relationships - Each of the three exchanges between the soldier and the NCS contain Task 3, OWN CALL, and Task 4, OVER. Conditional branching from these two tasks assures that the appropriate exchange is made. The messages themselves are transmitted sequentially in the loop consisting of Tasks 7, 8, and 9. Conditional branching at Task 8, MESSAGE, is used to break out of the loop when all messages have been transmitted.

3.8.3 Attributes - Two attributes are used to keep track of which exchange between soldier and NCS is currently being processed, and which of the three messages is currently being transmitted. IA(1) is used for the first purpose, and SA(1) for the second. In addition, SA(5) represents the number of messages to be sent. IA(1) is initialized to 0 at the beginning of a trial, then is set to 1 by the first NCS response (Task 10, RESP 1), and to 2 by the second NCS response (Task 11, RESP 2). Branching at Tasks 3 and 4 is based on the value of IA(1). SA(1) is also initialized at the beginning of a trial, and is incremented by USERF 1 when a message is transmitted in Task 8, MESSAGE. The loop of Tasks 7, 8, and 9 is broken when the number of messages sent in SA(1), equals the number to be sent in SA(5).

3.8.4 Resources - The two resources in the RADIOMSG model are the soldier and the NCS.

3.8.5 Times - Times for this task follow the convention set in other tasks, in which all tasks corresponding to PMs require time sampled from an exponential distribution.

3.9 Communicate Using Visual Signaling Techniques (SIGNALS)

This exercise models the procedure a soldier uses to guide a tank from the "Start" point to the "Finish" point of

a driving course using visual signals. The SAINT network of task elements is shown in Figure 3-9, and the SAINT input data are shown in Appendix B. Table 3-9 compares the tasks in the SAINT model with the performance measures identified in Appendix C. The SIGNALS model is probably the most difficult to understand by examining the network because much of the work of the model is accomplished through user-written functions.

3.9.1 Tasks - The SAINT simulation model for SIGNALS consists of thirty-two base tasks. Model variables are initialized by Task 48, INIT, and each trial begins with Task 49, SETUP. The branch from Task 50 to Task 49 is shown in the model to indicate that Task 48 is not performed for each trial. Task 1, SCHEDULE, represents starting the exercise for each set of signals. Task 2, DECIDE, represents the observable but uncertain choice process in which the next signal to demonstrate is selected.

The separate parts of fourteen distinct signals are scored by performance measures. For each signal, one task is included to initiate it and the separate parts of the signal are scored by PMs. For example, the first signal, "Gives Signal to Start Engine," uses Task 3, START TK to initiate it, and Task 11, ARMEXTND, and Task 12, ARM 360, to represent PM 1a, "Extends arm toward front to waist level," and PM 1b, "Moves arm in circular motion," respectively.

Tasks represent more than one PM when their associated action is repeated. For example, Task 21, CLSPHAND represents PM 5a, PM 10a and PM 13a, all of which are to "Clasps hands."

3.9.2 Precedence relationships - All but one of the task completions are followed by deterministic branching. Task 2, DECIDE, has probabilistic branching following its completion.

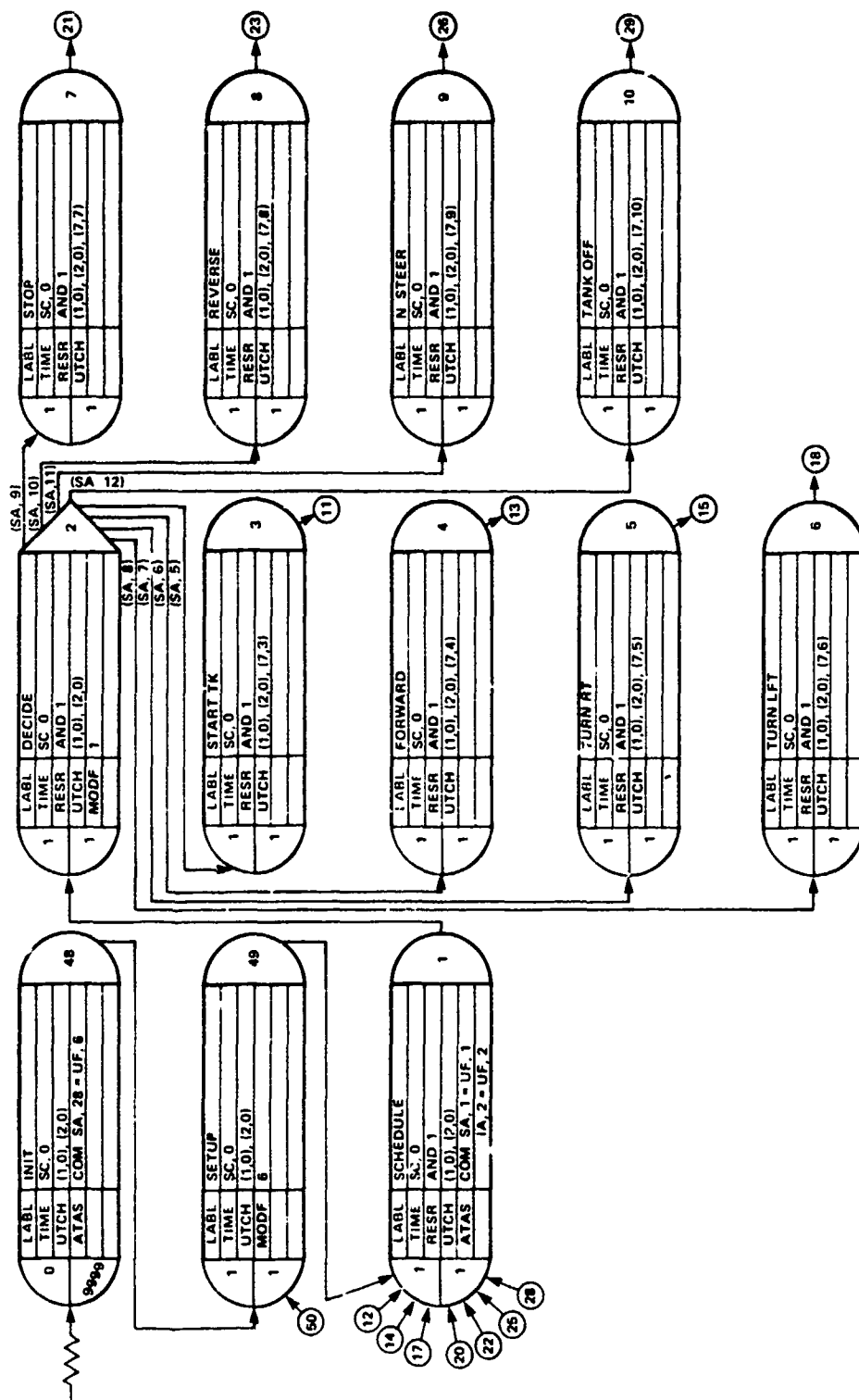


Figure 3-9
SAINT NETWORK FOR SIGNALS

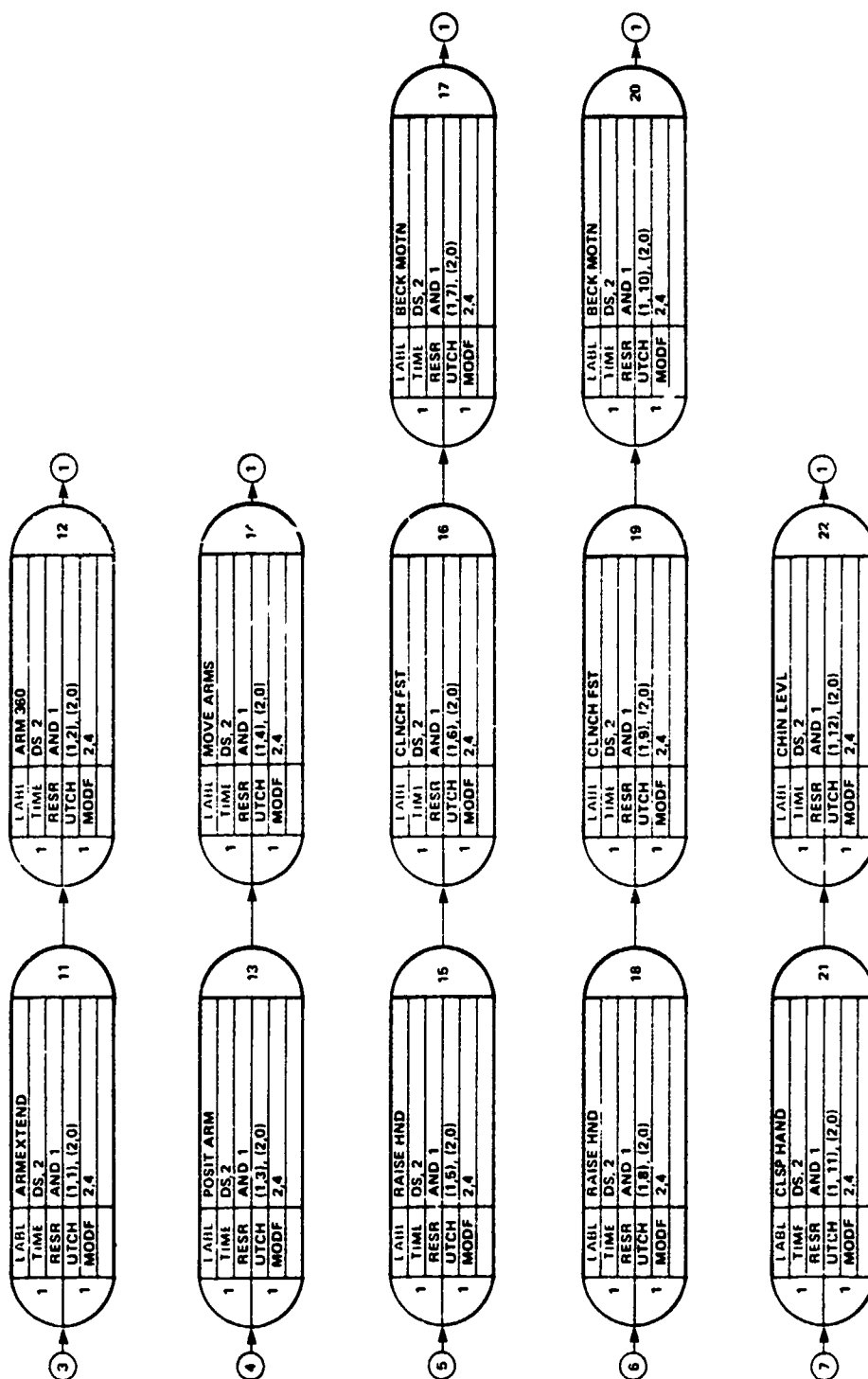


Figure 3-9
SAINT NETWORK FOR SIGNALS
(Continued)

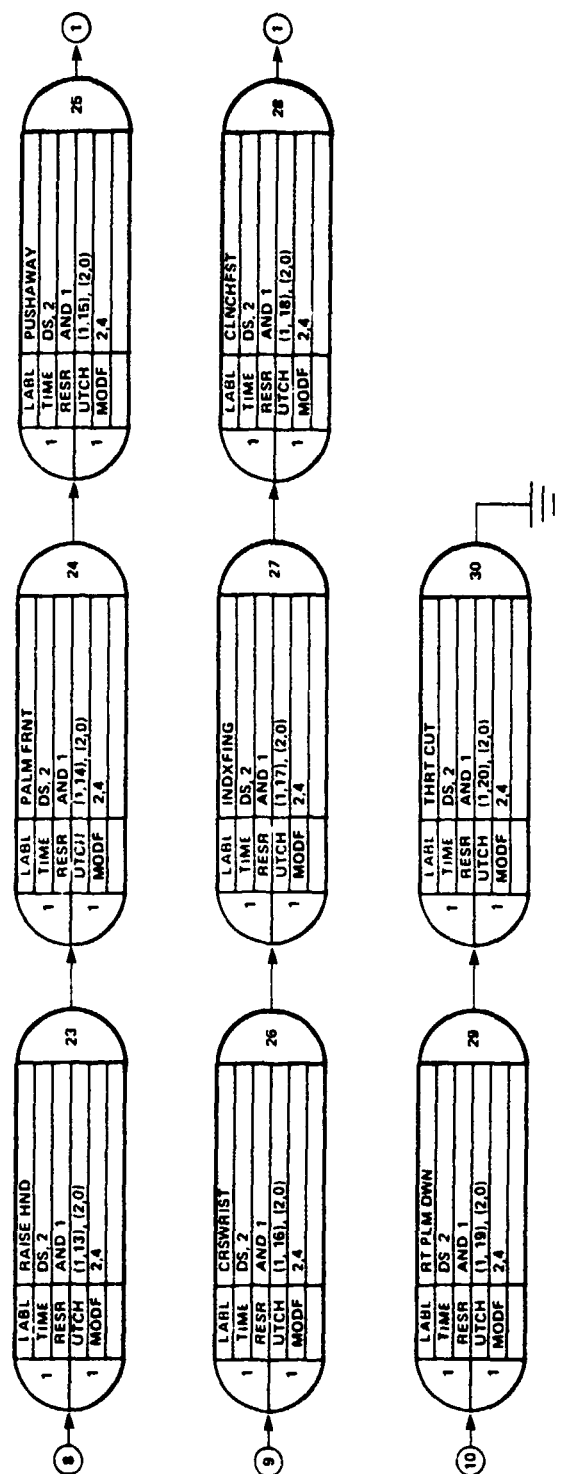


Figure 3-9
SAINT NETWORK FOR SIGNALS
(Continued)

TABLE 3-9. PERFORMANCE MEASURES CORRESPONDING TO
SAINT TASK FOR SIGNALS

<u>TASK</u>	<u>ASSOCIATED PERFORMANCE MEASURES</u>
1 SCHEDULE	An additional observable task.
2 DECIDE	An additional observable task.
3 START TK	Initiation of PM 1: Gives signal to start engine.
4 FORWARD	Initiation of PM 2, 4, 7, and 9.
5 TURN RT	Initiation of PM 8: Gives signal to turn right.
6 TURN LFT	Initiation of PM3: Gives signal to turn left.
7 STOP	Initiation of PM 5, 10, and 13: Gives signal to stop tank movement.
8 REVERSE	Initiation of PM 12: Gives signal to move in reverse.
9 N.STEER	Initiation of PM 6 and 11: Gives signal to neutral steer.
10 TANK OFF	Initiation of PM 14: Gives signal to stop engines.
11 ARMEXTND	PM 1a: Extends arm toward front at waist level.
12 ARM 360	PM 1b: Moves arm in circular motion.
13 POSITARM	PM 2a, 4a, 7a, and 9a: Position both palms toward chest.
14 MOVEARMS	PM 2b, 4b, 7b, and 9b: Move arms and hands backward and forward.
15 RAISEHND	PM 8a: Raises hands to shoulder level in front of body.
16 CLNCHFST	PM 8b: Forms clenched fist of arm indicating direction in which turn is to be made.
17 BECKMOTN	PM 8c: Makes beckoning motion with other arm to bring vehicle forward.

TABLE 3-9. PERFORMANCE MEASURES CORRESPONDING TO
SAINT TASK FOR SIGNALS (Continued)

TASK	ASSOCIATED PERFORMANCE MEASURES
18 RAISEHND	PM 3a: Raises hand to shoulder level in front of body.
19 CLNCHFST	PM 3b: Forms clenched fist of arm indicating direction turn is to be made.
20 BECKMOTN	PM 3c: Makes beckoning motion with other arm to bring vehicle forward.
21 CLSPHAND	PM 5a, 10a, and 13a: Clasps hands.
22 CHINLEVL	PM 5b, 10b, and 13b: Places hands at chin level.
23 RAISEHND	PM 12a: Raises both hands to shoulder level.
24 PALMFRNT	PM 12b: Places palms at front.
25 PUSHAWAY	PM 12c: Moves hand forward and backward as if pushing vehicle away.
26 CRSWRIST	PM 6a and 11a: Crosses wrists at throat.
27 INDXFING	PM 6b and 11b: Points index finger to tank driver's right.
28 CLNCHFST	PM 6c and 11c: Clenches fist of other hand.
29 RTPLMDWN	PM 14a: Positions right hand palm down.
30 THRT CUT	PM 15b: Draws hand across neck in "throat cutting" motion from left to right.
48 INIT	Used for network synthesis.
49 SETUP	Used for network synthesis.

This type of branching usually signifies that the order of the signals to be carried out is uncertain. However, in this model, all probabilities are set to be either zero or one by MODRF 1. A probabilistic branch was used at this point because ten SAINT tasks may follow a probabilistic branch, while only five may follow a conditional branch. The deterministic branching following the other task completions signifies that the soldier should follow a predetermined sequence of actions for each signal.

3.9.3 Attributes - Attributes play a critical role in SIGNALS. The model uses several system attributes, information attributes, and user task characteristics. The roles played by these variables are described below.

System attributes represent branching probabilities used in Task 2, DECIDE, and the sequence of signals that comprise the task. SA(5) through SA(12) represent the branching probabilities. Each of the attributes, SA(13) through SA(26), contains a number representing one of the signals. In addition, SA(1) is used as a counter, and SA(28) contains a series number which describes the particular order in which different ground-guiding tasks are presented in several trials.

A number identifying each task is stored in UTC(7). This number is relevant for Tasks 3 through 10. Conceptually, this number may be viewed as a condition under which the task will be performed. The requirements of the situation are represented in IA(2); a particular signal will be given when the situation represented in IA(2) matches the conditions of the signal represented in UTC(7).

3.9.4 User-defined functions - Most of the control in the SIGNALS model is accomplished through the use of several

user-defined functions. Listings of these functions are found in Appendix A. This section describes how they perform the task.

When the simulation begins, UINPT reads in two arrays that are stored in the labeled common block, SITVR. The arrays define the particular ground-guide courses, and the order in which these courses may be presented over trials. There are four different ground-guide courses, with each course represented by an ordering of signals. A soldier typically receives a different course on each trial. The series represents the order in which the courses are presented to the soldier. Four different series are represented in the model.

The model begins with Task 48, INIT. In this task, a series number for the particular simulated soldier is read from an input file and stored in SA(28). This number is used to generate the correct sequence of signals in Task 49, SETUP. In this task, MODRF 6 retrieves the series number and trial number from SA(28) and SA(3), respectively, and from these numbers determines the proper course. Then, it places the sequence of correct signals for that course in SA(13) through SA(26). All variables have now been set up for the particular trial.

Task 1, SCHEDULE, uses two user functions, USERF 1 and USERF 2. USERF 1 increments the step counter, which is initialized to zero at the beginning of a trial. USERF 2 places the appropriate system attribute value from SA(13) through SA(26), into IA(2). For example, on the first step, SA(1) will be incremented from 0 to 1 by USERF 1. USERF 2 will then give IA(2) the value of the first step; that is, the value of SA(13). On the next step, IA(2), will have the value of SA(14), and so forth.

Task 2, DECIDE, matches IA(2) with UTC(7) for the tasks which follow Task 2, and sets the branching probabilities. For example, suppose that the next step to be performed in a task is the signal for making a right turn, so that Task 2 should branch to Task 5, TURN RT. In this case IA(2) will have the value, 5; for Tasks 3 through 10, UTC(7) has the same value as the task number. MODRF 1 goes through Tasks 3 through 10 to see if UTC(7) matches IA(2). If no match is found, the probability of branching to that task is set to 0. If a match is found, the probability is set to 1. In the example, no match is found for Tasks 3 and 4; consequently SA(5) and SA(6) are set to 0. A match is found for Task 5, and SA(7) is set to 1. Similarly, SA(8)-SA(12) are set to 0. Thus, the model branches to the correct signal.

3.9.5 Resources - A single resource is required by this model representing the soldier performing the task.

3.9.6 Times - Times for this task are set according to the convention adopted in other tasks. Thus, Tasks 1 through 10, and Tasks 48 and 49 require no time to perform; all other tasks require a time sampled from an exponential distribution.

4.0 REPRESENTATION OF LEARNING AND PERFORMANCE PROCESSES

There are several alternatives for representing processes involved in acquisition, retention, and retrieval of information in memory. Sticha (1982) has identified two general modeling approaches which encompass these three processes. The first approach follows from the tradition of Hull (1943, 1952), and describes learning in terms of changes in the strength of memory traces. The second approach describes learning as a transition of an association from an unlearned state to a learned state. Both of these models are well-developed, and there have been successful applications of both strength (Wickelgren and Norman, 1966; Wickelgren, 1972, 1974a, b) and state (Greeno, 1974) approaches. In addition, Restle (1965) has shown that a strength model may be constructed which is equivalent to any state model.

The models developed in this project have incorporated the strength approach for two reasons. First, the strength approach gives a more parsimonious account for forgetting from long-term memory. This topic is often ignored by state theories, and simple methods for incorporating long term forgetting are contradicted by empirical results (see Wickelgren, 1972). Second, the strength approach is more easily extended to incorporate predictions of performance speed, as well as performance accuracy. In a state model, a task being learned will eventually be absorbed into a learned state. After that point, there is no mechanism within the model to represent further improvements in speed. The existence of improvements in speed long after perfect accuracy has been attained is a well-demonstrated empirical finding (Lewis, 1979; Newell and Rosenbloom, 1981; Snoddy 1926) that can be more easily incorporated within the strength framework.

Specific models were developed for the psychological processes involved in acquisition, retention, retrieval, and choice. The problem of modeling performance speed was addressed separately from performance accuracy. The following subsections describe the models that were developed for these processes.

4.1 Acquisition

The overall approach to learning and retention is based on the concept of the strength of an association. The strength of an association is assumed to be a normally distributed random variable. The probability of correctly retrieving this association is the probability that this random variable exceeds a threshold. Acquisition, according to this approach, is described by a function relating association strength to the amount of practice, or number of learning trials. The function used in the models assumes that strength increases at a constant rate to an asymptote; that is,

$$H_R = M - (M - H_1)b^{n-1}, \quad (1)$$

where H_R is the strength of the response, H_1 is the initial value of the strength, M is the strength asymptote, b is the learning rate, and n is the number of trials. The variance of H_R is assumed to be constant. A critical aspect of this model is its assumption that strength approaches an asymptote at a constant rate. The necessity of this assumption may be tested by comparing the fit of Equation 1 to the fit of a model assuming linear increase in strength. Sufficiency of the model may be assessed by comparing the fit of Equation 1 to that of a model that separately estimates the strength for each acquisition trial.

4.2 Retention

The major concern of the retention model is the description of forgetting from long-term memory. The distinction between long- and short-term memory is important because different functions describe memory dynamics for these two memory systems. When memory is tested for materials that are completely forgotten within a minute, forgetting occurs at a constant rate, leading to an exponential forgetting curve (Wickelgren and Norman, 1966; Wickelgren, 1970). For materials that are remembered over longer periods of time, the forgetting rate decreases with time, so that older memories are forgotten more slowly than newer memories (Wickelgren, 1972). Several forms of this forgetting function have been proposed. Two recent ones, both proposed by Wickelgren (1974a,b), have each received some empirical support. We will use the development of Wickelgren (1974a), termed strength-fragility theory, because of its ability to integrate long- and short-term forgetting into a single forgetting function. However, as a first approximation, we will be concerned only with the long-term component of this forgetting function.

According to strength-fragility theory, the change in trace strength over time may be described by the following equation:

$$\frac{ds}{dt} = -kfs - ps, \quad (2)$$

where s is the strength of the trace, f is its fragility, k is the decay rate operating over long time periods, and p is a measure of short-term decay related to the similarity of interpolated activities. Fragility is a decreasing function of time, governed by the following equation:

$$\frac{df}{dt} = -rf^2, \quad (3)$$

where r is the fragility decay rate. The overall strength function is determined by solving Equation 3, substituting the solution of Equation 3 into Equation 2, and solving Equation 2. The resulting strength function is given by the following equation.

$$s(t) = s_0(1 + rf_0t)^{-k/r} e^{-pt}. \quad (4)$$

where s_0 and f_0 represent initial values for s and f , respectively. Equations 2 and 4 show two forces acting on the memory trace to produce forgetting. The long-term force produces decay that behaves according to a power function. The short-term force produces exponential decay. When interference is high, as is the case when information is represented by a perceptual code, the exponential function dominates. When interference is low, as when information is represented in large, semantic chunks, the power function dominates. Thus this model integrates empirical findings of good fit for power functions in long-term forgetting (Wickelgren, 1972), with findings of exponential decay of short-term memory (Wickelgren and Norman, 1966; Wickelgren, 1970).

For the purposes of this model, we will assume that $p = 0$, and therefore, that forgetting can be described by a power function during periods in which no practice occurs. This model must be generalized to model situations in which practice periodically occurs during the retention interval. As it generally is not possible to determine exactly when practice occurs, it is necessary to model the effects of periodic practice either by simulating practice, or by approximating the effects of practice by a continuous function. Using the latter approach, the equation describing the dynamics of long-term memory would be the following:

$$\frac{ds}{dt} = -kfs + a(M - s), \quad (5)$$

where M is the strength asymptote, and " a " is a factor representing both the frequency and the effectiveness of training. The second term of Equation 5 representing acquisition is consistent with the model presented in section 4.1. Thus, the model assumes that the effect of practice during a retention interval is analogous to the effect of training trials, although they may be either more or less effective.

The solution to Equations 3 and 5 is given below:

$$s(t) = (1 + rf_0 t)^{-k/r} e^{-at} [s_0 + \int_0^t aM(1 + rf_0 t)^{k/r} e^{at} dt]. \quad (6)$$

The integral in Equation 6 can be solved analytically only when k/r is an integer. There are a variety of numerical methods that may be used to solve the integral, however. Equation 6 represents the basic model of retention with periodic practice which will be incorporated into the learning and performance models.

4.3 Retrieval

The basic task of performing the steps of a procedural task in the proper order is one of cued recall. Each step is cued by the completion of the previous step. A retrieval model for such a task (Wickelgren, 1974b) is a threshold model in which the probability of a correct response is the probability that the strength of the correct association exceeds a threshold. The procedure used to collect data involved the use of a number of prompts in levels varying from weak to strong. The hints may be viewed as momentarily raising the strength of the correct association, or equivalently, lowering the threshold for a correct response. Since three levels of hint were defined in the experimental procedure three thresholds are defined in the model. These thresholds are assumed to be constant for any specific task.

4.4 Performance Speed

Performance speed at the task element level is a function of many variables which are neither controlled nor measured in this research. In addition, we were not able to collect sufficient speed data to test models of task-element performance speed. The analysis of speed will concentrate on the time taken to perform the entire task. Two simple models will be used to characterize performance speed. In the first model, speed is a function of the difference between the strength and the strength asymptote. In the second model, speed is a function of the reciprocal of the strength. In each of these models, the time devoted to giving hints must also be taken into account. More detailed models of the retrieval process would yield better predictions of performance speed and might also account for the speed-accuracy tradeoff.

4.5 Choice

For a number of tasks there are several orders in which task elements may be performed. A choice mechanism is required to pick one task-element order from among several acceptable orders. The mechanism which is hypothesized to describe choice is based on the Law of Comparative Judgment (Thurstone, 1927). According to this model, each of the alternative correct task elements is characterized by a priority, which is a normally-distributed random variable. Each task elements differs in its mean priority; the standard deviation of the priority variables is assumed to be constant for all task elements. When an individual is faced with a choice between two or more task elements, he draws samples from the priority distributions of each of the candidate task elements, and chooses the step with the highest sampled priority.

4.6 Summary

This section has described the psychological models of learning, retention, retrieval, and choice which will be incorporated into a general model of acquisition and retention of military procedures. The descriptions in this section, however, have been at an abstract level. The specific methods by which these models are integrated with SAINT models describing the control of performance of task elements is presented in section 5.0.

5.0 MODEL REPRESENTATION OF PSYCHOLOGICAL PROCESSES

The psychological models described in the previous section are implemented in the SAINT models by four subroutines, LEARN, RETEN, MODRF 2, and PRIOR. The ways in which these subroutines represent the psychological models of acquisition, retention, retrieval, and choice, and determine performance speed are described in this section.

5.1 Acquisition

Subroutine LEARN updates the strength of a particular SAINT task according to the learning model presented in Equation 1. A listing of LEARN is presented in Figure 5-1. LEARN is called by MODRF 2, and hence, operates only on those SAINT tasks that correspond to performance measures. The first three lines of LEARN find the appropriate index to the common variables, and check to ensure that the task being learned corresponds to a performance measure. Then the current strength is retrieved, updated, and stored in UTC(3).

5.2 Retention

The retention function implemented in the SAINT models is the long-term (power decay) component of the function described in Equation 4. The retention function is implemented in the subroutine RETEN, the listing of which is displayed in Figure 5-2. The two parameters of the function, stored in common block PARMS, are STDEC and FRDEC, which represent k/r and r in Equation 4, respectively. Equation 6 is not represented in the SAINT model because practice on a task is simulated, rather than modeled through a differential equation. Simulating the effects of practice should allow the modeler more freedom in specifying the interval between practice periods.

```

      SUBROUTINE LEARN(ITASK)
      COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3TIM
      +          TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2(4
C *****
C * LEARN - THIS FUNCTION UPDATES THE STRENGTH OF ITASK IN UTC(3)
C * BY THE PROPORTION, TRLRN, TO THE ASYMPTOTE, SASMT. IT IS CALLED
C * BY MODRF 2.
C *****
      CALL GETTC(ITASK,1,VALUE)
      INDEX=VALUE
      IF (INDEX.EQ.0) GO TO 100
      CALL GETTC(ITASK,3,STRNT)
      STRNT=(1-TRLRN(INDEX))*STRNT + TRLRN(INDEX)*SASMT(INDEX)
      CALL PUTTC(ITASK,3,STRNT)
100 RETURN
      END

```

Figure 5-1. LISTING OF SUBROUTINE LEARN

```

      SUBROUTINE RETEN(ITASK,TIME)
      COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3T
+      TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2
C *****
C *   THIS SUBROUTINE UPDATES THE STRENGTH AND FRAGILITY OF A TRAC
C *   FOR TASK "ITASK" TO BE CURRENT TO TIME "TIME".
C *****
      CALL GETTC(ITASK,6,TLAST)
      IF (TIME.LE.TLAST) GO TO 100
      CALL GETTC(ITASK,1,VALUE)
      INDEX=VALUE
      IF (INDEX.EQ.0) GO TO 100
      CALL GETTC(ITASK,3,STRNT)
      CALL GETTC(ITASK,4,FRAGL)
      CALL PUTTC(ITASK,6,TIME)
      TDIFF=TIME-TLAST
      STRNT=STRNT*(1+FRDEC*FRAGL*TDIFF)**(-STDEC)
      CALL PUTTC(ITASK,3,STRNT)
      FRAGL=FRAGL/(1+FRDEC*FRAGL*TDIFF)
      CALL PUTTC(ITASK,4,FRAGL)
100  RETURN
      END

```

Figure 5-2. LISTING OF SUBROUTINE RETEN

Because the differential equation representing strength does not depend on time, except as it is mediated by trace fragility, it is not necessary to know the initial values of strength and fragility in order to calculate the appropriate decay proportions. The values of strength and fragility at any time may be used to update the strength and fragility to any later time. This strategy was used to update the two theoretical variables in RETEN.

The first five lines of RETEN ensure that the time to which strength and fragility are to be updated is after the time of last update, and that the SAINT task which is to be updated corresponds to a performance measure identified by the behavioral analysis. Then, strength and fragility are retrieved from UTC(3) and UTC(4), respectively, and the current time is stored in UTC(6). Strength is then updated using the value of fragility at the time of last update; the new value is stored in UTC(3). Similarly, fragility is updated and stored in UTC(4).

It should be noted that use of RETEN assumes that no practice has occurred during the interval between updates. Thus, when practice is being simulated and subroutine LEARN is called, RETEN must be called immediately prior to a call to LEARN. This procedure has been followed in the simulations (in MODRF 2).

5.3 Retrieval

Subroutine MODRF 2 embodies the retrieval model, sets the performance speed, integrates the SAINT model with the learning and retention functions, and stores statistical variables. A listing of MODRF 2 is given in Figure 5-3. The retrieval mechanism operates by comparing the momentary strength, stored in UTC(5), with the three response thresholds, stored in the common

```

C *****
C * MODRF 2 - THIS FUNCTION CHECKS THE TRACE STRENGTH AND BRANCHES
C * TO THE APPROPRIATE HINT LEVEL. TTIME IS ADJUSTED. HINT LEVEL
C * IS RECORDED AND OVERALL GO/NO GO IS UPDATED.
C *****
C
C      FIND THE HINT LEVEL BASED ON TRACE STRENGTH, UTC(5).
C
C      20 CALL GETTC(NTASK,1,VALUE)
C         INDEX=VALUE
C         IF (INDEX.EQ.0) GO TO 29
C         CALL GETTC(NTASK,5,STMOM)
C         XLEVEL=0.0
C         IF (STMOM.LT.THR50(INDEX)) XLEVEL=1.0
C         IF (STMOM.LT.THR51(INDEX)) XLEVEL=2.0
C         IF (STMOM.LT.THR52(INDEX)) XLEVEL=3.0
C
C      SET TTIME AND GO/NG VARIABLE, SA(2), DEPENDING ON HINT LE
C
C      CALL GETTC(NTASK,3,STRNT)
C      TTIME=TCONS+TSTRN*(SASMT(INDEX)-STRNT)+RNORM(1)*TSTDV**2
C      IF (XLEVEL.GT.0.9) TTIME=TTIME+P1TIM
C      IF (XLEVEL.GT.1.9) TTIME=TTIME+P2TIM
C      IF (XLEVEL.GT.2.9) TTIME=TTIME+P3TIM
C      TTIME=AMAX1(TTIME,0.0)
C      IF (XLEVEL.GT.0.0) CALL PUTSA(2,0.0)
C
C      STORE STATISTICAL VARIABLES FOR THIS SUBTASK.
C
C      CALL GETSA(3,TRIAL)
C      NTRIAL=TRIAL
C
C      ICLCT=(NUMPM+1)*(NTRIAL-1)+INDEX
C      XOUT=0.0
C      IF (XLEVEL.GT.0.5) XOUT=1.0
C      CALL UCLCT(XOUT,ICLCT)
C      CALL UHIST(XLEVEL,ICLCT)
C
C      ADJUST STRENGTH IN UTC(3) USING RETEN AND LEARN
C
C      TIME=TNOW+TTIME
C      CALL RETEN(NTASK,TIME)
C      CALL LEARN(NTASK)
C      29 RETURN

```

Figure 5-3. LISTING OF MODRF 2

block PARMS. The momentary strength, set in PRIOR, is the sum of the average strength stored in UTC(3), and a variable sampled from a standard normal distribution. The prompt level, XLEVEL, is set depending on the value of the momentary strength.

5.4 Performance Speed

MODRF 2 represents a model of performance speed in which speed is a function of the difference between trace strength and the strength asymptote. It is assumed that the variance in performance time is independent of the variance in momentary strength; that is, two independent samples are taken to determine momentary strength and performance speed. In order to conform to the procedure used to estimate model parameters (Miller and Grenno, 1978), performance time was assumed to be normally distributed.

The determination of performance time requires seven parameters in common block PARMS and the trace strength stored in UTC(3). The parameters represent the strength asymptote, a constant time required by the SAINT tasks, a multiplier on the difference between the strength and the strength asymptote, the standard deviation of the time distribution, and the times required by the three levels of hints. Time is constrained to be greater than zero.

5.5 Choice

The choice model is implemented in function PRIOR, shown in Figure 5-4. PRIOR has the dual purposes of setting the momentary strength and the priority of a SAINT task. When PRIOR is called, it first checks to see if it has already set the priority for that task. This check is performed by examining the momentary strength. Momentary strength is initialized to -11 and is reset to that value (by MODRF 4) whenever a task

```

      FUNCTION PRIOR(ITASK)
      COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NFRNT,NFUNCH,
+             NRNIT,NRENT,MNDC,NDC,NDTN,NNTC
      COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3TIM
+             TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2(
C *****
C * THIS FUNCTION PERFORMS THREE TASKS:
C *     1. IT READS UTC(2) AND SETS THE PRIORITY BASED ON DS,1.
C *     2. IT READS UTC(3) (STRENGTH) AND SETS UTC(5) (MOMENTARY
C *        STRENGTH) BASED ON DS,1.
C *     3. IT SETS THE PRIORITY TO -11 IF THE STRENGTH IS NOT
C *        GREATER THAN THRS0.
C * USE OF THIS FUNCTION DURING MULTIPLE-RUN SIMULATIONS (E.G. LEA
C * ING TRIALS) ASSUMES THAT UTC(5) IS RESET TO -11 WHEN THE TASK
C * PERFORMED. IN ADDITION, UTC(5) MUST BE INITIALIZED TO -11.
C * OTHERWISE PRIORITIES AND STRENGTHS WILL NOT CHANGE OVER TRIALS.
C * DYNAMIC PRIORITIES ARE REQUIRED FOR STARTANK, STOPTANK, GUNNER
C * LOADERPF, AND PRECFIRE.
C *****
C
C     SET MOMENTARY STRENGTHS.
C
      CALL GETTC(ITASK,1,VALUE)
      INDEX=VALUE
      CALL GETTC(ITASK,5,STMOM)
      IF (STMOM.GE.-10) GO TO 10
      CALL RETEN(ITASK,TNOW)
      CALL GETTC(ITASK,3,STMOM)
      STMOM=STMOM+RNORM(1)
      CALL PUTTC(ITASK,5,STMOM)
C
C     SET TASK PRIORITY.
C
      CALL GETTC(ITASK,2,VALUE)
      IF (VALUE.EQ.0) GO TO 10
      PRIOR=VALUE+RNORM(1)
      CALL PUTPR(ITASK,PRIOR)
      IF (INDEX.EQ.0) RETURN
      IF (STMOM.LT.THRS0(INDEX)) PRIOR=-11
      CALL PUTPR(ITASK,PRIOR)
      RETURN
C
C     GO HERE IF PRIORITY ALREADY SET.
C
10 CALL GETPR(ITASK,VALUE)
   PRIOR=VALUE
   RETURN
   END

```

Figure 5-4. LISTING OF FUNCTION PRIOR

is performed; if the momentary strength is greater than this value, then the momentary strength and priority need not be set again.

If the momentary strength is at its initial value, PRIOR determines the current value of strength by calling RETEN. Then, it calculates the momentary strength by adding a sample from a standard normal distribution to the average strength. If the momentary strength is not sufficiently strong to produce a correct response, then the priority is set to -11, a value that is lower than could be produced if the strength were sufficient for a correct response. This assignment of priority is made to reflect a characteristic of the experimental situation that the scorer cannot determine that a response is not known by a soldier unless that response is the only permissible response in that situation.

If the strength of the task is sufficient to produce a correct response, then the priority is set to be the sum of the average priority, stored in UTC(2), and a sample from a standard normal distribution. The assumptions of this procedure are equivalent to those of Case V of Thurstone's (1927) Law of Comparative Judgment.

6.0 PARAMETER ESTIMATION AND MODEL TESTING

The models described in the previous sections were developed further by estimating the values of the free parameters, and by examining the extent to which the models could predict the data collected from soldiers performing the eight tasks. Data were collected from two sources, soldiers in training who had just received their formal training on the tasks, and soldiers in an operational unit with less than two years experience since they received their initial training.

The data were used for the dual purposes of estimating model parameters and testing the models. Naturally, these two processes are closely related. The goal of parameter estimation is to find a set of parameter values that best characterizes the data in the sense that they maximize some measure of goodness-of-fit of the model. The goodness-of-fit value attained by the optimal parameter estimates is a test of the model. More specifically, the necessity and sufficiency of a model may be tested by comparing the goodness-of-fit of the optimal parameters for the models with simpler or more complicated models, respectively.

This section describes an analysis in which the goodness-of-fit for several general models is determined. A later report will present a more detailed analysis of issues uncovered by the results presented in this report.

6.1 Method

Performance data on the eight armor tasks were collected from two samples of soldiers: one from an operational unit, and one from soldiers in One Station Unit Training (OSUT).

6.1.1 Operational unit data collection - Data from the operational unit sample primarily addressed the parameters of the retention models.

Subjects - Subjects were 116 soldiers from operational units of Ft. Knox, Kentucky who had completed the OSUT training program within thirty-one months prior to the study.

Procedure - Soldiers from the operational unit were randomly assigned to one of eight test stations. Each soldier proceeded in a "round robin" fashion to the next station until he had performed all of the eight tasks. At each test station, the soldier was given one opportunity to perform a task. The scorer read a set of instructions to inform the soldiers of the task and any specific conditions to consider during performance (e.g., moving or stationary targets during precision fire engagements). After reading the instructions, the scorer did not intervene during the performance of the task unless the soldier made an error.

If the soldier committed an error on a step, the scorer gave the soldier some assistance. If this degree of assistance was not sufficient to produce correct performance, the scorer gave stronger assistance, until correct performance for the step was achieved. The following three levels of assistance were used:

Level 1 - Remind the soldier what the overall task is and tell him the steps he has performed up to that point.

Level 2 - Tell the soldier what the next step is.

Level 3 - Show the soldier how to do the step.

After the soldier demonstrated the step correctly,

he proceeded to the next step and continued until he had completed the task.

While the soldier performed the task, the scorer recorded data on correct performance of task steps, including the order in which steps were performed, the type of error committed, the level of assistance given, and the elapsed time. Questionnaires were used to collect information on each soldier's background and task-related job experience. In addition, Armed Service Vocational Aptitude Battery (ASVAB) scores and level of education were obtained from personnel records.

6.1.2 OSUT data collection - Data from four OSUT companies primarily addressed the parameters for the learning models.

Subjects - Subjects were 471 soldiers from four OSUT companies of an OSUT Brigade in Ft. Knox, Kentucky. The soldiers were in their fifth to tenth week of training.

Procedure - Testing and training trials included five acquisition trials and a retention trial, for a total of six performances by the soldier. Each soldier performed two of the eight tasks. For each task tested, they reported to the test site twice during a twelve-week data collection period. In the first session, the soldier performed the task five times using the same procedure described for the operational unit. Approximately four weeks after the first session, the soldier returned to perform the task one time. The first session was timed to coincide approximately with the time of initial training of the task; the second session coincided with the time of the gate test for that task.

6.1.3 Differences between the samples - Minor changes were made in the definitions of the tasks between the

operational unit and OSUT sessions. These changes were made to accommodate changes in Army training policy, or to reflect details about the data collection procedure. These changes are described below:

Load an M240 Coax Machinegun - One performance measure (pulls charger handle back and to side of gun) was deleted from the scoresheet since this step is not trained in OSUT.

Start the M60A1 Tank Engine - Two steps concerning the generator switch were deleted, since the driver training devices used for the OSUT study do not have this switch. Also, the sequencing requirements involved in setting the parking brake were modified slightly.

Stop the M60A1 Tank Engine - Sequencing requirements involved in setting the parking brake were modified slightly, consistent with the changes made in Start the M60A1 Tank Engine.

Perform Loader's Prepare-to-Fire Checks - Four performance measures, in which the loader told either the gunner or the tank commander to squeeze the triggers, were deleted.

Communicate over Tactical FM Radio - The procedure used in the operational unit was in error, and was changed to combine all three messages into a single message for the OSUT data collection. After training half the platoons, project staff learned that the procedure had been changed. The format was revised to comply with the Army's procedure. The procedure used for the second half of the platoons involved two fewer steps than for the first half of the platoons.

These changes have differing effects on the comparability between the two samples. The biggest effect is in the radio task; comparisons between the operational unit and OSUT scores will be uninterpretable for this task. Within the OSUT group, however, it will probably be possible to get a picture of the course of learning by combining data from the two procedures, where possible. The other changes are much more minor, and were handled by minor changes in scoring procedures for data from one of the samples, or by making assumptions about the comparability of the task elements common to both samples.

6.2 Overview of Analysis

Data consist of performance measures, measures of the extent of practice in the unit, soldier experience, and demographics. The major dependent variables for this analysis are the levels of assistance required for correct performance of the task elements and the time required to perform the tasks. The independent variable for the OSUT group is the trial number; for the operational unit the independent variable is the "retention interval," or the time between the soldier's graduation from OSUT and the performance test.

Our chief concern in this analysis is the prediction of error rates and associated levels of assistance required, rather than prediction of performance time. Consequently, the error data will be analyzed in considerably greater detail than performance times. Specifically, levels of assistance are analyzed for each performance measure, while performance time is analyzed only for the task as a whole. There are several reasons for restricting the analysis of performance times to the task level, rather than the task-element level. First, it was not possible to obtain accurate measures of time at the task-element level because some task elements required very little time to perform. Second, task-element performance

is affected by external variables such as the equipment and scorer behavior; these effects might hide the learning and retention effects that are being sought. Performance time for the entire task draws from a larger sample of behavior, and hence, learning and retention effects should be more evident. Finally, a model of performance time for an individual task element involves more details of the nature of that task element, instead of the procedural aspects of the task which are the focus of this study. For these reasons, performance time was modeled at the task level rather than the task-element level.

A common strategy for parameter estimation and testing of simulation models is to perform sensitivity analyses in which model parameters are varied through reasonable ranges, until a model that closely resembles to the data is found. We have chosen a much more rigorous and formal approach to parameter estimation and model testing which takes advantage of the fact that the psychological models can be tested outside the context of the SAINT framework. This approach involves the determination of maximum-likelihood estimates of the parameters, using a general, iterative optimization routine. The necessity and sufficiency of a model can be assessed by comparing the goodness-of-fit of that model to both more and less general models by using a log-likelihood chi-square test. Miller and Greeno (1978) have described procedures for testing models that predict both error probabilities and response times using maximum likelihood estimates.

Data from each task were divided into two groups. Model parameters were estimated from the first half of the data, termed the model-development group. These parameters were then applied to the second half of the data, termed the cross-validation group. The ability of a model to describe the behavior of a different set of individuals is an assessment of

the importance of individual differences in model parameters.

6.3 Results for OSUT Data

OSUT data consist of error and performance time data for five learning trials and one retention trial. The models for these data are principally concerned with various representations of learning. Since there is only one retention interval, it is not possible to test different functional forms of retention function. Therefore, retention models represent forgetting as a percentage decay over the retention interval. Models tested and results are described in the following subsections.

6.3.1 Hierarchy of models - Parameters were estimated for twelve learning and retention models. Six of these models predict the distribution of assistance levels over trials. The remaining six predict both the distribution of assistance level and performance time. The models will be referred to by LEARN 1 through LEARN 12, or by the short descriptive titles given in Table 6-1. The hierarchy of these models is presented in Figure 6-1. Two models connected by a line in Figure 6-1 are nested: that is, the model that is higher in the figure is a special case of the model that is lower in the figure. A brief description of the models is given in the remainder of this section.

The basic learning model presented in section 4.1 is represented in LEARN 4. This model postulates geometric increase of strength over trials to an asymptote. Between the final training trial and the retention trial, the strength

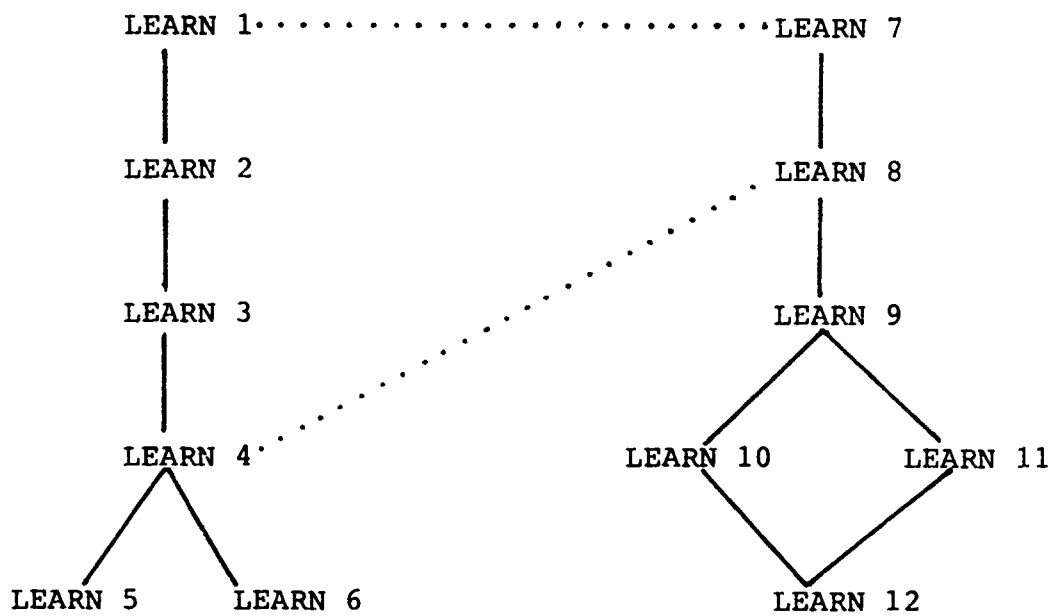


Figure 6-1. HIERARCHY OF LEARNING MODELS
FOR OSUT SAMPLE

TABLE 6-1. DESCRIPTION OF LEARNING MODELS
FOR OSUT SAMPLE

<u>Model</u>	<u>Description</u>	<u>Parameters</u>
LEARN 1	Constant Strength Model for Error Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3)
LEARN 2	Constant Increment Model for Error Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = SINCR
LEARN 3	Constant Rate Strength Model for Error Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = SINCR X(5) = SASMT
LEARN 4	Strength Model with Forgetting for Error Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = SINCR X(5) = SASMT X(6) = STDEC
LEARN 5	Trial Strength Model for Error Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = STRNT(2) X(5) = STRNT(3) X(6) = STRNT(4) X(7) = STRNT(5) X(8) = STRNT(6)
LEARN 6	Task-Element Groups Strength Model for Error Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = SINCR(1) X(5) = SASMT(1) X(6) = STDEC(1) X(7) = STRNT(2) X(8) = SINCR(2) X(9) = SASMT(2) X(10) = STDEC(2)
LEARN 7	Overall Baseline Model for Error and Time Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = TIME X(5) = TIME S.D.

TABLE 6-1. DESCRIPTION OF LEARNING MODELS
FOR OSUT SAMPLE (Continued)

<u>Model</u>	<u>Description</u>	<u>Parameters</u>
LEARN 8	Constant Time Learning Model for Error and Time Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = SINCR X(5) = SASMT X(6) = STDEC X(7) = TIME X(8) = TIME S.D.
LEARN 9	Independent Time Model for Error and Time Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = SINCR X(5) = SASMT X(6) = STDEC X(7) = TIME CONSTANT X(8) = P1 TIME X(9) = P2 TIME X(10) = P3 TIME X(11) = TIME S.D.
LEARN 10	Time Improvement Model (Difference) for Error and Time Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = SINCR X(5) = SASMT X(6) = STDEC X(7) = TIME CONSTANT X(8) = TIME STRENGTH X(9) = P1 TIME X(10) = P2 TIME X(11) = P3 TIME X(12) = TIME S.D.
LEARN 11	Time Improvement Model (Reciprocal) for Error and Time Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = SINCR X(5) = SASMT X(6) = STDEC X(7) = TIME CONSTANT X(8) = TIME STRENGTH X(9) = P1 TIME X(10) = P2 TIME X(11) = P3 TIME X(12) = TIME S.D. X(13) = INITIAL STRNT

TABLE 6-1. DESCRIPTION OF LEARNING MODELS
FOR OSUT SAMPLE (Continued)

<u>Model</u>	<u>Description</u>	<u>Parameters</u>
LEARN 12	Trail Time and Strength Model for Error and Time Prediction	X(1) = THRSH(1) X(2) = THRSH(2) X(3) = THRSH(3) X(4) = STRNT(2) X(5) = STRNT(3) X(6) = STRNT(4) X(7) = STRNT(5) X(8) = STRNT(6) X(9) = TIME(1) X(10) = TIME(2) X(11) = TIME(3) X(12) = TIME(4) X(13) = TIME(5) X(14) = TIME(6) X(15) = TIME S.D. X(16) = P1 TIME X(17) = P2 TIME X(18) = P3 TIME

decreases by a certain proportion. LEARN 4 has six parameters: three response thresholds, a learning rate, an asymptote, and a decay multiplier. This model requires measurement of strength on an interval scale. Therefore, the initial strength is not an identifiable parameter of the model, and may be set arbitrarily. For all models requiring only an interval measure of strength, an initial strength value of 5.0 was used.

Models LEARN 1 through LEARN 3 are special cases of LEARN 4 used to test the necessity of the basic learning model. LEARN 3 postulates the same learning function as LEARN 4, but assumes no forgetting during the retention interval. LEARN 2 assumes a constant increment in strength as the result of a training trial, a special case of LEARN 3 with an infinite asymptote for strength. Finally, LEARN 1 assumes no learning; strength is a constant across trials. The only parameters for LEARN 1 are the three response thresholds. LEARN 1 provides a baseline for assessing the improvements brought about by any model over knowledge of response probabilities alone.

LEARN 5 and LEARN 6 are generalizations of LEARN 4, used to test the sufficiency of the basic learning model. LEARN 5 tests the adequacy of the assumptions of the shape of the learning curve by providing a model in which the strength of each trial (after the first) is estimated individually. LEARN 6 is focused on the assumption of all previously discussed models that the learning parameters are the same for all task elements. In LEARN 6, the performance measures are divided into two classes, and parameter values are estimated separately for each. LEARN 6 performs a function similar to that of a cross validation, except that it is the task elements that are divided into two groups rather than the subjects.

LEARN 10 and LEARN 11 are the two basic models that predict both accuracy and speed as a function of strength.

In LEARN 10 performance time is a linear function of the difference between the strength and the strength asymptote. In LEARN 11, performance time is a linear function of the reciprocal of the strength. In addition, for both models estimates are made of the time required for different levels of assistance. LEARN 11 requires measurement of strength on a ratio scale; thus, for this model, initial strength must be estimated rather than set arbitrarily. Both models have six parameters in addition to those represented in LEARN 4. These parameters represent a time constant, the multiple on strength (or its reciprocal), the time required for each of the three levels of assistance, and the standard deviation of the time distribution.

LEARN 7 through LEARN 9 are special cases of both LEARN 10 and LEARN 11. In LEARN 9, time is not affected by strength, except inasmuch as it is affected by the greater assistance that is required when strength is lower. In LEARN 8, time is a constant, but accuracy is represented as in LEARN 4. Finally, in LEARN 7, both accuracy and time are represented as constants. LEARN 7 is used as the baseline for assessing the improvements caused by the more general models.

LEARN 12 is a generalization of both LEARN 10 and LEARN 11. This model is analogous to LEARN 5; that is, strength and time are estimated separately for each trial. This model provides a test of the shape of the learning curve of speed and accuracy. A model analogous to LEARN 6 was not developed, because it provided a more detailed analysis of time than was desired for this study.

For two tasks, Engage Targets Using Precision Fire Techniques, and Communicate Using Visual Signaling Techniques, individual task elements were practiced several times during one training trial. For these tasks, the basic models had to be revised to account for the added practice, which included

up to twenty-four trials. The changes in the models affected the number of parameters, and required ratio-scale measurement of strength for some models. The number of parameters for both versions of the models is shown in Table 6-2.

Necessity of a particular model was tested by comparing the goodness-of-fit measure for that model to the goodness-of-fit for the models directly above it in the model hierarchy (Figure 6-1). Thus, for example, LEARN 4 would be compared to LEARN 3, and LEARN 12 would be compared to two other models, LEARN 10 and LEARN 11. The value of the chi-square statistic for a model indicates the improvement in prediction brought about by adding the parameter in the more complex model to the simpler model. Thus, if LEARN 4 performs significantly better than LEARN 3, then adding the forgetting parameter significantly improves the quality of the model. This comparison may also be viewed as a test of the hypothesis that the forgetting proportion is equal to zero.

6.3.2 Models of accuracy - The first six acquisition models predict a probability distribution over levels of assistance. An iterative optimization routine was used to find the parameter values that maximized the likelihood of the obtained data according to the model. The resulting models, incorporating data from the model-development group, are discussed in this section. The principal model of interest, which combines geometric increase of strength to an asymptote and decay during the retention interval, has six parameters: three thresholds, a learning rate, an asymptote, and a forgetting proportion. These parameters are difficult to interpret and cannot be compared across tasks. Therefore, the parameters were transformed into six other parameters.

An indication of the initial strength level is given by subtracting the threshold for correct performance

TABLE 6-2. NUMBER OF PARAMETERS REQUIRED FOR
DIFFERENT VERSIONS OF LEARNING MODELS

		All Tasks Except <u>PRECFIRE and SIGNALS</u>	<u>PRECFIRE AND SIGNALS</u>
LEARN	1	3	3
LEARN	2	4	4
LEARN	3	5	5
LEARN	4	6	7
LEARN	5	8	32
LEARN	6	10	11
LEARN	7	5	5
LEARN	8	8	9
LEARN	9	11	12
LEARN	10	12	13
LEARN	11	13	13
LEARN	12	18	66

from the initial strength, which was set to be 5.0. This factor is the standard score corresponding to the proportion of correct responses on the first trial. The standardized initial strength is the first of the transformed parameters. The second parameter represents the amount of learning that occurs over the first five trials. This parameter is measured by the difference between the strength after the fifth training trial, and the (unstandardized) initial strength. The learning amount measures the improvement in performance in units of the standard deviation of the strength distribution. The third parameter, the learning rate, is unchanged from the original parameterization. It represents the degree of curvature in the learning curve. The fourth parameter, the decay amount, measures the predicted decrement of performance which occurs during the retention interval in standard units. The final two parameters measure the effectiveness of the first two levels of assistance (the third level is completely effective). They are measured by the difference between adjacent response thresholds.

The maximum-likelihood estimates for these parameters, based on data from the model-development group, are shown in Table 6-3 for the eight tasks. Most of the model parameters show considerable variation across tasks. For example, SIGNALS, STARTANK, and LOADMG started with fairly high strength, while the two prepare-to-fire tasks started with considerably lower strength. The degree of learning tended to compensate for the initial differences, with the greatest learning occurring for the prepare-to-fire checks. The learning rate and decay amount also show differences between tasks. Perhaps the most noticeable difference is that no decay occurs for SIGNALS, while the average decay for the other tasks is nearly two-thirds of a standard unit.

TABLE 6-3. PARAMETER ESTIMATES FOR OSUT MODEL OF PERFORMANCE ACCURACY
(LEARN 4) FOR MODEL DEVELOPMENT GROUP

	STANDARDIZED		LEARNING AMOUNT	LEARNING RATE	DECAY AMOUNT	LEVEL 1		LEVEL 2	
	INITIAL STRENGTH					EFFECTIVENESS		EFFECTIVENESS	
LOADMG	1.0322		1.5816	0.6075	0.5340	0.0985		1.6157	
STARTANK	1.2974		1.2816	0.0781	1.0404	0.1495		00	
STOPTANK	0.8026		1.7542	0.6367	0.5595	0.1277		2.0343	
GUNNERPF	-0.4183		2.3464	0.6236	0.6427	0.0342		2.9796	
LOADERPF	-0.8238		2.9759	0.6089	0.9002	0.0909		3.2605	
PRECFIRE	0.4211		1.1216	0.1652	0.6291	0.0095		0.3273	
RADIOMSG	0.7258		1.1078	0.2103	0.2604	0.0230		1.1944	
SIGNALS	1.4342		0.8819	0.2982	0.0000	0.0166		0.8991	

One area in which the tasks are generally similar is in the effectiveness of the different levels of assistance. The first level of assistance was generally not very effective, leading to an increase of strength of less than one-fifth of a standard unit. The second level of assistance was much more effective, and led to increases of over one standard unit for six of the tasks. In fact, for STARTANK, the third level of assistance was never required in the model-development group. The results represent the fact that if assistance was required, it was very seldom the case that the first level of assistance was sufficient, or that the third level of assistance was required.

Model fit was measured by twice the negative log likelihood of the data given the model. Lower values of this measure indicate better fit of the model. This measure of fit was chosen because differences in negative log likelihood between nested models are distributed according to a chi-square distribution with degrees of freedom equal to the difference in the number of parameters. Thus, a statistical test of the ability of a more complex model to offer a better account of data than a simpler model was possible. The fit of the accuracy models to the data from the model-development group is shown in Table 6-4. The results indicate that the general acquisition and forgetting model (LEARN 4) is significantly better than the simpler models tested for all tasks except RADIOMSG and SIGNALS, in which the forgetting proportion was not significantly greater than zero. Overall, there is a very good indication of the necessity of the model.

For four of the tasks, LOADMG, STARTANK, STOPTANK, and RADIOMSG, the shape of the learning curve could not be significantly improved upon by estimating the strength of each trial separately. For the remaining four tasks, there was a

TABLE 6-4. TWICE NEGATIVE LOG LIKELIHOOD OF DATA FROM MODEL DEVELOPMENT GROUP - OSUT SAMPLE

TASK	LEARN 1		LEARN 2		LEARN 3		LEARN 4		LEARN 5		LEARN 6	
	CONSTANT STRENGTH	3 Parameters	CONSTANT INCREMENT	4 Parameters	CONSTANT RATE	5 Parameters	FOREGETTING	6 Parameters (7)	TRIAL STRENGTH	8 Parameters (32)	TASK ELEMENT GROUPS*	10 Parameters (11)
LOADMG	1221.452		1080.560 $\chi^2(1)=140.892^{***}$		1016.920 $\chi^2(1)=63.640^{***}$		1009.648 $\chi^2(1)=7.272^{**}$		1009.374 $\chi^2(2)=0.274$		991.888 $\chi^2(4)=17.76^{**}$	
STARTTANK	1268.658		1250.784 $\chi^2(1)=17.874^{***}$		1229.722 $\chi^2(1)=21.062^{***}$		1201.818 $\chi^2(1)=27.904^{***}$		1199.242 $\chi^2(2)=2.576$		1080.75 $\chi^2(4)=121.068^{***}$	
STOPTANK	1525.468		1313.686 $\chi^2(1)=211.782^{***}$		1216.576 $\chi^2(1)=97.11^{***}$		1207.146 $\chi^2(1)=9.43^{**}$		1206.560 $\chi^2(2)=0.586$		1175.618 $\chi^2(4)=31.528^{***}$	
GUNNERPF	12019.7		9709.78 $\chi^2(1)=2309.92^{***}$		8450.36 $\chi^2(1)=1259.42^{***}$		8330.74 $\chi^2(1)=119.62^{***}$		8314.38 $\chi^2(2)=16.36^{***}$		8326.82 $\chi^2(4)=3.92$	
LOADERPF	2166.50		1631.246 $\chi^2(1)=535.254^{***}$		1323.178 $\chi^2(1)=308.068^{***}$		1290.99 $\chi^2(1)=32.188^{***}$		1274.378 $\chi^2(2)=16.61^{***}$		1278.794 $\chi^2(4)=12.196^*$	
PRECMIRE	5654.6		5103.36 $\chi^2(1)=551.24^{***}$		4996.44 $\chi^2(1)=106.92^{***}$		4922.74 $\chi^2(2)=73.7^{**}$		4852.58 $\chi^2(25)=70.16^{***}$		4840.68 $\chi^2(4)=82.06^{***}$	
RAI MSG	2353.58		2234.36 $\chi^2(1)=119.22^{***}$		2223.22 $\chi^2(1)=11.14^{***}$		2220.66 $\chi^2(1)=2.56$		2219.68 $\chi^2(2)=0.98$		2199.98 $\chi^2(4)=20.68^{***}$	
SIGNALS	2322.22		2187.26 $\chi^2(1)=134.96^{***}$		2149.24 $\chi^2(1)=38.02^{***}$		2149.24 $\chi^2(2)=0.0$		2101.04 $\chi^2(25)=48.2^{**}$		2146.74 $\chi^2(4)=2.5$	

* $p < .05$ ** $p < .01$ *** $p < .001$

significant improvement in fit between LEARN 4 and LEARN 5. The interpretation of this improvement is difficult for PRECFIRE and SIGNALS, because differences in the shape of the curve are confounded with task-element differences (because different task elements receive different numbers of training trials). However, the improvement in fit for GUNNERPF and LOADERPF unambiguously indicates that a more general learning function is required for these tasks. This difference occurs for the two tasks in which learning is the greatest, and hence, the test of the shape of the learning curve is the most severe. Further research should investigate whether other simple functions may be found to characterize the shape of the learning curve.

For six of the eight tasks, there were significant differences between task elements as indicated by the improvement in performance of LEARN 6 over LEARN 4. Although these differences are significant, they are not particularly large. With the exception of STARTANK, the improvement from LEARN 4 to LEARN 6 (4 d.f.) is less than 15.5% (for RADIOMSG) of the difference between LEARN 1 and LEARN 4 (3 d.f.). For STARTANK, the difference between task elements is substantial, and the difference between LEARN 4 and LEARN 6 is 181.1% of that between LEARN 1 and LEARN 4. Thus, although there are significant differences between task elements, a model which ignores these differences may provide an acceptable account of acquisition data for most tasks. Nevertheless, the analysis of task-element differences is a good area for further study. The use of an acquisition model should provide powerful tools for investigations of the causes of task-element differences.

Overall, the general learning and forgetting model provides a good account for the training data from the OSUT soldiers. However, two areas that require further investigation have been identified. These areas involve the shape of

the learning curve and differences in the values of model parameters between task elements.

6.3.3 Models of speed and accuracy - Models LEARN 7 through LEARN 12 predict the total performance time for a task in addition to the probability distribution over assistance levels for individual task elements. The results of goodness-of-fit tests for these six models are presented in Table 6-5. The improvement from LEARN 7 to LEARN 8 is equivalent to that from LEARN 1 to LEARN 4, and is significant in all cases. LEARN 9 adds three parameters to LEARN 8. The parameters represent the time required for each of the three levels of assistance. Addition of these parameters greatly increases the ability to predict performance time.

The major models of interest for prediction of performance time are LEARN 10 and LEARN 11. These two models differ in that LEARN 10 postulates that performance time is a linear function of the difference between the current strength and the strength asymptote, while LEARN 11 postulates that performance time is a linear function of the reciprocal of strength. LEARN 11 has an extra parameter because the initial strength must be estimated from the data. Table 6-5 indicates that LEARN 10 and LEARN 11 provide approximately equal improvements over LEARN 9 which are significant for all tasks except PRECFIRE. Thus, speed increases as a regular function of strength, but the nature of this function is not certain.

LEARN 12 provides a test of the LEARN 10 and LEARN 11 in a manner similar to the test that LEARN 5 provides for LEARN 4. The results are similar, as well. For four tasks, GUNNERPF, LOADERPF, PRECFIRE, and SIGNALS, LEARN 12 predicts

TABLE 6-5. TWICE NEGATIVE LOG LIKELIHOOD OF DATA FROM MODEL DEVELOPMENT GROUP - OSUT SAMPLE

TASK	LEARN 7		LEARN 8		LEARN 9		LEARN 10		LEARN 11		LEARN 12	
	BASELINE TIME	5 Parameters	CONSTANT TIME	8 Parameters (9)	INDEPENDENT TIME	11 Parameters (12)	DIFFERENCE MODEL	12 Parameters (13)	RECIPROCAL MODEL	13 Parameters	TRIAL TIME & STRENGTH	18 Parameters (66)
LOADMG		3839.86	3628.06 $\chi^2(3)=211.80^{***}$		3223.5 $\chi^2(3)=404.56^{***}$		3214.64 $\chi^2(1)=8.86^{**}$		3215.3 $\chi^2(2)=8.2^*$		3213.68 $\chi^2(6)=0.96$ $\chi^2(5)=1.62$	
STARTANK		4177.46	4110.62 $\chi^2(3)=66.84^{***}$		3970.42 $\chi^2(3)=140.2^{***}$		3879.86 $\chi^2(1)=90.56^{***}$		3868.12 $\chi^2(2)=102.3^{***}$		3863.36 $\chi^2(6)=16.5^*$ $\chi^2(5)=4.76$	
STOPTANK		5297.70	4979.36 $\chi^2(3)=318.34^{***}$		4673.28 $\chi^2(3)=306.08^{***}$		4571.40 $\chi^2(1)=101.88^{***}$		4572.62 $\chi^2(2)=100.66^{***}$		4566.96 $\chi^2(6)=4.44$ $\chi^2(5)=5.66$	
GUNNERPF		16516.76	12827.82 $\chi^2(3)=3688.94^{***}$		12605.50 $\chi^2(3)=222.32^{***}$		12482.38 $\chi^2(1)=123.12^{***}$		12492.98 $\chi^2(2)=112.52^{***}$		12452.56 $\chi^2(6)=32.86^{***}$ $\chi^2(5)=39.08^{***}$	
LOADERPF		5342.36	4466.86 $\chi^2(3)=875.5^{***}$		4289.56 $\chi^2(3)=177.3^{***}$		4260.8 $\chi^2(1)=28.76^{***}$		4268.96 $\chi^2(2)=20.6^{***}$		4240.30 $\chi^2(6)=20.5^{**}$ $\chi^2(5)=28.66^{***}$	
PRECFIRE		7737.46	7005.62 $\chi^2(4)=731.84^{***}$		6985.62 $\chi^2(3)=20^{***}$		6982.78 $\chi^2(1)=2.84$		6983.72 $\chi^2(1)=1.9$		6870.64 $\chi^2(53)=112.14^{***}$ $\chi^2(53)=113.08^{***}$	
RADIONSG		4211.14	4078.22 $\chi^2(3)=132.92^{***}$		4020.86 $\chi^2(3)=57.36^{***}$		4014.06 $\chi^2(1)=6.8^{**}$		4013.26 $\chi^2(2)=7.6^*$		4009.86 $\chi^2(6)=4.2$ $\chi^2(5)=3.4$	
SIGNALS		4650.94	4477.88 $\chi^2(4)=173.06^{***}$		4376.38 $\chi^2(3)=101.5^{***}$		4332.88 $\chi^2(1)=43.5^{***}$		4335.5 $\chi^2(1)=40.88^{***}$		4260.64 $\chi^2(53)=72.24^*$ $\chi^2(53)=74.86^*$	

* $p < .05$ ** $p < .01$ *** $p < .001$

the response probabilities and speeds better than both LEARN 11 and LEARN 10. For one task, STARTANK, LEARN 12 performs better than LEARN 10, but not better than LEARN 11. For the remaining tasks, the added parameters of LEARN 12 do not improve on either of the simpler models. As was true in the accuracy models, differences in the shape of the strength function are confounded with task-element differences for PRECFIRE and SIGNALS. The results for GUNNERPF and LOADERPF can be interpreted to indicate a failure of the theoretical learning function to account for the shape of the empirical learning curve, although this difference is relatively minor.

We are left with little evidence to choose between the difference model and the reciprocal model of performance speed. One criterion for choice between these models is the ability of the model to predict the results of the cross-validation sample. If those results are approximately equal, we will chose the difference model because it is somewhat simpler. It should be noted that the choice between these and other models is influenced by the functional form of the learning curve. Any change in the learning curve may need to be accompanied by a corresponding change in the relationship between strength and speed.

6.3.4 Application of models to cross-validation sample - The importance of individual differences in parameter estimates was assessed by applying the model with parameters developed from the model-development group to data from the cross-validation group. The model from the first half of subjects was used as a starting point for the optimization routine, which then found maximum-likelihood estimates for model parameters within the cross-validation group.

The initial values of the parameters, estimated from data from subjects in the model-development group, provide a model of acquisition and retention with no free parameters. Therefore, the reduction in negative log-likelihood from the initial parameters to the maximum-likelihood parameter estimates of the parameters for the cross-validation group has a chi-square distribution with degrees of freedom equal to the number of parameters in the model. A significant value of chi-square can be interpreted as indicating a difference in the parameter values between the two groups of subjects.

The initial and final values of the negative log-likelihood are presented in Table 6-6, along with the associated chi-square values for five of the models: the two baseline models, LEARN 1 and LEARN 7, and the three models of interest, LEARN 4, LEARN 10, and LEARN 11. For two of the accuracy models, LOADMG and STARTANK, the maximum likelihood parameters did not produce a significant improvement over the parameters estimated from an independent sample. For the other six accuracy models, as well as all of the models of both speed and accuracy, the maximum likelihood parameter estimates offered a significant improvement over the estimates from the model-development group.

The fit of the baseline models indicates that some of the discrepancy between parameter values may be attributed to differences in the overall response frequencies. These differences were significant for five of the eight tasks. In addition, differences in the distribution for STARTANK required a modification of the analysis for this task. In the model-development group there were no cases in which level 3 assistance was required. Thus, thresholds were set so that the likelihood of level 3 assistance was zero. In the cross-validation group, there were several instances of level 3 assistance, clearly an impossible event under the parameters esti-

TABLE 6-6. INITIAL AND FINAL BY LIKELIHOOD VALUES FOR CROSS-VALIDATION TESTS

TASK	LEARN 1		LEARN 4		LEARN 7		LEARN 10		LEARN 11	
	CONSTANT STRENGTH	3 Parameters	FORGETTING	6 Parameters (7)	BASELINE TIME	5 Parameters	DIFFERENCE MODEL	12 Parameters (13)	RECIPROCAL MODEL	13 Parameters
LOADMG	1194.064		1014.734		3790.24		3379.42		3378.72	
	1189.63		1009.102		3781.56		3291.72		3291.84	
	$\chi^2(3)=4.434$		$\chi^2(6)=5.632$		$\chi^2(5)=8.68$		$\chi^2(12)=87.7***$		$\chi^2(13)=86.88***$	
STARTANK	1264.49		1164.138		4078.34		3783.76		3774.4	
	1264.38		1157.706		4074.00		3761.88		3752.66	
	$\chi^2(2)=0.11$		$\chi^2(5)=6.432$		$\chi^2(4)=4.34$		$\chi^2(11)=21.88*$		$\chi^2(12)=21.74$	
STOPTANK	1379.164		1172.084		5149.00		4636.06		4629.84	
	1372.924		1159.404		5138.92		4584.42		4578.34	
	$\chi^2(3)=6.24$		$\chi^2(6)=12.68*$		$\chi^2(5)=10.08$		$\chi^2(12)=51.64***$		$\chi^2(13)=51.5***$	
GUNNERPF	12032.04		8297.38		16630.78		12525.90		12546.44	
	12023.70		8267.18		16615.02		12485.94		12501.48	
	$\chi^2(3)=8.34*$		$\chi^2(6)=30.2***$		$\chi^2(5)=15.76**$		$\chi^2(12)=39.96***$		$\chi^2(13)=44.96***$	
LOADERPF	2342.98		1448.636		5653.22		4562.18		4571.3	
	2313.2		1395.46		5577.92		4449.58		4458.9	
	$\chi^2(3)=29.78***$		$\chi^2(6)=99.874***$		$\chi^2(5)=75.3***$		$\chi^2(12)=112.6***$		$\chi^2(13)=112.4***$	
PRECIFIRE	5332.92		4629.52		7299.78		6573.16		6574.18	
	5256.52		4548.72		7191.4		6454.82		6454.84	
	$\chi^2(3)=76.4***$		$\chi^2(7)=80.8***$		$\chi^2(5)=108.38***$		$\chi^2(13)=118.34***$		$\chi^2(13)=119.34***$	
RADTOMSG (Group 2)	2276.26		2081.90		4267.20		4025.34		4024.66	
	2264.82		2057.86		4138.40		3864.88		3860.42	
	$\chi^2(3)=11.44**$		$\chi^2(6)=24.04***$		$\chi^2(5)=128.8***$		$\chi^2(12)=160.46***$		$\chi^2(13)=164.24***$	
SIGNALS (Group 1)	1929.87		1850.15		4204.52		4025.18		4029.02	
	1916.752		1829.744		4190.48		3988.76		3991.40	
	$\chi^2(3)=13.118**$		$\chi^2(7)=20.406**$		$\chi^2(5)=14.04**$		$\chi^2(13)=36.42***$		$\chi^2(13)=37.62***$	

*p < .05

**p < .01

***p < .001

mated for the model-development group. Thus, the initial parameters correspond to an infinite negative log-likelihood measure. To avoid this problem, the second and third assistance levels were combined in the models for STARTANK. Hence, these models have one fewer parameter than corresponding models for other tasks.

Even in other tasks, level 1 and level 3 prompts are rare. Given the low probability of these levels of assistance, the amount of data is probably not sufficient to provide stable estimates of the associated thresholds. Examination of Table 6-6 indicates that for three tasks, STOPTANK, PRECFIRE, and SIGNALS, the chi-square value for LEARN 4 is only slightly greater than that for LEARN 1. Thus, in five of the tasks, there is evidence that the parameter estimates are reasonably close for the different samples for the accuracy models.

Differences in the models predicting both speed and accuracy are more substantial and cannot be attributed entirely to differences in the baseline model for any tasks. In addition, the differences do not distinguish between the difference model and the reciprocal model of performance speed. Individual differences, therefore, have a greater impact on the models of performance speed than they do on performance accuracy.

Even if there are significant individual differences in the values of model parameters, parameters estimated from one set of subjects may provide a reasonable account for the behavior of another set of subjects. More specifically, the reduction from the baseline model to the initial parameter values may be nearly as great as the reduction to the optimal parameter values even if the additional improvement obtained by going to the optimal values is significant. Table 6-7

TABLE 6-7. IMPROVEMENT FROM BASELINE TO INITIAL
PARAMETER VALUES AS A PERCENTAGE OF IMPROVEMENT
TO OPTIMAL PARAMETERS

	<u>LEARN 4</u>	<u>LEARN 10</u>	<u>LEARN 11</u>
LOADMG	96.88%	82.10%	82.26%
STARTANK	93.97%	92.99%	93.23%
STOPTANK	94.06%	90.69%	90.81%
GUNNERPF	99.20%	99.03%	96.48%
LOADERPF	94.21%	90.02%	89.96%
PRECFIRE	88.58%	83.93%	83.80%
RADIOMSG	88.38%	41.34%	40.92%
SIGNALS	76.55%	81.95%	81.10%

shows that with two exceptions, the models with parameters estimated from the model-development group produces over three-quarters of the reduction in negative log-likelihood as the optimal model parameters.

The analysis of the cross-validation sample has identified significant individual differences in the values of model parameters. Further investigations should seek a simple way to characterize these differences. However, the parameter estimates taken from another sample account for a large proportion of the improvement in prediction that is produced by the model with optimal parameter estimates. Thus, the model may still have considerable utility, even if individual differences are not addressed.

6.3.5 Analysis of the combined sample - Data from the two groups were combined in order to obtain more precise parameter estimates. The resulting parameter estimates are shown in Table 6-8 and Table 6-9 for LEARN 4 and LEARN 10, respectively. The parameter estimates from the combined sample do not differ in any important respect from those obtained from the model-development group.

6.4 Results for Operational Unit Data

Data in the operational unit consist of errors and performance times for a single performance of the eight tasks. The models are concerned principally with skill retention, although some models make allowances for practice occurring during the retention interval. The models tested and the results are described in the following subsections.

TABLE 6-8. PARAMETER ESTIMATES FOR OSUT MODEL OF PERFORMANCE ACCURACY
(LEARN 4) FOR ALL SUBJECTS

STANDARDIZED INITIAL STRENGTH	LEARNING AMOUNT	LEARNING RATE	DECAY AMOUNT	LEVEL 1		LEVEL 2	
				EFFECTIVENESS		EFFECTIVENESS	
LOADMG	1.0692	0.5752	0.6174	0.1157		1.4966	
STARTANK	1.2335	0.1674	1.0860	0.1547		1.6574	
STOPTANK	0.8868	0.5672	0.6097	0.1326		2.1484	
GUNNERPF	-0.4131	0.6015	0.5826	0.0356		2.8384	
LOADERPF	-0.8280	0.6008	1.1074	0.1166		2.6879	
PRECIFIRE	0.4515	0.1549	0.6887	0.0162		0.2585	
RADIOMSG	0.6710	0.1678	0.2688	0.0194		1.3392	
SIGNALS	1.5339	0.2732	0.0126	0.0178		0.9634	

TABLE 6-9. PARAMETER ESTIMATES FOR OSUT MODEL OF PERFORMANCE SPEED AND ACCURACY (LEARN 10) FOR ALL SUBJECTS

TASK	STANDARDIZED INITIAL STRENGTH	LEARNING AMOUNT	LEARNING RATE	DECAY AMOUNT	LEVEL 1 EFFECT	LEVEL 2 EFFECT	TIME CONSTANT	STRENGTH MULTIPLE	LEVEL 1 TIME	LEVEL 2 TIME	LEVEL 3 TIME	TIME S.D.
LOADMG	1.0714	1.6117	0.5501	0.6456	0.1157	1.4960	0.9915	0.2164	8.9704	5.7249	17.0552	8.2159
STARTANK	1.1682	1.0350	0.4507	0.4631	0.1485	1.6414	3.1584	6.0741	17.4657	15.2365	5.1334	32.7704
STOPTANK	0.8790	1.5476	0.6482	0.2964	0.1324	2.1516	3.5832	4.2756	18.5412	5.0778	0.0	29.8616
GUNNERFF	-0.4117	2.3708	0.5948	0.5601	0.0356	2.8370	5.3299	4.3164	2.7872	0.0	52.4029	104.515
LOADERFF	-0.8147	3.1263	0.5742	1.2103	0.1164	2.6788	4.4236	2.1919	3.9138	0.0	22.0008	23.1395
PRECFFIRE	0.4613	1.0726	0.1453	0.6188	0.0163	0.2582	0.8987	0.1672	0.0001	0.7524	0.1161	11.4405
RADIONMSG	0.6600	1.2104	0.2276	0.1978	0.0194	1.3398	7.7022	5.2644	0.0	26.4781	0.0	69.4243
SIGNALS	1.5643	0.7505	0.2110	0.2159	0.0178	0.9592	2.1222	1.5665	8.1678	0.0	4.0338	26.1172

6.4.1 Hierarchy of models - Parameters were estimated for four retention models which predicted the distribution of assistance levels as a function of time since training. The models will be referred to as RETEN 1 through RETEN 4. The fit of the models of accuracy and a regression analysis of performance time indicated that it would not be appropriate to test models predicting both speed and accuracy. The models are increasingly general, with each model being a special case of the model following it.

The models are described briefly in Table 6-10. The basic retention model presented in Equation 6 is represented by RETEN 4. The value of the integral in Equation 6 is approximated numerically using Simpson's rule. RETEN 4 has eight parameters: three thresholds, the initial strength, the strength decay rate (the exponent of the decay function), the fragility decay rate (the multiplier on time in the decay function), the learning rate, and the strength asymptote.

The other models are all special cases of RETEN 4. In RETEN 3, practice is assumed to have no impact on retention; that is, the learning rate is assumed to be zero. RETEN 3 has six parameters corresponding to those of RETEN 4, except that the learning rate and strength asymptote are absent. RETEN 2 makes the further restriction that fragility is a constant. In this case, decay of strength is exponential. RETEN 2 requires five parameters: three thresholds, an initial strength, and a strength decay rate. RETEN 1 is the baseline, constant strength model. This three-parameter model is the same as LEARN 1.

6.4.2 Models of accuracy - The fit of the retention models is shown in Table 6-11, which presents twice the negative log-likelihood of the data for the maximum-likelihood estimates for the models. In general, it was very difficult

TABLE 6-10. DESCRIPTION OF RETENTION MODELS
FOR OPERATIONAL UNIT SAMPLE

MODEL	DESCRIPTION	PARAMETERS
RETEN 1	Constant Strength Model Model for Error Prediction	X(1)=THRSH (1) X(2)=THRSH (2) X(3)=THRSH (3)
RETEN 2	Exponential Decay Model for Error Prediction	X(1)=THRSH (1) X(2)=THRSH (2) X(3)=THRSH (3) X(4)=STDEC X(5)=SINIT
RETEN 3	Power Function Decay Model for Error Prediction	X(1)=THRSH (1) X(2)=THRSH (2) X(3)=THRSH (3) X(4)=STDEC X(5)=FRDEC X(6)=SINIT
RETEN 4	Power Function Decay Model with Practice for Error Prediction	X(1)=THRSH (1) X(2)=THRSH (2) X(3)=THRSH (3) X(4)=STDEC X(5)=FRDEC X(6)=RATE X(7)=SASMT X(8)=SINIT

to obtain stable estimates of model parameters. Neither of the simple decay models provided any improvement in prediction over the constant strength model. The most general model, RETEN 4, provided significant improvement over the simpler models for two tasks, GUNNERPF and LOADERPF. However, the optimal parameter estimates for these tasks indicate very high decay of both strength and fragility, combined with a low learning rate. Since this configuration of parameter values leads to a retention curve which is essentially flat over the retention interval sampled in this study, the significance of the improvement is probably spurious, caused by inaccuracies in the numerical approximation to the integral used in RETEN 4. Thus, it is reasonable to conclude that performance was constant over the retention interval used in the operational unit sample.

6.4.3 Models of speed and accuracy - The flat performance over time exhibited by the accuracy data suggests that there would be little to gain from testing models of performance speed, unless there is some evidence that performance speed is related to time since training. Consequently, before any models were developed, performance time was correlated with time since training. Since we expect that retention would be described by a power function, logarithms were taken of both variables. The results, displayed in Table 6-12, indicate that, except for LOADMG, correlations between performance time and time since training were very close to zero. In addition, there was a significant negative correlation for RADIOMSG. Consequently, models predicting both speed and accuracy were not tested for the operational unit data.

6.4.4 Analysis of the combined sample - Since acceptable fits were not found for the retention models of the operational unit data, an analysis of the cross-validation sample was not

TABLE 6-11. TESTS OF RETENTION MODELS FOR MODEL DEVELOPMENT GROUP

TASK	RETEN 1		RETEN 2		RETEN 3		RETEN 4	
	CONSTANT STRENGTH	3 Parameters	EXPONENTIAL DECAY	5 Parameters	POWER DECAY	6 Parameters	PRACTICE & DECAY	8 Parameters (7)
LOADMG	625.548		625.004 $\chi^2(2)=0.544$		625.004 ^a $\chi^2(1)=0.00$		625.004 ^a $\chi^2(2)=0.00$	
STARTANK	1238.544		1248.302 $\chi^2(2)=0.242$		1238.302 ^a $\chi^2(1)=0.00$		1238.302 ^a $\chi^2(2)=0.00$	
STOPTANK	933.692		933.692 $\chi^2(2)=0.00$		933.692 $\chi^2(1)=0.00$		930.550 ^b $\chi^2(2)=3.142$	
GUNNERPF	2445.30		2445.3 $\chi^2(2)=0.00$		2445.3 $\chi^2(1)=0.00$		2419.52 ^b $\chi^2(2)=25.78^{**}$	
LOADERPF	1054.776		1054.776 $\chi^2(2)=0.00$		1054.776 $\chi^2(2)=0.00$		1043.428 ^b $\chi^2(2)=11.348^*$	
PRECFIRE	1118.34		1118.34 $\chi^2(2)=0.00$		1118.34 $\chi^2(1)=0.00$		1118.16 ^b $\chi^2(2)=0.18$	
RADIOMSG	804.024		804.024 $\chi^2(2)=0.00$		804.024 $\chi^2(1)=0.00$		801.208 $\chi^2(2)=2.816$	
SIGNALS	213.49		213.49 $\chi^2(2)=0.00$		213.49 $\chi^2(1)=0.00$		212.294 $\chi^2(2)=1.196$	

^aConvergence at lower value not obtained. Values represent upper bound of negative likelihood measure.

^bConvergence not obtained. Values represent lowest obtained values of negative likelihood measure.

*p < .01

**p < .001

TABLE 6-12. CORRELATION BETWEEN PERFORMANCE
TIME AND TIME SINCE OSUT GRADUATION
IN OPERATIONAL UNIT SAMPLE

<u>TASK</u>	<u>N</u>	<u>CORRELATION</u>
LOADMG	116	.35**
STARTANK	58	-.02
STOPTANK	115	-.01
GUNNERPF	116	-.05
LOADERPF	116	-.07
PRECFIRE	113	.14
RADIOMSG	115	-.19*
SIGNALS	116	.003

*p < .05

**p < .01

performed. However, it may be that the added power gained from combining the samples would be sufficient to uncover a relationship between performance and time since training that could not be found using the data from the model-development group alone. Maximum-likelihood estimates were found for the parameters of the four retention models for the combined sample. The results, shown in Table 6-13, parallel those from the model-development group. Again, there is no evidence to reject the constant strength model within the retention interval investigated.

6.5 Summary

The goodness-of-fit of the learning and retention models proposed in section 4.0 were assessed by comparing their predictions to the data collected from two samples of soldiers, a sample from a unit in training, and a sample from an operational unit. The learning model provided a good account of the speed and accuracy data from the soldiers in training. However, deviations of the data from the model suggest several areas requiring further investigation. These areas include differences between task elements, the shape of the learning curve, and individual differences in learning and retention. The deviations from the model were greater in the models predicting both speed and accuracy than they were in models predicting accuracy alone. The modeling framework developed in this project provides powerful tools for investigating the issues raised by this preliminary analysis.

Performance in the operational unit was constant over the retention interval investigated. None of the retention models investigated provided any improvement over a baseline model assuming constant strength over the retention interval. Constant performance was exhibited when performance was measured by either accuracy or speed.

TABLE 6-13. TESTS OF RETENTION MODELS FOR ALL SUBJECTS IN OPERATIONAL UNIT

TASK	RETEN 1		RETEN 2		RETEN 3		RETEN 4	
	CONSTANT STRENGTH 3 Parameters		EXPONENTIAL DECAY 5 Parameters		POWER DECAY 6 Parameters		PRACTICE & DECAY 8 Parameters	
LOADMG	1287.936		1285.538 $\chi^2(2)=2.398$		1285.538 ^a $\chi^2(1)=0.0$		1285.538 ^a $\chi^2(2)=0.0$	
STARTANK	2410.88		2410.48 $\chi^2(2)=0.4$		2410.48 ^a $\chi^2(1)=0.0$		2410.48 ^a $\chi^2(2)=0.0$	
STOPTANK	1765.922		1765.862 $\chi^2(2)=0.06$		1765.862 $\chi^2(1)=0.0$		1765.862 ^a $\chi^2(2)=0.0$	
GUNNERPF	5117.26		5117.26 $\chi^2(2)=0.0$		5117.26 $\chi^2(1)=0.0$		5101.42 ^b $\chi^2(2)=15.84^*$	
LOADERPF	2023.50		2023.50 $\chi^2(2)=0.0$		2023.50 $\chi^2(1)=0.0$		2009.16 ^b $\chi^2(2)=14.34^*$	
PRECFIRE	2211.58		2211.58 $\chi^2(2)=0.0$		2211.58 $\chi^2(1)=0.0$		2210.30 ^b $\chi^2(2)=1.28$	
RADIOMSG	1608.046		1608.046 $\chi^2(2)=0.0$		1608.046 $\chi^2(1)=0.0$		1606.53 $\chi^2(2)=1.516$	
SIGNALS	315.540		315.540 $\chi^2(2)=0.0$		315.540 $\chi^2(1)=0.0$		315.470 $\chi^2(2)=5.07$	

^aConvergence at lower value not obtained. Values represent upper bound of negative likelihood measure.

^bConvergence not obtained. Values represent lowest obtained values negative likelihood measure.

*p < .01

7.0 EXPERIMENTAL SIMULATIONS

The general SAINT models described in section 3.0 were modified to represent specific details of the two samples for which data were collected. Since the data from the OSUT soldiers allowed for relatively precise estimation of model parameters, the specific models for this group provided a framework for representing these parameters, along with details about the data collection procedure within the SAINT models. However, good fit for the psychological models was not obtained for the operational unit sample. Consequently, the model of this sample has been made somewhat more general, so that the implications of various values of model parameters can be determined. Both of these models show how the basic SAINT models can be modified to represent specific situations.

7.1 OSUT Models

In order to represent the details of the OSUT data collection procedure, the basic models must be modified to account for the formal training that occurs before the five training trials are conducted as a part of this study, and for the activities that occur during the retention interval between the fifth and sixth trial. Figure 7-1 shows a framework that represents initial training and activities during the retention interval.

Initial training, and the time between initial training and the first training trial of the study, are nominally represented in Task 81, FRML TRN, and Task 82, TRN WAIT. Since no data were collected regarding these tasks, they do not effect the learning or retention of the procedures. Thus, for all practical purposes, training may be seen to begin with the first training trial. These tasks give the model user the capability to modify characteristics of the formal training,

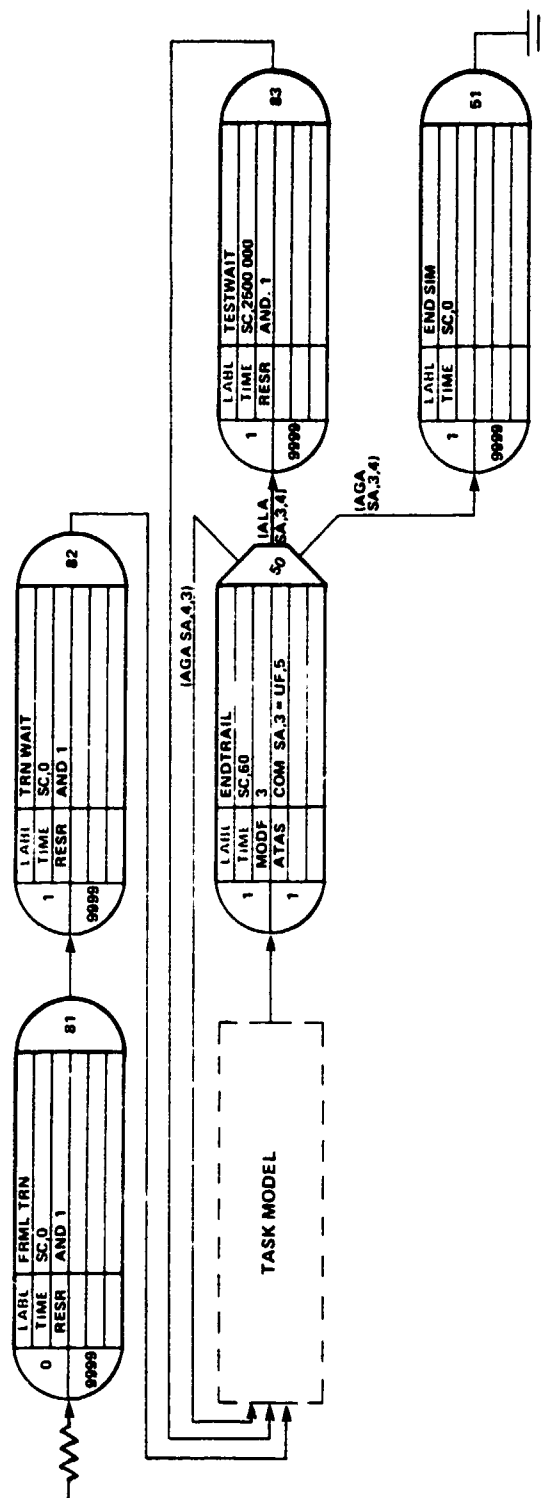


Figure 7-1
FRAMEWORK FOR OSUT SAMPLE SIMULATION

should this become important for a particular application of the model. Task 83, TESTWAIT, represents the delay between the fifth and sixth trials. In reality, this time is filled with practice on the task of interest, as well as on interfering tasks. However, inclusion of all these activities into the model would lead to a representation of the retention interval that is much more complex than could be validated by the data. A more complex representation of the retention interval would be required if the retention interval were different for different soldiers. Differences in the retention interval between soldiers for the OSUT data were very minor.

7.1.1 Implementing the models - Several small modifications in the general models are required to produce the OSUT simulations. First, descriptions of Tasks 81 through 83 must be supplied. The SAINT input data describing these tasks is shown in Figure 7-2. This input assumes that the general models begin with Task 1. For models in which this is not true, branching statements must be changed appropriately.

In addition to the changes described in Figure 7-2, two changes must be made to the SAINT input data for the general model. First, the description of Task 1 (the first task in the task model) must be modified so that it is no longer a source task. This modification requires changing the number of predecessor tasks required and denoting the task as non-specific. Second, the SAINT input statement describing the branching from Task 50 must be deleted. The replacement for this statement was included in the input shown in Figure 7-2.

The SAINT models may require other changes to make them consistent with details of the OSUT data collection procedure which differed from the situation as described by

```

TAS,81,FRML TRN,0,9999,SC,0,,0,SD,(15) A,1*
TAS,82,TRN WAIT,1,9999,SC,0,,0,NS,(15) A,1*
TAS,83,TESTWAIT,1,9999,SC,2500000,,0,NS,(15) A,1*
UTC,81,0,0,5,1,-11,0*
UTC,82,0,0,5,1,-11,0*
UTC,83,0,0,5,1,-11,0*
MOD,81,2,D,T,4,D,T*
MOD,82,2,D,T,4,D,T*
MOD,83,2,D,T,4,D,T*
DET,81,82*
DET,82,1*
DET,83,1*
CFI,50,1,AGA,3,4,SA,,83,ALA,4,3,SA,,51,AGA,4,3,SA*

```

Figure 7-2 SAINT INPUT DATA FOR
OSUT MODEL FRAMEWORK

the general model. Such changes were required for LOADMG, STARTANK, LOADERPF, and RADIOMSG. In general, these changes involved the elimination of model tasks which were not measured in the OSUT data collection procedure.

7.1.2 Model output - Each of the eight models was run using parameter estimates from LEARN 4 and LEARN 10. A single simulation run involved 100 simulated subjects. Three kinds of results were calculated as a function of trial: (1) the proportion of simulated soldiers who performed the task correctly; (2) the proportion of task elements performed correctly; and (3) the time required to perform the task. These results were compared to the actual data.

Because of a late change in the SAINT model for GUNNERPF, there are slight differences between the model represented in SAINT and the model for which the parameters were estimated. Specifically, three performance measures are not included in the SAINT model, and five performance measures, which are considered to be distinct in the parameter estimation procedure, were considered to be repetitions of other performance measures in the SAINT model. The effect of these changes on the learning and retention parameters is unknown. However, the changes will lead the SAINT models to underestimate performance time. In addition, there are discrepancies in RADIOMSG, caused by the fact that performance on some task elements was not recorded, while other performance measures did not correspond to task elements.

Figure 7-3 shows the predictions of the two models along with the actual proportion of soldiers who performed each of the eight tasks correctly. Only those performance measures that were represented in the SAINT model were included in determining whether or not a soldier received a "go" on a particular task. Thus, performance measures which did not represent

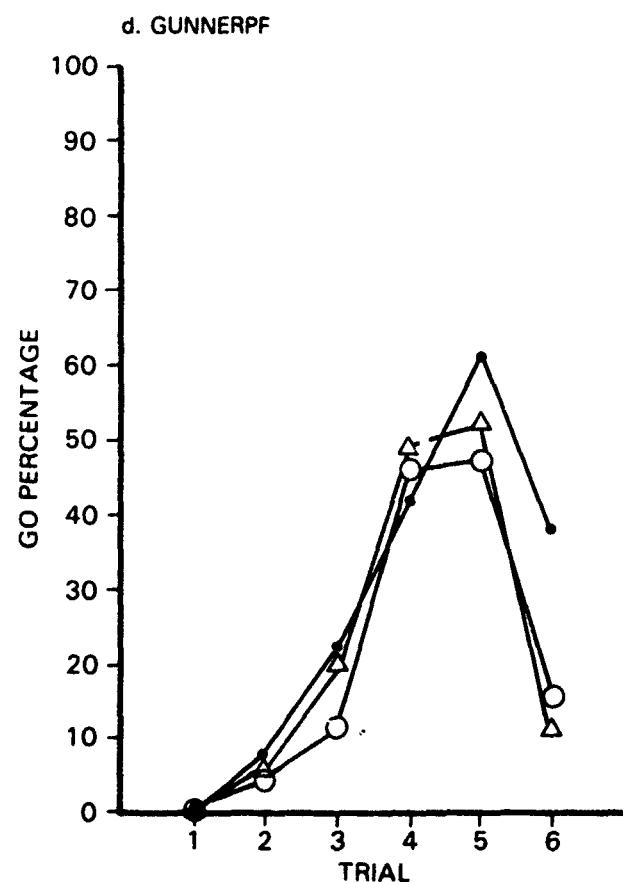
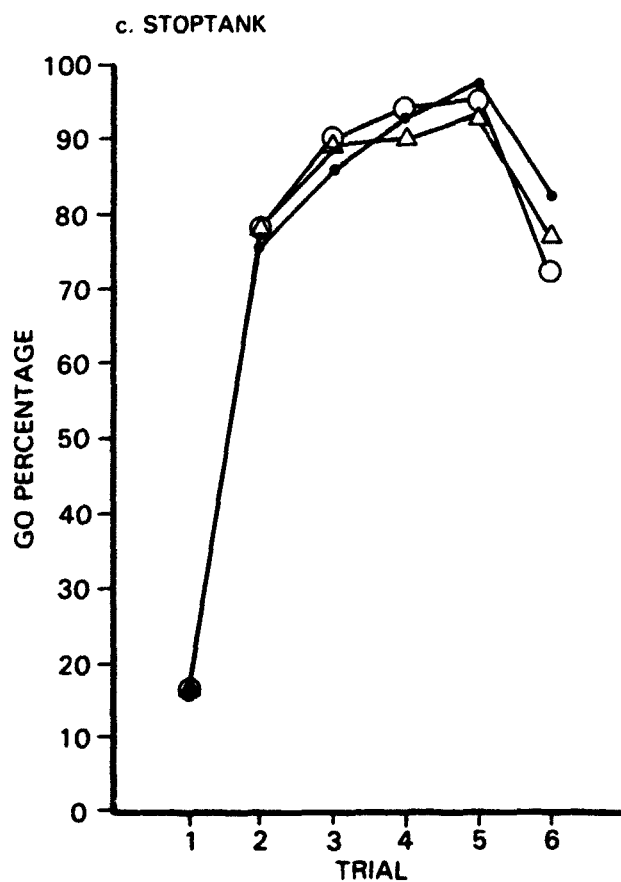
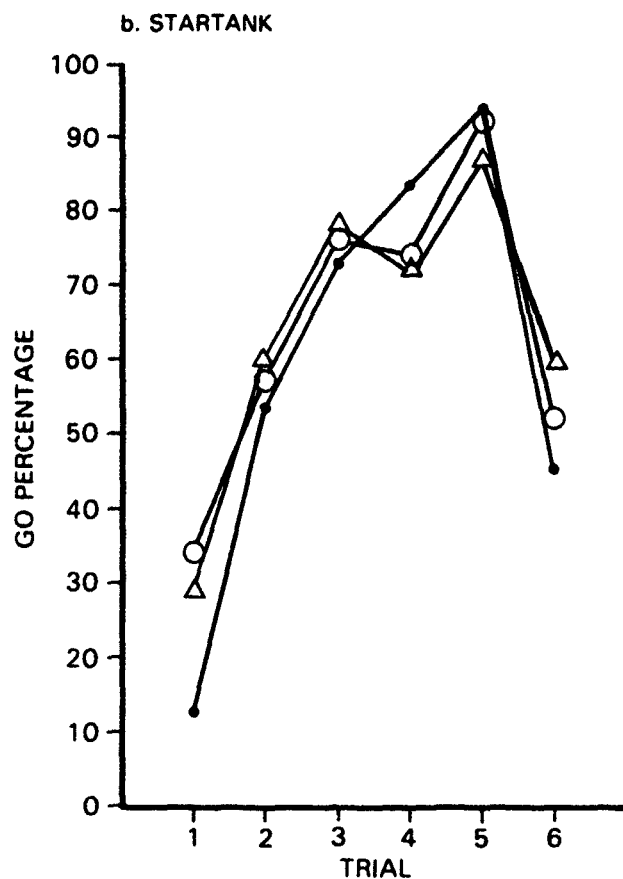
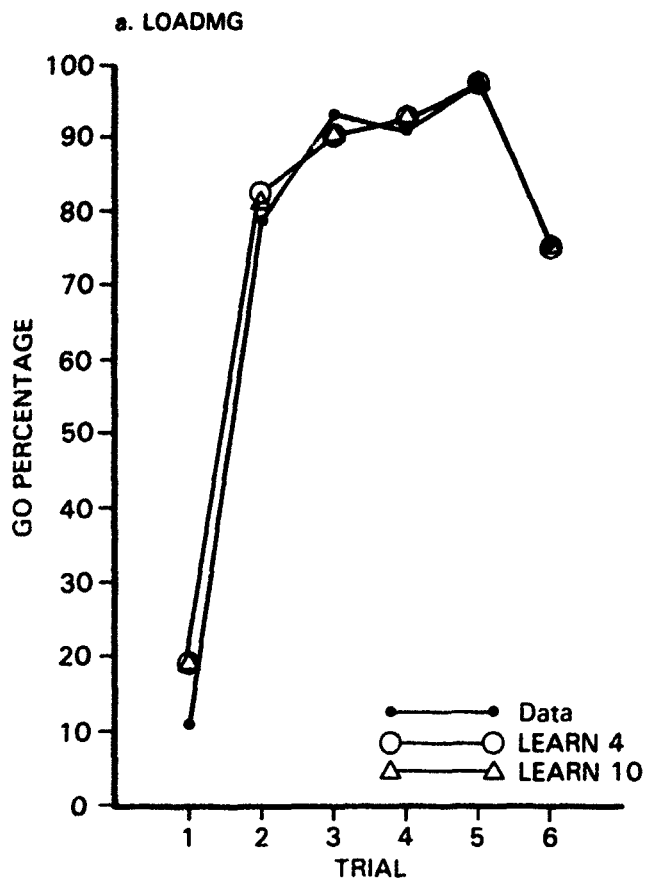


Figure 7-3
SIMULATED AND ACTUAL WHOLE-TASK GO RATES

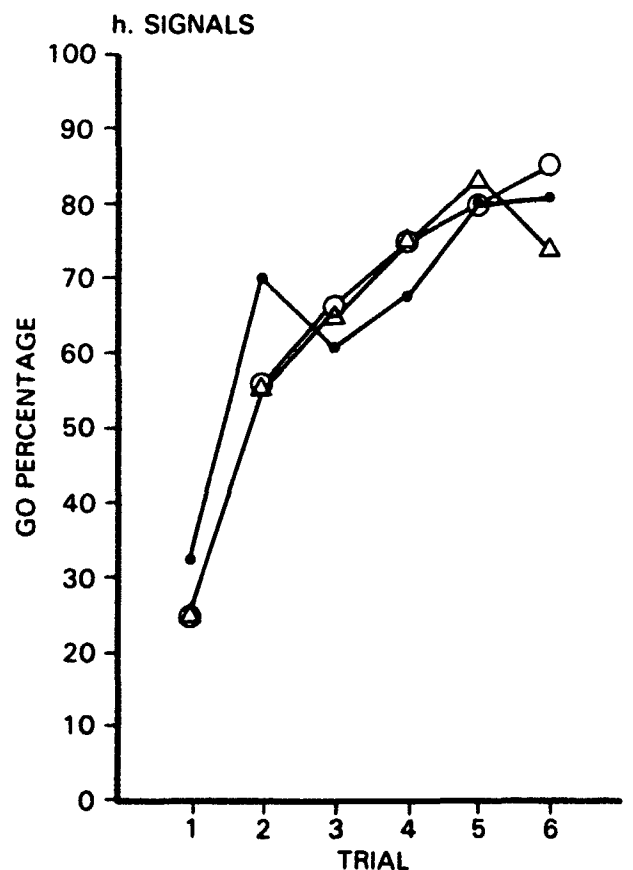
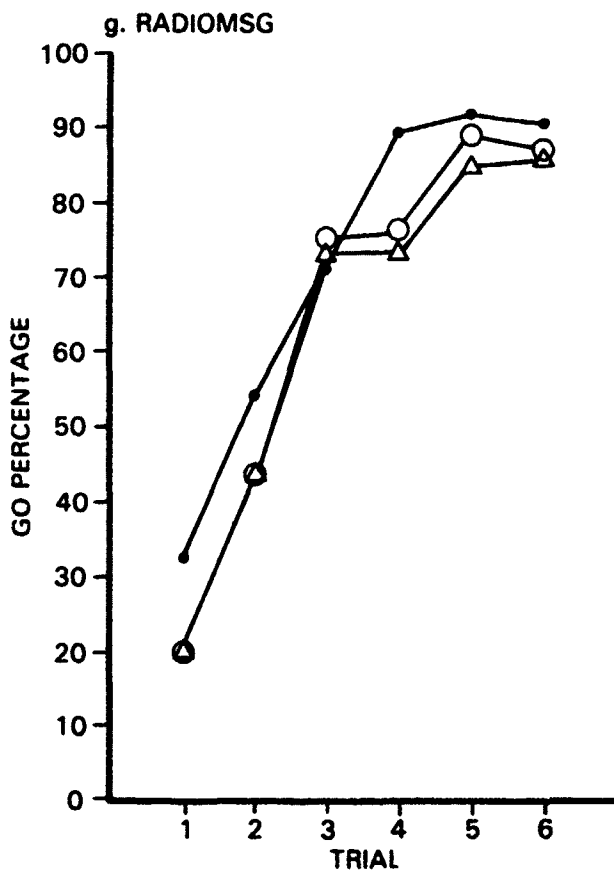
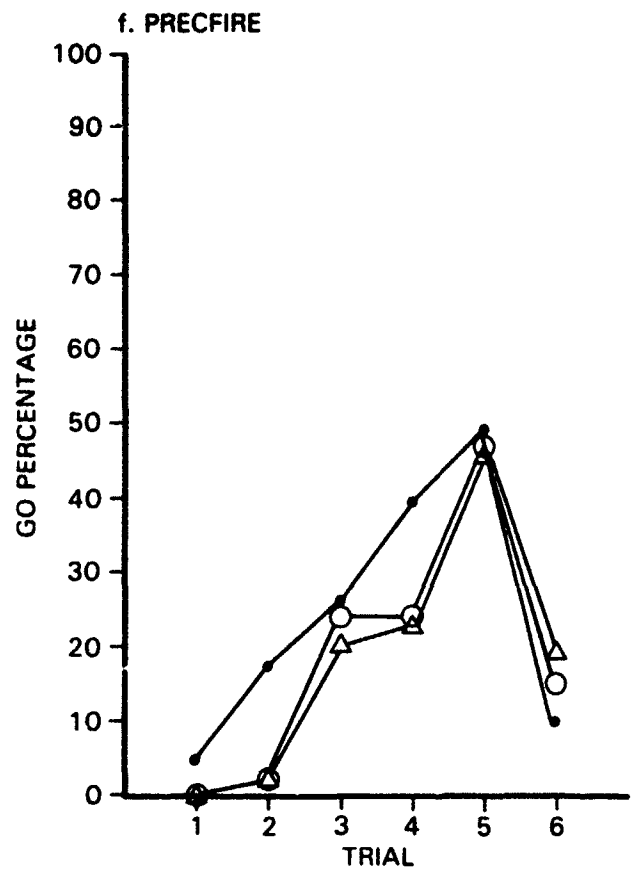
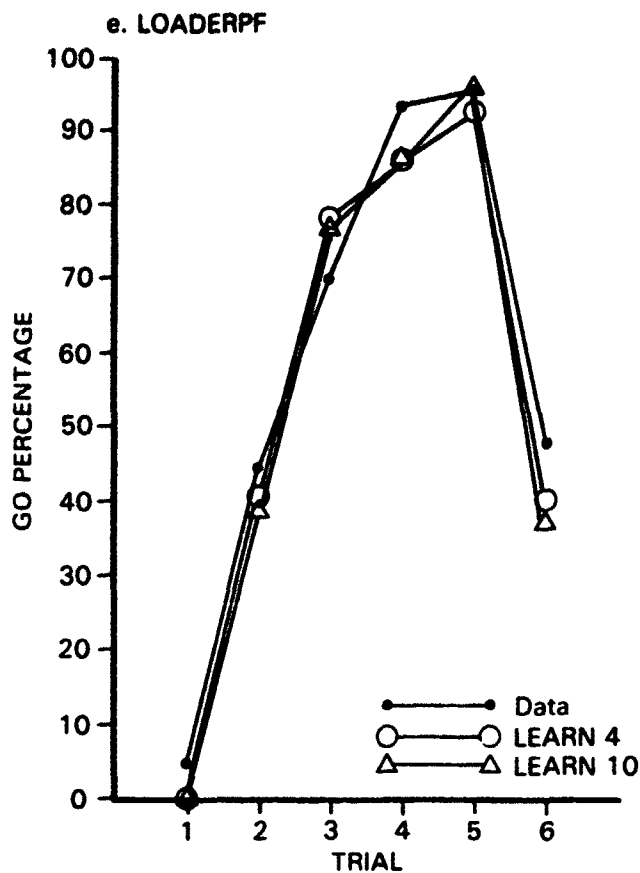


Figure 7-3
SIMULATED AND ACTUAL WHOLE-TASK GO RATES
(CONTINUED)

a specific action (in GUNNERPF and PRECFIRE), or which did not represent procedural steps (in RADIOMSG), were eliminated in the determination of the "go" rate. Figure 7-3 shows a very good fit of the "go" percentage data to the predictions of both learning models. Several of the discrepancies between the model and data, such as those on Trial 4 of STARTANK and RADIOMSG, and those on Trials 2 and 4 of PRECFIRE, appear to be caused by a particular sequence of random numbers selected in the simulations. Since the same numbers were used for both learning models, the same discrepancies occur for both LEARN 4 and LEARN 10. The two models were run with the same seed to the random number generator so that comparisons between the models would be not confounded with the particular random numbers selected in the simulation.

The models do better in predicting the proportion of steps performed correctly (Figure 7-4), because the parameter estimates were developed from individual step performance data. Both LEARN 4 and LEARN 10 perform well, and the loss in prediction, occurring when time as well as errors is being predicted (LEARN 10) is not noticeable in Figure 7-4.

The SAINT models embodying model LEARN 10 were concerned with prediction of the mean total performance time. Since the variability of performance times was not of primary concern, the SAINT models were run with the time standard deviation set arbitrarily low. Therefore, the variance of performance times in the simulated data will be substantially lower than that for the actual data. The predictions of the SAINT models are presented, along with the actual performance times, in Figure 7-5. For GUNNERPF, the time predictions were adjusted to account for differences in the number of performance measures between the parameter estimation methods and SAINT models.

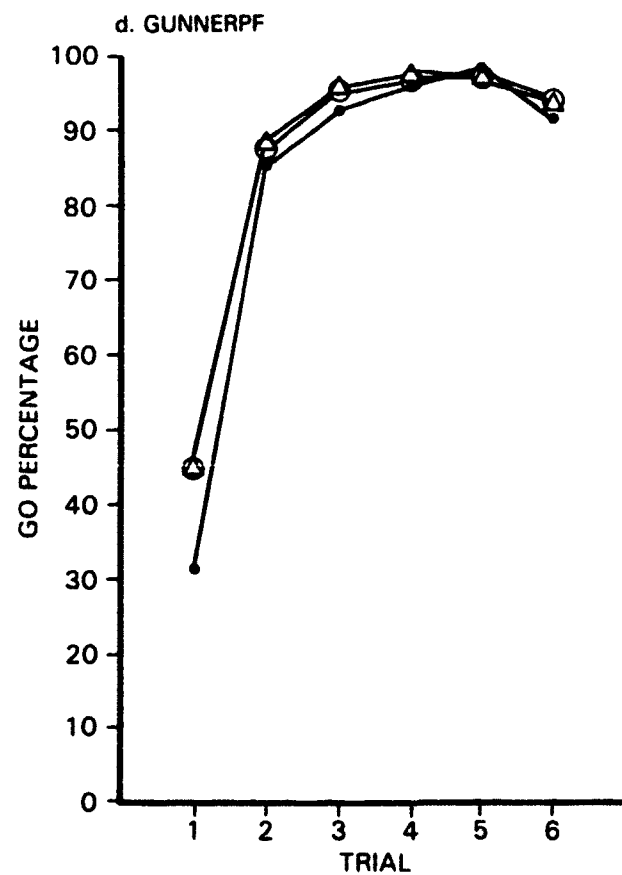
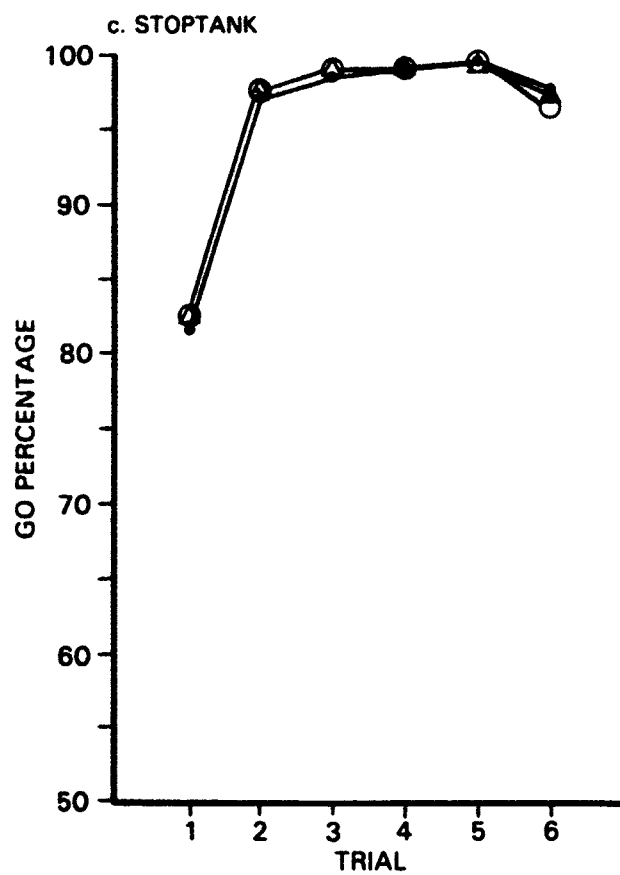
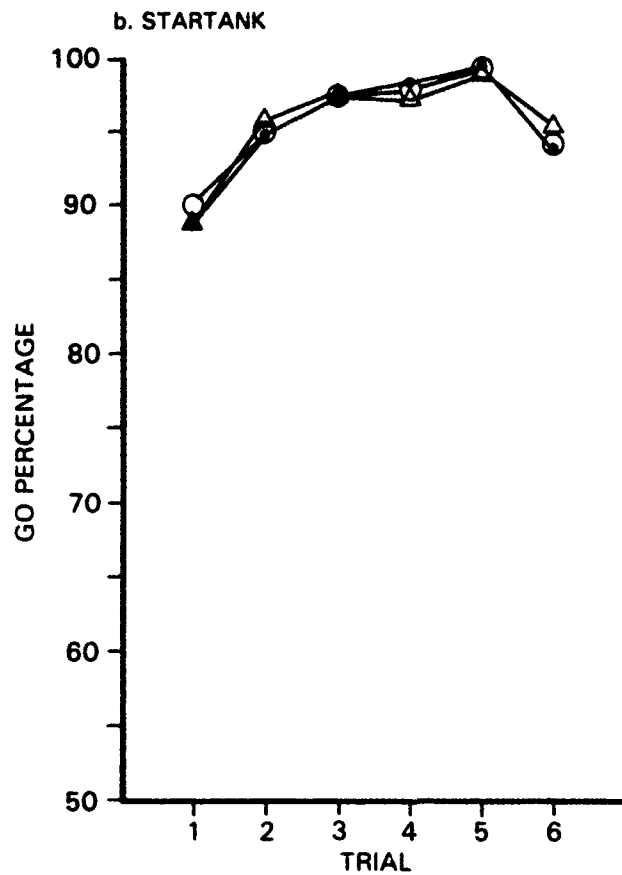
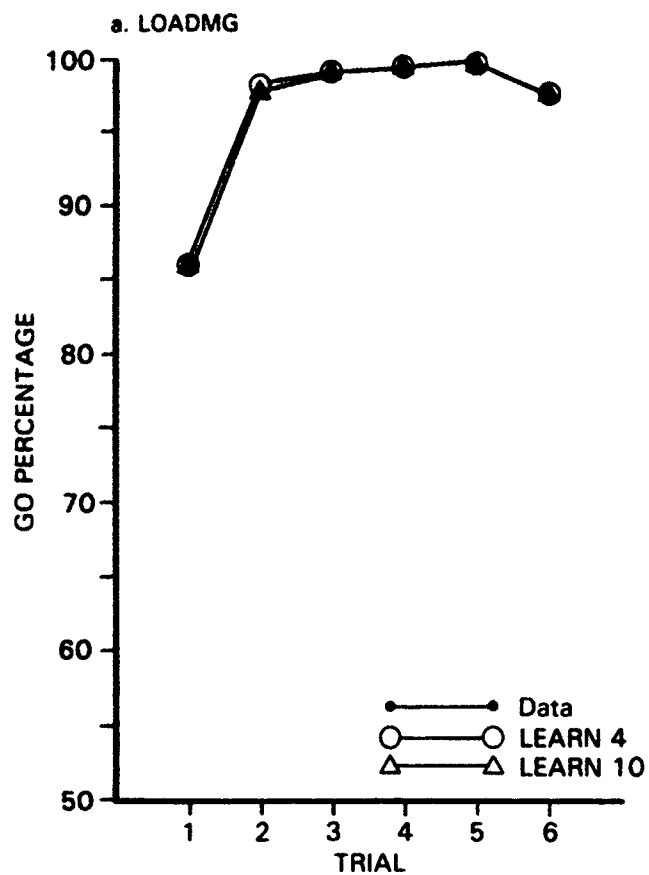


Figure 7-4
SIMULATED AND ACTUAL TASK-ELEMENT GO RATES

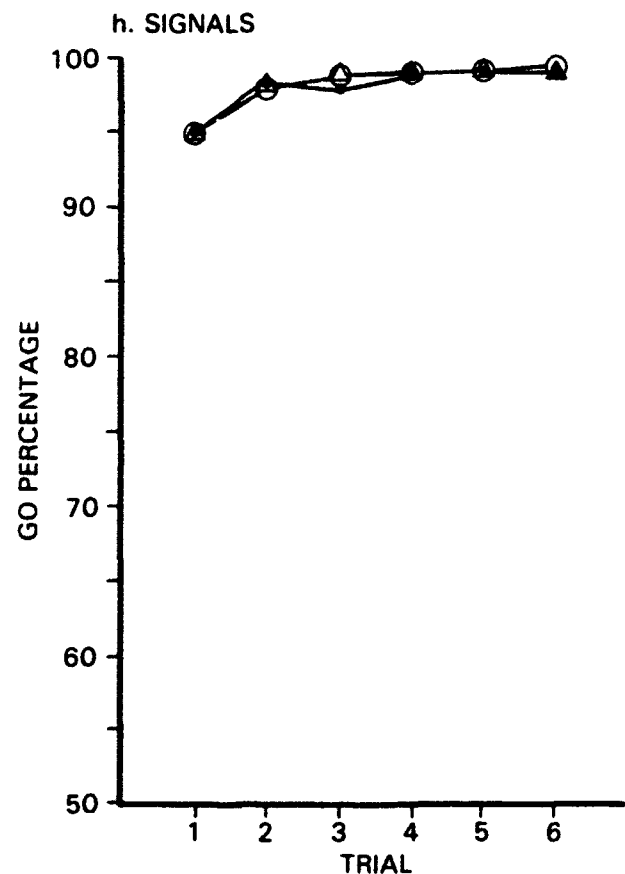
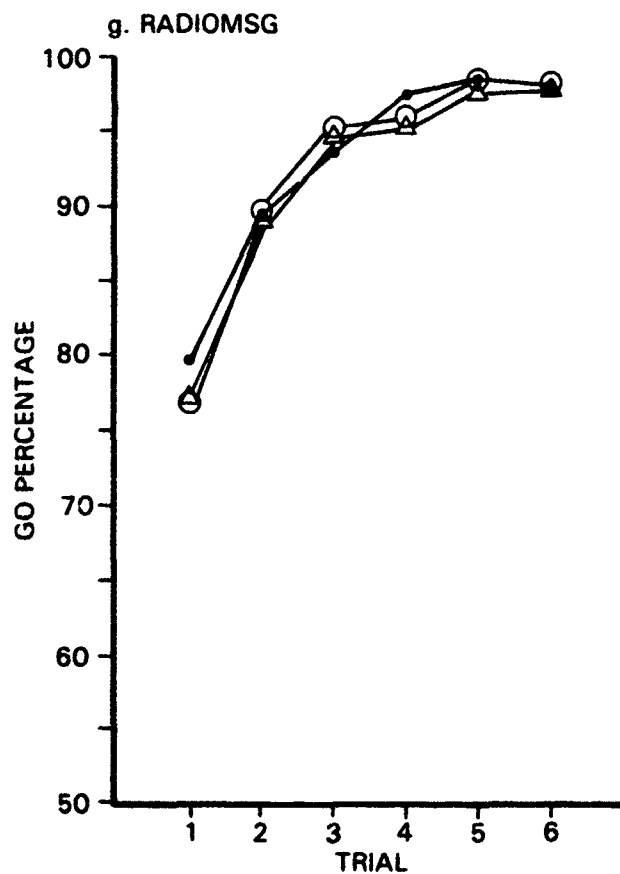
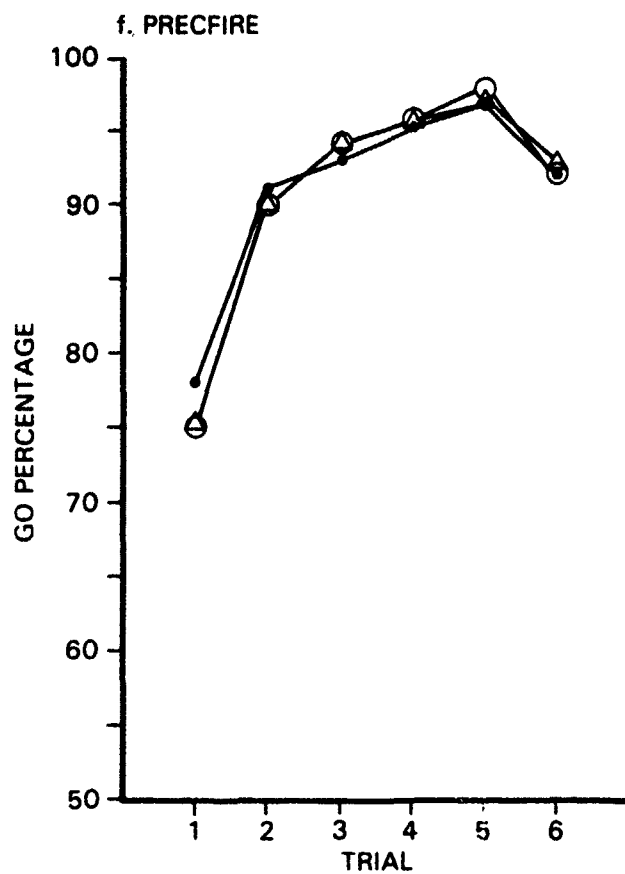
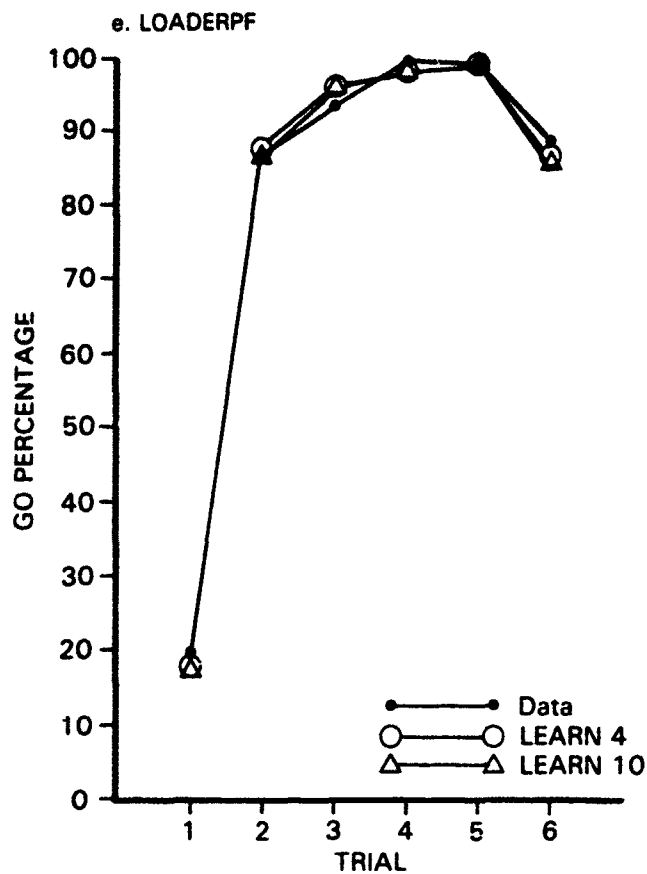


Figure 7-4
SIMULATED AND ACTUAL TASK-ELEMENT GO RATES
(CONTINUED)

Figure 7-5 shows an impressive fit between the data and predictions of the model on five of the tasks: LOADMG, STARTANK, STOPTANK, GUNNERPF, and LOADERPF. Predictions of the model are less impressive for PRECFIRE, RADIOMSG, and SIGNALS. The tasks in which the model predictions were best were those in which great improvements in speed were made over the learning trials. In these tasks, performance is at least three times faster on the fifth trial than it is on the first trial. On the other three tasks, the performance on the fifth trial is less than twice as fast as it is on the first trial. In fact, a logical examination of the three tasks can uncover some of the reasons that predictions of performance time are not as accurate for these tasks. In PRECFIRE, there was a forty second time limit on performance. Adherence to this limit would be expected to artificially lower the average performance time on early trial, and lead to less apparent improvement in performance time than is actually the case. In RADIOMSG and SIGNALS, the performance time is determined in great extent by the scorer. Since a large percentage of the performance time is accounted for by activities of the scorer, which are improving at a slower rate than those of the trainee, the fit of the models should not be as good for these two tasks.

7.2 Operational Unit Models

The data from the operational unit showed no evidence of forgetting during the retention interval sampled, although performance by the operational unit was substantially lower than the performance by the OSUT sample at the time of graduation. Therefore, we cannot use the SAINT model to verify the accuracy of the learning and retention models in the same manner as we did with the OSUT models. It is still helpful to build a framework for representing some of the factors characterizing the experience of a soldier in an operational unit that may effect retention of the procedures.

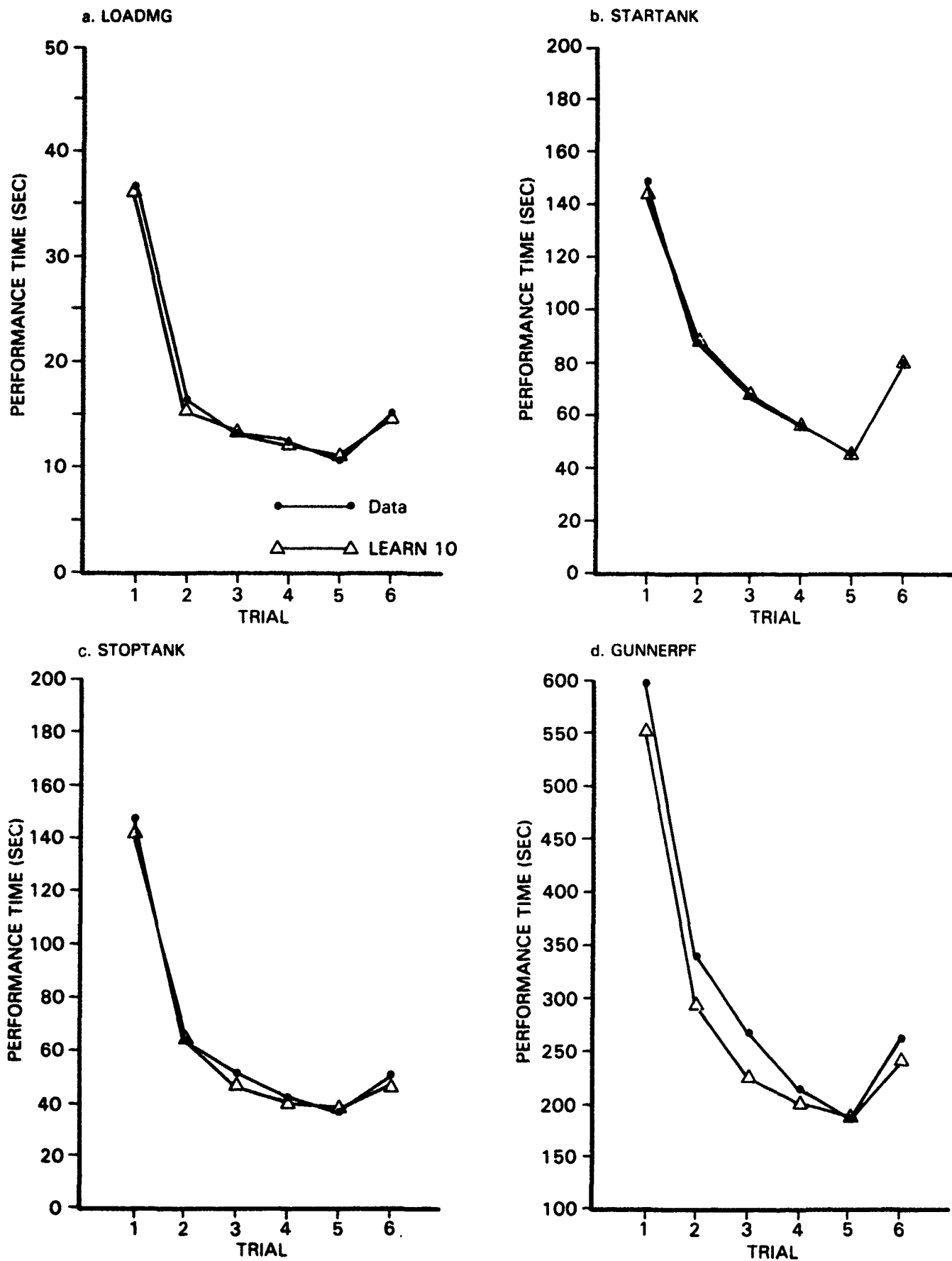


Figure 7-5
SIMULATED AND ACTUAL PERFORMANCE TIMES

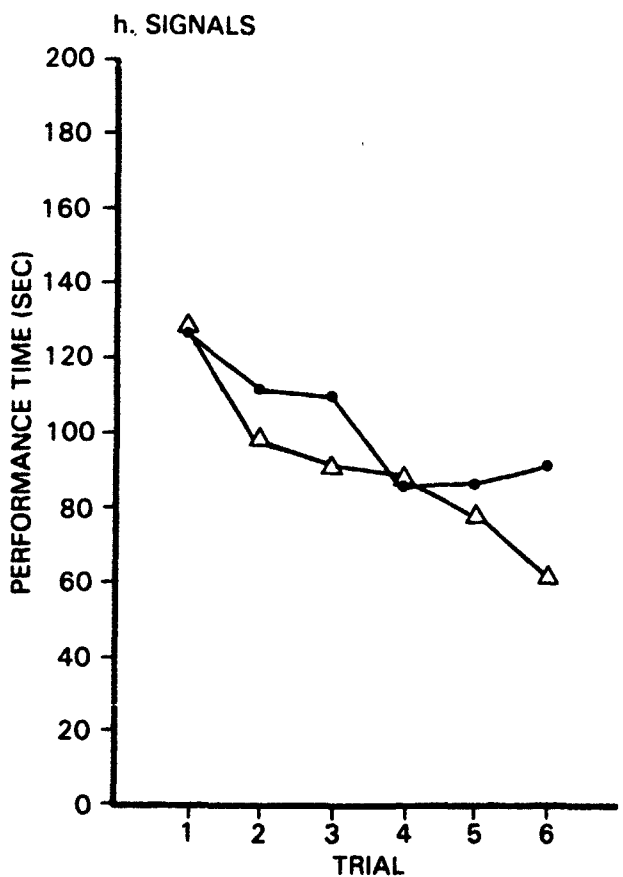
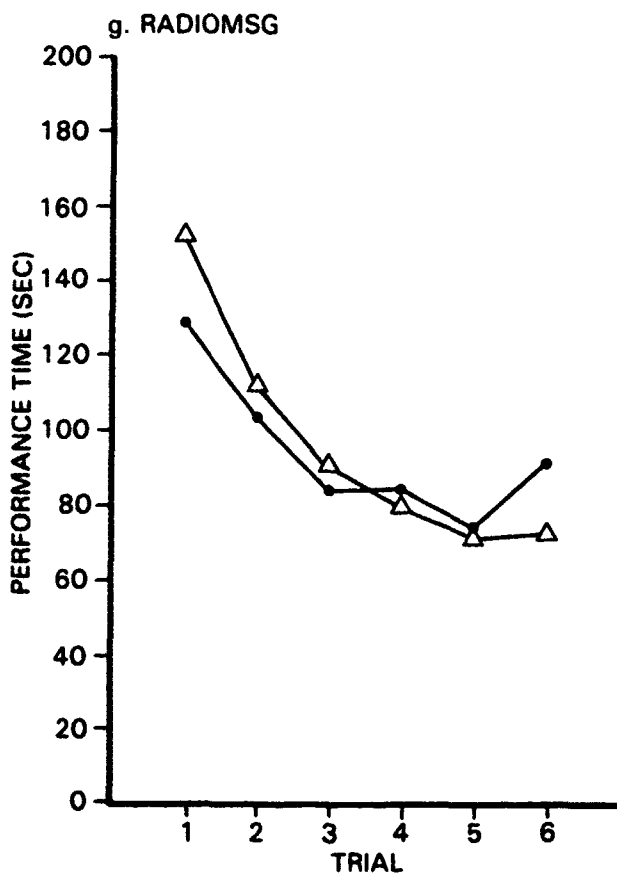
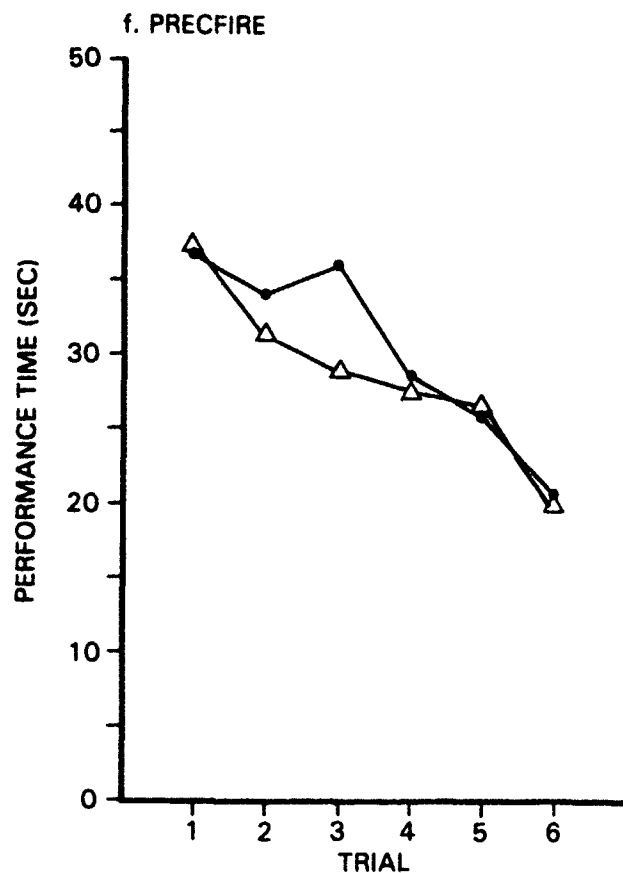
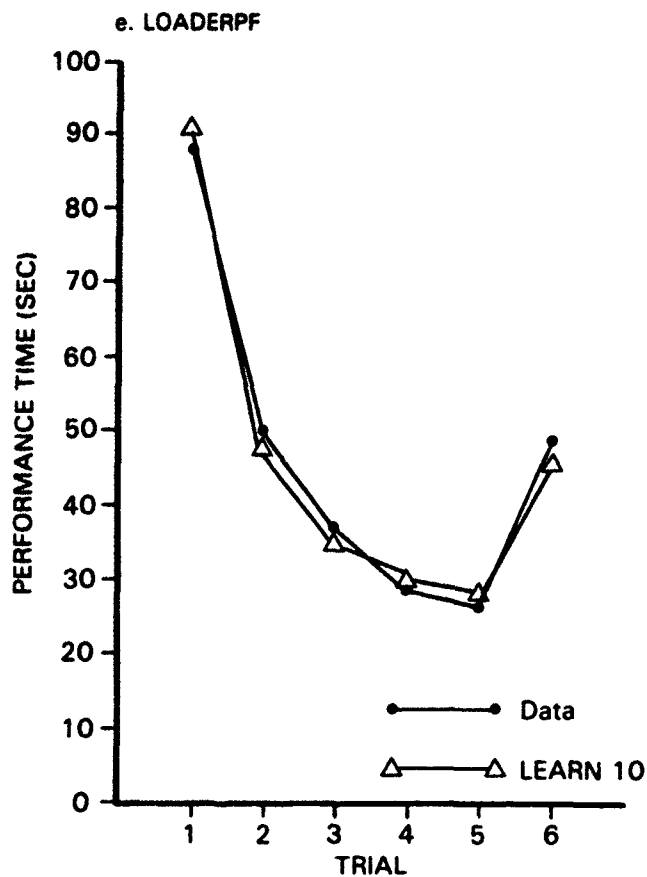


Figure 7-5
**SIMULATED AND ACTUAL PERFORMANCE TIMES
(CONTINUED)**

7.2.1 The OSUT framework - The framework, presented in Figure 7-6, consists of six tasks which are added to those in the general versions of the task models. In addition to the tasks, the operational unit version of the models uses four moderator functions, five resource attributes, and a different version of UINPT.

The tasks describe a variety of events which may occur in the experience of a soldier after graduation from OSUT. The operational models begin with Task 71, OSUT, which begins the simulation and reads (through MODRF 7) the values of five subject-specific resource attributes. These attributes represent the soldiers first duty position, his OSUT track, the duty position the soldier had at the time of the experiment, the time between graduation from OSUT and assignment of this duty position, and the time between graduation from OSUT and the experiment. Task 71 branches deterministically to three tasks, Task 72, DP WAIT, Task 73, EXP WAIT, and Task 74, WAIT.

Task 72, DP WAIT, represents the time that the soldier is in his first duty position. The time required for this task is read in by MODRF 7 and stored in the fourth resource attribute. At the completion of Task 72, USERF 7 changes the current duty position stored in RA(1,1) to the value of RA(1,3), which represents the duty position at the time of the experiment.

Task 73, EXP WAIT, is a counter which tracks the time between the soldiers graduation from OSUT and the experiment. The time required for this task is the time stored in RA(1,1). At the completion of Task 73, Tasks 72 and 74 are cleared, Task 75 is signaled, and SA(29) is set to two.

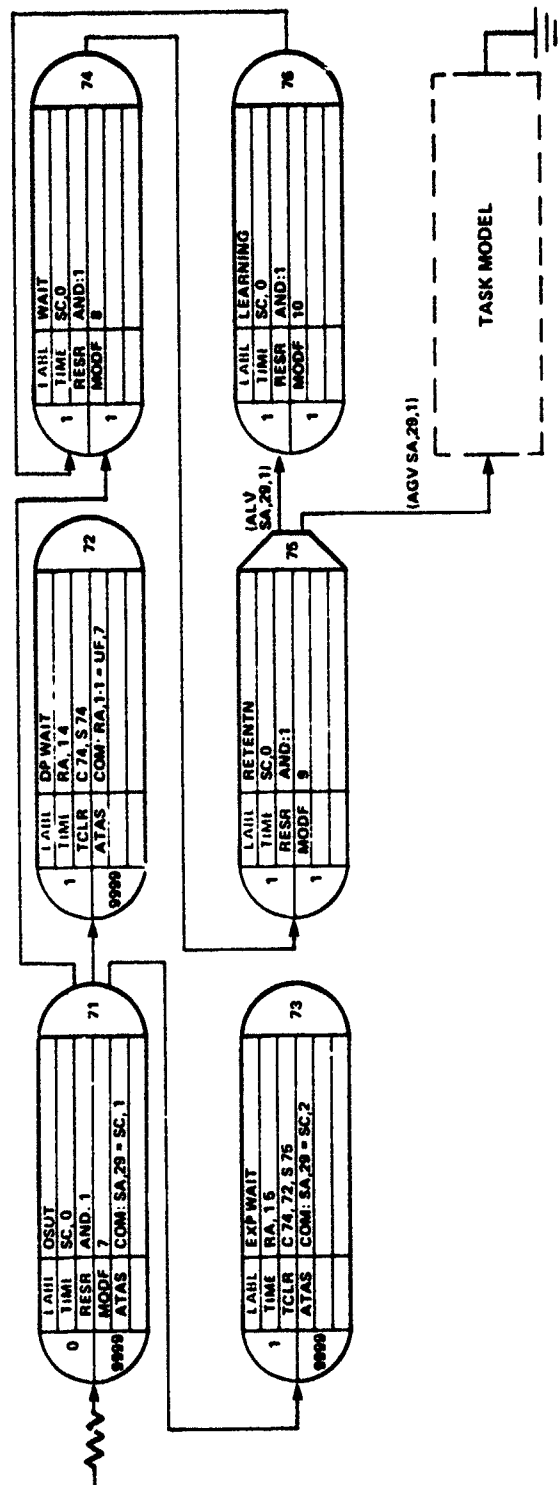


Figure 7-6
FRAMEWORK FOR OPERATIONAL UNIT MODELS

Task 74, WAIT, Task 75, RETENTN, and Task 76, LEARNING, form a loop representing periodic practice on the task. RETENTN, LEARNING, and the moderator functions associated with these tasks represent the changes in strength that occur during intervals of periodic practice. Task 74, WAIT, represents the intervals between practice, which are determined in MODRF 8, and which depend on the soldiers duty position. Since we are not collecting data on the performance of the soldiers during the retention interval, it is not necessary to simulate the task model during this interval. Rather, only the effects of practice on association strength are simulated. The loop represented by these three tasks is interrupted by the completion of Tasks 72 and 73. The loop is finally broken when Task 73 has been completed; then, Task 75 branches to the model of the specific procedure.

7.2.2 Implementing the framework - The SAINT input data to implement the operational unit framework is shown in Figure 7-7. These statements must be added to the general SAINT models. In addition, several other changes must be made to the models.

1. Data must be added to the end of the SAINT input data. These data represent the initial task strength as it relates to OSUT track, waiting time used in Task 74 as a function of duty position, and the number of base tasks in the model. Currently, there is provision for five strengths and four waiting times.
2. The program options (POP) instruction must be changed to allow for five resource attributes, twenty-nine system attributes, and ten moderator functions.

```

TAS,71,    OSUT,0,9999,SC,0, ,0,SO, (15) A,1*
TAS,72, DP WAIT,1,9999,RA,4,1,0,NS*
TAS,73,EXP WAIT,1,9999,RA,5,1,0,NS*
TAS,74,    WAIT,1,    1,SC,0, ,0,NS, (15) A,1*
TAS,75, RETENTN,1,    1,SC,0, ,0,NS, (15) A,1*
TAS,76,LEARNING,1,    1,SC,0, ,0,NS, (15) A,1*
UTC,71,0,0,5,1,-11,0*
UTC,72,0,0,5,1,-11,0*
UTC,73,0,0,5,1,-11,0*
UTC,74,0,0,5,1,-11,0*
UTC,75,0,0,5,1,-11,0*
UTC,76,0,0,5,1,-11,0*
MOD,71,2,D,T,4,D,T,7,A,T*
MOD,72,2,D,T,4,D,T*
MOD,73,2,D,T,4,D,T*
MOD,74,2,D,T,4,D,T,8,A,T*
MOD,75,2,D,T,4,D,T,9,A,T*
MOD,76,2,D,T,4,D,T,10,A,T*
ATA,71,COM,SA, ,29,SC,1*
ATA,72,COM,RA,1,1,UF,7*
ATA,73,COM,SA, ,29,SC,2*
DET,71,72,73,74*
DET,74,75*
DET,76,74*
CFI,75,76,ALV,1,29,SA, , 1,AGV,1,29,SA*
TCL,72,74,74*
TCL,73,74,75,72*

```

Figure 7-7. SAINT INPUT DATA FOR OPERATIONAL
UNIT MODEL FRAMEWORK

3. Task 1 must be changed so that it is not a source task and requires 1 predecessor completion for release.
4. An input file must be created and assigned to unit 2. There should be one entry in this file for each iteration of the simulation. Each entry should consist of the initial duty position number, the OSUT track number, the duty position at the time of the experiment, the time between OSUT graduation and assignment of the current duty position, and the time between OSUT graduation and the experiment. Values are currently read in F5.0 format, and the format may be changed in MODRF 7.

In addition to these changes, other changes in the model are required if the first task in the general task model is not Task 1. The additional changes involve the task numbers in branching statements. Finally, the general model for SIGNALS already reads data from unit two. The two statements may be alternated in the input data, or one of the read statements may be changed to be from another unit.

8.0 DISCUSSION

The analysis described in this report shows considerable success in the effort to develop a model that describes the learning and retention of procedural skills. For the OSUT sample, the learning and retention models embodied in the SAINT simulations provided good predictions of performance speed and accuracy as a function of training and the retention interval. Data from the operational unit are puzzling, however, and cannot be accounted for by any simple retention model. Nevertheless, the analytical model suggests explanations for the results in the operational unit. Specific aspects of procedural control, learning, and retention are discussed in this section along with prospects for future development and use of the model.

8.1 Procedural Control

The major two features characterizing the representation of procedural control taken in this project are the analysis of a procedure into task elements, and the representation of the interactions between task elements in a SAINT model. The analysis of a task into its components requires considerably more effort than the study of the task as a whole. The results of this modeling effort shed some light on the benefits to be obtained from task analysis in terms of understanding learning and retention.

8.1.1 Task analysis - The first benefit of task analysis comes from the fact that task elements may receive different amounts of training within a single procedure. These differences may be noted in the SIGNALS and PRECFIRE tasks and, to a lesser extent, in GUNNERPF. Practice of a procedure differentially trains the steps within the procedure. A task analysis can help to identify those steps that, because of

their extra training, would be expected to be learned more rapidly, and those which would require more practice.

Even after practice differences are accounted for in the learning models, differences in the learning rates for task elements were uncovered by the analysis. The existence of these differences, and the ability to measure their extent, offers the possibility of a greater understanding of what makes certain steps of a procedure especially difficult or easy. Although this analysis has not been conducted at the present time, an investigation of some of the variables affecting task-element difficulty is planned for this project.

A final benefit that comes from task analysis is that it provides a more sensitive measure of performance, and an opportunity to develop and interpret a greater number of experimental manipulations on training and performance measurement. The prompting procedure used in this study would not have been possible without the preliminary behavioral analyses performed on the tasks. In addition, the task analysis is the key activity that makes possible a method of performance measurement that is more sensitive than a simple assessment of "go" or "no go" on a task.

8.1.2 The SAINT representation - As was stated by Sticha (1982), the major test of a method for representing procedural control is its sufficiency for describing a variety of procedures. In this respect, SAINT provided a good framework for representing sequencing control in the tasks for which models were developed. Occasionally, problems arose that were a result of the particular implementation of the SAINT system. For example, limits on conditional branching made it inappropriate for use in SIGNALS, and probabilistics branching was substituted. In no case were there problems of a more concep-

tual nature with SAINT. Thus, SAINT appears to provide an adequate language for representing sequencing control in procedural tasks.

8.2 Results From OSUT Data

The results from the OSUT data provided both the best confirmation of the model and the greatest insight into where further model development is required. The learning model provided a good account of accuracy at both the task-element and whole-task levels, and of task performance speed. In addition, the analysis of the OSUT data identified task-element and individual differences that would benefit from further analysis. To a lesser extent, the OSUT data provided evidence for the retention model, but the evidence was not sufficient to identify the shape of the retention function.

8.3 Results from the Operational Unit

The main result from the operational unit is that performance is constant at different retention intervals. The modeling analysis did not identify the cause of this result, but it has eliminated some possible explanations. These explanations postulate that all changes in performance after training have occurred by the time that the soldiers were drawn for the operational unit sample. However, the level of performance in the operational unit sample is not consistent with such explanations, either under the assumption of simple decay, or of simple decay with intermittent practice.

If a simple decay function (either an exponential or a power function) describes the change in performance over the time since graduation from OSUT, then the decay rate must be so large that essentially all decay would have occurred in the first three months after OSUT graduation, before any soldiers

had been sampled for this research. However, the level of performance of the operational unit is approximately at the level of soldiers immediately after formal training, indicating that a moderate amount of information has been retained. It is difficult to imagine how a simple decay function would arrive at such a high asymptote. Consequently, there must be some effects of practice which are moderating the decay function.

If, on the other hand, practice is occurring, then the retention curve should be in the form of Equation 6. The asymptotic properties of this curve are controlled by the exponential learning portion of the equation. Thus, if the constant performance of the operational unit comes from the fact that the soldiers are at the asymptote of the retention function, they should be performing at a high level indicated by the maximum strength. Since performance by soldiers in the operational unit is substantially lower than performance at the time of the gate test, this explanation seems unlikely.

What seems to be the case for the operational unit is that they are receiving practice on part of the task, while another part of the task receives none. Since many of the tasks that were studied in this research are practiced fairly frequently, it is reasonable to postulate that all task elements are at an asymptote of their particular retention curves for soldiers in the operational unit. For those task elements that are practical in the unit, this asymptote would be at a high level, while for those that are not practiced, it would be low. If all soldiers tend to receive the same kind of practice in the unit, then we would expect to find that the same task elements would be forgotten by soldiers in operational units. This postulate would help explain the very low improvements in goodness-of-fit for the decay models. In addition, the explanation is consistent with the result that a significant fit for a retention model occurred only for GUNNERPF and LOADERPF, two

tasks which received little practice in the unit.

8.4 Future Prospects

The results of the analysis described in this report help identify some of the needs for further development, including further analysis of the current data and model development. In addition, the results may shed some light on the potential uses of the model in training research and management.

8.4.1 Analysis of current data - The current data include information which bears on many of the issues brought to light by this research. The four issues that are most relevant to the success of the modeling effort involve task-element differences, individual differences, the shape of the learning curve, and task differences.

The prediction of task-element differences seems to be the most important issue for further investigation. These differences were identified in the analysis of the OSUT data, and were hypothesized to account for the data from the operational unit. Part of the data collected for this research include ratings of task elements on several attributes thought to relate to skill learning and retention. One avenue for further analysis would attempt to relate model parameters to the values of these skill components. Favorable results of this analysis would allow one to predict how well a task element would be learned or retained from knowledge of these characteristics.

An analogous analysis may be done on individual differences. In this analysis, AFQT and ASVAB composite scores would be used to predict learning and decay rates. This analysis would attempt to relate model parameters to a variety of soldier aptitudes.

The shape of the learning curve has many implications on the model, particularly if extrapolations are made to many learning trials or to long retention intervals. In addition, the shape of the learning curve affects the relationship between association strength and performance speed. Although clear deviations from the exponential learning curve were found for only two of the tasks, these tasks were the ones that provided the most sensitive test of the shape of the learning curve because they involved the most learning. Although statistical rejection of the exponential model is not likely for the other tasks, it is worthwhile to examine the fit of the model to see if there are systematic deviations from the model which would suggest a different form of the learning curve.

With data for only eight tasks, it is unlikely that we will be able to find a definitive solution to the factors affecting learning and retention of tasks considered as wholes. Nevertheless, the tasks sampled do represent widely different kinds of procedural tasks. A preliminary analysis which related the values of model parameters to task characteristics, as well as task-element characteristics, might lead to some insights which would help to predict differences in learning and retention of tasks.

8.4.2 Needs for model development - The analyses described above will produce results that will lead to growth of the models described in this report. However, there is a need for further model development which cannot be supported by the current data, particularly in the area of retention of skills in the unit. The data for such model development should differ from the current data and focus more on repeated-measures designs, naturalistic observations, and greater documentation of practice. Data with one or more of these foci may already exist, and attempts to use existing data should be made wherever possible.

However, additional data should be collected to supplement current knowledge.

8.4.3 Use of the model - It is important at various stages of a research program to consider the ways in which the research products will be used. From the beginning of this project, two uses of the learning and retention models were envisioned:

1. As a method to organize the results of learning and retention experiments for researchers, and to guide in the design and interpretation of new research.
2. As a tool for training managers to predict the effects of various schedules for both initial and refresher training on performance levels.

In both cases, the model is designed to be the basis of a decision support system. Although the final form of the model is not settled, it is a good time to determine the functions that such a system would serve, so that the modeling research will take a course that best meets the needs of both training researchers and training managers. A reexamination of the ultimate goals of the model-development process is a critical part of further developments and analyses.

REFERENCES

- Aagard, J. A., & Braby, R. Learning guidelines and algorithms for types of training objectives (TAEG Report No. 23). Orlando, FL: Training and Evaluation Group, 1976.
- Duket, S. D., Wortman, D. B., Seifert, D. J., Hann, R. L., & Chubb, G. P. Analyzing SAINT output using SPSS (AMRL-TR-77-64). Dayton, OH: Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, 1978(a).
- Duket, S. D., Wortman, D. B., Seifert, D. J., Hann, R. L., & Chubb, G. P. Documentation for the SAINT simulation program (AMRL-TR-77-63). Dayton, OH: Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, 1978(b).
- Einhorn, H. J., & Hogarth, R. M. Behavioral decision theory: Processes of judgment and choice. Annual Review of Psychology, 1981, 32, 53-88.
- Greeno, J. G. Representation of learning as discrete transition in a finite state space. In D. H. Krantz, R. C. Atkinson, R. D. Luce, & P. Suppes (Eds.), Contemporary developments in mathematical psychology (Vol. 1). San Francisco: W. H. Freeman, 1974.
- Greeno, J. G. Psychology of learning, 1960-1980: One participant's observations. American Psychologist, 1980, 35(8), 713-728.
- Hann, R. L., & Kuperman, G. G. SAINT model of a choice reaction time paradigm. Proceedings of the Human Factors Society--19th Annual Meeting, 1975, 336-341.

- Hull, C. L. Principles of behavior. New York: Appleton-Century-Crofts, 1943.
- Hull, C. L. A behavior system: An introduction to behavior theory concerning the individual organism. New Haven, CN: Yale University Press, 1952.
- Kleinman, D. L., Baron, S., & Levison, W. H. An optimal control model of human response part I: Theory and validation. Automatica, 1970, 6, 357-369.
- Krantz, D. H. Measurement theory and qualitative laws in psychophysics. In D. H. Krantz, R. C. Atkinson, R. D. Luce, & P. Suppes (Eds.), Contemporary development in mathematical psychology (Vol. 2). San Francisco: W. H. Freeman, 1974.
- Lewis, C. H. Speed and practice. Unpublished manuscript, 1979. (Available from C. H. Lewis, IBM Research Center, Yorktown Heights, NY 10598)
- Miller, J. O., & Greeno, J. G. Goodness-of-fit tests for models of latency and choice. Journal of Mathematical Psychology, 1978, 17, 1-13.
- Newell, A., & Rosenbloom, P. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, NJ: Erlbaum, 1981.
- Newell, A., & Simon, H. A. Human problem solving. Englewood Cliffs, NJ: Prentice-Hall, 1972.
- Pew, R. W., & Baron, S. Perspectives on human performance modelling. Proceedings of the Conference on Analysis, Design and Evaluation of Man-Machine Systems, 1982.

- Pew, R. W., Baron, S., Feehrer, C. E., & Miller, D.C. Critical review and analysis of performance models applicable to man-machine systems evaluation (Report No. 3446). Cambridge, MA: Bolt, Beranek, and Newman, Inc., March 1977.
- Restle, F. The significance of all-or-none learning. Psychological Bulletin, 1965, 69, 329-324.
- Seifert, D. J., Koeplinger, G., & Hoyland, C. M. REDIMEN: SAINT redimensioning program (AFAMRL-TR-80-5). Dayton, OH: Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, 1980.
- Snoddy, G. S. Learning and stability. Journal of Applied Psychology, 1926, 84, 127-190.
- Sticha, P. J. Review of analytic models of procedural learning and performance (DDI/PR 82-14-334). McLean, VA: Decisions and Designs, Inc., 1982.
- Thurstone, L. L. A law of comparative judgment. Psychological Review, 1927, 34, 273-286.
- Wickelgren, W. A. Time, interference, and rate of presentation on short-term recognition memory for items. Journal of Mathematical Psychology, 1970, 7, 219-235.
- Wickelgren, W. A. Trace resistance and the decay of long-term memory. Journal of Mathematical Psychology, 1972, 9, 418-455.
- Wickelgren, W. A. Single-trace fragility theory of memory dynamics. Memory and Cognition, 1974(a), 2, 775-780.

- Wickelgren, W. A. Strength/resistance theory of the dynamics of memory storage. In D. H. Krantz, R. C. Atkinson, R. D. Luce, & P. Suppes (Eds.), Contemporary developments in mathematical psychology (Vol. 1). San Francisco: W. H. Freeman, 1974(b).
- Wickelgren, W. A., & Norman, D. A. Strength models and serial position in short-term recognition memory. Journal of Mathematical Psychology, 1966, 3, 316-347.
- Wortman, D. B., Ducket, S. D., Seifert, D. J., Hann, R. L., & Chubb, G. P. Simulation using SAINT: A user-oriented instruction manual (AMRL-TR-77-61). Dayton, OH: Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, 1978(a).
- Wortman, D. B., Ducket, S. D., Seifert, D. J., Hann, R. L., & Chubb, G. P. The SAINT users manual (AMRL-TR-77-62). Dayton, OH: Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, 1978(b).
- Wortman, D. B., & Hixson, A. F., III. A SAINT model of the AN/TSQ-73 missile minder: User's guide (Research Note RN 79-6). West LaFayette, IN: Pritsker Associates, Inc., August 1978. (NTIS No. AD A068998)

APPENDIX A.

LISTING OF USER-WRITTEN
SUBROUTINES FOR SAINT MODELS

```

      SUBROUTINE LEARN(ITASK)
      COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3TIM,
+             TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2(40)
C *****
C * LEARN - THIS FUNCTION UPDATES THE STRENGTH OF ITASK IN UTC(3)
C * BY THE PROPORTION, TRLRN, TO THE ASYMPTOTE, SASMT. IT IS CALLED
C * BY MODRF 2.
C *****
      CALL GETTC(ITASK,1,VALUE)
      INDEX=VALUE
      IF (INDEX.EQ.0) GO TO 100
      CALL GETTC(ITASK,3,STRNT)
      STRNT=(1-TRLRN(INDEX))*STRNT + TRLRN(INDEX)*SASMT(INDEX)
      CALL PUTTC(ITASK,3,STRNT)
100 RETURN
      END

```

```

      SUBROUTINE MODRF(MODFN,NTASK)
      COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NPUNCH,
+               NRNIT,NRENT,MNDC,NDC,NDTN,NNTC
      COMMON /COM22/ TTIME,PFIRB
      COMMON /SITVR/ NUMTR,NUMPM,NUMGO(10),ICNTL(6,12),ISGNL(4,15),
+               ISERS(4,6)
      COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3TIM,
+               TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2(40)
      COMMON /VERS2/ NUMTSK,STRNTH(4),WAIT(5)
      DIMENSION XRA(5)
      GO TO (10,20,30,40,50,60,70,80,90,100),MODFN
C *****
C * MODRF 1 - THIS FUNCTION COMPARES IA(2) TO TC(7) FOR TASKS 3 - 10
C * AND SETS BRANCHING PROBABILITIES IN SA(5)-SA(12) ACCORDINGLY.
C * USED BY SIGNALS.
C *****
      10 CALL GETIA(2,RQMTS)
      DO 11 I=3,10
      CALL GETTC(I,7,VALUE)
      J=I+2
      AMATCH=0.0
      IF (ABS(VALUE-RQMTS).LT.0.01) AMATCH=1.0
      11 CALL PUTSA(J,AMATCH)
      RETURN
C *****
C * MODRF 2 - THIS FUNCTION CHECKS THE TRACE STRENGTH AND BRANCHES
C * TO THE APPROPRIATE HINT LEVEL. TTIME IS ADJUSTED. HINT LEVEL
C * IS RECORDED AND OVERALL GO/NO GO IS UPDATED.
C *****
C
C               FIND THE HINT LEVEL BASED ON TRACE STRENGTH, UTC(5).
C
      20 CALL GETTC(NTASK,1,VALUE)
      INDEX=VALUE
      IF (INDEX.EQ.0) GO TO 29
      CALL GETTC(NTASK,5,STMOM)
      XLEVEL=0.0
      IF (STMOM.LT.THRS0(INDEX)) XLEVEL=1.0
      IF (STMOM.LT.THRS1(INDEX)) XLEVEL=2.0
      IF (STMOM.LT.THRS2(INDEX)) XLEVEL=3.0
C
C               SET TTIME AND GO/NG VARIABLE, SA(2), DEPENDING ON HINT LEVEL
C
      CALL GETTC(NTASK,3,STRNT)
      TTIME=TCONS+TSTRN*(SASMT(INDEX)-STRNT)+RNORM(1)*TSTDV**2
      IF (XLEVEL.GT.0.9) TTIME=TTIME+P1TIM
      IF (XLEVEL.GT.1.9) TTIME=TTIME+P2TIM
      IF (XLEVEL.GT.2.9) TTIME=TTIME+P3TIM
      TTIME=AMAX1(TTIME,0.0)
      IF (XLEVEL.GT.0.0) CALL PUTSA(2,0.0)
C
C               STORE STATISTICAL VARIABLES FOR THIS SUBTASK.
C
      CALL GETSA(3,TRIAL)
      NTRIAL=TRIAL

```

```

        ICLCT=(NUMPM+1)*(NTRIAL-1)+INDEX
        XOUT=0.0
        IF (XLEVEL.GT.0.5) XOUT=1.0
        CALL UCLCT(XOUT,ICLCT)
        CALL UHIST(XLEVEL,ICLCT)
C
C
C         ADJUST STRENGTH IN UTC(3) USING RETEN AND LEARN
C
        TIME=TNOW+TTIME
        CALL RETEN(NTASK,TIME)
        CALL LEARN(NTASK)
    29 RETURN
C *****
C *  MODRF 3 - THIS FUNCTION IS CALLED AT THE END OF EACH LEARNING *
C *  TRIAL.  IT COLLECTS ALL STATISTICS ABOUT THE TRIALS AND RESETS *
C *  THE TRIAL GO/NG COUNTER. *
C *****
C
C         ADD GO/NG TO TRIAL COUNTER.
C
    30 CALL GETSA(2,GONG)
        CALL GETSA(3,TRIAL)
        NTRIAL=TRIAL
        NUMGO(NTRIAL)=NUMGO(NTRIAL)+GONG
C
C
C         COLLECT STATISTICS ON GO/NG PROPORTION.
C
        ICOUNT=(NUMPM+1)*NTRIAL
        CALL UCLCT(GONG,ICOUNT)
        CALL UHIST(GONG,ICOUNT)
        TIME=TNOW-TMARK(IDUM)
        ICOUNT=6*(NUMPM+1)+NTRIAL
        CALL UCLCT(TIME,ICOUNT)
        CALL UHIST(TIME,ICOUNT)
C
C
C         RESET GO/NO VARIABLE, SA(2).
C
        CALL PUTSA(2,1.0)
        RETURN
C *****
C *  MODRF 4 - THIS FUNCTION IS CALLED BY EACH TASK.  IT RESETS THE *
C *  VALUE OF UTC(5), MOMENTARY STRENGTH, TO -11, SO THAT PRIORITIES *
C *  WILL BE SET CORRECTLY ON THE NEXT TRIAL. *
C *****
    40 CALL PUTTC(NTASK,5,-11.0)
        RETURN
C *****
C *  MODRF 5 - THIS FUNCTION IS CALLED BY THE FIRST TASK IN PRECFIRE. *
C *  IT GETS THE APPROPRIATE INFORMATION FROM ICNTL AND PUTS IT INTO *
C *  SA(8) (AMMO TYPE), SA(9) (TARGET MOVING), AND SA(10) (PERISCOPE *
C *  OK). *
C *****
    50 CALL GETSA(3,TRIAL)
        NTRIAL=TRIAL
        CALL GETSA(1,ENG)

```

```

      NENG=3*(ENG-1)
      DO 51 I=1,3
        J=7+I
        XCNTL=ICNTL(NTRIAL,NENG+I)
51    CALL PUTSA(J,XCNTL)
      RETURN
C *****
C *  MODRF 6 - THIS FUNCTION SETS UP THE COURSE, THAT IS, THE
C *  SEQUENCE OF TASKS THAT IS USED IN SIGNALS.  THE SEQUENCE DEPENDS
C *  ON THE SERIES NUMBER, IN SA(28), AND THE TRIAL, IN SA(3).  THE
C *  SEQUENCE IS RETRIEVED FROM ISGNL.
C *****
      60 CALL GETSA(28,SERIES)
        NSRS=SERIES
        CALL GETSA(3,TRIAL)
        NTRIAL=TRIAL
        NCORS=ISERS(NSRS,NTRIAL)
        DO 61 I=1,14
          J=I+12
          VALUE=ISGNL(NCORS,I)
61    CALL PUTSA(J,VALUE)
      RETURN
C *****
C *  MODRF 7 - THIS FUNCTION IS CALLED BY EACH TASK IN VERSION 2.  IT
C *  SETS THE MEAN STRENGTHS DEPENDING ON THE SUBJECT'S OSUT TRACK,
C *  AND READS IN RA(1,1) - RA(1,5).
C *****
      70 READ(2,71) (XRA(I),I=1,5)
      71 FORMAT(5F5.0)
        DO 72 I=1,5
          XXRA=XRA(I)
62    CALL PUTRA(1,I,XXRA)
        CALL GETRA(1,2,TRACK)
        TRACK=TRACK+.005
        ITRACK=INT(TRACK)
        VALUE=STRNTH(ITRACK)
        DO 76 I=1,NUMTSK
76    CALL PUTTC(I,3,VALUE)
        CALL PUTTC(50,3,VALUE)
        CALL PUTTC(51,3,VALUE)
        DO 77 I=71,76
77    CALL PUTTC(I,3,VALUE)
      RETURN
C *****
C *  MODRF 8 - THIS FUNCTION IS CALLED BY EACH VERSION 2 TASK.  IT
C *  CALCULATES THE WAIT TIME AS A FUNCTION OF THE CURRENT DUTY POSITI-
C *  TION, RA(1,1).
C *****
      80 CALL GETRA(1,1,DUTY)
        DUTY=DUTY+.005
        IDUTY=INT(DUTY)
        TTIME=WAIT(IDUTY)
      RETURN
C *****
C *  MODRF 9 - THIS FUNCTION IS CALLED BY EACH VERSION 2 TASK.  IT

```

```

C * CALLS RETENTION FOR EVERY BASE MODEL TASK.
C *****
  90 DO 91 I=1,NUMTSK
  91 CALL RETEN(I,TNOW)
    RETURN
C *****
C * MODRF 10 - THIS FUNCTION IS CALLED BY EACH VERSION 2 TASK. IT
C * CALLS LEARN FOR EVERY BASE MODEL TASK.
C *****
  100 DO 101 I=1,NUMTSK
  101 CALL LEARN(I)
    RETURN
    END

```

```

      FUNCTION PRIOR(ITASK)
      COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NPUNCH,
+      NRNIT,NRENT,MNDC,NDC,NDTN,NNTC
      COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3TIM,
+      TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2(40)
C *****
C * THIS FUNCTION PERFORMS THREE TASKS:
C *   1. IT READS UTC(2) AND SETS THE PRIORITY BASED ON DS,1.
C *   2. IT READS UTC(3) (STRENGTH) AND SETS UTC(5) (MOMENTARY
C *      STRENGTH) BASED ON DS,1.
C *   3. IT SETS THE PRIORITY TO -11 IF THE STRENGTH IS NOT
C *      GREATER THAN THRS0.
C * USE OF THIS FUNCTION DURING MULTIPLE-RUN SIMULATIONS (E.G. LEARN-
C * ING TRIALS) ASSUMES THAT UTC(5) IS RESET TO -11 WHEN THE TASK IS
C * PERFORMED. IN ADDITION, UTC(5) MUST BE INITIALIZED TO -11.
C * OTHERWISE PRIORITIES AND STRENGTHS WILL NOT CHANGE OVER TRIALS
C * DYNAMIC PRIORITIES ARE REQUIRED FOR STARTANK, STOPTANK, GUNNERPF,
C * LOADERPF, AND PRECFIRE.
C *****
C
C      SET MOMENTARY STRENGTHS.
C
      CALL GETTC(ITASK,1,VALUE)
      INDEX=VALUE
      CALL GETTC(ITASK,5,STMOM)
      IF (STMOM.GE.-10) GO TO 10
      CALL RETEN(ITASK,TNOW)
      CALL GETTC(ITASK,3,STMOM)
      STMOM=STMOM+RNORM(1)
      CALL PUTTC(ITASK,5,STMOM)
C
C      SET TASK PRIORITY.
C
      CALL GETTC(ITASK,2,VALUE)
      IF (VALUE.EQ.0) GO TO 10
      PRIOR=VALUE+RNORM(1)
      CALL PUTPR(ITASK,PRIOR)
      IF (INDEX.EQ.0) RETURN
      IF (STMOM.LT.THRS0(INDEX)) PRIOR=-11
      CALL PUTPR(ITASK,PRIOR)
      RETURN
C
C      GO HERE IF PRIORITY ALREADY SET.
C
10 CALL GETPR(ITASK,VALUE)
   PRIOR=VALUE
   RETURN
   END

```

```

SUBROUTINE RETEN(ITASK,TIME)
COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3TIM,
+             TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2(40)
C *****
C *   THIS SUBROUTINE UPDATES THE STRENGTH AND FRAGILITY OF A TRACE
C *   FOR TASK "ITASK" TO BE CURRENT TO TIME "TIME".
C *****
CALL GETTC(ITASK,6,TLAST)
IF (TIME.LE.TLAST) GO TO 100
CALL GETTC(ITASK,1,VALUE)
INDEX=VALUE
IF (INDEX.EQ.0) GO TO 100
CALL GETTC(ITASK,3,STRNT)
CALL GETTC(ITASK,4,FRAGL)
CALL PUTTC(ITASK,6,TIME)
TDIFF=TIME-TLAST
STRNT=STRNT*(1+FRDEC*FRAGL*TDIFF)**(-STDEC)
CALL PUTTC(ITASK,3,STRNT)
FRAGL=FRAGL/(1+FRDEC*FRAGL*TDIFF)
CALL PUTTC(ITASK,4,FRAGL)
100 RETURN
END

```

```

SUBROUTINE UINPT
C *****
C ***** VERSIONS 1 AND 3 ***** VERSIONS 1 AND 3 *****
C *
C * UINPT - THIS ROUTINE READS VALUES FOR COMMON BLOCKS SITVR AND
C * PARMS. SITVR CONTAINS SIX VARIABLES: NUMTR, WHICH INDICATES
C * THE NUMBER OF LEARNING TRIALS; NUMPM, THE NUMBER OF PERFORMANCE
C * MEASURES IN THE TASK; NUMGO, WHICH RECORDS THE NUMBER OF "GO"
C * ITERATIONS OVER THE SIMULATION; ICNTL, WHICH REFLECTS THE ORDER
C * OF THE ENGAGEMENTS IN PRECFIRE; ISGNL, WHICH REPRESENTS THE
C * FOUR COURSES IN SIGNALS; AND ISERS, WHICH REPRESENTS FOUR
C * ORDERS OF COURSES USED IN SIGNALS. PARMS CONTAINS MODEL
C * PARAMETERS FOR THE STRENGTH-FRAGILITY MODEL. STDEC AND FRDEC
C * ARE DECAY PARAMETERS FOR STRENGTH AND FRAGILITY RESPECTIVELY.
C * TRLRN IS THE LEARNING INCREMENT FOR TRAINING TRIALS; AND SASMT
C * IS THE STRENGTH ASYMPTOTE. THRS0, THRS1, AND THRS2 ARE THE
C * RETRIEVAL THRESHOLDS FOR HINT LEVELS 0, 1, AND 2, RESPECTIVELY.
C * TSTRN, TSTDV, P1TIM, P2TIM, AND P3TIM ARE TIME PARAMETERS.
C *****
COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NFUNCH,
+ NRNIT,NRENT,MNDC,NDC,NDTN,NNTC
COMMON /SITVR/ NUMTR,NUMPM,NUMGO(10),ICNTL(6,12),ISGNL(4,15),
+ ISERS(4,6)
COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3TIM,
+ TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2(40)
DO 10 I=1,10
10 NUMGO(I)=0
READ(NCRDR,1) NUMTR
TRIAL=NUMTR
CALL PUTSA(4,TRIAL)
READ(NCRDR,1) NUMPM
READ(NCRDR,2) STDEC
READ(NCRDR,2) FRDEC
READ(NCRDR,2) TCONS
READ(NCRDR,2) TSTRN
READ(NCRDR,2) TSTDV
READ(NCRDR,2) P1TIM
READ(NCRDR,2) P2TIM
READ(NCRDR,2) P3TIM
READ(NCRDR,3) (TRLRN(I),I=1,NUMPM)
READ(NCRDR,3) (SASMT(I),I=1,NUMPM)
READ(NCRDR,3) (THRS0(I),I=1,NUMPM)
READ(NCRDR,3) (THRS1(I),I=1,NUMPM)
READ(NCRDR,3) (THRS2(I),I=1,NUMPM)
1 FORMAT(I5)
2 FORMAT(F10.0)
3 FORMAT(16F5.0)
READ(NCRDR,1) MORE
GO TO(100,200,300),MORE
C
C GO HERE IF TASK IS PRECFIRE.
C
200 READ(NCRDR,4) ((ICNTL(I,J),J=1,12),I=1,6)
4 FORMAT(12I5)
GO TO 100

```

```

C
C      GO HERE IF TASK IS SIGNALS.
C
300 READ(NCRDR,5) ((ISGNL(I,J),J=1,15),I=1,4)
      READ(NCRDR,6) ((ISERS(I,J),J=1,6),I=1,4)
      5 FORMAT(15I5)
      6 FORMAT(6I5)
100 RETURN
      END

```

```

SUBROUTINE UINPT
C *****
C ***** VERSION 2   VERSION 2   VERSION 2   *****
C
C * UINPT - THIS ROUTINE READS VALUES FOR COMMON BLOCKS SITVR AND
C * PARMS. SITVR CONTAINS SIX VARIABLES: NUMTR, WHICH INDICATES
C * THE NUMBER OF LEARNING TRIALS; NUMPM, THE NUMBER OF PERFORMANCE
C * MEASURES IN THE TASK; NUMGO, WHICH RECORDS THE NUMBER OF "GO"
C * ITERATIONS OVER THE SIMULATION; ICNTL, WHICH REFLECTS THE ORDER
C * OF THE ENGAGEMENTS IN PRECFIRE; ISGNL, WHICH REPRESENTS THE
C * FOUR COURSES IN SIGNALS; AND ISERS, WHICH REPRESENTS FOUR
C * ORDERS OF COURSES USED IN SIGNALS. PARMS CONTAINS MODEL
C * PARAMETERS FOR THE STRENGTH-FRAGILITY MODEL. STDEC AND FRDEC
C * ARE DECAY PARAMETERS FOR STRENGTH AND FRAGILITY RESPECTIVELY.
C * TRLRN IS THE LEARNING INCREMENT FOR TRAINING TRIALS; AND SASMT
C * IS THE STRENGTH ASYMPTOTE. THRS0, THRS1, AND THRS2 ARE THE
C * RETRIEVAL THRESHOLDS FOR HINT LEVELS 0, 1, AND, 2, RESPECTIVELY.
C * TSTRN, STSDV, P1TIM, P2TIM, AND P3TIM ARE TIME PARAMETERS.
C * FOR VERSION2, COMMON VERS2 CONTAINS (1)STRNTH-THE MEAN STRENGTH
C * DEPENDING ON THE OSUT TRACK, (2)WAIT-THE WAIT TIME AS A FUNCTION
C * OF THE DUTY POSITION & (3)NUMTSK- NUMBER OF BASE SUBTASKS.
C *****
COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NPUNCH,
+ NRNIT,NRENT,MNDC,NDC,NDTN,NNTC
COMMON /SITVR/ NUMTR,NUMPM,NUMGO(10),ICNTL(6,12),ISGNL(4,15),
+ ISERS(4,6)
COMMON /PARMS/ STDEC,FRDEC,TCONS,TSTRN,TSTDV,P1TIM,P2TIM,P3TIM,
+TRLRN(40),SASMT(40),THRS0(40),THRS1(40),THRS2(40)
COMMON /VERS2/ NUMTSK,STRNTH(4),WAIT(5)
DO 10 I=1,10
10 NUMGO(I)=0
READ(NCRDR,1) NUMTR
TRIAL=NUMTR
CALL PUTSA(4,TRIAL)
READ(NCRDR,1) NUMPM
READ(NCRDR,2) STDEC
READ(NCRDR,2) FRDEC
READ(NCRDR,2) TCONS
READ(NCRDR,2) TSTRN
READ(NCRDR,2) TSTDV
READ(NCRDR,2) P1TIM
READ(NCRDR,2) P2TIM
READ(NCRDR,2) P3TIM
READ(NCRDR,3) (TRLRN(I),I=1,NUMPM)
READ(NCRDR,3) (SASMT(I),I=1,NUMPM)
READ(NCRDR,3) (THRS0(I),I=1,NUMPM)
READ(NCRDR,3) (THRS1(I),I=1,NUMPM)
READ(NCRDR,3) (THRS2(I),I=1,NUMPM)
1 FORMAT(I5)
2 FORMAT(F10.0)
3 FORMAT(16F5.0)
READ(NCRDR,3) (STRNTH(I),I=1,4)
READ(NCRDR,3) (WAIT(I),I=1,5)
READ(NCRDR,1) NUMTSK
READ(NCRDR,1) MORE

```

```

      GO TO(100,200,300),MORE
C
C      GO HERE IF TASK IS PRECFIRE.
C
200 READ(NCRDR,4) ((ICNTL(I,J),J=1,12),I=1,6)
   4 FORMAT(12I5)
      RETURN
C
C      GO HERE IF TASK IS SIGNALS.
C
300 READ(NCRDR,5) ((ISGNL(I,J),J=1,15),I=1,4)
      READ(NCRDR,6) ((ISERS(I,J),J=1,6),I=1,4)
   5 FORMAT(15I5)
   6 FORMAT(6I5)
100 RETURN
      END

```

SUBROUTINE UOTPT

```
*****
* UOTPT - THIS SUBROUTINE PREPARES AND PRINTS A PLOT OF WHOLE-TASK
* GO RATE BY TRIAL, AND OUTPUTS OTHER STATISTICS, NAMELY, THE
* TASK-ELEMENT GO RATE AND THE PERFORMANCE TIME. OUTPUT IS IN
* TABULAR FORM. OPTIONALLY, HISTOGRAMS MAY BE OBTAINED BY REMOVING
* COMMENTED CALLS TO UHIST IN THIS ROUTINE AND IN MODRF 2 AND 3.
*****
COMMON /SITVR/ NUMTR,NUMPM,NUMGO(10),ICNTL(6,12),ISGNL(4,15),
+             ISERS(4,6)
DO 10 I=1,NUMTR
  TRIAL=I
  VALUE=NUMGO(I)
10 CALL UPLOTT(VALUE,TRIAL,1)
  CALL UCLCT(1,0)
  CALL UHIST(1,0)
  CALL UPLOTT(1,1,0)
RETURN
END
```

```

      FUNCTION USERF(MM)
      GO TO (10,20,30,40,50,60,70,80),MM
C *****
C * USER FUNCTION 1 - THIS FUNCTION INCREMENTS SA(1).  USED BY      *
C * GUNNERPF, LOADERPF, RADIMSG, AND SIGNALS.                      *
C *****
      10 CALL GETSA(1,VALUE)
         USERF=VALUE+1
         RETURN
C *****
C * USER FUNCTION 2 - THIS FUNCTION RETURNS A VALUE FROM          *
C * SA(13) - SA(26) AS INDEXED IN SA(1).  USED BY SIGNALS.      *
C *****
      20 CALL GETSA(1,VALUE)
         I=VALUE+12
         CALL GETSA(I,VALUE)
         USERF=VALUE
         RETURN
C *****
C * USER FUNCTION 3 - THIS FUNCTION RETURNS THE VALUE OF SA(8).  USED *
C * BY PRECFIRE                                                    *
C *****
      30 CALL GETSA(8,VALUE)
         USERF=VALUE
         RETURN
C *****
C * USER FUNCTION 4 - THIS FUNCTION CHECKS THE COMPATIBILITY OF SA(8)*
C * AND SA(6).  (1,1), (2,2), AND (3,2) COMBINATIONS RETURN 1; OTHER *
C * COMBINATIONS RETURN 0.  USED BY PRECFIRE.                    *
C *****
      40 USERF=0
         CALL GETSA(8,CMD)
         CALL GETSA(6,RTCL)
         IF(CMD.EQ.RTCL) USERF=1
         IF(CMD*RTCL.GT.5) USERF=1
         RETURN
C *****
C * USER FUNCTION 5 - THIS FUNCTION INCREMENTS THE TRIAL NUMBER,  *
C * SA(3).  IT IS CALLED AT THE END OF EACH LEARNING TRIAL.      *
C *****
      50 CALL GETSA(3,VALUE)
         USERF=VALUE+1
         RETURN
C *****
C * USER FUNCTION 6 - THIS FUNCTION READS THE SERIES NUMBER FROM  *
C * UNIT 2.  IT IS USED BY SIGNALS AT THE START OF EACH ITERATION *
C *****
      60 READ(2,1) ISRS
         1 FORMAT(I5)
         USERF=ISRS
         RETURN
C *****
C * USER FUNCTION 7 - THIS FUNCTION IS USED BY EACH VERSION 2 TASK.
C * IT CHANGES THE CURRENT DUTY POSITION RA(1,1) TO THE DUTY POSITION
C * THE TIME OF THE EXPERIMENT.

```

```

C *****
  70 CALL GETRA(1,3,VALUE)
    USERF=VALUE
    RETURN
C *****
C * USER FUNCTION 8 - THIS FUNCTION CHANGES THE RETICLE SETTING,
C * SA(6) FROM 1 (2) TO 2 (1). IT IS CALLED BY TASK 11 OF PRECFIRE.
C *****
  80 CALL GETSA(6,VALUE)
    USERF=3.0-VALUE
    RETURN
    END

```

APPENDIX B.

SAINT INPUT DATA FOR GENERAL
VERSION OF PROCEDURE MODELS

GEN,LOADMG1,12,16,1982,11,100,1,125865,1,N*
 POPTIONS,2,0,0,05,4,(9) 2*
 OUTPUT OPTIONS,0,0,0,0,0,(14)N,N,N,N*
 DIS,1,NO,0,-10,10,1*
 DIS,2,ER,1,0,10,1*
 UB0, 1, CHARGER, 2, SAFE, 3, COVER UP, 4, FEEDT UP, 5, HNDLBACK,
 6, EXAMINE, 7, FEEDT DN, 8, ROUND IN, 9, AMMO IN, 10, COVER DN,
 11, FIRE, 12, SAYS UP, 13, GO/NG 1*
 UB0, 14, CHARGER, 15, SAFE, 16, COVER UP, 17, FEEDT UP, 18, HNDLBACK,
 19, EXAMINE, 20, FEEDT DN, 21, ROUND IN, 22, AMMO IN, 23, COVER DN,
 24, FIRE, 25, SAYS UP, 26, GO/NG 2*
 UB0, 27, CHARGER, 28, SAFE, 29, COVER UP, 30, FEEDT UP, 31, HNDLBACK,
 32, EXAMINE, 33, FEEDT DN, 34, ROUND IN, 35, AMMO IN, 36, COVER DN,
 37, FIRE, 38, SAYS UP, 39, GO/NG 3*
 UB0, 40, CHARGER, 41, SAFE, 42, COVER UP, 43, FEEDT UP, 44, HNDLBACK,
 45, EXAMINE, 46, FEEDT DN, 47, ROUND IN, 48, AMMO IN, 49, COVER DN,
 50, FIRE, 51, SAYS UP, 52, GO/NG 4*
 UB0, 53, CHARGER, 54, SAFE, 55, COVER UP, 56, FEEDT UP, 57, HNDLBACK,
 58, EXAMINE, 59, FEEDT DN, 60, ROUND IN, 61, AMMO IN, 62, COVER DN,
 63, FIRE, 64, SAYS UP, 65, GO/NG 5*
 UB0, 66, CHARGER, 67, SAFE, 68, COVER UP, 69, FEEDT UP, 70, HNDLBACK,
 71, EXAMINE, 72, FEEDT DN, 73, ROUND IN, 74, AMMO IN, 75, COVER DN,
 76, FIRE, 77, SAYS UP, 78, GO/NG 6*
 UB0, 79, TIME 1, 80, TIME 2, 81, TIME 3, 82, TIME 4, 83, TIME 5, 84, TIME 6*
 UHI, 1, CHARGER, 3, 0, 1, 2, SAFE, 3, 0, 1, 3, COVER UP, 3, 0, 1,
 4, FEEDT UP, 3, 0, 1, 5, HNDLBACK, 3, 0, 1, 6, EXAMINE, 3, 0, 1,
 7, FEEDT DN, 3, 0, 1, 8, ROUND IN, 3, 0, 1, 9, AMMO IN, 3, 0, 1*
 UHI, 10, COVER DN, 3, 0, 1, 11, FIRE, 3, 0, 1, 12, SAYS UP, 3, 0, 1,
 13, GO/NG 1, 1, 0, 1*
 UHI, 14, CHARGER, 3, 0, 1, 15, SAFE, 3, 0, 1, 16, COVER UP, 3, 0, 1,
 17, FEEDT UP, 3, 0, 1, 18, HNDLBACK, 3, 0, 1, 19, EXAMINE, 3, 0, 1,
 20, FEEDT DN, 3, 0, 1, 21, ROUND IN, 3, 0, 1, 22, AMMO IN, 3, 0, 1*
 UHI, 23, COVER DN, 3, 0, 1, 24, FIRE, 3, 0, 1, 25, SAYS UP, 3, 0, 1,
 26, GO/NG 2, 1, 0, 1*
 UHI, 27, CHARGER, 3, 0, 1, 28, SAFE, 3, 0, 1, 29, COVER UP, 3, 0, 1,
 30, FEEDT UP, 3, 0, 1, 31, HNDLBACK, 3, 0, 1, 32, EXAMINE, 3, 0, 1,
 33, FEEDT DN, 3, 0, 1, 34, ROUND IN, 3, 0, 1, 35, AMMO IN, 3, 0, 1*
 UHI, 36, COVER DN, 3, 0, 1, 37, FIRE, 3, 0, 1, 38, SAYS UP, 3, 0, 1,
 39, GO/NG 3, 1, 0, 1*
 UHI, 40, CHARGER, 3, 0, 1, 41, SAFE, 3, 0, 1, 42, COVER UP, 3, 0, 1,
 43, FEEDT UP, 3, 0, 1, 44, HNDLBACK, 3, 0, 1, 45, EXAMINE, 3, 0, 1,
 46, FEEDT DN, 3, 0, 1, 47, ROUND IN, 3, 0, 1, 48, AMMO IN, 3, 0, 1*
 UHI, 49, COVER DN, 3, 0, 1, 50, FIRE, 3, 0, 1, 51, SAYS UP, 3, 0, 1,
 52, GO/NG 4, 1, 0, 1*
 UHI, 53, CHARGER, 3, 0, 1, 54, SAFE, 3, 0, 1, 55, COVER UP, 3, 0, 1,
 56, FEEDT UP, 3, 0, 1, 57, HNDLBACK, 3, 0, 1, 58, EXAMINE, 3, 0, 1,
 59, FEEDT DN, 3, 0, 1, 60, ROUND IN, 3, 0, 1, 61, AMMO IN, 3, 0, 1*
 UHI, 62, COVER DN, 3, 0, 1, 63, FIRE, 3, 0, 1, 64, SAYS UP, 3, 0, 1,
 65, GO/NG 5, 1, 0, 1*
 UHI, 66, CHARGER, 3, 0, 1, 67, SAFE, 3, 0, 1, 68, COVER UP, 3, 0, 1,
 69, FEEDT UP, 3, 0, 1, 70, HNDLBACK, 3, 0, 1, 71, EXAMINE, 3, 0, 1,
 72, FEEDT DN, 3, 0, 1, 73, ROUND IN, 3, 0, 1, 74, AMMO IN, 3, 0, 1*
 UHI, 75, COVER DN, 3, 0, 1, 76, FIRE, 3, 0, 1, 77, SAYS UP, 3, 0, 1,
 78, GO/NG 6, 1, 0, 1*
 UHI, 79, TIME 1, , 80, TIME 2, , 81, TIME 3, , 82, TIME 4, , 83, TIME 5, ,

84, TIME 6*
 UPL, 1, TRIAL, 4, B, 1*
 UVA, 1, 1, 0, NUM GO, V, V, 0, 100*
 IMO, 2, A, 4, A*
 ISA, 2, SC, 1, 3, SC, 1, 5, SC, 1*
 LRE, 1, FIRST, 2, SECOND*
 TAS, 1, CHARGER, 0, 1, DS, 2, 0, 0, SD, (15)A, 1, 2*
 TAS, 2, SAFE, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 3, COVER UP, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 4, FEEDT UP, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 5, HNDLBACK, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 6, HOLDHNDL, 1, 1, SC, 41, 0, 0, NS, (15)A, 1*
 TAS, 7, EXAMINE, 1, 1, DS, 2, 0, 0, NS, (15)A, 2*
 TAS, 8, FEEDT DN, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 9, ROUND IN, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 10, AMMO IN, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 11, COVER DN, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 12, FIRE, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 13, SAYS UP, 1, 1, DS, 2, 0, 0, NS, (15)A, 1, 2*
 TAS, 50, ENDTRIAL, 1, 1, SC, 60, 0, 0, NS*
 TAS, 51, END SIM, 1, 9999, SC, 0, 0, 0, SI*
 STA, 1, H, STA*
 UTC, 1, 1, 0, 5, 1, -11, 0*
 UTC, 2, 2, 0, 5, 1, -11, 0*
 UTC, 3, 3, 0, 5, 1, -11, 0*
 UTC, 4, 4, 0, 5, 1, -11, 0*
 UTC, 5, 5, 0, 5, 1, -11, 0*
 UTC, 6, 0, 0, 5, 1, -11, 0*
 UTC, 7, 6, 0, 5, 1, -11, 0*
 UTC, 8, 7, 0, 5, 1, -11, 0*
 UTC, 9, 8, 0, 5, 1, -11, 0*
 UTC, 10, 9, 0, 5, 1, -11, 0*
 UTC, 11, 10, 0, 5, 1, -11, 0*
 UTC, 12, 11, 0, 5, 1, -11, 0*
 UTC, 13, 12, 0, 5, 1, -11, 0*
 UTC, 50, 0, 0, 5, 1, -11, 0*
 UTC, 51, 0, 0, 5, 1, -11, 0*
 MOD, 6, 2, D, T*
 MOD, 50, 2, D, T, 3, A, T, 4, D, T*
 MOD, 51, 2, D, T, 4, D, T*
 ATA, 2, COM, SA, , 5, SC, 0*
 ATA, 12, COM, SA, , 5, SC, 1*
 ATA, 50, COM, SA, , 3, UF, 5*
 DET, 1, 2*
 DET, 2, 3*
 DET, 3, 4*
 DET, 4, 5*
 DET, 5, 6, 7*
 DET, 7, 8*
 DET, 8, 9*
 DET, 9, 10*
 DET, 10, 11*
 DET, 11, 12*
 DET, 12, 13*
 DET, 13, 50*

CFI,50,1,ALA,4,3,SA,,51,AGA,4,3,SA*

TCL,7,6,0*

FIN*

5

12

.000

.00

.9915

.2164

8.216

8.970

5.725

17.06

.5501.5501.5501.5501.5501.5501.5501.5501.5501.5501.5501

6.6426.6426.6426.6426.6426.6426.6426.6426.6426.6426.642

3.9293.9293.9293.9293.9293.9293.9293.9293.9293.9293.929

3.8133.8133.8133.8133.8133.8133.8133.8133.8133.8133.813

2.3172.3172.3172.3172.3172.3172.3172.3172.3172.3172.317

1

NUMTR

NUMPM

STDEC

FRDEC

TCONS

TSTRN

TSTDV

P1TIM

P2TIM

P3TIM

TRLRN

SASMT

THRS0

THRS1

THRS2

MORE

GEN, STARTAN1, 12, 21, 1982, 21, 100, 1, 71268659, 1, N*
 POP, 2, 0, 0, 10, 4, , 2*
 OUT, 0, 0, (14) N, N, N, N*
 DIS, 1, NO, 0, -10, 10, 1
 DIS, 2, ER, 1, 0, 10, 1
 DIS, 3, CO, 0*
 DIS, 4, UN, , 100, 300*
 UBO, 1, PRES BRK, 2, PARK, 3, REL BRAK, 4, CK DRN V, 5, CKFUEL SO,
 6, CKFUELPU, 7, ELECEQOF, 8, MB SW ON, 9, FUEL LVL, 10, CK GEN,
 11, ACCEL DN, 12, STARTER, 13, GENBLOWR, 14, GO/NG 1*
 UBO, 15, PRES BRK, 16, PARK, 17, REL BRAK, 18, CK DRN V, 19, CKFUEL SO,
 20, CKFUELPU, 21, ELECEQOF, 22, MB SW ON, 23, FUEL LVL, 24, CK GEN,
 25, ACCEL DN, 26, STARTER, 27, GENBLOWR, 28, GO/NG 2*
 UBO, 29, PRES BRK, 30, PARK, 31, REL BRAK, 32, CK DRN V, 33, CKFUEL SO,
 34, CKFUELPU, 35, ELECEQOF, 36, MB SW ON, 37, FUEL LVL, 38, CK GEN,
 39, ACCEL DN, 40, STARTER, 41, GENBLOWR, 42, GO/NG 3*
 UBO, 43, PRES BRK, 44, PARK, 45, REL BRAK, 46, CK DRN V, 47, CKFUEL SO,
 48, CKFUELPU, 49, ELECEQOF, 50, MB SW ON, 51, FUEL LVL, 52, CK GEN,
 53, ACCEL DN, 54, STARTER, 55, GENBLOWR, 56, GO/NG 4*
 UBO, 57, PRES BRK, 58, PARK, 59, REL BRAK, 60, CK DRN V, 61, CKFUEL SO,
 62, CKFUELPU, 63, ELECEQOF, 64, MB SW ON, 65, FUEL LVL, 66, CK GEN,
 67, ACCEL DN, 68, STARTER, 69, GENBLOWR, 70, GO/NG 5*
 UBO, 71, PRES BRK, 72, PARK, 73, REL BRAK, 74, CK DRN V, 75, CKFUEL SO,
 76, CKFUELPU, 77, ELECEQOF, 78, MB SW ON, 79, FUEL LVL, 80, CK GEN,
 81, ACCEL DN, 82, STARTER, 83, GENBLOWR, 84, GO/NG 6*
 UBO, 85, TIME 1, 86, TIME 2, 87, TIME 3, 88, TIME 4, 89, TIME 5, 90, TIME 6*
 UHI, 1, PRES BRK, 3, 0, 1, 2, PARK, 3, 0, 1, 3, REL BRAK, 3, 0, 1,
 4, CK DRN V, 3, 0, 1, 5, CKFUEL SO, 3, 0, 1, 6, CKFUELPU, 3, 0, 1,
 7, ELECEQOF, 3, 0, 1, 8, MB SW ON, 3, 0, 1, 9, FUEL LVL, 3, 0, 1*
 UHI, 10, CK GEN, 3, 0, 1, 11, ACCEL DN, 3, 0, 1, 12, STARTER, 3, 0, 1,
 13, GENBLOWR, 3, 0, 1, 14, GO/NG 1, 1, 0, 1*
 UHI, 15, PRES BRK, 3, 0, 1, 16, PARK, 3, 0, 1, 17, REL BRAK, 3, 0, 1,
 18, CK DRN V, 3, 0, 1, 19, CKFUEL SO, 3, 0, 1, 20, CKFUELPU, 3, 0, 1,
 21, ELECEQOF, 3, 0, 1, 22, MB SW ON, 3, 0, 1, 23, FUEL LVL, 3, 0, 1*
 UHI, 24, CK GEN, 3, 0, 1, 25, ACCEL DN, 3, 0, 1, 26, STARTER, 3, 0, 1,
 27, GENBLOWR, 3, 0, 1, 28, GO/NG 2, 1, 0, 1*
 UHI, 29, PRES BRK, 3, 0, 1, 30, PARK, 3, 0, 1, 31, REL BRAK, 3, 0, 1,
 32, CK DRN V, 3, 0, 1, 33, CKFUEL SO, 3, 0, 1, 34, CKFUELPU, 3, 0, 1,
 35, ELECEQOF, 3, 0, 1, 36, MB SW ON, 3, 0, 1, 37, FUEL LVL, 3, 0, 1*
 UHI, 38, CK GEN, 3, 0, 1, 39, ACCEL DN, 3, 0, 1, 40, STARTER, 3, 0, 1,
 41, GENBLOWR, 3, 0, 1, 42, GO/NG 3, 1, 0, 1*
 UHI, 43, PRES BRK, 3, 0, 1, 44, PARK, 3, 0, 1, 45, REL BRAK, 3, 0, 1,
 46, CK DRN V, 3, 0, 1, 47, CKFUEL SO, 3, 0, 1, 48, CKFUELPU, 3, 0, 1,
 49, ELECEQOF, 3, 0, 1, 50, MB SW ON, 3, 0, 1, 51, FUEL LVL, 3, 0, 1*
 UHI, 52, CK GEN, 3, 0, 1, 53, ACCEL DN, 3, 0, 1, 54, STARTER, 3, 0, 1,
 55, GENBLOWR, 3, 0, 1, 56, GO/NG 4, 1, 0, 1*
 UHI, 57, PRES BRK, 3, 0, 1, 58, PARK, 3, 0, 1, 59, REL BRAK, 3, 0, 1,
 60, CK DRN V, 3, 0, 1, 61, CKFUEL SO, 3, 0, 1, 62, CKFUELPU, 3, 0, 1,
 63, ELECEQOF, 3, 0, 1, 64, MB SW ON, 3, 0, 1, 65, FUEL LVL, 3, 0, 1*
 UHI, 66, CK GEN, 3, 0, 1, 67, ACCEL DN, 3, 0, 1, 68, STARTER, 3, 0, 1,
 69, GENBLOWR, 3, 0, 1, 70, GO/NG 5, 1, 0, 1*
 UHI, 71, PRES BRK, 3, 0, 1, 72, PARK, 3, 0, 1, 73, REL BRAK, 3, 0, 1,
 74, CK DRN V, 3, 0, 1, 75, CKFUEL SO, 3, 0, 1, 76, CKFUELPU, 3, 0, 1,
 77, ELECEQOF, 3, 0, 1, 78, MB SW ON, 3, 0, 1, 79, FUEL LVL, 3, 0, 1*
 UHI, 80, CK GEN, 3, 0, 1, 81, ACCEL DN, 3, 0, 1, 82, STARTER, 3, 0, 1,

83,GENBLOWR,3,0,1,84, GO/NG 6,1,0,1*
 UHI,85,TIME 1,,,,86,TIME 2,,,,87,TIME 3,,,,88,TIME 4,,,,
 89,TIME 5,,,,90,TIME 6*
 ISA,2,SC,1,3,SC,1*
 LRE,1,DRIVER,2,TIMER*
 UPL,1,TRIAL,4,B,1*
 UVA,1,1,0,NUMGO,V,V,0,100*
 IMO,2,A,4,A*
 TAS,1, BEGIN,0,1,,,,,SO,(15) A,1*
 TAS,2,SETPKBRK,1,1,SC,0,0,0,(15) A,1*
 TAS,3,CK TRANS,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,4, PARK,1,1,SC,0,0,51,NS,(15) A,1*
 TAS,5,PRES BRK,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,6,RELBRAKE,2,2,DS,2,0,21,NS,(15) A,1*
 TAS,7,ELECEQOF,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,8,CK DRN V,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,9,FRONT DV,1,1,SC,0,0,0,NS,(15) A,1*
 TAS,10, CLOSE F,1,1,SC,0,0,51,NS,(15) A,1*
 TAS,11, REAR DV,1,1,SC,0,0,0,NS,(15) A,1*
 TAS,12, CLOSE R,1,1,SC,0,0,51,NS,(15) A,1*
 TAS,13,CKFUELSD,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,14,FSOFF ON,1,1,SC,0,0,21,NS,(15) A,1*
 TAS,15,CKFUELPU,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,16, PUMP ON,1,1,SC,0,0,21,NS,(15) A,1*
 TAS,17, CK GEN,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,18, GEN ON,1,1,SC,0,0,21,NS,(15) A,1*
 TAS,19,MB SW ON,6,6,DS,2,0,0,NS,(15) A,1*
 TAS,20,FUEL LVL,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,21,RT LEVEL,1,1,SC,0,0,0,NS,(15) A,1*
 TAS,22,LF LEVEL,1,1,SC,0,0,0,NS,(15) A,1*
 TAS,23,ACCEL DN,3,3,DS,2,0,0,NS,(15) A,1*
 TAS,24, STARTER,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,25, HOLD,1,1,DS,3,0,0,NS,(15) A,1*
 TAS,26,GENBLOWR,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,27, 15 SEC,1,1,SC,15,0,0,NS,(15) A,2*
 TAS,28,ACCEL UP,1,1,SC,0,0,0,NS,(15) A,1*
 TAS,29,WT3-5MIN,1,1,DS,4,0,0,NS,(15) A,1*
 TAS,50,ENDTRIAL,1,1,SC,60,0,0,NS*
 TAS,51, END SIM,1,9999,SC,0,0,0,SI*
 STA, 1,M,STA*
 UTC,1,0,0,5,1,-11,0*
 UTC,2,0,5,5,1,-11,0*
 UTC,3,2,35,5,1,-11,0*
 UTC,4,0,0,5,1,-11,0*
 UTC,5,1,35,5,1,-11,0*
 UTC,6,3,0,5,1,-11,0*
 UTC,7,7,5,5,1,-11,0*
 UTC,8,4,5,5,1,-11,0*
 UTC,9,0,35,5,1,-11,0*
 UTC,10,0,0,5,1,-11,0*
 UTC,11,0,35,5,1,-11,0*
 UTC,12,0,0,5,1,-11,0*
 UTC,13,5,5,5,1,-11,0*
 UTC,14,0,0,5,1,-11,0*
 UTC,15,6,5,5,1,-11,0*

UTC,16,0,0,5,1,-11,0*
 UTC,17,10,5,5,1,-11,0*
 UTC,18,0,0,5,1,-11,0*
 UTC,19,8,5,5,1,-11,0*
 UTC,20,9,5,5,1,-11,0*
 UTC,21,0,35,5,1,-11,0*
 UTC,22,0,35,5,1,-11,0*
 UTC,23,11,0,5,1,-11,0*
 UTC,24,12,0,5,1,-11,0*
 UTC,25,0,0,5,1,-11,0*
 UTC,26,13,0,5,1,-11,0*
 UTC,27,0,0,5,1,-11,0*
 UTC,28,0,0,5,1,-11,0*
 UTC,29,0,0,5,1,-11,0*
 UTC,50,0,0,5,1,-11,0*
 UTC,51,0,0,5,1,-11,0*
 MOD,1,2,D,T,4,D,T*
 MOD,2,2,D,T*
 MOD,4,2,D,T,4,D,T*
 MOD,9,2,D,T*
 MOD,10,2,D,T,4,D,T*
 MOD,11,2,D,T*
 MOD,12,2,D,T,4,D,T*
 MOD,14,2,D,T,4,D,T*
 MOD,16,2,D,T,4,D,T*
 MOD,18,2,D,T,4,D,T*
 MOD,21,2,D,T*
 MOD,22,2,D,T*
 MOD,25,2,D,T,4,D,T*
 MOD,27,2,D,T,4,D,T*
 MOD,28,2,D,T,4,D,T*
 MOD,29,2,D,T,4,D,T*
 MOD,50,2,D,T,3,A,T,4,D,T*
 MOD,51,2,D,T,4,D,T*
 ATA,1,COM,SA,,5,SC,1,SA,,6,SC,1,SA,,7,SC,1,SA,,8,SC,0,SA,,9,SC,0,SA,,10,SC,0*
 ATA,4,COM,SA,,5,SC,0*
 ATA,10,COM,SA,,6,SC,0*
 ATA,12,COM,SA,,7,SC,0*
 ATA,14,COM,SA,,8,SC,1*
 ATA,16,COM,SA,,9,SC,1*
 ATA,18,COM,SA,,10,SC,1*
 ATA,50,COM,SA,,3,UF,5*
 DET,1,2,7,8,13,15,17*
 DET,2,3,5*
 DET,4,6*
 DET,5,6*
 DET,6,19*
 DET,7,19*
 DET,8,9,11*
 DET,10,19*
 DET,12,19*
 DET,14,19*
 DET,15,19*
 DET,18,23*
 DET,19,20*

DET, 21, 23*

DET, 23, 24*

DET, 25, 26*

DET, 27, 23, 5

DET, 29, 23*

CFI,9,19,ALV,0,6,SA,,10,AGV,0,6,SA*

CFI,13,14,ALV,0,8,SA,,19,AGV,0,8,SA*

CFI,17,18,ALV,0,10,SA,,23,AGV,0,10,S

TCL, 25, 27, 0.0*

FINISH*

NUMTR

NUMPM

STDEC

STDEC
FRDEC

PRDC
TCONS

FLORIS
TSTRN

TSTDV

TSTDV
B4T3M

F1TIM
F2TIM

P2TIM
P3TIM

[illegible]

3.8323.8323.8323.8323.8323.8323.8323.8323.8323.8323.8323.832 THRS0

3.6783.6783.6783.6783.6783.6783.6783.6783.6783.6783.678 THRS1

2.0372.0372.0372.0372.0372.0372.0372.0372.0372.0372.0372.037 THRS2

1 MORE

B-8

FILE: STOPTANK VERS1 A1 DECISIONS AND DESIGNS, INC. - VM/SP R01 PLC04

```

GEN,STOPTAN1,12,21,1982,31,100,1,952713,1,N*
POP,1,0,0,5,4,,,2*
OUT,0,0,0,0,0,0,(14)N,N,N,N*
DIS,1,NO,0,-10,10,1*
DIS,2,ER,1,0,10,1*
UBO,1,PRESSBRK,2,CK TRANS,3,RELBRAKE,4,PRSS ACC,5,WAIT 5MN,6,RLAX ACC,
7,WAIT 3MN,8,ELECEQOF,9,STOP ENG,10,MB SW OF,11,GO/NG 1*
UBO,12,PRESSBRK,13,CK TRANS,14,RELBRAKE,15,PRSS ACC,16,WAIT 5MN,17,RLAX ACC,
18,WAIT 3MN,19,ELECEQOF,20,STOP ENG,21,MB SW OF,22,GO/NG 2*
UBO,23,PRESSBRK,24,CK TRANS,25,RELBRAKE,26,PRSS ACC,27,WAIT 5MN,28,RLAX ACC,
29,WAIT 3MN,30,ELECEQOF,31,STOP ENG,32,MB SW OF,33,GO/NG 3*
UBO,34,PRESSBRK,35,CK TRANS,36,RELBRAKE,37,PRSS ACC,38,WAIT 5MN,39,RLAX ACC,
40,WAIT 3MN,41,ELECEQOF,42,STOP ENG,43,MB SW OF,44,GO/NG 4*
UBO,45,PRESSBRK,46,CK TRANS,47,RELBRAKE,48,PRSS ACC,49,WAIT 5MN,50,RLAX ACC,
51,WAIT 3MN,52,ELECEQOF,53,STOP ENG,54,MB SW OF,55,GO/NG 5*
UBO,56,PRESSBRK,57,CK TRANS,58,RELBRAKE,59,PRSS ACC,60,WAIT 5MN,61,RLAX ACC,
62,WAIT 3MN,63,ELECEQOF,64,STOP ENG,65,MB SW OF,66,GO/NG 6*
UBO,67,TIME 1,68,TIME 2,69,TIME 3,70,TIME 4,71,TIME 5,72,TIME 6*
UHI,1,PRESSBRK,3,0,1,2,CK TRANS,3,0,1,3,RELBRAKE,3,0,1,4,PRSS ACC,3,0,1,
5,WAIT 5MN,3,0,1,6,RLAX ACC,3,0,1,7,WAIT 3MN,3,0,1,8,ELECEQOF,3,0,1*
UHI,9,STOP ENG,3,0,1,10,MB SW OF,3,0,1,11,GO/NG 1,1,0,1*
UHI,12,PRESSBRK,3,0,1,13,CK TRANS,3,0,1,14,RELBRAKE,3,0,1,15,PRSS ACC,3,0,1,
16,WAIT 5MN,3,0,1,17,RLAX ACC,3,0,1,18,WAIT 3MN,3,0,1,19,ELECEQOF,3,0,1*
UHI,20,STOP ENG,3,0,1,21,MB SW OF,3,0,1,22,GO/NG 2,1,0,1*
UHI,23,PRESSBRK,3,0,1,24,CK TRANS,3,0,1,25,RELBRAKE,3,0,1,26,PRSS ACC,3,0,1,
27,WAIT 5MN,3,0,1,28,RLAX ACC,3,0,1,29,WAIT 3MN,3,0,1,30,ELECEQOF,3,0,1*
UHI,31,STOP ENG,3,0,1,32,MB SW OF,3,0,1,33,GO/NG 3,1,0,1*
UHI,34,PRESSBRK,3,0,1,35,CK TRANS,3,0,1,36,RELBRAKE,3,0,1,37,PRSS ACC,3,0,1,
38,WAIT 5MN,3,0,1,39,RLAX ACC,3,0,1,40,WAIT 3MN,3,0,1,41,ELECEQOF,3,0,1*
UHI,42,STOP ENG,3,0,1,43,MB SW OF,3,0,1,44,GO/NG 4,1,0,1*
UHI,45,PRESSBRK,3,0,1,46,CK TRANS,3,0,1,47,RELBRAKE,3,0,1,48,PRSS ACC,3,0,1,
49,WAIT 5MN,3,0,1,50,RLAX ACC,3,0,1,51,WAIT 3MN,3,0,1,52,ELECEQOF,3,0,1*
UHI,53,STOP ENG,3,0,1,54,MB SW OF,3,0,1,55,GO/NG 5,1,0,1*
UHI,56,PRESSBRK,3,0,1,57,CK TRANS,3,0,1,58,RELBRAKE,3,0,1,59,PRSS ACC,3,0,1,
60,WAIT 5MN,3,0,1,61,RLAX ACC,3,0,1,62,WAIT 3MN,3,0,1,63,ELECEQOF,3,0,1*
UHI,64,STOP ENG,3,0,1,65,MB SW OF,3,0,1,66,GO/NG 6,1,0,1*
UHI,67,TIME 1,,,,68,TIME 2,,,,69,TIME 3,,,,70,TIME 4,,,,71,TIME 5,,,,
72,TIME 6*
UPL,1,TRIAL,4,B,1*
UVA,1,1,*,NUM GO,V,V,0,100*
IMO,2,A,4,A*
ISA,2,SC,1,3,SC,1,5,SC,1*
LRE,1,DRIVER*
TAS,1,PRESSBRK,1,1,DS,2,0,0,NS,(15) A,1*
TAS,2,CK TRANS,1,1,DS,2,0,0,NS,(15) A,1*
TAS,3,PARK,1,1,SC,0,0,21,NS,(15) A,1*
TAS,4,RELBRAKE,2,2,DS,2,0,21,NS,(15) A,1*
TAS,5,PRSS ACC,1,1,DS,2,0,0,NS,(15) A,1*
TAS,6,WAIT 5MN,1,1,DS,2,0,0,NS,(15) A,1*
TAS,7,RLAX ACC,1,1,DS,2,0,0,NS,(15) A,1*
TAS,8,WAIT 3MN,1,1,DS,2,0,0,NS,(15) A,1*
TAS,9,ELECEQOF,1,1,DS,2,0,0,NS,(15) A,1*
TAS,10,STOP ENG,1,1,DS,2,0,0,NS,(15) A,1*
TAS,11,MB SW OF,2,2,DS,2,0,0,NS,(15) A,1*
TAS,12,BEGIN ,0,1,SC,0,0,0,SD,(15) A,1*

```

TAS,50,ENDTRIAL,1,1,SC,60,0,0,NS*
 TAS,51,ENDSIM,1,9999,SC,0,0,0,SI*
 STA,12,M,STA*
 UTC,1,1,5,5,1,-11,0*
 UTC,2,2,5,5,1,-11,0*
 UTC,3,0,0,5,1,-11,0*
 UTC,4,3,0,5,1,-11,0*
 UTC,5,4,5,5,1,-11,0*
 UTC,6,5,5,5,1,-11,0*
 UTC,7,6,5,5,1,-11,0*
 UTC,8,7,5,5,1,-11,0*
 UTC,9,8,5,5,1,-11,0*
 UTC,10,9,5,5,1,-11,0*
 UTC,11,10,0,5,1,-11,0*
 UTC,12,0,0,5,1,-11,0*
 UTC,50,0,0,5,1,-11,0*
 .,0,0,5,1,-11,0*
 MOD,3,2,D,T,4,D,T*
 MOD,12,2,D,T,4,D,T*
 MOD,50,2,D,T,3,A,T,4,D,T*
 MOD,51,2,D,T,4,D,T*
 ATA,50,COM,SA,,3,UF,5*
 ATA,3,COM,SA,,5,SC,0*
 ATA,12,COM,SA,,5,SC,1*
 DET,12,1,2,9*
 DET,1,4*
 DET,3,4*
 DET,4,5*
 DET,5,6*
 DET,6,7*
 DET,7,8*
 DET,8,10*
 DET,9,11*
 DET,10,11*
 DET,11,50*
 CFI,50,12,ALA,4,3,SA,0,51,AGA,4,3,SA*
 CFI,2,4,ALV,0,5,SA,,3,AGV,0,5,SA*
 FIN*

5	NUMTR
10	NUMPM
.004	STDEC
.02	FRDEC
3.593	TCONS
4.276	TSTRN
29.86	TSTDV
18.54	P1TIM
5.078	P2TIM
.0000	P3TIM
.6482.6482.6482.6482.6482.6482.6482.6482.6482	
6.5566.5566.5566.5566.5566.5566.5566.5566.5566	
4.1214.1214.1214.1214.1214.1214.1214.1214.121	
3.9893.9893.9893.9893.9893.9893.9893.9893.989	
1.8371.8371.8371.8371.8371.8371.8371.8371.837	
1	MORE

TRLRN
 SASMT
 THRS0
 THRS1
 THRS2

GEN,GUNNERP1,12,22,1982,41,100,1,,1,N*
 FOP,4,0,2,4,4,,,2*
 OUT,0,0,(14) N,N,N,N*
 DIS,1,NO,0,-10,10,1,*
 DIS,2,ER,1,0,10,1*
 UBO, 1,PLUNGER, 2,RELEASE, 3,CK OIL, 4,SAY UNLK, 5,SAYPOWER,
 6,ELTRAVON, 7,ROTATE, 8,ELEVLOWR, 9,LK EYEPC, 10,ALINE +,
 11,PRSAZKNB, 12,ALINPTRS, 13,ZEROPTRS, 14,RLS KNOB, 15,MNTRV360,
 16,TRVALINE, 17,CK AZIND, 18,LK EYEPC, 19,TRAV RT, 20,TRAV LFT,
 21,SUDNSTOP, 22,ELTRVOFF, 23,TRVTURRT, 24,SAYPOWER*
 UBO, 25,ELTRV ON, 26,GO/NG 1*
 UBO, 27,PLUNGER, 28,RELEASE, 29,CK OIL, 30,SAY UNLK, 31,SAYPOWER,
 32,ELTRAVON, 33,ROTATE, 34,ELEVLOWR, 35,LK EYEPC, 36,ALINE +,
 37,PRSAZKNB, 38,ALINPTRS, 39,ZEROPTRS, 40,RLS KNOB, 41,MNTRV360,
 42,TRVALINE, 43,CK AZIND, 44,LK EYEPC, 45,TRAV RT, 46,TRAV LFT,
 47,SUDNSTOP, 48,ELTRVOFF, 49,TRVTURRT, 50,SAYPOWER*
 UBO, 51,ELTRV ON, 52,GO/NG 2*
 UBO, 53,PLUNGER, 54,RELEASE, 55,CK OIL, 56,SAY UNLK, 57,SAYPOWER,
 58,ELTRAVON, 59,ROTATE, 60,ELEVLOWR, 61,LK EYEPC, 62,ALINE +,
 63,PRSAZKNB, 64,ALINPTRS, 65,ZEROPTRS, 66,RLS KNOB, 67,MNTRV360,
 68,TRVALINE, 69,CK AZIND, 70,LK EYEPC, 71,TRAV RT, 72,TRAV LFT,
 73,SUDNSTOP, 74,ELTRVOFF, 75,TRVTURRT, 76,SAYPOWER*
 UBO, 77,ELTRV ON, 78,GO/NG 3*
 UBO, 79,PLUNGER, 80,RELEASE, 81,CK OIL, 82,SAY UNLK, 83,SAYPOWER,
 84,ELTRAVON, 85,ROTATE, 86,ELEVLOWR, 87,LK EYEPC, 88,ALINE +,
 89,PRSAZKNB, 90,ALINPTRS, 91,ZEROPTRS, 92,RLS KNOB, 93,MNTRV360,
 94,TRVALINE, 95,CK AZIND, 96,LK EYEPC, 97,TRAV RT, 98,TRAV LFT,
 99,SUDNSTOP, 100,ELTRVOFF, 101,TRVTURRT, 102,SAYPOWER*
 UBO, 103,ELTRV ON, 104,GO/NG 4*
 UBO, 105,PLUNGER, 106,RELEASE, 107,CK OIL, 108,SAY UNLK, 109,SAYPOWER,
 110,ELTRAVON, 111,ROTATE, 112,ELEVLOWR, 113,LK EYEPC, 114,ALINE +,
 115,PRSAZKNB, 116,ALINPTRS, 117,ZEROPTRS, 118,RLS KNOB, 119,MNTRV360,
 120,TRVALINE, 121,CK AZIND, 122,LK EYEPC, 123,TRAV RT, 124,TRAV LFT,
 125,SUDNSTOP, 126,ELTRVOFF, 127,TRVTURRT, 128,SAYPOWER*
 UBO, 129,ELTRV ON, 130,GO/NG 5*
 UBO, 131,PLUNGER, 132,RELEASE, 133,CK OIL, 134,SAY UNLK, 135,SAYPOWER,
 136,ELTRAVON, 137,ROTATE, 138,ELEVLOWR, 139,LK EYEPC, 140,ALINE +,
 141,PRSAZKNB, 142,ALINPTRS, 143,ZEROPTRS, 144,RLS KNOB, 145,MNTRV360,
 146,TRVALINE, 147,CK AZIND, 148,LK EYEPC, 149,TRAV RT, 150,TRAV LFT,
 151,SUDNSTOP, 152,ELTRVOFF, 153,TRVTURRT, 154,SAYPOWER*
 UBO, 155,ELTRV ON, 156,GO/NG 6*
 UBO, 157,TIME 1, 158,TIME 2, 159,TIME 3, 160,TIME 4, 161,TIME 5, 162,TIME 6*
 UHI, 1, PLUNGER, 3, 0, 1, 2, RELEASE, 3, 0, 1, 3, CK OIL, 3, 0, 1,
 4, SAY UNLK, 3, 0, 1, 5, SAYPOWER, 3, 0, 1, 6, ELTRAVON, 3, 0, 1,
 7, ROTATE, 3, 0, 1, 8, ELEVLOWR, 3, 0, 1, 9, LK EYEPC, 3, 0, 1*
 UHI, 10, ALINE +, 3, 0, 1, 11, PRSAZKNB, 3, 0, 1, 12, ALINPTRS, 3, 0, 1,
 13, ZEROPTRS, 3, 0, 1, 14, RLS KNOB, 3, 0, 1, 15, MNTRV360, 3, 0, 1,
 16, TRVALINE, 3, 0, 1, 17, CK AZIND, 3, 0, 1, 18, LK EYEPC, 3, 0, 1*
 UHI, 19, TRAV RT, 3, 0, 1, 20, TRAV LFT, 3, 0, 1, 21, SUDNSTOP, 3, 0, 1,
 22, ELTRVOFF, 3, 0, 1, 23, TRVTURRT, 3, 0, 1, 24, SAYPOWER, 3, 0, 1,
 25, ELTRV ON, 3, 0, 1, 26, GO/NG 1, 1, 0, 1*
 UHI, 27, PLUNGER, 3, 0, 1, 28, RELEASE, 3, 0, 1, 29, CK OIL, 3, 0, 1,
 30, SAY UNLK, 3, 0, 1, 31, SAYPOWER, 3, 0, 1, 32, ELTRAVON, 3, 0, 1,
 33, ROTATE, 3, 0, 1, 34, ELEVLOWR, 3, 0, 1, 35, LK EYEPC, 3, 0, 1*
 UHI, 36, ALINE +, 3, 0, 1, 37, PRSAZKNB, 3, 0, 1, 38, ALINPTRS, 3, 0, 1,

39,ZEROPTRS,3,0,1, 40,RLS KNOB,3,0,1, 41,MNTRV360,3,0,1,
 42,TRVALINE,3,0,1, 43,CK AZIND,3,0,1, 44,LK EYEPC,3,0,1*
 UHI, 45, TRAV RT,3,0,1, 46,TRAV LFT,3,0,1, 47,SUDNSTOP,3,0,1,
 48,ELTRVOFF,3,0,1, 49,TRVTURRT,3,0,1, 50,SAYPOWER,3,0,1,
 51,ELTRV ON,3,0,1, 52, GO/NG 2,1,0,1*
 UHI, 53, PLUNGER,3,0,1, 54, RELEASE,3,0,1, 55, CK OIL,3,0,1,
 56,SAY UNLK,3,0,1, 57,SAYPOWER,3,0,1, 58,ELTRAVON,3,0,1,
 59, ROTATE,3,0,1, 60,ELEVLOWR,3,0,1, 61,LK EYEPC,3,0,1*
 UHI, 62, ALINE +,3,0,1, 63,PRSAZKNB,3,0,1, 64,ALINPTRS,3,0,1,
 65,ZEROPTRS,3,0,1, 66,RLS KNOB,3,0,1, 67,MNTRV360,3,0,1,
 68,TRVALINE,3,0,1, 69,CK AZIND,3,0,1, 70,LK EYEPC,3,0,1*
 UHI, 71, TRAV RT,3,0,1, 72,TRAV LFT,3,0,1, 73,SUDNSTOP,3,0,1,
 74,ELTRVOFF,3,0,1, 75,TRVTURRT,3,0,1, 76,SAYPOWER,3,0,1,
 77,ELTRV ON,3,0,1, 78, GO/NG 3,1,0,1*
 UHI, 79, PLUNGER,3,0,1, 80, RELEASE,3,0,1, 81, CK OIL,3,0,1,
 82,SAY UNLK,3,0,1, 83,SAYPOWER,3,0,1, 84,ELTRAVON,3,0,1,
 85, ROTATE,3,0,1, 86,ELEVLOWR,3,0,1, 87,LK EYEPC,3,0,1*
 UHI, 88, ALINE +,3,0,1, 89,PRSAZKNB,3,0,1, 90,ALINPTRS,3,0,1,
 91,ZEROPTRS,3,0,1, 92,RLS KNOB,3,0,1, 93,MNTRV360,3,0,1,
 94,TRVALINE,3,0,1, 95,CK AZIND,3,0,1, 96,LK EYEPC,3,0,1*
 UHI, 97, TRAV RT,3,0,1, 98,TRAV LFT,3,0,1, 99,SUDNSTOP,3,0,1,
 100,ELTRVOFF,3,0,1, 101,TRVTURRT,3,0,1, 102,SAYPOWER,3,0,1,
 103,ELTRV ON,3,0,1, 104, GO/NG 4,1,0,1*
 UHI, 105, PLUNGER,3,0,1, 106, RELEASE,3,0,1, 107, CK OIL,3,0,1,
 108,SAY UNLK,3,0,1, 109,SAYPOWER,3,0,1, 110,ELTRAVON,3,0,1,
 111, ROTATE,3,0,1, 112,ELEVLOWR,3,0,1, 113,LK EYEPC,3,0,1*
 UHI, 114, ALINE +,3,0,1, 115,PRSAZKNB,3,0,1, 116,ALINPTRS,3,0,1,
 117,ZEROPTRS,3,0,1, 118,RLS KNOB,3,0,1, 119,MNTRV360,3,0,1,
 120,TRVALINE,3,0,1, 121,CK AZIND,3,0,1, 122,LK EYEPC,3,0,1*
 UHI, 123, TRAV RT,3,0,1, 124,TRAV LFT,3,0,1, 125,SUDNSTOP,3,0,1,
 126,ELTRVOFF,3,0,1, 127,TRVTURRT,3,0,1, 128,SAYPOWER,3,0,1,
 129,ELTRV ON,3,0,1, 130, GO/NG 5,1,0,1*
 UHI, 131, PLUNGER,3,0,1, 132, RELEASE,3,0,1, 133, CK OIL,3,0,1,
 134,SAY UNLK,3,0,1, 135,SAYPOWER,3,0,1, 136,ELTRAVON,3,0,1,
 137, ROTATE,3,0,1, 138,ELEVLOWR,3,0,1, 139,LK EYEPC,3,0,1*
 UHI, 140, ALINE +,3,0,1, 141,PRSAZKNB,3,0,1, 142,ALINPTRS,3,0,1,
 143,ZEROPTRS,3,0,1, 144,RLS KNOB,3,0,1, 145,MNTRV360,3,0,1,
 146,TRVALINE,3,0,1, 147,CK AZIND,3,0,1, 148,LK EYEPC,3,0,1*
 UHI, 149, TRAV RT,3,0,1, 150,TRAV LFT,3,0,1, 151,SUDNSTOP,3,0,1,
 152,ELTRVOFF,3,0,1, 153,TRVTURRT,3,0,1, 154,SAYPOWER,3,0,1,
 155,ELTRV ON,3,0,1, 156, GO/NG 6,1,0,1*
 UHI, 157, TIME 1, , , , 158, TIME 2, , , , 159, TIME 3, , , , 160, TIME 4, , , ,
 161, TIME 5, , , , 162, TIME 6*
 UPL,1, TRIAL,4,8,1*
 UVA,1,1,*,NUM GO,V,V,0,100*
 IMO,2,A,4,A*
 ISA,2,SC,1,3,SC,1*
 LRE,1,GUNNER,2,LOADER,3,DRIVER,4,TANK CDR*
 TAS, 1, PLUNGER,0,1,DS,2,, 0,SO,(15) A,1*
 TAS, 2, RELEASE,1,1,DS,2,, 0,NS,(15) A,1*
 TAS, 3, CK OIL,1,1,DS,2,, 0,NS,(15) A,1*
 TAS, 4,SAY UNLK,1,1,DS,2,, 0,NS,(15) A,1*
 TAS, 5,LDR UNLK,1,1,SC,0,, 0,NS,(15) A,2*
 TAS, 6,SAYPOWER,1,1,DS,2,, 0,NS,(15) A,1*
 TAS, 7,POWER UP,1,1,SC,0,, 0,NS,(15) A,3*

TAS, 8, ELTRAVON, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 9, ROTATE, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 10, ROTLEFT, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 11, ROTRIGHT, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 12, ELEVLOWR, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 13, ELEVATE, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 14, LOWER, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 15, TCHNDLOP, 4, 4, SC, 0, , 0, NS, (15) A, 4*
 TAS, 16, ACCURACY, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 17, LK EYEPC, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 18, ID AIMPT, 1, 1, SC, 0, , 21, NS, (15) A, 1, 4*
 TAS, 19, ALIGN +, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 20, PRSAZKHB, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 21, ALINPTRS, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 22, ZEROPTRS, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 23, RLS KNOB, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 24, MNTRV360, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 25, TRVALINE, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 26, SLIPPAGE, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 27, RIGHT, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 28, LEFT, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 29, LK EYEPC, 1, 1, DS, 2, , 51, NS, (15) A, 1*
 TAS, 30, TRAV RGT, 1, 1, DS, 2, , 51, NS, (15) A, 1*
 TAS, 31, TRAV LFT, 1, 1, DS, 2, , 51, NS, (15) A, 1*
 TAS, 32, SUDNSTOP, 1, 1, DS, 2, , 51, NS, (15) A, 1*
 TAS, 33, ELTRVOFF, 1, 1, DS, 2, , 51, NS, (15) A, 1*
 TAS, 34, TRVTURRT, 1, 1, DS, 2, , 51, NS, (15) A, 1*
 TAS, 35, SAYPOWER, 1, 1, DS, 2, , 51, NS, (15) A, 1*
 TAS, 36, ELTRAVON, 1, 1, DS, 2, , 51, NS, (15) A, 1*
 TAS, 37, CK AZIND, 1, 1, DS, 2, , 61, NS, (15) A, 1*
 TAS, 38, TELL TC, 1, 1, SC, 0, , 61, NS, (15) A, 1*
 TAS, 39, CONTINUE, 3, 3, SC, 0, , 0, NS, (15) A, 1*
 TAS, 50, ENDTRIAL, 1, 1, SC, 60, 0, 0, NS*
 TAS, 51, END SIM, 1, 9999, SC, 0, 0, 0, SI*
 STA, 1, M, STA*
 UTC, 1, 1, 0, 5, 1, -11, 0*
 UTC, 2, 2, 0, 5, 1, -11, 0*
 UTC, 3, 3, 0, 5, 1, -11, 0*
 UTC, 4, 4, 0, 5, 1, -11, 0*
 UTC, 5, 0, 0, 5, 1, -11, 0*
 UTC, 6, 5, 0, 5, 1, -11, 0*
 UTC, 7, 0, 0, 5, 1, -11, 0*
 UTC, 8, 6, 0, 5, 1, -11, 0*
 UTC, 9, 7, 5, 5, 1, -11, 0*
 UTC, 10, 0, 35, 5, 1, -11, 0*
 UTC, 11, 0, 35, 5, 1, -11, 0*
 UTC, 12, 8, 5, 5, 1, -11, 0*
 UTC, 13, 0, 35, 5, 1, -11, 0*
 UTC, 14, 0, 35, 5, 1, -11, 0*
 UTC, 15, 0, 0, 5, 1, -11, 0*
 UTC, 16, 0, 5, 5, 1, -11, 0*
 UTC, 17, 9, 0, 5, 1, -11, 0*
 UTC, 18, 0, 0, 5, 1, -11, 0*
 UTC, 19, 10, 0, 5, 1, -11, 0*
 UTC, 20, 11, 0, 5, 1, -11, 0*

UTC,21,12, 0,5,1,-11,0*
 UTC,22,13, 0,5,1,-11,0*
 UTC,23,14, 0,5,1,-11,0*
 UTC,24,15, 0,5,1,-11,0*
 UTC,25,16, 0,5,1,-11,0*
 UTC,26, 0, 5,5,1,-11,0*
 UTC,27, 0,35,5,1,-11,0*
 UTC,28, 0,35,5,1,-11,0*
 UTC,29,18, 0,5,1,-11,0*
 UTC,30,19, 0,5,1,-11,0*
 UTC,31,20, 0,5,1,-11,0*
 UTC,32,21, 0,5,1,-11,0*
 UTC,33,22, 0,5,1,-11,0*
 UTC,34,23, 0,5,1,-11,0*
 UTC,35,24, 0,5,1,-11,0*
 UTC,36,25, 0,5,1,-11,0*
 UTC,37,17, 0,5,1,-11,0*
 UTC,38, 0, 0,5,1,-11,0*
 UTC,39, 0, 0,5,1,-11,0*
 UTC,50, 0, 0,5,1,-11,0*
 UTC,51, 0, 0,5,1,-11,0*
 MOD,5,2,D,T,4,D,T*
 MOD,7,2,D,T,4,D,T*
 MOD,10,2,D,T*
 MOD,11,2,D,T*
 MOD,13,2,D,T*
 MOD,14,2,D,T*
 MOD,15,2,D,T,4,D,T*
 MOD,16,2,D,T*
 MOD,26,2,D,T*
 MOD,27,2,D,T*
 MOD,28,2,D,T*
 MOD,38,2,D,T,4,D,T*
 MOD,39,2,D,T,4,D,T*
 MOD,50,2,D,T,3,A,T,4,D,T*
 MOD,51,2,D,T,4,D,T*
 ATA, 1,COM,SA,,1,SC,0*
 ATA,27,COM,IA,,2,SC,1,SA,,1,UF,1*
 ATA,28,COM,IA,,2,SC,2,SA,,1,UF,1*
 ATA,50,COM,SA,,3,UF,5,IA,,1,SC,0*
 DET,1,2*
 DET,2,3*
 DET,3,4*
 DET,4,5*
 DET,5,6*
 DET,6,7*
 DET,7,8*
 DET,8,9,12*
 DET,9,10,11*
 DET,10,15*
 DET,11,15*
 DET,12,13,14*
 DET,13,15*
 DET,14,15*
 DET,15,16,26*

GEN,LOADERP1,12,22,1892,51,100,1,,1,N*
 POP,4,0,0,4,4,(9) 2*
 OUT,0,(14) N,N,N,N*
 DIS,1,NO,0,-10,10,1*
 DIS,2,ER,1,0,10,1*
 UBO, 1,CLS BRCH, 2,TESTR IN, 3, SW FIRE, 4, SAYS UP, 5,TELL GNR,
 6, TELL TC, 7, SW SAFE, 8,TELL GNR, 9,SAYNOGUN,10,TESTROUT,
 11, GO/NG 1*
 UBO,12,CLS BRCH,13,TESTR IN,14, SW FIRE,15, SAYS UP,16,TELL GNR,
 17, TELL TC,18, SW SAFE,19,TELL GNR,20,SAYNOGUN,21,TESTROUT,
 22, GO/NG 2*
 UBO,23,CLS BRCH,24,TESTR IN,25, SW FIRE,26, SAYS UP,27,TELL GNR,
 28, TELL TC,29, SW SAFE,30,TELL GNR,31,SAYNOGUN,32,TESTROUT,
 33, GO/NG 3*
 UBO,34,CLS BRCH,35,TESTR IN,36, SW FIRE,37, SAYS UP,38,TELL GNR,
 39, TELL TC,40, SW SAFE,41,TELL GNR,42,SAYNOGUN,43,TESTROUT,
 44, GO/NG 4*
 UBO,45,CLS BRCH,46,TESTR IN,47, SW FIRE,48, SAYS UP,49,TELL GNR,
 50, TELL TC,51, SW SAFE,52,TELL GNR,53,SAYNOGUN,54,TESTROUT,
 55, GO/NG 5*
 UBO,56,CLS BRCH,57,TESTR IN,58, SW FIRE,59, SAYS UP,60,TELL GNR,
 61, TELL TC,62, SW SAFE,63,TELL GNR,64,SAYNOGUN,65,TESTROUT,
 66, GO/NG 6*
 UBO,67,TIME 1,68,TIME 2,69,TIME 3,70,TIME 4,71,TIME 5,72,TIME 6*
 UHI, 1,CLS BRCH,3,0,1, 2,TESTR IN,3,0,1, 3, SW FIRE,3,0,1,
 4, SAYS UP,3,0,1, 5,TELL GNR,3,0,1, 6, TELL TC,3,0,1,
 7, SW SAFE,3,0,1, 8,TELL GNR,3,0,1, 9,SAYNOGUN,3,0,1*
 UHI,10,TESTROUT,3,0,1,11, GO/NG 1,1,0,1*
 UHI,12,CLS BRCH,3,0,1,13,TESTR IN,3,0,1,14, SW FIRE,3,0,1,
 14, SAYS UP,3,0,1,16,TELL GNR,3,0,1,17, TELL TC,3,0,1,
 18, SW SAFE,3,0,1,19,TELL GNR,3,0,1,20,SAYNOGUN,3,0,1*
 UHI,21,TESTROUT,3,0,1,22, GO/NG 2,1,0,1*
 UHI,23,CLS BRCH,3,0,1,24,TESTR IN,3,0,1,25, SW FIRE,3,0,1,
 26, SAYS UP,3,0,1,27,TELL GNR,3,0,1,28, TELL TC,3,0,1,
 29, SW SAFE,3,0,1,30,TELL GNR,3,0,1,31,SAYNOGUN,3,0,1*
 UHI,32,TESTROUT,3,0,1,33, GO/NG 3,1,0,1*
 UHI,34,CLS BRCH,3,0,1,35,TESTR IN,3,0,1,36, SW FIRE,3,0,1,
 37, SAYS UP,3,0,1,38,TELL GNR,3,0,1,39, TELL TC,3,0,1,
 40, SW SAFE,3,0,1,41,TELL GNR,3,0,1,42,SAYNOGUN,3,0,1*
 UHI,43,TESTROUT,3,0,1,44, GO/NG 4,1,0,1*
 UHI,45,CLS BRCH,3,0,1,46,TESTR IN,3,0,1,47, SW FIRE,3,0,1,
 48, SAYS UP,3,0,1,49,TELL GNR,3,0,1,50, TELL TC,3,0,1,
 51, SW SAFE,3,0,1,52,TELL GNR,3,0,1,53,SAYNOGUN,3,0,1*
 UHI,54,TESTROUT,3,0,1,55, GO/NG 5,1,0,1*
 UHI,56,CLS BRCH,3,0,1,57,TESTR IN,3,0,1,58, SW FIRE,3,0,1,
 59, SAYS UP,3,0,1,60,TELL GNR,3,0,1,61, TELL TC,3,0,1,
 62, SW SAFE,3,0,1,63,TELL GNR,3,0,1,64,SAYNOGUN,3,0,1*
 UHI,65,TESTROUT,3,0,1,66, GO/NG 6,1,0,1*
 UHI,67,TIME 1,,,,68,TIME 2,,,,69,TIME 3,,,,70,TIME 4,,,,
 71,TIME 5,,,,72,TIME 6*
 UPL,1,TRIAL,4,B,1*
 UVA,1,1,*,NUM GO,V,V,0,100*
 IMO,2,A,4,A*
 ISA,2,SC,1,3,SC,1*
 LRE,1,LOADER,2,GUNNER,3,DRIVER,4,TANK CDR

TAS, 1,CLS BRCH,0,1,DS, 2,, 0,SO,(15) A,1*
 TAS, 2,TESTR IN,1,1,DS, 2,, 0,NS,(15) A,1*
 TAS, 3, SW FIRE,1,1,DS, 2,, 0,NS,(15) A,1*
 TAS, 4, SAYS UP,1,1,DS, 2,, 0,NS,(15) A,1*
 TAS, 5,MB SW ON,1,1,SC, 0,, 0,NS,(15) A,3*
 TAS, 6, GUN ON,1,1,SC, 0,, 0,NS,(15) A,2*
 TAS, 7,TCPALMSW,1,1,SC, 0,, 0,NS,(15) A,4*
 TAS, 8,TELL GNR,1,1,DS, 2,, 0,NS,(15) A,1*
 TAS, 9,ONTHEWAY,1,1,SC, 0,,21,NS,(15) A,1,2*
 TAS,10, TRIGGER,1,1,SC, 0,,21,NS,(15) A,2*
 TAS,11, TELL TC,1,1,DS, 2,, 0,NS,(15) A,1*
 TAS,12,ONTHEWAY,1,1,SC, 0,,21,NS,(15) A,1,4*
 TAS,13, TRIGGER,1,1,SC, 0,,21,NS,(15) A,4*
 TAS,14, SW SAFE,2,2,DS, 2,, 0,NS,(15) A,1*
 TAS,15,TELL GNR,1,1,DS, 2,, 0,NS,(15) A,1*
 TAS,16,ONTHEWAY,1,1,SC, 0,, 0,NS,(15) A,2*
 TAS,17, TRIGGER,1,1,SC, 0,, 0,NS,(15) A,2*
 TAS,18,WATCH LT,1,1,SC,11,,21,NS,(15) A,1*
 TAS,19,SAYNOGUN,1,1,DS, 2,, 0,NS,(15) A,1*
 TAS,20, GUN OFF,1,1,SC, 0,, 0,NS,(15) A,2*
 TAS,21,TESTROUT,1,1,DS, 2,, 0,NS,(15) A,1*
 TAS,50,ENDTRIAL,2,2,SC,60,, 0,NS*
 TAS,51, END SIM,1,9999,SC,0,,0,SI*
 STA,1,M,STA*
 UTC, 1, 1,0,5,1,-11,0*
 UTC, 2, 2,0,5,1,-11,0*
 UTC, 3, 3,0,5,1,-11,0*
 UTC, 4, 4,0,5,1,-11,0*
 UTC, 5, 0,0,5,1,-11,0*
 UTC, 6, 0,0,5,1,-11,0*
 UTC, 7, 0,0,5,1,-11,0*
 UTC, 8, 5,5,5,1,-11,0*
 UTC, 9, 0,0,5,1,-11,0*
 UTC,10, 0,0,5,1,-11,0*
 UTC,11, 6,5,5,1,-11,0*
 UTC,12, 0,0,5,1,-11,0*
 UTC,13, 0,0,5,1,-11,0*
 UTC,14, 7,0,5,1,-11,0*
 UTC,15, 8,0,5,1,-11,0*
 UTC,16, 0,0,5,1,-11,0*
 UTC,17, 0,0,5,1,-11,0*
 UTC,18, 0,0,5,1,-11,0*
 UTC,19, 9,0,5,1,-11,0*
 UTC,20, 0,0,5,1,-11,0*
 UTC,21,10,0,5,1,-11,0*
 UTC,50, 0,0,5,1,-11,0*
 UTC,51, 0,0,5,1,-11,0*
 MOD,5,2,D,T,4,D,T*
 MOD,6,2,D,T,4,D,T*
 MOD,7,2,D,T,4,D,T*
 MOD,9,2,D,T,4,D,T*
 MOD,10,2,D,T,4,D,T*
 MOD,12,2,D,T,4,D,T*
 MOD,13,2,D,T,4,D,T*
 MOD,16,2,D,T,4,D,T*

MOD,17,2,D,T,4,D,T*
 MOD,18,2,D,T,4,D,T*
 MOD,20,2,D,T,4,D,T*
 MOD,50,2,D,T,3,A,T,4,D,T*
 MOD,51,2,D,T,4,D,T*
 ATA,8,STA,SA,0,1,SC,0*
 ATA,9,STA,SA,0,1,UF,1*
 ATA,15,STA,SA,0,1,SC,0*
 ATA,16,STA,SA,0,1,UF,1*
 ATA,50,COM,SA,,3,UF,5,SA,,1,SC,0*
 DET,1,2*
 DET,2,3*
 DET,3,4*
 DET,4,5*
 DET,5,6*
 DET,6,7,18*
 DET,7,8,11*
 DET,8,9*
 DET,9,10,18*
 DET,11,12*
 DET,12,13,18*
 DET,13,14*
 DET,14,15*
 DET,15,16*
 DET,16,17,18*
 DET,19,20,21*
 DET,20,50*
 DET,21,50*
 CFI,10,9,ALV,3,1,SA,,14,AGV,3,1,SA*
 CFI,17,16,ALV,1,1,SA,,19,AGV,1,1,SA*
 CFI,50,1,ALA,4,3,SA,0,51,AGA,4,3,SA*
 TCL,7,18,0*
 TCL,10,18,0*
 TCL,13,18,0*
 TCL,17,18,0*
 FINISH*
 2
 10
 .004
 .02
 10.02
 1.02
 5.02
 10.02
 10.02
 10.02
 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0
 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0
 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 1
 NUMTR
 NUMPM
 STDEC
 FRDEC
 TCONS
 TSTRN
 TSTDV
 P1TIM
 P2TIM
 P3TIM
 TRLRN
 SASMT
 THRS0
 THRS1
 THRS2
 MORE

GEN,PREC FIR,12,23,1982,61, 2,1,71268659,1,N*
 POP,2,0,0,10,5,,,2*
 OUT,1,1,(14)N,N,N,N*
 DIS,1,NO,0,-10,10,1*
 DIS,2,ER,1,0,10,1*
 UBO, 1, GUN ON, 2,CHK COMP, 3, SAY ID, 4,TELESCOP, 5,PERISCOP,
 6, CK HEAT, 7,SABOTHEP, 8,CENTER +, 9, 5.0 MIL,10, 2.5 MIL,
 11, 7.5 MIL,12,ONTHEWAY,13, GO/NG 1*
 UBO,14, GUN ON,15,CHK COMP,16, SAY ID,17,TELESCOP,18,PERISCOP,
 19, CK HEAT,20,SABOTHEP,21,CENTER +,22, 5.0 MIL,23, 2.5 MIL,
 24, 7.5 MIL,25,ONTHEWAY,26, GO/NG 2*
 UBO,27, GUN ON,28,CHK COMP,29, SAY ID,30,TELESCOP,31,PERISCOP,
 32, CK HEAT,33,SABOTHEP,34,CENTER +,35, 5.0 MIL,36, 2.5 MIL,
 37, 7.5 MIL,38,ONTHEWAY,39, GO/NG 3*
 UBO,40, GUN ON,41,CHK COMP,42, SAY ID,43,TELESCOP,44,PERISCOP,
 45, CK HEAT,46,SABOTHEP,47,CENTER +,48, 5.0 MIL,49, 2.5 MIL,
 50, 7.5 MIL,51,ONTHEWAY,52, GO/NG 4*
 UBO,53, GUN ON,54,CHK COMP,55, SAY ID,56,TELESCOP,57,PERISCOP,
 58, CK HEAT,59,SABOTHEP,60,CENTER +,61, 5.0 MIL,62, 2.5 MIL,
 63, 7.5 MIL,64,ONTHEWAY,65, GO/NG 5*
 UBO,66, GUN ON,67,CHK COMP,68, SAY ID,69,TELESCOP,70,PERISCOP,
 71, CK HEAT,72,SABOTHEP,73,CENTER +,74, 5.0 MIL,75, 2.5 MIL,
 76, 7.5 MIL,77,ONTHEWAY,78, GO/NG 6*
 UBO,79, TIME 1,80, TIME 2,81, TIME 3,82, TIME 4,83, TIME 5,84, TIME 6*
 UHI, 1, GUN ON,3,0,1, 2,CHK COMP,3,0,1, 3, SAY ID,3,0,1,
 4,TELESCOP,3,0,1, 5,PERISCOP,3,0,1, 6, CK HEAT,3,0,1,
 7,SABOTHEP,3,0,1, 8,CENTER +,3,0,1, 9, 5.0 MIL,3,0,1*
 UHI,10, 2.5 MIL,3,0,1,11, 7.5 MIL,3,0,1,12,ONTHEWAY,3,0,1,
 13, GO/NG 1,1,0,1*
 UHI,14, GUN ON,3,0,1,15,CHK COMP,3,0,1,16, SAY ID,3,0,1,
 17,TELESCOP,3,0,1,18,PERISCOP,3,0,1,19, CK HEAT,3,0,1,
 20,SABOTHEP,3,0,1,21,CENTER +,3,0,1,22, 5.0 MIL,3,0,1*
 UHI,23, 2.5 MIL,3,0,1,24, 7.5 MIL,3,0,1,25,ONTHEWAY,3,0,1,
 26, GO/NG 2,1,0,1*
 UHI,27, GUN ON,3,0,1,28,CHK COMP,3,0,1,29, SAY ID,3,0,1,
 30,TELESCOP,3,0,1,31,PERISCOP,3,0,1,32, CK HEAT,3,0,1,
 33,SABOTHEP,3,0,1,34,CENTER +,3,0,1,35, 5.0 MIL,3,0,1*
 UHI,36, 2.5 MIL,3,0,1,37, 7.5 MIL,3,0,1,38,ONTHEWAY,3,0,1,
 39, GO/NG 3,1,0,1*
 UHI,40, GUN ON,3,0,1,41,CHK COMP,3,0,1,42, SAY ID,3,0,1,
 43,TELESCOP,3,0,1,44,PERISCOP,3,0,1,45, CK HEAT,3,0,1,
 46,SABOTHEP,3,0,1,47,CENTER +,3,0,1,48, 5.0 MIL,3,0,1*
 UHI,49, 2.5 MIL,3,0,1,50, 7.5 MIL,3,0,1,51,ONTHEWAY,3,0,1,
 52, GO/NG 4,1,0,1*
 UHI,53, GUN ON,3,0,1,54,CHK COMP,3,0,1,55, SAY ID,3,0,1,
 56,TELESCOP,3,0,1,57,PERISCOP,3,0,1,58, CK HEAT,3,0,1,
 59,SABOTHEP,3,0,1,60,CENTER +,3,0,1,61, 5.0 MIL,3,0,1*
 UHI,62, 2.5 MIL,3,0,1,63, 7.5 MIL,3,0,1,64,ONTHEWAY,3,0,1,
 65, GO/NG 5,1,0,1*
 UHI,66, GUN ON,3,0,1,67,CHK COMP,3,0,1,68, SAY ID,3,0,1,
 69,TELESCOP,3,0,1,70,PERISCOP,3,0,1,71, CK HEAT,3,0,1,
 72,SABOTHEP,3,0,1,73,CENTER +,3,0,1,74, 5.0 MIL,3,0,1*
 UHI,75, 2.5 MIL,3,0,1,76, 7.5 MIL,3,0,1,77,ONTHEWAY,3,0,1,
 78, GO/NG 6,1,0,1*
 UHI,79, TIME 1,,,80, TIME 2,,,81, TIME 3,,,82, TIME 4,,,

83, TIME 5, , , , 84, TIME 6*
 UPL, 1, TRIAL, 4, B, 1*
 UVA, 1, 1, 8, NUM GO, V, V, 0, 100*
 IMO, 2, A, 4, A*
 ISA, 2, SC, 1, 3, SC, 1*
 ISA, 8, SC, 3, 9, SC, 1, 10, SC, 0, 5, SC, 2, 6, SC, 2*
 LRE, 1, GUNNER, 2, LOADER*
 TAS, 1, BEGIN, 0, 1, SC, 0, , 0, SO, (15) A, 1*
 TAS, 2, GUN ON, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 3, CHK COMP, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 4, CHS SITE, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 5, INDX AMO, 1, 1, SC, 0, , 21, NS, (15) A, 1*
 TAS, 6, TELESCOP, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 7, PERISCOP, 1, 1, DS, 2, , 21, NS, (15) A, 1*
 TAS, 8, SAY ID, 3, 3, DS, 2, , 0, NS, (15) A, 1*
 TAS, 9, CK HEAT, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 10, SABOTHEP, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 11, CHNG RET, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 12, CENTER +, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 13, RANGE LN, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 14, CNTR LN, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 15, SET LEAD, 1, 1, SC, 0, , 0, NS, (15) A, 1*
 TAS, 16, 5.0 MIL, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 17, 2.5 MIL, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 18, 7.5 MIL, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 19, ONTHEWAY, 1, 1, DS, 2, , 0, NS, (15) A, 1*
 TAS, 20, FIRES, 2, 2, SC, 0, , 0, NS, (15) A, 1*
 TAS, 21, WAIT, 1, 1, DS, 2, , 0, NS, (15) A, 2*
 TAS, 22, SAY UP, 1, 1, SC, 0, , 0, NS, (15) A, 2*
 TAS, 50, ENDTRIAL, 1, 1, SC, 60, 0, 0, NS*
 TAS, 51, END SIM, 1, 1, SC, 0, 0, 0, SI*
 STA, 1, M, STA*
 UTC, 1, 0, 0, 5, 1, -11, 0*
 UTC, 2, 1, 5, 5, 1, -11, 0*
 UTC, 3, 2, 5, 5, 1, -11, 0*
 UTC, 4, 0, 5, 5, 1, -11, 0*
 UTC, 5, 0, 0, 5, 1, -11, 0*
 UTC, 6, 4, 0, 5, 1, -11, 0*
 UTC, 7, 5, 0, 5, 1, -11, 0*
 UTC, 8, 3, 0, 5, 1, -11, 0*
 UTC, 9, 6, 0, 5, 1, -11, 0*
 UTC, 10, 7, 0, 5, 1, -11, 0*
 UTC, 11, 0, 0, 5, 1, -11, 0*
 UTC, 12, 8, 0, 5, 1, -11, 0*
 UTC, 13, 0, 0, 5, 1, -11, 0*
 UTC, 14, 0, 0, 5, 1, -11, 0*
 UTC, 15, 0, 0, 5, 1, -11, 0*
 UTC, 16, 9, 0, 5, 1, -11, 0*
 UTC, 17, 10, 0, 5, 1, -11, 0*
 UTC, 18, 11, 0, 5, 1, -11, 0*
 UTC, 19, 12, 0, 5, 1, -11, 0*
 UTC, 20, 0, 0, 5, 1, -11, 0*
 UTC, 21, 0, 0, 5, 1, -11, 0*
 UTC, 22, 0, 0, 5, 1, -11, 0*
 UTC, 50, 0, 0, 5, 1, -11, 0*

GEN,RADIONS1,12,23,1982,71, 2,1,71268659,1,N*
 POP,2,0,2,5,4,,,2*
 OUT,0,0,(14)N,N,N,N*
 DIS,1,NO,0,-10,10,1*
 DIS,2,ER,1,0,10,1*
 UBO, 1, SWITCH, 2,NCS CALL, 3,OWN CALL, 4,3RTN MSG, 5, MESSAGE,
 6, BREAK, 7, OVER, 8, GO/NG 1*
 UBO, 9, SWITCH,10,NCS CALL,11,OWN CALL,12,3RTN MSG,13, MESSAGE,
 14, BREAK,15, OVER,16, GO/NG 2*
 UBO,17, SWITCH,18,NCS CALL,19,OWN CALL,20,3RTN MSG,21, MESSAGE,
 22, BREAK,23, OVER,24, GO/NG 3*
 UBO,25, SWITCH,26,NCS CALL,27,OWN CALL,28,3RTN MSG,29, MESSAGE,
 30, BREAK,31, OVER,32, GO/NG 4*
 UBO,33, SWITCH,34,NCS CALL,35,OWN CALL,36,3RTN MSG,37, MESSAGE,
 38, BREAK,39, OVER,40, GO/NG 5*
 UBO,41, SWITCH,42,NCS CALL,43,OWN CALL,44,3RTN MSG,45, MESSAGE,
 46, BREAK,47, OVER,48, GO/NG 6*
 UBO,49,TIME 1,50,TIME 2,51,TIME 3,52,TIME 4,53,TIME 5,54,TIME 6*
 UHI, 1, SWITCH,3,0,1, 2,NCS CALL,3,0,1, 3,OWN CALL,3,0,1,
 4,3RTN MSG,3,0,1, 5, MESSAGE,3,0,1, 6, BREAK,3,0,1,
 7, OVER,3,0,1, 8, GO/NG 1,1,0,1*
 UHI, 9, SWITCH,3,0,1,10,NCS CALL,3,0,1,11,OWN CALL,3,0,1,
 12,3RTN MSG,3,0,1,13, MESSAGE,3,0,1,14, BREAK,3,0,1,
 15, OVER,3,0,1,16, GO/NG 2,1,0,1*
 UHI,17, SWITCH,3,0,1,18,NCS CALL,3,0,1,19,OWN CALL,3,0,1,
 20,3RTN MSG,3,0,1,21, MESSAGE,3,0,1,22, BREAK,3,0,1,
 23, OVER,3,0,1,24, GO/NG 3,1,0,1*
 UHI,25, SWITCH,3,0,1,26,NCS CALL,3,0,1,27,OWN CALL,3,0,1,
 28,3RTN MSG,3,0,1,29, MESSAGE,3,0,1,30, BREAK,3,0,1,
 31, OVER,3,0,1,32, GO/NG 4,1,0,1*
 UHI,33, SWITCH,3,0,1,34,NCS CALL,3,0,1,35,OWN CALL,3,0,1,
 36,3RTN MSG,3,0,1,37, MESSAGE,3,0,1,38, BREAK,3,0,1,
 39, OVER,3,0,1,40, GO/NG 5,1,0,1*
 UHI,41, SWITCH,3,0,1,42,NCS CALL,3,0,1,43,OWN CALL,3,0,1,
 44,3RTN MSG,3,0,1,45, MESSAGE,3,0,1,46, BREAK,3,0,1,
 47, OVER,3,0,1,48, GO/NG 6,1,0,1*
 UHI,48,TIME 1,,,,50,TIME 2,,,,51,TIME 3,,,,52,TIME 4,,,,
 53,TIME 5,,,,54,TIME 6*
 UPL,1,TRIAL,4,B,1*
 UVA,1,1,*,NUM GO,V,V,0,100*
 IMO,2,A,4,A*
 ISA,1,SC,0,5,SC,3,2,SC,1,3,SC,1*
 LRE,1,SOLDIER,2,NCS*
 TAS, 1, SWITCH,0,1,DS,2,0,0,SO,(15) A,1*
 TAS, 2,NCS CALL,1,1,DS,2,0,0,NS,(15) A,1*
 TAS, 3,OWN CALL,1,1,DS,2,0,0,NS,(15) A,1*
 TAS, 4, OVER,1,1,DS,2,0,0,NS,(15) A,1*
 TAS, 5,3RTN MSG,1,1,DS,2,0,0,NS,(15) A,1*
 TAS, 6, MSG FOL,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 7, MSG ID,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 8, MESSAGE,1,1,DS,2,0,0,NS,(15) A,1*
 TAS, 9, BREAK,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,10, RESP 1,1,1,SC,0,0,0,NS,(15) A,2*
 TAS,11, RESP 2,1,1,SC,0,0,0,NS,(15) A,2*
 TAS,12,FIN RESP,1,1,SC,0,0,0,NS,(15) A,2*

```

TAS,50,ENDTRIAL,1,1,SC,60,0,0,NS*
TAS,51,ENDSIM,1,9999,SC,0,0,0,SI*
STA,1,M,STA*
UTC,1,1,0,5,1,-11,0*
UTC,2,2,0,5,1,-11,0*
UTC,3,3,0,5,1,-11,0*
UTC,4,7,0,5,1,-11,0*
UTC,5,4,0,5,1,-11,0*
UTC,6,0,0,5,1,-11,0*
UTC,7,0,0,5,1,-11,0*
UTC,8,5,0,5,1,-11,0*
UTC,9,6,0,5,1,-11,0*
UTC,10,0,0,5,1,-11,0*
UTC,11,0,0,5,1,-11,0*
UTC,12,0,0,5,1,-11,0*
UTC,50,0,0,5,1,-11,0*
UTC,51,0,0,5,1,-11,0*
MOD,6,2,D,T,4,D,T*
MOD,7,2,D,T,4,D,T*
MOD,10,2,D,T,4,D,T*
MOD,11,2,D,T,4,D,T*
MOD,12,2,D,T,4,D,T*
MOD,50,2,D,T,3,A,T,4,D,T*
MOD,51,2,D,T,4,D,T*
ATA,1,STA,SA,,1,SC,0,IA,,2,SC,0*
ATA,8,COM,SA,,1,UF,1*
ATA,10,COM,IA,,2,SC,1*
ATA,11,COM,IA,,2,SC,2*
ATA,50,COM,SA,,3,UF,5,SA,,5,SC,3*
DET,1,2*
DET,2,3*
DET,5,4*
DET,6,7*
DET,7,8*
DET,9,7*
DET,10,3*
DET,11,3*
DET,12,50*
CFI,3,4,ALV,0,2,IA,,5,ALV,1,2,IA,,6,AGV,1,2,IA*
CFI,4,10,ALV,0,2,IA,,11,ALV,1,2,IA,,12,AGV,1,2,IA*
CFI,8,9,AGA,1,5,SA,,4,ALA,1,5,SA*
CFI,50,1,ALA,4,3,SA,,51,AGA,4,3,SA*
FINISH*
      2          NUMTR
      7          NUMPM
      .004        STDEC
      .02         FRDEC
      4.02        TCONS
      2.02        TSTRN
      1.02        TSTDV
      10.02       P1TIM
      10.02       P2TIM
      10.02       P3TIM
      0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
      6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0 6.0
      3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0
      2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1          MORE
      TRLRN
      SASMT
      THRS0
      THRS1
      THRS2

```

GEN,SIGNALS1,12,23,1982,81, 2,1,71268659,1,N*
 POP,1,0,2,28,6,,,2*
 OUT,1,1,(14)N,Y,N,N*
 DIS,1,NO,0,-10,10,1*
 DIS,2,ER,1,0,10,1*
 UBO, 1,ARMEXTND, 2, ARM360, 3,POSITARM, 4,MOVEARMS, 5,RAISEHND,
 6,CLNCHFST, 7,BECKMOTN, 8,RAISEHND, 9,CLNCHFST, 10,BECKMOTN,
 11,CLSPHAND, 12,CHINLEVL, 13,RAISEHND, 14,PALMFRNT, 15,PUSHAWAY,
 16,CRSWRIST, 17,INDXFING, 18,CLNCHFST, 19,RTPLMDWN, 20,THRT CUT,
 21, GO/NG 1*
 UBO, 22,ARMEXTND, 23, ARM360, 24,POSITARM, 25,MOVEARMS, 26,RAISEHND,
 27,CLNCHFST, 28,BECKMOTN, 29,RAISEHND, 30,CLNCHFST, 31,BECKMOTN,
 32,CLSPHAND, 33,CHINLEVL, 34,RAISEHND, 35,PALMFRNT, 36,PUSHAWAY,
 37,CRSWRIST, 38,INDXFING, 39,CLNCHFST, 40,RTPLMDWN, 41,THRT CUT,
 42, GO/NG 2*
 UBO, 43,ARMEXTND, 44, ARM360, 45,POSITARM, 46,MOVEARMS, 47,RAISEHND,
 48,CLNCHFST, 49,BECKMOTN, 50,RAISEHND, 51,CLNCHFST, 52,BECKMOTN,
 53,CLSPHAND, 54,CHINLEVL, 55,RAISEHND, 56,PALMFRNT, 57,PUSHAWAY,
 58,CRSWRIST, 59,INDXFING, 60,CLNCHFST, 61,RTPLMDWN, 62,THRT CUT,
 63, GO/NG 3*
 UBO, 64,ARMEXTND, 65, ARM360, 66,POSITARM, 67,MOVEARMS, 68,RAISEHND,
 69,CLNCHFST, 70,BECKMOTN, 71,RAISEHND, 72,CLNCHFST, 73,BECKMOTN,
 74,CLSPHAND, 75,CHINLEVL, 76,RAISEHND, 77,PALMFRNT, 78,PUSHAWAY,
 79,CRSWRIST, 80,INDXFING, 81,CLNCHFST, 82,RTPLMDWN, 83,THRT CUT,
 84, GO/NG 4*
 UBO, 85,ARMEXTND, 86, ARM360, 87,POSITARM, 88,MOVEARMS, 89,RAISEHND,
 90,CLNCHFST, 91,BECKMOTN, 92,RAISEHND, 93,CLNCHFST, 94,BECKMOTN,
 95,CLSPHAND, 96,CHINLEVL, 97,RAISEHND, 98,PALMFRNT, 99,PUSHAWAY,
 100,CRSWRIST,101,INDXFING,102,CLNCHFST,103,RTPLMDWN,104,THRT CUT,
 105, GO/NG 5*
 UBO,106,ARMEXTND,107, ARM360,108,POSITARM,109,MOVEARMS,110,RAISEHND,
 111,CLNCHFST,112,BECKMOTN,113,RAISEHND,114,CLNCHFST,115,BECKMOTN,
 116,CLSPHAND,117,CHINLEVL,118,RAISEHND,119,PALMFRNT,120,PUSHAWAY,
 121,CRSWRIST,122,INDXFING,123,CLNCHFST,124,RTPLMDWN,125,THRT CUT,
 126, GO/NG 6*
 UBO,127,TIME 1,128,TIME 2,129,TIME 3,130,TIME 4,131,TIME 5,132,TIME 6*
 UHI, 1,ARMEXTND,3,0,1, 2, ARM360,3,0,1, 3,POSITARM,3,0,1,
 4,MOVEARMS,3,0,1, 5,RAISEHND,3,0,1, 6,CLNCHFST,3,0,1,
 7,BECKMOTN,3,0,1, 8,RAISEHND,3,0,1, 9,CLNCHFST,3,0,1*
 UHI, 10,BECKMOTN,3,0,1, 11,CLSPHAND,3,0,1, 12,CHINLEVL,3,0,1,
 13,RAISEHND,3,0,1, 14,PALMFRNT,3,0,1, 15,PUSHAWAY,3,0,1,
 16,CRSWRIST,3,0,1, 17,INDXFING,3,0,1, 18,CLNCHFST,3,0,1*
 UHI, 19,RTPLMDWN,3,0,1, 20,THRT CUT,3,0,1, 21, GO/NG 1,1,0,1*
 UHI, 22,ARMEXTND,3,0,1, 23, ARM360,3,0,1, 24,POSITARM,3,0,1,
 25,MOVEARMS,3,0,1, 26,RAISEHND,3,0,1, 27,CLNCHFST,3,0,1,
 28,BECKMOTN,3,0,1, 29,RAISEHND,3,0,1, 30,CLNCHFST,3,0,1*
 UHI, 31,BECKMOTN,3,0,1, 32,CLSPHAND,3,0,1, 33,CHINLEVL,3,0,1,
 34,RAISEHND,3,0,1, 35,PALMFRNT,3,0,1, 36,PUSHAWAY,3,0,1,
 37,CRSWRIST,3,0,1, 38,INDXFING,3,0,1, 39,CLNCHFST,3,0,1*
 UHI, 40,RTPLMDWN,3,0,1, 41,THRT CUT,3,0,1, 42, GO/NG 2,1,0,1*
 UHI, 43,ARMEXTND,3,0,1, 44, ARM360,3,0,1, 45,POSITARM,3,0,1,
 46,MOVEARMS,3,0,1, 47,RAISEHND,3,0,1, 48,CLNCHFST,3,0,1,
 49,BECKMOTN,3,0,1, 50,RAISEHND,3,0,1, 51,CLNCHFST,3,0,1*
 UHI, 52,BECKMOTN,3,0,1, 53,CLSPHAND,3,0,1, 54,CHINLEVL,3,0,1,
 55,RAISEHND,3,0,1, 56,PALMFRNT,3,0,1, 57,PUSHAWAY,3,0,1,

58,CRSWRIST,3,0,1, 59,INDXFING,3,0,1, 60,CLNCHFST,3,0,1*
 UHI, 61,RTPLMDWN,3,0,1, 62,THRT CUT,3,0,1, 63,GO/NG 3,1,0,1*
 UHI, 64,ARMEXTND,3,0,1, 65, ARM360,3,0,1, 66,POSITARM,3,0,1,
 67,MOVEARMS,3,0,1, 68,RAISEHND,3,0,1, 69,CLNCHFST,3,0,1,
 70,BECKMOTN,3,0,1, 71,RAISEHND,3,0,1, 72,CLNCHFST,3,0,1*
 UHI, 73,BECKMOTN,3,0,1, 74,CLSPHAND,3,0,1, 75,CHINLEVL,3,0,1,
 76,RAISEHND,3,0,1, 77,PALMFRNT,3,0,1, 78,PUSHAWAY,3,0,1,
 79,CRSWRIST,3,0,1, 80,INDXFING,3,0,1, 81,CLNCHFST,3,0,1*
 UHI, 82,RTPLMDWN,3,0,1, 83,THRT CUT,3,0,1, 84,GO/NG 4,1,0,1*
 UHI, 85,ARMEXTND,3,0,1, 86, ARM360,3,0,1, 87,POSITARM,3,0,1,
 88,MOVEARMS,3,0,1, 89,RAISEHND,3,0,1, 90,CLNCHFST,3,0,1,
 91,BECKMOTN,3,0,1, 92,RAISEHND,3,0,1, 93,CLNCHFST,3,0,1*
 UHI, 94,BECKMOTN,3,0,1, 95,CLSPHAND,3,0,1, 96,CHINLEVL,3,0,1,
 97,RAISEHND,3,0,1, 98,PALMFRNT,3,0,1, 99,PUSHAWAY,3,0,1,
 100,CRSWRIST,3,0,1,101,INDXFING,3,0,1,102,CLNCHFST,3,0,1*
 UHI,103,RTPLMDWN,3,0,1,104,THRT CUT,3,0,1,105,GO/NG 5,1,0,1*
 UHI,106,ARMEXTND,3,0,1,107, ARM360,3,0,1,108,POSITARM,3,0,1,
 109,MOVEARMS,3,0,1,110,RAISEHND,3,0,1,111,CLNCHFST,3,0,1,
 112,BECKMOTN,3,0,1,113,RAISEHND,3,0,1,114,CLNCHFST,3,0,1*
 UHI,115,BECKMOTN,3,0,1,116,CLSPHAND,3,0,1,117,CHINLEVL,3,0,1,
 118,RAISEHND,3,0,1,119,PALMFRNT,3,0,1,120,PUSHAWAY,3,0,1,
 121,CRSWRIST,3,0,1,122,INDXFING,3,0,1,123,CLNCHFST,3,0,1*
 UHI,124,RTPLMDWN,3,0,1,125,THRT CUT,3,0,1,126,GO/NG 6,1,0,1*
 UHI,127,TIME 1,,,,128,TIME 2,,,,129,TIME 3,,,,130,TIME 4,,,,
 131,TIME 5,,,,132,TIME 6*
 UPL,1,TRIAL,4,B,1*
 UVA,1,1,4,NUM GO,V,V,0,100*
 IMO,2,A,4,A*
 ISA,2,SC,1,3,SC,1,12,SC,1*
 LRE,1,SIGNALER*
 TAS,48, INIT,0,1,SC,0,0,0,SO,(15) A,1*
 TAS,49, SETUP,1,1,SC,0,0,0,NS,(15) A,1*
 TAS,1,SCHEDULE,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 2, DECIDE,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 3,START TK,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 4, FORWARD,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 5, TURN RT,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 6,TURN LFT,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 7, STOP,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 8, REVERSE,1,1,SC,0,0,0,NS,(15) A,1*
 TAS, 9, N STEER,1,1,SC,0,0,0,NS,(15) A,1*
 TAS,10,TANK OFF,1,1,SC,0,0,0,NS,(15) A,1*
 TAS,11,ARMEXTND,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,12, ARM360,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,13,POSITARM,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,14,MOVEARMS,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,15,RAISEHND,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,16,CLNCHFST,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,17,BECKMOTN,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,18,RAISEHND,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,19,CLNCHFST,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,20,BECKMOTN,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,21,CLSPHAND,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,22,CHINLEVL,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,23,RAISEHND,1,1,DS,2,0,0,NS,(15) A,1*

TAS,24,PALMFRNT,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,25,PUSHAWAY,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,26,CRSWRIST,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,27,INDXFING,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,28,CLNCHFST,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,29,RTPLMDWN,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,30,THRT CUT,1,1,DS,2,0,0,NS,(15) A,1*
 TAS,50,ENDTRIAL,1,1,SC,60,0,0,NS*
 TAS,51, END SIM,1,9999,SC,0,0,0,SI*
 STA,1,H,STA*
 UTC,48, 0,0,5,1,-11,0*
 UTC,49, 0,0,5,1,-11,0*
 UTC, 1, 0,0,5,1,-11,0*
 UTC, 2, 0,0,5,1,-11,0*
 UTC, 3, 0,0,5,1,-11,0,3*
 UTC, 4, 0,0,5,1,-11,0,4*
 UTC, 5, 0,0,5,1,-11,0,5*
 UTC, 6, 0,0,5,1,-11,0,6*
 UTC, 7, 0,0,5,1,-11,0,7*
 UTC, 8, 0,0,5,1,-11,0,8*
 UTC, 9, 0,0,5,1,-11,0,9*
 UTC,10, 0,0,5,1,-11,0,10*
 UTC,11, 1,0,5,1,-11,0*
 UTC,12, 2,0,5,1,-11,0*
 UTC,13, 3,0,5,1,-11,0*
 UTC,14, 4,0,5,1,-11,0*
 UTC,15, 5,0,5,1,-11,0*
 UTC,16, 6,0,5,1,-11,0*
 UTC,17, 7,0,5,1,-11,0*
 UTC,18, 8,0,5,1,-11,0*
 UTC,19, 9,0,5,1,-11,0*
 UTC,20,10,0,5,1,-11,0*
 UTC,21,11,0,5,1,-11,0*
 UTC,22,12,0,5,1,-11,0*
 UTC,23,13,0,5,1,-11,0*
 UTC,24,14,0,5,1,-11,0*
 UTC,25,15,0,5,1,-11,0*
 UTC,26,16,0,5,1,-11,0*
 UTC,27,17,0,5,1,-11,0*
 UTC,28,18,0,5,1,-11,0*
 UTC,29,19,0,5,1,-11,0*
 UTC,30,20,0,5,1,-11,0*
 UTC,50, 0,0,5,1,-11,0*
 UTC,51, 0,0,5,1,-11,0*
 MOD,48,2,D,T,4,D,T*
 MOD,49,2,D,T,4,D,T,6,A,T*
 MOD, 1,2,D,T,4,D,T*
 MOD,2,1,A,T,2,D,T,4,D,T*
 MOD,3,2,D,T,4,D,T*
 MOD,4,2,D,T,4,D,T*
 MOD,5,2,D,T,4,D,T*
 MOD,6,2,D,T,4,D,T*
 MOD,7,2,D,T,4,D,T*
 MOD,8,2,D,T,4,D,T*
 MOD,9,2,D,T,4,D,T*

MOD,10,2,D,T,4,D,T*
 MOD,50,2,D,T,3,A,T,4,D,T*
 MOD,51,2,D,T,4,D,T*
 ATA,48,COM,SA,,28,UF,6*
 ATA,1,COM,SA,,1,UF,1,IA,,2,UF,2*
 ATA,50,COM,SA,,3,UF,5,SA,,1,SC,0,IA,,2,SC,0*
 DET,48,49*
 DET,49,1
 DET,1,2*
 DET,3,11*
 DET,11,12*
 DET,12,1*
 DET,4,13*
 DET,13,14*
 DET,14,1*
 DET,5,15*
 DET,15,16*
 DET,16,17*
 DET,17,1*
 DET,6,18*
 DET,18,19*
 DET,19,20*
 DET,20,1*
 DET,7,21*
 DET,21,22*
 DET,22,1*
 DET,8,23*
 DET,23,24*
 DET,24,25*
 DET,25,1*
 DET,9,26*
 DET,26,27*
 DET,27,28*
 DET,28,1*
 DET,10,29*
 DET,29,30*
 DET,30,50*
 CFI,50,49,ALA,4,3,SA,0,51,AGA,4,3,SA*
 PRO,2,SA,,3,5,4,6,5,7,6,8,7,9,8,10,9,11,10,12*
 FINISH*

	2	20	.004	.02	2.02	1.02	1.00	10.00	10.00	10.00										
	NUMTR	NUMPM	STDEC	FRDEC	TCONS	TSTRN	TSTDV	F1TIM	F2TIM	F3TIM										
5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0				
5.0	5.0	5.0	5.0			TSTRN														
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
0.0	0.0	0.0	0.0			SASMT														
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0				
2.0	2.0	2.0	2.0			THRS0														
1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5				
1.5	1.5	1.5	1.5			THRS1														
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
0.0	0.0	0.0	0.0			THRS2														
3	MORE																			
3	8	7	4	5	4	7	9	4	6	4	7	10	0	0						
3	8	7	4	6	4	7	9	4	5	4	7	10	0	0						
3	4	6	4	7	9	4	5	4	7	9	8	7	10	0						
3	4	5	4	7	9	4	6	4	7	9	8	7	10	0						
2	3	1	4	2	1															
4	2	1	3	1	2															
2	1	4	3	3	2															
3	0	0	0	0	0															

APPENDIX C.

**SAMPLE SCORESHEETS SHOWING
PERFORMANCE MEASURES**

NAME _____

SSAN _____

UNIT _____

TEST ☐ 1 ☐ 2 ☐ 3TRAIN ☐ 1 ☐ 2 ☐ 3

LOAD AN M240 COAX MACHINEGUN

INSTRUCTIONS TO SOLDIER

"At this station you will demonstrate your ability to load an M240 coaxial machinegun. Assume the machinegun will be fired immediately after it is loaded. Do you understand the instructions?" (NOTE TO SCORER: If the soldier does not, read the instructions again.) "BEGIN."

PERFORMANCE MEASURES

	<u>YES</u>	<u>NO</u>	<u>PTS</u>	<u>TIME</u>
1. Clears the machinegun.				
a. Pulls charger handle rearward to lock bolt back.	___	___	1 2 3	___
b. Places safety on S.	___	___	1 2 3	___
c. Raises cover.	___	___	1 2 3	___
d. Lifts feedtray.	___	___	1 2 3	___
e. Pulls charger handle back and to side of gun, then holds handle against cover.	___	___	1 2 3	___
f. Looks and feels empty chamber.	___	___	1 2 3	___
g. Lowers feedtray.	___	___	1 2 3	___
2. Loads the machinegun.				
a. Places first round in feedtray with open side of belt face down.	___	___	1 2 3	___
b. Pushes ammunition in feedtray until it comes in contact with cartridge stops.	___	___	1 2 3	___
c. Closes cover.	___	___	1 2 3	___
d. Places safety in F.	___	___	1 2 3	___
e. Announces "UP" when machinegun loaded.	___	___	1 2 3	___

TOTAL TIME _____

NAME _____

SSAN _____

UNIT _____

TEST ☐ 1 ☐ 2 ☐ 3

TRAIN ☐ 1 ☐ 2 ☐ 3

The soldier has satisfactorily completed the task if he scores a "YES" on all of the standards listed below:

<u>STANDARDS</u>	<u>YES</u>	<u>NO</u>
1. Completes all performance measures without assistance from scorer.	_____	_____
2. Steps are performed in sequence.	_____	_____
3. Ammunition is in feedtray and doesn't pull out when jerked.	_____	_____
TOTAL SCORE	_____	_____
TOTAL TIME	_____	_____

REASON(S) FOR "NO" SCORE

NAME _____

SSAN _____

UNIT _____

TEST ☐ 1 ☐ 2 ☐ 3TRAIN ☐ 1 ☐ 2 ☐ 3

START THE M60A1 TANK ENGINE

INSTRUCTIONS TO SOLDIER

"You are the driver of an M60A1 tank. You are to start the engine, assuming normal weather conditions. I will act as other crew positions when necessary. Do you understand the instructions?" (NOTE TO SCORER: If the soldier has questions, read the instructions again.) "BEGIN."

PERFORMANCE MEASURES

YES NO PROMPTS TIME

- | | | | | | | |
|--|-----|-----|---|---|---|-----|
| 1. Sets parking brake by pushing brake pedal until pressure reaches between 750-900 psi. | ___ | ___ | 1 | 2 | 3 | ___ |
| 2. Places transmission in PARK. | ___ | ___ | 1 | 2 | 3 | ___ |
| 3. Releases brake pedal. | ___ | ___ | 1 | 2 | 3 | ___ |
| 4. Closes both drain valves. | ___ | ___ | 1 | 2 | 3 | ___ |
| 5. Places fuel shut-off valve handle in ON position. | ___ | ___ | 1 | 2 | 3 | ___ |
| 6. Places fuel pump switch in ON position. | ___ | ___ | 1 | 2 | 3 | ___ |
| 7. Asks crew if their electronic equipment is OFF. | ___ | ___ | 1 | 2 | 3 | ___ |

(NOTE TO SCORER: Tell soldier the electronic equipment is OFF.)

(NOTE TO SCORER: Insure all the electronic equipment is OFF before master battery switch is turned ON.)

- | | | | | | | |
|--|-----|-----|---|---|---|-----|
| 8. Turns master battery switch ON. | ___ | ___ | 1 | 2 | 3 | ___ |
| 9. Check fuel levels in both tanks. | | | | | | |
| a. Sets FUEL TANKS switch to position L. | ___ | ___ | 1 | 2 | 3 | ___ |
| <u>OR</u> | | | | | | |
| b. Sets FUEL TANKS switch to position R. | ___ | ___ | 1 | 2 | 3 | ___ |

(NOTE TO SCORER: If a soldier performs A or B, he should be given a "YES" for PM 9.)

- | | | | | | | |
|---|-----|-----|---|---|---|-----|
| 10. Turns generator switch ON. | ___ | ___ | 1 | 2 | 3 | ___ |
| 11. Depresses accelerator pedal | ___ | ___ | 1 | 2 | 3 | ___ |
| 12. Presses starter switch until engine starts (for up to 15 seconds, whichever comes first.) | | | 1 | 2 | 3 | |

NAME _____

SSAN _____

UNIT _____

TEST ☐ 1 ☐ 2 ☐ 3

TRAIN ☐ 1 ☐ 2 ☐ 3

PERFORMANCE MEASURES

13. Holds piece of cloth in front of air intake screen to see if generator is blowing.

YES NO PROMPTS TIME

_____ 1 2 3 _____

TOTAL TIME _____

The soldier has satisfactorily completed the task if he scores a "YES" on all of the Standards listed below:

STANDARDS

YES NO

1. Completes all performance measures without assistance from scorer. _____
2. Asks if electronic equipment is OFF before turning master battery switch ON. _____
3. Tank engine starts. _____
4. Performs performance measures in sequence when necessary (see sequence flowchart on next page.) _____

TOTAL SCORE _____

TOTAL TIME _____

REASON(S) FOR "NO" SCORE

(NOTE TO SCORER: For the 194th Brigade proceed directly to the "Stop the Tank Engine." Tell the soldier "Assume you are getting ready to drive. Release the parking brake and put transmission in neutral. Now assume you are finished driving. Demonstrate the procedure for stopping the engine. I will act as the crew when necessary. Do you understand the instructions?"

NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

STOP THE M60A1 TANK ENGINE

INSTRUCTIONS TO SOLDIER

"You are the driver of an M60A1 tank. You are to demonstrate the procedure for stopping the tank's engine. I will act as tank commander or gunner when necessary. Do you understand the instructions?" (NOTE TO SCORER: If the soldier has questions, read the instructions again.) "BEGIN."

<u>PERFORMANCE MEASURES</u>	<u>YES</u>	<u>NO</u>	<u>PROMPTS</u>	<u>TIME</u>
1. Sets parking brake by pushing brake pedal until pressure reaches between 750-900 psi.	___	___	1 2 3	___
2. Places transmission in PARK.	___	___	1 2 3	___
3. Releases brake pedal.	___	___	1 2 3	___
4. Presses accelerator so that engine idles at 1000-1200 rpm.	___	___	1 2 3	___
(NOTE TO SCORER: Ask soldier how long engine should idle at this rpm.)				
5. Soldier says engine idles at 1000-1200 rpm for <u>5</u> minutes.	___	___	1 2 3	___
(NOTE TO SCORER: Tell soldier to continue to next step.)				
6. Releases accelerator and idles engine at 700-750 rpm.	___	___	1 2 3	___
(NOTE TO SCORER: Ask soldier how long engine should idle at this rpm.)				
7. Soldier says engine idles at 750 rpm for <u>3</u> minutes.	___	___	1 2 3	___
8. Turns lighting control (ON-OFF) switch OFF.	___	___	1 2 3	___

NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

PERFORMANCE MEASURES

YES NO PROMPTS TIME

9. Asks Gunner and TC if their electronic equipment is OFF.

(NOTE TO SCORER: Scorer tells soldier the equipment is OFF.)

10. Holds engine fuel shut-off switch in "SHUT-OFF" (Up) until engine stops.

_____ 1 2 3 _____

11. Turns master battery OFF, after engine stops.

_____ 1 2 3 _____

TOTAL TIME _____

The soldier has satisfactorily completed the task if he scores a "YES" on all of the Standards listed below:

STANDARDS

YES NO

1. Completes all performance measures without assistance from scorer.
2. Turns master battery switch OFF, after engine stops.
3. All driver's electronic equipment switches are in OFF position.
4. Engine stops.
5. Performs performance measures in sequence when necessary (see sequence flowchart on next page).

TOTAL SCORE

TOTAL TIME

REASON(S) FOR "NO" SCORE

NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

PERFORM GUNNER'S PREPARE-TO-FIRE CHECKS
 (CHECK GUN CONTROLS)

INSTRUCTIONS TO SOLDIER

"You are the gunner of an M60A1 tank. You are doing Prepare-to-Fire checks and have already checked the firing switches. You will perform the sequence "CHECK GUN CONTROLS" after I give you the command. The turret is in manual operation. I will act as the other crew positions when necessary. Do you understand the instructions?" (NOTE TO SCORER: If the soldier has questions, read the instructions again.) "Remember, the turret must be placed into power operation before checking the azimuth indicator for accuracy and slippage."

(NOTE TO SCORER: Start the trial by saying "CHECK GUN CONTROLS.")

<u>PERFORMANCE MEASURES</u>	<u>YES</u>	<u>NO</u>	<u>PROMPTS</u>	<u>TIME</u>
1. Places turret into power operation.				
a. Holds down power solenoid plunger while rotating gunner's control handle either left or right.	___	___	1 2 3	___
b. Holds gunner control handle in position described in (a) until zero pressure is indicated on pressure gage.	___	___	1 2 3	___
c. Checks hydraulic power pack oil level by removing dipstick of oil level gage.	___	___	1 2 3	___
d. Tells loader to unlock turret traverse lock.	___	___	1 2 3	___
(NOTE TO SCORER: Unlock turret lock.)				
e. Announces "POWER."	___	___	1 2 3	___
(NOTE TO SCORER: Announces "POWER UP.")				
f. Turns ELEV/TRAV power switch ON.	___	___	1 2 3	___
g. Squeezes magnetic brake switch while rotating gunner's power control handles to left and right.	___	___	1 2 3	___
h. Moves handles rearward to elevate gun, forward to lower gun, while squeezing magnetic brake switch.	___	___	1 2 3	___

(NOTE TO SCORER: PM g and h may be done as listed or reversed [h then g].)

NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

PERFORMANCE MEASURES

YES NO PROMPTS TIME

(NOTE TO SCORER: Tell soldier TC's power control handles have been operated.)

2. Checks azimuth indicator for accuracy.

- a. Looks through eyepiece on gunner's day-light periscope.

___ ___ 1 2 3 ___

(NOTE TO SCORER: Tell soldier the aiming point.)

- b. Alines cross on aiming point using manual elevating and traversing handles.

___ ___ 1 2 3 ___

(NOTE TO SCORER: Verify soldier has alined cross on aiming point.)

- c. Sets azimuth indicator to zero.

- Presses resetter knob.

___ ___ 1 2 3 ___

- Turns resetter knob to aline middle scale pointer with inner scale pointer.

___ ___ 1 2 3 ___

- Turns resetter knob moving both pointers to zero.

___ ___ 1 2 3 ___

- Releases resetter knob.

___ ___ 1 2 3 ___

- d. Traverses turret through complete circle using manual traversing handle.

___ ___ 1 2 3 ___

- e. Brings aiming cross back on same aiming point.

___ ___ 1 2 3 ___

(NOTE TO SCORER: Verify the aiming cross is on original aiming point by looking through periscope.)

- f. Turns head to check that azimuth indicator middle scale pointer is within acceptable area.

___ ___ 1 2 3 ___

(NOTE TO SCORER: Use scoring aid when determining if the pointer is within the acceptable area.)

- g. 1) Proceeds to next check if middle scale pointer is within acceptable area.

___ ___ 1 2 3 ___

OR

- 2) Notifies tank commander (TC) pointer is not within acceptable area.

___ ___ 1 2 3 ___

NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

PERFORMANCE MEASURES

3. Checks azimuth indicator for slippage.

Right Side

- | | <u>YES</u> | <u>NO</u> | <u>PROMPTS</u> | <u>TIME</u> |
|---|------------|-----------|----------------|-------------|
| a. Looks through eyepiece of gunner's day-light periscope. | ___ | ___ | 1 2 3 | ___ |
| b. Uses gunner's control handles to traverse rapidly to right. | ___ | ___ | 1 2 3 | ___ |
| c. Stops turret suddenly while traversing. | ___ | ___ | 1 2 3 | ___ |
| d. Turns ELEV/TRAV power switch OFF. | ___ | ___ | 1 2 3 | ___ |
| e. Traverse turret left using manual tra-verse handle until cross is alined with original aiming point. | ___ | ___ | 1 2 3 | ___ |

(NOTE TO SCORER: Verify the aiming cross is on original aiming point by looking through periscope.)

- | | | | | |
|--|-----|-----|-------|-----|
| f. Turns head to check that azimuth indi-cator middle scale pointer is within acceptable area. | ___ | ___ | 1 2 3 | ___ |
|--|-----|-----|-------|-----|

(NOTE TO SCORER: Use scoring aid when determining if the pointer is within acceptable area.)

- | | | | | |
|--|-----|-----|-------|-----|
| g. 1) Proceeds to left side check if middle scale indicator pointer is within acceptable area. | ___ | ___ | 1 2 3 | ___ |
| <u>OR</u> | | | | |
| 2) Notifies TC if both pointers are not within acceptable area. | ___ | ___ | 1 2 3 | ___ |

- | | | | | |
|-------------------------------------|-----|-----|-------|-----|
| h. Announces POWER. | ___ | ___ | 1 2 3 | ___ |
| i. Turns ELEV/TRAV power switch ON. | ___ | ___ | 1 2 3 | ___ |

Left Side

- | | | | | |
|--|-----|-----|-------|-----|
| a. Looks through eyepiece of gunner's daylight periscope. | ___ | ___ | 1 2 3 | ___ |
| b. Uses gunner's control handles to traverse rapidly to left. | ___ | ___ | 1 2 3 | ___ |
| c. Stops turret suddenly while traversing. | ___ | ___ | 1 2 3 | ___ |
| d. Turns ELEV/TRAV power switch OFF. | ___ | ___ | 1 2 3 | ___ |
| e. Traverses turret right using manual traverse handle until cross is alined with original aiming point. | ___ | ___ | 1 2 3 | ___ |

NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

PERFORMANCE MEASURES

YES NO PROMPTS TIME

(NOTE TO SCORER: Verify aiming cross is on original aiming point by looking through periscope.)

- f. Turns head to check that middle scale pointer is within acceptable area.

___ ___ 1 2 3 ___

(NOTE TO SCORER: Use scoring aid when determining if the pointers are within acceptable area.)

- g. 1) Stops check if pointer is within acceptable area.

___ ___ 1 2 3 ___

OR

- 2) Notifies TC if pointers are not within acceptable area.

___ ___ 1 2 3 ___

TOTAL TIME _____

The soldier has satisfactorily completed the task if he scores a "YES" on all of the standards listed below:

STANDARDS

YES NO

1. Completes all performance measures.
2. Announced "POWER" before turning ELEV/ TRAV switch ON.
3. Pointer of azimuth indicator is within range shown on scoring aid after accuracy.
4. Pointers of azimuth indicator are within range shown on scoring aid after each slippage test.
5. Cross is alined with aiming point after accuracy checks.
6. Performs performance measures in sequence when necessary (see sequence flowchart on next page).

___ ___

___ ___

___ ___

___ ___

___ ___

___ ___

TOTAL SCORE

___ ___

TOTAL TIME

NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

PERFORM LOADER'S PREPARE-TO-FIRE CHECKS
 (CHECK MAIN GUN FIRING SWITCHES)

INSTRUCTIONS TO SOLDIER

"You are the loader of an M60A1 tank. You are doing the Prepare-to-Fire checks and will demonstrate the section "Check Main Gun Firing Switches." I will act as the other crew positions when necessary. I cannot answer questions or assist you once the test has started. Do you understand the instructions?" (NOTE TO SCORER: If the soldier has questions, read the instructions again.)

(NOTE TO SCORER: Start the trial by saying "CHECK MAIN GUN FIRING SWITCHES.")

<u>PERFORMANCE MEASURES</u>	<u>YES</u>	<u>NO</u>	<u>PROMPTS</u>	<u>TIME</u>
1. Closes breech by tripping extractors with block of wood.	___	___	1 2 3	___
2. Inserts circuit tester into opening between rear face of gun tube and front face of breechblock.	___	___	1 2 3	___
3. Moves main gun safety switch to FIRE position.	___	___	1 2 3	___
4. Announces "UP."	___	___	1 2 3	___
(NOTE TO SCORER: Turn master battery switch ON, then turn the main gun switch ON. Momentarily press the commander's control handle palm switch. Circuit tester should not light.)				
5. Tells gunner to squeeze main gun triggers.	___	___	1 2 3	___
(NOTE TO SCORER: Squeeze the trigger on each handle and the trigger on manual elevation control. Rotate the manual firing handle very rapidly in a clockwise direction. Announce ON THE WAY each time you squeeze a trigger. Circuit tester should light.)				
6. Tells TC to squeeze main gun trigger.	___	___	1 2 3	___
(NOTE TO SCORER: Squeeze and hold override palm handle, then squeeze trigger. Announce "ON THE WAY." Circuit tester should light.)				

NAME _____

SSAN _____

UNIT _____

TEST ☐ 1 ☐ 2 ☐ 3

TRAIN ☐ 1 ☐ 2 ☐ 3

PERFORMANCE MEASURES

7. Moves main gun safety switch to SAFE.

8. Tells gunner to press trigger on manual firing handle.

(NOTE TO SCORER: Squeeze the trigger on manual firing handle. Announce "ON THE WAY." Turn manual firing handle very rapidly in clockwise direction. Announce "ON THE WAY." Circuit tester should not light.)

9. Tells gunner to turn main gun OFF.

10. Removes circuit tester from breechblock.

YES NO PRO:PTS TIME

___ ___ 1 2 3 ___

___ ___ 1 2 3 ___

___ ___ 1 2 3 ___

___ ___ 1 2 3 ___

TOTAL TIME ___

STANDARD

1. The soldier has satisfactorily completed the task if he scores a "YES" on all of the performance measures.

2. Performs performance measure in sequence when necessary (see sequence flowchart on next page).

TOTAL TIME

YES NO

___ ___

___ ___

REASON(S) FOR "NO" SCORE

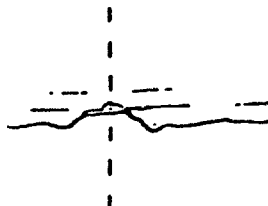
NAME _____
 SSAN _____
 UNIT _____

TEST

SERIES 1

Engagement 1
 Gunner
 SABOT
 Tank
 Fire

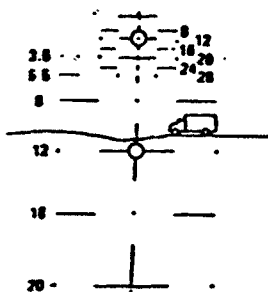
1. Turns main gun switch ON.
2. Indexes ammunition.
3. Looks through Unity Window.
4. Announces IDENTIFIED.
- NOTE: Scorer says UP.
- NOTE: Scorer says FIRE.
5. Looks through correct sight.
 Periscope ☒
 Telescope _____
6. Selects correct reticle.
 Periscope ☒
 SABOT/HEP _____
 HEAT _____
7. Lays crosshair at center of the target
 (with lead applied).
 Periscope crosshair ☒
 SABOT 2000M range line, 2.5 mil lead _____
 HEP 1000M range line, 7.5 mil lead _____
 HEAT 1800M range line, 5.0 mil lead _____
8. Says ON THE WAY.



Yes	No	Prompts			Time
		1	2	3	
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____

Engagement 2
 (Periscope damaged)
 Gunner
 HEP
 Moving Truck
 1000
 Fire

1. Turns main gun switch ON.
2. Indexes ammunition.
3. Looks through Unity Window.
4. Announces IDENTIFIED.
- NOTE: Scorer says UP.
- NOTE: Scorer says FIRE.
5. Looks through correct sight.
 Periscope _____
 Telescope ☒
6. Selects correct reticle.
 Periscope _____
 SABOT/HEP ☒
 HEAT _____
7. Lays crosshair at center of the target
 (with lead applied).
 Periscope crosshair _____
 SABOT 2000M range line, 2.5 mil lead _____
 HEP 1000M range line, 7.5 mil lead ☒
 HEAT 1800M range line, 5.0 mil lead _____
8. Says ON THE WAY.



Yes	No	Prompts			Time
		1	2	3	
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____

NAME _____
SSAN _____
UNIT _____

TEST

SERIES 1

Engagement 3
(Periscope damaged)
Gunner
HEAT
Moving Tank
1800
Fire

1. Turns main gun switch ON.
 2. Indexes ammunition.
 3. Looks through Unity Window.
 4. Announces IDENTIFIED.
- NOTE: Scorer says UP.
NOTE: Scorer says FIRE.

5. Looks through correct sight.
Periscope _____
Telescope ✓

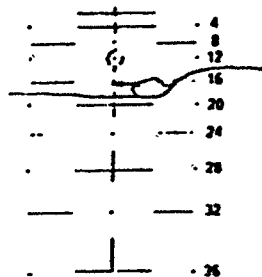
6. Selects correct reticle.
Periscope _____
SABOT/HEP _____
HEAT ✓

7. Lays crosshair at center of the target
(with lead applied).

Periscope crosshair _____
SABOT 2000M range line, 2.5 mil lead _____
HEP 1000M range line, 7.5 mil lead _____
HEAT 1800M range line, 5.0 mil lead ✓

8. Says ON THE WAY.

Yes	No	Prompts			Time
		1	2	3	
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____



Engagement 4
(Periscope damaged)
Gunner
SABOT
Moving Tank
2000
Fire

1. Turns main gun switch ON.
 2. Indexes ammunition.
 3. Looks through Unity Window.
 4. Announces IDENTIFIED.
- NOTE: Scorer says UP.
NOTE: Scorer says FIRE.

5. Looks through correct sight.
Periscope _____
Telescope ✓

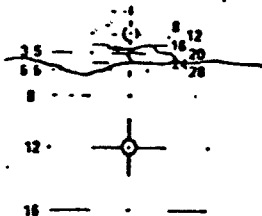
6. Selects correct reticle.
Periscope _____
SABOT/HEP ✓
HEAT _____

7. Lays crosshair at center of the target
(with lead applied).

Periscope crosshair _____
SABOT 2000M range line, 2.5 mil lead ✓
HEP 1000M range line, 7.5 mil lead _____
HEAT 1800M range line, 5.0 mil lead _____

8. Says ON THE WAY.

Yes	No	Prompts			Time
		1	2	3	
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____
_____	_____	1	2	3	_____



NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

COMMUNICATE OVER TACTICAL FM RADIO AN/VRC-64

<u>PERFORMANCE MEASURES</u>	<u>YES</u>	<u>NO</u>	<u>PROMPTS</u>	<u>TIME</u>
1. Places CVC helmet switch in center position.	___	___	1 2 3	
2. Calls net control station.	___	___	1 2 3	___
3. Identifies himself before giving the messages.	___	___	1 2 3	
4. Tells net control station number of messages.	___	___	1 2 3	
5. Tells net control station precedence of messages.	___	___	1 2 3	
6. Transmits Message #1.	___	___	1 2 3	___
7. Uses phonetic alphabet as required.	___	___	1 2 3	
8. Pronounces numbers correctly.	___	___	1 2 3	
9. Says BREAK at end of Message #1.	___	___	1 2 3	___
10. Transmits Message #2.	___	___	1 2 3	
11. Uses phonetic alphabet as required.	___	___	1 2 3	
12. Pronounces numbers correctly.	___	___	1 2 3	___
13. Says BREAK at end of Message #2.	___	___	1 2 3	___
14. Transmits Message #3.	___	___	1 2 3	___
15. Uses phonetic alphabet as required.	___	___	1 2 3	
16. Pronounces numbers correctly.	___	___	1 2 3	
17. Says OVER after Message #3.	___	___	1 2 3	___

<u>STANDARDS</u>	<u>YES</u>	<u>NO</u>
1. Each performance measure completed with a YES.	___	___
2. Steps are performed in sequence.	___	___
TOTAL TIME	_____	

NAME _____
 SSAN _____
 UNIT _____
 TEST ☐ 1 ☐ 2 ☐ 3
 TRAIN ☐ 1 ☐ 2 ☐ 3

COMMUNICATE USING VISUAL SIGNALLING TECHNIQUES:
 GROUND GUIDING

INSTRUCTIONS TO SOLDIER

"At this station you will be tested on your ability to ground guide a tank from the START point to the FINISH point of a driving course. The course is clearly marked for you. I will be the tank; whatever direction I am facing will be the direction the tank is facing. I am parked in the motor pool. My engine is not running. When we get to the FINISH point, assume that I won't be moving anymore today. Do you understand the instructions?" (NOTE TO SCORER: If soldier does not understand the instructions, reread them.) "BEGIN."

<u>PERFORMANCE MEASURES</u>	<u>SIGNAL GIVEN</u>	<u>YES</u>	<u>NO</u>	<u>PROMPTS</u>	<u>TIME</u>
1. Gives signal to Start Engines.					—
a. Extends arm toward front at waist level.	—	—	—	1 2 3	
b. Moves arm in circular motion.	—	—	—	1 2 3	
2. Gives signal to Move Vehicle Forward.					—
a. Positions both palms toward chest.	—	—	—	1 2 3	
b. Moves arms and hands backward and forward.	—	—	—	1 2 3	
3. Gives signal to Turn Left.					—
a. Raises hands to shoulder level in front of body.	—	—	—	1 2 3	
b. Forms clenched fist of arm indicating direction turn is to be made (as seen by tank driver).	—	—	—	1 2 3	
c. Makes beckoning motion with other arm to bring vehicle forward.	—	—	—	1 2 3	
4. Gives signal to Move Vehicle Forward.					—
a. Positions both palms toward chest.	—	—	—	1 2 3	
b. Moves arms and hands backward and forward.	—	—	—	1 2 3	
5. Gives signal to Stop Tank Movement.					—
a. Clasps hands.	—	—	—	1 2 3	
b. Places hands at chin level.	—	—	—	1 2 3	
6. Gives signal to Steer Neutral (Right).					—
a. Crosses wrists at throat.	—	—	—	1 2 3	
b. Points index finger to tank driver's right.	—	—	—	1 2 3	
c. Clenches fist of other hand.	—	—	—	1 2 3	
NOTE: If soldier gives right turn signal, tell him to give the signal for neutral steer right. <u>Do not</u> mark the PM "NO."					
7. Gives signal to Move Vehicle Forward.					—
a. Positions both palms toward chest.	—	—	—	1 2 3	
b. Moves arms and hands backward and forward.	—	—	—	1 2 3	
8. Gives signal to Turn Right.					—
a. Raises hands to shoulder level in front of body.	—	—	—	1 2 3	
b. Forms clenched fist of arm indicating direction turn is to be made (as seen by tank driver).	—	—	—	1 2 3	
c. Makes beckoning motion with other arm to bring vehicle forward.	—	—	—	1 2 3	
9. Gives signal to Move Vehicle Forward.					—
a. Positions both palms toward chest.	—	—	—	1 2 3	
b. Moves arms and hands backward and forward.	—	—	—	1 2 3	

	<u>SIGNAL GIVEN</u>	<u>YES</u>	<u>NO</u>	<u>PROMPTS</u>	<u>TIME</u>
10. Gives signal to Stop Tank Movement.					
a. Clasps hands.				1 2 3	
b. Places hands at chin level.				1 2 3	
NOTE: If the soldier gives a Right/Left Turn signal, tell him to back the tank to the FINISH point. <u>Do not</u> mark the PM "NO."					
11. Gives signal to Neutral Steer (Right).					
a. Crosses wrists at throat.				1 2 3	
b. Points index finger to tank driver's right.				1 2 3	
c. Clenches fist of other hand.				1 2 3	
NOTE: If soldier gives Back-Up--Hold, Left Track Signal, tell him to give Neutral Steer (Right) and back the tank to the finish point. <u>Do not</u> mark the PM "NO."					
12. Gives signal to Move in Reverse.					
a. Raises both hands to shoulder level.				1 2 3	
b. Places palms to front.				1 2 3	
c. Moves hands forward and backward as if pushing vehicle away.				1 2 3	
13. Gives signal to Stop Tank Movement.					
a. Clasps hands.				1 2 3	
b. Places hands at chin level.				1 2 3	
14. Gives signal to Stop Engines.					
a. Positions right hand palm down.				1 2 3	
b. Draws hand across neck in "throat cutting" motion from left to right.				1 2 3	
<u>STANDARD</u>		<u>YES</u>	<u>NO</u>		
1. Completes each performance measure without assistance from scorer.					
2. Ground guide signals given in sequence indicated.					
3. Correct ground guide signal given.					
		TOTAL TIME _____			

REASON(S) FOR "NO" SCORE
