ornl

ORNL/TM-13147

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

RAM Simulation Model for SPH/RSV Systems

J. C. Schryver
A. H. Primm
S. C. Nelson

AMMUNITION LOGISTICS PROGRAM

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-13147

Ammunition Logistics Program

RAM SIMULATION MODEL FOR SPH/RSV SYSTEMS

J. C. Schryver
Computer Science and Mathematics Division

A. H. Primm
Robotics and Process Systems Division

S. C. Nelson
Engineering Technology Division

Date Published: December 1995

**Research sponsored by the
Army's Project Manager, Crusader
under the Interagency Agreement 1892-A078-A1
between the Department of Energy and the
Armament Research, Development, and Engineering Center
at the Picatinny Arsenal**

**Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
managed by
LOCKHEED MARTIN ENERGY SYSTEMS, INC.
for the
DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400**

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The U.S. Army's Project Manager, Crusader is sponsoring the development of technologies that apply to the Self-Propelled Howitzer (SPH), formerly the Advanced Field Artillery System (AFAS), and Resupply Vehicle (RSV), formerly the Future Armored Resupply Vehicle (FARV), weapon system. Oak Ridge National Laboratory (ORNL) is currently performing developmental work in support of the SPH/RSV Crusader system. Supportive analyses of reliability, availability, and maintainability (RAM) aspects were also performed for the SPH/RSV effort. During FY 1994 and FY 1995 ORNL conducted a feasibility study to demonstrate the application of simulation modeling for RAM analysis of the Crusader system. Following completion of the feasibility study, a full-scale RAM simulation model of the Crusader system was developed for both the SPH and RSV. This report provides documentation for the simulation model as well as instructions in the proper execution and utilization of the model for the conduct of RAM analyses.

2. DISCRETE EVENT SIMULATION MODELING

Simulation is a powerful modeling framework for analysis of complex system performance. Simulation models are structured in the *time domain*, where the flow of events and materials can be observed and evaluated. Discrete-event simulation is a suitable tool for modeling systems that can be decomposed into a set of discrete chronological steps or tasks. The advantages of decomposing a complex system into smaller steps are that: (1) it is often easier to describe the behavior of constituent parts of a process than to describe the whole and (2) the performance of the whole system can be studied by varying the behavior of the constituent parts.

In discrete event simulation, time is not incremented in small, equal-sized steps but in irregular steps which mark the instantaneous start and completion times of events. Elapsed time during individual task execution is not explicitly represented. A discrete event simulation may be deterministic or stochastic. A deterministic model executes in identical fashion each time it runs. A stochastic model schedules task completion times by taking random draws from probability density functions (PDFs). The PDFs are used to represent the potential execution times for each task. Therefore, instead of modeling the execution time for a task with a point estimate, a range of possible execution times are used. Interval estimates allow the simulation model to represent not only the typical behavior of the system, but a wide range of expected system behavior.

One type of discrete event simulation is task network simulation. Network simulation offers a powerful technique for analyzing RAM characteristics of complex systems. Although a detailed description of task network simulation is beyond the scope of this document, additional information can be found elsewhere.[1] One of the available commercial software packages for task network simulation is MicroSaint, which is marketed by Micro Analysis and Design, Inc.

A network diagram shows the constituent parts of the simulation model. It is essentially a task block flow diagram where nodes represent activities of the system. In order to create a network, a task analysis is performed to identify the activities of the system. In addition, decisions that determine paths followed after completion of each activity are established. The network diagram can be implemented as a computer-based network simulation model using a network simulation package such as MicroSaint.

A MicroSaint task network consists of four primitives: nodes, branches, entities, and queues. Nodes are subtasks which represent discrete activity of the system. Discrete activities are punctuated by start times and stopping times and have clearly identifiable predecessors and followers. The network has a unique start task and may have multiple stop tasks. The duration of a specific task is determined by a random draw from a time distribution. The user provides values for free parameters of the distribution selected to model task duration. Typically, the mean and standard deviation are selected. Many distribution types are supported, including the normal, gamma, exponential, Weibull, etc. By selecting a new random number seed at the start of each run of the model, different results may be obtained. Entities are modeled as concurrent processes which traverse the simulation network. Entities are the actors which provide dynamic simulation capability by actually executing tasks residing in the static network. An entity might be an operator or a machine, but in the present context most of the entities are RSV or SPH vehicles. The model also spawns control entities to perform periodic adminstrative tasks and failure scheduling.

Branches define predecessor relationships among subtasks. Four types of branching may be employed: single, multiple, probabilistic, and tactical. A single branch decision type is deterministic. The following task is always executed after termination of the current task. Network models that only utilize single branching logic trace a deterministic route during every run; no other routes are possible. The introduction of conditional branching logic into the network allows for the possibility of different routes being taken over the course of multiple runs. At a node with a probabilistic decision type, only one of several candidate paths is randomly selected. A tactical decision is similar to a probabilistic decision except that the path with the greatest "value" according to user-provided criteria is selected. These four branching types allow for complex decision modeling to be employed in the network.

Queues are coupled with specific nodes such that entities arriving at the node wait in the queue until release conditions for the task have been satisfied.

Simple networks process tasks one at a time in sequential fashion. More complex networks allow multiple entities to propagate through the network in parallel fashion. Thus it is possible for multiple tasks to be processed concurrently in such networks. Tactical and multiple branching also create the possibility that entities will follow novel paths through the network on different simulation runs.

A network of tasks may be organized as an abstraction hierarchy, where the top level reveals the most general view of the operation of the network. Each box in the top level contains a subnetwork, which in turn may contain its own subnetwork, etc. Successive levels in the hierarchy show more detail of smaller parts of the entire system.

3. OVERVIEW OF THE TASK NETWORK

The missions of the SPH and RSV were modeled using the MicroSaint network simulation toolkit. Failures and repairs were also modeled for major RSV and SPH systems and the RSV resupply subsystem. Figure A.1 provides an overview of the top-level network. Simulation parameters are initialized in the Start task. Upon completion of the Start task, the RSV, SPH, and failure management task networks are initiated in parallel.

The RSV subnetwork is shown in Fig. A.2. The RSV uploads at the logistical rearm point (LRP) and then travels to a hide position. When the RSV receives a resupply request from the SPH, it travels to the SPH; otherwise, it defaults to a wait mode. At periodic intervals, the RSV subnetwork checks for evidence of SPH communication and evaluates its duration of stay in a single location. If the SPH has not been in contact and the RSV has remained idle long enough, it performs a tactical move and then returns to the wait mode. Otherwise, no action is taken. After rearming, the RSV returns to the LRP if it is low on rounds; otherwise it travels to a new hide position. After traveling to the LRP, crew rotation occurs if a rest period is required. The RSV continues to the first hide position after uploading.

The SPH begins by advancing to a firing position, as represented in Fig. A.3. If the time interval since the last firing mission has exceeded the threshold, another mission is executed. Otherwise the SPH enters a wait mode. At periodic intervals in the wait mode, the SPH evaluates its options. If the SPH has rounds, a mission is executed if enough time has elapsed. Otherwise, a tactical move is scheduled if the SPH has remained stationary for a sufficient time period. If the SPH is low on rounds, the RSV may be summoned. The conditions which must be met for a summons are that the RSV is not being uploaded and a mission or tactical move is not scheduled. No action is taken if a mission, tactical move, or rearming is not required. The same logic is followed if the SPH is not armed except that mission execution is not examined as an option. A firing mission is always followed by a survivability move and then a decision as to whether rearming is required. The SPH waits to rendezvous with the RSV if the number of rounds is sufficiently small and the RSV is available. Otherwise, the SPH enters the wait mode. After rearming, the SPH responds with a tactical move. If the SPH is in the wait mode for a long period, it also performs a tactical move.

Scheduling of failures is tracked independently in the failure management subnetwork, as depicted in Fig. A.4. Fourteen systems are tracked for both vehicles. The RSV resupply system (see Fig. A.5) is modeled to the level of eight subsystems. Each system and subsystem is managed by its own task. The tasks run independently for each vehicle. (It is possible for two versions of the same task to overlap and be active at the same time.) An operational time to failure is assigned to each system or subsystem on the basis of a random draw from a Gamma distribution. During the execution of operational tasks in the SPH and RSV subnetworks, the failure times are decremented until they are equal to or less than zero. At this point a failure is scheduled for the appropriate system. A flag is set which allows the failure management task to select a new time to failure. The procedures are accomplished with the use of user-defined functions. The functions may be viewed by selecting the *Function Library* option under the *Display* menu. The failure management functions are denoted by the "*_tmods" naming convention.

The network activity terminates after the first resupply following the specified stop time. Typical run length is 1 week. Time steps are in units of minutes.

3.1 SIMULATION MODEL ASSUMPTIONS

The major assumptions inherent in the network simulation model are provided in this section. The battlefield configuration consists of a single RSV coupled with a single SPH. The RSV and SPH are modeled separately in their own subnetworks, but their interactions are synchronized. Rearming is initiated and extends through identical time intervals for both RSV and SPH.

All systems begin as new and are considered repairable as new. Failures in the RSV and SPH are modeled at the system level. RSV resupply is modeled at the subsystem level. Mean Time Between Failures (MTBF) is constant throughout the entire mission. Failure occurrence times are stochastic and independent. Failures are independent samples from an exponential distribution with mean equal to the appropriate MTBF. All failures involve essential functions which abort the mission and are detected immediately. Repair activity begins immediately and occurs at the location in which the failure is discovered. Failures and repair processes do not affect the time required to complete the interrupted task other than to introduce a time lag representing the repair activity.

Task times are sampled from a gamma distribution. Mean times are user selectable. The default values are based on available data. The model provides an estimate for the standard deviation using a nonlinear function of the mean.

4. INPUT VARIABLES

The simulation input parameters may be modified in the Start (1) task. The Start task is where all parameters are initialized at the beginning of each simulation run. An input parameter value can be modified by editing the right-hand side of the assignment statement in which the parameter name appears. Natural language descriptions of each input parameter name and their initial values are listed in Text Boxes 1–4. The values in units of minutes of MTBF for each system are provided in Text Box 1. MTBF variables are exponentially distributed in the simulation model. Text Box 2 contains the mean repair times for various systems and subsystems. Mean travel times are given in Text Box 3. They are modeled using the gamma distribution, where the standard deviation is computed as a nonlinear function of the mean. All other input parameters are described in Text Box 4.

Six multiplier input variables are defined at the top of the input variable assignments in the Start task. They are described as FARV failure/MTBF multiplier (*ffm*), AFAS failure/MTBF multiplier (*afm*), FARV repair multiplier (*frm*), AFAS repair multiplier (*arm*), FARV travel time multiplier (*ftm*), and AFAS travel time multiplier (*atm*). These constants can be used to simultaneously modify entire classes of input variables. The default values of the multipliers are unity, so that they will have no effect on the following assignment statements. If, for example, the variable *ffm* is set to 0.5, then all MTBF variables for the FARV vehicle will be halved (and failure rates will double). Setting the *ffm* variable to 2.0 doubles the MTBF for all FARV systems (and halves the failure rates).

Text Box 1		
ammo_mtbf	21720	SPH auto. ammunition handling system mtbf
ammop_mtbf	373320	RSV ammunition processing subsystem mtbf
aux_mtbf[1]	38940	RSV auxiliary systems mtbf
aux_mtbf[2]	61020	SPH auxiliary systems mtbf
comm_mtbf[1]	20071900	RSV communication/ident. sys mtbf
comm_mtbf[2]	17985600	SPH communication/ident. sys mtbf
defarm_mtbf[1]	100000000	RSV defensive armament mtbf
defarm_mtbf[2]	22754	SPH defensive armament mtbf
display_mtbf[1]	819360	RSV data display and controls system mtbf
display_mtbf[2]	975240	SPH data display and controls system mtbf
dock_mtbf	697680	RSV docking subsystem mtbf
fire_mtbf[1]	100000000	RSV fire control system mtbf
fire_mtbf[2]	31920	SPH fire control system mtbf
fuel_mtbf	269040	RSV fuel transfer subsystem mtbf
navig_mtbf[1]	31000000	RSV navigation system mtbf
navig_mtbf[2]	17305125	SPH navigation system mtbf
nbc_mtbf[1]	161400	RSV NBC and environmental control system mtbf
nbc_mtbf[2]	144600	SPH NBC and environmental control system mtbf
power_mtbf[1]	13920	RSV power package and drive train mtbf
power_mtbf[2]	39240	SPH power package and drive train mtbf
prarm_mtbf	20040	SPH primary armament systems mtbf
prjsh_mtbf	210540	RSV projectile S&H subsystem mtbf
prjtr_mtbf	126060	RSV projectile transfer subsystem mtbf
prpsh_mtbf	80880	RSV propellant S&H subsystem mtbf
prptr_mtbf	2608680	RSV propellant transfer subsystem mtbf
prpup_mtbf	2608680	RSV propellant upload subsystem mtbf
struct_mtbf[1]	1246140	RSV vehicle housing and structure mtbf
struct_mtbf[2]	1554720	SPH vehicle housing and structure mtbf
surv_mtbf[1]	100000000	RSV survivability system mtbf
surv_mtbf[2]	100380	SPH survivability system mtbf
suspen_mtbf[1]	104280	RSV suspension and steering mtbf
suspen_mtbf[2]	145680	SPH suspension and steering mtbf
velec_mtbf[1]	62280	RSV vehicle electronics system mtbf
velec_mtbf[2]	28200	SPH vehicle electronics system mtbf

Text Box 2		
ammo_rep	154	SPH auto. ammunition handling mean repair time
ammop_rep	32	RSV ammo processing subsystem mean repair time
aux_rep[1]	111	RSV auxiliary systems mean repair time
aux_rep[2]	124	SPH auxiliary systems mean repair time
comm_rep[1]	14	RSV communication/ident. mean repair time
comm_rep[2]	10	SPH communication/ident. mean repair time
defarm_rep[1]	15	RSV defensive armament mean repair time
defarm_rep[2]	15	SPH defensive armament mean repair time
display_rep[1]	124	RSV data display & controls system mean repair time
display_rep[2]	129	SPH data display & controls system mean repair time
dock_rep	60	RSV docking subsystem mean repair time
fire_rep[1]	55	RSV fire control system meanrepair time
fire_rep[2]	60	SPH fire control system meanrepair time
fuel_rep	66	RSV fuel transfer subsystem mean repair time
navig_rep[1]	56	RSV navigation system mean repair time
navig_rep[2]	49	SPH navigation system mean repair time
nbc_rep[1]	45	RSV NBC/environmental control system mean repair time
nbc_rep[2]	46	SPH NBC/environmental control system mean repair time
power_rep[1]	455	RSV power package & drive mean repair time
power_rep[2]	623	SPH power package & drive mean repair time
prarm_rep	59	SPH primary armament system mean repair time
prjsh_rep	95	RSV projectile S&H subsystem mean repair time
prjtr_rep	88	RSV projectile transfer subsystem mean repair time
prpsh_rep	63	RSV propellant S&H subsystem mean repair time
prptr_rep	60	RSV propellant transfer subsystem mean repair time
prpup_rep	30	RSV propellant upload subsystem mean repair time
struct_rep[1]	46	RSV vehicle housing and structure mean repair time
struct_rep[2]	53	SPH vehicle housing and structure mean repair time
surv_rep[1]	72	RSV survivability system mean repair time
surv_rep[2]	73	SPH survivability system mean repair time
suspen_rep[1]	68	RSV suspension and steering mean repair time
suspen_rep[2]	132	SPH suspension and steering mean repair time
velec_rep[1]	30	RSV vehicle electronics system mean repair time
velec_rep[2]	31	SPH vehicle electronics system mean repair time

Text Box 3		
ex_miss_tm	7.5	mean time to execute mission (min)
rearm_rt	0.25	mean time to rearm SPH (minutes per round)
rest_tm	15	resting or shift rotation period (min)
set_tm	4	mean setup time for rearming SPH (min)
surv_tm	1.6	mean time to perform survivability move (min)
tact_tm	3	mean time for RSV to perform tactical maneuver (min)
tact2_tm	3	mean time for SPH to perform tactical maneuver (min)
tv_afas_tm	3.22	mean time to reach SPH (min)
tv_fire_tm	12	mean time for SPH to travel to firing position (min)
tv_hid1_tm	18	mean time to reach first hide position (min)
tv_hid2_tm	1.395	mean time to reach second hide position (min)
tv_lrp_tm	21.325	mean time to travel to LRP (min)
upload_tm	65	mean time to upload at LRP (min)

Text Box 4		
end_run	10080	time to end run
exe_int	60	preferred interval between missions (min)
full	60	maximum load for SPH
miss_round	16	number of rounds used on a single mission
move_thr	120	time limit for RSV to remain in single location (min)
move2_thr	30	time limit for SPH to remain in single location (min)
min_rounds	60	initial minimum number of rounds
Rounds2	60	initial number of rounds on SPH
rrm_thr	2	rearm threshold (no. of missions)
sdcon1	0.555	linear parameter for std. dev. of time
sdcon2	0.004	quadratic parameter for std. dev. of time
shift	480	shift length (min)
up_round	130	number of rounds uploaded to RSV
wait_inc	15	time to wait to check for call from RSV or move (min)

5. PERFORMANCE (OUTPUT) VARIABLES

The simulation model creates three output files following each batch of runs. The blueprints for these files can be reviewed by selecting *Snapshots* under the *Display* menu. The three snapshot or output files are named ENDRUN, FAILCOUNTS, and QUARTER-HOURLY. Summary statistics contained in the ENDRUN file are listed in Text Box 5.

Availability metrics for the SPH and RSV are computed according to the following expression:

$$\frac{\text{Uptime}}{(\text{Uptime} + \text{Downtime})}, \tag{1}$$

where Downtime is essentially total time to repair.

Vehicle MTBF metrics were computed by dividing total run time by the total number of failures for the vehicle. The vehicle MTBF is undefined in case there are no failures and is designated by the value 9999 in the snapshot file. For the default case, where run lengths are approximately 1 week, vehicle MTBF values are undefined for approximately 75% of the total number of runs. An important consequence of this fact is that summary statistics for vehicle MTBFs must be handled uniquely.

Due to the stochastic nature of discrete event simulation, many runs are required to produce reliable interval estimates of output variables. Normally, measures of central tendency for each variable are obtained by taking the simple mean of that variable over the set of runs. Since vehicle MTBF is frequently undefined for a single run, the procedure of the sample mean is not viable for this statistic. The vehicle MTBF for a single run is defined as:

$$\frac{\text{Run length}}{\text{Number of vehicle failures}}. \tag{2}$$

The vehicle MTBF for a set of runs is:

$$\Sigma_i = \frac{\text{Run length}_i}{\text{Number of vehicle failures}_i}, \tag{3}$$

where Eq. (3) is summed over the entire sample. Equation (3) is just the sum of run lengths divided by the sum of the number of vehicle failures. These sums are easily obtained from a spreadsheet analysis. Most spreadsheets can import MicroSaint snapshot files because they are ASCII text files.

Text Box 5	
AFAS_empty	number of times SPH is empty
afas_mtbf	omnibus SPH mean time between failure
afas_num	total number of SPH failures
afas_out	total time SPH is out of service or failed (min)
afas_per	total time SPH is out of service or failed (%)
avail	RSV availability
avail2	SPH availability
ave_lead	average time to wait for resupply (min)
crus_mtbf	omnibus Crusader system mean time between failure
fail_tot	total time RSV is out of service due to failure (min)
fail2_tot	total time SPH is out of service due to failure (min)
farv_mtbf	omnibus RSV mean time between failure
farv_num	total number of RSV failures
lead_per	total time SPH waits for resupply (%)
lead_tot	total time SPH waits for resupply (min)
LRP_trips	number of trips to LRP
max_lead	maximum time to wait for resupply (min)
min_rounds	minimum number of rounds on SPH
miss_num	number of executed missions
outserv_per	total time SPH is without rounds (%)
outserv_tot	total time SPH is without rounds (min)
rearm_per	total rearming time (%)
rearm_tot	total rearming time (min)
rest_per	total time used for crew rotation (min)
rest_tot	total time used for crew rotation (min)
resupply	number of times SPH is resupplied
rounds_phr	average number of rounds transferred per hour
tact_moves	number of tactical moves by RSV
tact_moves2	number of tactical moves by SPH
tact_per	total tactical move time for RSV (%)
tact_tot	total tactical move time for RSV (min)
tv_afas_per	total travel time from hide position to SPH (%)
tv_afas_tot	total travel time from hide position to SPH (min)
tv_hid1_per	total travel time from LRP to hide position (%)
tv_hid1_tot	total travel time from LRP to hide position (min)
tv_hid2_per	total travel time from SPH to hide position (%)
tv_hid2_tot	total travel time from SPH to hide position (min)
tv_lrp_per	total travel time to LRP (%)
tv_lrp_tot	total travel time to LRP (min)
upload_per	total uploading time (%)
upload_tot	total uploading time (min)
wait_per	total time RSV is in wait mode (%)
wait_tot	total time RSV is in wait mode (min)

The end-of-run snapshot variable descriptions, listed in Text Box 6, are self-explanatory for the most part. A few output variables require a little further explanation. The variable "ave_lead" is obtained by dividing "lead_tot" by "resupply," or the total lead time divided by the number of resupplies. The lead time for a given resupply period is the interval from the moment the SPH communicates with the RSV to the completion of actual resupply efforts. The communication normally occurs immediately following the execution of a mission, but it can also occur following a resupply period during which the RSV did not have sufficient rounds to equip the SPH. In this case, the RSV returns to the LRP for uploading, and the SPH waits to receive its full complement of rounds. The lead time can reach large values if the RSV fails during the period it is attempting to reach the SPH for resupply.

The SPH is not necessarily idle during the entire lead time period. The SPH may enact a survivability move or execute a mission (if it is still armed) during that period if conditions require these actions. However, most of the lead time period is normally completed by the SPH in the wait mode.

The variable "wait_tot" should not be confused with "lead_tot." The "wait_tot" refers to the time spent in wait mode by the RSV, whereas "lead_tot" is specific to the SPH. The time spent by the SPH in wait mode is not tracked by the Crusader model in Version 1.0.

The following "count" performance variables are found in the FAILCOUNTS snapshot file. Count variables sort failures either by system (*_num) or RSV activity (*_fail).

Nine performance variables (see Text Box 7) are tracked and snapshots are saved at 15-min intervals in the snapshot file called QUARTER-HOURLY. All variables stored in this file are also stored in ENDRUN, except ROUNDS and ROUNDS2.

Text Box 6	
ammo_num[2]	number of SPH auto ammunition handling system failures
ammop_num	number of RSV resupply ammunition processing subsystem failures
aux_num[1]	number of RSV auxiliary system failures
aux_num[2]	number of SPH auxiliary system failures
comm_num[1]	number of RSV communication/identification system failures
comm_num[2]	number of SPH communication/identification system failures
defarm_num[1]	number of RSV defensive armament system failures
defarm_num[2]	number of SPH defensive armament system failures
display_num[1]	number of RSV data display and control system failures
display_num[2]	number of SPH data display and control system failures
dock_num	number of RSV resupply docking subsystem failures
fuel_num	number of RSV resupply fuel transfer subsystem failures
fire_num[1]	number of RSV fire control system failures
fire_num[2]	number of SPH fire control system failures
navig_num[1]	number of RSV navigation system failures
navig_num[2]	number of SPH navigation system failures
nbc_num[1]	number of RSV NBC and env. control system failures
nbc_num[2]	number of SPH NBC and env. control system failures
power_num[1]	number of RSV power package and drive train system failures
power_num[2]	number of SPH power package and drive train system failures
prarm_num[2]	number of SPH primary armament system failures
prjsh_num	number of RSV resupply projectile S&H subsystem failures
prjtr_num	number of RSV resupply projectile transfer subsystem failures
prpsh_num	number of RSV resupply propellant S&H subsystem failures
prptr_num	number of RSV resupply propellant transfer subsystem failures
prpup_num	number of RSV resupply propellant upload subsystem failures
struct_num[1]	number of RSV vehicle structure system failures
struct_num[2]	number of SPH vehicle structure system failures
surv_num[1]	number of RSV survivability system failures
surv_num[2]	number of SPH survivability system failures
suspen_num[1]	number of RSV suspension and steering system failures
suspen_num[2]	number of SPH suspension and steering system failures
velec_num[1]	number of RSV vehicle electronics system failures
velec_num[2]	number of SPH vehicle electronics system failures
rrm_fail	number of RSV failures during rearming
tact_fail	number of RSV failures during tactical moves
tafas_fail	number of RSV failures during travel to SPH
thd1_fail	number of RSV failures during travel to first hide position
thd2_fail	number of RSV failures during travel to second hide position
tlrp_fail	number of RSV failures during travel to LRP
upld_fail	number of RSV failures during uploading

Text Box 7	
afas_out	total time SPH is out of service or failed (min)
avail	RSV availability
avail2	SPH availability
fail_tot	total time RSV is out of service due to failure (min)
fail2_tot	total time SPH is out of service due to failure (min)
lead_tot	total time SPH waits for resupply (min)
outserv_tot	total time SPH is without rounds (min)
Rounds	total number of rounds aboard the RSV
Rounds2	total number of rounds aboard the SPH

6. RUNNING THE RAM SIMULATION MODEL WITH MICROSAINT

The instructions provided in this section are specific to the version of MicroSaint[5] for Macintosh, but many procedures will be identical for the Windows version. Windows-specific instructions are italicized in parentheses.

6.1 HARDWARE AND SOFTWARE REQUIREMENTS

The Crusader RAM simulation model will execute on Windows 3.0 (or higher) and Macintosh platforms. The only software requirement is the MicroSaint application (Version 1.2 for the Macintosh and Version 1.3 for Windows). At least 4 MB of random access memory (RAM) are required for both Windows and Macintosh platforms; however, 8 MB of RAM are recommended. A hard drive containing at least 5 MB and a floppy disk drive are also required. A color monitor is recommended. At least 512 KB of internal memory and a mouse are needed for an IBM-compatible platform, or a MacPlus or higher.

The simulation models have been tested on a Power Macintosh 7100/66AV with 24 MB of RAM and a Northgate 80486DX-33MHz with 16 MB of RAM. A single 1-week run of the Action View animation version of the model on the Power Macintosh under default conditions uses on average about 7 min for execution under the fastest speed, and about 45 s for initialization. The version of the model without *Action View* animation can execute in about 4.5 min with about 20 s required for initialization. The MicroSaint application has not been optimized for the PowerPC chip, and so it must run in a considerably slower emulation mode on a Power Macintosh. The same model executes in approximately 56 s as a Windows application with virtually no time required for initialization. The version of the model that does not include the *Action View* animation executes in about 24 s. Presumably, a Pentium platform would execute the model in even less time than an 80486 processor.

6.2 LAUNCHING AND SETTING UP THE MODEL

Launch MicroSaint by double-clicking on the application icon, or directly on the model document icon. Select *Open* from the *File* menu, and then select "CRUS10.MOD" or "CRUS10AV.MOD" from the list of models. The top-layer network should be visible in the workspace. If you desire to view subnetworks, select the subnetwork of interest by clicking on the appropriate rectangle. Then choose *Open Diagram* from the *Display* menu. *(Press the "Down" button from the top of the network window.)* Other windows may be opened using the same procedure. *(In the Windows version, only one network view may be open at a time.)* Note that double-clicking the icon instead of using *Open Diagram* will cause a dialogue box for the subnetwork or task to appear.

The tool palette on the left side of the network window is not normally used during model execution. Most of the tools are used for network development. Normally the user should ensure that the top "selection" or "arrow" tool is highlighted. The magnifying glass tool can be used to enlarge the network view, and the telescoping tool will shrink the network view. Telescoping may be useful if the user desires to simultaneously view many subnetworks.

The *Display* menu indicates that other objects that can be viewed are the *Variable Catalog*, *Function Library*, *Event Queue*, *Snapshots*, and *Execution Monitor*. The variable catalog will present a list of all model variables together with their current values. The execution monitor is a useful runtime tool that the user may configure to show a short list of the current values of the most informative variables. It also shows the current simulation time in the upper right-hand corner and the current run number in the upper left-hand corner. Select the "Add" button to add new variables to the list shown on the *Execution Monitor*. To delete a variable, simply highlight it and press the delete key. The values of complex expressions, such as the addition of a list of variables, can be added to the execution monitor as well as single variables. Each statement must end with a semicolon. Text Box 8 shows the default expressions listed on the execution monitor in CRUS10.MOD.

Text Box 8	
avail;	{FARV availability}
lead_tot;	{total AFAS waiting time}
fail2_tot;	{AFAS tot. failure time}
Rounds;	{FARV}
Rounds2;	{AFAS}
stny_tm;	{FARV time in one pos.}
stny2_tm;	{AFAS time in one pos.}
last_miss;	{time of last mission}
fail_tot;	{FARV total failure time}
avail2;	{AFAS availability}
outserv_tot;	{tot time AFAS w/o rnds}
afas_out;	{oos and failure time}

The *Function Library* and *Snapshots* windows do not present any dynamic runtime information of interest. However, when performing a large sequence of runs, it is best to leave one of these two windows open. Close all other MicroSaint windows. This practice will result in faster model execution due to the reduced memory requirements for window management.

The *Event Queue* provides a dynamic list of all scheduled events in the model from closest to furthest away in simulation time. It is sometimes interesting to observe the event queue, but it normally slows model execution significantly; therefore, it is usually advisable to keep this window closed.

If the user selects *Show* under the *Action View* menu, the iconic animation for the model is revealed. The user should resize the window so that the entire background is visible. The background should appear if the "CRUS10AV.MOD" or AV version of the model is loaded. The background shows the LRP region on the left side of the window, and a column of enemy tanks on the right side of the window. At the upper left is a status box with labels for the RSV and SPH. During a simulation run, a cross appears to the right of each vehicle label to indicate that the system is fully operational. The presence of a system abort and repair process is shown by a circle instead of a cross. Icons representing the RSV and SPH move across the background to indicate their activities. The SPH moves up and down in a forward position to execute missions and perform survivability moves. Mission execution is depicted as a release of ammunition rounds. The RSV icon docks with the SPH icon during resupply, and an arrow icon represents the "boom" connecting the two vehicles. Normally the RSV remains in a rear position close to the center of the window, moving only when required for survivability. The RSV icon visits the LRP region for uploading, which is depicted by a human icon walking out to the RSV icon.

The various model windows can be tiled or overlapped to present an optimum array of information flow. Since monitor space is limited, it is best to restrict the number of windows open at a given time to one or two.

Another important option for executing the model is *Settings...* under the *Execute* menu. This dialog box allows the user to customize several aspects of the actual execution. The top option specifies the random number seed. The default seed value, 1, is adequate for most purposes. If the user desires to rerun a particular run, he may enter the seed number for that run, and the model will replicate the results for that run.

Four checkboxes follow the random number seed box. The simulation user will want to keep *Trace of Tasks* and *Standard Deviations Set to Zero* unchecked. The *ActionView Functions* box should always be checked to allow the iconic animation to run should it be needed. The *Snapshots of Variables* box should be checked only if the user requires the generation of simulation output files. Otherwise it should be unchecked. This box can remain checked even if output files are not desired, but the model will run more slowly.

Below the checkboxes is another number box where the user indicates the number of times to execute the model. The number of runs required for Monte Carlo sensitivity studies is addressed in Sect. 6.4.

The radio button marked *Symbols* should be selected at the bottom of the dialogue box in the area labeled *Show Entities With:*.

6.3 MODEL EXECUTION

The *Execution* menu contains the relevant options for controlling execution of the Crusader simulation model. The model execution is initiated with the *Go* command and terminated with the *Halt* command. Model execution can be stopped temporarily by selecting the *Pause* command. If the *Go* option is selected after *Pause*, the simulation will resume, preserving the state of the system when *Pause* was first selected. *Single-step* is a run mode used primarily for debugging and will advance the simulation by a single event before pausing. *Halt* stops model execution like *Pause*, except that the current state of the system is lost. If *Go* is selected after *Halt*, the simulation will reinitialize and begin a new run or set of runs. Normal termination of the simulation follows with a sequence of dialog boxes which prompt the user to provide filenames for the snapshot files. Selecting *Halt* will also result in the appearance of the same dialog boxes.

The rate of model execution can also be controlled in the *Execution* menu. *Normal speed* is the default option. Selecting *Top speed* will ramp model execution to its fastest rate. The execution speed can be incremented or decremented from the current rate by selecting *Faster* or *Slower*.

In the *Action View* menu, in addition to selecting animation speed, the user is given the options labeled *dots*, *lines*, and *refresh*. Checking either the *dots* or *lines* options will leave trails or time histories of icon movements for both the SPH and RSV. Checking *refresh* will erase the time history lines.

6.4 NUMBER OF RUNS

The user selects more than one run in the *Settings...* menu in case more stable estimators for simulation performance variables are desired. For example, the value of SPH availability obtained from a single run cannot be regarded as being very reliable. Since runs are not deterministic, a different value will likely be obtained by executing the model a second time. A different result is ensured because the model begins with a new seed number for the second (and each subsequent) run. It is often desirable to establish a typical value for each performance measure, and an interval around that value which contains the "true" value of the performance measure with a very high degree of confidence. The precision of the estimator is directly related to interval size, where more precision is associated with smaller interval sizes. A power analysis is performed to establish interval sizes as a function of sample size. The analyst can then work backward from required interval size in order to find the required sample size.

The mean value of the sample is normally taken as the point estimate or "typical" value for the performance measure (although the mode is also frequently used). Procedures for interval estimation, or obtaining confidence intervals, can be found in any introductory Statistics textbook.[2] Most procedures take advantage of the fact that the distribution of sample means is Normal. A symmetric interval estimate (confidence interval) for a population mean value can be formed using the sample mean according to Eq. (4):

$$CI_\alpha = \overline{X} \pm z_\alpha \cdot \sigma_{\bar{x}} \ldots \tag{4}$$

Equation (4) states that the confidence interval (CI) is the sample mean plus or minus the standard error multiplied by a constant. The standard error is just the standard deviation of the sample mean, and is computed by dividing the sample standard deviation by the square root of the sample size. Fortunately, the distribution of the sample mean is always the same, regardless of the distribution of the underlying variable, if the sample size is large enough. The distribution of the sample mean is asymptotically Normal. Therefore the Standard Normal Table (which can be found in any standard statistics book (e.g., ref. 2) can be used to choose a multiplier constant that will generate a CI of desired reliability/precision.

The reliability of a CI is usually expressed as a percentage (100*(1-α)). For example, a 90% CI is an interval about the sample mean that has a 0.9 probability of containing the "true" population mean. A higher percentage signifies a more robust CI, but it is achieved at the expense of less precision. There is always a tradeoff between precision and reliability in forming a CI for a fixed sample size. An effective strategy to simultaneously increase both precision and reliability is to increase sample size.

Tables 1 through 5 list values for the size or precision of the CI using the method embodied in Eq. (4). These values were obtained under baseline conditions. The simulation analyst should recognize that different results may be obtained if the model is run under a different set of conditions. For example, higher failure rates will probably increase the standard error of the mean.

The size of the CI is given for several sample sizes, including 50, 100, 250, 500, 1000, 2500, 5000, and 10,000. Five levels of reliability for the confidence interval are given in each column, including 90%, 95%, 97.5%, 99%, and 99.9%. Linear interpolations between cells may be used to obtain approximate CI sizes for sample sizes not included in the tables.

Table 1. Sample size values based on RSV availability

RSV availability	90%	95%	97.5%	99%	99.9%
50	0.438	0.520	0.594	0.684	0.875
100	0.309	0.368	0.420	0.484	0.619
250	0.196	0.232	0.266	0.306	0.391
500	0.138	0.164	0.188	0.216	0.277
1000	0.098	0.116	0.133	0.153	0.196
2500	0.062	0.074	0.084	0.097	0.124
5000	0.044	0.052	0.059	0.068	0.088
10000	0.031	0.037	0.042	0.048	0.062

Table 2. Sample size values based on SPH availability

SPH availability	90%	95%	97.5%	99%	99.9%
50	0.285	0.338	0.387	0.445	0.570
100	0.201	0.239	0.273	0.315	0.403
250	0.127	0.151	0.173	0.199	0.255
500	0.090	0.107	0.122	0.141	0.180
1000	0.064	0.076	0.086	0.100	0.127
2500	0.040	0.048	0.055	0.063	0.081
5000	0.028	0.034	0.039	0.045	0.057
10000	0.020	0.024	0.027	0.031	0.040

Table 3. Sample size values based on average lead time

Average lead time	90%	95%	97.5%	99%	99.9%
50	0.754	0.895	1.023	1.179	1.508
100	0.533	0.633	0.724	0.833	1.066
250	0.337	0.400	0.458	0.527	0.674
500	0.238	0.283	0.324	0.373	0.477
1000	0.169	0.200	0.229	0.264	0.337
2500	0.107	0.127	0.145	0.167	0.213
5000	0.075	0.090	0.102	0.118	0.151
10000	0.053	0.063	0.072	0.083	0.107

Table 4. Sample size values based on number of resupplies

Number of resupplies	90%	95%	97.5%	99%	99.9%
50	0.380	0.451	0.516	0.594	0.760
100	0.269	0.319	0.365	0.420	0.537
250	0.170	0.202	0.231	0.266	0.340
500	0.120	0.143	0.163	0.188	0.240
1000	0.085	0.101	0.115	0.133	0.170
2500	0.054	0.064	0.073	0.084	0.107
5000	0.038	0.045	0.052	0.059	0.076
10000	0.027	0.032	0.036	0.042	0.000

Table 5. Sample size values based on total wait time (RSV)

Total wait time	90%	95%	97.5%	99%	99.9%
50	50.1	59.5	67.9	78.3	100.1
100	35.4	42.0	48.0	55.3	70.8
250	22.4	26.6	30.4	35.0	44.8
500	15.8	18.8	21.5	24.7	31.7
1000	11.2	13.3	15.2	17.5	22.4
2500	7.1	8.4	9.6	11.1	14.2
5000	5.0	5.9	6.8	7.8	10.0
10000	3.5	4.2	4.8	5.5	7.1

CI sizes were computed for RSV availability, SPH availability, average lead time, number of resupplies, and total wait time. It was not possible to perform a power analysis on the MTBF statistic, for reasons described in Sect. 4. Since many performance measures are potentially of interest to the analyst, it is best to perform the power analysis for a number of variables instead of only one variable.

One way to use the tables is to select a pair of values representing minimum reliability and minimum interval size or precision. The reliability level will fix the column, and the interval size will determine the row. Read across to the leftmost column from the fixed row and column to find the required sample size. For example, suppose we want to be 99% certain that the population mean differs from the sample mean by no more than 0.100. For simplicity we will consider only RSV availability (Table 1) in performing the power analysis. Looking down the column headed by 99%, we see 0.153 and 0.097 at the 5th and 6th rows. The sixth row is the most conservative because it is slightly less than 0.100. Now if we read to the left of the value 0.097, we notice that a sample size of 2500 is needed to meet our reliability and precision requirements. We will probably want to perform the same table lookup using a few other performance measures with the other tables to see if a sample size of 2500 is consistently adequate.

The method of obtaining the CIs used above is based on distributional assumptions that may not be accurate, even for fairly large sample sizes. An example is availability, which frequently may have a value of 100 for large MTBFs. Therefore the sampling distribution of mean availability will be highly skewed toward the value of 100. An alternate method for estimating CI size is called the bootstrap method. Bootstrapping is a statistical estimation technique that does not rely on any distributional assumptions. The technique involves resampling with replacement using the original sample values in order to assemble an empirical distribution based on characteristics of the original sample. The areas under the empirical distribution can be calculated directly and exactly. The bootstrap procedure is inherently useful for analysis of simulation output data because it is, itself, a simulation procedure.[3,4]

Bootstrap interval estimators were obtained using resamples of size 40 from simulation baseline data. Tables 6 and 7 respectively provide 90% and 95% CI estimators for the five selected performance measures. Values for sample sizes 50, 100, 250, 500, and 1000 are tabled. Bootstrap estimates are not necessarily symmetric about the sample mean, so both lower and upper confidence limits are listed. The size of the interval estimates can be calculated by subtracting lower from upper. The sample means are respectively 99.24%, 99.61%, 28.35 min, 67.23, and 6871 min for RSV availability, SPH availability, average lead time, number of resupplies, and total wait time.

Table 6. 90% CI bootstrap estimates

	RSV availability		SPH availability		Average lead time		Number of resupplies		Total wait time (RSV)	
90% CI	L	U	L	U	L	U	L	U	L	U
50	98.685	99.478	99.274	99.890	27.711	29.288	66.460	67.200	6780.3	6888.6
100	98.653	99.306	99.390	99.776	28.050	29.067	66.640	67.170	6798.6	6868.1
250	99.183	99.459	99.463	99.735	27.882	28.393	67.064	67.436	6853.3	6897.3
500	99.136	99.445	99.528	99.690	28.060	28.596	67.044	67.346	6860.8	6892.3
1000	99.217	99.402	99.541	99.661	28.186	28.513	67.160	67.340	6865.9	6884.9

Note: L = Lower; U = Upper.

Table 7. 95% CI bootstrap estimates

	RSV availability		SPH availability		Average lead time		Number of resupplies		Total wait time (RSV)	
95% CI	L	U	L	U	L	U	L	U	L	U
50	98.546	99.508	99.173	99.921	27.469	30.283	66.180	67.200	6730.2	6894.3
100	98.530	99.326	99.205	99.783	27.946	29.182	66.520	67.390	6796.2	6897.5
250	99.147	99.516	99.438	99.737	27.787	28.401	67.040	67.456	6843.5	6901.0
500	99.058	99.457	99.518	99.702	28.022	28.599	67.022	67.394	6855.4	6896.6
1000	99.214	99.405	99.533	99.661	28.125	28.601	67.101	67.374	6854.6	6885.8

Note: L = Lower; U = Upper.

The simulation analyst may use either or both sets of tables to complete the power analysis. It is recommended that at least 500 independent trials are run in order to generate reliable interval estimators of the mean. In general, scheduling more than 2500 runs offers significantly diminished returns in terms of increased statistical power.

7. SENSITIVITY ANALYSIS: AN EXAMPLE

Setting up the Crusader Model for sensitivity analysis is fairly simple. All required modifications to the model can be performed in the Start (1) task. As explained in Sect. 6.4, the user may schedule more than one run in the *Settings...* menu in order to generate reliable estimators of critical performance measures.

An example is presented to illustrate the use of the Crusader model to perform sensitivity studies. An important parameter in the simulation model is the MTBF for vehicle systems and subsystems. Special multiplier variables have been created to allow the user to simultaneously modify all MTBF variables for the RSV and SPH. First, double-click on the Start node to pop up the dialogue box. We specify a test sequence for the variable *ffm* to include values 1.0, 0.75, 0.5, 0.25, and 0.1 respectively in order to progressively decrease the MTBF for the RSV. Similarly, the variable *afm* is modified to concurrently decrease the MTBF for the SPH. Joint manipulation of the two multiplier variables has the effect of uniformly decreasing the MTBF for the entire Crusader system. The multiplier value equal to 1.0 is equivalent to the baseline MTBF; 0.5 generates values one-half the baseline MTBF value. In this example 500 runs were executed for each level of MTBF in the test sequence, for a total of 2500 runs. The values of *ffm* and *afm* are first edited to equal 1.0, 500 runs are executed, and the variables *ffm* and *afm* are next set to 0.75 before running another 500 trials. This procedure is repeated for each of the five values in the test sequence.

Figure 1 shows mean availability as an increasing function of MTBF. Error bars are shown to indicate 95% confidence limits for the population mean. The parametric curves depict a "knee" or precipitous drop in availability at about 25% times the baseline value of the mean. At 75% of the baseline value of MTBF the decrease in availability is not substantial.

The average lead time for resupply declined steadily as MTBF increased, as shown in Fig. 2. Again, the region about 25% of the baseline seems to be most sensitive to changes in MTBF. The effect of increasing MTBF by an order of magnitude seems to be an increase in average lead time from 28 to 39 min. Figure 3 shows that the average number of times the SPH is without rounds similarly decreased as MTBF was increased. The shape of the curves in Figs. 2 and 3 are quite comparable, although the variability demonstrated by the error bars is somewhat smaller in the curve for average lead time.

Finally, Fig. 4 reveals that the average mission count increased with MTBF. Like the other parametric curves, the region of greatest sensitivity was found at about 25% of baseline, and performance at 75% of baseline did not differ significantly from baseline performance. The results for all five performance measures are consistent with the expectation that increasing the interval between failures allows the entire system to perform more efficiently and according to design specifications. It is probably not prudent to place too much trust in the absolute values produced by the simulation model for each of the performance measures. However, the general shape of the curves is a more robust feature of sensitivity analysis, and it is this aspect of the analysis that should be given the most attention and credibility.

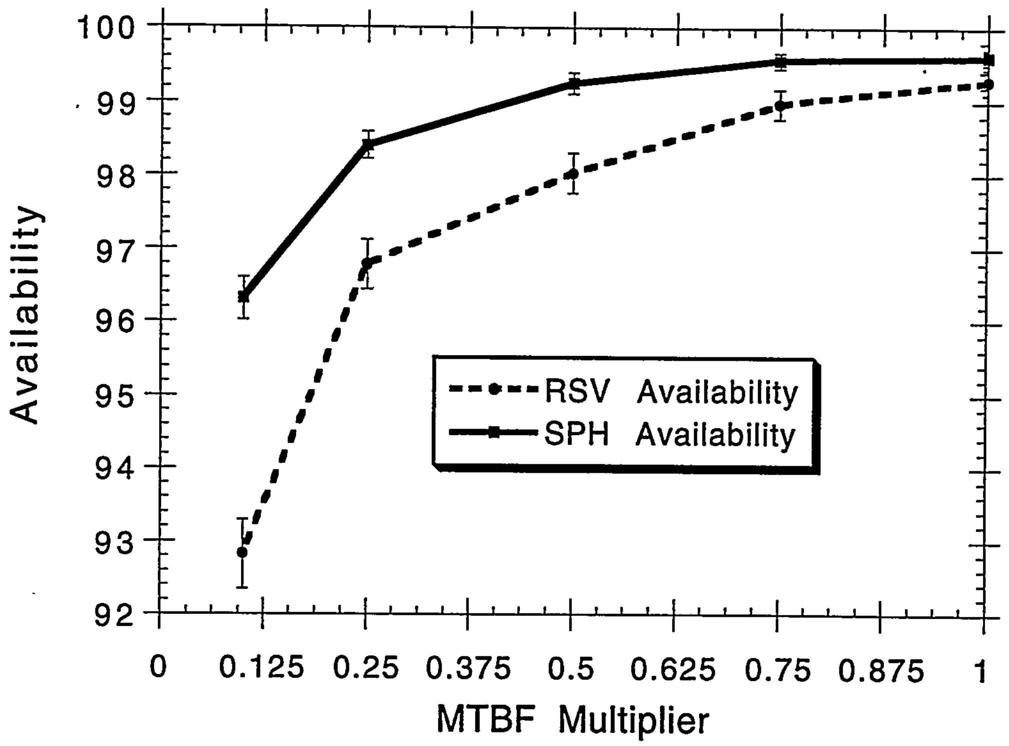

Fig. 1. Sensitivity of availability to MTBF.

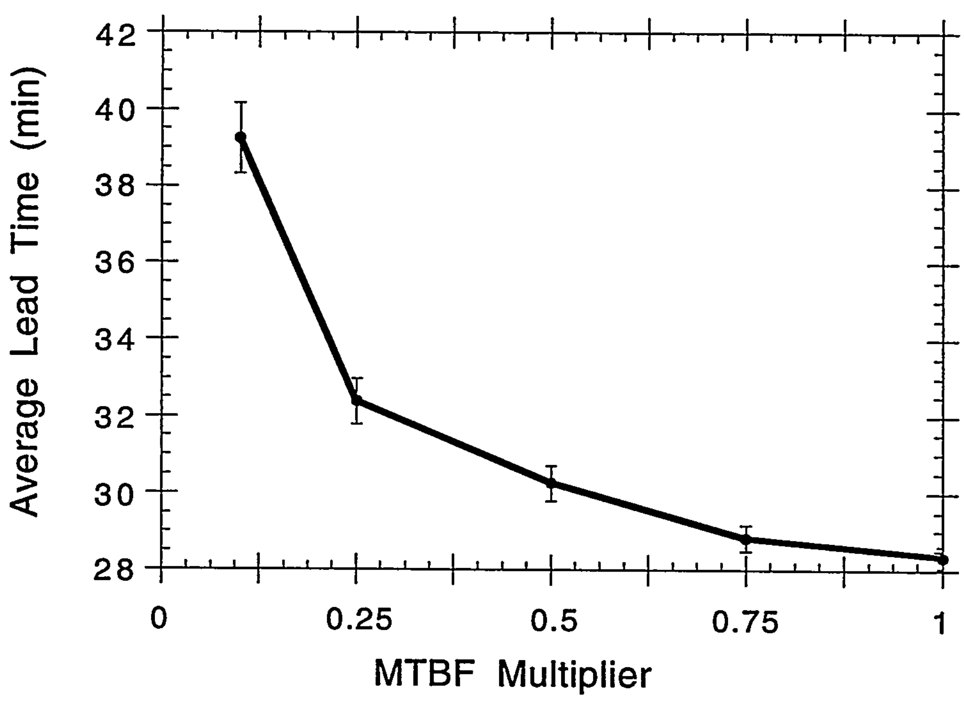

Fig. 2. Sensitivity of lead time to MTBF.

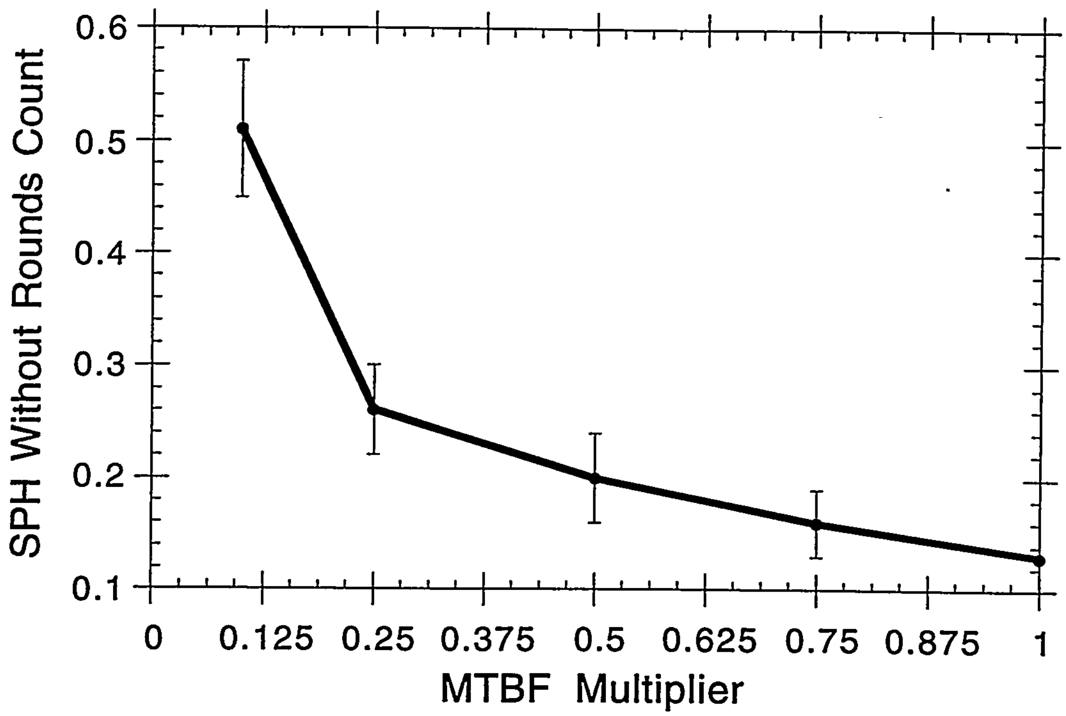

Fig. 3. Sensitivity of rounds count to MTBF.

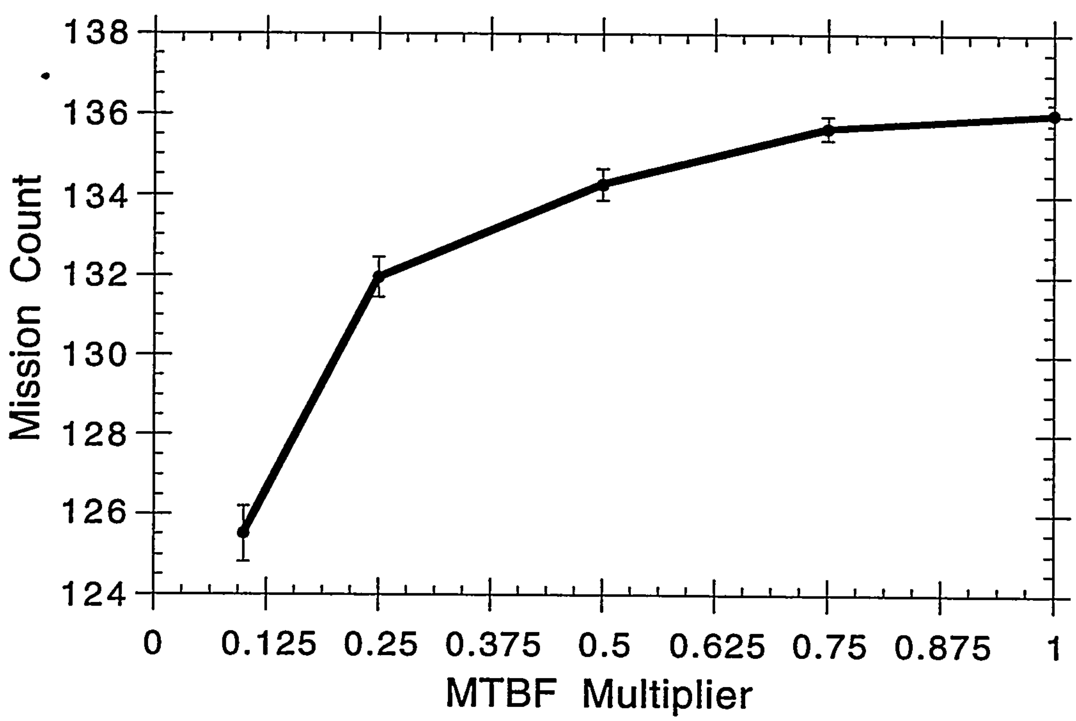

Fig. 4. Sensitivity of mission count to MTBF.

8. SUMMARY AND CONCLUSION

A model demonstrating the application of discrete-event simulation to RAM calculations and system performance analysis has been developed in support of SPH/RSV systems design. The simulation model runs on Windows and Macintosh platforms under the MicroSaint simulation package. Actual operations of an SPH/RSV pair in the field can be simulated under a variety of initial conditions. Reporting capabilities include statistics for availability and other measures of system performance. User instructions are provided to allow the analyst to modify initial conditions, execute the model, and interpret the findings.

Several tables have been developed as decision aids in the determination of the required number of replications to ensure convergence. The values in these tables suggest that very reliable estimators can be developed within a reasonable amount of clock time if the calculations are performed on a capable desk-top computer such as a Pentium or 80486-based processor. In general, from 500 to 2500 replications should be adequate.

The results of an MTBF sensitivity study offer conceptual validation for the simulation model by showing that availability increases at a slower-than-linear rate with MTBF. The capability to reveal qualitative trends and functional relationships among independent variables and measures of system performance is perhaps more informative to the design process than generation of quantitative estimators. For example, the shapes of availability curves generated by the model suggest that MTBF could be decreased by as much as a factor of four without severely impacting system availability.

Although the RAM simulation model is complete, several programming extensions could be implemented to improve usability and value as a tool for RAM calculations and system performance analysis. For example, a more general representation of the operations of multiple SPH/RSV interactions is a natural extension. A multiple SPH/RSV model could be used to determine the optimum number of RSVs needed to service a fixed number of SPH vehicles. Another area of potential enhancement is the repair process, which is currently conceptualized primarily as a "change-out" activity. This simplistic treatment of repair activity effectively places an upper bound on the availability estimators. Finally, if the assumption that all failures are "system abort" were relaxed, it would be possible to model the benefits of diagnostics/prognostics and preventive system maintenance.

REFERENCES

1. G. P. Chubb, K. R. Laughery, and A. A. B. Pritsker, "Simulating Manned Systems," pp. 1298–1327 in *Handbook of Human Factors*, ed. G. Salvendy, Wiley, New York, 1987.

2. W. L. Hays, *Statistics for the Social Sciences*, Holt, Rinehart and Winston, New York, 1973.

3. B. Efron and R. Tibshirani "Statistical Data Analysis in the Computer Age," *Science* **253** (July), 390–95 (1991).

4. L. W. Friedman and H. H. Friedman, "Analyzing Simulation Output Using the Bootstrap Method," *Simulation* **64**(2), 95–100 (1995).

5. *MicroSaint User's Guide*, Micro Analysis and Design Simulation Software, Inc., Boulder, Colorado, 1990.

APPENDIX
SIMULATION NETWORK DIAGRAMS

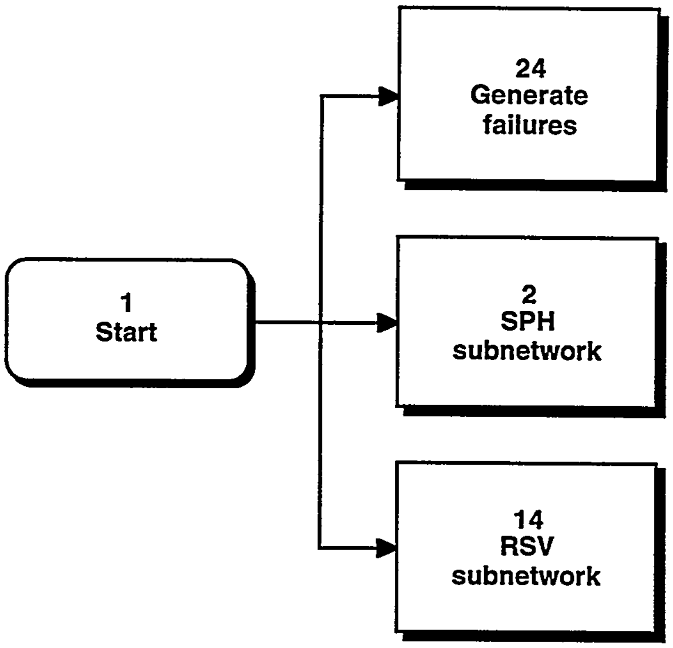

Fig. A.1. Top-level network.

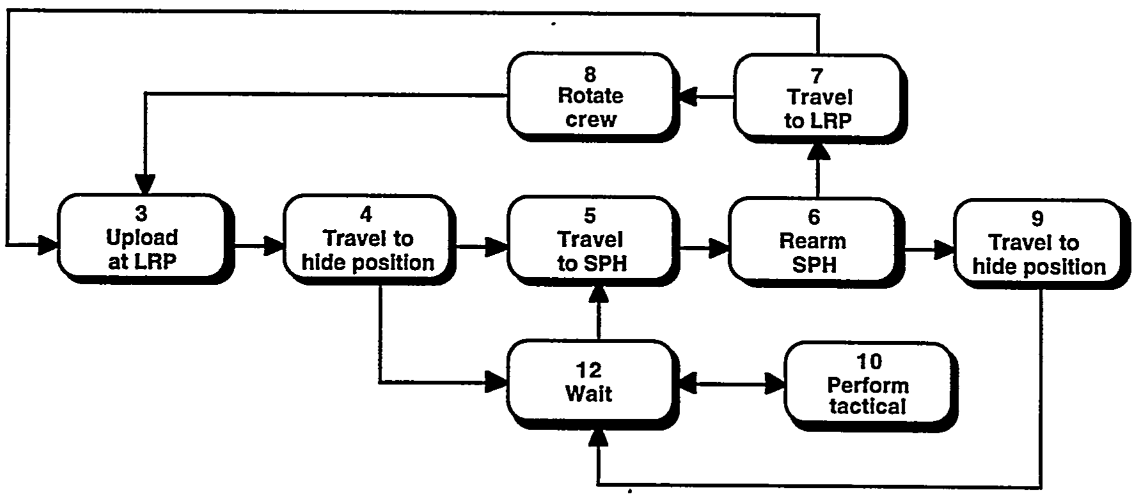

Fig. A.2. RSV subnetwork.

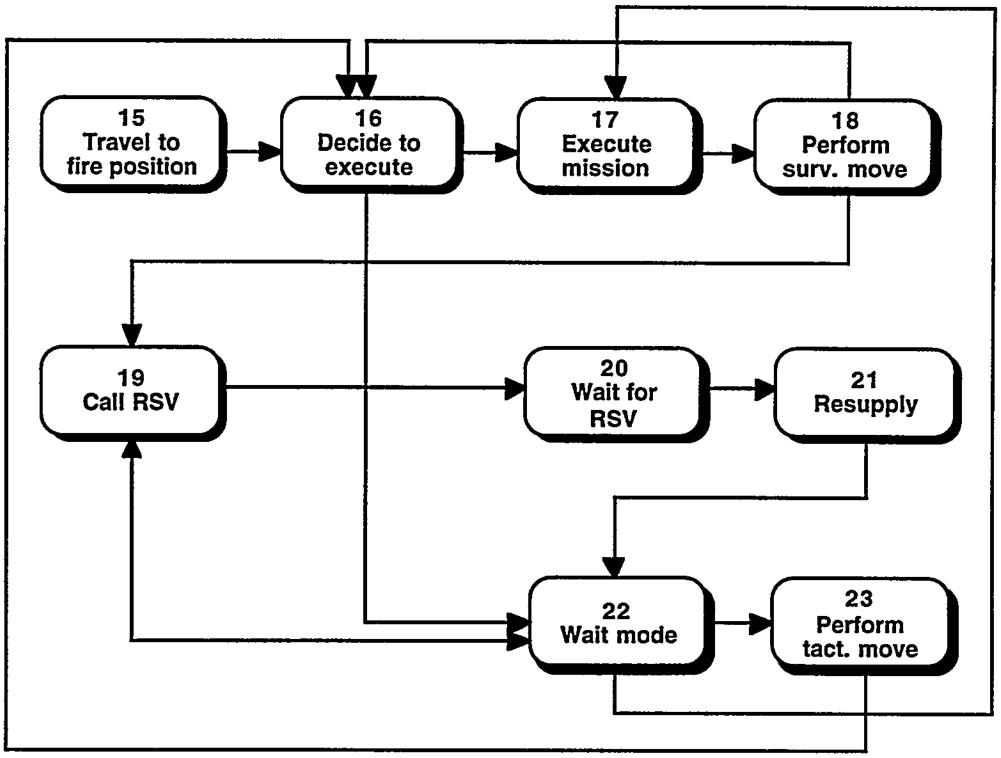

Fig. A.3. SPH subnetwork.

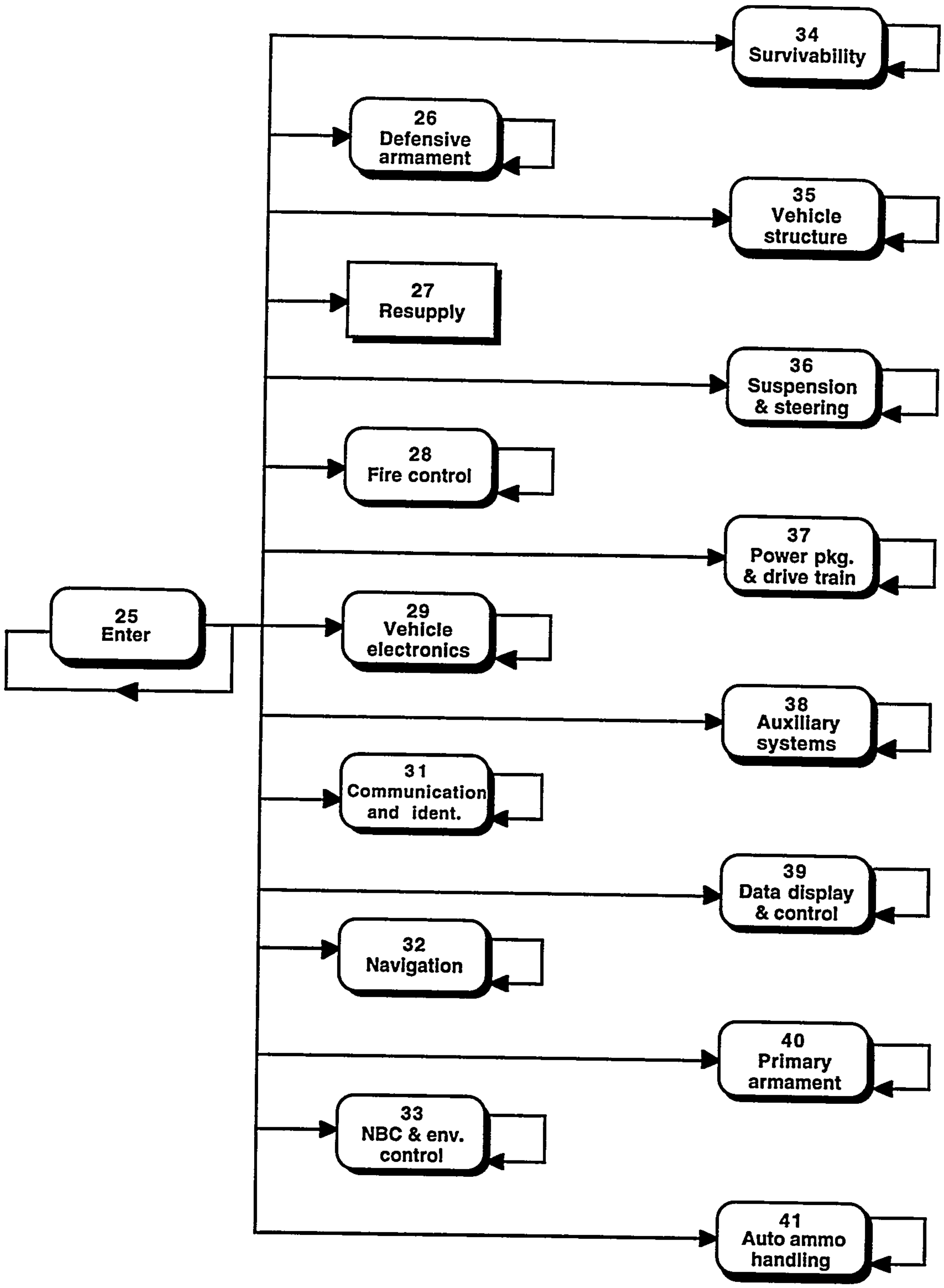

Fig. A.4. General system failures.

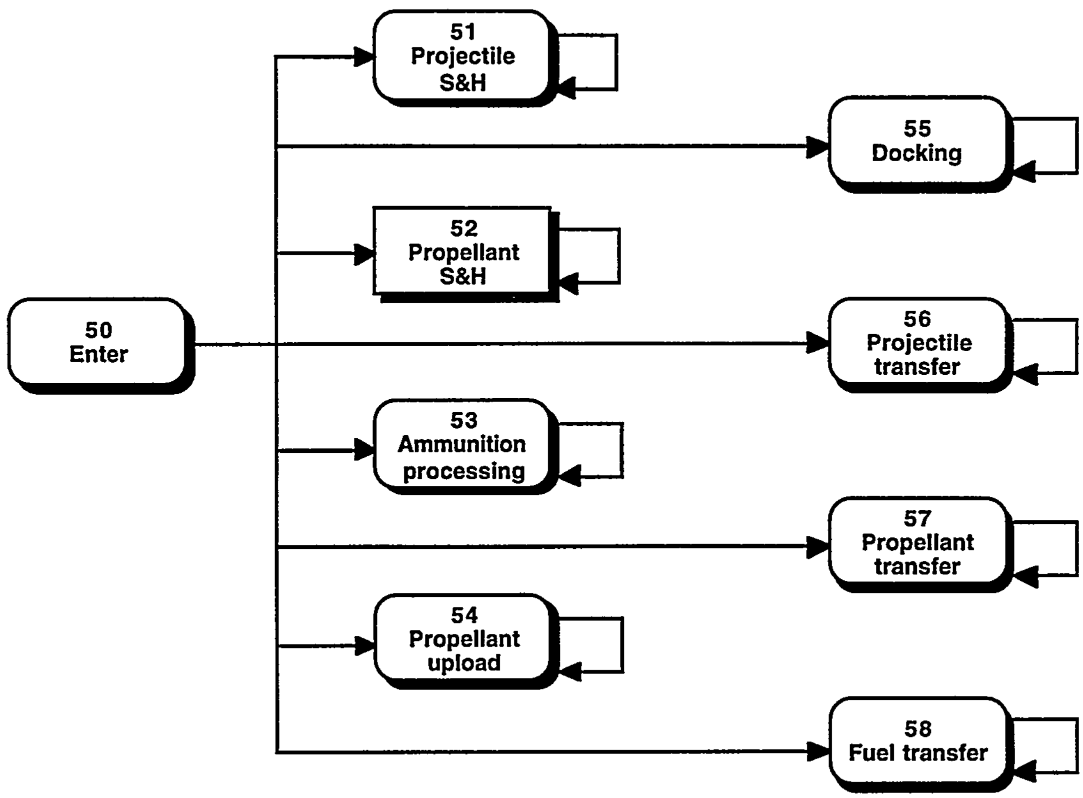

Fig. A.5. RSV resupply subsystem failures.

INTERNAL DISTRIBUTION

1. J. H. Clift
2. D. C. Cunningham
3. R. M. Davis
4. M. J. Haire
5. D. C. Haley
6. W. R. Hamel
7. J. H. Hannah
8-10. J. N. Herndon
11. R. F. Holdaway
12. H. E. Knee
13. C. T. Kring
14. L. M. Kyker
15. R. C. Mann
16-18. S. C. Nelson
19. C. E. Oliver
20. K. E. Plummer
21-23. A. H. Primm
24. S. L. Schrock
25-27. J. C. Schryver
28-29. Laboratory Records
30. Laboratory Records (ORNL-RC)
31. RPSD Publications Office
32. ORNL Patent Section
33. Central Research Library
34. Document Reference Section

EXTERNAL DISTRIBUTION

35. C. M. Bloomhardt, Lockheed Martin Armament Systems, Lakeside Avenue, Burlington, Vermont 05401-4985.

36. T. M. Brown, U.S. Army, SIMCAR-FSS-DF, Attention Tom Brown, Building 322, Picatinny Arsenal, New Jersey 07806-5000.

37. M. Dolecki, Commander, USA TACOM, Attention: AMSTA-ZED (Mr. Matt Dolecki), Warren, Michigan 48397-5000.

38. J. Fedewitz, PM-AMMOLOG, HQ ARDEC, AMCPM-AL Picatinny Arsenal, New Jersey 07801-5001.

39. T. Fitzgerald, Project Manager, FARV, Attention: SFAE-ASM-FR, Picatinny Arsenal, New Jersey 07806-5000.

40. N. F. Gravenstede, PM-AMMOLOG, HQ ARDEC, AMCPM-AL Picatinny Arsenal, New Jersey 07801-5001.

41. G. L. Kent, PM-AMMOLOG, HQ ARDEC AMCPM-AL Picatinny Arsenal, New Jersey 07801-5001.

42. A. Simon, U.S. Army, SIMCAR-FSS-DF, Attention: Alex Simon, Building 3159, Picatinny Arsenal, New Jersey 07806-5000.

43. Office of Assistant Manager for Energy Research and Development, Oak Ridge Operations Office, Department of Energy, P.O. Box 2008, Oak Ridge, Tennessee 37831-6269.

44-45. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, Tennessee 37831.