

AD-A116 922

JAYCOR ALBUQUERQUE NM

F/6 15/7

USER'S GUIDE FOR THE SAS (STAND-OFF ATTACK SIMULATION) COMPUTER--ETC(U)

JAN 82 C M CLARK, J T HUMPHREY, L W KENNEDY DNA001-79-C-0463

DNA-5845T

NL

UNCLASSIFIED

1 of 2
4/4/2022

AD-E301009

(12)

DNA 5845T

AD A116922

USER'S GUIDE FOR THE SAS (STAND-OFF ATTACK SIMULATION) COMPUTER MODEL

Carl M. Clark
Joseph T. Humphrey
Dr. Lynn W. Kennedy
Stephen G. Reynolds
JAYCOR
8001 Mountain Road Place, N.E.
Albuquerque, New Mexico 87110

15 January 1982

Topical Report for Period 15 January 1981—15 January 1982

CONTRACT No. DNA 001-79-C-0463

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B382081466 A99QAXFD00001 H2590D.

DTIC FILE COPY

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, D. C. 20305

DTIC
ELECTE
JUL 13 1982
S B D

82 06 11 028

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA 5845T	2. GOVT ACCESSION NO. AD-A116 922	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) USER'S GUIDE FOR THE SAS (STAND-OFF ATTACK SIMULATION) COMPUTER MODEL		5. TYPE OF REPORT & PERIOD COVERED Topical Report for Period 15 Jan 81—15 Jan 82
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Carl M. Clark Lynn W. Kennedy Joseph T. Humphrey Stephen G. Reynolds		8. CONTRACT OR GRANT NUMBER(s) DNA 001-79-C-0463
9. PERFORMING ORGANIZATION NAME AND ADDRESS JAYCOR 8001 Mountain Road Place, NE Albuquerque, New Mexico 87110		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Subtask A99QAXFD000-01
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305		12. REPORT DATE 15 January 1982
		13. NUMBER OF PAGES 168
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This work sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B382081466 A99QAXFD00001 H2590D.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Stand-Off Attack Computer Simulation Modeling Force-on-Force Engagement Security Analysis Survivability Analysis		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This guide describes the use and application of the SAS (Stand-Off Attack Simula- tion) computer model. SAS is an effective survivability and security system design tool which allows an analyst to compare the relative effectiveness of selected survivability and/or security system upgrades to a threat involving the use of stand-off direct or indirect weapon fire. The program evaluates damage to both personnel and material targets. Scenarios which require either fixed (static) or mobile (dynamic) situations, or both may be modeled.		

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
INTRODUCTION	7
1 ANALYSIS METHODOLOGY	10
1.1 BACKGROUND	10
1.2 MODELING TECHNIQUE	10
1.3 AREAS OF SPECIFIC APPLICATION	19
1.4 ADVANTAGES OF THE SAS MODEL	20
1.5 DISADVANTAGES OF THE SAS MODEL	21
1.6 UNIQUE FEATURES OF THE SAS MODEL	21
1.7 AREAS FOR FUTURE DEVELOPMENT	23
2 INPUT DATA DESCRIPTION	26
2.1 TARGET DATA DESCRIPTION	27
2.2 FIRER DESCRIPTION	31
2.3 AIM POINT DATA DESCRIPTION	31
2.4 TARGET STATUS CHANGE DATA	33
2.5 OTHER INPUT DATA	34
3 EXAMPLE APPLICATION	35
3.1 SCENARIO OVERVIEW	35
3.2 SCENARIO DETAILS	37
3.3 INPUT DATA SUMMARY	62
3.4 SAS OUTPUT	62
4 EXAMPLE TERMINAL SESSION	69
REFERENCES	76
 <u>Appendix</u>	
A STANDARDIZED INPUT DATA FORMS	77
B GENERAL MODEL INFORMATION	101
B.1 SAS MODEL FLOW STRUCTURE	101
B.2 STORAGE REQUIREMENTS	102

TABLE OF CONTENTS (Continued)

<u>Appendix</u>	<u>Page</u>
B.3 SYSTEM CONFIGURATION	102
B.4 SOURCE LANGUAGE	102
C DOCUMENTED SAS CODE LISTING	104
C.1 INCLUDE FILE	104
C.2 DOCUMENTED SAS CODE LISTING	104
D GENERATION OF SHOT HIT POINT	159

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



LIST OF ILLUSTRATIONS

	PAGE
FIGURE 1.1 - Parallel Plane Concept Used to Describe the Size and Position of Opposing Force Personnel and Material Targets.....	11
FIGURE 1.2 - Monte-Carlo Techniques Used to Evaluate Probability of Hit.....	12
FIGURE 3.1 - "Bird's Eye" View of Ambush Scenario Just Prior to Mine Detonation.....	36
FIGURE 3.2 - Defender Target Locations, Truck 1, Position 0.	40
FIGURE 3.3 - Defender Target Locations, Truck 2, Position 0.	41
FIGURE 3.4 - Defender Target Locations, Truck 3, Position 0.	42
FIGURE 3.5 - Defender Target Locations, Truck 4, Position 0.	43
FIGURE 3.6 - Defender Target Locations, Truck 1, Position 1.	44
FIGURE 3.7 - Defender Target Locations, Truck 2, Position 1.	45
FIGURE 3.8 - Defender Target Locations, Truck 3, Position 1.	46
FIGURE 3.9 - Defender Target Locations, Truck 4, Position 1.	47
FIGURE 3.10 - Defender Target Locations, Truck 1, Position 2	48
FIGURE 3.11 - Defender Target Locations, Truck 2, Position 2	49
FIGURE 3.12 - Defender Target Locations, Truck 3, Position 2	50
FIGURE 3.13 - Defender Target Locations, Truck 4, Position 2	51
FIGURE 3.14 - Attacker Target Locations.....	53
FIGURE 3.15 - Truck Locations Versus Elapsed Time After Mine Detonation.....	56
FIGURE 3.16 - SAS Output Showing Probability Targets Exist with Time.....	63
FIGURE 3.17a - SAS Output Showing Expected Number of Targets Destroyed with Time.....	65
FIGURE 3.17b - SAS Output Showing Expected Number of Targets Destroyed with Time.....	66
FIGURE 3.18a - Plot of Expected Number of Targets Destroyed Versus Time From Beginning of Simulation.....	67

FIGURE 3.18b - Plot of Expected Number of Targets Destroyed Versus Time From Beginning of Simulation.....	68
FIGURE A.1a - Data Forms Used to Acquire SAS Input Data (Input Data Form, Page 1).....	78
FIGURE A.1b - Data Forms Used to Acquire SAS Input Data (Target Data Form).....	79
FIGURE A.1c - Data Forms Used to Acquire SAS Input Data (Aim Point Data Form).....	80
FIGURE A.1d - Data Forms Used to Acquire SAS Input Data (Input Data Form, Page 2).....	81
FIGURE A.1e - Data Forms Used to Acquire SAS Input Data (Input Data Form, Page 3).....	82
FIGURE A.2a - Completed Input Data Form, Page 1.....	83
FIGURE A.2b - Completed Target Data Form, Targets 1-13.....	84
FIGURE A.2c - Completed Target Data Form, Targets 14-26.....	85
FIGURE A.2d - Completed Target Data Form, Targets 27-39.....	86
FIGURE A.2e - Completed Target Data Form, Targets 40-52.....	87
FIGURE A.2f - Completed Target Data Form, Targets 53-65.....	88
FIGURE A.2g - Completed Target Data Form, Targets 66-78.....	89
FIGURE A.2h - Completed Target Data Form, Targets 79-91.....	90
FIGURE A.2i - Completed Target Data Form, Target 92.....	91
FIGURE A.2j - Completed Aim Point Data Form, Aim Points 1-13.....	92
FIGURE A.2k - Completed Aim Point Data Form, Aim Points 14-26.....	93
FIGURE A.2l - Completed Aim Point Data Form, Aim Points 27-39.....	94
FIGURE A.2m - Completed Aim Point Data Form, Aim Points 40-44.....	95
FIGURE A.2n - Completed Input Data Form, Page 2, Targets 1-19.....	96
FIGURE A.2o - Completed Input Data Form, Page 2, Targets 20-38.....	97
FIGURE A.2p - Completed Input Data Form, Page 2, Targets 39-46.....	98

FIGURE A.2q - Completed Input Data Form, Page 3.....	99
FIGURE B.1 - SAS Routine Flow Structure.....	103
FIGURE D.1 - Illustration of Aim Error.....	160
FIGURE D.2 - Hit Point Expressed in Polar Coordinates.....	161

LIST OF TABLES

	PAGE
TABLE 3.1 Weapon Assumptions.....	61

INTRODUCTION

The SAS computer model was developed by JAYCOR to assist in the evaluation of survivability and security of weapon movement within the TNF (Theater Nuclear Forces). In performing this work, it was quickly discovered that many scenarios involved both security and survivability considerations. The interplay between security and survivability was often so tightly knit that analysis of the two factors independently would not provide a realistic solution. For example, consider the case of a ground movement scenario which involves the attack on a ground convoy by a land based special operations team (SOT). The objective of the SOT is to destroy or inflict sufficient damage to the weapons being transferred as to render them inoperative.

Survivability models have been developed which will evaluate the damage delivered to these weapons from sustained direct or indirect fire. Likewise, security models will enable the analyst to evaluate the number of defense and attack personnel incapacitated during engagement. Unfortunately, in our investigation of available computer models used to solve problems of this nature, we were unable to locate a model which would adequately combine the aspects of security and survivability in a realistic manner.

In general, security models simulated force-on-force engagement encounters, concentrating on attack force personnel firing at defense personnel and vice versa. We were unable to find a security model which could evaluate personnel firing at a combination of "material" type targets and "personnel" type targets.

Survivability models which were investigated permitted analysis of both material and personnel type targets; however, they did not allow for modification in defense force or attack force strategy. Realizing these deficiencies, an attempt was made to develop a methodology which would integrate necessary components of survivability and security, providing damage analysis for both material and personnel targets, as well as allowing the analyst to modify defense and attack force strategy. The methodology to

combine both key survivability and security concerns has been developed and implemented into a computer model called SAS (Stand-Off Attack Simulation).

The objective of this report is to familiarize the reader with the methodology, use, and potential application of the SAS computer model. SAS analysis methodology is described in Section 1. This section also includes a description of specific areas of model application, a discussion of the advantages and disadvantages of the modeling technique used, and a list of possible improvements to the model which would simplify data input, increase simulation detail, and expand the areas of possible model application.

A summary of input required to execute the SAS model is presented in Section 2.

Section 3 describes an example scenario in detail. This scenario will be used to illustrate a possible application of the SAS model and step the user through an example execution. All input data is described in the text and examples of simulation output is included. The usefulness of SAS output data is best illustrated through an example such as that provided in Section 3. For this reason, this section also contains a description of the type of output data generated.

Section 4 contains a terminal session description for the example scenario outlined in Section 3. This section steps the user through the SAS execution of the example scenario, illustrating the interactive questions used to obtain input data and the format used to enter input data in response to each question type.

Appendix A contains sample standardized input data forms which are used to describe the details unique to a particular analysis. Following the blank data forms are completed forms showing all data required to perform execution of the example scenario described in Section 3.

Appendix B provides a description of the SAS model flow structure. Also included are descriptions of the expected storage requirements, operating

system, source language used, and description of the computer system on which the code is currently installed.

Appendix C contains a fully documented code listing of the SAS model, including all subroutines used (excluding system furnished routines) and the INCLUDE file. The INCLUDE file contains parameter values which define the size of arrays and constants used within the SAS model.

Appendix D contains additional mathematical detail not provided in Section 1.

SECTION 1

ANALYSIS METHODOLOGY

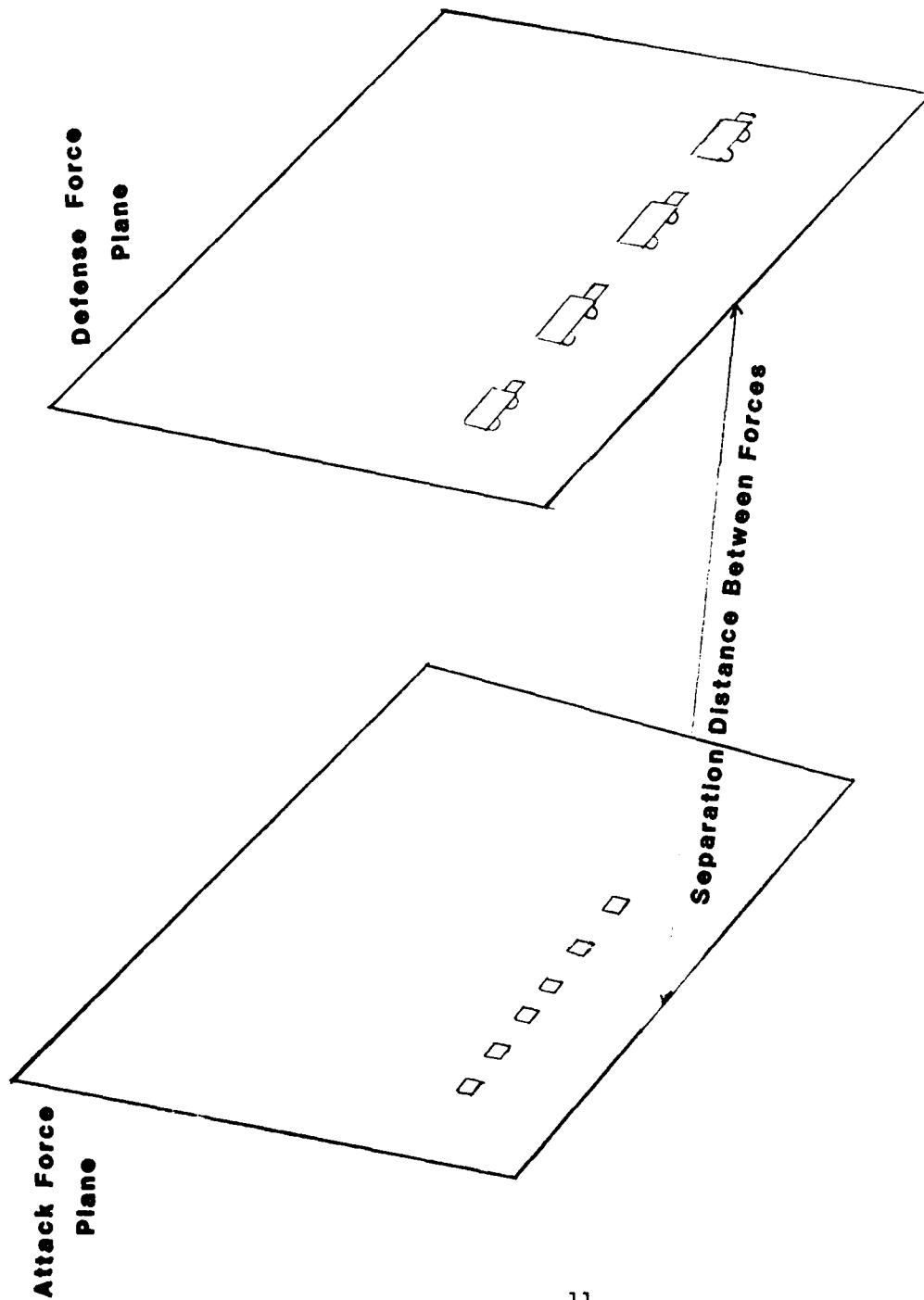
1.1 BACKGROUND

SAS is a two-dimensional simulation engagement model. Engagements involve the analysis of opposing force personnel and may include damage to selected material targets if desired. Attack force personnel and material targets are assumed to lie in a geometric plane parallel to the plane containing defense personnel and material targets (Figure 1.1). Movement of personnel and material targets is permitted during the simulation provided all movement is in either a horizontal or vertical direction within the respective attack and defense force plane. The sizes of personnel and material targets are governed by the vulnerable cross-sectional area of the target as viewed from the opposing plane. Targets, either of personnel or material type, are described in terms of rectangular shapes. Target position is specified relative to a fixed origin by x and y coordinate values corresponding to the lower left-hand corner of the rectangular target.

Attack or defense force fire is directed toward aim points located in the opposing force reference plane. Aim points are generally located in the center of an opposing target; however, they need not be directly related to any target nor appear within any opposing target area. This flexibility permits SAS to simulate "scattered" weapon fire from either attack or defense force personnel.

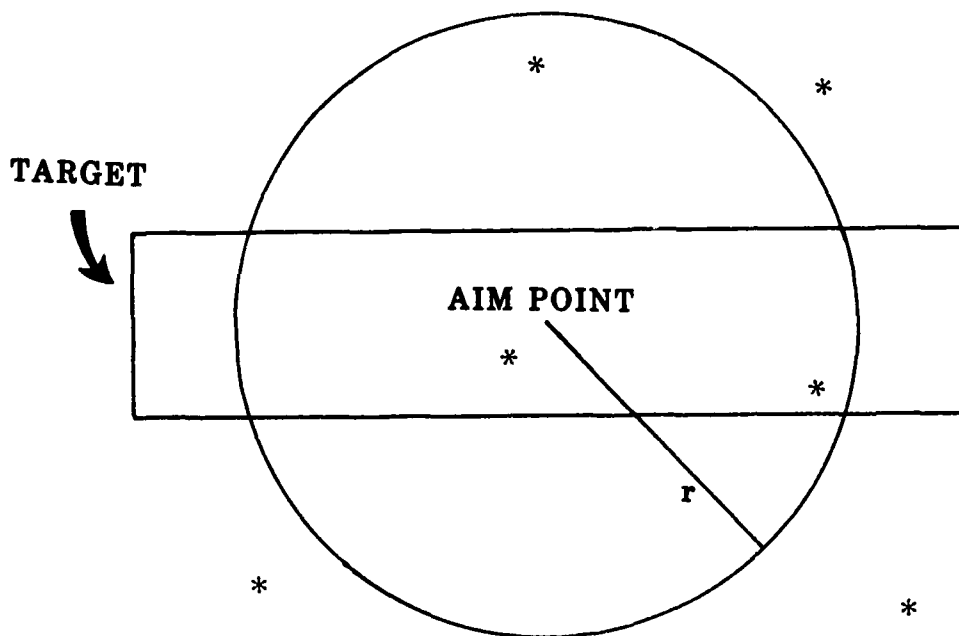
1.2 MODELING TECHNIQUE

The model uses Monte Carlo sampling techniques to determine the probability a given target is hit by opposing weapons fire. To obtain the probability that a player, firing at a particular aim point, may hit a particular target, a given number of shots are fired by the player at the aim point. The number of times the target is hit is divided by the total number of shots fired (refer to Figure 1.2). Once probabilities of target hit have



Personnel and Material Target Size is Represented as the Cross-sectional Area Which is Vulnerable to Opposing Weapon fire.

Figure 1.1 Parallel Plane Concept Used to Describe Size and Position of Opposing Force Personnel and Material Targets.



$$P(\text{Target is hit}) = \frac{\sum (\text{Number of shots hitting target})}{\text{Total number of shots fired}}$$

$$= \frac{2}{6} = .33, \text{ for this example.}$$

where,

r - Defines the radius of a circle around the aim point within which one-half of all shots fired are expected to land .

***** - Denotes a sample hit point selected at random.

Figure 1.2 Monte-Carlo Techniques Used to Evaluate Probability of Hit

3
been determined for all possible players, aim points, and targets, these hit probabilities are used to modify the probability that a particular target will still exist with time. An event scheduling array is maintained which allows SAS to identify the next event to occur. An "event" can be any one of the following:

- a player may be selected to fire;
- a player may select a new weapon to fire, or the firing rate of the player's weapon may be modified;
- an aiming delay may be specified before a particular player may fire his next round;
- a player or target may stop or start moving in a different direction;
- the strategy of a particular player may change (i.e., a modified set of aim points may be selected for a particular player);
- the cover provided a target may change.

Strategy for attackers or defenders is defined by assigning a set of aim points, which are ordered by priority for each attacker or defender involved in the simulation. The probability that an attacker will continue to fire at a specific aim point is proportional to the probability that the primary target associated with that aim point is not destroyed.

SAS begins by computing probabilities of target hit by use of Monte Carlo techniques for each possible combination of firer, aim point, and target. These probabilities are stored in a three-dimensional array, denoted by $P_{NF,NAM,NT}$, where NF is the maximum number of firers, NAM is the number of possible aim points, and NT is the number of possible targets. $P_{i,j,k}$ contains the probability firer i fires at aim point j and hits target k.

In order to compute each $P_{i,j,k}$, a specified number of shots are fired at

aim point j. Each shot fired is assumed to follow a Rayleigh distribution with an aiming error defining the firer's accuracy. The Rayleigh distribution function is given by:

$$P_I = \int_0^{\infty} \frac{r}{\sigma^2} e^{(-r^2/2\sigma^2)} dr$$

where r is the distance from the aim point and P_I is the probability that the shot fired lands within a circle of radius r centered around the aim point.

For purposes of the SAS model, aiming error is defined as the radius of a circle in which one half of the shots fired are expected to land. The "hit point" of each shot within the opposing force reference plane is expressed by x and y coordinates relative to the reference plane origin. To generate x and y coordinates from the Rayleigh distribution, uniform random variables are generated. These uniform random variables are then transformed to normal random variables having mean $\mu=0$ and variance $\sigma=.8493r$. The variance σ , expressed as a function of r, is used to insure that the expected probability of a shot landing within r distance of the aim point is 0.5. Refer to Appendix D for the detailed description of how random variables are generated from the Rayleigh distribution.

Factors which effect aiming error are divided into four categories: mechanical, environmental, human, and "other". Mechanical factors would include the weapon bench CEP, amount of weapon recoil, anomalies in the symmetrical weight of the round fired and the trueness of the bore of the weapon barrel, for example. Wind (which could effect projectile flight), the degree of darkness (i.e., daylight, dusk, twilight, etc.), fog, terrain, or smoke (which could reduce the probability that the firer selects the proper aim point at which to fire) would be considered as environmental factors. Human factors would include the amount of training the firer has had, the recency of training, and the effects of stress on the individual.

Other factors which affect aiming error include:

- the amount of time available to the firer to aim between rounds fired,

- the degree of which either the firer or the target is moving,
- possible target camouflage,
- distance between firer and target,
- firer's weapon type,
- maximum effective range of weapon.

SAS uses deterministic techniques to combine the target hit probabilities generated by Monte Carlo methods into the probability a certain target exists after N seconds have elapsed from the start of the simulation. The probability that a given set of targets exists with time can be combined to yield the expected number of targets destroyed with time.

SAS maintains a one-dimensional array which contains the probability that each target involved in the simulation exists with time. The probability of existence for a given target is modified whenever a member of the opposing force is selected to fire and may either hit the target directly or inflict damage to the target collaterally. If a firer is selected to fire and may hit the target directly or indirectly, the probability that the target exists after the firer fires his first shot is modified by the probability that the target is not destroyed by the firing event. Let $P_{TE}^t = P(\text{the target exists before firing event occurs})$. For the direct hit situation, P_{TE}^{t+1} is computed to reflect the probability that the target still exists after the firing event occurs as follows:

$$P_{TE}^{t+1} = P_{TE}^t (1 - P_{AS} P_{FE} P_{TH})$$

where:

P_{TE}^{t+1} = Probability target exists after shot is fired.

P_{TE}^t = Probability target exists before shot is fired.

P_{AS} = Probability that this particular aim point is selected.

P_{FE} = Probability that the firer still exists to fire (i.e., the probability that this firer has not been destroyed previously).

P_{TH} = Probability that the target is hit given the firer fires at the particular aim point.

Recall that the probability a given target (i) is hit, assuming a particular firer (j) fires at a particular aim point (k), has been computed previously and is stored in $P_{i,j,k}$. P_{TH} may be replaced by $P_{i,j,k}$ when referring to a specific target, firer, and aim point.

The variables enclosed in parentheses, $1 - P_{AS}P_{FE}P_{TH}$, represent the probability that the target is not destroyed by this shot. Thus, the probability the target exists after the shot is fired is equal to the probability it exists before the shot is fired multiplied by the probability that the target is not destroyed by this shot.

Whenever a firer is selected to fire, he may have more than one aim point at which to fire. Assume that N contains the number of aim points that the firer may select. Then the probability that target i exists after firer k fires his shot is expressed by:

$$P_i^{t+1} = P_i^t \sum_{k=1}^N (1 - P_k P_j P_m)$$

where:

$$P_i^{t+1} = P(\text{target i exists after firer j fires the shot})$$

$$P_i^t = P(\text{target i exists before firer j fires the shot})$$

$$P_k = P(\text{firer j selects aim point k to fire at})$$

$$P_j = P(\text{firer } j \text{ exists to fire})$$

$$P_m = P(\text{target } i \text{ is hit if firer } j \text{ fires at aim point } k)$$

This expression can be further reduced to:

$$P_i^{t+1} = NP_i^t - P_i^t P_j \sum_{k=1}^N P_k P_m$$

To analyze the effect of indirect target damage, the probability that the target exists may be expressed as:

$$P_i^{t+1} = \sum_{n=1}^M P_n^t (1 - P_j \sum_{k=1}^N P_k P_m)$$

where:

P_i^{t+1} , P_j , P_m , P_k , and N have been described previously,

M = Number of targets which if hit would result in collateral damage to target i ,

P_n^t = P (that target n exists before firer j fires next shot), where target n is a target which if hit causes collateral damage to target i ,

P_m = P (that firer j fires at aim point k and hits target n), and

N = Number of aim points that firer j may fire at and hit target n .

Probabilities of target existence are modified as the simulation progresses by both the direct hit and indirect hit expressions. After the simulation time reaches the maximum time desired by the analyst, the simulation model stops and lists the probability of target existence for each target involved in the simulation. The analyst may also list the probability

of target existence at intermediate times during the simulation.

After probabilities of target existence have been developed, targets possessing similar characteristics (such as defense or attack force personnel) can be grouped together and the expected number of targets destroyed or damaged can be calculated over time. This allows the analyst to estimate the expected size of the remaining defense force or attack force at preselected points in time. Any targets, such as number of vehicles destroyed, number of (nuclear) weapons damaged, etc., may be grouped in this manner with their expected number remaining with time calculated.

Standard conditional probability techniques are used to determine the expected number of grouped targets that are destroyed at a particular point since the beginning of the simulation. Assume that the array P_i for $i = 1, N$ contains the probability that attack force personnel 1 through N exist after 10 seconds have elapsed in the simulation. From this data, it is possible to obtain the expected number of attack force personnel which have been killed. First, the probability that exactly r attack force personnel remain after 10 seconds is computed. For this example, r would range from $r = 0$ (no attack personnel remain) to $r = N$ (all personnel remain). Let $P_r = P(\text{exactly } r$

attack personnel remain). $P_{r=0} = \prod_{i=1}^N (1 - P_i)$, which is the probability that

attacker 1 has been killed $\times$ probability that attacker 2 has been killed $\times$...probability that attacker N has been killed. The computer can rapidly evaluate all combinations required to generate P_r for each possible value of r .

After all P_r values have been obtained, the expected number of attack force personnel killed (denoted by E_k) is evaluated via the expression:

$$E_k = \sum_{i=1}^N i P_r$$

by definition of expected value.

For a more complete description of the methodology used by the SAS computer model, the user is requested to read the documentation provided within the computer code listing, Appendix C.

1.3 SPECIFIC AREAS OF APPLICATION

The SAS code is a generalized survivability model applicable to a wide variety of situations in which adversaries fire weapons at guards and at critical system components from prescribed distances. This includes scenarios involving ambush of a ground or air convoy, mortar or RPG attacks on a fixed site during outloading or maintenance, missile or bombing attacks on a secured fixed site, and attacks by any kind of stand-off weapon on unhardened field sites or dispersal locations. The model is specifically not applicable to attacks involving theft or sabotage of a covert nature. The computer model PANL (Path ANALysis, Reference 3) should be used for covert or stealth type attacks.

Specific applicability of the model to a large number of situations with currently deployed or under-development weapon systems is foreseen. These potential applications are discussed in a general way in the following paragraphs. Included in the discussion are Pershing, Lance, and the Ground Launched Cruise Missile (GLCM).

Pershing, because of the requirement for quick reaction capability, is vulnerable at QRA (Quick Reaction Alert) sites to conventional bombing or to attack by terrorists or special operations teams (SOT) using a variety of weapons. Mortars, rocket propelled grenades, missiles, automatic and semi-automatic weapons and small arms are a few of the weapons which terrorists or SOTs might use in attacking QRA sites from a standoff position. Trucks, tanks, helicopters or fixed-wing aircraft might be used in the attack.

SAS is applicable to the analysis of Pershing QRA sites when various types of survivability or security enhancements are being considered. Berms, revetments, and various types of soft and semi-soft shelters may be analyzed. Armored blankets and personnel shelters or armored fighting positions may be included in the analysis as well.

The attack modes described above may be mounted against other systems during uploading for dispersal or for non-emergency relocation. GLCM and LANCE must be mobilized and formed into convoys for dispersal, and smaller weapons, whether or not they are protected by armored storage containers, undergo a period of high vulnerability when they are removed from igloos and loaded onto trucks, helicopters, or aircraft. A well-planned standoff attack, timed to coincide with a dispersal or relocation operation, could constitute a significant threat during the period when efficient mobilization for war is imperative. Analysis of scenarios involving this kind of attack, with variation on mobilization and dispersal procedures and equipment, could provide significant improvement.

The standoff mode of attack is the one most likely to be used against Pershing, Lance, or GLCM while in convoy to deployed positions or while hiding in the woods. Such an attack requires less detailed planning by an aggressor than does a penetration attack, and there is less risk to him. He will maintain his position at a distance from the target, and thus will be able to escape more easily. This is true whether the attacking weapons are bombs, missiles, or long-range rifles, and whether the adversary mobility is by aircraft, trucks, or on foot.

It seems apparent that SAS could and should be used to analyze the vulnerability of real systems to a number of real threats. Improvement options now under development could be analyzed to determine whether they provide increased survivability under attack situations which are highly credible. We believe that SAS can provide a useful methodology in a realm which bridges the area between the security threat and considerations which have totally to do with survivability.

1.4 ADVANTAGES OF THE SAS MODEL

The SAS computer model offers the analyst complete control over all parameters which significantly effect the outcome of a small scale engagement situation. The SAS model accomplishes this goal while enabling the analyst to complete an analysis in a relatively short time-frame. By use of standardized data input forms, the completeness and correctness of all input data can be

4
assured before executing the model. After the model is executed, data is entered from the input forms directly in a specified order, thus minimizing data entry time. SAS execution time is minimized by limiting the amount of Monte Carlo analysis to determining only probabilities of hit. All other calculations are analytical in nature, involving the use of conditional probability and expected value techniques, which are more efficiently handled by the computer. SAS evaluates the probability that personnel and material targets remain undestroyed with time. Thus definition of engagement termination criteria, such as the attack force refusing to continue after 25 percent of their total force has been killed, is left entirely for the analyst to define.

1.5 DISADVANTAGES OF THE SAS MODEL

Some simulation models are entirely Monte Carlo in nature, allowing the model to determine at any point in time whether a particular player is still alive or has been killed previously. This feature is useful if the analyst requires a deep level of detail. Using a model of this type, the analyst could develop a type of "script language" which could specify the actions of the various players based on other player's situations. For example, the analyst could model a scenario in which player 1 would move to a new location based on whether player 2 is still alive or has been killed. With SAS, the analyst has the capability to move a particular player to a new location by specifying the time relative to the beginning of the simulation when the movement of the player is to occur. Because SAS considers only the probability that a particular player has been killed, decisions cannot be made based on the "condition" of individual players. If this amount of detail is required, a code which employs Monte Carlo techniques throughout is suggested.

1.6 UNIQUE FEATURES OF THE SAS MODEL

Several unique features were built into the SAS model which were not generally available in other security models surveyed. These features have been found to be extremely useful in evaluating security/survivability concerns. (Reference 4)

- Specified groups of targets may be selected for further analysis to determine the expected number of targets remaining with time. In addition to the typical engagement outcome results describing damage to the defending and attacking forces, the user is able to obtain engagement damage to any set of targets. The type of targets which may be analyzed include vehicles, both ground based and air-borne, and any vital cargo being transported. Thus, in addition to providing security/survivability outcome, the output from SAS may be useful in logistics analysis where the expected number of vehicles destroyed, for example, would be important.
- Some targets may contain other targets; for example, a truck may contain a weapon container assembly which may contain a warhead, a fuzing mechanism, and an arming mechanism. This permits detailed analysis of each vulnerable component involved in the security system analysis.
- The model is also capable of evaluating the probability of target destruction in which the exact position of the target is not known, for example, the unknown target position of a weapon container in the bed of a canvas-covered truck, or personnel hidden by thick foliage in which their exact location cannot be determined.
- The degree of obscurity that a particular target or set of targets may possess can change during the simulation, for example, smoke canisters may be released during an engagement which could decrease the ability of firers to determine the exact location of enemy targets.
- Automatic weapon fire covering a random area may be modeled with SAS, as well as weapons which provide blast kill capability such as grenade launchers or rocket batteries, for example.
- SAS also has the capability of analyzing collateral target damage. Certain targets, if hit, may destroy or inflict damage

on other targets. For example, vehicle fuel tanks, when hit with incendiary fire, could ignite, inflicting damage on personnel and material targets within the vehicle.

1.7 AREAS FOR FUTURE DEVELOPMENT

Possible improvements to the SAS model are discussed in this section. Each improvement or refinement will be described and the reason the improvement would be useful and the additional capability it will provide is discussed. Improvements to be discussed fall primarily within the following major areas:

- Simplification of Input Data Entry

Those improvements which increase the ability of an analyst to generate and enter the data necessary to perform a simulation. These improvements would also reduce the time required to modify original data in order to analyze additional improvement options.

- Provide Additional Simulation Detail

Improvements which increase the detail of the simulation or provide a more "real world" analysis capability.

1.7.1 Target Movement

Presently, the SAS user must model target movement by analyzing the target in a series of fixed positions along the movement route. This technique may be used to analyze target movement by means of situation "snap shots" which freeze the target's position for a finite number of steps along the movement path. An improvement would permit the analyst to specify a beginning and ending position for a particular target along with its' movement rate. The computer would calculate the target's position at a particular instant in time. This capability would reduce the amount of input data required to describe target movement and would eliminate discontinuities in

output which often appear when the movement route is broken into a series of fixed positions.

1.7.2 Graphics Tablet for Data Entry

Interactive data entry using a digitizing graphics tablet would greatly reduce the amount of time required to specify terrain, cover, and target shape. Since input time would be reduced, more targets could be modeled in more detail.

1.7.3 Graphics Package for Verification of Input Data

Graphics software would display target shapes viewed at various perspective angles verifying the correctness of input data prior to simulation execution. This would permit an analyst to identify the vulnerable cross-sectional area of the target as viewed from a particular opposing force member's point of reference. The analyst could then use this information, or a subroutine could be developed, to evaluate the probability that the opposing force member could hit the target using the proper target orientation.

1.7.4 Data Base Development

A generalized data base would be constructed containing the sizes and physical characteristics of various targets and weapon types. The SAS user could then refer to this data base to acquire the data required for his particular simulation. This data base could either be computerized, allowing the analyst to select and incorporate the data automatically, or the data could be contained in document form in a manner compatible with the data entry format used by SAS.

1.7.5 Confidence Interval Estimation

The simulation model could be expanded to include confidence interval estimation. Generation of confidence intervals helps the analyst interpret the accuracy of the simulation results.

1.7.6 Relaxation of the Parallel Plane Restriction

Currently, opposing force personnel and material must reside in either the attack or defense force planes. Movement of personnel or equipment out of these planes is presently not permitted. This improvement would permit the analyst to specify all target positions in terms of x, y, and z coordinates. The coordinates specified would be relative to a single, fixed point origin in space, thereby allowing the analyst to place attack and defense force targets at any position desired.

1.7.7 Three-Dimensional Targets

Targets may be represented as three-dimensional volumes. Targets would be constructed of adjoining quadrilateral plates. The targets would be rotated to the "line of fire" from opposing force personnel and then the vulnerable cross-sectional area of the target would be computed for the firer's weapon type before the probability of hit is calculated.

1.7.8 Target Protection

Targets which cannot be destroyed or damaged by one round of opposing fire are not presently considered. For example, a payload well protected with armored blankets could not be damaged by an attack force using small caliber weapons assuming a single round direct hit, because the round would not penetrate target protection. It is understood, however, that subsequent rounds fired on or near the impact point of the first round could weaken the target protection sufficiently to result in payload damage or destruction.

The "multiple hit" possibility for those targets which require more than one direct hit to penetrate target protection should be incorporated into the SAS model. This improvement will permit the SAS model to analyze more complex survivability-oriented scenarios.

SECTION 2

INPUT DATA DESCRIPTION

Input data required for a typical SAS execution is divided into five major categories:

- Target Data

Target data is used to describe the size and shape of all personnel and material targets the user wishes to consider in the simulation.

- Firer Data

Data which describes a potential firer's characteristics, such as weapon type and response time, are included within this section.

- Aim Point Data

The aim point data section describes locations of aim points which may be fired at by opposing force personnel during the simulation.

- Target Status Change Description

Data describing a particular target's movement, size or shape change, or selection of new weapon type during the simulation would be included in this section.

- Other Input Data

This section contains all other input data not included within the first four categories. Names used for output files created during the simulation execution would be one example of the type of data required for this category.

The data required for each category is discussed in more detail in the text which follows.

2.1 TARGET DATA DESCRIPTION

For purposes of the SAS model, targets are objects whose destruction or damage benefits the opposing force in the engagement. If we assume a ground movement scenario in which a ground convoy is ambushed by a ground based attack force, possible targets of interest for the attack force would be defense personnel, the vehicles used to transport defense personnel and payload, and the payload itself. Attack personnel would be considered as possible targets for defense personnel. Targets are modeled by SAS as two-dimensional rectangular areas.

To allow for situations in which the exact position of the target is not revealed to the attacker, it is possible to specify an area in which the target may be placed. To clarify this feature of the SAS model, assume that the payload, which the attacking force is attempting to destroy, is located somewhere on the bed of a large truck in which the rear section of the truck is covered with canvas, preventing the attack force from directly determining the exact payload location. Further, assume that the size of the payload is small relative to the size of the canvas-covered truck bed. A target of this type can be modeled using SAS by defining the size of the target (the payload in this example) and specifying a random area in which the target may be located (the rear of the canvas-covered truck).

In some situations, certain targets, if hit, may inflict damage to other targets near by. For example, suppose the fuel tank of a vehicle was modeled as a target. The tank, when hit with incendiary attacker fire, could ignite and inflict damage to other targets such as defense personnel or vital payload. Lacking a better term, we will call this type of indirect target damage "collateral" damage. For each target identified and defined as input to the model, a list of targets which would be collaterally damaged by hitting this target must be specified. Before executing the SAS model, all targets must be identified and their size and position defined.

Due to aiming error, a firer may attempt to hit one target and actually hit another. For this reason, a list of targets which could be hit by firing at a specific aim point is required.

A standardized target data collection form used to obtain all required target information is presented in Appendix A. This data form also specifies data required to describe firer characteristics for personnel targets capable of firing at opposing forces.

There are several data items required to define each target to the SAS model. The format and use of each data item will be discussed separately as they appear on the data form, Appendix A, Figure A.1b.

2.1.1 Target Number

First, the user must select a target number. The target number selected for each target must be unique. Normally, targets are assigned numbers in sequence (1,2,...); however, the user may choose to assign target numbers in any order.

2.1.2 Target Description

Following the target number is the target description. Target description consists of at most 10 characters which are used to uniquely identify targets to the user. Any printable character may be included in the target description.

2.1.3 Target Height, Width

The height and width of the rectangular target area is measured in meters.

2.1.4 Random or Fixed

The user must now decide whether the target is random (lies within a random area, exact location is unknown to the opposing force) or fixed (the opposing force can recognize exact target location). An 'R' is entered

indicating the target is found within a random area or an 'F' is entered indicating the target is in a fixed location. The user must specify either an 'R' or an 'F' for each target listed.

2.1.5 Lower Left Coordinate

For fixed targets, the user enters the x and y coordinates specifying the position of the lower left corner of the rectangular target areas. The coordinate values are relative to the origin in target's reference plane. The x coordinate is the distance in meters from the origin in the horizontal direction. The y coordinate is the distance from the origin to the lower left corner in the vertical direction.

For random targets, the user enters the x and y coordinates specifying the lower left corner of a random target area within which the target is located. The x and y coordinate values are expressed in meters defining the horizontal and vertical distance for the geometric plane origin.

2.1.6 Random Area Height, Width

For random targets, these parameters specify the height and width of the random target area in meters. Random area height and width are left blank if a fixed target is specified.

2.1.7 Can Target Fire?

If the target is a personnel target capable of firing at the opposing force, the user enters a 'Y' indicating the target can fire; otherwise, an 'N' is entered. Either a 'Y' or 'N' must be specified for each target involved in the simulation.

2.1.8 Weapon Type

The user identifies the type of weapon used by entering a one or two digit number. The weapons currently defined for use within the SAS model are listed below with their associated weapon-type number.

Weapon Type	Weapon Description
-------------	--------------------

1	M16
2	AK47
3	M60
4	50 Caliber
5	M79
6	RPG
7	M19

The weapon type is left blank if the target cannot fire.

2.1.9 Number of Rounds

The user enters the number of rounds of ammunition available. This parameter is left blank if the target cannot fire.

2.1.10 Response Time

Response time specifies the time at which the firer begins firing at opposing forces relative to the beginning of the simulation (time = 0) measured in seconds. The parameter is left blank if the target cannot fire.

2.1.11 Aim Points, Ordered by Priority

Aim point numbers of those aim points which the firer may fire at during the simulation are ordered on their importance to the firer. When fire is first initiated, the firer will attempt to hit the first aim point listed. As the simulation progresses, the probability that the firer continues to fire at the first aim point decreases proportionally to the probability the primary target associated with the aim point has been previously destroyed. Aim point numbers are left blank if this target cannot fire.

2.1.12 Collaterally Damaged Targets

A list of those targets which will be destroyed if this target is hit by

the opposing force is the last data item required for target description. The order of the target numbers specified is not important. If fewer targets are specified than spaces on the data sheet, leave the rest of the spaces blank.

2.2 FIRER DESCRIPTION

Each attack and defense force member firing at the opposing force must be described. The description includes specification of a weapon type, response time (time when firer begins shooting), amount of ammunition available, and strategy. A summary of the input parameters describing a firer's characteristics (can target fire, weapon type, number of rounds, response time, and aim points ordered by priority) appearing on the target data form have been discussed in the previous section.

2.3 AIM POINT DATA DESCRIPTION

A firer's strategy is defined by providing a set of aim points which are ordered in terms of priority. Each aim point is associated with a primary target. Normally, the aim point is defined in the center of the particular situation modeled. In addition to the primary target, a list of other targets in close proximity to the primary target which may be hit by firing at this aim point must be provided. In some cases, it is more desirable to define a region for an attacker or defender to fire into than a specific point at which to aim. This is especially true if the exact position of the intended target cannot be determined. SAS allows spray of fire into an area by defining a rectangular shaped area in place of a fixed aim point. Shots are fired randomly into the rectangular area assuming each section of the area is equally likely to be fired into.

An sample form used to gather required aim point data is presented in Appendix A, Figure A.1c. The data items required for aim point description are in this section.

2.3.1 Aim Point Number

The aim point number is a unique number which the user assigns to each

aim point considered in the simulation. Aim point numbers are usually defined in sequence (1,2,...); however, they may be assigned in any order which may be more convenient to the user.

2.3.2 Aim Point Description

The user must specify a unique aim point description which consists of at most 10 characters for each aim point desired. Any printable character is permissible within the aim point description. Normally, the aim point description references the primary target associated with the aim point.

2.3.3 Random or Fixed

If the user wants to define the aim point as fixed, an 'F' is entered; otherwise, a 'R' is used. Fixed aim points specify fixed points located in the plane containing the opposing force personnel and equipment to be fired at. Fixed aim points are specified whenever the opposing force can obtain an exact location of the opposing target area center.

Random aim areas, indicated by a 'R' entry, are used whenever the exact location of an opposing target area is unknown or the strategy is to spray automatic weapon fire evenly throughout an area without aiming specifically at any target.

The user must define each aim point considered as either random or fixed.

2.3.4 X Coordinate, Y Coordinate

If the aim point is fixed, the x and y coordinate values specify the location of the aim point in the opposing force reference plane. The x and y coordinate values measure, in meters, the horizontal and vertical distance from the origin to the aim point. Normally, for fixed aim points, the aim point is placed in the center of the primary target associated with the aim point.

If the aim point is random, the x and y coordinates specify the location

of the lower left corner of a random aim area measured in meters from the opposing plane origin.

2.3.5 Height, Width

Height and width are only specified when a random aim point is desired. The height and width, expressed in meters, indicates the size of the random aim area. These data fields are left blank if a fixed aim point is used.

2.3.6 Targets Which May be Hit by Firing at This Aim Point

The last data entered on the aim point data form involves a list of possible targets which may be hit when this aim point is fired at. The first target number entered is considered the primary target associated with the aim point. Order of subsequent target numbers is not important. Should the user specify fewer target numbers than spaces on the data form, the remaining spaces should remain blank. The target which is closest to the aim point or most likely to be hit by firing at the aim point should be listed as the primary target.

2.4 TARGET STATUS CHANGE DATA

A target status change is specified by entering four parameters: the old target number, the new target number, the time the status change occurs, and the delay time for the new target to commence firing after the status change occurs. The data form used to obtain target status change information appears in Appendix A, Figure A.1d. The target status change parameters are now discussed in more detail.

2.4.1 Old Target Number

The old target number is the number of the original target assigned at the beginning of the simulation. This number identifies the target whose status will be modified.

2.4.2 New Target Number

The new target number points to a new set of target information which will replace the target information previously obtained when referencing the old target number. Information pertaining to the new target is transferred to the old target, destroying previous information used to define the old target characteristics.

2.4.3 Status Change Time

Status change time specifies the time in seconds from the start of the simulation when the target status change is to occur.

2.4.4 Delay Time

The user may choose to specify a delay time before the new target commences firing after the status change is complete. The delay time parameter can be used to specify a player's reload time, aim time, or time required to acquire a new weapon .

2.5 OTHER INPUT DATA

Other data required to perform a SAS simulation primarily concerns the specification of input and output files, names, starting and ending time for simulation results, and manner in which results are stored when the simulation is complete. Due to the self-explanatory nature of these parameters, they are not described within this section. Refer to Appendix A, Figure A.1a and Figure A.1e, for questions asked to obtain additional input parameters required to perform a SAS simulation.

SECTION 3

EXAMPLE APPLICATION

The application of the SAS model to evaluate the outcome of an ambush attack on a nuclear convoy will be presented. The objective of the attack is to destroy or render the weapons being transported by the convoy inoperative. The scenario details and modeling assumptions required to evaluate the engagement outcome are presented in two sections: scenario overview and scenario details. The scenario overview describes the scenario in a general sense and is used to acquaint the reader with the situation just prior to ambush. The second section, scenario details, provides more detailed assumptions concerning movement rates, weapon characteristics, number and location of defense and attack force personnel, vehicle type and size, and both attack/defense personnel cover and protection.

3.1 SCENARIO OVERVIEW

A nuclear convoy is transporting weapons to a fixed storage site. The convoy is enroute on a public highway when the attack occurs. The convoy consists of nine vehicles which carry defense personnel, nuclear weapons and required communication equipment. The attack on the convoy occurs without forewarning; hence, the defense force is taken totally by surprise. The attack is initiated by detonation of pre-placed mines under the roadway used by the convoy. Detonation of these mines is timed to effectively destroy the lead weapon-carrying vehicle and prevent further progression of remaining weapon-carrying vehicles beyond the mine detonation point. The mine detonation also isolates vehicles and personnel which have proceeded beyond the mine detonation point from involvement in the ensuing engagement between defense and attack forces concerned with protection and destruction of weapons in remaining weapon-carrying vehicles. Figure 3.1 shows a "bird's eye" view of the scenario situation just prior to mine detonation.

The attack force consists of 11 men who have pre-positioned themselves in well-hidden, protective foxholes or behind large trees to minimize the amount

LMG(Light Machine Gun)

RPG(Rocket Propelled Grenade Launcher)

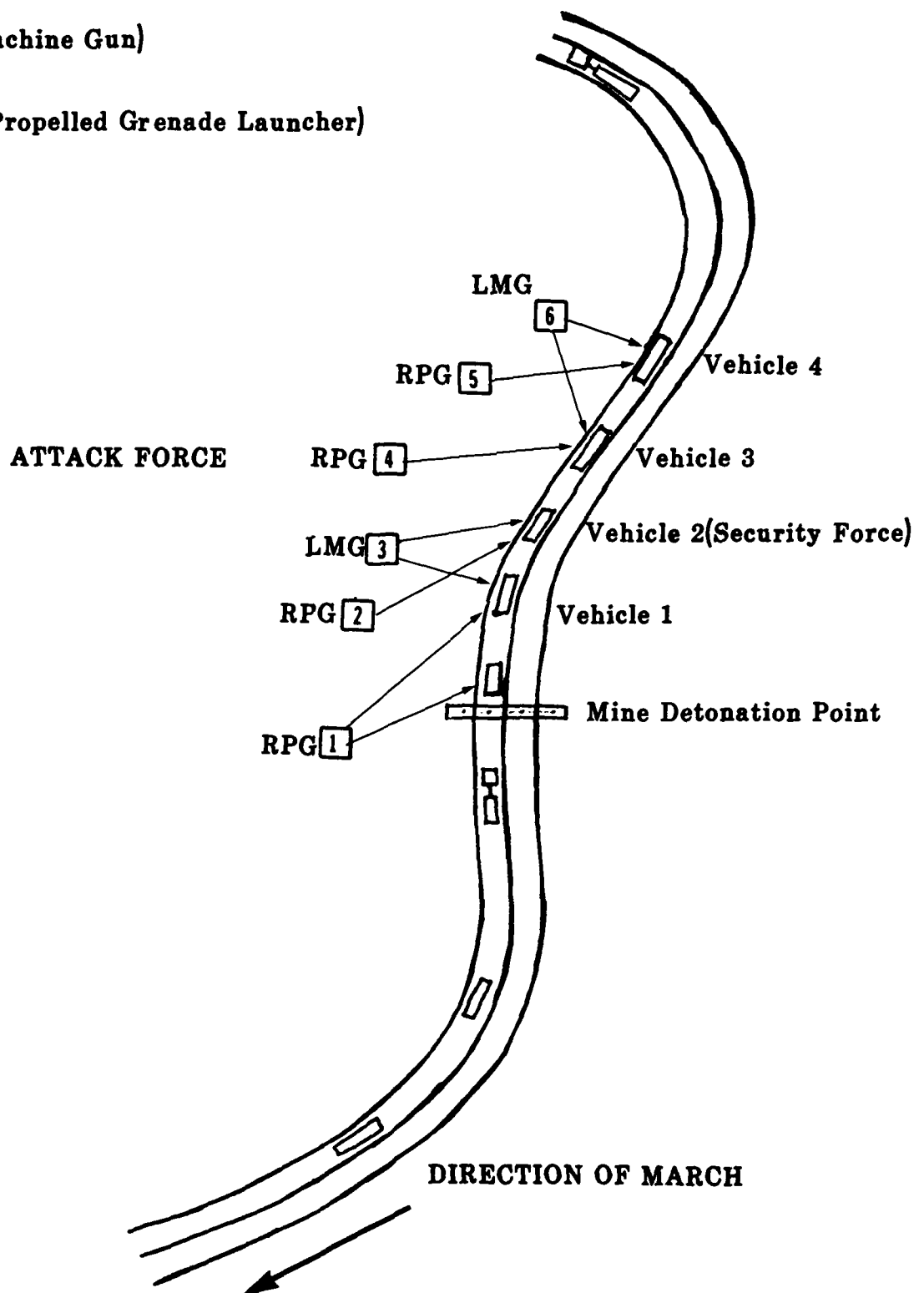


Figure 3.1 "Bird's Eye" View of Ambush Scenario Just Prior to Mine Detonation

6
of body area exposed to defense force weapon fire. Each attack force member has well-defined objectives and strategy when the engagement starts. A portion of the attack force is used to tie down the three lead vehicles and the one trailing vehicle, preventing personnel on these vehicles from directly assisting personnel located on weapon carrying vehicles throughout the engagement. It will be assumed that the attack force personnel used to pin down defense personnel in the lead and trailing vehicles will not enter into the engagement between defense personnel on remaining weapon carrying vehicles. Thus, the encounter situation which will be modeled by SAS involves the remaining six attackers (numbered 1-6 in Figure 3.1) and the defense personnel on trucks (numbered 1-4).

Since the attack occurs without forewarning, defense personnel do not respond with return fire immediately. In all cases, the first person responding from each vehicle is the assistant driver. After the vehicles have stopped, defense personnel located in the rear section of each vehicle, along with the vehicle driver, will respond. The vehicles will attempt to proceed through the ambush area, if possible. Although the defense vehicles have limited off-road capability, the attack force has carefully chosen its ambush site to prevent vehicles from continuing beyond the mine detonation point. The vehicles are assumed to stop on the roadway grouped fairly closely together, with the lead vehicle approximately five meters from the mine detonation point.

The convoy is in continuous communication with an additional back-up response force. Once the attack has been determined, the back-up response force is notified via personnel in the command-control and communications vehicle. The time required for the back-up force to respond to the ambush attack is five minutes. Due to the expected short duration of the engagement, additional support provided to the defense personnel on Trucks 1-4 by the back-up response force will not be considered.

3.2 SCENARIO DETAILS

All necessary assumptions required to perform the simulation, with the exception of specific individual player strategy and collateral target damage

assumptions, are described in this section. The purpose of this section is to:

- describe assumptions used in this particular example, and
- familiarize the reader with the parameters which must be identified and defined prior to performing any simulation involving the use of the SAS computer model.

3.2.1 Convoy Composition

The nuclear convoy, shown in Figure 3.1, is composed of four load-carrying vehicles, one security personnel vehicle and the lead and trailing escort vehicles. Each load-carrying vehicle contains three nuclear projectiles within a standard container. The security vehicle contains six security personnel and equipment. The vehicles composing the convoy are ordered as follows:

- Route recon vehicle
- Command-control and communications vehicle
- Security-lead vehicle (M115)
- Two load-carrying vehicles
- Security force vehicle
- Two load-carrying vehicles
- Trailing security vehicle

3.2.2 Security Force Distribution

All vehicles contain a driver and assistant driver, both trained and equipped with issue M16 rifles. All load-carrying vehicles are equipped with an M60 as well as the M16 rifles. The ambush procedure of mine detonation under the first load-carrying vehicle isolates the three leading vehicles from the main body, while the trailing security vehicle is also diverted by direct fire. These security elements are, therefore, unable to directly assist the main body. The force-on-force encounter then involves principally security force personnel (drivers and guards) of load-carrying mission vehicles and the security guard vehicle personnel, excluding the defense personnel on the lead

load-carrying vehicle which was destroyed by the mine detonation.

3.2.3 Personnel Assumptions

Defense personnel are assumed to remain completely exposed to fire. Since time involved in the stand-off attack phase of the SOT assault is only 20 seconds, it is assumed that if a defender sustains a hit, he will be considered ineffective in providing retaliatory fire. For this reason, defenders, while in moving vehicles, are assumed to have 0.9 square meters of cross sectional area exposed to fire. Defenders in moving vehicles are modeled as rectangles with height of 1.2 meter, width of 0.75 meters. Figures 3.2 through 3.13 show the size and position of defense personnel within their respective vehicles.

Notice that in each of the figures there is a number associated with each vehicle and each defender within a particular vehicle. These numbers will be used to uniquely identify each defender involved in the simulation. For example, D1T1 denotes Defender 1, Truck 1. Similar numbers are used to uniquely identify weapons, weapon containers, defense force vehicle engine areas and members of the attacking force. W1T1, K1T1, and E1T1, for example, will be used to denote Weapon 1, Truck 1, container for Weapon 1, Truck 1, and Truck 1 engine area, respectively. A1 is used to uniquely identify attack force member 1.

After vehicles stop, defense personnel are assumed to be limited in their selection of possible defense positions. By scenario definition, the convoy is being attacked from the right. On the left side is a steep drop which cannot be used effectively for defensive positions and precludes vehicle maneuvering. Thus security personnel are forced to take defensive positions behind their respective vehicles. Security personnel remain in prone positions exposing approximately 0.04 square meters of their body to fire, modeled as a rectangular target 0.15 meters high, 0.25 meters wide. Figures 3.10-3.13 show the position of defense personnel after vehicles have stopped, while Figures 3.6-3.9 are the intermediate positions.

Attack personnel are in defensive positions exposing only their heads to

Truck 1 Just Prior to Mine Detonation

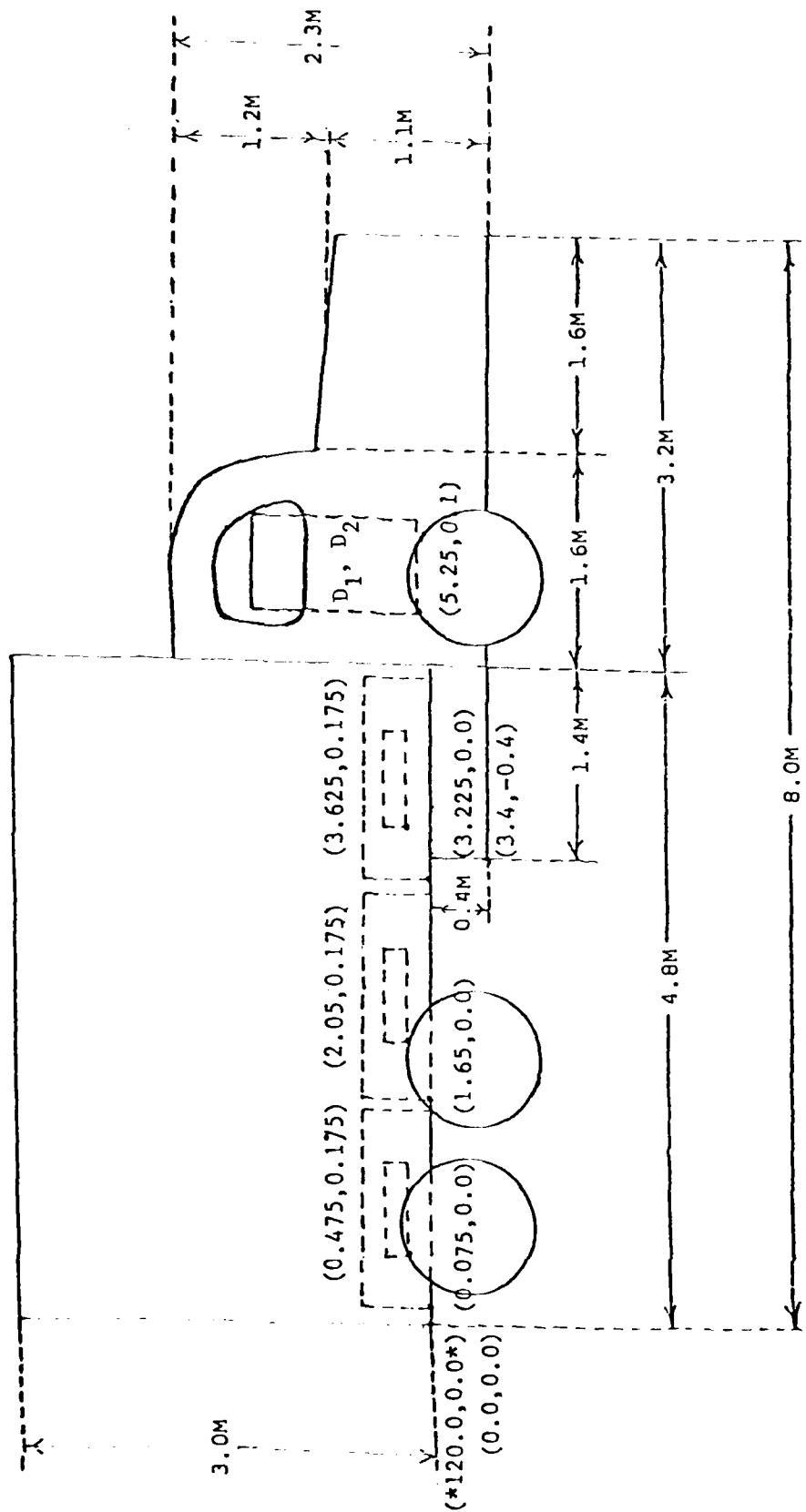


Figure 3.2 Defender Target Locations, Truck 1, Position 0

Truck 2 Just Prior to Mine Detonation

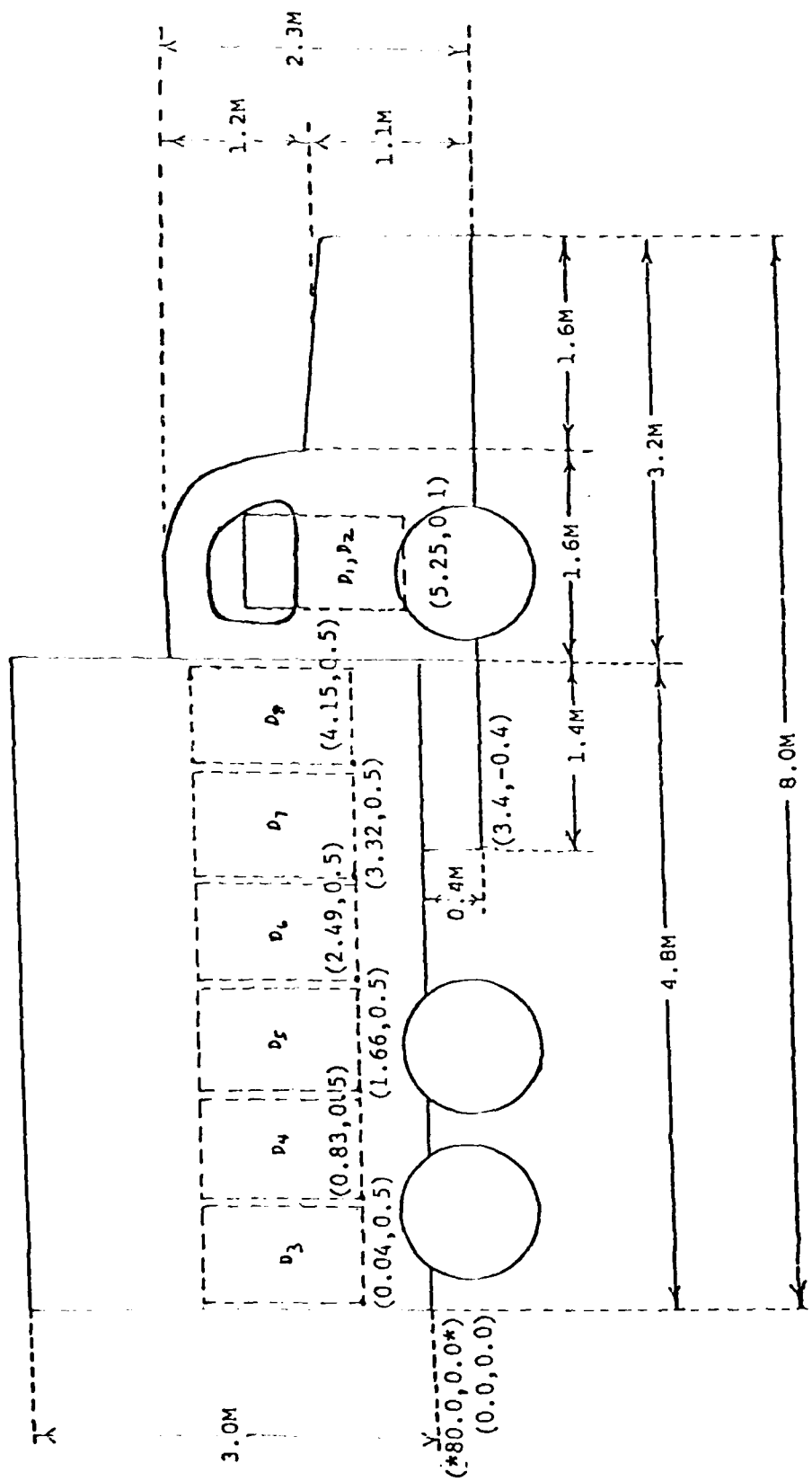


Figure 3.3 Defender Target Locations, Truck 2, Position 0

Truck 3 Just Prior to Mine Detonation

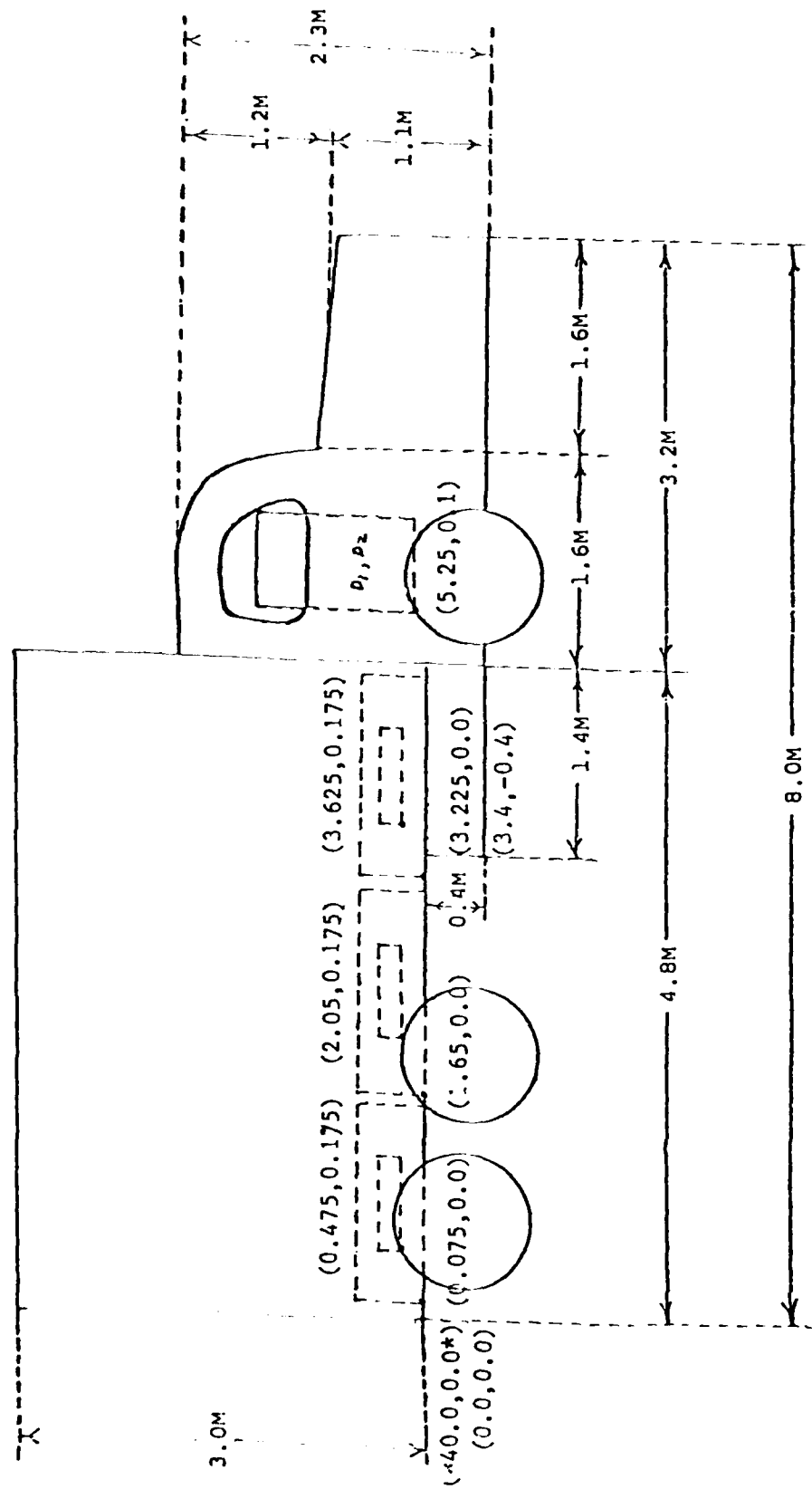


Figure 3.4 Defender Target Locations, Truck 3, Position 0

Truck 4 Just Prior to Mine Detonation

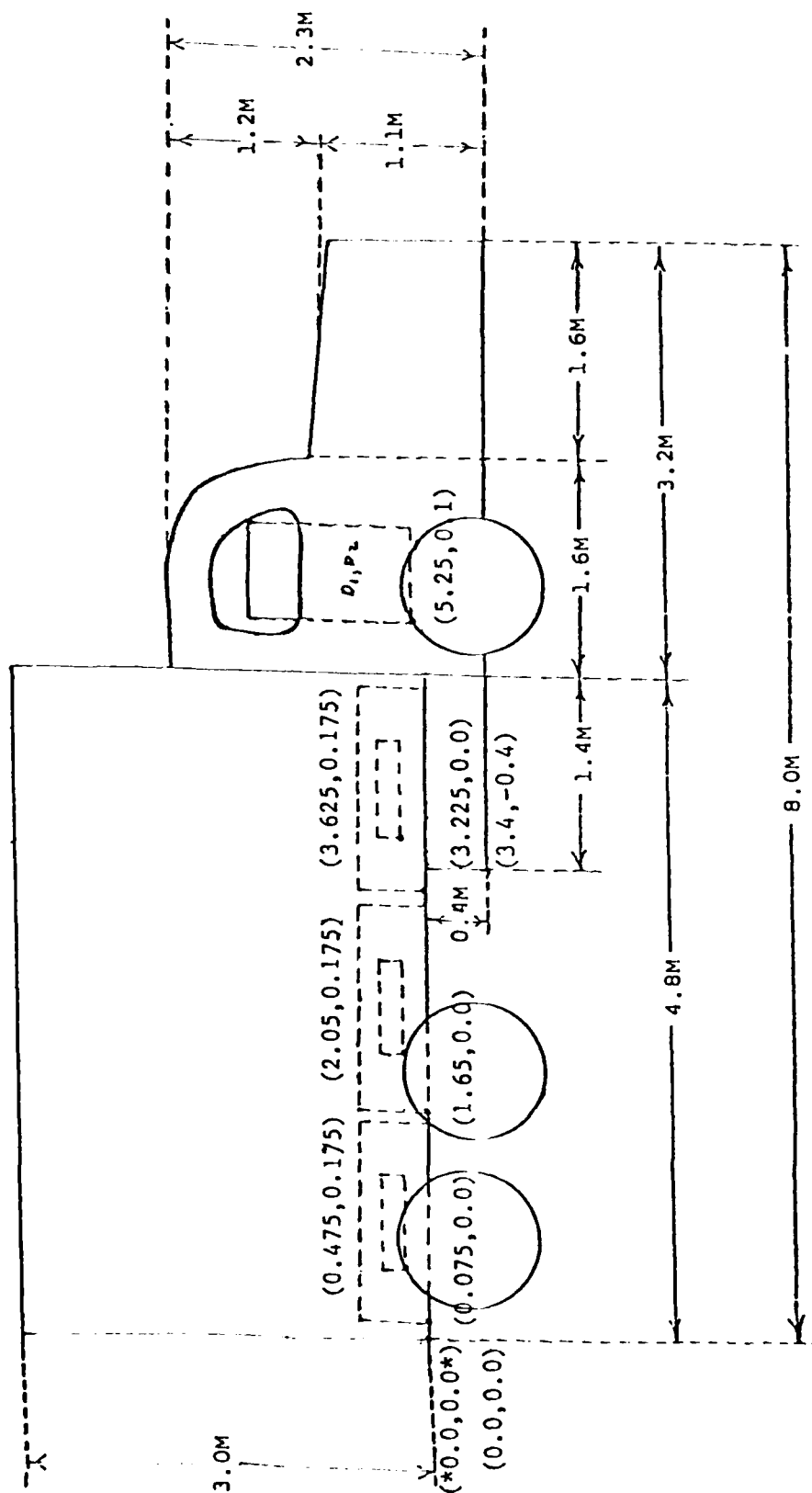


Figure 3.5 Defender Target Locations, Truck 4, Position 0

Truck 1 Moving State

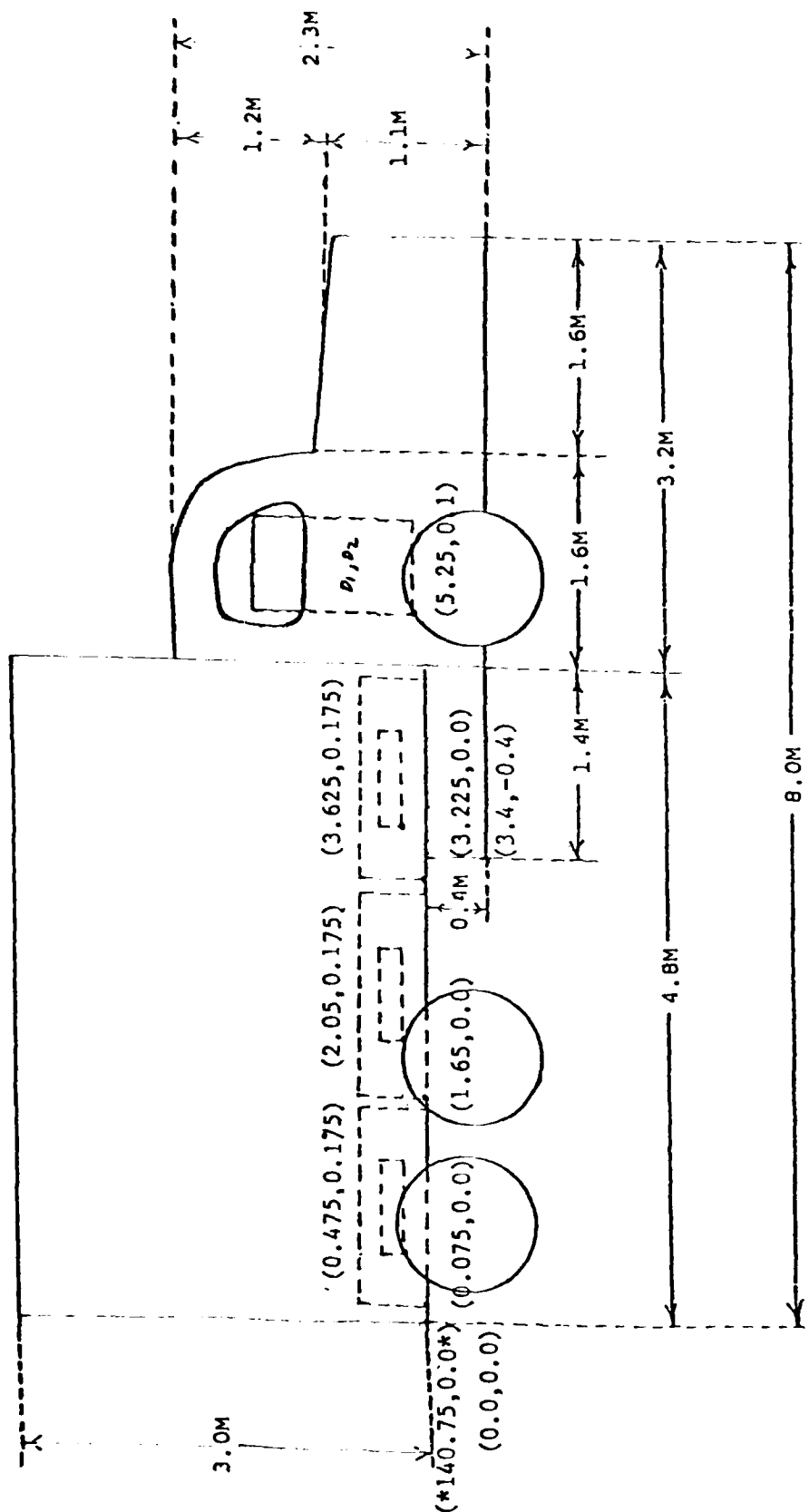


Figure 3.6 Defender Target Locations, Truck 1, Position 1

Truck 2
Moving State

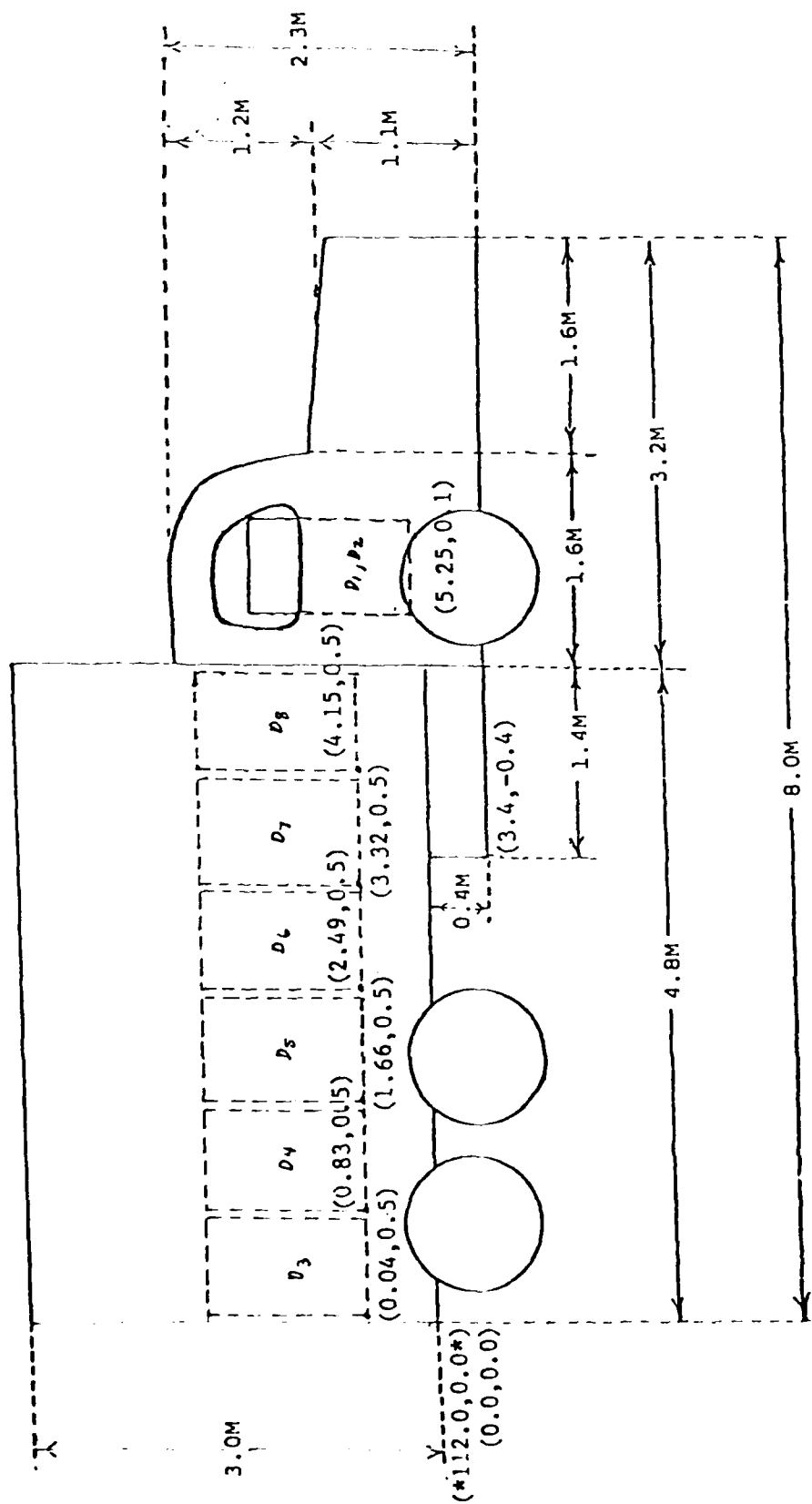


Figure 3.7 Defender Target Locations, Truck 2, Position 1

Truck 3 Moving State

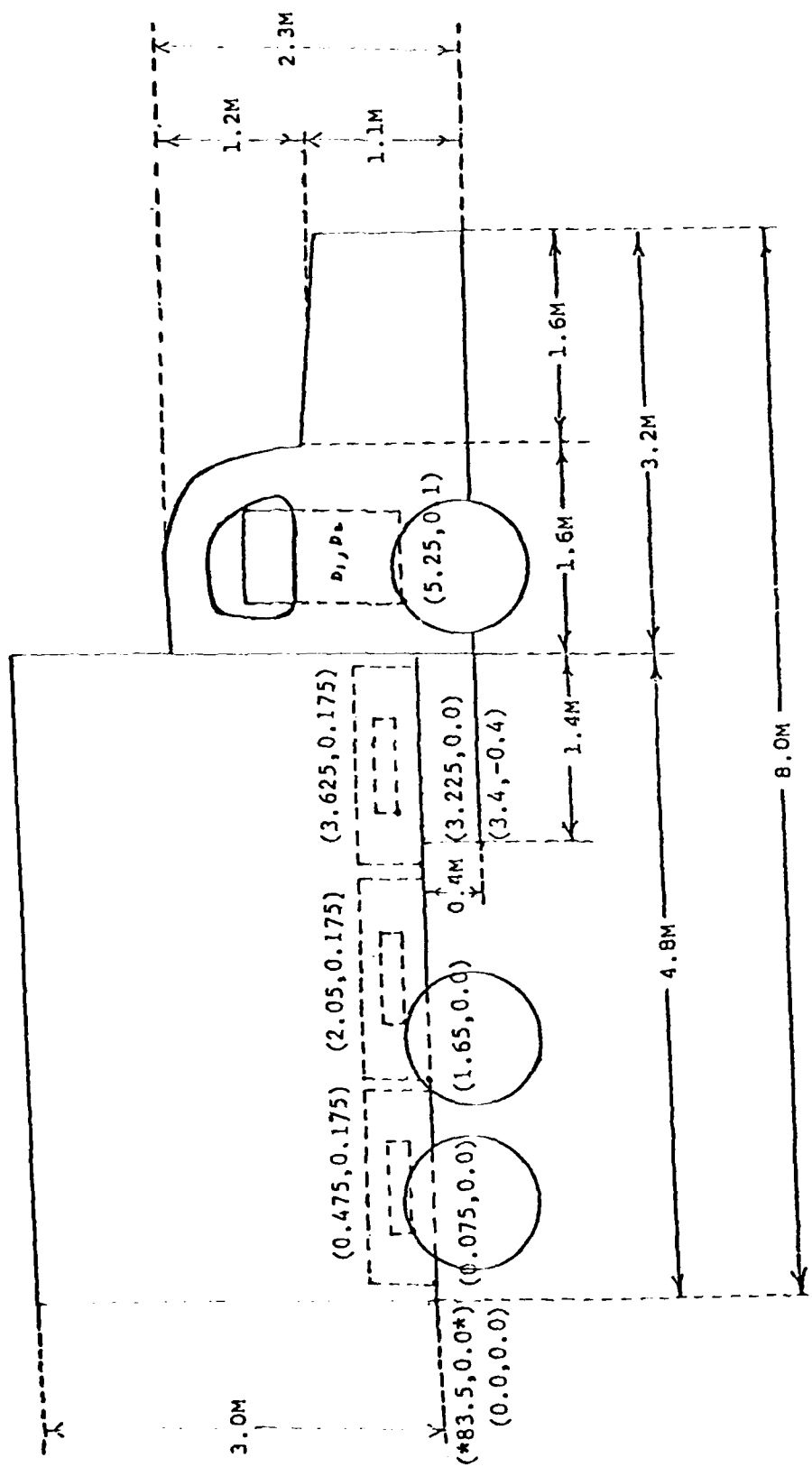


Figure 3.8 Defender Target Locations, Truck 3, Position 1

47

Truck 1
Stopped State

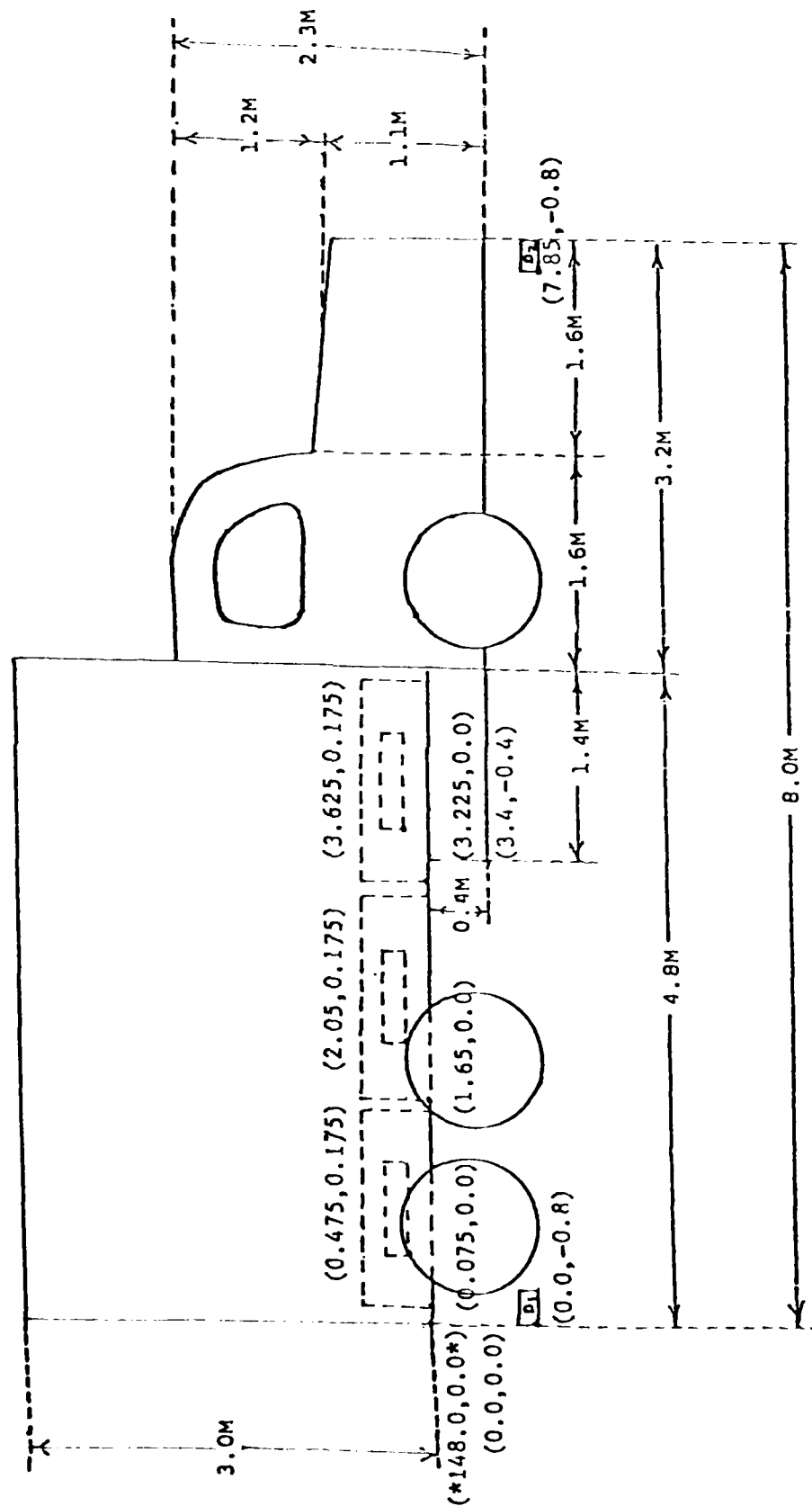


Figure 3.10 Defender Target Locations, Truck 1, Position 2

Truck 2 Stopped State

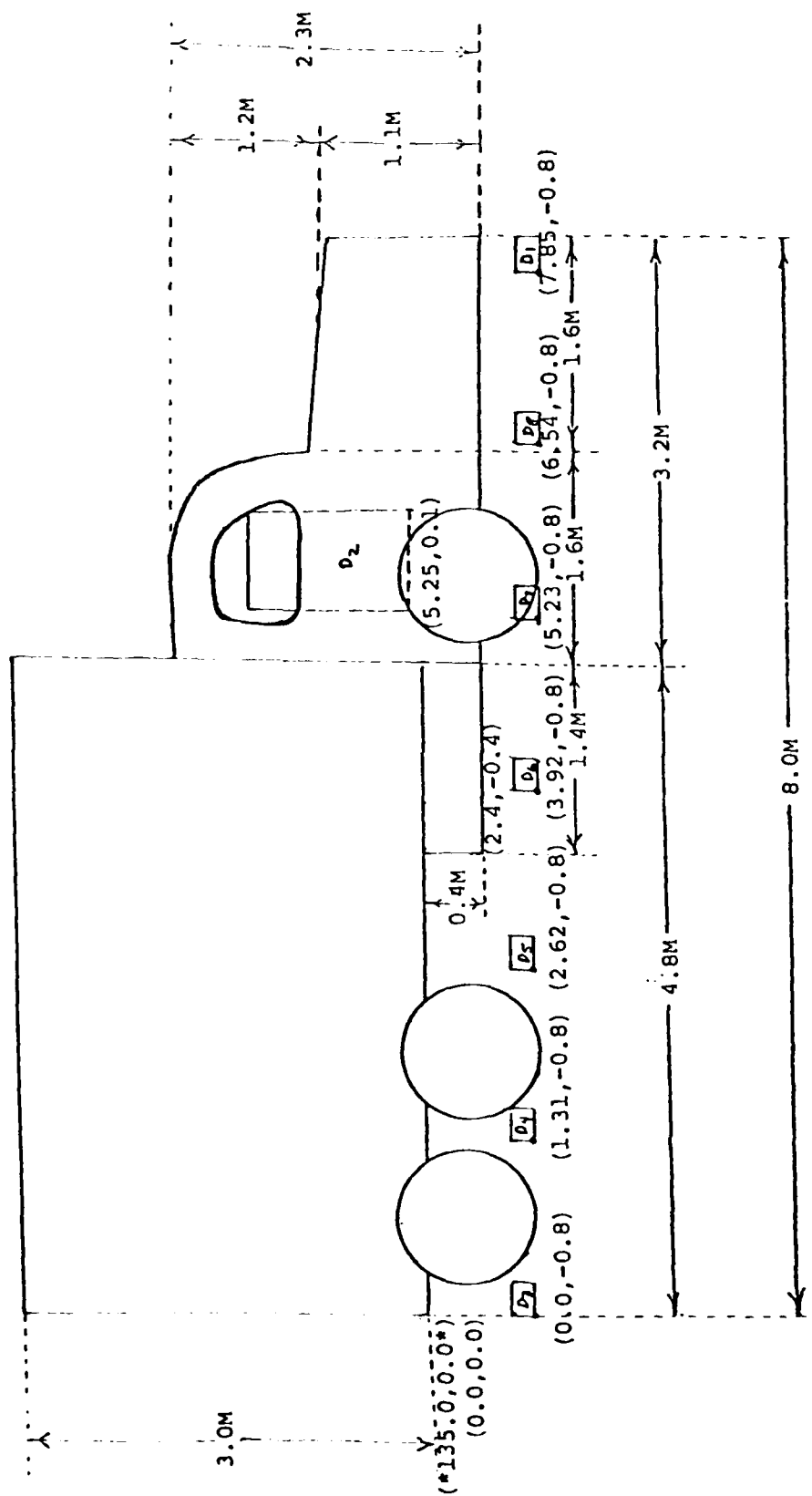


Figure 3.11 Defender Target Locations, Truck 2, Position 2

Truck 3 Stopped State

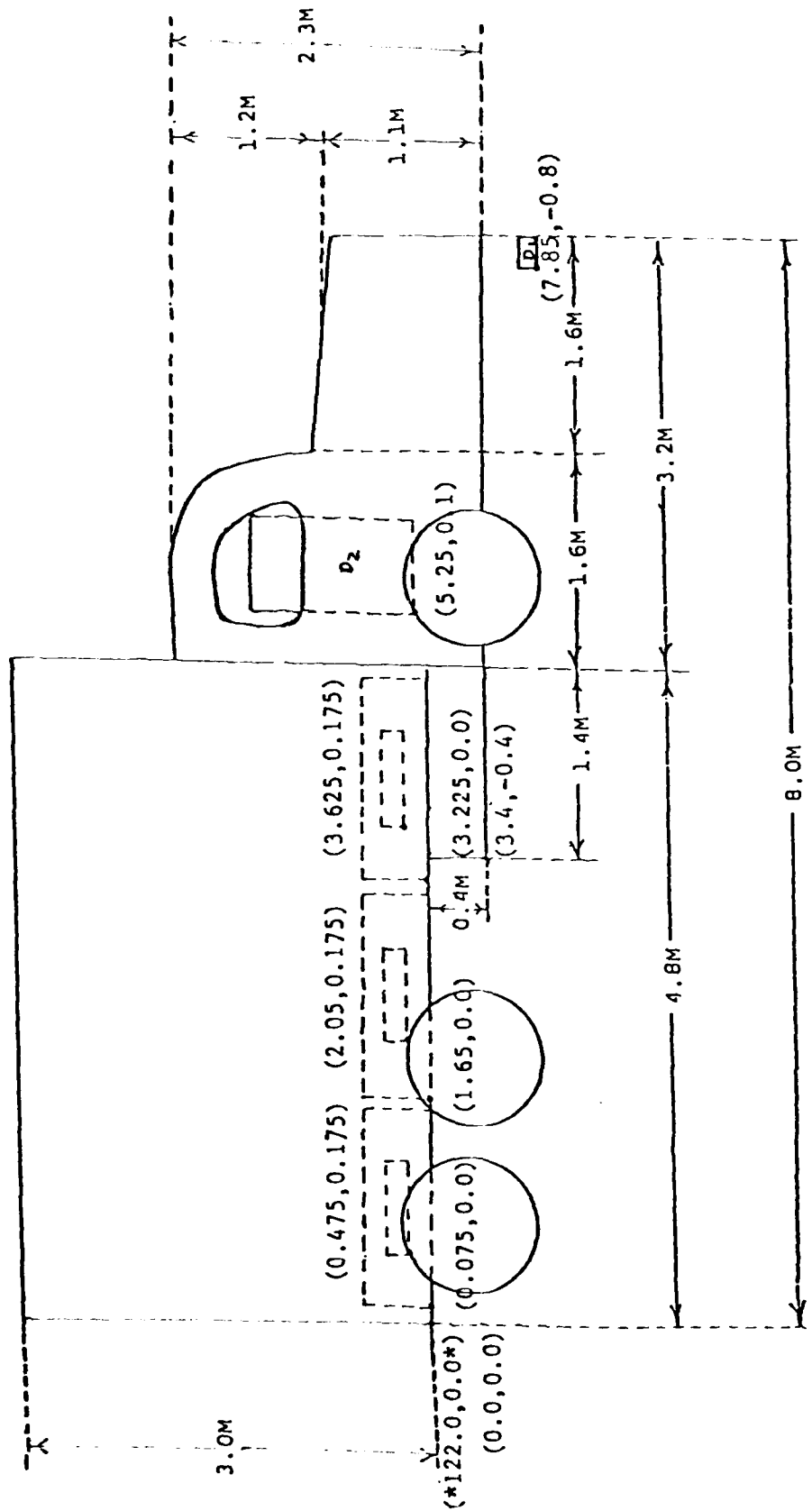


Figure 3.12 Defender Target Locations, Truck 3, Position 2

Truck 4 Stopped State

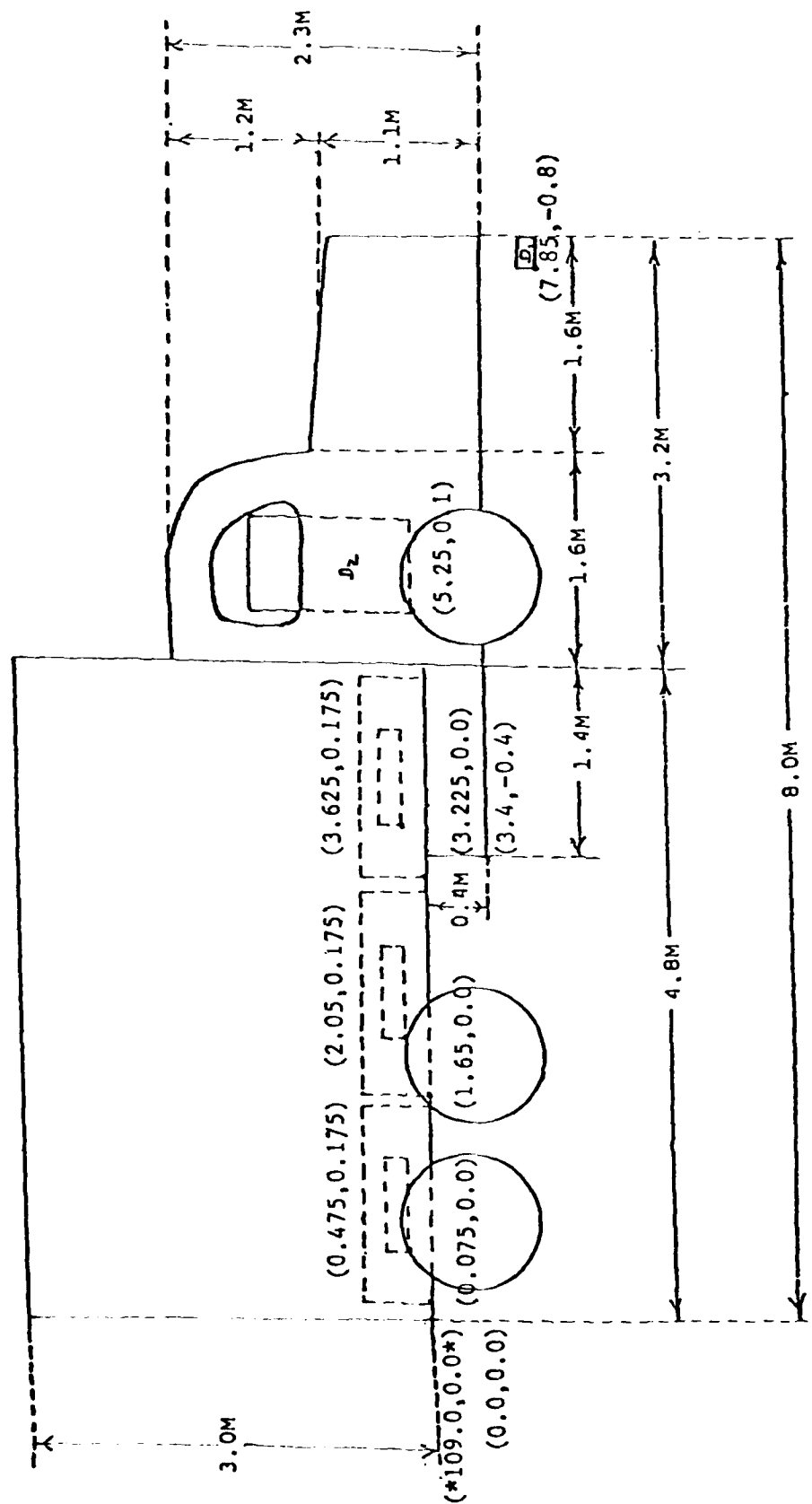


Figure 3.13 Defender Target Locations, Truck 4, Position 2

direct fire from the security defensive personnel. It is assumed that roughly 0.02 square meters of each attacker's body is exposed to direct fire, modeled as a square target measuring 0.125 meters on each side. Refer to Figure 3.14 for size and location of attack force members. A particular attack force member is identified by the character 'A' followed by the number of the attacker. A1, for example, is used to refer to the attack force member 1.

3.2.4 Vehicle Assumptions

Detailed dimensions of each vehicle involved in the stand-off attack simulation are given in Figures 3.2-3.13. All vehicles involved are assumed to be canvas-covered flatbed stake trucks. Each truck is 8.0 meters in total length. The engine section of each vehicle measures 1.1 meters in height by 1.6 meters in width. Vehicles are assumed destroyed if hit by an RPG in the engine section, or by fire if the fuel tank is ignited. The cab section of each vehicle is assumed to measure 2.3 meters in height by 1.6 meters wide. The canvas-covered rear section of each vehicle is modeled as a rectangle 4.8 meters long and 3.0 meters high. The fuel tank is located directly behind the vehicle cab below the truck bed. The fuel tank is also modeled as a rectangular target measuring 0.4 meters high by 1.4 meters in length.

Vehicle engine areas are denoted by 'ET' followed by the truck number. For example, ET1 is used to refer to the engine area of Truck 1. The rear section of the canvas-covered truck is uniquely referred to by 'CT' followed by the truck number. Thus, CT1 would be used to identify the canvas-covered rear section of Truck 1. Vehicle fuel tanks are denoted by 'GT' followed by the truck number. GT1 is used to refer to the fuel tank on Truck 1.

3.2.5 Vehicle Movement Assumptions

At the point of mine detonation, all convoy vehicles are assumed to be traveling at 40 KPH. Immediately after mine detonation, all vehicles decelerate uniformly. The distance required for each vehicle to come to a complete halt was determined by subtracting the starting position (x coordinate value in Figures 3.2 through 3.5) from their stopped position (x coordinate value in Figures 3.10 through 3.13). The stopped positions were

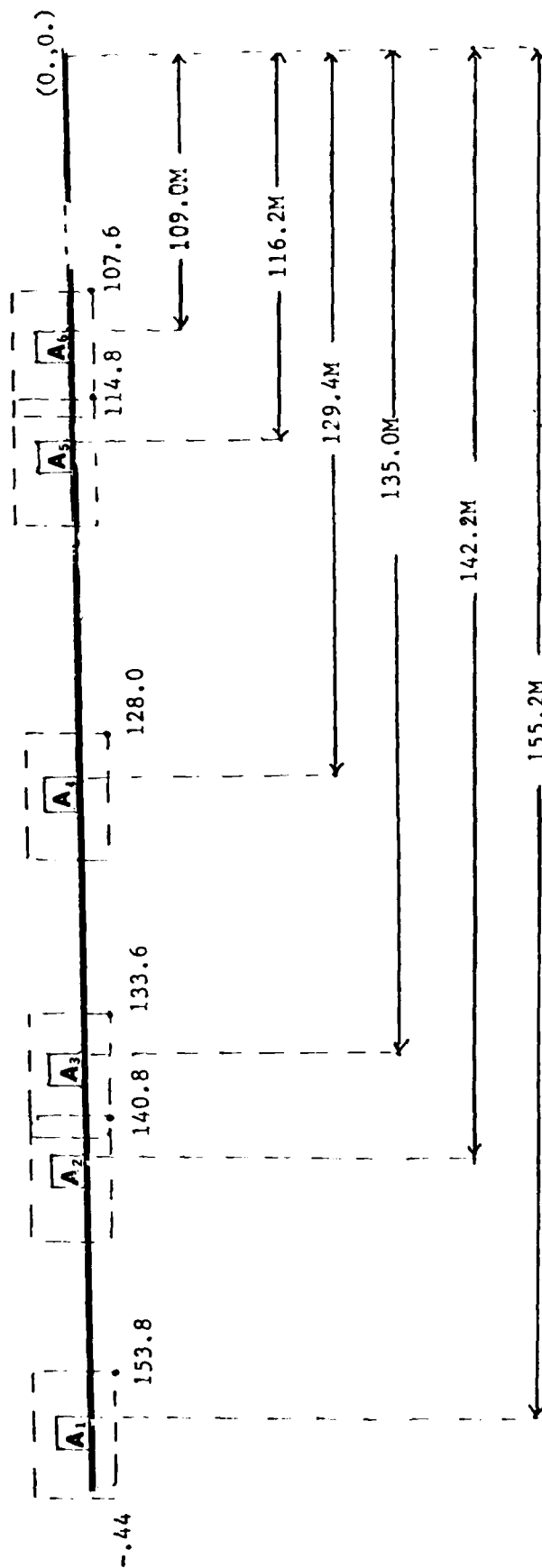


Figure 3.14 Attacker Target Locations

selected to provide a minimum of 5 meters separation between each vehicle. The distance for Truck 1 to come to a complete stop is $148 - 120 = 28$ meters, for example. Assuming each truck decelerates uniformly, it is possible to calculate average positions of each vehicle. Truck 1, which is closest to the mine detonation is assumed to stop in 5 seconds. The last vehicle, Truck 4, is assumed to require 10 seconds. The other two vehicles are evenly spaced over the 5-10 second interval. Truck 2 is assumed to stop in 6.67 seconds, and Truck 3 is assumed to stop in 8.33 seconds.

The location of each truck after mine detonation is then given by the expression:

$$x(t) = x(o) + v(o)t - 1/2 at^2$$

where:

- $x(t)$ = location of truck in meters from origin (along x axis)
- $x(o)$ = initial truck position in meters from origin (along x axis)
- $v(o)$ = initial truck velocity in meters/second (40 KPH = 11 meters/second)
- a = constant value of deceleration in meters/second²
- t = time elapsed since mine detonation in seconds.

The location of each vehicle when stopped is given by:

$$x(t_s) = x(o) + v(o)t_s + 1/2 at_s^2$$

where:

- $x(t_s)$ = location of truck when stopped measured from the origin along the x axis
- t_s = time required for truck to stop (point when $v(t) = 0$.)

Because the deceleration of each truck is constant, it is possible to solve for "a" in each expression and set the resulting expressions equal:

$$a = 2/t^2 [x(t) - x(0) - v(0)t]$$

$$a = 2/t_s^2 [x(t_s) - x(0) - v(0)t_s]$$

Thus,

$$2/t^2 [x(t) - x(0) - v(0)t] = 2/t_s^2 [x(t_s) - x(0) - v(0)t_s]$$

or

$$x(t) = x(0) + v(0)t + x[t_s - x(0) - v(0)t_s][t/t_s]^2.$$

The average position of each truck is determined at time $t = t_s/2$ in that deceleration is constant.

For example, the average position of Truck 1, x_1 , is then given by:

$$\begin{aligned} x_1 &= 120. + 11(5./2) + [148 - 120 - 11(5.)]\left[\frac{5/2}{5}\right]^2 \\ &= 140.75 \text{ meters.} \end{aligned}$$

Figure 3.15 illustrates the location of all trucks in terms of distance from the origin along the x axis in meters varying with time from mine detonation. For analysis purposes, the average truck positions will be used during the period when the trucks are decelerating.

3.2.6 Weapon Assumptions

Nuclear weapon projectiles are assumed to have a cross sectional area of 0.105 meters, which, if hit by either automatic weapon fire or RPG fire, will result in damage or destruction. The cross-sectional area resulting in destruction was assumed to be 0.15 meters high by 0.7 meters wide. Each weapon-carrying vehicle is loaded with three nuclear projectiles. The projectiles/containers are secured in the center of the truck bed, arranged in a linear fashion equally spaced over the bed length.

Standard containers were placed around weapons which, if hit by rifle fire outside the dimensions of the nuclear round, would cause the weapons within the container to be rejected, pending a detailed inspection and certification process. Components such as the fuzing and arming mechanisms would be included within the weapon container target. Each container was assumed to contain one nuclear weapon. Container targets with associated damage and destruction hit regions are shown in Figures 3.10-3.13. Each container is assumed to be 0.5 meters high and 1.5 meters long.

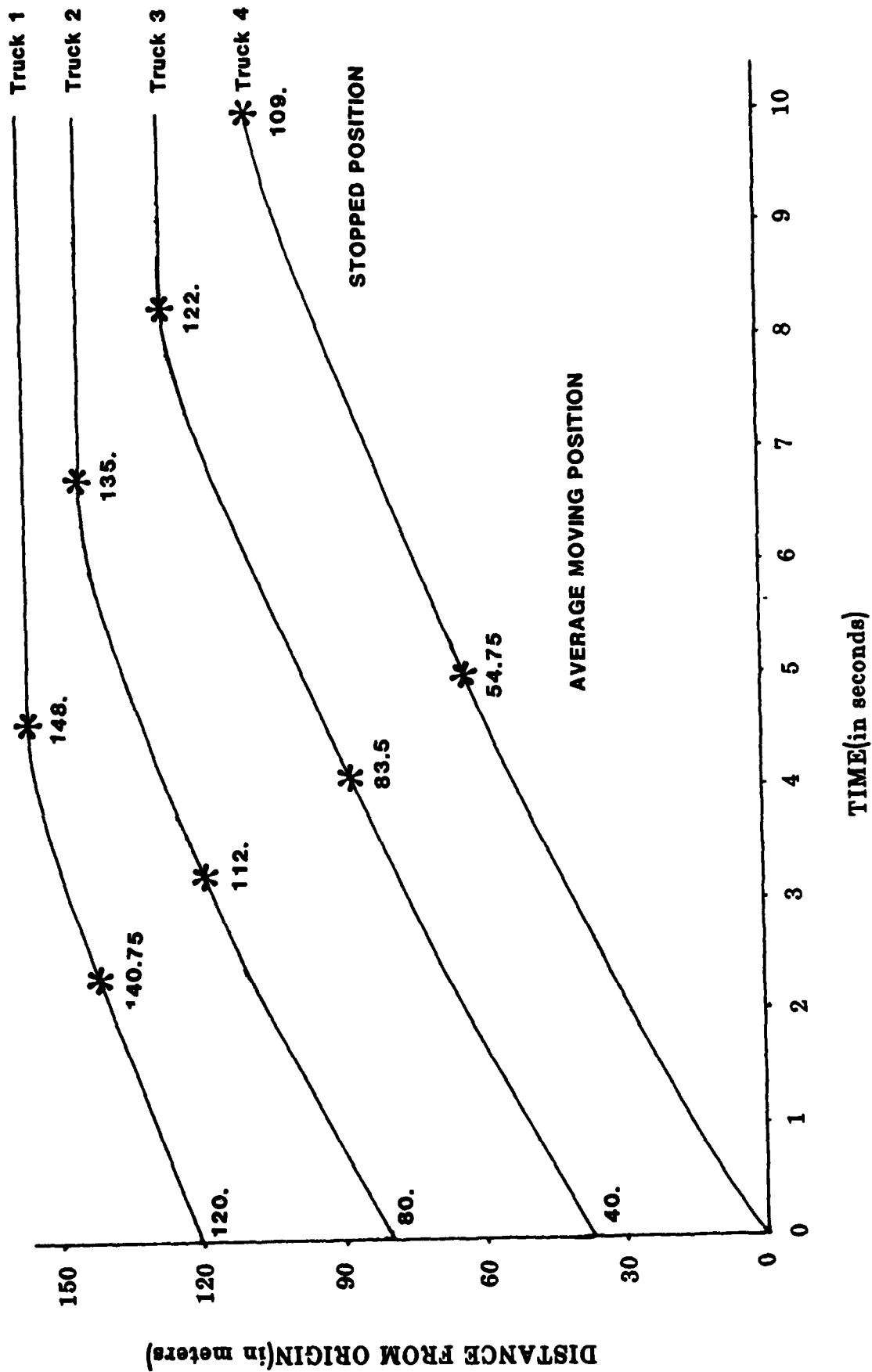


Figure 3.15 Truck Locations Versus Elapsed Time After Mine Detonation

The weapon area resulting in destruction if hit is denoted by the weapon number followed by the truck number. W1T1 is used to refer to Weapon 1, Truck 1, for example. A particular weapon container is denoted by the container number followed by the truck number: K1T1 is used to refer to Weapon Container 1, Truck 1. The average position of the weapons and weapon containers within moving vehicles (considered Position 1) is shown in Figures 3.6 through 3.9. The position of weapons and weapon containers within stopped vehicles (considered Position 2) is shown in Figures 3.10 through 3.13. The second position of all targets is denoted by adding 'P2' to the end of their unique three or four character designation. Thus, K1T1P2 is used to refer to Container 1, Truck 1, position 2, for example. Character designators without the P2 suffix refer to targets in Position 1.

3.2.7 Defender Strategy Assumptions

Once the convoy is aware of the attack, an attempt will be made to continue with the convoy movement and push through the ambush area. However, by scenario definition, the roadway is completely blocked by the mined vehicle and debris generated from the initial mine detonation. As the vehicles decelerate, the assistant drivers are assumed to be the only defensive personnel capable of immediate response. Each assistant driver will direct counter fire at the source of received automatic weapons fire (attackers designated numbers 3 and 6). After both attackers with automatic weapons are defeated, defensive fire will be directed at attack force personnel having RPGs. After all vehicles have halted, defenders use trucks for cover and concentrate counterfire primarily on attackers with automatic weapons. When attackers possessing automatic weapons are defeated, fire will then be directed at the source of RPG fire (Attackers 1, 2, 4 and 5).

3.2.8 Attacker Strategy Assumptions

In general, Attackers numbered 1, 2, 4 and 5 attempt to destroy Trucks 1, 2, 3 and 4, respectively, by firing RPGs at the truck engines. After trucks have been stopped, the attackers use the RPGs to fire at rear sections of trucks, attempting to destroy either personnel or weapons. Attackers 3 and 6 attempt to eliminate defense personnel. Attacker 3 is assumed to concentrate

fire on defense personnel in Trucks 1 and 2, while attacker 6 attempts to defeat security personnel in Trucks 3 and 4. Attackers 3 and 6 will place most emphasis on hitting both drivers and assistant drivers before firing at additional security force personnel in the rear section of Truck 2. Fire will not be directed at truck fuel tanks. The fuel tanks may be hit from indirect small arms fire, but will not be considered as primary targets.

3.2.9 Aim Point Assumptions

Aim Points Used by Attack Force. Attackers employing RPGs use the center of the forward engine section as an aim point when attempting vehicle destruction. The center of the rear canvas-covered truck bed is the aim point when trying to hit weapons or security force personnel. Attackers employing automatic weapons aim at the center of defender targets whenever those targets can be visually acquired. Security force personnel located in the rear section of the security vehicle cannot be seen directly. These personnel are therefore attacked by spraying the rear canvas-covered truck sections randomly with automatic weapon fire. After each vehicle has stopped and defenders have assumed defensive positions behind their vehicle, all defenders are modeled as fixed aim points.

Aim Points Used by Defense Force. Attackers are assumed to be concealed in brush and surrounding foliage. Identification of the exact location of the Special Operations Team members is therefore difficult. Key signatures are used such as sound, intermittent muzzle flashes, smoke, and movement to identify regions in which attackers are located. Defenders direct automatic weapon fire into those regions which they have identified as the sources of fire being received. Attackers remain in preselected positions throughout the 20 second duration of the force-on-force attack sequence.

3.2.10 Defense Force Weapon Assumptions

Drivers in each vehicle are armed with M16 rifles. The firing rate is established as 175 rounds per minute for the 20 second encounter, and is based upon an equal combination of semi- and fully-automatic firing modes. The time required to exchange magazines was also incorporated into this average firing

rate. In addition to the drivers, two security guards in the rear section of Truck 2 are also armed with M16 rifles and are assigned the same firing rate.

Assistant drivers, or guards, and three security personnel in the rear section of Truck 2 are armed with M60 machine guns. A firing rate of 550 rounds per minute was used for this weapon. Selection of this firing rate was based on similar factors used to determine the firing rate used for the M16.

The remaining security force personnel in the rear section of Truck 2 are armed with the M79 grenade launcher. The M79 rate of fire is 15 rounds per minute. A lethal kill radius of 5 meters was established for each M79 round. Thus an attacker would be considered a casualty if the distance between the attacker and the point of detonation of the M79 round was less than or equal to 5 meters.

3.2.11 Attack Force Weapon Assumptions

Attackers 1 and 3 are armed with the AK47 rifle with an average firing rate of 200 rounds per minute. The remaining attackers are armed with an equivalent RPG weapon with firing rate of 5 rounds per minute. RPGs were assumed to be armor piercing weapons, effective for destruction of vehicles, should vehicles be hit in the forward engine section. RPGs were not modeled as having a lethal kill radius for personnel or weapons, because the canvas-covered rear section of the truck would not provide projectile detonation. Refer to Table 3.1 for a summary of weapon types and associated firing rates for weapons used for both the attack and defense forces in the simulation.

3.2.12 Attack and Defense Force Separation Distance Assumptions

For simplicity in simulation input, the defense force and attack force are assumed to reside in two parallel planes 75 meters apart. Thus, the minimum distance between and attacker and defender is 75 meters.

3.2.13 Aiming Error Assumptions

For purposes of this analysis, weapon accuracy was considered to be a

rate. In addition to the drivers, two security guards in the rear section of Truck 2 are also armed with M16 rifles and are assigned the same firing rate.

Assistant drivers, or guards, and three security personnel in the rear section of Truck 2 are armed with M60 machine guns. A firing rate of 550 rounds per minute was used for this weapon. Selection of this firing rate was based on similar factors used to determine the firing rate used for the M16.

The remaining security force personnel in the rear section of Truck 2 are armed with the M79 grenade launcher. The M79 rate of fire is 15 rounds per minute. A lethal kill radius of 5 meters was established for each M79 round. Thus an attacker would be considered a casualty if the distance between the attacker and the point of detonation of the M79 round was less than or equal to 5 meters.

3.2.11 Attack Force Weapon Assumptions

Attackers 1 and 3 are armed with the AK47 rifle with an average firing rate of 200 rounds per minute. The remaining attackers are armed with an equivalent RPG weapon with firing rate of 5 rounds per minute. RPGs were assumed to be armor piercing weapons, effective for destruction of vehicles, should vehicles be hit in the forward engine section. RPGs were not modeled as having a lethal kill radius for personnel or weapons, because the canvas-covered rear section of the truck would not provide projectile detonation. Refer to Table 3.1 for a summary of weapon types and associated firing rates for weapons used for both the attack and defense forces in the simulation.

3.2.12 Attack and Defense Force Separation Distance Assumptions

For simplicity in simulation input, the defense force and attack force are assumed to reside in two parallel planes 75 meters apart. Thus, the minimum distance between an attacker and defender is 75 meters.

3.2.13 Aiming Error Assumptions

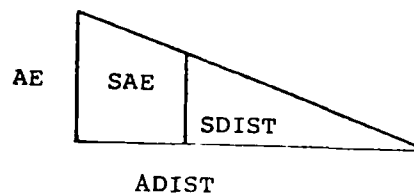
For purposes of this analysis, weapon accuracy was considered to be a

function of:

- separation distance between the target and firer,
- velocity of the target, and
- aiming errors inherent in the particular weapon employed.

Environmental factors, such as wind, were not considered. Aiming error inherent to the particular type of weapon employed was determined from references, or selected based on expert opinion.

Any change in separation distance between the firer and target was assumed to effect the aiming error as a ratio of the total separation distance over the standard separation distance. The calculation of the modified aiming error, assuming the separation distance is more than 75 meters, can be illustrated by the following geometric figure.



AE (Aiming Error) at distance ADIST is given by: $AE = SAE * ADIST / SDIST$.
where: SDIST - Standard Separation Distance. Distance between firer and target used as a baseline reference.

SAE - Standard Aiming Error. Radius in which 1/2 of all shots fired are expected to land, assuming a distance between firer and target of SDIST.

ADIST - Actual Distance between target and firer.

AE - Aiming Error assuming distance between firer and target is ADIST.

The standard aiming error is expressed as a distance in meters. This distance defines the radius of a circle in which one-half of all rounds fired

are expected to hit. The aiming error used for automatic weapons was 0.5 meters when the separation distance was 75 meters. Reference 1 indicates small variations in aiming error exist between the M16, AK47, and M60; however, ranges given for aiming error of these weapons significantly depends on the firer's training and environmental factors. The aiming error values for a separation distance of 75 meters were averaged for each automatic weapon type. The value of 0.5 meters was thus selected as representative.

Aiming error for the RPG and M79 were calculated as 1.12 meters and 15 meters, respectively, for a standard separation distance of 75 meters.

Due to the average target velocity of 20 KPH, an increase to the static aiming error by a factor of 1.5 was used. Table 3.1 summarizes the selected aiming errors associated with each weapon type for targets in a moving or static state.

Table 3.1 - WEAPON ASSUMPTIONS

<u>Weapon Type</u>	<u>Target Velocity (KPH)</u>	<u>CEP (M)</u>	<u>Firing Rate (R/Min)</u>
M16	0.	.5	175.
AK47	0.	.5	200.
M60	0.	.5	550.
50 Cal	0.	.5	550.
M79	0.	15.	15.
RPG	0.	1.12	5.
M19	0.	.5	450.
M16	20.	.75	175.
AK47	20.	.75	200.
M60	20.	.75	550.
50 Cal	20.	.75	550.
M79	20.	22.5	15.
RPG	20.	1.68	5.
M19	20.	.75	450.

3.3 INPUT DATA SUMMARY

Refer to Appendix A for a complete listing of input data used to simulate the scenario previously described. Standardized data forms which aid in organizing the large amount of simulation input data are also included in Appendix A. These forms help insure that all information required for the simulation has been obtained. After the forms have been completed, the model is executed and form information entered interactively in the same order that it was originally written on the forms. Thus, by using the standardized data forms, an analyst having a minimum amount of prior computer experience may be able to perform a SAS simulation.

3.4 SAS OUTPUT

Output from the SAS model consists of probabilities that a particular target will still exist after a fixed amount of time has elapsed from the beginning of the simulation. Figure 3.16 shows a typical SAS output listing for the analysis of the baseline ground convoy scenario described in detail earlier in this section. Across the top of the output listing is time elapsed in seconds from the beginning of the simulation. Output is printed each second for the first 20 seconds into the simulation. All targets are listed in the column nearest the left hand side of the page. The remaining information within the output listing gives the probability that each target exists after a given amount of time has elapsed. The following information is supplied to help correlate the targets appearing on the left hand column of the output listing with those mentioned in the report previously:

- D represents defender,
- A represents attacker,
- W represents weapon (critical component),
- K represents weapon casing and container,
- G represents gas tank or fuel tank,
- E represents engine,
- T represents truck.

The first number on the far left is the target number, and numbers

following characters denote the specific target. For example, target number 18, which is given by D8T2, represents Defender(D) 8 on Truck(T) 2.

Figure 3.17 (a and b) shows the conditional probability that a given number of targets within a particular group remain undestroyed with time. The probabilities for Group 4 (Figure 3.17b, Attackers) are the conditional probabilities that a specific number of attackers remain with time. For example, after 10 seconds into the simulation, the probability that one attacker has been killed is 0.458. Notice that probabilities sum to one for any column of probabilities. This is because all possibilities are represented (i.e., for any amount of elapsed time into the simulation, the probability that no attackers have been killed + the probability that one attacker has been killed +...+ the probability all six attackers have been killed must sum to 1.). The figure also gives the expected number of attackers killed with time. For example, after 20 seconds, approximately three (2.64) attackers have been killed. The model does not attempt to establish an attack or defense force quit criteria. Definition of when the engagement terminates is left to the analyst.

Graphs such as the one shown in Figures 3.18a and 3.18b may be constructed to show the expected number of targets which have been destroyed as the simulation progresses in time. This graph is intended only as an example of one method to visually display SAS results. The figures from the output listing do not necessarily correspond to values shown on the graph.

C:\SAS\DATA\TARGETS\SAS\TARGETS.DAT														
SAS TARGETS														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195
196	197	198	199	200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
256	257	258	259	260	261	262	263	264	265	266	267	268	269	270
271	272	273	274	275	276	277	278	279	280	281	282	283	284	285
286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315
316	317	318	319	320	321	322	323	324	325	326	327	328	329	330
331	332	333	334	335	336	337	338	339	340	341	342	343	344	345
346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390
391	392	393	394	395	396	397	398	399	400	401	402	403	404	405
406	407	408	409	410	411	412	413	414	415	416	417	418	419	420
421	422	423	424	425	426	427	428	429	430	431	432	433	434	435
436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465
466	467	468	469	470	471	472	473	474	475	476	477	478	479	480
481	482	483	484	485	486	487	488	489	490	491	492	493	494	495
496	497	498	499	500	501	502	503	504	505	506	507	508	509	510
511	512	513	514	515	516	517	518	519	520	521	522	523	524	525
526	527	528	529	530	531	532	533	534	535	536	537	538	539	540
541	542	543	544	545	546	547	548	549	550	551	552	553	554	555
556	557	558	559	560	561	562	563	564	565	566	567	568	569	570
571	572	573	574	575	576	577	578	579	580	581	582	583	584	585
586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615
616	617	618	619	620	621	622	623	624	625	626	627	628	629	630
631	632	633	634	635	636	637	638	639	640	641	642	643	644	645
646	647	648	649	650	651	652	653	654	655	656	657	658	659	660
661	662	663	664	665	666	667	668	669	670	671	672	673	674	675
676	677	678	679	680	681	682	683	684	685	686	687	688	689	690
691	692	693	694	695	696	697	698	699	700	701	702	703	704	705
706	707	708	709	710	711	712	713	714	715	716	717	718	719	720
721	722	723	724	725	726	727	728	729	730	731	732	733	734	735
736	737	738	739	740	741	742	743	744	745	746	747	748	749	750
751	752	753	754	755	756	757	758	759	760	761	762	763	764	765
766	767	768	769	770	771	772	773	774	775	776	777	778	779	780
781	782	783	784	785	786	787	788	789	790	791	792	793	794	795
796	797	798	799	800	801	802	803	804	805	806	807	808	809	810
811	812	813	814	815	816	817	818	819	820	821	822	823	824	825
826	827	828	829	830	831	832	833	834	835	836	837	838	839	840
841	842	843	844	845	846	847	848	849	850	851	852	853	854	855
856	857	858	859	860	861	862	863	864	865	866	867	868	869	870
871	872	873	874	875	876	877	878	879	880	881	882	883	884	885
886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915
916	917	918	919	920	921	922	923	924	925	926	927	928	929	930
931	932	933	934	935	936	937	938	939	940	941	942	943	944	945
946	947	948	949	950	951	952	953	954	955	956	957	958	959	960
961	962	963	964	965	966	967	968	969	970	971	972	973	974	975
976	977	978	979	980	981	982	983	984	985	986	987	988	989	990
991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005
1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020
1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035
1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050
1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065
1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080
1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095
1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110
1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125
1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140
1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155
1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170
1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185
1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200
1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230
1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245
1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260
1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275
1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290
1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305
1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320
1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335
1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350
1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365
1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380
1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395
1396	1397	1398	1399	1400	1401	14								

EXPECTED NUMBER OF TARGETS DESTROYED AFTER TIME ELAPSED:

	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
3.95	5.80	6.98	7.81	8.15	8.25	8.35	8.45	8.53	8.59	8.65	8.69	8.75	8.79	8.81	8.84	8.86	8.88	8.91	8.91	8.92

CONDITIONAL PROBABILITIES FOR GROUP #4, CONSISTING OF TARGETS:

	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
0	1.000	1.000	0.898	0.815	0.735	0.675	0.715	0.747	0.720	0.685	0.641	0.595	0.545	0.495	0.445	0.395	0.345	0.295	0.245	0.195
1	0.000	0.000	0.100	0.185	0.265	0.315	0.211	0.234	0.257	0.269	0.258	0.236	0.215	0.194	0.174	0.154	0.134	0.114	0.094	0.074
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

EXPECTED NUMBER OF TARGETS DESTROYED AFTER TIME ELAPSED:

	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00

CONDITIONAL PROBABILITIES FOR GROUP #5, CONSISTING OF TARGETS:

	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
1	1.000	1.000	0.998	0.995	0.992	0.989	0.986	0.983	0.980	0.977	0.974	0.971	0.968	0.965	0.962	0.959	0.956	0.953	0.950	0.947
2	0.000	0.000	0.002	0.005	0.008	0.011	0.014	0.017	0.020	0.023	0.026	0.029	0.032	0.035	0.038	0.041	0.044	0.047	0.050	0.053
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

EXPECTED NUMBER OF TARGETS DESTROYED AFTER TIME ELAPSED:

	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00

FIGURE 3.17b SAS Output Showing Expected Number of Targets Destroyed With Time

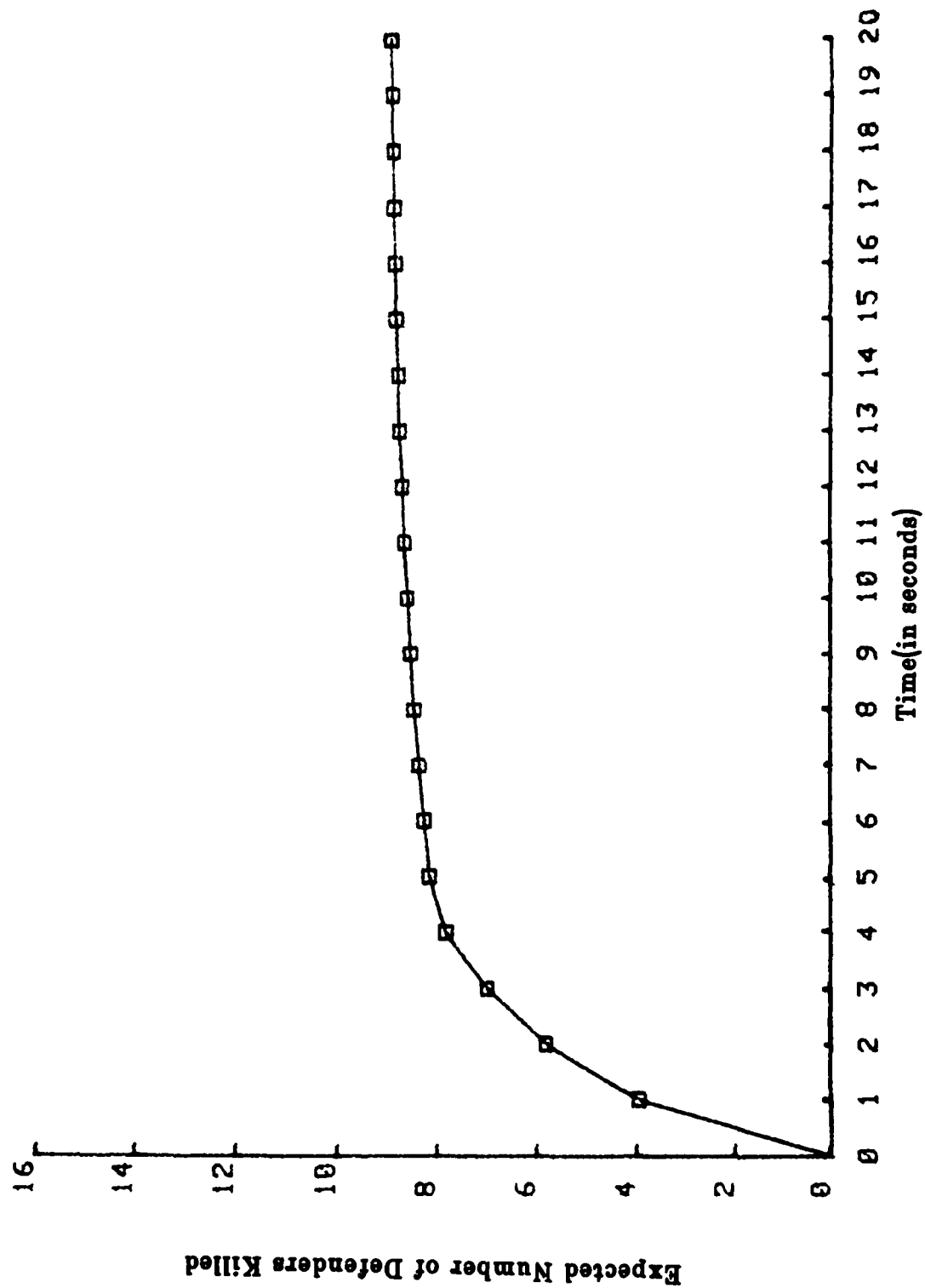


Figure 3.18a Plot of Expected Number of Targets Destroyed Versus Time From Beginning of Simulation

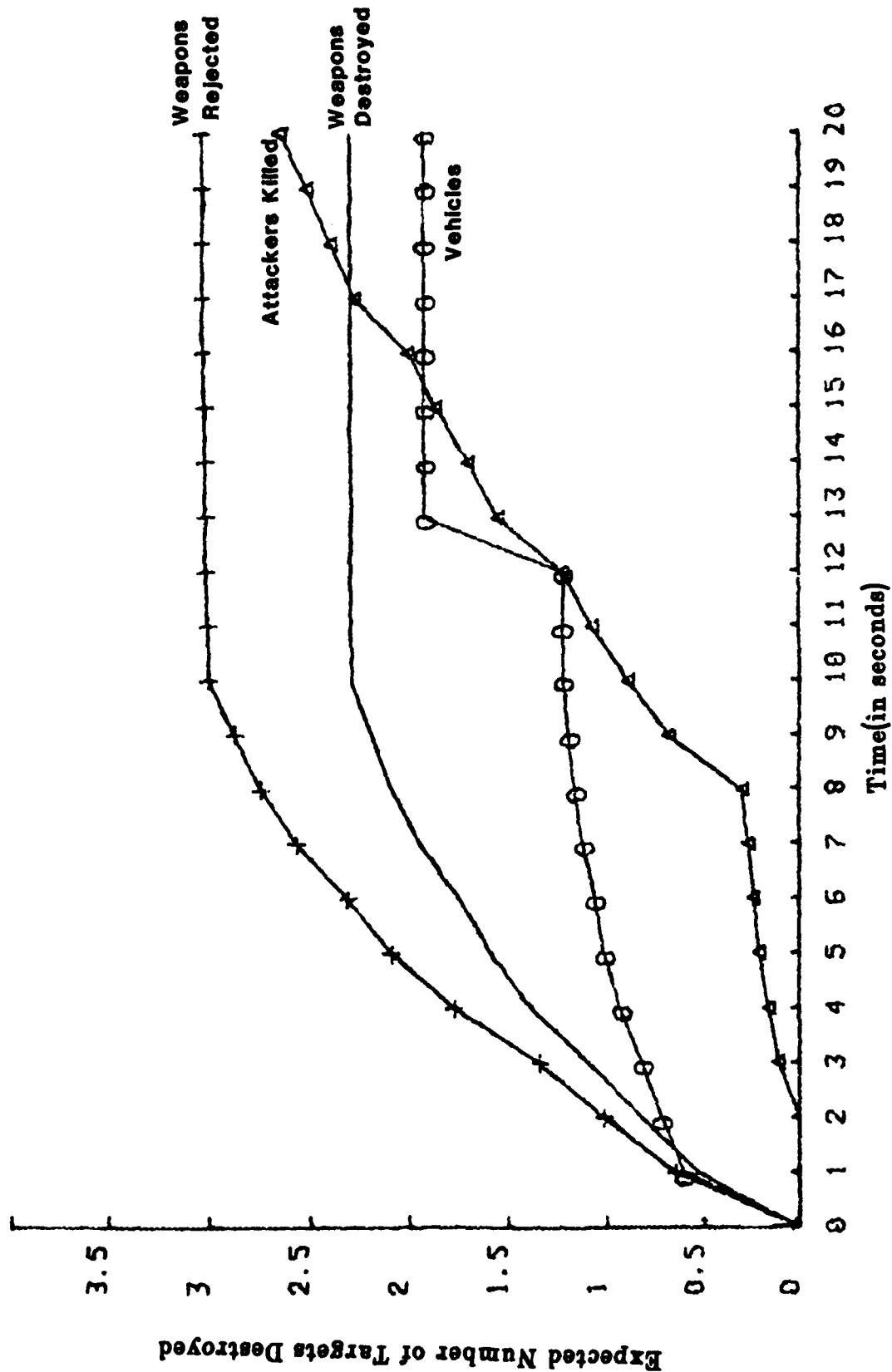


Figure 3.18b Plot of Expected Number of Targets Destroyed Versus Time From Beginning of Simulation

SECTION 4

EXAMPLE TERMINAL SESSION

```
*****
*           A typical user terminal session is presented to illustrate *
*           how SAS would be used to perform analysis of the example scenario *
*           presented in Chapter 4, Example Application. This information is *
*           intended to aid the user in understanding the SAS data entry pro- *
*           cedure. Comments, enclosed in asterisks, contain explanatory *
*           remarks. *
*           Following the normal log-on procedure, the user begins the *
*           terminal session which follows. *
*****
```

```
$RUN SAS
NAME OF INPUT DATA FILE?
INPUT.DAT
MODIFY TARGETS OF RUN? M-MODIFY, R-RUN
R
START TIME, END TIME FOR SIMULATION OUTPUT?
```

```
*****
*           Since this is the first SAS execution of this scenario, the *
*           data base files have not been initialized. In order to initialize *
*           these files prior to data entry, the user presses 'Z' while holding *
*           the CTRL key down(denoted by CTRL Z). After data base files have *
*           been initialized, this question is again asked and at that point the *
*           user will enter the starting and ending simulation times. *
*****
```

```
CTRL Z
1-INITIALIZE PROBABILITY DATA BASE
2-INITIALIZE TARGET DATA BASE
3-INITIALIZE AIM POINT DATA BASE
4-EXAMINE PROB. DATA BASE VALUE
5-MODIFY PROB. DATA BASE VALUE
1
NAME OF NEW PROBABILITY DATA BASE FILE?
PROB.DAT
1-INITIALIZE PROBABILITY DATA BASE
2-INITIALIZE TARGET DATA BASE
3-INITIALIZE AIM POINT DATA BASE
4-EXAMINE PROB. DATA BASE VALUE
5-MODIFY PROB. DATA BASE VALUE
2
NAME OF NEW TARGET DATA BASE FILE?
TARGET.DAT
1-INITIALIZE PROBABILITY DATA BASE
2-INITIALIZE TARGET DATA BASE
3-INITIALIZE AIM POINT DATA BASE
4-EXAMINE PROB. DATA BASE VALUE
5-MODIFY PROB. DATA BASE VALUE
3
NAME OF NEW AIM POINT DATA BASE FILE?
AIM.DAT
```

1-INITIALIZE PROBABILITY DATA BASE
 2-INITIALIZE TARGET DATA BASE
 3-INITIALIZE AIM POINT DATA BASE
 4-EXAMINE PROB. DATA BASE VALUE
 5-MODIFY PROB. DATA BASE VALUE

 * At this point the user has initialized all of the data files *
 * required for SAS execution. The user presses "Z" while holding down *
 * the CTRL key to end data base initialization. *

CTRL Z
 START TIME, END TIME FOR SIMULATION OUTPUT?
 1.0 20.0
 TIME INTERVAL AT WHICH RESULTS ARE DISPLAYED?
 1.0
 NUMBER OF HARDWARE OPTION TITLES?
 3
 TITLE 1?
 SGT ATTACK ON GROUND CONVOY
 TITLE 2?
 STATE: TRANSITION
 TITLE 3?
 BASELINE
 NAME OF FILE TO CONTAIN RESULTS?
 OUTPUT.DAT
 NAME OF PROBABILITY DATA BASE FILE?
 PROB.DAT
 TARGET DATA FILE NAME?
 TARGET.DAT
 ENTER TARGET INFORMATION.
 TARGET NUMBER?
 1
 TARGET DESCRIPTION?
 D1T1
 TARGET HEIGHT, WIDTH?
 1.2 .75
 RANDOM OR FIXED TARGET AREA? F-FIXED, R-RANDOM
 F
 TARGET AREA, LOWER: X COOR., Y COOR.?
 146. .1
 CAN TARGET FIRE? Y-YES, N-NO
 Y
 WEAPON TYPE?
 15
 NUMBER OF ROUNDS AVAILABLE?
 360
 RESPONSE TIME?
 7.
 AIM POINTS(ORDERED BY PRIORITY)?
 19 22 17 14 20 21 0 0 0 0
 TARGETS DAMAGED BY HITTING THIS TARGET?
 0 0 0 0 0 0 0 0 0 0
 TARGET NUMBER?

2
 TARGET DESCRIPTION?
 D211
 TARGET HEIGHT, WIDTH?
 1.2 .75
 RANDOM OR FIXED TARGET AREA? F-FIXED, R-RANDOM
 F
 TARGET AREA, LOWER: X COOR., Y COOR.?
 146. .1
 CAN TARGET FIRE? Y-YES, N-NO
 Y
 WEAPON TYPE?
 17
 NUMBER OF ROUNDS AVAILABLE?
 3000
 RESPONSE TIME?
 2.
 AIM POINTS(ORDERED BY PRIORITY)?
 19 22 17 16 20 21 0 0 0 0
 TARGETS DAMAGED BY HITTING THIS TARGET?
 0 0 0 0 0 0 0 0 0 0

 * To avoid unnecessary repetition, all other data contained in *
 * the target data forms(Appendix A, Figure A2) for those targets in *
 * their initial positions is entered in a similar fashion. We will now *
 * continue with entry of data for the last target in it's initial *
 * position. *

TARGET NUMBER?
 46
 TARGET DESCRIPTION?
 A6
 TARGET HEIGHT, WIDTH?
 .125 .125
 RANDOM OR FIXED TARGET AREA? F-FIXED, R-RANDOM
 F
 TARGET AREA, LOWER: X COOR., Y COOR.?
 109. 0.
 CAN TARGET FIRE? Y-YES, N-NO
 Y
 WEAPON TYPE?
 16
 NUMBER OF ROUNDS AVAILABLE?
 1000
 RESPONSE TIME?
 0.
 AIM POINTS(ORDERED BY PRIORITY)?
 14 10 13 9 15 11 7 3 0 0
 TARGETS DAMAGED BY HITTING THIS TARGET?
 0 0 0 0 0 0 0 0 0 0
 TARGET NUMBER?

```

*****
*           All target information on the target data forms has been      *
*           entered for targets in their initial positions. The user now types *
*           a '2' while depressing the CTRL key to end the target data input  *
*           section.                                                         *
*****

```

```

CTRL Z
AIM POINT DATA FILE NAME?
AIM.DAT
ENTER AIM POINT INFORMATION.
AIM POINT NUMBER?
1
AIM POINT DESCRIPTION?
D111
RANDOM OR FIXED AIM POINT? F-FIXED, R-RANDOM
F
COORDINATES OF AIM POINT X, Y?
146.375 .7
TARGETS WHICH MAY BE HIT BY FIRING AT THIS AIM POINT?
1 2 5 8 9 0 0 0 0 0 0 0 0 0 0
AIM POINT NUMBER?
2
AIM POINT DESCRIPTION?
D211
RANDOM OR FIXED AIM POINT? F-FIXED, R-RANDOM
F
COORDINATES OF AIM POINT X, Y?
146.375 .7
TARGETS WHICH MAY BE HIT BY FIRING AT THIS AIM POINT?
2 1 5 8 9 0 0 0 0 0 0 0 0 0 0

```

```

*****
*           Again to avoid undue repetition, only the last aim point will*
*           now be entered. All other aim points which were not entered pre- *
*           viously are entered in a similar manner.                         *
*****

```

```

AIM POINT NUMBER?
4*
AIM POINT DESCRIPTION?
E14P2
RANDOM OR FIXED AIM POINT? F-FIXED, R-RANDOM
F
COORDINATES OF AIM POINT X, Y?
116.2 .55
TARGETS WHICH MAY BE HIT BY FIRING AT THIS AIM POINT?
40 31 32 33 34 35 36 37 38 39 0 0 0 0 0 0

```

```

*****
*           To end the aim point data gathering section the user types      *
*           '7' while depressing the CTRL key.                               *
*****

```

```

CTRL Z
ENTER STATUS CHANGE INFORMATION.

```

```

*****
*           Status change information includes both the target data for *
*           all targets in their non-initial positions and the status change time*
*           and target numbers in the input data forms.
*****

```

```

TARGET NUMBER?
47
TARGET DESCRIPTION?
01T1P2
TARGET HEIGHT, WIDTH?
.15 .25
RANDOM OR FIXED TARGET AREA? F=FIXED, R=RANDOM
F
TARGET AREA, LOWER: X COOR., Y COOR.?
148. -.5
CAN TARGET FIRE? Y=YES, N=NO
Y
WEAPON TYPE?
4
NUMBER OF ROUNDS AVAILABLE?
360

```

```

*****
*           Notice that RESPONSE TIME is no longer required for targets *
*           in their non-initial positions. In place of response time, an *
*           initial delay is requested later in the interactive terminal session.*
*           This delay is not relative to the beginning of the simulation, as is *
*           response time, rather it is relative to the time at which the target *
*           status change occurs.
*****

```

```

AIX POINTS(ORDERED BY PRIORITY)?
19 22 17 18 20 21 0 0 0 0
TARGETS DAMAGED BY HITTING THIS TARGET?
0 0 0 0 0 0 0 0 0 0
STATUS CHANGE TIME, INITIAL DELAY, OLD TARGET NUMBER?
5. 0. 1
TARGET NUMBER?
48
TARGET DESCRIPTION?
02T1P2
TARGET HEIGHT, WIDTH?
.15 .25
RANDOM OR FIXED TARGET AREA? F=FIXED, R=RANDOM
F
TARGET AREA, LOWER: X COOR., Y COOR.?
155.750 -0.5
CAN TARGET FIRE? Y=YES, N=NO
Y
WEAPON TYPE?
6
NUMBER OF ROUNDS AVAILABLE?
3000

```

AIM POINTS(ORDERED BY PRIORITY)?
 19 22 17 18 20 21 0 0 0 0
 TARGETS DAMAGED BY HITTING THIS TARGET?
 0 0 0 0 0 0 0 0 0 0
 STATUS CHANGE TIME, INITIAL DELAY, OLD TARGET NUMBER?
 5. 0. 2

 * To avoid unnecessary repetition, only the data for the last *
 * non-initial target will be given. *

TARGET DESCRIPTION?
 A6P2
 TARGET HEIGHT, WIDTH?
 .125 .125
 RANDOM OR FIXED TARGET AREA? F=FIXED, R-RANDOM
 F
 TARGET AREA, LOWER: X COOR., Y COOR.?
 109. 0.
 CAN TARGET FIRE? Y=YES, N=NO
 Y
 WEAPON TYPE?
 5

NUMBER OF ROUNDS AVAILABLE?
 1000
 AIM POINTS(ORDERED BY PRIORITY)?
 42 38 41 37 29 30 31 32 33 34
 TARGETS DAMAGED BY HITTING THIS TARGET?
 0 0 0 0 0 0 0 0 0 0
 STATUS CHANGE TIME, INITIAL DELAY, OLD TARGET NUMBER?
 10. 0. 46

 * After all status change information has been entered, type *
 * "Z" while holding down the CTRL key. *

CTRL Z
 NUMBER OF TARGETS IN GROUP?
 9
 TARGETS FOR WHICH CONDITIONAL PROBABILITY SHOULD BE DISPLAYED?
 49 50 51 69 70 71 79 80 81

 * The first target group corresponds to all weapon targets. *
 * Notice that the final target numbers are used when specifying the tar- *
 * get group, not the initial numbers. *

NUMBER OF TARGETS IN GROUP?
 9
 TARGETS FOR WHICH CONDITIONAL PROBABILITY SHOULD BE DISPLAYED?
 52 53 54 72 73 74 82 83 84

```

*****
*           The second group specifies all weapon containers which if hit*
*           would result in a possible weapon rejection.           *
*****

```

NUMBER OF TARGETS IN GROUP?

14

TARGETS FOR WHICH CONDITIONAL PROBABILITY SHOULD BE DISPLAYED?

47 48 57 58 59 60 61 62 63 64 67 68 77 78

```

*****
*           The third group of targets correspond to the set of defend- *
*           ders.                                           *
*****

```

NUMBER OF TARGETS IN GROUP?

6

TARGETS FOR WHICH CONDITIONAL PROBABILITY SHOULD BE DISPLAYED?

87 88 89 90 91 92

```

*****
*           The fourth group corresponds to all attacker targets.      *
*****

```

NUMBER OF TARGETS IN GROUP?

4

TARGETS FOR WHICH CONDITIONAL PROBABILITY SHOULD BE DISPLAYED?

56 66 76 86

```

*****
*           The fifth group contains all vehicle engines.              *
*****

```

```

*****
*           The user has now specified all target groups he wishes to  *
*           consider for this simulation. It so happens that in this case the *
*           maximum number of groups(5 groups) was desired. When the maximum *
*           number of groups have been chosen, the model automatically proceeds *
*           to the next data item. Since the target groups are the last data *
*           item to enter before the simulation is run, this is all the user has *
*           to input for this example. Should you desire fewer target groups, *
*           this input loop may be terminated by entering 'Z' while holding the *
*           CTRL key down.                                           *
*****

```

```

*****
*           The simulation is complete when a 's' prompt is returned to *
*           the user's terminal. Simulation output can be found in the output *
*           file specified by the user(in this case the file OUTPUT.DAT was used.*
*           The user may type the file to his screen by typing TYPE OUTPUT.DAT or *
*           may list the contents of the file on the line printer by typing *
*           PRINT OUTPUT.DAT after the 's' prompt.                  *
*           A sample of the output for this example is presented in Figure 1. *
*****

```

REFERENCES

1. G. E. P. Box and M. E. Muller, "A Note on the Generation of Random Normal Deviates", *Analytical Mathematical Statistics*, Volume 29, 1958, pp. 610-611.
2. G. S. Fishman, "Concepts and Methods in Discrete Event Digital Simulation", Chapter 8, pp. 211-213, John Wiley & Sons, New York, 1973.
3. C. M. Clark and M. N. Cravens, "Path Analysis (PANL) User's Guide", Sandia Laboratories, SAND80-1888.
4. R. McGowan and D. Packard, "Survivability and Security Assessment of Movement-of-Weapons Issue, Volume I-Analysis, JAYCOR, unpublished.

APPENDIX A

STANDARDIZED INPUT DATA FORMS

1 Input information required for SAS execution is classified into three areas: target data, aim point data, and other input data. Other input data consists of all data which is not related to target or aim point data directly. The target data, aim point data and input data files will be used to store all required input for a SAS simulation. To facilitate obtaining and entering the large amount of data required for a typical SAS execution, standard data forms have been specifically designed to obtain the required data for each of the three information categories. Examples of blank data forms to gather necessary target data, aim point data and other input data are presented in Figures A.1 (a through e). The input data forms are useful in organizing input data and assuring the analyst that all necessary data has been obtained prior to execution of the SAS simulation. After data forms have been filled out and the SAS model is executed, the data on the forms is entered interactively in the same order which it is filled in on the form.

The information which is required for each data form is contained in completed data forms for this particular example, which are presented separately in Figure A.2 (a through q).

INPUT DATA FORM, page 1

Name of input data file (10 chars. max.)? _____
Start, end time for simulation output? _____
Time interval at which results are displayed? _____
Number of hardware option titles (1-5)? _____
Hardware option title 1? _____
Hardware option title 2 (if no more titles, leave rest of option titles blank)? _____
Hardware option title 3 (if no more titles, leave rest of option titles blank)? _____
Hardware option title 4 (if no more titles, leave rest of option titles blank)? _____
Hardware option title 5 (if no more titles, leave rest of option titles blank)? _____
Name of file to contain results (10 chars. max.)? _____
Name of probability data base file (10 chars. max.)? _____
Target data file name (10 chars. max.)? _____
(File out Target Data Form next)

Figure A.1a Data Forms Used to Acquire SAS Input Data (Input Data Form, page 1)

INPUT DATA FORM, page 3

Number of targets in Group 1? _____
Target numbers in Group 1? _____
Number of targets in Group 2? _____
Target numbers in Group 2? _____
Number of targets in Group 3? _____
Target numbers in Group 3? _____
Number of targets in Group 4? _____
Target numbers in Group 4? _____
Number of targets in Group 5? _____
Target numbers in Group 5? _____
Number of targets in Group 6? _____
Target numbers in Group 6? _____
Number of targets in Group 7? _____
Target numbers in Group 7? _____
Number of targets in Group 8? _____
Target numbers in Group 8? _____
Number of targets in Group 9? _____
Target numbers in Group 9? _____
Number of targets in Group 10? _____
Target numbers in Group 10? _____

When all target groups have been entered, all data required for the simulation is complete.

Figure A.1e Data Forms Used to Acquire SAS Input Data (Input Data Form, page 3)

INPUT DATA FORM, page 1

Name of input data file (10 chars. max.)? INPUT.DAT

Start, end time for simulation output? 1.0 20.0

Time interval at which results are displayed? 1.0

Number of hardware option titles (1-5)? 2

Hardware option title 1? SOT ATTACK ON GROUND CONVOY

Hardware option title 2 (if no more titles, leave rest of option titles blank)?
STATE: TRANSITION

Hardware option title 3 (if no more titles, leave rest of option titles blank)?

Hardware option title 4 (if no more titles, leave rest of option titles blank)?

Hardware option title 5 (if no more titles, leave rest of option titles blank)?

Name of file to contain results (10 chars. max.)? OUTPUT.DAT

Name of probability data base file (10 chars. max.)? PROB.DAT

Target data file name (10 chars. max.)? TARGET.DAT

(File out Target Data Form next)

Figure A.2a Completed Input Data Form, Page 1

Target Number	Target Description	Target		Random or Fixed	Lower Left Coord.		Random Area		Can Target fire? (Y or N)	Weapon Type	Number of Rounds	Response Time	Aim Points, Ordered by Priority										Collaterally Damaged Targets																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
		Height	Width		X	Y	Height	Width ²																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
													1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
1	D1T1	1.2	.75	F	146.	0.1			Y	15	360	7.	19	22	17	18	20	21																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												

Notes:

1 - If random, X and Y are lower left corner coordinates of random target area
2 - height and width for the random target area are not used if target is fixed

After all target information has been entered, fill out aimpoint data form, otherwise continue entering target data information on new target data form.

Figure A.2b Completed Target Data Form, Targets 1-13

Target Number	Target Description	Target Height Width		Random or Fixed	Lower Left Coord.		Random Area Height Width ²		Can target fire? (Y or N)	Weapon Type of Rounds	Response Time	Aim Points, Ordered by Priority										Collaterally Damaged Targets																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
		Height	Width		X	Y	Height	Width ²				1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
14	D4T2	1.2	.75	F	112.8	.5			Y	17	3000	8.67	19	22	18	17	20	21																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

Notes:

1 - If random, X and Y are lower left corner coordinates of random target area
 2 - height and width for the random target area are not used if target is fixed

After all target information has been entered, fill out aimpoint data form, otherwise continue entering target data information on new target data form.

Figure A.2c Completed Target Data Form, Targets 14-26

Target Number	Target Description	Target		Random or Fixed	Lower Left Coord.		Random Area Height/Width ²		Can target fire? (Y or N)	Weapon Type	Number of Rounds	Response Time	Aim Points, Ordered by Priority										Collaterally Damaged Targets																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
		Height	Width		X	Y	Height	Width					1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
27	K2T3	.5	1.5	F	85.15	0.			N																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

Notes:

1 - if random, X and Y are lower left corner coordinates of random target area
2 - height and width for the random target area are not used if target is fixed

After all target information has been entered, fill out aimpoint data form, otherwise continue entering target data information on new target data form.

Figure A.2d Completed Target Data Form, Targets 27-39

Target Number	Target Description	Target		Random or Fixed	Lower Left Coord.		Random Area		Can Target fire? (Y or N)	Weapon Type	Number of Rounds	Response Time	Aim Points, Ordered by Priority										Collaterally Damaged Targets																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
		Height	Width		X	Y	Height	Width					1 2 3 4 5 6 7 8 9 10										1 2 3 4 5 6 7 8 9 10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

Notes:

1 - If random, X and Y are lower left corner coordinates of random target area
2 - height and width for the random target area are not used if target is fixed

After all target information has been entered, fill out aimpoint data form, otherwise continue entering target data information on new target data form.

Figure A.2e Completed Target Data Form, Targets 40-52

Target Number	Target Description	Target		Random or Fixed	Lower Left Coord. 1		Random Area Height Width ²	Can Target fire? (Y or N)	Weapon Type of Rounds	Response Time	Aim Points, Ordered by Priority										Collaterally Damaged Targets																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
		Height	Width		Coord. 1						1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
					X	Y																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
53	K2TIP2	.5	1.5	F	149.6	0.		N																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

Notes:
1 - if random, X and Y are lower left corner coordinates of random target area
2 - height and width for the random target area are not used if target is fixed

After all target information has been entered, fill out aimpoint data form, otherwise continue entering target data information on new target data form.

Figure A.2f Completed Target Data Form, Targets 53-65

Target Number	Target Description	Target Height	Target Width	Random or Fixed	Lower Left Corner		Random Area Height	Can Target (Y or N)	Weapon Type	Number of Rounds	Response Time	Aim Points, Ordered by Priority										Collaterally Damaged Targets									
					X	Y						1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
66	E2P2	1.1	1.6	F	141.4	0.		N																							
67	D1P2	.15	.25	F	129.7	-0.8		Y	4	360	10.33	22	19	20	18	21	17														
68	D2P2	1.2	.75	F	127.2	.1		Y	6	3000	2.	22	19	20	18	21	17														
69	W1P2	.15	.7	F	122.6	.175		N														67	68	70	71	72	73	74	75	76	
70	W2P2	.15	.7	F	124.	.175		N														67	68	69	71	72	73	74	75	76	
71	W3P2	.15	.7	F	125.4	.175		N														67	68	69	70	72	73	74	75	76	
72	K1P2	.5	1.5	F	122.0	0.		N																							
73	K2P2	.5	1.5	F	123.6	0.		N																							
74	K3P2	.5	1.5	F	125.2	0.		N																							
75	G1P2	.4	1.4	F	125.4	-0.4		N														67	68	69	70	71	72	73	74	76	
76	E1P2	1.1	1.6	F	128.4	0.		N																							
77	D1P2	.15	.25	F	116.7	-0.8		Y	4	360	12.	22	19	21	20	18	17														
78	D2P2	1.2	.75	F	114.2	.1		Y	6	3000	2.	22	19	21	20	18	17														

Notes:

1 - if random, X and Y are lower left corner coordinates of random target area
2 - height and width for the random target area are not used if target is fixed

After all target information has been entered, fill out aimpoint data form, otherwise continue entering target data information on new target data form.

Figure A.2g Completed Target Data Form, Targets 66-78

Target Number	Target Description	Target		Random or Fixed	Lower Left Coord.		Random Area		Can target fire? (Y or N)	Weapon Type	Number of Rounds	Response Time	Aim Points, Ordered by Priority										Collaterally Damaged Targets																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
		Height	Width		X	Y	Height	Width ²					1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
79	W1T4P2	.15	.7	F	109.6	.175			N																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		

Notes:

1 - If random, X and Y are lower left corner coordinates of random target area
2 - height and width for the random target area are not used if target is fixed

After all target information has been entered, fill out aimpoint data form, otherwise continue entering target data information on new target data form.

Figure A.2h Completed Target Data Form, Targets 79-91

Aim Point Number	Aim Point Description	Random or Fixed	X Coord.	Y Coord.	Height	Width	Targets which may be hit by firing at this aim point									
							1	2	3	4	5	6	7	8	9	10
1	D1T1	F	146.375	0.7			1	2	5	8	9					
2	D2T1	F	146.375	0.7			2	1	5	8	9					
3	CT1	R	140.75	0.	3.	4.8	9	1	2	3	4	5	6	7	8	
4	ET1	F	147.95	.55			10	1	2	3	4	5	6	7	8	9
5	D1T2	F	117.625	0.7			11	12	17	18	19					
6	D2T2	F	117.625	0.7			12	11	17	18	19					
7	CT2	R	112.	0.	3.	4.8	16	11	12	13	14	15	17	18	19	
8	ET2	F	119.2	.55			20	11	12	13	14	15	16	17	18	19
9	D1T3	F	89.125	0.7			21	22	25	28	29					
10	D2T3	F	89.125	0.7			22	21	25	28	29					
11	CT3	R	83.5	0.	3.	4.8	29	21	22	23	24	25	26	27	28	
12	ET3	F	90.7	.55			30	21	22	23	24	25	26	27	28	29
13	D1T4	F	60.375	0.7			31	32	35	38	39					

Notes:

1 - If random, X and Y are coordinates of lower left corner of random firing area

2 - height and width specify the height and width of random aim point area

After all aim points have been entered, fill out Input Data Form - Page 2, otherwise continue entering aim point data information on new aim point data form.

Figure A.2j Completed Aim Point Data Form, Aim Points 1-13

Aim Point Number	Aim Point Description	Random or Fixed	X Coord.	Y Coord.	Height 1	Width 2	Targets which may be hit by firing at this aim point									
							1	2	3	4	5	6	7	8	9	10
14	D2T4	F	60.375	0.7			32	31	35	38	39					
15	CT4	R	54.75	0.	3.	4.8	39	31	32	33	34	35	36	37	38	
16	ET4	F	61.95	.55			40	31	32	33	34	35	36	37	38	39
17	A1	R	153.763	-0.438	1.	3.	41	42	43	44	45	46				
18	A2	R	140.763	-0.438	1.	3.	42	41	43	44	45	46				
19	A3	R	133.563	-0.438	1.	3.	43	41	42	44	45	46				
20	A4	R	127.963	-0.438	1.	3.	44	41	42	43	45	46				
21	A5	R	114.763	-0.438	1.	3.	45	41	42	43	44	46				
22	A6	R	107.563	-0.438	1.	3.	46	41	42	43	44	45				
23	D1T1P2	F	148.125	-0.725			1	3	6	7						
24	D2T1P2	F	155.875	-0.725			2									
25	CT1P2	R	148.	0.	3.	4.8	9	1	3	4	5	6	7	8		
26	ET1P2	F	155.2	.55			10	1	2	3	4	5	6	7	8	9

Notes:

1 - if random, X and Y are coordinates of lower left corner of random firing area

2 - height and width specify the height and width of random aim point area

After all aim points have been entered, fill out Input Data Form - Page 2, otherwise continue entering aim point data information on new aim point data form.

Figure A.2k Completed Aim Point Data Form, Aim Points 14-26

Aim Point Number	Aim Point Description	Random or Fixed	X Cbdr.	Y Coord.	Height	Width ²	Targets which may be hit by firing at this aim point									
							1	2	3	4	5	6	7	8	9	10
27	D1T2P2	F	142.875	-0.725			11	18								
28	D2T2P2	F	140.375	0.7			12	16	17	18	19					
29	D3T2P2	F	135.125	-0.725			13	14								
30	D4T2P2	F	136.415	-0.725			14	13	15	16						
31	D5T2P2	F	137.705	-0.725			15	14	16	19						
32	D6T2P2	F	138.755	-0.725			16	15	17	19						
33	D7T2P2	F	140.295	-0.725			17	16	18	19						
34	D8T2P2	F	141.585	-0.725			18	17	19							
35	CT2P2	R	135.	0.	3.	4.8	15	12	13	14	16	17	18	19		
36	ET2P2	F	142.2	.55			20	11	12	13	14	15	16	17	18	19
37	D1T3P2	F	129.875	-0.725			21									
38	D2T3P2	F	127.625	0.7			22	25	28	29						
39	CT3P2	F	122.	0.	3.	4.8	29	22	23	24	25	26	27	28		

Notes:

1 - If random, X and Y are coordinates of lower left corner of random firing area

2 - height and width. specify the height and width of random aim point area

After all aim points have been entered, fill out Input Data Form - Page 2, otherwise continue entering aim point data information on new aim point data form.

Figure A.21 Completed Aim Point Data Form, Aim Points 27-39

AD-A116 922

JAYCOR ALBUQUERQUE NM

F/G 15/7

USER'S GUIDE FOR THE SAS (STAND-OFF ATTACK SIMULATION) COMPUTER--ETC(U)

JAN 82 C M CLARK, J T HUMPHREY, L W KENNEDY

DNA001-79-C-0463

UNCLASSIFIED

DNA-5845T

NL

2 of 2

140 2

16900



END

DATE

FILED

8-82

DTIC

Old Target Number	New Target Number	Status Change Time	Delay Time
1	47	5.0	0.
2	48	5.0	0.
3	49	5.0	0.
4	50	5.0	0.
5	51	5.0	0.
6	52	5.0	0.
7	53	5.0	0.
8	54	5.0	0.
9	55	5.0	0.
10	56	5.0	0.
11	57	6.67	0.
12	58	6.67	0.
13	59	6.67	0.
14	60	6.67	0.
15	61	6.67	0.
16	62	6.67	0.
17	63	6.67	0.
18	64	6.67	0.
19	65	6.67	0.

If all status changes have been entered, fill out Input Data Form - Page 3, otherwise continue entering status changes on new Input Data Form - Page 2.

Figure A.2n Completed Input Data Form, Page 2, Targets 1-19

Old Target Number	New Target Number	Status Change Time	Delay Time
20	66	6.67	0.
21	67	8.33	0.
22	68	8.33	0.
23	69	8.33	0.
24	70	8.33	0.
25	71	8.33	0.
26	72	8.33	0.
27	73	8.33	0.
28	74	8.33	0.
29	75	8.33	0.
30	76	8.33	0.
31	77	10.0	0.
32	78	10.0	0.
33	79	10.0	0.
34	80	10.0	0.
35	81	10.0	0.
36	82	10.0	0.
37	83	10.0	0.
38	84	10.0	0.

If all status changes have been entered, fill out Input Data Form - Page 3, otherwise continue entering status changes on new Input Data Form - Page 2.

Figure A.2o Completed Input Data Form, Page 2, Targets 20-38

INPUT DATA FORM, page 3

Number of targets in Group 1? 9

Target numbers in Group 1? 49 50 51 69 70 71 79 80 81

Number of targets in Group 2? 9

Target numbers in Group 2? 52 53 54 72 73 74 82 83 84

Number of targets in Group 3? 14

Target numbers in Group 3? 47 48 57 58 59 60 61 62 63 64 67 68 77 78

Number of targets in Group 4? 6

Target numbers in Group 4? 87 88 89 90 91 92

Number of targets in Group 5? 4

Target numbers in Group 5? 56 66 76 86

Number of targets in Group 6? _____

Target numbers in Group 6? _____

Number of targets in Group 7? _____

Target numbers in Group 7? _____

Number of targets in Group 8? _____

Target numbers in Group 8? _____

Number of targets in Group 9? _____

Target numbers in Group 9? _____

Number of targets in Group 10? _____

Target numbers in Group 10? _____

Figure A.2q Completed Input Data Form, Page 3

APPENDIX B

GENERAL MODEL INFORMATION

B.1 SAS MODEL FLOW STRUCTURE

The SAS computer model consists of a main executive routine and five subroutines. The major function of the executive routine is to modify the probability of target existence array and schedule the next event to occur as the simulation progresses. The main routine directly communicates with three of the five subroutines. Subroutine GINPUT is the first routine called by the main program. GINPUT is called to obtain all necessary input required to perform the simulation. This routine also insures that probabilities of hit have been computed for all target/aim point/firer combinations when targets are in their initial positions.

As the simulation proceeds, requests to modify the status of certain targets will occur. Subroutine STATCHG is called to insure the parameter data describing the new target status has been stored and probabilities of hit are computed for the target whose status has been modified.

After the simulation has finished, subroutine CONDPRO is executed to evaluate the expected number of targets destroyed and conditional probability that a fixed number of targets remain with time.

Subroutine GETPTH is called from both subroutines GINPUT and TAREXEC. Subroutine GETPTH determines the probability of hit for a specific target, firer, and aim point combination.

Subroutine TAREXEC is called from subroutine STATCHG. Subroutine TAREXEC is used to handle data base storage and retrieval of information required to perform the target status change. Once the proper information describing the new target status has been obtained and stored, subroutine GETPTH is called to evaluate the probability of hit for the status change target.

The general flow structure of routine calls within the SAS model is summarized in Figure B.1.

B.2 STORAGE REQUIREMENTS

SAS creates a direct access data base file used to store probability of hit for each combination of target, aim point, and firer. Disk storage requirements used to store the SAS program source code, object code, and data base file are approximately 20K blocks, where each block contains 512 bytes or characters. The amount of storage required depends on the complexity of the simulation. The estimate of 20K blocks is based on a simulation with a comparable level of detail as the example presented in Section 3.

B.3 SYSTEM CONFIGURATION

The SAS model was developed on a VAX 11/780 computer system utilizing the following peripherals:

Memory: 768K bytes Extended Core Storage (ECC)
MOS memory (with floating point accelerator)

Disk Drives: RMO3-67 megabyte hard disk drive

Printer: LPO5-600 lines per minute

Terminal: VT-100

The operating system used while running the SAS simulation model was the VAX/VMS virtual memory operating system.

B.4 SOURCE LANGUAGE

The SAS computer model is written entirely in extended VAX FORTRAN IV. Although an extended type FORTRAN was used, an attempt was made to minimize the use of instructions which would not be available in ANSI standard FORTRAN, thus allowing the code to be transferred to another type computer more quickly.

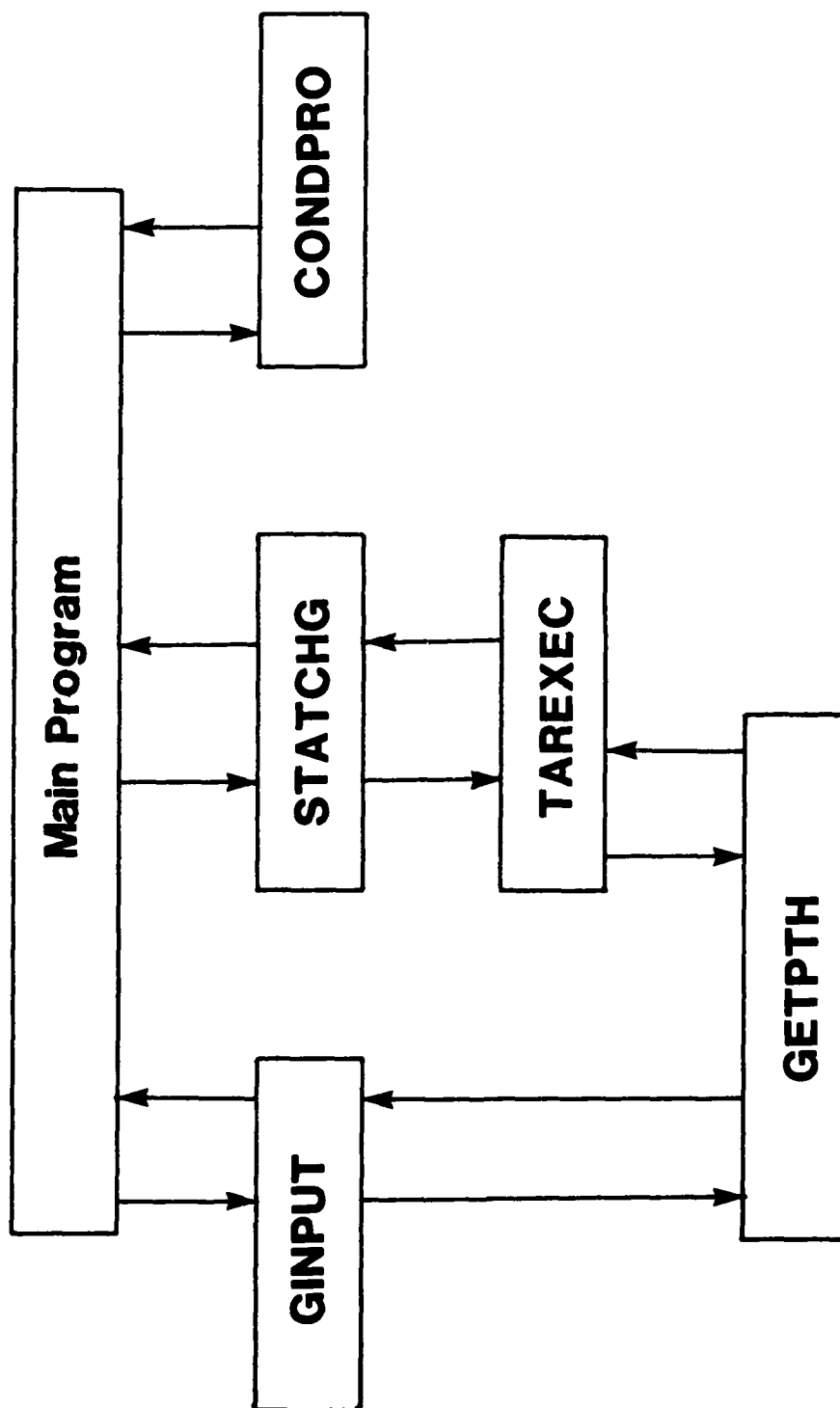


Figure B.1 SAS Routine Flow Structure

APPENDIX C

DOCUMENTED SAS CODE LISTING

C.1 INCLUDE FILE

The include file used by SAS contains a set of parameter statements used to define array dimensions and selected variable values used by the SAS model. Should the size of an array need to be altered by the user for a particular application, the user may appropriately adjust the array dimension parameters defined in the include file without modifying the SAS model. The model may then be recompiled and executed with the new array dimensions used. Whenever the statement INCLUDE 'W.INC' appears in the documented SAS computer listing, all statements contained in the include file are inserted.

```
C      W.INC -- Constants used in SAS.
PARAMETER MAIMPRY = 10      ! Maximum number of aim points per
C                             ! target, ordered by priority
PARAMETER MAIMPTS = 60      ! Maximum number of aim points
PARAMETER MGROUPS = 10      ! Maximum number of groups in TGROUPS
PARAMETER WSEED1 = 23       ! Initial seed value for RANDU
PARAMETER WSEED2 = 41       ! Initial seed value for RANDU
PARAMETER WSEPOIS = 75      ! Minimum separation distance
PARAMETER WSHOTS = 100      ! Number of shots fired to find a
C                             ! probability from, using Monte Carlo techniques
PARAMETER WSTATC = 99       ! Maximum number of status changing targets
PARAMETER WTARGS = 140      ! Maximum number of targets
PARAMETER WICOLAT = 15      ! Maximum number of collateral targets
C                             ! destroyed when a target is destroyed
PARAMETER MTMITGA = 16      ! Maximum number of targets than can be
C                             ! hit, given an aim point
PARAMETER MTITLEL = 30      ! Maximum length (in 2-byte words) of hardware
C                             ! option title
PARAMETER MTITLES = 5       ! Maximum number of hardware option titles
PARAMETER MTMINTS = 20      ! Maximum number of time intervals
PARAMETER MWYPES = 22       ! Maximum number of weapon types
C      end of W.INC.
```

C.2 DOCUMENTED SAS CODE LISTING

A documented code listing of the SAS computer model is provided in this section. The listing is intended for the reader who desires specific details on the operation of the model.

```

PROGRAM SAS
INCLUDE "A.INC"
COMMON /L1/ *CEP(*MTYPES)      !Aiming error array.
COMMON /L2/ *RATE(*MTYPES)      !Weapon firing rate array.
COMMON /L3/ *PEXIST(*MTARGS)     !Probability that a particular
                                ! target exists with time.
COMMON /L4/ *EVENTIM(*MTARGS)    !EVENTIM(I) = Next status
                                ! change event time for tar-
                                ! get I.
COMMON /L5/ *NUMTAR(*MTARGS)     !Array of target numbers.
COMMON /L6/ *ITARDES(*MTARGS, 5) !Array of 10 character target
                                ! descriptions.
COMMON /L7/ *TARCHAR(*MTARGS, 6) !Array containing target data.
COMMON /L8/ *WTYPE(*MTARGS)      !Array containing firer's
                                ! weapon type.
COMMON /L9/ *ROUNDS(*MTARGS)      !Number of rounds available to
                                ! each firer.
COMMON /L10/ *NAIMPRY(*MTARGS, *NAIMPRY) !Array listing aim points by
                                ! priority for each firer.
                                ! NAIMPRY(I,1 thru NAIMPRY)
                                ! contains all aim points for
                                ! firer I, with NAIMPRY(I,1)
                                ! being most important,
                                ! NAIMPRY(I,2) being second,
                                ! and so on.
COMMON /L12/ *MTCOLAT(*MTARGS, *MTCOLAT) !Array specifying targets which
                                ! may be collaterally damaged.
                                ! MTCOLAT(I,1 thru MTCOLAT)
                                ! contains all targets which
                                ! may be collaterally damaged
                                ! by hitting target NUMTAR(I).
COMMON /L13/ *NUMAIMP(*NAIMPTS)    !Array of aim point numbers.
COMMON /L14/ *AIMDES(*NAIMPTS, 5)  !Array of 10 character aim point
                                ! descriptions.
COMMON /L15/ *AIMCHAR(*NAIMPTS, 4) ! Array of aim point data.
COMMON /L16/ *MTHITGA(*NAIMPTS, *MTHITGA) !Array specifying a list of tar-
                                ! gets which could be hit by
                                ! firing at aim point
                                ! NUMAIMP(I).
COMMON /L17/ *RESULTS(*MTARGS, *MTHITS) !Array used to store simulation
                                ! results. RESULTS(I,J) would
                                ! contain the probability that
                                ! target I would still exist
                                ! after J time intervals have
                                ! elapsed.
COMMON /L18/ *PTARHIT(*MTARGS, *NAIMPRY, *MTHITGA) !Array used to store
                                ! probabilities of hit.
                                ! PTARHIT(I,J,K) contains the
                                ! the probability that firer
                                ! NUMTAR(I) fires at aim point
                                ! NUMAIMP(J) and hits target
                                ! NUMTAR(K).
COMMON /L19/ *NAMEOPF(5)           !Name of output file where
                                ! results are to be written.
                                ! maximum of 10 characters.

```

```

REAL PTXSV(MTARGS), RTIME(MTARGS)      ! ulation.
                                           ! PTXSV is used to temporarily
                                           ! save the PTXIST array while
                                           ! computing collateral damage
                                           ! effects.
                                           ! RTIME is used to print out
                                           ! time headings in results
                                           ! output.
INTEGER NFIDX(MTARGS)                   ! NFIDX is an array used to determine
                                           ! the index of the next at-
                                           ! tacker or defender to fire
                                           ! Entries into the NFIDX array
                                           ! are pointers into the
                                           ! PTXIST and VUMTAR arrays.
CHARACTER *2 NAMEORF, NAMEIDE, NAMEORF, NAMEIDF, NAMEADF, IAINDES,
1      ITARDES

Initialize aiming error for each weapon type.

DATA ACEP/0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 15.0, 1.12, 0.5,
1      0.75, 0.75, 0.75, 0.75, 0.75, 0.75, 0.75, 0.75, 22.5,
2      1.68, 0.75, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 30.0,
3      2.24, 1.67

Initialize firing rate for each weapon type.

DATA RATE/1.714, 6.316, 6.316, 0.343, 0.3, 0.109, 6.0, 0.109, 4.0,
1      12.0, 0.133, 1.714, 6.316, 6.316, 0.343, 0.3, 0.109,
2      6.0, 0.109, 4.0, 12.0, 0.133, 1.714, 6.316, 6.316, 3.0, 1.2,
3      1.2, 6.0, 0.109, 4.0, 12.0, 0.133/

Initialize maximum weapon kill radius for each weapon type.

DATA MAXRAD/0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 5.0, 0.0, 0.0,
1      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 5.0, 0.0, 0.0,
2      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 5.0, 0.0, 0.0/

GINPUT is called to insure all data has been obtained for the simula-
tion and all probabilities of hit have been determined.

CALL GINPUT

If an error occurred while obtaining input, branch is taken to avoid
incorrect simulation results.

IF (JERROR.NE. 0) GO TO 4100

Seed values for the random number generation routine RANDU are initial-
ized.

ISEED1 = 35901
ISEED2 = 45902

Initialize RUN to begin the simulation.

```



```

0000 saved in the NFIX array. If EVENTM(I) is less than the lowest firing
0000 time found so far, this firer becomes the prime candidate to fire. All
0000 information concerning previous candidates is destroyed by setting
0000 NFIXED to 1. The index of this firer is saved, and the TMIN value is
0000 reduced to the firing time of this firer.

0000 IF (EVENTM(I)-TMIN) 300, 400, 500

0000 Set TMIN to the next minimum event time found thus far.

300 TMIN = EVENTM(I)

0000 Set the number of firers found having firing times equal to the new
0000 minimum event time to 1.

0000 NFIXED = 1

0000 Store the index of the firer having the new minimum event time into the
0000 NFIX array.

0000 NFIX(1)=I
0000 GO TO 500

0000 Increment the number of firers having minimum event time.

400 NFIXED = NFIXED+1

0000 Branch back to top of loop if the event time for all firers has not
0000 been checked. After all firer's firing times have been compared, TMIN
0000 will contain the minimum firing time, NFIX will contain indices of
0000 all firers having this minimum firing time, and NFIXED will contain the
0000 number of firers with equal minimum firing times.

500 CONTINUE

0000 Determine whether the next set of results should be saved.

0000 TRES is the time at which the last set of results was saved.
0000 DELTAT is the time difference between result output, hence TRES is
0000 the time at which the next set of results is to be saved.

600 TRES = TRES+DELTAT

0000 If the time of next result save is less than(or equal to) the time that
0000 the next firing occurs, then results for the simulation at time TRES
0000 are saved as the current value in the PTEXIST array, since only firing
0000 situations can effect PTEXIST values.

0000 IF (TRES .GT. TMIN) GO TO 800

0000 Save simulation results after TRES units of time have elapsed in the
0000 simulation.

0000 DO 700 I = 1, NTAES
0000 RESULTS(I, ILCES) = PTEXIST(I)

```

```

700  CONTINUE

Increment the index into the results array to point to the location to
store results for next time increment.
ILNRS = ILNRS+1

Increment the time at which the next set of simulation results should
be obtained.

TNRES = TNRES
GO TO 600

If the time of the next firing is greater than the maximum time desired
for the simulation, branch to output results and end.

800  IF (TMIN .GT. TMAX) GO TO 3200

If the time of the next status change is less than or equal to the time
at which the next player fires, perform the status change before
allowing the player to fire.

IF (TMIN .LT. STMIN) GO TO 900

Maintain seed values so that varying the number of status changes, etc.
does not affect the selection of random numbers used by the main
program.

ISEED1 = ISEED1
ISEED2 = ISEED2

Perform status change.

CALL STATUS

If an error occurred during the status change, branch and end execution.

IF (JERROR .NE. 0) GO TO 4100

Reestablish possibly modified seed values.

ISEED1 = ISEED1
ISEED2 = ISEED2

Select the index of the next player to fire. Initially assume that
there is a single player who has a unique minimum response time. If
there is a unique player with minimum response time, the index of this
player would be the first element in the NFIDX array.

900  NFIRER = NFIDX(1)

If only one firer is selected, branch. The firer's index is stored in
NFIRER.

IF (NFIRED .EQ. 1) GO TO 1400

```

```

00      If there exists more than one player with equally minimum firing time,
00      then a selection must be made as to which of these players is to fire.
00      The methodology to select which of these players is to fire is based on
00      the premise that each player has an equal chance of firing first. A
00      uniform random number on the real interval (0,1) is selected. This num-
00      ber is then transformed to the real interval (0,NMINEQ), where NMINEQ is
00      the number of players having equally minimum firing times, by multi-
00      plying the U(0,1) random variable by NMINEQ. The U(0,NMINEQ) random
00      variable is then transformed to the interval (1,NMINEQ+1) by adding
00      1 to the U(0,NMINEQ) value. Next, the U(1,NMINEQ+1) random variable
00      value is truncated obtaining the index into the NFIDX array of the
00      next player to fire. The truncation is performed automatically by
00      assigning the real value between (1,NMINEQ+1) to an integer variable,
00      in this case NEXTF.

00      CALL RANDU (ISEED1, ISEED2, U)

00      Obtain index to next firer, NEXTF, which is an integer number between
00      1 and NMINEQ.

00      NEXTF=U*FLOAT(NMINEQ)+1.

00      Obtain index of player to fire in NFIRER. NUMTAR(NFIRER) is the actual
00      firer number.

00      NFIRER=NFIDX(NEXTF)

00      Save the probability of existence array temporarily so that collateral
00      damage may be assessed correctly.

1400  DO 1500 I = 1, NTARS
00      PTEXS(I) = PTEXIST(I)
1500  CONTINUE

00      Obtain the probability(stored in PFIREX) that the firer which has been
00      selected to fire, is still around to fire(ie - PFIREX is the probab-
00      ility that the firer has not been killed previously).

00      PFIREX = PTEXIST(NFIRER)

00      PAINC is the cumulative probability that the firer has selected an aim
00      point to fire at.

00      PAINC = 0.0

00      JJ is an index into the NAIMPRY array specifying a particular aim point
00      from the prioritized list of aim points the firer NFIRER may aim at.

00      JJ = 1

00      The aim point number is stored in NAIMPT. This is used to index into
00      the NTHIIGA(umber of Targets HIT Given Aim point) and the PTARHIT(
00      Probability Target is HIT) arrays.

1500  NAIMPT = NAIMPRY(NFIRER, JJ)

```

NAIMP=0 indicates all aim points that firer #FIRER has selected to fire at have been evaluated. The branch is taken, completing the 'perform firing event code section'.

IF (NAIMP.EQ. 0) GO TO 3000

P is the probability that firer #FIRER has not fired at an aim point investigated previously.

P = 1.0-PAINC

P=0. indicates that the firer, given he is still alive to fire, has already selected aim points to fire at with probability equal to 1. Thus, he will not fire at other possible aim points, and hence, the branch is taken ending the 'perform firing process'.

IF (P.EQ. 0.0) GO TO 3000

NTARGET is the number of the primary target associated with the aim point whose index is given by NAIMP.

NTARGET = NTHIGA(NAIMP, 1)

The methodology used to determine the probability that the firer fires at a particular aim point is outlined in the following text.

The target array is searched to find the index of NTARGET. The index to NTARGET is stored in ITARGET when found. If not found, an error has occurred in defining target status change in input and an error message is issued.

The following statements are used to determine the probability that a particular aim point is chosen.

STATE 1) The probability that the primary target, associated with the aim point the firer is firing at, exists(stored in PAIX), or

STATE 2) the probability that the firer has not fired at an aim point previously(stored in P).

Each time a player is selected to fire, he will either select one aim point to fire at or choose not to fire at any aim point because of high probability that each primary target has been destroyed previously. This concept may be expressed in terms of probability as follows:

RULE 1) The probability that the firer either fires at one of the aim points in his prioritized aim point list or chooses not to fire must sum to 1.

Normally, the probability that the firer fires at a particular aim point on his prioritized list of aim points is the probability that the primary target associated with that aim point still exists(RULE 1). Early in the simulation however, the probability that each of the pri-

many targets(associated with the prioritized list of aim points that the firer is interested in) still exists is quite high.

Since the probability that he fires at each aim point is proportional to the probability that it's primary target exists(STMT 1), a probability that he selects one of the aim points in his prioritized list in could exceed 1. This violates the condition specified in RULE 1, hence, STMT 2 is necessary to avoid the situation where the probability would exceed 1 for the firer selecting aim points in his prioritized list to fire at.

The following example illustrates in better detail how STMTs 1 and 2 are used in the simulation. Suppose a firer has the choice of firing at one of these possible prioritized aim points(A1, A2 or A3) or not to fire at all because the primary targets associated with each of these aim points may have been destroyed previously. The following table gives the probability that the primary targets T1, T2 and T3 (associated with aim points A1, A2 and A3 respectively) have not been previously destroyed.

T1	T2	T3
.5	.3	1.

In this example we will assume that the firer's strategy is to first fire at aim point A1 attempting to destroy target T1, and after it is destroyed, he will fire at aim point A2 attempting to destroy target T2, and so on. The probability that the firer chooses to fire at A1 is given by the minimum between PAIM(computed by STMT 1) and P (computed by STMT 2). PAIM as computed by STMT 1 for the primary aim point of interest, A1, would be:

$PAIM = 1 - .5$, where .5 is the probability that the primary target (T1) associated with aim point A1 still exists.
Thus, $PAIM = .5$ for aim point A1.

P, which is the probability that the firer has not fired at a previous aim point is initially 1, since this is the first aim point that may be chosen.

Thus, the probability that the firer chooses aim point A1 to fire at is given by the minimum of $PAIM(.5)$ or $P(1.)$. This probability, which is stored in the variable PFIREAP, is .5 for aim point A1. This leaves the firer with a maximum probability equal to .5 of being able to fire at aim point A2.

The PAIM value for firing at aim point A2 is the probability that the primary target associated with aim point A2 still exists($PAIM = .3$ from table). P is the probability that he did not fire at A1($P = .5$). Therefore the probability that aim point A2 is selected to be fired at is given by $PFIREAP = .3$.

The PAIM value for firing at aim point A3 is 1., obtained from the table. P is the probability that aim point A1 was not chosen and aim point A2 was not chosen, which is expressed by:

$$P = 1 - (.5(\text{probability A1 selected}) + .3(\text{probability A2 selected})) = .2$$

PFIREAP is given as the minimum between $PAIM = 1$, or $P = .2$. Thus,

the probability that aim point A3 is selected to be fired at is
PFIREAP=.2.

Notice that if we add the probabilities that the firer fires at either A1, A2 or A3 we get 1. (ie = .5+.3+.2). This, however would not be the case if the probability that primary target T3 existed was less than .2. Then there would be a probability that neither aim point A1, A2 or A3 would be selected (ie = the firer would not fire, since the primary targets associated with aim points of interest to him would have been destroyed previously. The following data summarizes the probability that each of the aim points A1, A2 or A3 were selected:

A1	A2	A3
.5	.3	.2

This ends the aim point selection description section.

```

00 1700 ITARGET = 1, NTARS
      IF (NNTAR(ITARGET) .EQ. NTARGET) GO TO 1800
1700 CONTINUE
      TYPE *, 'TARGET ', ITARGET, ' COULD NOT BE FOUND(3).'
      GO TO 4100

Get the probability that the primary target associated with aim point
NAIPT still exists, and store into PAIM.

1800 PAI4 = PTEXSV(ITARGET)

Determine whether to use PAIM or P as the probability NAIPT is select-
ed to be fired at.

      IF (PAI4 .GE. P) GO TO 1900

Set the probability of firing at this aim point to the probability that
the primary target associated with this aim point has not been destroy-
ed.

PFIREAP = PAIM

Increment the cumulative probability that the firer has selected an aim
point previously by adding in the probability that this aim point was
selected.

PAINC = PAINC+PFIREAP
GO TO 2000

The probability that the firer fires at this aim point, PFIREAP, is the
remaining probability that he has not selected a aim point previously.

1900 PFIREAP = P

In this situation, the firer has definitely selected an aim point from
those evaluated previously, hence PAINC is set to 1.

PAINC = 1.0

```

The next section modifies the probability of existence array, PTEXIST, appropriately for all targets possibly effected by NFIRER firing at aim point NAIMPT.

At this point a firer, NFIRER, has been selected to fire, an aim point has been chosen, NAIMPT, and the probability that this aim point is fired at has been determined and is stored in PFIREAP. The effect of this event on the target existence probability for all targets which may be damaged will now be determined.

KK is used to index a particular target which may be hit by firing at aim point NAIMPT. KK is initially set to 1, specifying the first target (the primary target) associated with aim point NAIMPT. The actual target number corresponding to aim point NAIMPT and index KK is given by NTHITGA(NAIMPT, KK) and stored in NTARGET. The probability that the target specified by KK is hit when NFIRER fires at aim point NAIMPT is obtained from the PTARHIT array and stored in PTH.

The probability that this target still exists after NFIRER fires a shot at aim point NAIMPT is computed by multiplying:

- 1) the probability that the target has not been destroyed previously, from the PTEXIST array. (This is also the probability that this target exists before this shot is fired. For simplicity the phrase "this shot fired" will be used to avoid repetition of the phrase NFIRER fires a shot at aim point NAIMPT.) X
- 2) the probability that the target still exists after this shot is fired.

This is the probability that the target was not hit previously and continues to remain undamaged after this shot is fired.

The probability that the target still exists after this shot is fired is equal to:

1.-the probability that the target is hit by this shot.

The probability that the target is hit by this shot is expressed by multiplying:

- 1) the probability that the firer still exists to fire the shot (ie - he has not been killed previously), stored in PFIREX. X
- 2) the probability that, given the firer still exists (ie - can fire), he selects aim point NAIMPT to fire at, stored in PFIREAP. X
- 3) the probability that, given the firer still exists and he fires at aim point NAIMPT, he can hit the target, stored in PTH.

Collateral damage is assessed by decreasing the probability


```

0000 destruction of target NTARGET have been processed, and branch to modify
0000 probability of existence of NTARGET value, PEXIST(ITARGET).
0000
0000 IF (NCOLTAR .EQ. 0) GO TO 2700
0000
0000 Obtain the index of NCOLTAR in the NUNTAR array.
0000
0000 DO 2500 ICOLTAR = 1, NUNTAR
0000     IF (NUNTAR(ICOLTAR) .EQ. NCOLTAR) GO TO 2600
2500 CONTINUE
0000 TYPE *, 'TARGET ', NCOLTAR, ' COULD NOT BE FOUND(2).'
0000 GO TO 4100
0000
0000 Modify the probability that the collaterally damaged target NCOLTAR
0000 exists by multiplying it's previous existence probability by the prob-
0000 ability that target NTARGET is not hit by this shot. Collateral damage
0000 is assessed to target NCOLTAR assuming that if target NTARGET is hit,
0000 NCOLTAR will also suffer equal damage. The model does not presently
0000 allow the user to input a probability of collateral damage to target
0000 NCOLTAR if target NTARGET is hit. The probability that target NCOLTAR
0000 is damaged if NTARGET is hit is 1.
0000
2500 PEXIST(ICOLTAR) = PEXIST(ICOLTAR)*PTARDES
0000
0000 KK is incremented to point to the next collaterally damaged target.
0000
0000 KK = KK+1
0000
0000 If KK is less than the maximum number of targets which can be collater-
0000 ally damaged by hitting target NTARGET, branch and process next collat-
0000 erally damaged target, otherwise the probability of existence has been
0000 notified for all collaterally damaged targets.
0000
0000 IF (KK .LE. NCOLLAR) GO TO 2600
0000
0000 Modify the probability of existence for target NTARGET to reflect the
0000 change in existence probability due to firing this shot.
0000
2700 PEXIST(ITARGET) = PEXIST(ITARGET)*PTARDES
0000
0000 Increment KK to point to the next target which may be hit by firing at
0000 aim point NUNTAR.
0000
2800 KK = KK+1
0000
0000 If KK is greater than the maximum number of targets which can be hit by
0000 firing at a given aim point, all targets have been processed for this
0000 aim point, otherwise branch to modify all target existence probabil-
0000 ities effected by hitting the next target associated with this aim
0000 point.
0000
0000 IF (KK .LE. NTHETA) GO TO 2100
0000
0000 Increment JJ to point to the next aim point available to the firer for
0000 selection.

```

```

0
2900 JJ = JJ+1
0000
0000 If JJ is less than or equal to the maximum number of aim points permit-
0000 ted for a given firer, branch and process next aim point, otherwise all
0000 aim points that this firer may shoot at have been processed.
0000
0000 IF (JJ .LE. MAXIPRY) GO TO 1600
0000
0000 NWT is a key indicating the type of weapon used by NFIRER, the player
0000 who just fired the last round.
0000
3000 NWT = NWTTYPE(NFIRER)
0000
0000 EVENTIM(NFIRER) is the time at which this player is expected to fire
0000 his next round. The expected time to aim and fire the next shot stored
0000 in the weapon firing rate array(*RATE), is added to the present value
0000 of EVENTIM(NFIRER).
0000
0000 EVENTIM(NFIRER) = EVENTIM(NFIRER)+*RATE(NWT)
0000
0000 The number of rounds fired by NFIRER is incremented.
0000
0000 NROUNDOF(NFIRER) = NROUNDOF(NFIRER)+1
0000
0000 If NFIRER has run out of ammunition, a message is printed and the time
0000 at which he fires his next shot and the firer's next shot event time
0000 is set greater than the total simulation time to insure that he will
0000 not fire again(unless he acquires more ammunition at a later date in
0000 the simulation).
0000
0000 IF (NROUNDOF(NFIRER) .GT. NROUNDS(NFIRER)) GO TO 200
0000 WRITE (3, 3100) NFIRER, EVENTIM(NFIRER)
3100 PRINT (' FIRMER: ', I3, ' RAN OUT OF AMMUNITION AFTER ', F10.2,
0000 1 ' SECONDS.')
0000 EVENTIM(NFIRER) = TMAX+1.0
0000 GO TO 200
0000
0000 The simulation is complete with probability of existence results stored
0000 for all targets at each DELTAT time interval in the RESULTS array.
0000
0000 Decrement IURES to the exact number of time interval "snap shots" saved
0000 in the RESULTS array.
0000
3200 IURES = IURES-1
0000
0000 Fill the RTIME array with the time at which each "snap shot" of target
0000 existence probabilities was taken.
0000
0000 RTIME(1) = TSTART
0000 DO 3300 I = 2, IURES
0000 RTIME(I) = RTIME(I-1)+DELTAT
3300 CONTINUE
0000 JT = 2*MTITLE+2
0

```


000 000 0

CLOSE (UNIT = 3)

Close target data base file.

CLOSE (UNIT = 4)

Close aim point data base file.

CLOSE (UNIT = 7)

STOP

END

SUBROUTINE CONOPRO

CONOPRO -- A VAX/VMS FORTRAN subroutine which calculates conditional probabilities, written for JAYCOR by Joe Humphrey, Mar 1980.

ITGROUP contains a number of groups of targets, each of which has a probability of being destroyed (in RESULTS). For each group, the conditional probability that i targets are destroyed, as i goes from 0 to the number of targets in the group, are calculated and written to unit 3. This is done for several times of interest (TSTART, TSTART+DELTAT, TSTART+2*DELTAT, . . ., TSTART+(ILRES-1)*DELTAT).

Define constants and variables.

INCLUDE "4.INC"

PARAMETER OUT = 3 ! Unit number of output file

CHARACTER *2 ITARDES(NTARGS, 5) ! Target descriptors

INTEGER FIRST ! Pointer to last destroyed target,
!- after trailing destroyed targets
!- have been restored (E. G. if KILLED
!- = FFFIFFIT, FIRST = 1)

INTEGER GIND ! Index to target in ITGROUP
INTEGER GROUP ! Index to group in ITGROUP
INTEGER GTIND(NTARGS) ! Conversion from GIND to FIND
INTEGER ILRES ! Number of time increments
INTEGER IIGROUP(NTARGS, NGROUPS) ! Target numbers of targets in groups
INTEGER LAST ! Pointer to last destroyed target
INTEGER LENGTH ! Number of trailing destroyed targets
!- (E. G. if KILLED = FFFIFFIT,
!- LENGTH = 3)

INTEGER NDEST ! Number of targets to be destroyed
INTEGER NTARGS ! Number of targets in group
INTEGER NNTAR(NTARGS) ! Conversion from FIND to target
!- number

INTEGER TIAC ! Number of time increments passed
INTEGER TIID ! Index to target in target array
!- (RESULTS, TARNUM, etc.)
INTEGER WORD ! Pointer to word of ITARDES

LOGICAL KILLED(NTARGS) ! .TRUE. if the corresponding target
!- in ITGROUP is destroyed; .FALSE.
!- otherwise

REAL DELTAT ! Length of time increment (seconds)
REAL EPROR ! Probability of an event (A
!- particular configuration of KILLED)

REAL EXPNDES(NTINTS) !
REAL GPROB(NTINTS) ! Probability of a group of events (E.
!- G. that 1 target is destroyed in

```

C      REAL RESULTS(MTARGS, MTWINTS)      !- ITGROUP)
C                                          ! Probabilities that a target will be
C      REAL TSTART                        !- around after so many time increments
C                                          ! Starting time

COMMON /L6/ NUMTAR, /L7/ ITARDES, /L17/ RESULTS, /L24/ DELTAT, /L44/
1      TSTART, /L46/ ITGROUP, /L47/ ILRES

C      Find number of targets in group, converting GINDs to TINDs.
C      Return if 0 targets in group.

DO 2600 GROUP = 1, NGROUPS
  DO 100 TINC = 1, ILRES
    EXPIDES(TINC) = 0.0
100    CONTINUE
    DO 400 GIND = 1, MTARGS
      IF (ITGROUP(GIND, GROUP) .LE. 0) GO TO 500
      DO 200 TIND = 1, MTARGS
        IF (ITGROUP(GIND, GROUP) .EQ. NUMTAR(TIND)) GO TO 300
200      CONTINUE
      TYPE *, "Target", ITGROUP(GIND, GROUP), " in ITGROUP not "//
1      "found."
      CALL EXIT

300      GTIND(GIND) = TIND
400      CONTINUE
      GIND = MTARGS+1
      MTARGS = GIND-1
      IF (MTARGS .LE. 0) GO TO 2700

C      Write title, target numbers and descriptors, and times of
C      interest.

IF (GROUP .EQ. 1) WRITE (OUT, 600)
600  FORMAT ("1")
      WRITE (OUT, 700) GROUP, (NUMTAR(GTIND(GIND))),
1      (ITARDES(GTIND(GIND), WORD), WORD = 1, 5), GIND = 1,
2      MTARGS)
700  FORMAT (///X, "CONDITIONAL PROBABILITIES FOR GROUP ", I1,
1      ", CONSISTING OF TARGETS:", /, (X, 9(I3, X, 5A2)))
      WRITE (OUT, 800) (TSTART+FLOAT(TINC-1)*DELTAT, TINC = 1, ILRES)
300  FORMAT (/X, 3A, <ILRES>(X, F5.1))

C      Start out with no targets destroyed. Write number of targets
C      to be destroyed. Initialize group probability.

DO 900 GIND = 1, MTARGS
  KILLED(GIND) = .FALSE.
900  CONTINUE

DO 2300 NDEST = 0, MTARGS

  DO 2000 TINC = 1, ILRES
    GPROR(TINC) = 0.0

```

```

C          Find event probability. Sum into group probability.
1000      EPROB = 1.0
          DO 1100 GIND = 1, NTARGS
              IF (KILLED(GIND)) THEN
                  EPROB = EPROB*(1.0-RESULTS(GTIND(GIND), TINC))
              ELSE
                  EPROB = EPROB*RESULTS(GTIND(GIND), TINC)
              END IF
1100      CONTINUE
          GPROB(TINC) = GPROB(TINC)+EPROB

C          Find last target destroyed. If not at end of list,
C          restore that target and destroy the next one.

          DO 1200 LAST = NTARGS, 1, -1
              IF (KILLED(LAST)) GO TO 1300
1200      CONTINUE
          LAST = NTARGS
1300      IF (LAST.LE. NTARGS) THEN
              KILLED(LAST) = .FALSE.
              KILLED(LAST+1) = .TRUE.
              GO TO 1000
          END IF

C          Last target destroyed == find number of targets
C          destroyed at end of list. Restore those targets.

          DO 1400 GIND = LAST, 1, -1
              IF (.NOT. KILLED(GIND)) GO TO 1500
              KILLED(GIND) = .FALSE.
1400      CONTINUE
          GIND = 0
1500      LGIND = LAST-GIND

C          Find last destroyed target in revised list. If none
C          are destroyed, then all events for that number of
C          destroyed targets have been found == find it for next
C          time increment.

          DO 1600 FIRST = LAST, 1, -1
              IF (KILLED(FIRST)) GO TO 1800
1600      CONTINUE
          IF (LGIND.GT. 0) THEN
              DO 1700 GIND = 1, LENGTH
                  KILLED(GIND) = .TRUE.
1700      CONTINUE
          END IF
          GO TO 2000

C          Destroyed target found == restore it, destroy next
C          LENGTH+1 targets, and find event probability.

1800      KILLED(FIRST) = .FALSE.
          DO 1900 GIND = FIRST+1, FIRST+LENGTH+1

```

```

                KILLED(GIND) = .TRUE.
1900          CONTINUE
                GO TO 1000
2000          CONTINUE

C          Group probabilities for all times of interest have been
C          found (for NDEST destroyed targets). Write them and find
C          group probabilities for NDEST+1 destroyed targets.

                IF (LENGTH .LT. NTARGS) KILLED(LENGTH+1) = .TRUE.
                WRITE (OUT, 2100) NDEST, (GPROB(TINC), TINC = 1, ILRES)
2100          FORMAT (X, I3, X, <ILRES>(X, F5.3))
                IF (NDEST .GT. 0) THEN
                    DO 2200 TINC = 1, ILRES
                        EXPNDES(TINC) = EXPNDES(TINC)+GPROB(TINC)*NDEST
2200          CONTINUE
                END IF
2300          CONTINUE
                WRITE (OUT, 2400)
2400          FORMAT (//X, ' EXPECTED NUMBER OF TARGETS DESTROYED AFTER',
1              ' TIME ELAPSED:')
                WRITE (OUT, 600) (TSTART+FLRAT(TINC-1)*DELTAT, TINC = 1, ILRES)
                WRITE (OUT, 2500) (EXPNDES(TINC), TINC = 1, ILRES)
2500          FORMAT (4X, <ILRES>(X, F5.2))
2600          CONTINUE

C          All probabilities found -- return.

2700          RETURN
                END

```

SUBROUTINE GETPTH

PURPOSE - Determine the probability that firer AF fires at aim point XAM and hits target AT. The probability of hit is returned in PTH. The routine is based on a Monte-Carlo sampling technique which fires a number of shots at a target, counting the number of hits which occurred. The number of hits is divided by the number of tries and a probability of hit is obtained. Aiming error is defined in terms of a radius (contained in array ACEP) within which 1/2 of all shots are expected to land. The radius in which 1/2 of the shots land defines a Raleigh distribution with mean equal to the radius, r. In order to generate random shots from the Raleigh distribution, the model first draws random numbers from a uniform distribution, then converts these to normal random variables. The normal random variables are then used as absolute X and Y offsets from the intended aim point for a particular shot. The miss distance from the aim point is expressed as:

$$DIST = \sqrt{X^2 + Y^2}$$

In order to generate normal random variables for the X and Y offsets, the normal distribution from which these variables are drawn must provide random variables X and Y which satisfy the assumption that 1/2 the time the distance, given by $\sqrt{X^2 + Y^2}$, is $\leq$ or $=$ r. The mean for each X and Y random variable is zero, since it is just as likely to be off high as it is low. It can be shown mathematically that the variables X and Y drawn from a $N(0, r^2/2.493)$ (a normal distribution with mean=0, variance= $r^2/2.493$) will satisfy $\sqrt{X^2 + Y^2} \leq$ or $=$ r 50% of the time, where r is the distance from the aim point in which 50% of all shots fired are expected to land.

The model presently accepts targets only rectangular in shape, however, by using the Monte-Carlo sampling technique, any target shape may be used, provided the code is written to determine whether the shots fired lie within the target area or outside the target area.

This routine also evaluates situations where random aim areas are used instead of fixed aim points. When random aim areas are used, shots are fired at aim points selected uniformly from a rectangular region whose coordinates are defined by the analyst. Each time a new shot is to be fired, a new aim point for the shot is selected at random from within the area.

The number of shots fired in determining the probability of hit may be modified by changing the value of NSHOTS in the include file W.INC.

```

INCLUDE 'W.INC'
COMMON /L1/ ACEP(NTYPES)
COMMON /L6/ NMTAR(NTARGS)
COMMON /L7/ ITARDES(NTARGS, 5)
COMMON /L8/ TARCHAR(NTARGS, 6)
COMMON /L9/ NTYPE(NTARGS)

```

```

COMMON /L13/ TUNATYP(MAINPTS)
COMMON /L14/ TATNPTS(MAINPTS, 5)
COMMON /L15/ ATACHAR(TATNPTS, 4)
COMMON /L27/ ISSE01, ISSE02
COMMON /L29/ RF
COMMON /L30/ RAN
COMMON /L31/ RT
COMMON /L32/ RTH
COMMON /L39/ IDXFTX
COMMON /L41/ IDXAIX
COMMON /L45/ IDXFXR
COMMON /L48/ WXXRAD(RTIPES)
CHARACTER *2 ITARVES, IAIWDES

```

WRT is used to accumulate the number of hits on target NT as the shots are fired.

```

300 WRT = 0

```

The seed values are saved on input and restored on completion so that reloading the numbers of shots fired will not effect the sequence of random numbers generated in other sections of the code.

```

ISSE01 = SSEE1
ISSE02 = SSEE2

```

ISSE01 is used to accumulate the total number of shots fired.

```

ISSE01 = 0

```

W is set to a key identifying the firer's weapon type.

```

W = WTYPE(IDXFIR)

```

WRAD is the weapon kill radius. This is a lethal radius in which the effects of the weapon fired will destroy the target.

```

WRAD = WXXRAD(W)

```

CRP is the radius, converted to apply to a circular normal distribution, within which one-half of all shots fired are expected to land.

```

CRP = 4CRP(W)*0.6443

```

XLFC - X coordinate of Lower left Firer Corner. X coordinate of lower left corner defining fixed position of firer. This value is used as the X coordinate of the lower left corner of a random area in which the firer may be located, if a random area for the firer's position is specified(specified by FRAHGT>0.).

```

XLFC = TARCHAR(IDXFIR, 3)

```

YLFC - Y coordinate of Lower left Firer Corner. Y coordinate of lower left corner defining fixed position of firer. This value is used as the Y coordinate of the lower left corner of a random

500 RWIDTH = FRAWID

Use random firer area height to determine expected point from which shots are fired from.

FHEIGHT = FRAHGT
GO TO 400

RAIMHGT - Random AIM point area HEIGHT. The random aim point area is a rectangular area in which the firer is expected to aim. Before each shot is fired an aim point is selected from this area at random assuming all points within this area are assumed equally likely.

600 RAIMHGT = AIMCHAR(NAM, 3)

RAIMWID - Random AIM point area WIDTH.

RAIMWID = AIMCHAR(NAM, 4)

XALC - X coordinate of random Aim point area Lower left Corner. If a fixed aim point is used, this is the Y coordinate of the fixed aim point in the target reference plane.

XALC = AIMCHAR(NAM, 1)

YALC - Y coordinate of random Aim point area Lower left Corner. If a fixed aim point is used, this is the Y coordinate of the fixed aim point in the target reference plane.

YALC = AIMCHAR(NAM, 2)

THEIGHT - Height of rectangular target area.

THEIGHT = TARCHAR(1, 1)

TWIDTH - Width of rectangular target area.

TWIDTH = TARCHAR(1, 2)

Obtain the center of rectangular target area with respect to height(
THD2 = Target Height Divided by 2).

THD2 = THEIGHT/2.0

Obtain the center of rectangular target area in terms of width(TWD2 =
Target Width Divided by 2).

TWD2 = TWIDTH/2.0

FRAHGT - Target Random Area Height. The random target area is a rectangular area in which the target itself is positioned. The position of the target within the random area is selected assuming each possible position is equally likely.

TPANGT = TARCHAR(NT, 5)

XLTC - X coordinate of the lower left Target Corner. Normally, XLTC is the X coordinate of lower left corner defining a fixed target position. If a random target area is specified, this is the X coordinate of the lower left corner of the random target area.

XLTC = TARCHAR(NT, 3)

YLTC - Y coordinate of the lower left Target Corner. Normally, YLTC is the Y coordinate of lower left corner defining a fixed target position. If a random target area is specified, this is the Y coordinate of the lower left corner of the random target area.

YLTC = TARCHAR(NT, 4)

XCTRTA - X coordinate for the Center of the Random Target Area.
TARCHAR(NT,6) is the width of the random target area.

XCTRTA = TARCHAR(NT, 6)-TWIDTH

YCTRTA - Y coordinate for the Center of the Random Target Area. The point (XCTRTA,YCTRTA) is the center of the random target area relative to the origin in the target reference plane.

YCTRTA = TPANGT-THEIGHT

XTCTR - X coordinate of Target Center.

XTCTR = TWD2+XLTC

YTCTR - Y coordinate of Target Center. The point (XTCTR,YTCTR) is the center of the target area relative to the origin in the target reference plane.

YTCTR = THD2+YLTC

Compute the distance from the firer(point where the shot is fired) and the target center if the firer is projected into the target reference plane.

XYDIST = SQRT((XFCTR-YTCTR)**2+(YTCTR-YTCTR)**2)

Compute the absolute distance between the firer(point where shot is fired) and target center by including the separation distance between reference planes.

DIST = SQRT(XYDIST**2+VSEPDIS**2)

The distance factor, DFACTDP, is used to modify the radius in which 1/2 or all shots are expected to land(CBP). The distance factor is computed by taking the ratio of absolute distance to the reference plane separation distance, since the CBP value was determined using the plane separation distance.

DFACTOR = DIST/SEPDIS

SIGMA is modified by the distance factor to generate random hit points which are expected to fall within a circle with increased radius proportional to the increase in distance between the firer and target normalized by the plane separation distance.

SIGMA = CEP*DFACTOR

SIGMAP is used to store the value of $2 * SIGMA^2$ for optimization reasons, preventing this calculation from being re-evaluated several times when it is later used to find the hit point for this shot.

SIGMAP = $-2 * SIGMA * SIGMA$

XAIMCOR = X AIM point COordinate value relative to the target reference plane origin.

XAIMCOR = AIRCHAR(44, 1)

YAIMCOR = Y AIM point COordinate value relative to the target reference plane origin. For a fixed aim point, XAIMCOR and YAIMCOR are the coordinate values of the fixed aim point. If a random aim area is desired, XAIMCOR and YAIMCOR will be redefined before they are used by selecting a random aim point from the random aim area.

YAIMCOR = AIRCHAR(44, 2)

Increment the number of shots firer to indicate firing next shot.

1100 NSHOT = NSHOT + 1

If we have exceeded the maximum number of shots necessary, the probability of hit can be determined and the branch is taken.

IF (NSHOT .GT. NSHOTS) GO TO 1700

If a fixed aim point is used, RAIHGT=0 and branch is taken, otherwise compute new aim point coordinates from random aim point area.

IF (RAIHGT .EQ. 0.0) GO TO 1200

Obtain new aim point X coordinate from random aim area. Select a point uniformly from the width of the random aim point area(RAIMWID) and add the absolute position of the lower left corner of the random area.

CALL RANDU (ISEED1, ISEED2, U)

XAIMCOR = U*IDIM+XALC

Obtain new Y coordinate from random aim point area.

CALL RANDU (ISEED1, ISEED2, U)

YAIMCOR = U*YRIGHT+YALC

If a fixed target area was specified (TRAHGT=0), the point (XTCTR,YTCTR) already identifies the location of the lower left corner of the fixed target area relative to the origin in the target reference plane. If a random target area was desired, values for XTCTR and YTCTR are reassigned in the following section before the next shot is fired.

1200 IF (TRAHGT .EQ. 0.0) GO TO 1300

Compute new X coordinate for lower corner of target in random target area. The X coordinate of the lower left corner is selected randomly from a line segment whose length is given by the width of the random target area minus the width of the target area. This places the lower left coordinate relative to the lower left corner of the random target area. To obtain the center of the target relative to the lower left corner of the random target area, one-half the target width is added (TWD2). Finally, to place the X coordinate of the target center relative to the origin of the target reference plane, the lower left corner of the random target area (XLTC) is added.

CALL RANDU (ISEED1, ISEED2, U)
XTCTR = U*XTCTRTA+TWD2+XLTC
CALL RANDU (ISEED1, ISEED2, U)

The new Y coordinate for the target center is obtained in a similar fashion independently. The point (XTCTR, YTCTR) now contains the position of the recently placed target area relative to the origin in the target reference plane.

YTCTR = U*YCTRTA+TWD2+YLTC
XYDIST = SQRT((XFCIR-XTCTR)**2+(YFCIR-YTCTR)**2)

Now that the new target position has been determined, the distance between the firer and target center is computed, the distance factor is obtained, and SIGMA redefined for the possible change in separation distance from the previous target position. This allows a new sigma factor, SIG*AF, to be used in obtaining the coordinates for the next shot.

DIST = SQRT(XYDIST**2+MSEPDIS**2)
DFACTOR = DIST/MSEPDIS
SIGMA = CEP*DFACTOR
SIG*AF = -2.*SIGMA*SIGMA

At this point, the distribution for shots fired at the target has been determined (by calculating the distribution variance stored in SIGMA), the aim point has been determined (stored in XAIMCOR, YAIMCOR), and the target has been positioned (center of target in XTCTR, YTCTR). We "fire the shot" by selecting two uniform random variables (U1 and U2) from the real interval (0.,1.). These values are then converted to two normal random variables (X and Y) having mean=0., variances SIGMA**2 via the transformation:

X=mean+SQRT(-2*variance*LOG(U1))*COS(2*pi*U2)

$Y = \text{mean} + \text{SQRT}(-2 * \text{variance} * \text{LOG}(U1)) * \text{SIN}(2 * \pi * U2)$

where $\pi = 3.1459$

NOTE - This technique for generating normal variates from the uniform distribution is described in the book: "Concepts and Methods in Discrete Event Digital Simulation", by George S. Fishman, P. 213.

```
1300 CALL RANDU (ISEED1, ISEED2, U)
    IF (U .GT. 0.9999) U = 0.9999
    IF (U .LT. 0.0001) U = 0.0001
    S2 = SIGMAF*ALOG(U)
    CALL RANDU (ISEED1, ISEED2, U)
    XSHOT = XAIMCOR + SQRT(S2)*COS(6.28319*U)
    CALL RANDU (ISEED1, ISEED2, U)
    It is necessary when evaluating the ALOG function that the U(0,1)
    value does not approach zero too closely, as the function is undefined
    at that point. It is also necessary that the U(0,1) value not
    approach 1. too closely as this would result eventually in the square
    root of a number very close to zero.
```

```
    IF (U .GT. 0.9999) U = 0.9999
    IF (U .LT. 0.0001) U = 0.0001
    S2 = SIGMAF*ALOG(U)
    CALL RANDU (ISEED1, ISEED2, U)
    YSHOT = YAIMCOR + SQRT(S2)*SIN(6.28319*U)
```

The point (XSHOT, YSHOT) is the point at which the shot hits in the target reference plane relative to the target origin. If the X coordinate of the shot falls to the left of the left side of the target, or to the right side of the target, the shot misses and the branch is taken to prevent registering a direct hit.

```
IF (XSHOT .LT. XTCTR-TWD2 .OR. XSHOT .GT. XTCTR+TWD2) GO TO 1600
```

If the Y coordinate is below the bottom of the target, or above the top of the target, the shot misses, and the branch is taken to avoid registering a direct hit.

```
IF (YSHOT .LT. YTCTR-TWD2 .OR. YSHOT .GT. YTCTR+TWD2) GO TO 1600
```

The shot has hit the target area. The number of target hits is incremented.

```
1500 NHIT = NHIT+1
```

Branch to fire next shot.

```
GO TO 1100
```

This shot has missed the target, however, depending on its proximity to the target, may inflict enough damage to the target to register a hit. If the target center is within the lethal radius of the weapon

```

round, a hit is registered.
1600 IF (WKRAD .EQ. 0.0) GO TO 1100
Compute the Distance From HIT Center(DFHTCTR).
DFHTCTR = SQRT((XSHOT-XICTR)**2+(YSHOT-YICTR)**2)
If the distance from the hit point is within the weapons lethal kill
radius, the branch is taken to register a hit.
IF (DFHTCTR .LE. WKRAD) GO TO 1500
The shot has landed outside the maximum lethal weapon radius, branch
to fire next shot.
GO TO 1100
Compute the probability of hit by dividing the number of hits by the
number of shots fired.
1700 PTH = FLOAT(NHIT)/FLOAT(NSHOTS)
Output the firer number identifier, aim point number/identifier, tar-
get number/identifier and probability of hit. Should unexplained
results be found when examining the simulation output, it is often
useful to view the probabilities of hit to locate possible errors in
input data. For a large simulation there may be a considerable number
of probabilities printed. Should this statement generate too much
output, the user may choose to make this statement a comment (by
placing a 'C' in column 1) without affecting the simulation results.
WRITE (3, 1300) NF, (ITARDES(IDXFIR, I), I = 1, 5), NUMAIMP(NAM),
1 (IAIMDES(NAM, I), I = 1, 5), NUMTAR(NT), (ITARDES(NT, I), I =
2 1, 5), PTH
1800 FORMAT (3(3X, I3, 1X, 5A2), 5X, F10.7)
RETURN
END

```

SUBROUTINE GINPUT

PURPOSE - The main purpose of the GINPUT subroutine is to get all necessary input needed to perform the SAS simulation. This includes all input in the input data file, the aim point data file and the target data file. An attempt is made to read each of these files. If all information has not been stored in these files previously, it will be obtained interactively. This routine also allows interactive modification of data contained within these files. This permits the user to set up the data to perform analysis of a baseline situation, then modify certain elements of the data to define an improved option, for example, easily due to the interactive method of data entry and retrieval.

The GINPUT routine is also used to initialize the aim point, target and probability of hit data base files and to make sure correct probabilities of hit have been obtained for all targets in their initial positions.

```

I CCLUDE 'A.INC'
COMMON /L4/ PTEXIST(*TARGS)
COMMON /L5/ EVENTID(*TARGS)
COMMON /L6/ NUMTAR(*TARGS)
COMMON /L7/ LTARDES(*TARGS, 5)
COMMON /L8/ TARCHAR(*TARGS, 6)
COMMON /L9/ NTYPE(*TARGS)
COMMON /L10/ TROUNDS(*TARGS)
COMMON /L11/ MATPRY(*TARGS, *MATPRY)
COMMON /L12/ ITCOLAT(*TARGS, *ITCOLAT)
COMMON /L13/ MATPR(*TARGS, *MATPR)
COMMON /L14/ TAINDES(*TARGS, 5)
COMMON /L15/ ATCHAR(*TARGS, 4)
COMMON /L16/ THITGA(*TARGS, *THITGA)
COMMON /L17/ PTARHIT(*TARGS, *MATPRY, *THITGA)
COMMON /L18/ LANEPR(5)
COMMON /L19/ LANEID(5)
COMMON /L20/ LANEPR(5)
COMMON /L21/ LANEPR(5)
COMMON /L22/ JERPR
COMMON /L23/ TMAX
COMMON /L24/ DELTAT
COMMON /L25/ LPARS
COMMON /L26/ LPARS
COMMON /L27/ ISEED1, ISEED2
COMMON /L28/ GROUND(*TARGS)
COMMON /L29/ LF
COMMON /L30/ LAA
COMMON /L31/ BT
COMMON /L32/ PTM
COMMON /L33/ MATPR(5)
COMMON /L34/ LANEPR(5)
COMMON /L35/ ISTATUS(*STATC, 2)
COMMON /L36/ STATUS(MSTATC, 2)
COMMON /L37/ *STATC
COMMON /L38/ *STATC
COMMON /L39/ IOXPR

```

```

COMMON /L40/ YTHA
COMMON /L41/ IOXA1M
COMMON /L44/ TSTART
COMMON /L45/ IOX1AR
COMMON /L46/ ITGROUP(NTARGS, NGROUPS)
COMMON /L49/ ITITLE(NTFILES, MTITLEL)
COMMON /L50/ NFILES
CHARACTER *1 ICHAR, IDEBUG, ISCHAR
CHARACTER *2 NAMEOPF, NAMEIDF, NAMEEDF, NAMEODF, NAMEADF, IAINDFS,
1 ITARDES
CHARACTER *10 NOBF, NTDF, AADF, NOPF, NIDF

```

The names of various input and output files used by the SAS model are described as follows. Each file name may be at most 10 characters in length.

```

NOBF = Name for probability of hit data base file.
NTDF = Name of target data base file.
AAOF = Name of aim point data base file.
NOPF = Name of output file.
NIDF = Name of input data file.

```

The contents of each of the data files is summarized as follows:

NOBF - contains the probability of hit for a given firer, aim point and target. An entry of -1, for a particular firer, aim point and target indicates that this probability has not yet been determined. This file is filled with -1's when the file is initialized.

NTDF - The target data file contains 10 items about each target in the simulation. These items are listed, as they are stored in the data base, as follows:

- 1) Target number which uniquely identifies each target,
- 2) 10 character target description,
- 3) target size, expressed as height and width,
- 4) flag indicating whether target is fixed or random('F' indicates fixed, 'R' indicates random)
 - a) If fixed, the file contains the fixed position of the lower left corner of the target relative to the origin in the target's reference plane and is given in terms of X and Y coordinates.
 - b) If random, the file contains the height and width of a random area in which the target is positioned and the fixed position of the lower left corner of the random area relative to the target's reference plane given by an X and Y coordinate.
- 5) flag indicating whether or not the target can fire a weapon ('Y' indicates yes, 'N' indicates no),
- 6) key indicating the weapon type, if the firer has no weapon type(e - can not fire) a zero is entered,
- 7) number of rounds of ammunition available to the firer,
- 8) the firer's response time(this is the time before the firer fires his first shot),

- 9) Air points which the firer may fire at listed by priority, and
- 10) A list of targets which may be collaterally damaged.

ADPF - The air point data file contains 4 items which define key air point information required for the simulation. These items are listed as follows:

- 1) Air point number which identifies each air point uniquely,
- 2) 10 character air point description (normally this description corresponds to the primary target associated with this air point),
- 3) a flag indicating whether the air point is fixed or random ("F" indicates fixed, "R" indicates random) where,
 - a) if fixed, the fixed position of the air point is given relative to the origin in the target reference plane (not the firer reference plane) by an X and a Y coordinate,
 - b) if random, the position and size of a rectangular random air area is given by specifying the lower left corner of the area by an X and a Y coordinate, and the height and width of the random area, and
- 4) a list of targets which may be hit by firing at each air point.

ADPR - This file contains summarized results obtained from the simulation. Results are classified into three main groups:

- 1) Probability of hit,
- 2) probability that a particular target exists with time, and
- 3) expected number of targets left and conditional probability that a specific number of targets remain with time.

ADPI - The input data file contains information which identifies the location of air point, target, and probability of hit data previously entered, gives the duration of the simulation, specifies how often simulation results should be printed, and defines when all target status changes occur.

```
EQUIVALENCE (ADPF, IADPF(1))
EQUIVALENCE (ADPR, IADPR(1))
EQUIVALENCE (ADPI, IADPI(1))
EQUIVALENCE (ADPF, IADPF(1))
EQUIVALENCE (ADPR, IADPR(1))
EQUIVALENCE (ADPI, IADPI(1))
```

Initialize flags used in reading input.

PI = -1

FLAG is used to initialize the values used in the probability of hit data base.

```
FLAG = -1.0
N1 = 0
Z = 0.0
ISTARCF = 0
JSTARCS = 0
ISTARCS = 0
```

NTAPE is used to indicate the logical unit number of the primary input device. NTAPE is initially set to 5, indicating that input is to be obtained interactively from the terminal. NTAPE is later changed to 1 if the input data has been previously stored in a disk file.

NTAPE = 5

Set the record size(in bytes) of each record contained in the target data base. The basic record length is 59 plus the length required for aim points and collaterally effected targets. Each aim point and collateral target number requires 3 characters.

NTRECSZ = 68+3*VAINPRY+3*NTCOLAT

Set the record size(in bytes) for records contained in the aim point data file. 3 characters are allotted for each target number which may be hit by firing at a given aim point.

NARECSZ = 41+3*IT-ITGA

NTMA = NTRECSZ*VAINPRY

100 TYPE *, 'NAME OF INPUT DATA FILE?'

200 READ (5, 200, ERR = 100, END = 100) (NAMEIDF(I), I = 1, 5)

200 FORMAT (5I2)

OPEN (UNIT = 1, NAME = NAME, TYPE = 'UNKNOWN', ERR = 14000)

If the user wishes to modify data in either the aim point or target data base files prior to simulation execution, an 'M' is entered in response to the next question. If all input has been entered previously and the user is ready to run, a 'R' is entered.

300 TYPE *, 'MODIFY TARGETS OR RUN? M-MODIFY, R-RUN'

READ (5, 400, ERR = 300, END = 300) IDEBUG

IF (IDEBUG .NE. 'M' .AND. IDEBUG .NE. 'R') GO TO 300

An attempt is made to read the first record from the input data file. If the read is successful, the flag NTAPE is set to 1, specifying that input should come from this file not from the terminal.

TSTART is the time at which results from the simulation are first printed, the simulation itself is always assumed to begin at TIME=0. TMAX is the maximum time for the simulation to last.

READ (1, *, ERR = 14200, END = 900) TSTART, TMAX

NTAPE = 1

DELTAT is the time between result output displays.

READ (1, *, ERR = 14200, END = 14400) DELTAT

NTITLES is the number of titles used to describe this simulation.

READ (1, *, ERR = 14200, END = 14400) NTITLES

DO 500 I = 1, NTITLES

```

0000      ITITLE is a character array containing text which describes
0000      the current simulation to the user.

0000      READ (1, 400, ERR = 14200, END = 14400) (ITITLE(I, J), J = 1,
0000      1      4TITLFL)
400      FORMAT (<4TITLFL>A2)
500      CONTINUE

0000      Read other file names from the input file and open these files.

0000      READ (1, 200, ERR = 14200, END = 14400) (NAMEOPF(I), I = 1, 5)
0000      OPEN (UNIT = 3, NAME = NAMEOPF, TYPE = 'NEW', ERR = 14900, RECORDSIZE =
0000      1      133)
0000      READ (1, 200, ERR = 14200, END = 14400) (NAMEDBF(I), I = 1, 5)
0000      OPEN (UNIT = 2, NAME = NAMEDBF, TYPE = 'UNKNOWN', ERR = 14600, ACCESS =
0000      1      'DIRECT', RECORDSIZE = 1)

0000      Read the total number of targets, aim points and target status changes
0000      into JIARS, JAIMS and JSTATCS respectively.

0000      READ (1, 600, ERR = 14200, END = 14400) JIARS, JAIMS, JSTATCS,
0000      1      (NAMESTDF(I), I = 1, 5), (NAMEADF(I), I = 1, 5)
600      FORMAT (3I3, 10A2, 10A2)
0000      OPEN (UNIT = 4, NAME = NAMESTDF, TYPE = 'UNKNOWN', ERR = 14700, ACCESS =
0000      1      'DIRECT', RECORDSIZE = 4TREC SZ, FORM = 'FORMATTED', RECORDTYPE
0000      2      = 'FIXED')
0000      OPEN (UNIT = 7, NAME = NAMEADF, TYPE = 'UNKNOWN', ERR = 14800, ACCESS =
0000      1      'DIRECT', RECORDSIZE = 4TREC SZ, FORM = 'FORMATTED', RECORDTYPE
0000      2      = 'FIXED')
0000      GO TO 3100
700      FORMAT (I4, 5A2, 6F7.3, I2, I4, F7.2, <NAMEIMPR>I3, <NAMECOLAT>I3)
900      FORMAT (I3, 5A2, 4F7.3, <NAMEHIGA>I3)

0000      Receive input interactively, an empty input file was found. If an end-
0000      of-file indicator(CTRL Z) is given in response to the next question,
0000      the initialization of aim point, target and probability data bases may
0000      be performed. Examination or modification of specific values in the
0000      probability data base can also be accomplished.

0000      900      TYPE *, ' START TIME, END TIME FOR SIMULATION OUTPUT?'
0000      READ (5, *, ERR = 900, END = 1000) TSTART, TMAX
0000      IF (TSTART .GE. 0.0) GO TO 2200
0000      CLOSE (UNIT = 1)
0000      GO TO 100

0000      1000     TYPE *, ' 1-INITIALIZE PROBABILITY DATA BASE'
0000      TYPE *, ' 2-INITIALIZE TARGET DATA BASE'
0000      TYPE *, ' 3-INITIALIZE AIM POINT DATA BASE'
0000      TYPE *, ' 4-EXAMINE PROB. DATA BASE VALUE'
0000      TYPE *, ' 5-MODIFY PROB. DATA BASE VALUE'
0000      READ (5, *, ERR = 1000, END = 900) I
0000      GO TO (1100, 1300, 1500, 1700, 2000)I
0000      GO TO 1000

0000      1100     TYPE *, ' NAME OF NEW PROBABILITY DATA BASE FILE?'
0000      READ (5, 200, ERR = 1100, END = 1100) (NAMEDBF(I), I = 1, 5)
0000      OPEN (UNIT = 2, NAME = NAMEDBF, TYPE = 'UNKNOWN', ERR = 14600, ACCESS =

```

```

1      "DIRECT", RECORDSIZE = 1)
NRECMAX = NTHA*(NTARGS-1)+NTARGS*(NTHPTS-1)+NTARGS
DO 1200 I = 1, NRECMAX
    WRITE (2'I) FLAG
1200  CONTINUE
    CLOSE (UNIT = 2)
    GO TO 1000
1300  TYPE *, " NAME OF NEW TARGET DATA BASE FILE?"
    READ (5, 200, ERR = 1300, END = 1300) (NAMEADF(I), I = 1, 5)
    OPEN (UNIT = 4, NAME = NADF, TYPE = "UNKNOWN", ERR = 14700, ACCESS =
1      "DIRECT", RECORDSIZE = NRECSZ, FORM = "FORMATTED", RECORDTYPE
2      = "FIXED")
    DO 1400 J = 1, NTARGS
        WRITE (4'J, 700) N1, (NAMEADF(I), I = 1, 5), (Z, I = 1, 6), NZ,
1      NZ, Z, (NZ, I = 1, NTHPTS), (NZ, I = 1, NTHLAT)
1400  CONTINUE
    CLOSE (UNIT = 4)
    GO TO 1000
1500  TYPE *, " NAME OF NEW AIM POINT DATA BASE FILE?"
    READ (5, 200, ERR = 1500, END = 1500) (NAMEADF(I), I = 1, 5)
    OPEN (UNIT = 7, NAME = NADF, TYPE = "UNKNOWN", ERR = 14800, ACCESS =
1      "DIRECT", RECORDSIZE = NRECSZ, FORM = "FORMATTED", RECORDTYPE
2      = "FIXED")
    DO 1600 J = 1, NTHPTS
        WRITE (7'J, 800) N1, (NAMEADF(I), I = 1, 5), (Z, I = 1, 4), (NZ, I
1      = 1, NTHLAT)
1600  CONTINUE
    CLOSE (UNIT = 7)
    GO TO 1000
1700  TYPE *, " NAME OF PROBABILITY DATA BASE FILE?"
    READ (5, 200, ERR = 1700, END = 1700) (NAMEADF(I), I = 1, 5)
    OPEN (UNIT = 2, NAME = NADF, TYPE = "UNKNOWN", ERR = 14600, ACCESS =
1      "DIRECT", RECORDSIZE = 1)
1800  TYPE *, " ENTER: VFIRER, NTHPTS, NTARGET."
    READ (5, *, ERR = 1800, END = 1800) I, J, K
    NREC = NTHA*(I-1)+NTARGS*(J-1)+K
    READ (2'NREC) PROB
    WRITE (5, 1900) PROB
1900  FORMAT (' PROB.= ', F10.7)
    CLOSE (UNIT = 2)
    GO TO 1000
2000  TYPE *, " NAME OF PROBABILITY DATA BASE FILE?"
    READ (5, 200, ERR = 1700, END = 1700) (NAMEADF(I), I = 1, 5)
    OPEN (UNIT = 2, NAME = NADF, TYPE = "UNKNOWN", ERR = 14600, ACCESS =
1      "DIRECT", RECORDSIZE = 1)
2100  TYPE *, " ENTER: VFIRER, NTHPTS, NTARGET, PROB."
    READ (5, *, ERR = 2100, END = 2100) I, J, K, PROB
    NREC = NTHA*(I-1)+NTARGS*(J-1)+K
    WRITE (2'NREC) PROB
    CLOSE (UNIT = 2)
    GO TO 1000
2200  TYPE *, " TIME INTERVAL AT WHICH RESULTS ARE DISPLAYED?"
    READ (5, *, ERR = 2200, END = 2200) DELTAT
    I = (TMAX-TSTART)/DELTAT
    IF (I .GT. NTHPTS) GO TO 2200

```

```

2300  TYPE *, ' NUMBER OF HARDWARE OPTION TITLES?'
      READ (5, *, ERR = 2300, END = 2300) NTITLES
      IF (NTITLES .GT. MTITLES) GO TO 2300
      DO 2600 I = 1, NTITLES
2400      TYPE *, ' TITLE ', I, '?'
          READ (5, 2500, ERR = 2400, END = 2400) (ITITLE(I, J), J = 1,
1          MTITLE)
2500      FORMAT (<'TITLE>A2)
2600  CONTINUE
2700  TYPE *, ' NAME OF FILE TO CONTAIN RESULTS?'
      READ (5, 200, ERR = 2700, END = 2700) (NAMEOPF(I), I = 1, 5)
      OPEN (UNIT = 3, NAME = VOPF, TYPE = 'UNKNOWN', ERR = 14900, RECORDSIZE
1          = 133)
2800  TYPE *, ' NAME OF PROBABILITY DATA BASE FILE?'
      READ (5, 200, ERR = 2800, END = 2800) (NAMEDBF(I), I = 1, 5)
      OPEN (UNIT = 2, NAME = VDBF, TYPE = 'UNKNOWN', ERR = 14600, ACCESS =
1          'DIRECT', RECORDSIZE = 1)
2900  REWIND 1
3000  TYPE *, ' TARGET DATA FILE NAME?'
      READ (5, 200, ERR = 3000, END = 3000) (NAMEPDF(I), I = 1, 5)
      OPEN (UNIT = 4, NAME = VPDF, TYPE = 'UNKNOWN', ERR = 14700, ACCESS =
1          'DIRECT', RECORDSIZE = NTRECSZ, FORM = 'FORMATTED', RECORDTYPE
2          = 'FIXED')
3100  NTARS = 0
      IF (NTAPE .EQ. 1) GO TO 3200
      TYPE *, ' ENTER TARGET INFORMATION.'
3200  NTARS = NTARS+1
      JFIREF = 0
      JTARCF = 0
3300  IF (NTAPE .EQ. 5) GO TO 3600
      IF (ISTATCF .EQ. 0) GO TO 3400
      IF (JSTATCS .LT. NSTATCS) GO TO 11200
      READ (1, *, ERR = 14200, END = 14400) NU-TAR(NTARS), STATUS(NSTATCS,
1          1), STATUS(NSTATCS, 2), ISIATUS(NSTATCS, 1)
      GO TO 3500
3400  IF (JTARS .LT. NTARS) GO TO 3400
      READ (1, *, ERR = 14200, END = 14400) NU-TAR(NTARS)
3500  IF (IDBUG .EQ. '4') TYPE *, ' TARGET NUMBER: ', NU-TAR(NTARS)
      GO TO 3700
3600  TYPE *, ' TARGET NUMBER?'
      READ (5, *, ERR = 3300, END = 3400) NU-TAR(NTARS)
3700  READ (4'NU-TAR(NTARS), 700) ISKIP, (ITARDES(NTARS, I), I = 1, 5),
1          (IARCHAR(NTARS, I), I = 1, 6), (AIYPE(NTARS), NROUNDS(NTARS),
2          EVENTH(NTARS), (NAIMPRY(NTARS, I), I = 1, NAIMPRY),
3          (MICOLAT(NTARS, I), I = 1, MICOLAT)
      IF (ISKIP .EQ. -1) GO TO 4100
3800  CONTINUE
      IF (NTAPE .EQ. 1 .AND. IDBUG .EQ. 'R') GO TO 7500
      TYPE *, ' 1-DESCRIPTION, 2-TARGET HEIGHT, WIDTH'
      TYPE *, ' 3-RANDOM TARGET AREA LOWER COORDINATE'
      TYPE *, ' 4-RANDOM TARGET AREA HEIGHT, WIDTH'
      TYPE *, ' 5-TARGET AREA LOWER COORDINATE'
      TYPE *, ' 6-WEAPON TYPE, 7-AMMUNITION, 8-RESPONSE TIME'
      TYPE *, ' 9-AIM POINTS, 10-COLLATERAL DAMAGE TARGETS'
      READ (5, *, ERR = 3300, END = 7500) I

```

```

GO TO (3900, 4300, 4600, 5100, 5300, 5900, 6200, 6500, 7000, 7300), I
GO TO 3800
3900 WRITE (6, 4000) (ITARDES(NTARS, I), I = 1, 5)
4000 FORMAT (' TARGET DESCRIPTION: ', 5A2)
4100 TYPE *, ' TARGET DESCRIPTION?'
READ (5, 20), ERR = 4100, END = 4200) (ITARDES(NTARS, I), I = 1, 5)
4200 IF (ISKIP .EQ. -1) GO TO 4400
GO TO 3800
4300 TYPE *, ' TARGET HEIGHT, WIDTH: ', TARCHAR(NTARS, 1),
1 TARCHAR(NTARS, 2)
4400 TYPE *, ' TARGET HEIGHT, WIDTH?'
READ (5, *, ERR = 4400, END = 4500) TARCHAR(NTARS, 1), TARCHAR(NTARS,
1 2)
JTARGE = 1
4500 IF (ISKIP .EQ. -1) GO TO 4700
GO TO 3800
4600 TYPE *, ' RANDOM TARGET AREA, LOWER: X COOR., Y COOR.: ',
1 TARCHAR(NTARS, 3), TARCHAR(NTARS, 4)
GO TO 4700
4700 TYPE *, ' RANDOM OR FIXED TARGET AREA? F=FIXED, R=RANDOM'
READ (5, 4600, ERR = 4700, END = 4700) ICHAR
4800 FORMAT (A1)
IF (ICHAR .NE. 'F' .AND. ICHAR .NE. 'R') GO TO 4700
IF (ICHAR .EQ. 'F') GO TO 5400
4900 TYPE *, ' RANDOM TARGET AREA, LOWER: X COOR., Y COOR.?'
READ (5, *, ERR = 4700, END = 5000) TARCHAR(NTARS, 3), TARCHAR(NTARS,
1 4)
JTARGE = 1
5000 IF (ISKIP .EQ. -1) GO TO 5200
GO TO 3800
5100 TYPE *, ' RANDOM TARGET AREA HEIGHT, WIDTH: ', TARCHAR(NTARS, 5),
1 TARCHAR(NTARS, 6)
5200 TYPE *, ' RANDOM TARGET AREA HEIGHT, WIDTH?'
READ (5, *, ERR = 5200, END = 5600) TARCHAR(NTARS, 5), TARCHAR(NTARS,
1 6)
JTARGE = 1
GO TO 5600
5300 TYPE *, ' TARGET AREA, LOWER: X COOR., Y COOR. ', TARCHAR(NTARS,
1 3), TARCHAR(NTARS, 4)
5400 TYPE *, ' TARGET AREA, LOWER: X COOR., Y COOR.?'
READ (5, *, ERR = 5400, END = 5500) TARCHAR(NTARS, 3), TARCHAR(NTARS,
1 4)
JTARGE = 1
5500 TARCHAR(NTARS, 5) = 0.0
TARCHAR(NTARS, 6) = 0.0
5600 IF (ISKIP .EQ. -1) GO TO 3800
5700 TYPE *, ' CAN TARGET FIRE? Y=YES, N=NO'
READ (5, 4800, ERR = 5700, END = 5700) ICHAR
IF (ICHAR .NE. 'Y' .AND. ICHAR .NE. 'N') GO TO 5700
IF (ICHAR .EQ. 'Y') GO TO 5800
NATYPE(NTARS) = 0
EVENTIM(NTARS) = 999.0
GO TO 7400
5900 IF (ISKIP .EQ. -1) GO TO 6000
5900 TYPE *, ' WEAPON TYPE: ', NATYPE(NTARS)

```

```

6000  TYPE *, ' WEAPON TYPE?'
      READ (5, *, ERR = 6000, END = 6100) NWTTYPE(NTARS)
      JFIREF = 1
      IF (NWTTYPE(NTARS) .LE. 0) EVENTTM(NTARS) = 9999.0
      IF (NWTTYPE(NTARS) .GT. NTYPES) GO TO 6000
6100  IF (ISKIP .EQ. -1) GO TO 6300
      GO TO 3800
6200  TYPE *, ' NUMBER OF ROUNDS AVAILABLE: ', NROUNDS(NTARS)
6300  TYPE *, ' NUMBER OF ROUNDS AVAILABLE?'
      READ (5, *, ERR = 6300, END = 6400) NROUNDS(NTARS)
6400  IF (ISKIP .EQ. -1) GO TO 6600
      GO TO 3800
6500  IF (ISTATCF .EQ. 1) GO TO 6700
      TYPE *, ' RESPONSE TIME: ', EVENTTM(NTARS)
6600  IF (ISTATCF .EQ. 0) GO TO 6800
6700  EVENTTM(NTARS) = 0.0
      GO TO 6900
6800  TYPE *, ' RESPONSE TIME?'
      READ (5, *, ERR = 6800, END = 6900) EVENTTM(NTARS)
6900  IF (ISKIP .EQ. -1) GO TO 7100
      GO TO 3800
7000  TYPE *, ' AIM POINTS(ORDERED BY PRIORITY): ', (NAIMPRY(NTARS, I),
1      I = 1, NAIMPRY)
7100  TYPE *, ' AIM POINTS(ORDERED BY PRIORITY)?'
      READ (5, *, ERR = 7100, END = 7200) (NAIMPRY(NTARS, I), I = 1,
1      NAIMPRY)
7200  IF (ISKIP .EQ. -1) GO TO 7400
      GO TO 3800
7300  TYPE *, ' TARGETS DAMAGED BY HITTING THIS TARGET: ',
1      (NITCOLAT(NTARS, I), I = 1, NITCOLAT)
7400  TYPE *, ' TARGETS DAMAGED BY HITTING THIS TARGET?'
      READ (5, *, ERR = 7400, END = 7500) (NITCOLAT(NTARS, I), I = 1,
1      NITCOLAT)
      IF (ISKIP .EQ. -1) GO TO 3800
7500  WRITE (4, 100) NTAR(NTARS), 700) NUTAR(NTARS), (NIPARDES(NTARS, I), I
1      = 1, 5), (NTARCHAR(NTARS, I), I = 1, 6), NWTTYPE(NTARS),
2      NROUNDS(NTARS), EVENTTM(NTARS), (NAIMPRY(NTARS, I), I = 1,
3      NAIMPRY), (NITCOLAT(NTARS, I), I = 1, NITCOLAT)
      IF (ISTATCF .EQ. 0) GO TO 7800
      IF (NUTAPE .EQ. 1) GO TO 7700
7600  TYPE *, ' STATUS CHANGE TIME, INITIAL DELAY, OLD TARGET NUMBER?'
      READ (5, *, ERR = 7600, END = 7600) STATUS(NSTATICS, 1),
1      STATUS(NSTATICS, 2), ISTATUS(NSTATICS, 1)
7700  ISTATUS(NSTATICS, 2) = NUTAR(NTARS)
      NSTATICS = NSTATICS+1
7800  IF (ISKIP .EQ. -1 .OR. IDDEBUG .EQ. 'M') GO TO 3200
      NF = NUTAR(NTARS)
      IF (NUTARGE .EQ. 0) GO TO 8100
      DO 7900 J = 1, NAITPTS
          NREC = NITPA*(I-1)+NTARSG*(J-1)+NF
          WRITE (2, 100) FLAG
7900      CONTINUE
8000  CONTINUE
8100  IF (JFIREF .EQ. 0 .OR. IDDEBUG .EQ. 'M') GO TO 3200

```

```

      GO 4300 I = 1, NAIMPS
      DO 8200 J = 1, NTAAPS
        NREC = NTAAPS*(J-1)+NTARPS*(I-1)+J
        WRITE (2,NREC) FLAG
8200      CONTINUE
8300      CONTINUE
      GO TO 3200
8400      IF (ISTATCF .EQ. 1) GO TO 11200
      NTAAPS = NTAAPS-1
      NAIMS = 0
      IF (NTAPE .EQ. 5) GO TO 8500
      READ (1, 4800, ERR = 14200, END = 14400) ISCHAR
      GO TO 8600
8500      TYPE *, ' AIM POINT DATA FILE NAME?'
      READ (5, 200, ERR = 4500, END = 9500) (NAMEADF(I), I = 1, 5)
      OPEN (UNIT = 7, NAME = NAMEADF, TYPE = 'UNKNOWN', ERR = 14300, ACCESS =
1        'DIRECT', RECORDSIZE = NARECSZ, FORM = 'FORMATTED', RECORNTYPE
2        = 'FIXED')
      TYPE *, ' ENTER AIM POINT INFORMATION.'
8600      NAIMS = NAIMS+1
      JAIMF = 0
8700      IF (NTAPE .EQ. 5) GO TO 8800
      IF (JAIMS .LT. NAIMS) GO TO 11000
      READ (1, *, ERR = 14200, END = 14400) (NMAIMP(NAIMS)
      IF (IPEBUB .EQ. 'N') TYPE *, ' AIM POINT NUMBER ', NMAIMP(NAIMS)
      GO TO 4400
8800      TYPE *, ' AIM POINT NAME?'
      READ (5, *, ERR = 8700, END = 11000) NMAIMP(NAIMS)
8900      READ (7,NMAIMP(NAIMS), 500) YSKIP, (IAIMDES(NAIMS, I), I = 1,
1        5), (AIMCHAR(NAIMS, I), I = 1, 4), (NTHITGA(NAIMS, I), I = 1,
2        NTHITGA)
      IF (ISKIP .EQ. -1) GO TO 9300
9000      CONTINUE
      IF (NTAPE .EQ. 1 .AND. 100000 .EQ. 'R') GO TO 8600
      TYPE *, ' 1-DESCRIPTION'
      TYPE *, ' 2-RANDOM AIM POINT AREA LOWER COORDINATE'
      TYPE *, ' 3-RANDOM AIM POINT AREA HEIGHT, WIDTH'
      TYPE *, ' 4-COORDINATES OF FIXED AIM POINT'
      TYPE *, ' 5-TARGETS WHICH MAY BE HIT BY FIRING AT THIS AIM POINT'
      READ (5, *, ERR = 9000, END = 10700) I
      GO TO (4100, 9500, 9600, 9900, 10100, 10400), I
      GO TO 9000
9100      WRITE (6, 9200) (IAIMDES(NAIMS, I), I = 1, 5)
9200      FORMAT (' AIM POINT DESCRIPTION: ', SA2)
9300      TYPE *, ' AIM POINT DESCRIPTION?'
      READ (5, 200, ERR = 9300, END = 9400) (IAIMDES(NAIMS, I), I = 1, 5)
9400      IF (ISKIP .EQ. -1) GO TO 9400
      GO TO 4000
9500      TYPE *, ' RANDOM AIM POINT AREA LOWER COOR. ', AIMCHAR(NAIMS, 1),
1        AIMCHAR(NAIMS, 2)
      GO TO 9700
9600      TYPE *, ' RANDOM OR FIXED AIM POINT? F=FIXED, R=RANDOM'
      READ (5, 4800, ERR = 9400, END = 9600) ISCHAR
      IF (ISCHAR .EQ. 'F') .AND. ISCHAR .EQ. 'R') GO TO 9600
      IF (ISCHAR .EQ. 'F') GO TO 10200
9700      TYPE *, ' RANDOM AIM POINT NAME, COOR: X COOR., Y COOR?'

```



```

1          F7.2, 30X, I3, 2(1X, F7.3)/,
2          1X, 3X, 1X, 10X, 1X, <MAMPY>(I3, 1X), <MTCOLAT>(1X, I3))
12700 CONTINUE
      WRITE (3, 12200)
      WRITE (3, 12800)
12800 FORMAT (' AIM POINT DATA:')
      DO 13000 I = 1, NAI4S
        WRITE (3, 12900) MINAIMP(I), (IAIMDES(I, J), J = 1, 5),
1          (AINCHAP(I, J), J = 1, 4), (MTHITGA(I, J), J = 1, MTHITGA)
12900 FORMAT (1X, I3, 1X, 5A2, 4(1X, F7.3), <MTHITGA>(1X, I3))
13000 CONTINUE
      WRITE (3, 12200)
      JT = JT+1
      NIARS = NIARS
      NSTATCS = NSTATCS+1
      DO 13200 I = 1, NGROUPS
        DO 13100 J = 1, MTARGS
          ITGROUP(J, I) = 0.0
13100 CONTINUE
13200 CONTINUE
      NTGROUP = 1
13300 IF (NTAPE .EQ. 5) GO TO 13400
      C
      C Read optional target groupings for conditional probability calculation
      C after simulation is complete.
      C
      READ (1, *, ERR = 14200, END = 13900) NTING
      GO TO 13500
13400 TYPE *, ' NUMBER OF TARGETS IN GROUP?'
      READ (5, *, ERR = 13400, END = 13800) NTING
13500 TYPE *, ' TARGETS FOR WHICH CONDITIONAL PROBABILITY SHOULD BE '//
1          ' DISPLAYED?'
      READ (5, *, ERR = 13500, END = 13500) (ITGROUP(I, NTGROUP), I = 1,
1          NTING)
      GO TO 13700
13500 READ (1, *, ERR = 14200, END = 14400) (ITGROUP(I, NTGROUP), I =
1          1, NTING)
13700 NTGROUP = NTGROUP+1
      IF (NTGROUP .GE. NGROUPS) GO TO 13300
13800 NTGROUP = NTGROUP-1
      RETURN
13900 JERROR = 1
      RETURN
14000 WRITE (6, 14100) (NAMEIDF(I), I = 1, 5)
14100 FORMAT (' ERROR OPENING FILE: ', 5A2)
      GO TO 13900
14200 WRITE (6, 14300) (NAMEIDF(I), I = 1, 5)
14300 FORMAT (' ERROR READING FILE: ', 5A2)
      GO TO 13900
14400 WRITE (6, 14500) (NAMEIDF(I), I = 1, 5)
14500 FORMAT (' ERROR - EOF ENCOUNTERED ON: ', 5A2)
      GO TO 13900
14600 WRITE (6, 14100) (NAMEIDF(I), I = 1, 5)
      GO TO 13900
14700 WRITE (6, 14100) (NAMEIDF(I), I = 1, 5)

```

```
      GO TO 13900
14800  WRITE (6, 14100) (NAMEADF(I), I = 1, 5)
      GO TO 13900
14900  WRITE (6, 14100) (NAMEUPF(I), I = 1, 5)
      GO TO 13900
      END
```

SUBROUTINE STATCHG

PURPOSE - Subroutine STATCHG is used to schedule a target status change or group of target status changes. A target status change is a modification to the current qualities that define the target to the model. Any of the items which are associated with a target in the target data base may be modified. Examples of modifications include:

- 1) change in target size or shape,
- 2) increase or decrease in the amount of ammunition available,
- 3) selection of new weapon type,
- 4) change in attack or defense force strategy,
- 5) change in aiming time or weapon firing rate, and
- 6) collateral damage effects.

Status changes will be performed by this routine until the time in which the next status change to occur (STMIN) would exceed the time of the next firing event.

```

INCLUDE "A.INC"
COMMON /L6/ NMTAR(-TARGS)
COMMON /L22/ JERROR
COMMON /L23/ TMAX
COMMON /L25/ NTARS
COMMON /L29/ NF
COMMON /L35/ TMIN
COMMON /L36/ ISTATUS(NSTATC, 2)
COMMON /L37/ STATUS(NSTATC, 2)
COMMON /L38/ NSTATCS
COMMON /L39/ IDXFTR
COMMON /L42/ NSTATC
COMMON /L43/ STMIN

```

Search the status change array to determine the index of the last target having minimum status change time. The index of the next target to undergo a status change is stored in NSTATC and the time at which the status change occurs is stored in STMIN at completion of the following loop.

```

100  DO 300 I = 1, NSTATCS
      IF (STATUS(I, 1)-STMIN) 200, 300, 300
200      STMIN = STATUS(I, 1)
      NSTATC = I
300  CONTINUE

```

If the time at which the next firing event is less than the next status change time return to process firing events.

```

IF (TMIN .LE. STMIN) RETURN

```

NF is the number of the target which is to be changed. This will be considered the old target number. When the status change is made, the old target number in NF is changed indicating that the status change

```

0000      WAS completed.
0000
0000      NF = ISTATUS(NSTATC, 1)
0000
0000      Locate the index of the old target number in the NUMTAR array. If
0000      this target can not be found, an error occurred in the initial input
0000      data and an error message is printed.
0000
0000      DO 400 IDXFIR = 1, NTARS
0000          IF (NF .EQ. NUMTAR(IDXFIR)) GO TO 600
400      CONTINUE
0000      WRITE (6, 500) NF
500      FORMAT (' STATUS CHANGE FOR TARGET ', I3, ' IS NOT POSSIBLE.')
0000      STOP
0000
0000      Subroutine TAREXEC is called to modify desired parameters associated
0000      with the old target number, and assign the new status change target
0000      number indicating the target status change has been successfully
0000      completed.
0000
0000      600      CALL TAREXEC
0000
0000      The time of the next status change is set greater than the total sim-
0000      ulation time forcing a new minimum status change time to be selected
0000      before STATCHG returns.
0000
0000      SPMIN = TMAX+1.0
0000
0000      If an error has occurred previously in TAREXEC, branch to determine the
0000      time and target number for the next status change.
0000
0000      IF (JERRPR .EQ. 0) GO TO 100
700      RETURN
0000      END

```

SUBROUTINE TAREXEC

PURPOSE - Subroutine TAREXEC is called to modify desired parameters associated with the old target number, and assign the new status change number indicating the target status change has been successfully completed.

```

INCLUDE 'A.INC'
COMMON /L5/ EVENTM(MTARGS)
COMMON /L6/ NUMTAR(MTARGS)
COMMON /L7/ ITARDES(MTARGS, 5)
COMMON /L8/ TARCHAR(MTARGS, 6)
COMMON /L9/ MTYPE(MTARGS)
COMMON /L10/ GROUND(MTARGS)
COMMON /L11/ MAMPY(MTARGS, MAMPY)
COMMON /L12/ MCOLAT(MTARGS, MCOLAT)
COMMON /L13/ MUMTAR(MAMPY)
COMMON /L16/ MTHITGA(MAMPY, MTHITGA)
COMMON /L19/ PTARHIT(MTARGS, MAMPY, MTHITGA)
COMMON /L22/ JERRR
COMMON /L25/ TARS
COMMON /L26/ MAMS
COMMON /L29/ IF
COMMON /L30/ LA
COMMON /L31/ MT
COMMON /L32/ PTH
COMMON /L36/ ISTATUS(MSTATC, 2)
COMMON /L37/ STATES(MSTATC, 2)
COMMON /L39/ IOXFR
COMMON /L40/ MIVA
COMMON /L41/ IOXALM
COMMON /L42/ MSTATC
COMMON /L43/ STATV
COMMON /L45/ IDXTAR
CHARACTER *2 NAMEOPF, NAMEIDF, NAMEEDF, NAMEODF, NAMEADF, IAIMDES,
1 ITARDES

```

Obtain the old target number pointing to the target information which was previously used in reference to this target.

```
MFOLD = ISTATUS(MSTATC, 1)
```

Obtain the new number which will be used to access the target information after the status change has occurred. This number is used by TAREXEC to locate the data in the target data base which replaces data obtained by referencing the old target number.

```
MFNEW = ISTATUS(MSTATC, 2)
```

Set MF to the new target number.

```
MF = MFNEW
```

Read information associated with new target number.

```

READ (4*WNEW, 100) (TARCHAW(IXFIR, I), I = 1, 6), N*TYPE(IXFIR),
1  WROUWOS(IXFIR), EVENTTC, (WAI4PRY(IXFIR, I), I = 1,
2  WAI4PRY), (MTCOLAT(IXFIR, I), I = 1, MTCOLAT)
100 FCRMAT (3X, 10X, 6F7.3, 12, 14, F7.2, <WAI4PRY>I3, <MTCOLAT>I3)

A weapon type of zero indicates that this weapon cannot fire (ie - this
is a material type target such as a truck, etc. and not a personnel
target). If this is a material type target, branch.

IF (N*TYPE(IXFIR) .EQ. 0) GO TO 1100

If the time before the next player fires is greater than zero, schedule
this firer into the queue of players waiting to fire by entering the
present time at which the status change is occurring (STMIN) plus the
time before the player fires his first shot (STATUS(WSTATC, 2)).

IF (STATUS(WSTATC, 2) .GT. 0.0) EVENTTC(IXFIR) = STMIN + STATUS(WSTATC,
1  2)

The following section insures that all probabilities associated with
this new firer firing at the targets which he can hit are correct.

IXAIM is used as an index to specify a particular aim point which the
firer can fire at. IXAIM is set to 1 initially to specify the first
aim point.

IXAIM = 1

Obtain the aim point number.
200 NAM = WAI4PRY(IXFIR, IXAIM)

If the aim point number is zero, this indicates that all aim points
which the firer can fire at have been examined. In this case the
branch is taken and all probabilities have been found which involve
this firer firing at other targets.

IF (NAM .EQ. 0) GO TO 1100

IXPAR points to the next target which can be hit by firing at this
aim point (NAM).

IXPAR = 1

It is the target number of the target which can be hit.
300 IT = JTHITGA(NAM, IXPAR)

IF=0 indicates all targets which may be hit by firing at aim point NAM
have been investigated and probabilities of hit have been generated.

IF (IT .EQ. 0) GO TO 1000

WREC is the record number in the probability data base of the record
containing the probability that the firer (WF) fires at aim point NAM

```

```

and hits target NT.
NREC = NTAR*(NF-1)+NTAR*(NAN-1)+NT
The probability of hit is stored in PTH.
400 READ (2*NREC) PTH
    FORMAT (F10,7)

-1. is used as a special flag which indicates that the probability
of hit has not been calculated previously. If PTH is not equal to
-1. it is assumed that it contains the correct probability of hit.

IF (PTH .NE. -1.0) GO TO 400

Locate the index of the target which may be hit in the NUMTAR array.
If the target could not be found, an error has occurred in the input
data and an error message is printed.

DO 500 I = 1, NTARS
    IF (NUMTAR(I) .EQ. NT) GO TO 600
500 CONTINUE
    TYPE *, ' TARGET: ', NT, ' COULD NOT BE LOCATED.'
    GO TO 2400

NT is redefined as the index of the target which may be hit.

600 NT = I

Locate the index of the aim point in the NUMAIMP array. If the index
could not be found, an error message is printed indicating an error in
input data has been detected.

DO 700 J = 1, NAIMS
    IF (NUMAIMP(J) .EQ. NAN) GO TO 800
700 CONTINUE
    TYPE *, ' AIM POINT ', NAN, ' COULD NOT BE FOUND.'
    GO TO 2200

NAN is redefined as the index of the aim point in the NUMAIMP array.

800 NAN = J

GETPTH is called to obtain the probability that firer NF fires at aim
point NUMAIMP(NAN) and hits target NUMTAR(NT).

CALL GETPTH

Store the probability of hit into the data base.

WRITE (2*NREC) PTH

Store the probability of hit into the PTARHIT array for use by the
main program.

```

```

900  PTARHIT(IDXFIR, IDXAIM, IDXTAR) = PTH
    The index to the next target which may be hit by firing at this aim
    point is incremented.
    IDXTAR = IDXTAR+1
    If the index does not exceed the maximum number of targets which may
    be hit by firing at a given aim point, branch to determine probability
    that this target is hit.
    IF (IDXTAR .LE. MTHITGA) GO TO 300
    Increment index pointing to next aim point which firer may choose to
    fire at, probabilities of hit have been determined for all targets
    which may have been hit by firing at the previous aim point.
1000  IDXAIM = IDXAIM+1
    If the number of aim points examined is less than or equal to the max-
    imum number of aim points allowed, branch to obtain probabilities of
    hit for all targets associated with the next aim point, otherwise all
    aim points have been examined and probabilities of hit have been de-
    termined for all targets which the firer may hit.
    IF (IDXAIM .LE. MAXAIM) GO TO 200
    The following section insures that probabilities of hit have been
    found for all firers which can hit this target, where 'this target'
    refers to the target undergoing the status change.
    Save the index of this target.
1100  IDXFSV = IDXFIR
    IDXFIR is used to index the next firer which can hit the status change
    target.
    IDXFIR = 1
    If the weapon type of this firer is 0, this target cannot fire, hence
    it is ignored because it cannot hit the status change target.
1200  IF (WTYPE(IDXFIR) .EQ. 0) GO TO 2500
    Obtain the firer number for the firer indexed by IDXFIR.
    NF = NU4TAR(IDXFIR)
    Ignore examination of the target which did contain the old information
    before this status change was performed.
    IF (NF .EQ. NFOLD) GO TO 2500
    Search through all of the targets which may be collaterally damaged by

```

```

0000      hitting the target pointed to by IDXFIR. Make sure that if the status
0000      change target is one of the targets which may be collaterally damaged,
0000      the probability of hit has been calculated.

0000      DO 1300 I = 1, NCOLLAT
0000
0000          Obtain the index of the next collaterally damaged target.
0000
0000          NIC = NCOLLAT(IDXFIR, I)
0000
0000          If all collaterally damaged targets have been checked, branch.
0000
0000          IF (NIC .EQ. 0) GO TO 1400
0000
0000          Ignore examination of the target which did contain the old infor-
0000          mation before this status change was performed.
0000
0000          IF (NIC .NE. NCOLL) GO TO 1300
0000
0000          Store the new status change target number in place of old target
0000          number in collaterally damaged target array.
0000
0000          NCOLLAT(IDXFIR, I) = NNEW
1300      CONTINUE
0000
0000      This section insures that if this firer should fire at an aim point and
0000      and hit the status change target, that the probability that the status
0000      change target is hit has been computed previously.
0000
0000      IDXAIM is initially set to 1, pointing to the first aim point which may
0000      be hit by this firer.
1400      IDXAIM = 1
0000
0000      Obtain the aim point number.
1500      NAM = NAIMPRY(IDXFIR, IDXAIM)
0000
0000      If the aim point index is zero, all aim points which this firer may
0000      select have been examined.
0000
0000      IF (NAM .EQ. 0) GO TO 2500
0000
0000      The index to the number of targets(IDXTAR) which may be hit by firing
0000      at the current aim point(NAM) is initially set to 1.
0000
0000      IDXTAR = 1
0000
0000      The target number of the next target which may be hit is obtained.
1600      NT = NTHITGA(NAM, IDXTAR)
0000
0000      The branch is taken after all targets which may be hit by firing at
0000      this aim point have been examined.

```

```

IF (NT .EQ. 0) GO TO 2400

If this target number does not match either the old target number or
the new status change target number, then branch. We are only inter-
ested in verifying probabilities of hit associated with these target
numbers and changing the old target number to the new target status
change number.

IF (NT .NE. NFOLD .AND. NT .NE. NFNEW) GO TO 2300

Make sure the new target status change number replaces the old target
number in the NTHITGA array.

NTHITGA(NAM, IDXTAR) = NFNEW

Determine the record number of the record containing the probability
that this new status change target is hit.

NREC = 4PMA*(NF-1)+NTPGS*(NAY-1)+NFNEW

Read the probability that firer NF fires at aim point NAM and hits
the new status change target NFNEW.

READ (2*NREC) PTH
FOR IAF (F10.7)

1700 Branch if the probability of hit has been previously determined.

IF (PTH .IE. -1.0) GO TO 2200

Make sure that NT contains the old target number.

IF (NT .EQ. NFNEW) NT = NFOLD

Find the index of the old target number in the NUMTAR array so that
the new status change target number may be entered in place of the
old target number in this array.

DO 1800 I = 1, NTPRS
  IF (NUMTAR(I) .EQ. NT) GO TO 1900
1800 CONTINUE
TYPE *, ' TARGET: ', NT, ' COULD NOT BE LOCATED.'
GO TO 2900

Set NT to the index in the array NUMTAR pointing to the old target
number.

1900 NT = I

Obtain the index of the aim point number in the NUMAIMP array.

DO 2000 J = 1, NAIMS
  IF (NUMAIMP(J) .EQ. NAM) GO TO 2100
2000 CONTINUE
TYPE *, ' AIM POINT ', NAM, ' COULD NOT BE FOUND.'

```

```

GO TO 2300

Set NAM to the index of the aim point being fired at in the NUMAIMP
array.

2100 NAM = J

Replace the old target number with the new target status change number
in the NUMTAR array.

NUMTAR(NT) = NFNEW

GETPTH is called to obtain the new probability of hit for the firer NF
aiming at the aim point indexed by NAM and hitting the new status
change target indexed by NT.

CALL GETPTH

Store the old target number back into the NUMTAR array so the original
target number will be printed in the simulation output. This statement
should be removed when processing more than double level target sub-
stitution (ie - when one target assumes more than two positions during
the simulation.

NUMTAR(NT) = NFOLD

Store the newly obtained probability of hit into the probability data
base and PIARHIT arrays.

WRITE (2*PREC) PTH
2200 PIARHIT(IXAFIR, IXAIM, IXTAR) = PTH

Increment IXTAR to point to the next target which may be hit by firing
at this aim point.

2300 IXTAR = IXTAR+1

If IXTAR is less than or equal to the next target which may be hit by
firing at this aim point branch to check if this target number matches
either the old target number or the new status change target number.
If IXTAR is greater than the maximum number of targets which may be
hit, all targets associated with this aim point have been checked.

IF (IXTAR .LE. NTHITGA) GO TO 1600

Increment IXAIM to point to the next aim point which may be selected
by this firer.

2400 IXAIM = IXAIM+1

If the number of aim points examined is less than or equal to the max-
imum number of aim points which may be selected, branch to process all
targets which may be hit by firing at the next aim point. If not, all
aim points have been examined for this firer.

```

```

IF (IOXAIM .LE. MAXPPRY) GO TO 1500

Increment the number of firers which have been looked at.

2500 IOXFIR = IOXFIR+1

If the number of firers is less than or equal to the total number of
targets in the simulation, branch and check to see whether the next
firer could hit the new status change target. If not, all possible
firers have been examined and probabilities of another firer hitting
the new status change target have been correctly obtained in the
probability data base.

IF (IOXFIR .LE. NTARS) GO TO 1200

Make sure that all old target numbers have been replaced with the new
target status change number in the array containing targets which may
be hit by firing at a given aim point, NTHITGA. This is necessary
since the firers examined in the above section may not fire at all
aim points in the array and thus the target numbers associated with
these aim points have not been checked.

Loop for all possible aim points.

DO 2700 I = 1, NAIMS

    Loop for all possible targets which may be hit by firing at aim
    point indexed by I.

    DO 2600 J = 1, NTHITGA

        Obtain the target number in NT.

        NT = NTHITGA(I, J)

        If all targets which could be hit by firing at this aim point
        have been checked, branch to examine next aim point.

        IF (NT .EQ. 0) GO TO 2700

        If the target number is not equal to the old target number,
        branch and obtain the next target number associated with this
        aim point.

        IF (NT .NE. NFOLO) GO TO 2600

        Store the new status change target number in place of the old
        target number.

        NTHITGA(I, J) = NNEW

2600 CONTINUE
2700 CONTINUE

Store the new status change target number in place of the old target

```

```

00      number in the NUNPAR array.
00
00000  NUNPAR(IXPSV) = NPNEN
00000
00000  Modify the old target's status change time, indicating that the status
00000  change is complete and the old target will not change status again
00000  during the simulation.
00000
00000  STATUS(NSTATC, 1) = 9999.0
00000  RETURN
2300  JERROR = 1
00000  RETURN
00000  END

```

APPENDIX D

GENERATION OF SHOT HIT POINT

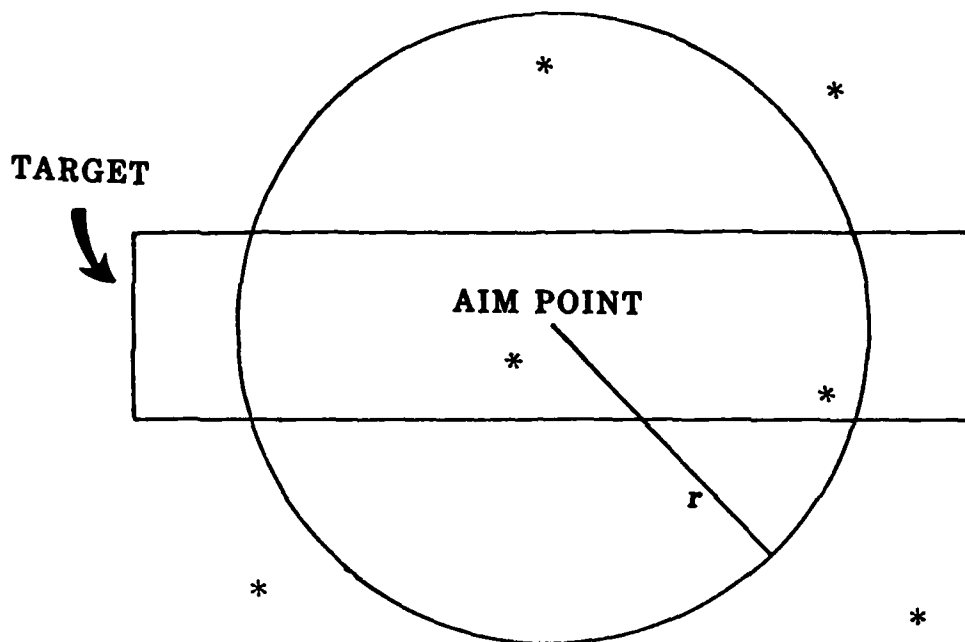
Several factors exist which effect the ability of a firer to hit a particular target at which he may be aiming. Factors effecting a firer's accuracy may be jointly combined to define a probability distribution function which describes the expected dispersion of shots fired at a particular aim point. Accuracy of each weapon type modeled by SAS is defined by a radius (r). The radius defines a circular area centered around the aim point within which one half of all shots fired are expected to land (refer to Figure D.1).

Each shot fired at a particular aim point can be considered to penetrate the target reference plane at a distance r from the aim point (or origin) at an angle θ . The angle θ is measured from the horizontal as shown in Figure D.2. The assumption is made that the firer is equally likely to miss the aim point in any direction. For example, referring to Figure D.2, the probability that he hits above the aim point is equal to the probability he hits below the aim point. By assuming the probability of hit is equal in any direction, the figure could be rotated at any angle, and the probability of hit above the aim point would equal the probability of hit below the aim point. This assumption implies that θ , the angle in which shots are expected to hit around the aim point, is uniformly distributed over all possible values 0° to 360° .

The distance from the aim point to the hit point (r) is generated by selecting x and y coordinates each from a Gaussian normal distribution having mean μ and standard deviation σ . The density function of the normal random variable x is expressed as:

$$n(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1/2 \left\{ \frac{x-\mu}{\sigma} \right\}^2}$$

Since the probability of hit is equally likely in any direction, the mean value for both x and y is zero, and they will possess the same standard deviation. By assuming that the coordinates x and y are independent, their joint density can be expressed as the product of their individual densities.



$$P(\text{Target is hit}) = \frac{\sum (\text{Number of shots hitting target})}{\text{Total number of shots fired}}$$

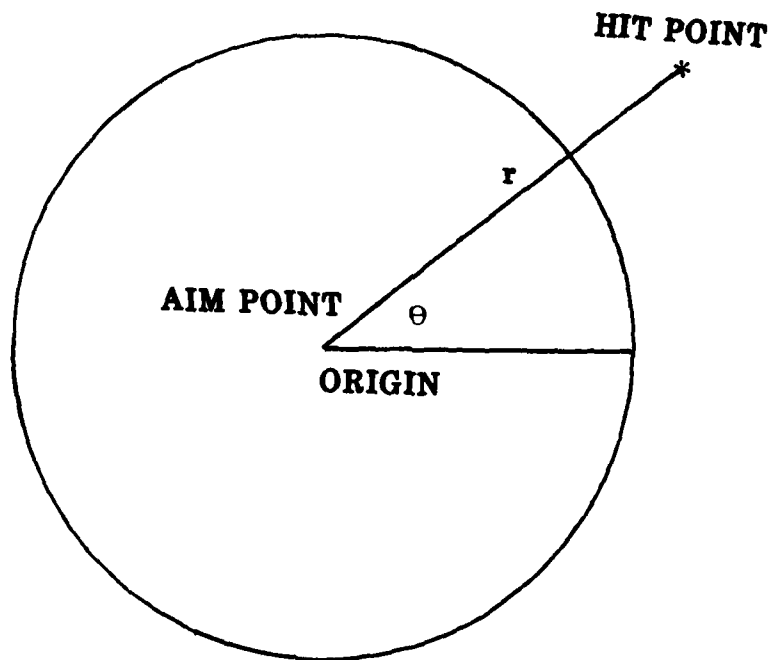
$$= \frac{2}{6} = .33, \text{ for this example.}$$

where,

r - Defines the radius of a circle around the aim point within which one-half of all shots fired are expected to land.

* - Denotes a sample hit point selected at random.

Figure D.1 Illustration of Aim Error



where,

r - Distance from origin to hit point.

θ - Angle of hit point from horizontal.

Figure D.2 Hit Point Expressed in Polar Coordinates

Thus,

$$n(x,y;0,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\left\{\frac{x^2+y^2}{2\sigma^2}\right\}}$$

This density function can be expressed in terms of polar coordinates (r,θ) by use of identities:

$$r^2 = x^2 + y^2, \quad x = r\cos\theta, \quad y = r\sin\theta, \quad \text{and} \quad dx dy = r dr d\theta$$

The density function in terms of the distance r from the aim point is given as :

$$f(r) = \frac{r}{\sigma^2} e^{-\left\{r^2/2\sigma^2\right\}}$$

The density function $f(r)$, known as the Raleigh density, is integrated from 0 to r to obtain the probability that a particular shot will lie within a distance r from the aim point. Performing this integration from 0 to r , where the value of r is the distance from the aim point within which one half of all shots fired are expected to land, allows σ to be expressed as a function of r .

$$\begin{aligned} P(\text{shot falls within } r \text{ of aim point}) &= \int_0^r f(r) dr \\ 0.5 &= \int_0^r \frac{r}{\sigma^2} e^{-\left\{\frac{r^2}{2\sigma^2}\right\}} dr \\ &= -e^{-\left\{\frac{r^2}{2\sigma^2}\right\}} \Big|_0^r \\ &= -e^{-\left\{\frac{r^2}{2\sigma^2}\right\}} + 1 \\ &= e^{-\left\{\frac{r^2}{2\sigma^2}\right\}} \\ \sigma &= r \sqrt{\frac{-1}{2 \ln(.5)}} \\ \sigma &= 0.8493r \end{aligned}$$

Thus, selecting x and y coordinate values from a normal distribution

having mean $\mu = 0$ and standard deviation $\sigma = .8493r$ will insure that one half of all shots fired will land within a distance r from the aim point.

To have the computer generate x and y shot coordinates, random numbers U_1 and U_2 are generated by a standard random number generation algorithm. U_1 and U_2 are selected uniformly from the real interval $[0., 1.]$. These uniform random variables are then used in the following expressions (References 1 and 2) to generate variates which are normally distributed with variance $\sigma = 0.8493r$ and mean $\mu = 0$.

$$x = \cos(2\pi U_2) \sqrt{-1.6986r \log(U_1)}$$

$$y = \sin(2\pi U_2) \sqrt{-1.6986r \log(U_1)}$$

The x and y coordinate values are added to the x and y coordinate values of the aim point to position the shot hit point relative to the origin in the target reference plane. The shot hit point is then checked to see whether or not it penetrated the rectangular target area. If so, a hit is registered. If the weapon has a non-zero lethal kill radius, the distance between the impact point (hit point) of the weapon round within the target reference plane and each corner of the rectangular target area is calculated. If the distance between the hit point and any target area corner is less than the weapon's maximum lethal radius, a hit is registered.

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Defense Intelligence Agency
ATTN: RTS-2C, Tech Svcs & Spt
ATTN: DT-1

Defense Nuclear Agency
ATTN: NASD
4 cy ATTN: TITL

Defense Technical Information Center
12 cy ATTN: DD

Field Command
Defense Nuclear Agency
ATTN: FCTT, G. Ganong
ATTN: FCTT, W. Summa

Field Command
Defense Nuclear Agency
Livermore Branch
ATTN: FCPRL

Field Command
Defense Nuclear Agency
Los Alamos Branch
ATTN: MS-635, FCPRA

DEPARTMENT OF THE ARMY

Harry Diamond Laboratories
Department of the Army
ATTN: DELHD-TD, W. Carter
ATTN: DELHD-NW-RA
ATTN: DELHD-NW-P, F. Balicki

U.S. Army Nuclear & Chemical Agency
ATTN: Library
ATTN: MONA-OPS, LTC Brent
ATTN: Library for MONA-ZB
ATTN: Library for MONA-SAL

U.S. Army TRADOC Sys Analysis Actvy
ATTN: ATAA-TDC, J. Hesse

U.S. Army Training and Doctrine Comd
ATTN: ATOO-NCO
ATTN: ATCD-NN

USAMICOM
Department of the Army
ATTN: DRSMI-YDR Foreign Intelligence Office
ATTN: DRCPM-PE, Pershing Proj Mgr
ATTN: DRSMI-SF

DEPARTMENT OF THE NAVY

Naval Ocean Systems Center
ATTN: Research Library

Naval Postgraduate School
ATTN: Code 1424, Library

DEPARTMENT OF THE AIR FORCE

Headquarters
Air Force Systems Command
ATTN: Maj B. Mills

Air Force Weapons Laboratory
Air Force Systems Command
ATTN: SUL
ATTN: NSSB
ATTN: AFWL SA
ATTN: NTN

Air University Library
Department of the Air Force
ATTN: AUL-LSE

Assistant Chief of Staff
Studies & Analyses
Department of the Air Force
ATTN: AF/SAMI, Tech Info Div

DEPARTMENT OF ENERGY CONTRACTORS

Lawrence Livermore National Lab
ATTN: L-9, D. Blumenthal
ATTN: L-21, M. Gustavson
ATTN: L-9, R. Barker

Los Alamos National Laboratory
ATTN: G. Best

Sandia National Lab
ATTN: 5612, J. Keizur
ATTN: 5613, R. Stratton

DEPARTMENT OF DEFENSE CONTRACTORS

Advanced Research & Applications Corp
ATTN: R. Armistead

AVCO Research & Systems Group
ATTN: J. Gilmore
ATTN: G. Grant

Battelle Memorial Institute
ATTN: D. Hamman

BDM Corp
ATTN: J. Braddock

BDM Corp
ATTN: T. McWilliams

Computer Sciences Corp
ATTN: F. Eisenbarth

General Research Corp
ATTN: P. Lowry
ATTN: H. Schroeder

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Hudson Institute, Inc
ATTN: H. Kahn

JAYCOR
ATTN: E. Almquist
ATTN: R. Sullivan

JAYCOR
4 cy ATTN: C. Clark
4 cy ATTN: J. Humphrey
4 cy ATTN: L. Kennedy
4 cy ATTN: S. Reynolds

Kaman Sciences Corp
ATTN: F. Shelton

Kaman Tempo
ATTN: DASIAC

Lockheed-California Co
ATTN: G. Busch

Mission Research Corp
ATTN: Tech Library

Pacific-Sierra Research Corp
ATTN: H. Brode, Chairman SAGE

R & D Associates
ATTN: S. Cohen
ATTN: A. Lynn
ATTN: P. Haas

S-CUBED
ATTN: K. Pyatt

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Rand Corp
ATTN: W. Jones
ATTN: Library

Santa Fe Corp
ATTN: D. Paolucci

Science Applications, Inc
ATTN: M. Drake
ATTN: J. Martin

Science Applications, Inc
ATTN: W. Layson

SRI International
ATTN: D. Elliott
ATTN: P. Dolan

SRI International
ATTN: R. Foster

Tetra Tech, Inc
ATTN: F. Bothwell

TRW Electronics & Defense Sector
ATTN: N. Lipner

TRW Electronics & Defense Sector
ATTN: P. Dai

Vector Research, Inc
ATTN: S. Bonder