REPORT JTCG/AS-81-S-001

JTCG/AS

ASSESSMENT OF SURVIVABILITY AGAINST LASER THREATS; THE ASALT-I COMPUTER PROGRAM

Frederick J. Steenrod
John E. Musch

September 1981

This document has not been approved by the Joint Logistics Command and does not represent DA[illegible] COM, [illegible], AF[illegible], and [illegible] Commands.

Prepared for

JOINT TECHNICAL COORDINATING GROUP
ON
AIRCRAFT SURVIVABILITY

FOREWORD

This report presents the results of research performed under Naval Weapons Center, China Lake, California Contract N00123-80-D-0033.

The work was sponsored by the JTCG/AS and conducted under the direction of the survivability Assessment Subgroup as Project SA-001.

The contractor was Armament Systems, Inc.

The authors would like to acknowledge the assistance of Carol A. Gillespie, Code 3381, of the Naval Weapons Center, in the understanding and documentation of the ASALT programs.

DISCLAIMER NOTICE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
JTCG/AS-81-S-004

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Assessment of Survivability Against Laser Threats; The ASALT-I Computer Program

5. TYPE OF REPORT & PERIOD COVERED
Final

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Frederick J. Steenrod and John E. Musch

8. CONTRACT OR GRANT NUMBER(s)
N00123-80-D-0033

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Armament Systems, Inc.
712-F North Valley Street
Anaheim, CA 92801

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
SA-001

11. CONTROLLING OFFICE NAME AND ADDRESS
JTCG/AS Central Office, AIR-5164J
Naval Air Systems Command
Washington, D.C. 20361

12. REPORT DATE
September 1981

13. NUMBER OF PAGES
157

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Naval Weapons Center
Code 3381
China Lake, CA 93555

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
~~This document has not been approved by the Joint Logistics Command and does not represent DARCOM, AMC, AFLC and AFSC Commands~~

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

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

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
ASALT
Laser
Computer program
Survivability
Survivability methodology

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

See reverse.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Assessment of Survivability Against Laser Threats; the ASALT-I Computer Program (U), by Frederick J. Steenrod and John F. Musch. Armament Systems Inc. China Lake, Calif., Naval Weapons Center for Joint Technical Coordinating Group/Aircraft Survivability., September 1981. 157 pp. (JTCG/AS-81-S-004, publication UNCLASSIFIED.)

ASALT-I is a FORTRAN computer program used to evaluate the effectiveness of a high-energy laser weapon against an aircraft flying a path previously evaluated for various encounter conditions. The laser weapon system is described by a flux emission function, aiming errors caused by jitter, and slewing limits of the tracking mechanism. The target aircraft is characterized by a set of components which are combined using a fault tree structure. The program output includes a summary for the whole mission which presents probabilities of kill for the total aircraft, its subgroups, and components. This manual contains descriptions for the mathematical concepts, the input requirements, and the output for the ASALT-I program.

ACKNOWLEDGMENT

Development of the ASALT-I computer program began in the spring of 1980 in an effort to fulfill the need for a survivability assessment model which combined the susceptibility of an aircraft being engaged by a laser weapon system with the vulnerability of that aircraft to irradiation. The program was developed and documented by Frederick John Steenrod and John E. Musch, of Armament Systems, Inc. with the guidance and supervision of Carol A. Gillespie and John Morrow of the Naval Weapons Center. Their assistance is gratefully acknowledged.

SUMMARY

The ASALT-I computer program provides a method for evaluating the effectiveness of a high-energy laser beam against an aircraft flying a path previously evaluated for various encounter conditions. The laser weapon system is described by a flux emission function, aiming errors caused by jitter, and slewing limits of the tracking mechanism. The target aircraft is described with a set of components which are combined in a fault tree structure. Each component has a set of rectangular presented areas and Pk functions associated with it. An atmospheric model is used to account for laser beam power degradation before it reaches the target due to interaction with molecules in the air and an optional smoke corridor. The ASALT-I program is used to determine when the laser can be fired and compute the total amount of energy that can be accumulated on each component. The component Pk functions and aircraft fault tree structure are then used to compute the total aircraft probability of kill. The Pk computations can be repeated for as many as 10 distinct aim points and three different fault trees (kill categories) in one program execution. The output of this program may include a time trace of the flight path which shows total aircraft Pk's for each aim point and kill category at regular time intervals in the flight path simulation. In addition, a summary for the whole mission presents the final probabilities of kill for each kill category of the total aircraft model, subgroups in the fault tree structure designated by the user, and each component.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Concluded)

LIST OF TABLES

SECTION I

INTRODUCTION

Improvements in laser and tracking technology have made laser air defense systems a potential threat to combat aircraft in the near term future. In order to fulfill the need to analyze this threat, the model, Assessment of Survivability Against Laser Threats, ASALT-I, is being developed under the cognizance of the Survivability Evaluation Branch, Aircraft Survivability and Lethality Division of the Naval Weapons Center. Recent revisions have been made to the program to provide a more elaborate method of combining components into several levels of redundant or singly vulnerable subgroups. This revision to the ASALT-I documentation includes descriptions of the new input required to define an aircraft fault tree and the new output produced by the model.

The ASALT-I computer program provides a method for evaluating the effectiveness of a high-energy laser against an aircraft flying a path previously evaluated for various encounter conditions. The output from this program is the accumulated aircraft kill probability versus flight path time. In addition, a summary for the whole mission presents the final probabilities of kill for each kill category of the total aircraft model, each subgroup, and each component. Each level of Pk computation can be duplicated for up to 10 distinct aim points. The combination of the Engagement Model[1] and the ASALT-I Model can be used to obtain a survivability estimate for an entire mission involving one high-energy laser weapon attacking one aircraft with consideration given to engagement conditions, tracking requirements, beam propagation, and target vulnerability. The procedures for assessing survivability against a laser air defense system by using these programs are:

1. Generate a flight path for the aircraft. Program FLYGEN[2] is one method of accomplishing this.

2. Select a weapon location and a set of engagement conditions for the laser weapon system.

3. Run the Engagement Model to determine the subsets of the flight path which can be engaged.

---

[1] Steenrod, Frederick J., and Musch, John E., Engagement Model Computer Program (ENGAGE) Analyst/User Manual, Armament Systems Inc., August 1980, Unclassified

[2] Virbila, John P., Aircraft Flight Path Generator Computer Program (FLYGEN), Joint Technical Coordinating Group for Munitions Effectiveness, April 1976, Unclassified.

4. Determine the vulnerability of the aircraft's components to laser radiation at 26 look-angles using a model such as the QKLOOK[3] programs.

5. Run the ASALT-I Model.

This manual contains a description of the mathematical concepts, the input requirements, and an output description for the ASALT-I Model. Section II, the Mathematical Model, is used to explain the coordinate systems, tracking computations, beam propagation model, and method of damage assessment used in this program. The definitions, units, and required order for all input parameters are explained in Section III. Examples of the line printer output and the binary output file are discussed in Section IV. A complete listing of the FORTRAN program, including comment cards, is presented in the appendix.

REQUIREMENTS AND CONSTRAINTS

The ASALT-I Model is written in FORTRAN and requires approximately $140{,}000_8$ (49152_{10}) words of memory on a Hewlett-Packard 3000 computer system. The program structure is modular and flexible so that any changes and/or improvements may be easily implemented. Execution of the program requires two input files and produces two output files. Peripheral device requirements are one card reader, one line printer, and two tape units or other devices for sequential files. Simpler arrangements are possible depending on the computer system. The program in its present form has the following constraints:

1. Only one laser weapon system in a fixed location may be evaluated.

2. The laser flux emission function, which varies with time, may contain a maximum of 10 entries.

3. The atmospheric attenuation function and the corresponding range arguments may contain no more than 10 elements.

4. If a smoke corridor is modeled, its length and location must be defined as a line segment between two end points. The omission of a smoke corridor is allowed.

---

[3] Steenrod, F.J. and Musch, J.E., QKLOOK Computer Programs Analyst and User Manual, JTCG/AS-79-V-008, Joint Technical Coordinating Group on Aircraft Survivability, May 1980

5. The maximum number of aim points on the target is 10.

6. The maximum number of components in the target model is 100.

7. A maximum of three different fault tree structures (kill categories) can be evaluated in one run.

8. The maximum number of elements in any one subgroup is eight.

9. The component vulnerability model requires exactly 10 entries in the function defining Pk at increasing energy levels.

10. Component presented areas and widths are required at 26 standard look-angles.

Some constraints may be overcome by executing the program sequentially several times.

CONCEPTUAL FLOWCHART

The sequence of steps employed in the ASALT-I Model is depicted in Figure 1-1 utilizing a flowchart format. The steps in the flowchart are discussed in the following narrative consisting of paragraphs corresponding to the letters in small hexagons on the flowchart. For the sake of continuity in documentation, Figure 1-1 follows the discussions of all steps.

Step A

Execution of this program begins by reading the data deck from Logical Unit #5 and making some preliminary computations. These data include parameters defining the laser weapon system and its tracking system, the atmospheric conditions, and the aircraft fault tree model.

Step B

This step is the beginning of the time loop in the program. The aircraft flight path data for each new time increment is computed from data on the Flight Path Input File. After all computations for this time increment are completed, program control will return to this step to begin the cycle again for the next time increment. Two tests are made before program control continues with Step C. If the end of the flight path is reached, control branches to Step I to terminate program execution. The second test is used to check the results of the engagement conditions tested during execution of the Engagement Model. If this test indicates that the laser cannot engage the aircraft at the current time, program control branches to Step H, the end of the time loop.

Step C

Step C is the tracking module of the program wherein all conditions involving the weapon's tracking system are evaluated. These conditions include the minimum prefire tracking time and the maximum slewing rates. If either condition is not satisfied, program control branches to Step H, bypassing the laser firing steps. If both tracking conditions are satisfied, the laser flux emission rate is computed from the input parameters defining the weapon system; then program control continues with the next step.

Step D

In this step, any decrease in beam intensity occurring while the beam propagates through the atmosphere to the target is determined, and used to compute the intensity reaching the target. The factors which influence the degradation of beam intensity include an attenuation function which varies with range, and attenuation when the weapon-to-aircraft geometry intersects a smoke corridor.

Step E

This step is the beginning of a loop which iterates for each aim point on the target. The computations inside this loop are used to determine aircraft damage when the laser is directed at the current aim point. Associated with each aim point is an envelope of look-angles which specify the geometrical conditions required to fire at the aim point. If the aim point cannot be hit, program control jumps to the end of the aim point loop at Step G; otherwise execution continues with Step F.

Step F

In this step, the laser energy on each component is accumulated and the resulting damage is evaluated. This is done by executing an inner loop for every component in the target model which includes: computing the expected time for the beam on each component; accumulating the total energy that has reached the component for the current aim point; and determining the damage caused by that level of accumulated energy.

Step G

This is the last step in the aim point loop and is executed only after the component loop has been completed. After the current level of damage for each component has been computed in the preceding step, Subroutine FALTRE is executed which uses the fault tree structure for each kill category to compute the damage to the total aircraft. The decision block in this step represents the end of the aim point loop, branching back to Step E until all aim points have been considered.

Step H

Step H is the last step in the time loop. Aircraft damage up to the current time for each aim point is printed during this step if requested by the user. Program execution then continues with the next time increment at Step B.

Step I

This step is reached only after the entire flight path has been processed and is the concluding step in program execution. In this step a summary of damage to components, subgroups, and the total aircraft is printed for each aim point before program execution halts.

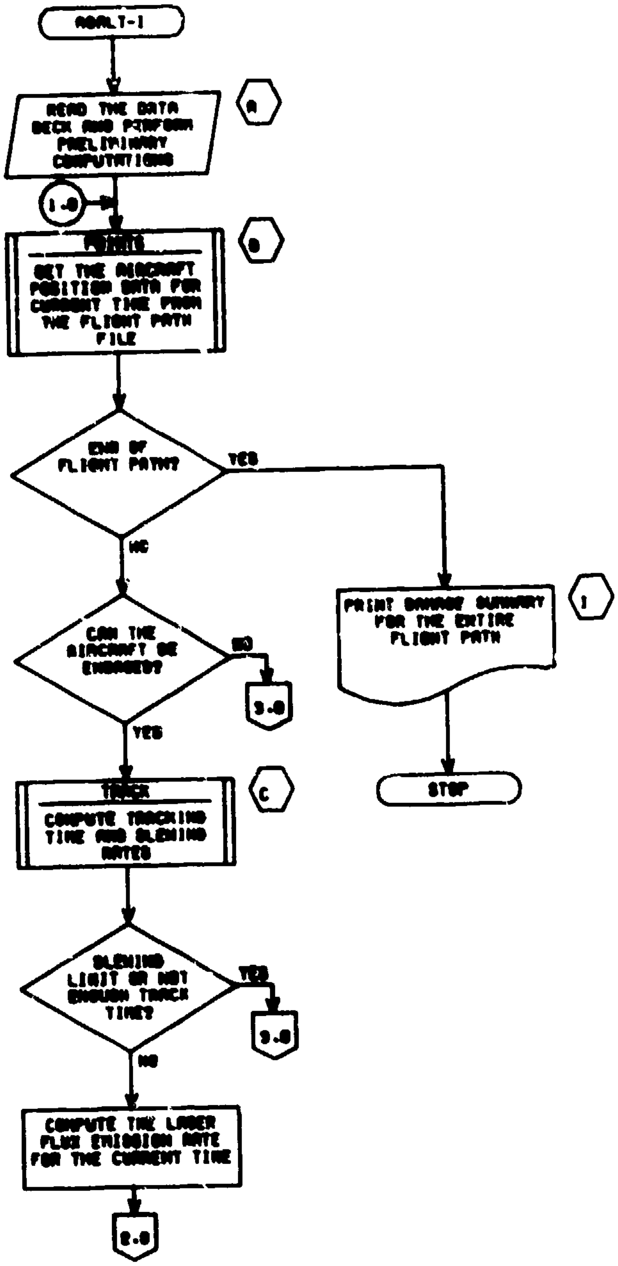


Figure 1-1. ASALT-I Model Conceptual Flowchart (Page 1 of 3).

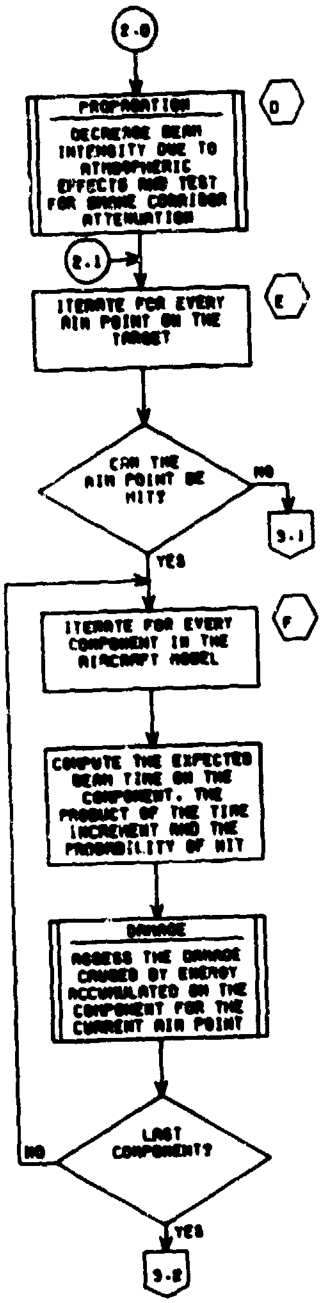


Figure 1-1. ASALT-I Model Conceptual Flowchart (Page 2 of 3).

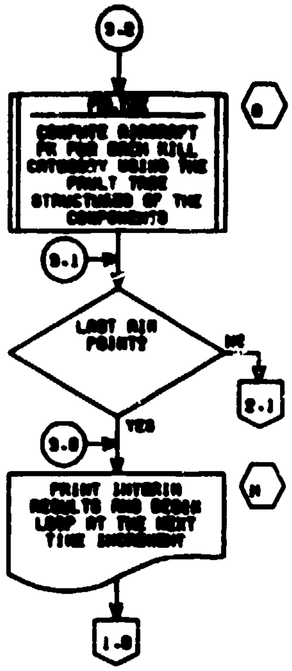


Figure 1-1. ASALT-I Model Conceptual Flowchart (Page 3 of 3).

SECTION II

MATHEMATICAL MODEL

The mathematical concepts used in the ASALT-I Model are presented in this section. There are five subsections of the Mathematical Model as follows:

1. Coordinate Systems
2. Tracking Computations
3. Beam Propagation
4. Damage
5. List of Abbreviations and Symbols.

In the first subsection, geometrical computations used in the program are presented. These include the coordinate systems, transformations between coordinate systems, and look-angle computations. The next three subsections correspond to the three basic modules in the program. The Tracking Computations subsection includes a derivation of the slewing rate computations. In the Beam Propagation subsection, the models for atmospheric attenuation and laser beam intersection with a smoke corridor are presented. The Damage subsection is used to describe the models for accumulating energy on each component and combining component Pk's into total target Pk's using the fault tree. Symbols used in the mathematical equations are defined in the text, and in a complete list in the final subsection.

COORDINATE SYSTEMS

The four coordinate systems used in the ASALT-I Model are depicted in Figure 2-1, where the subscripts on each axis identify the system name. The General Coordinate System is the primary system for this model. It is used for the laser weapon location, the smoke field location, the tracking rate computations, and the printed flight path coordinates. The Aircraft Coordinate System has its origin at the target aircraft center of gravity and is used in defining component and aim point locations on the aircraft. All data on the Flight Path Input File are in the Flight Path Coordinate System, and are transformed into the General Coordinate System as soon as they are read. These three systems are identical to the coordinate systems used in the Engagement Model and have the same names. The Encounter Coordinate System is the only added system. It is used for computations involving the laser beam encountering the target, such as computing the probability of hit for a component. All of these systems have orthogonal right-handed axes.

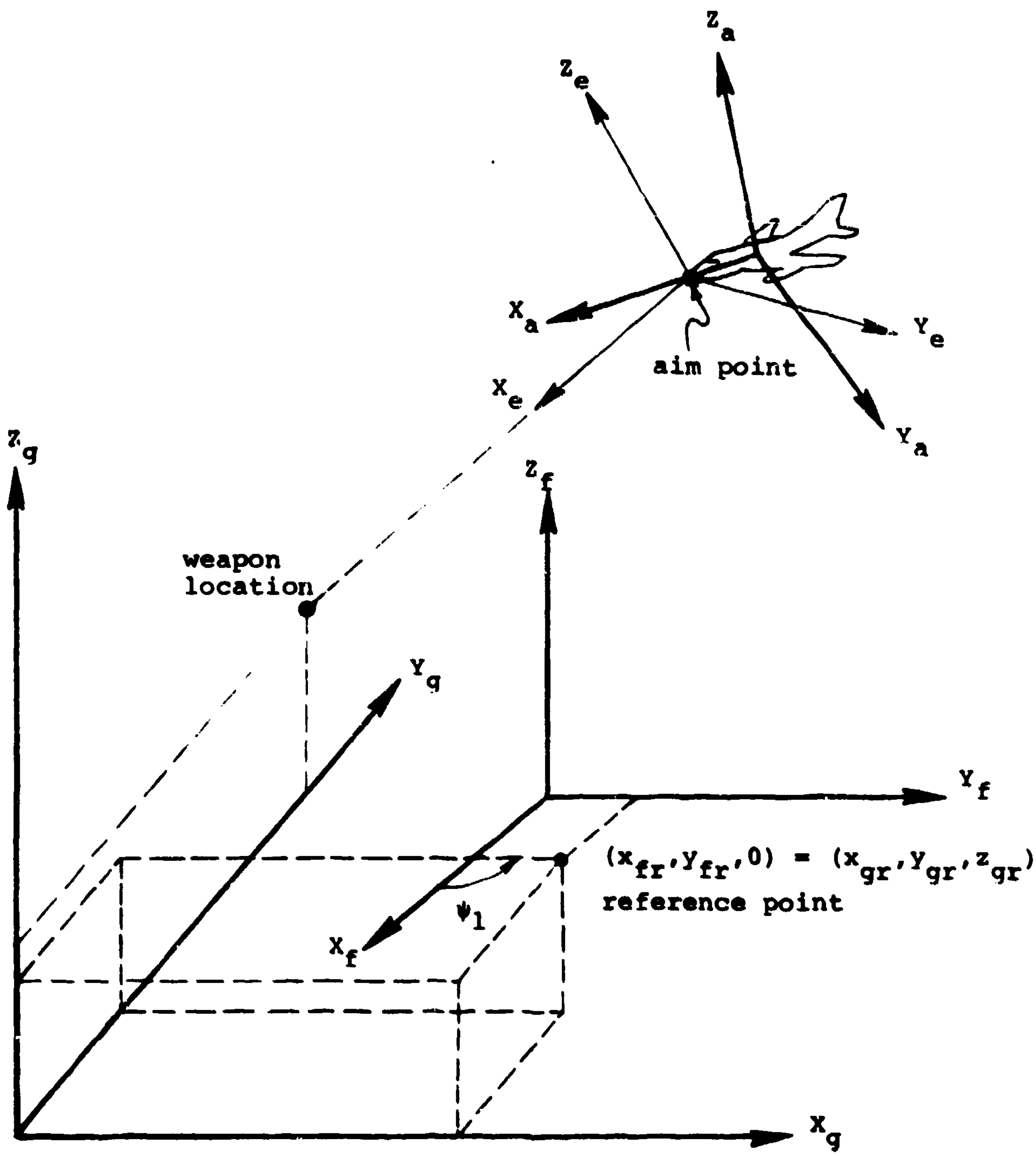


Figure 2-1. General (g), Flight Path (f), Encounter (e), and Aircraft (a) Coordinate Systems.

General Coordinate System

The General Coordinate System has a horizontal XY-plane with the X-axis pointing east and the Y-axis pointing north. The Z-axis points in the direction of increasing altitude. The locations for the laser weapon and the smoke corridor specified in the input data deck are in this coordinate system. Additionally, several computations and the aircraft flight path locations appearing on the line printer output are in the General Coordinate System.

Flight Path Coordinate System

All data on the Flight Path File read from Logical Unit #10 are in the Flight Path Coordinate System. This system must be a right-handed coordinate system, and have a horizontal XY-plane as well as a Z-axis pointing in the direction of increasing altitude. When these conditions are met, the user can facilitate the transformation of data from the Flight Path Coordinate System into the General Coordinate System by suppling the coordinates of a reference point and a rotation angle. For this program the reference point may be any point in the XY-plane of the Flight Path Coordinate Sytem selected by the user.

Let

$(x_{fr}, y_{fr}, 0)$ = reference point coordinates in the Flight Path Coordinate System

(x_{gr}, y_{gr}, z_{gr}) = coordinates of the same reference point in the General Coordinate System

ψ_1 = rotation angle from the X-axis of the Flight Path Coordinate System to the X-axis of the General Coordinate System (a positive rotation is counterclockwise when viewed from above, i.e. the positive Z-axis)

These data supplied by the user on card 2 relate the two Coordinate systems. Any aircraft location in the Flight Path Coordinate System, (x_f, y_f, z_f), can be transformed into an equivalent point in the General Coordinate System, (x_g, y_g, z_g), by executing this equation:

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} x_{gr} \\ y_{gr} \\ z_{gr} \end{bmatrix} + \begin{bmatrix} \cos\psi_1 & \sin\psi_1 & 0 \\ -\sin\psi_1 & \cos\psi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_f - x_{fr} \\ y_f - y_{fr} \\ z_f \end{bmatrix} \qquad (2\text{-}1)$$

where

(x_f, y_f, z_f) = an aircraft location in the Flight Path Coordinate System

(x_g, y_g, z_g) = the location in the General Coordinate System equivalent to (x_f, y_f, z_f)

Similarly any vector, such as the velocity or acceleration vectors, in the Flight Path Coordinate System, (v_{xf}, v_{yf}, v_{zf}), can be rotated into the equivalent vector in the General Coordinate System, (v_{xg}, v_{yg}, v_{zg}), by using this equation:

$$\begin{bmatrix} v_{xg} \\ v_{yg} \\ v_{zg} \end{bmatrix} = \begin{bmatrix} \cos\psi_1 & \sin\psi_1 & 0 \\ -\sin\psi_1 & \cos\psi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} v_{xf} \\ v_{yf} \\ v_{zf} \end{bmatrix} \qquad (2\text{-}2)$$

where

(v_{xf}, v_{yf}, v_{zf}) = a vector in the Flight Path Coordinate System

(v_{xg}, v_{yg}, v_{zg}) = the vector in the General Coordinate System equivalent to the vector (v_{xf}, v_{yf}, v_{zf})

The flight path data also include heading, dive, and roll angles for the aircraft. The dive and roll angles are equivalent in both the Flight Path and General Coordinate Systems. The heading angle must be transformed into the General Coordinate System by executing this equation:

$$\psi = \psi_f - \psi_1 \qquad (2\text{-}3)$$

where

ψ_f = aircraft heading angle in the Flight Path Coordinate System

ψ = aircraft heading angle in the General Coordinate System equivalent to ψ_f

The data on the Flight Path File are transformed to the General Coordinate System by executing Equations 2-1, 2-2, and 2-3 immediately after reading each record during execution of Subroutine READ1U.

Aircraft Coordinate System

The Aircraft Coordinate System has its origin at some fixed point on the aircraft. The X-axis points out the nose of the aircraft, the Y-axis points out the fuselage on the side with the left wing, and the Z-axis points out the top of the aircraft. This system is used to specify component and aim point locations, and to compute look-angles to the target.

Transformations

Vectors are transformed from the General to the Aircraft Coordinate System using a transformation matrix, T, determined by the heading, dive, and roll angles which relate the two coordinate systems. The derivation of matrix T is dependent on the order of the rotation angles and the direction of each angle. The order of rotations used in this program is heading, followed by dive, and then roll.

Figure 2-2 is used to show an arbitrary coordinate system with axes X_1, Y_1, and Z_1 being rotated through the sequence of heading, dive, and roll angles. In the top diagram, the original coordinate system with axes X_1, Y_1, and Z_1 is being rotated through a heading angle, _, to obtain a system with axes X_2, Y_2, and Z_2. This new system is then rotated in the middle diagram through a dive angle, _, resulting in the system with axes X_3, Y_3, and Z_3. Finally the roll angle transformation is shown in the bottom diagram, resulting in the system with axes X_4, Y_4, and Z_4. Figure 2-2 is also used to show the direction of positive heading, dive, and roll angles, along with the corresponding transformation matrices. Let

$(x, y, z)_1$ = vector in the coordinate system with axes X_1, Y_1, and Z_1.

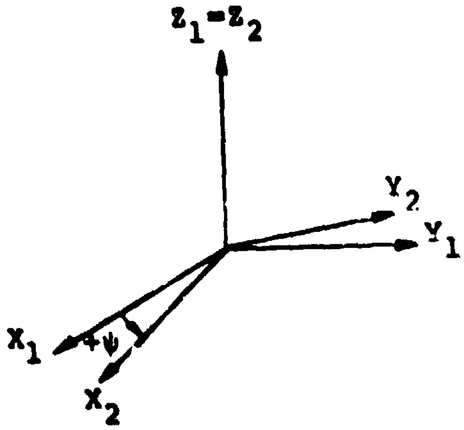
The equivalent vector in the coordinate system with axes X_4, Y_4, and Z_4 can be computed by:

$$(x, y, z)_4 = (x, y, z)_1 * [H] * [D] * [R] \qquad (2\text{-}4)$$

where

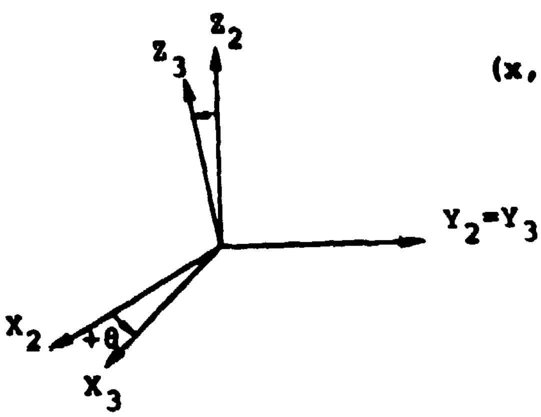
H = heading transformation matrix
D = dive transformation matrix
R = roll transformation matrix

Since matrix multiplication is associative, the vector transformation may use matrix T:



$$(x,y,z)_2 = (x,y,z)_1 * \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Heading Angle Transformation



$$(x,y,z)_3 = (x,y,z)_2 * \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Dive Angle Transformation

$$x,y,z)_4 = (x,y,z)_3 * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

Roll Angle Transformation

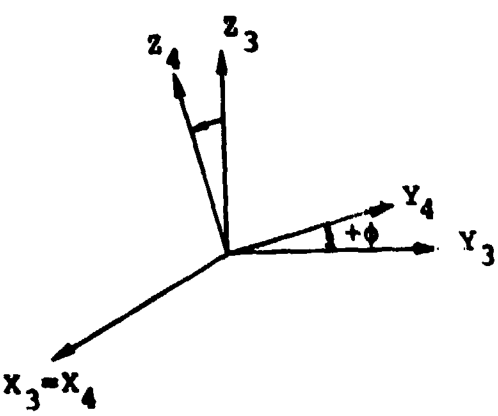


FIGURE 2-2. Heading, Dive and Roll Transformations.

$$(x, y, z)_4 = (x, y, z)_1 * [T] \tag{2-5}$$

where

$$[T] = [H] * [D] * [R] \tag{2-6}$$

Substituting the matrices given in Figure 2-2,

$$[T] = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{2-7}$$

$$[T] = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{2-8}$$

The transformation derived in Equations 2-4 through 2-8 is used in converting from the General Coordinate System to the Aircraft Coordinate System. In Figure 2-2, as well as Equation 2-5, the system with axes X_1, Y_1, and Z_1 corresponds to the General Coordinate System, and the system with axes X_4, Y_4, and Z_4 corresponds to the Aircraft Coordinate System. The transformation matrix in Equation 2-8 is computed and stored by executing Subroutine MATRIX with the heading, dive, and roll angles given as arguments. Since the matrix in Equation 2-8 is orthogonal, its inverse is simply the transpose of matrix T:

$$[T]^{-1} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \tag{2-9}$$

This matrix is also stored when Subroutine MATRIX is executed and is used in transforming from the Aircraft to the General Coordinate Systems. Transformation of any vector from one coordinate system to another is done by executing Subroutine VXMAT with the vector and desired transformation specified in the argument list.

Look-Angles

To compute the presented area for a component, the azimuth and elevation look-angles of the line from the weapon to the component must be computed. In Figure 2-3 the orientations of the look-angles around the aircraft centroid are shown. The azimuth look-angle is measured from the rear of the aircraft in a counterclockwise direction when viewed from the top of the aircraft. The elevation look-angle is measured from the bottom of the aircraft (0.0 degrees) to the top (180.0 degrees). The look-angles to a component use the same orientation, but the system origin is first translated to the component location. The look-angles to a component are computed by converting the vector from the laser location to the aircraft centroid into the Aircraft Coordinate System and adding the vector locating the component on the aircraft.

$$G_{ta} = G_{tg} * [T] \tag{2-10}$$

$$G_{ca} = G_{ta} + C_a \tag{2-11}$$

where

G_{tg} = vector from the laser location to the target center in the General Coordinate System

G_{ta} = vector in the Aircraft Coordinate System equivalent to G_{tg}

C_a = vector locating the component in the Aircraft Coordinate System

G_{ca} = vector from the laser location to the component in the Aircraft Coordinate System with components (c_x, c_y, c_z)

The look-angles to the component are then computed using:

$$A_{lc} = \tan^{-1}(c_y/c_x) \tag{2-12}$$

$$E_{lc} = \pi/2 - \tan^{-1} (c_z/(c_x^2 + c_y^2)^{1/2}) \tag{2-13}$$

where

A_{lc} = azimuth look-angle of the line from the laser location to the component; $0.0 \leq A_{lc} \leq 2\pi$

E_{lc} = elevation look-angle of the line from the laser location to the component; $0.0 \leq E_{lc} \leq \pi$

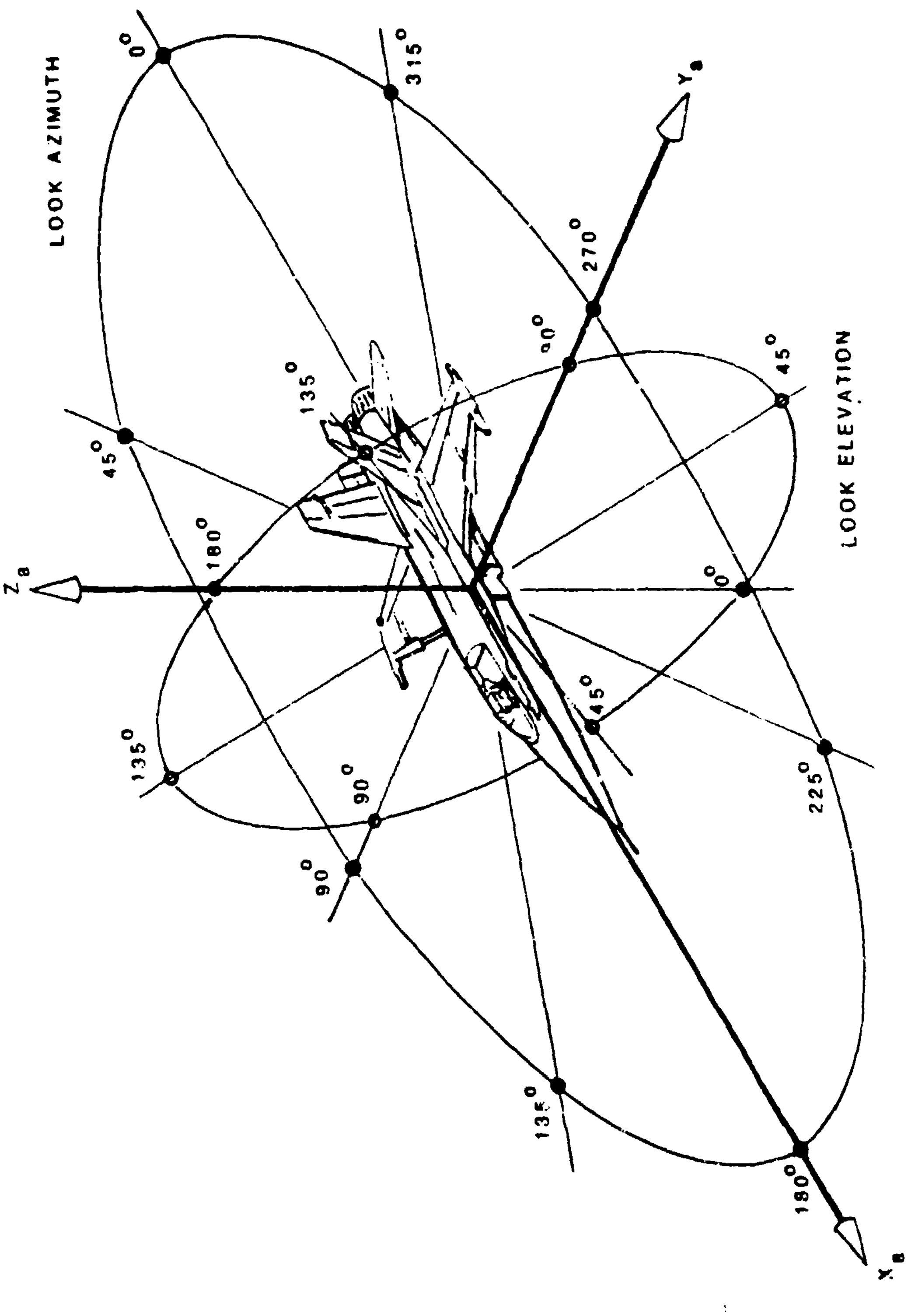


Figure 2-3. Aircraft Coordinate System, Look-azimuth, and Look-elevation.

The look-angles computed by using Equations 2-12 and 2-13 result in angles which, when oriented as shown in Figure 2-3, are the proper angles to point a vector from the component back towards the laser location; that is, the look-angles define the negative of the vector G_{ca}. The relationship of the vector G_{ca} and the look-angles is shown in Figure 2-4. Note that the coordinate system in this figure is obtained by translating the origin of the Aircraft Coordinate System to the component center. The range of values for the angles A_{lc} and E_{lc} is achieved by using the ATAN2 FORTRAN function and some IF statements in Subroutine LOKANG. The azimuth and elevation look-angles to an aim point on the target are computed by the same procedure substituting the aim point location for the component location.

Encounter Coordinate System

The final coordinate system is used for computations involving the laser beam interacting with the target. The Encounter Coordinate System has its origin at one of the aim points on the target. The X-axis points along the line-of-sight toward the laser location, so that the YZ-plane is perpendicular to the line-of-sight. The angular transformation from the Aircraft to the Encounter Coordinate Systems involves a heading rotation, ψ, of the XY-plane, followed by a dive rotation, θ, of the new XZ-plane. The rotation angles are computed using

$$\psi = A_{la} - \pi \tag{2-14}$$

$$\theta = \pi/2 - E_{la} \tag{2-15}$$

where

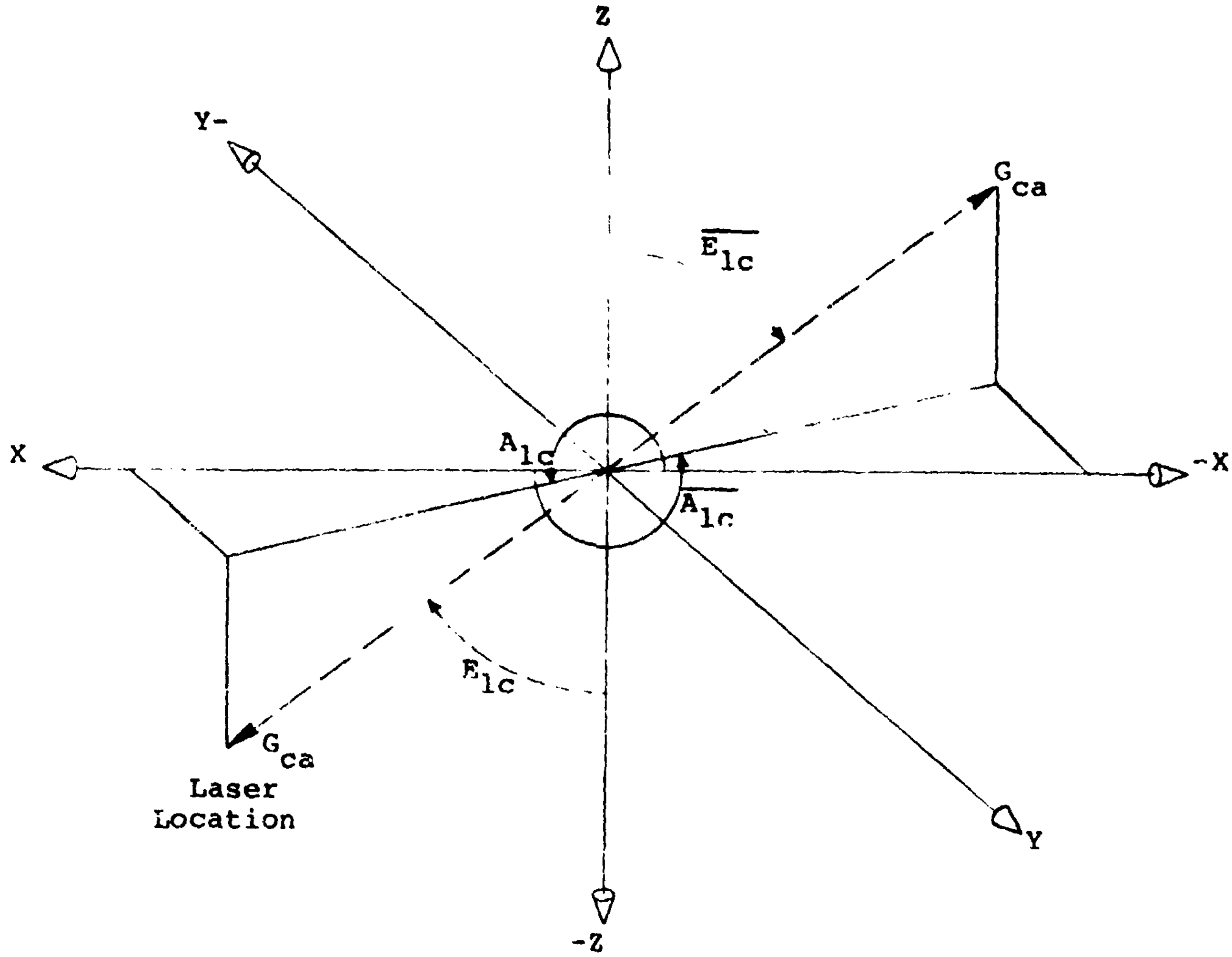
A_{la} = azimuth look-angle of the line from the laser location to the aim point; $0.0 \leq A_{la} \leq 2\pi$

E_{la} = elevation look-angle of the line from the laser location to the aim point; $0.0 \leq E_{la} \leq \pi$

ψ = rotation angle for the XY-plane; $-\pi \leq \psi \leq \pi$

θ = rotation angle for the XZ-plane after rotation through ψ; $-\pi/2 \leq \theta \leq \pi/2$

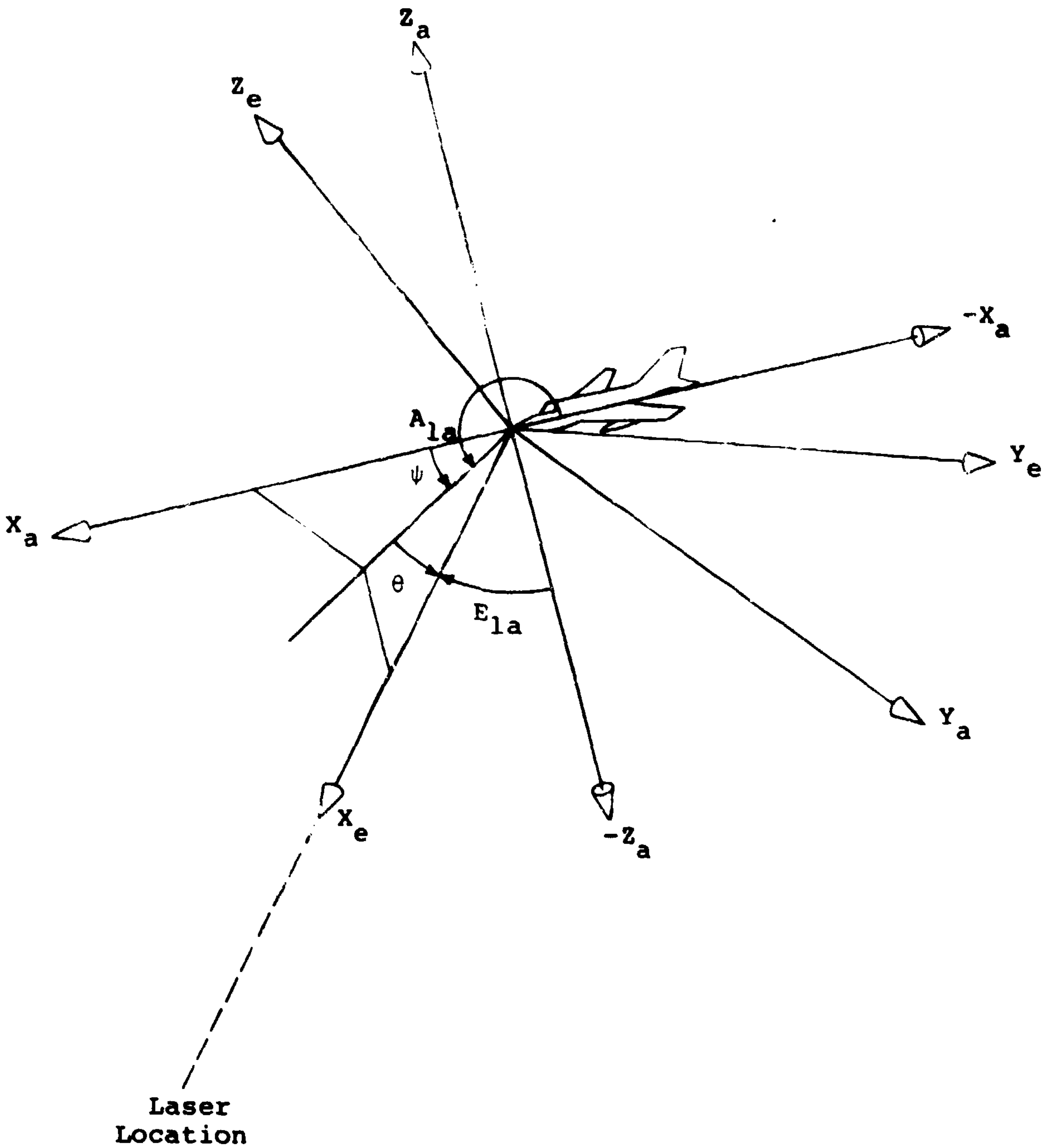
Figure 2-5 is used to show the relationship of the angles ψ, A_{la}, θ and E_{la} in the Encounter Coordinate System. These rotations are equivalent to the first two shown in Figure 2-2. By assigning a roll angle equal to 0.0, the Aircraft to Encounter Coordinate System transformation matrix can be computed using Equation 2-8 with heading and dive angles from Equations 2-14 and 2-15. This is



$\overline{A_{1c}}$ is the angle computed by using Equation 2-12 in the translated coordinate system, and is equivalent to the azimuth look-angle A_{1c}.

$\overline{E_{1c}}$ is the angle computed by using Equation 2-13 in the translated coordinate system and is equivalent to the elevation look-angle E_{1c}.

Figure 2-4. Look-angle Computation.



X_a, Y_a, and Z_a are the Aircraft Coordinate System Axes translated to the aim point.

X_e, Y_e, and Z_e are the Encounter Coordinate System Axes.

Figure 2-5. Rotation Angles into the Encounter Coordinate System.

done by simply invoking Subroutine MATRIX with the new rotation angles as the arguments.

TRACKING COMPUTATIONS

The tracking module in the ASALT-I Model is used to evaluate two conditions which are prerequisites for the simulation of laser firing. The first condition is that the minimum prefire track time must be satisfied. This computation is a simple comparison of time values. The second condition is the comparison of the weapon slewing rates with the maximums for the system established by the user on Card 7 of the input deck. The slewing rates are computed by evaluating the first derivatives of the weapon-to-target azimuth and elevation angles. The azimuth and elevation angles of the vector from the weapon to the target are computed by

$$A_z = \tan^{-1}(y/x) \qquad (2\text{-}16)$$

$$E_l = \tan^{-1}\left(z/(x^2 + y^2)^{1/2}\right) \qquad (2\text{-}17)$$

where

A_z = azimuth angle of the line from the weapon to the target

E_l = elevation angle of the line from the weapon to the target

(x,y,z) = vector from the laser location to the target in the General Coordinate System

Using the prime notation for derivatives with respect to time, the azimuth slew rate equation is

$$A_z' = (y/x)'/[1 + (y/x)^2] \qquad (2\text{-}18)$$

$$A_z' = (xy' - x'y)/(x^2 + y^2) \qquad (2\text{-}19)$$

The elevation slew rate equation is:

$$E_l' = \left\{z/\left[(x^2 + y^2)^{1/2}\right]\right\}' \div \left\{1 + \left\{z/\left[(x^2 + y^2)^{1/2}\right]\right\}^2\right\} \qquad (2\text{-}20)$$

$$E_l' = \left\{z' - z\left[(xx' + yy' + zz')/(x^2 + y^2 + z^2)\right]\right\} * \left[1/(x^2 + y^2)^{1/2}\right] \qquad (2\text{-}21)$$

where

A_z' = azimuth slew rate

E_l' = elevation slew rate

(x',y',z') = rate of change in the aircraft position vector; the aircraft velocity vector

BEAM PROPAGATION

As the laser beam propagates through the air, its power decreases due to interaction with molecules and particulate matter in the atmosphere. The propagation module of the program uses an array of attenuation factors which are a function of range to simulate the effects of atmospheric attenuation on the beam.

One possible countermeasure for use against a laser beam is a smoke corridor. The principle is that a thick cloud of smoke would further degrade the beam power reaching the aircraft. A smoke corridor may be modeled in the program by specifying two end points in the General Coordinate System for the corridor on Card 16 of the input deck. The power degradation due to smoke interference occurs only when the vector from the laser location to the target intersects the line between the smoke corridor end points as shown in Figure 2-6. This intersection requires satisfaction of two mathematical conditions: first, the azimuth angle from the laser to the aircraft must be between the azimuth angles from the laser to the end points of the smoke corridor; second, the range from the laser to the aircraft must be greater than the range from the laser to the point of intersection. The first condition is evaluated by simply comparing azimuth angles. The second condition requires that the point of intersection (x_i, y_i) be determined so that the ranges may be compared. The computation of the intersection point involves the solution of two simultaneous equations. The point of intersection may be expressed as:

$$x_i = x_w + s_w(x_a - x_w) \qquad (2\text{-}22)$$

$$y_i = y_w + s_w(y_a - y_w) \qquad (2\text{-}23)$$

where

x_i = x-coordinate of the point of intersection
y_i = y-coordinate of the point of intersection
x_w = x-coordinate of the weapon location
y_w = y-coordinate of the weapon location
x_a = x-coordinate of the aircraft location
y_a = y-coordinate of the aircraft location

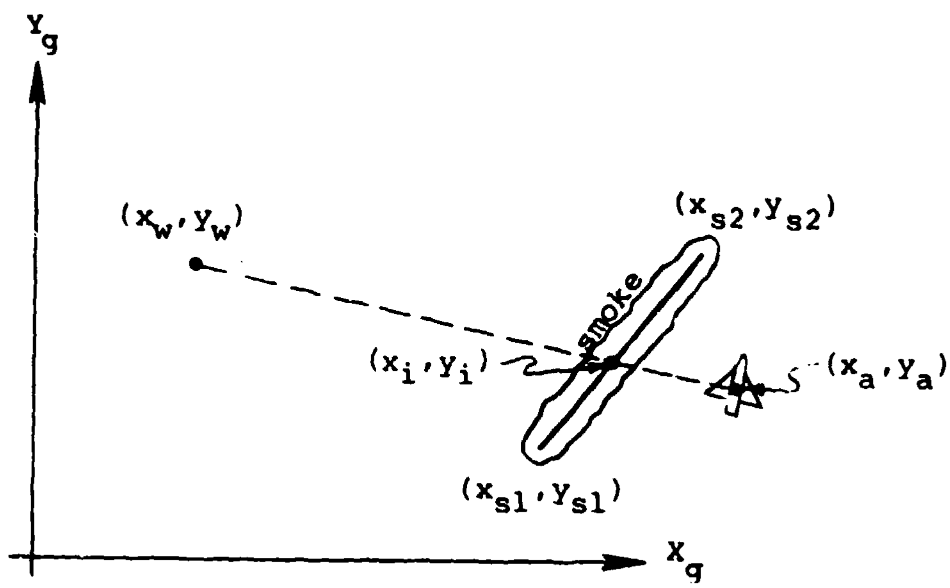


Figure 2-6. Smoke Corridor Geometry.

s_w = fraction of the horizontal distance between the laser and aircraft at which the point of intersection lies

The same point of intersection may be expressed as:

$$x_i = x_{s1} + s_s(x_{s2} - x_{s1}) \tag{2-24}$$

$$y_i = y_{s1} + s_s(y_{s2} - y_{s1}) \tag{2-25}$$

where

x_{s1} = x-coordinate of the smoke corridor first end point
y_{s1} = y-coordinate of the smoke corridor first end point
x_{s2} = x-coordinate of the smoke corridor second end point
y_{s2} = y-coordinate of the smoke corridor second end point
s_s = fraction of the distance between the first and second end points at which the point of intersection lies.

Equations 2-22 and 2-24 as well as 2-23 and 2-25 can be equated resulting in two simultaneous equations with two unknowns, s_w and s_s, as shown below.

$$x_w + s_w(x_a - x_w) = x_{s1} + s_s(x_{s2} - x_{s1}) \tag{2-26}$$

$$y_w + s_w(y_a - y_w) = y_{s1} + s_s(y_{s2} - y_{s1}) \tag{2-27}$$

Solving Equation 2-26 for s_w and substituting into Equation 2-27 results in an expression which can be solved for the term s_s.

$$s_s = \frac{(y_a - y_w)(x_{s1} - x_w) + (x_a - x_w)(y_w - y_{s1})}{(y_{s2} - y_{s1})(x_a - x_w) - (x_{s2} - x_{s1})(y_a - y_w)} \tag{2-28}$$

By evaluating Equation 2-28 first, the fraction s_s, can be substituted into Equations 2-24 and 2-25 resulting in the values for the coordinates of the point of intersection. The rest of the smoke corridor problem consists of simple distance computations and comparisons.

DAMAGE

In order for a laser beam to damage an aircraft, the power in the laser beam must accumulate over a period of time until the total energy absorbed by some component is adequate. In the ASALT-I Model, the user selects up to 10 aim points on the aircraft. Associated with each aim point is an envelope defining the range of look-angles at which the aim point can be hit, and a pair of standard deviations for the errors in locating and holding a beam on the aim point. The probability of hitting a component is computed by determining the rectangular presented area of the component, computing the total standard deviations, and integrating an offset Gaussian probability density function over the component presented area. An example target with components, aim points, and an aim point envelope is shown in Figure 2-7. The probability of hit is multiplied by an integration time interval to determine the expected time duration of the laser beam center on the component of interest. The expected time multiplied by the attenuated beam power results in the amount of energy reaching the component during the time interval. By summing the added energies for each time interval, the total expected energy on the component is obtained. Component Pk is dependent on the total energy accumulated. The component Pk's are then combined using fault tree structures resulting in subgroup and total target Pk's for each kill category. All of the damage computations are evaluated separately for every aim point.

Component Rectangular Presented Area

The user assembling the input data deck must include a location, as well as 26 presented areas and widths for each component on Cards 13 and 14. These data are used in the ASALT-I Program to determine the location and boundaries of the component presented area based on the current weapon-to-aircraft geometry. Each presented area and width pair may be interpreted as the area and horizontal length as seen from the weapon location when the look-angles define the orientation between the weapon and component. In Figure 2-8 the width and presented area for a component are shown for an azimuth look-angle equal to 225 degrees and an elevation look-angle equal to 135 degrees (refer to Figure 2-3 for look-angle orientations). The figure depicts the target as it would be seen from the weapon location.

The 26 standard sets of azimuth and elevation look-angles used in this program are listed in Table 2-1. For an arbitrary set of azimuth and elevation look-angles from the laser location to the component, the presented area and width are interpolated from the 26 sets of input values. Subroutine INT26 is used to select four of the 26 standard look-angles which geometrically surround the current look-angles, and to perform an interpolation between the corresponding sets of presented areas and widths.

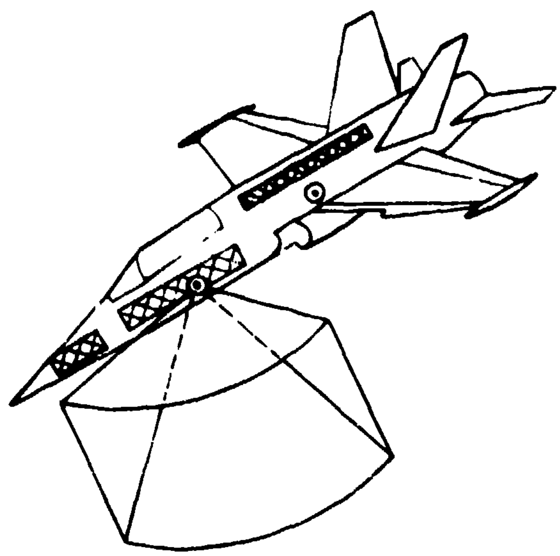


Figure 2-7. Components, Aim Points, and Aim Point Envelopes in the Target Model.

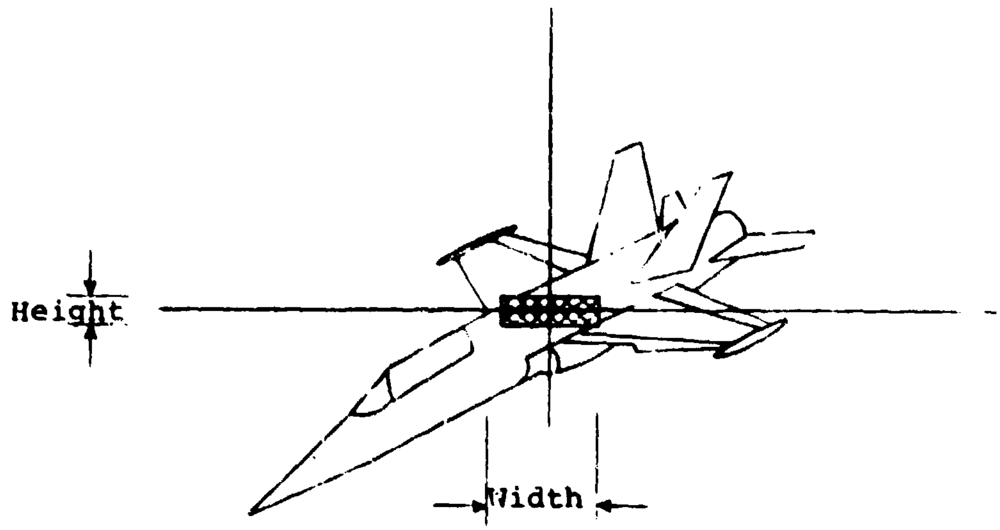


Figure 2-8. Example Component Presented Area and Width.

Table 2-1. Standard Look-angles for Component Presented Areas and Widths.

INDEX	LOOK-ANGLES AZIMUTH	LOOK-ANGLES ELEVATION	INDEX	LOOK-ANGLES AZIMUTH	LOOK-ANGLES ELEVATION
1	0	0	14	180	90
2	0	45	15	225	90
3	45	45	16	270	90
4	90	45	17	315	90
5	135	45	18	0	135
6	180	45	19	45	135
7	225	45	20	90	135
8	270	45	21	135	135
9	315	45	22	180	135
10	0	90	23	225	135
11	45	90	24	270	135
12	90	90	25	315	135
13	135	90	26	0	180

Probability of Hit

The probability of hitting the component is computed assuming the component has a rectangular presented area and that the normal (Gaussian) probability density function describes the accuracy of the beam center hitting the aim point. With these assumptions, the probability of hit is computed by integrating a two-dimensional normal probability density function centered at the aim point, over limits defined by the offset component presented area. The mathematical expression used is:

$$P_H = \frac{1}{2\pi\sigma_y\sigma_z} \int_{y_1 - g_y/2}^{y_1 + g_y/2} \int_{z_1 - g_z/2}^{z_1 + g_z/2} \exp\left[-\frac{y^2}{2\sigma_y^2} - \frac{z^2}{2\sigma_z^2}\right] dzdy \quad (2\text{-}29)$$

where

σ_y = total standard deviation in the direction of the Y-axis of the Encounter Coordinate System

σ_z = total standard deviation in the direction of the Z-axis of the Encounter Coordinate System

y_1 = y-coordinate of the component centroid in the Encounter Coordinate System

z_1 = z-coordinate of the component centroid in the Encounter Coordinate System

g_y = width of the component presented area

g_z = height of the component presented area

P_H = probability of hitting a rectangular component offset from the aim point

The standard deviations in Equation 2-29 are computed by combining standard deviations in aim point location and jitter using these equations

$$\sigma_y = \left(\sigma_{jy}^2 + \sigma_{ay}^2\right)^{1/2} \quad (2\text{-}30)$$

$$\sigma_z = \left(\sigma_{jz}^2 + \sigma_{az}^2\right)^{1/2} \quad (2\text{-}31)$$

where

σ_{jy} = standard deviation due to jitter of the beam in the direction of the Y-axis of the Encounter Coordinate System

σ_{jz} = standard deviation due to jitter of the beam in the direction of the Z-axis of the Encounter Coordinate System

σ_{ay} = standard deviation of the error in locating and tracking the aim point in the direction of the Y-axis of the Encounter Coordinate System

σ_{az} = standard deviation of the error in locating and tracking the aim point in the direction of the Z-axis of the Encounter Coordinate System

The probability of hit, P_H, in Equation 2-29 is computed as the product of two integrals

$$P_H = \left[\frac{1}{\sqrt{2\pi}\,\sigma_y} \int_{y_1 - g_y/2}^{y_1 + g_y/2} \exp\left(-y^2/2\sigma_y^2\right) \right] dy$$

$$* \left[\frac{1}{\sqrt{2\pi}\,\sigma_z} \int_{z_1 - g_z/2}^{z_1 + g_z/2} \exp\left(-z^2/2\sigma_z^2\right) \right] dz \qquad (2\text{-}32)$$

Each of these integrals is evaluated by using a modified version of an approximation from Approximations for Digital Computers by Hastings.[4]

---

[4] Hastings, Cecil Jr., assisted by Hayward, Jeanne T., and Wong, James P. Jr., Approximations for Digital Computers, page 187, Princeton University Press (1955)

Hastings approximates this integral

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt \tag{2-33}$$

by using

$$\Phi(x) = 1 - \left[1\Big/\left(1 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6\right)^{16}\right] \tag{2-34}$$

where

a_1 = 0.0705230784
a_2 = 0.0422820123
a_3 = 0.0092705272
a_4 = 0.0001520143
a_5 = 0.0002765672
a_6 = 0.0000430638

The probability of hit along one axis is one of the factors from Equation 2-32. Using the Y-axis as an example

$$P_{Hy} = \frac{1}{\sqrt{2\pi}\,\sigma_y} \int_{y_1 - g_y/2}^{y_1 + g_y/2} \exp\left(-y^2/2\sigma_y{}^2\right)\, dy \tag{2-35}$$

$$P_{Hy} = \left[\frac{1}{\sqrt{2\pi}\,\sigma_y} \int_{-\infty}^{y_1 + g_y/2} \exp\left(-y^2/2\sigma_y{}^2\right)\, dy\right]$$

$$- \left[\frac{1}{\sqrt{2\pi}\,\sigma_y} \int_{-\infty}^{y_1 - g_y/2} \exp\left(-y^2/2\sigma_y{}^2\right)\, dy\right] \tag{2-36}$$

$$P_{Hy} = P\left[y_1 + (g_y/2)\right] - P\left[y_1 - (g_y/2)\right] \tag{2-37}$$

where

$P(y_1 + g_{y/2})$ = integral of the normal probability density function evaluated from $-\infty$ to $(y_1 + g_y/2)$

$P(y_1 - g_{y/2})$ = integral of the normal probability density function evaluated from $-\infty$ to $(y_1 - g_y/2)$

P_{Hy} = probability of hit within the y directional limits of the component presented area

Since the normal probability density function used in Equations 2-35 and 2-36 has a mean at y=0, and the integral of this function from $-\infty$ to the mean equals one-half, the first term in Equation 2-37 may be written as:

$$P\left[y_1 + (g_y/2)\right] = \frac{1}{2} + \frac{1}{\sqrt{2\pi}\,\sigma_y} \int_0^{y_1 + g_y/2} \exp\left(-y^2/2\sigma_y{}^2\right) dy \quad (2\text{-}38)$$

Substituting

$$t = \frac{y}{\sqrt{2}\,\sigma_y}$$

$$dt = \frac{1}{\sqrt{2}\,\sigma_y}$$

$$P\left(\frac{y_1 + g_y/2}{\sqrt{2}\,\sigma_y}\right) = \frac{1}{2} + \frac{1}{\sqrt{\pi}} \int_0^{\frac{y_1 + g_y/2}{\sqrt{2}\,\sigma_y}} \exp\left(-t^2\right) dt \quad (2\text{-}39)$$

Now let

$$x = \frac{y_1 + g_y/2}{\sigma_y}$$

$$P\left(\frac{x}{\sqrt{2}}\right) = \frac{1}{2} + \frac{1}{2}\left[\frac{2}{\sqrt{\pi}} \cdot \int_0^{\frac{x}{2}} \exp\left(-t^2\right) dt\right] \quad (2\text{-}40)$$

and using the Hastings approximation from Equations 2-33 and 2-34.

$$P\left(\frac{x}{\sqrt{2}}\right) = \frac{1}{2} + \frac{1}{2}\,\Phi\left(\frac{x}{\sqrt{2}}\right) \qquad (2\text{-}41)$$

$$P\left(\frac{x}{\sqrt{2}}\right) = \frac{1}{2} + \frac{1}{2}\left\{1 - 1/\left[1 + a_1\frac{x}{\sqrt{2}} + a_2\left(\frac{x}{\sqrt{2}}\right)^2 + a_3\left(\frac{x}{\sqrt{2}}\right)^3 + a_4\left(\frac{x}{\sqrt{2}}\right)^4 + a_5\left(\frac{x}{\sqrt{2}}\right)^5 + a_6\left(\frac{x}{\sqrt{2}}\right)^6\right]^{16}\right\} \qquad (2\text{-}42)$$

The approximation used in Function DFN to implement Equation 2-42 uses the equivalent equation:

$$P\left(\frac{x}{\sqrt{2}}\right) = \left[1 - 0.5/\left(1 + b_1x + b_2x^2 + b_3x^3 + b_4x^4 + b_5x^5 + b_6x^6\right)^{16}\right] \qquad (2\text{-}43)$$

where

$b_1 = a_1/2^{1/2} = 0.0498673469$

$b_2 = a_2/2^{2/2} = 0.0211410061$

$b_3 = a_3/2^{3/2} = 0.0032776263$

$b_4 = a_4/2^{4/2} = 0.0000380036$

$b_5 = a_5/2^{5/2} = 0.0000488906$

$b_6 = a_6/2^{6/2} = 0.000005383$

This approximation for function, P, is used for both terms in Equation 2-37 and also for the probability of hit along the Z-axis, so Equation 2-32 is evaluated in Function PHIT using the equation:

$$P_H = \left[P\left(\frac{y_1 + g_y/2}{\sigma_y}\right) - P\left(\frac{y_1 - g_y/2}{\sigma_y}\right) \right]$$

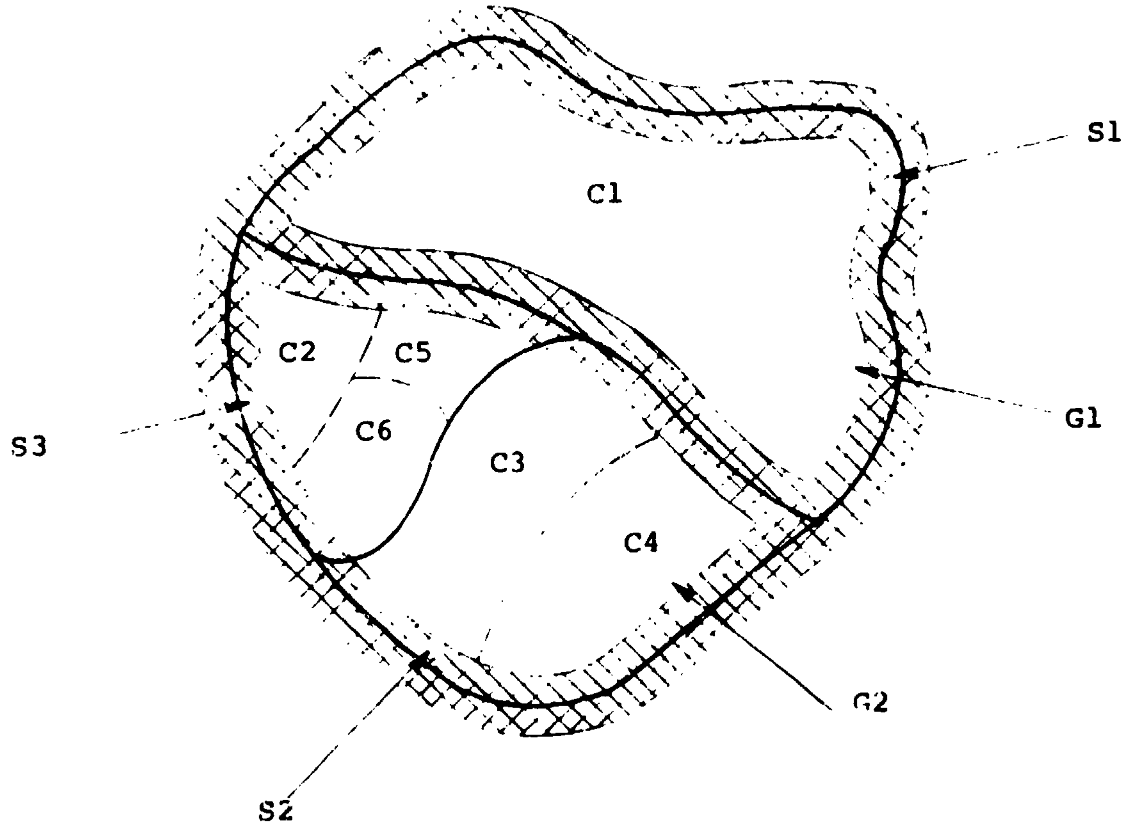
$$* \left[P\left(\frac{z_1 + g_z/2}{\sigma_z}\right) - P\left(\frac{z_1 - g_z/2}{\sigma_z}\right) \right] \qquad (2\text{-}44)$$

Pk Computations

Component Pk's are determined by the total amount of energy accumulated on the component. On Card 15 of the input data deck, the user must specify component Pk at ten levels of accumulated energy for every component. Once the accumulated energy is known, component Pk is computed by linear interpolation of the component Pk values for the accumulated energy level.

The last set of cards in the input data deck (Card 17) is used to define as many as three aircraft fault tree structures. These structures are used to determine the method for combining component Pk's into total aircraft Pk's for each kill category, possibly utilizing several levels of subgroups. Subroutine FALTRE is used to interpret the fault tree structures stored in array MUL and compute Pk's for each group by properly combining Pk's for the subgroups. The mathematical technique for computing a total aircraft Pk using a fault tree description is discussed in the text that follows using an example fault tree with two intermediate levels. In this discussion the word, subgroup, refers to a set of components; and the word, group, refers to a set of subgroups.

Referring to Figure 2-9, a target is illustrated in space as a symbolic shape. The most elemental building block of the target is a component, of which six appear in the example. Combinations of components form subgroups, three of which are presented in the example. Subgroups are combined into groups, there being two groups in the displayed target. A fault tree diagram of the same target is shown in Figure 2-10. The analogy between Figure 2-9 and the appearance of a political map is directly applicable. For example, if Sven Forkbeard asks his field marshal, "What happens to Targetsland if we take Essthree County?", the field marshal must answer, "I am told by my spies that we can control the State, Geetwo, depending upon the momentary defenses of Esstwo County." The vulnerability of a group must therefore be described by how many of its subgroups must be killed in order to kill the group containing them.



Components: C1, C2, C3, C4, C5, C6 — Component Boundary

Subgroups: S1, S2, S3 — Subgroup Boundary

Groups: G1, G2 — Group Boundary

Figure 2-9. Construction of a Target with Redundant Subgroups.

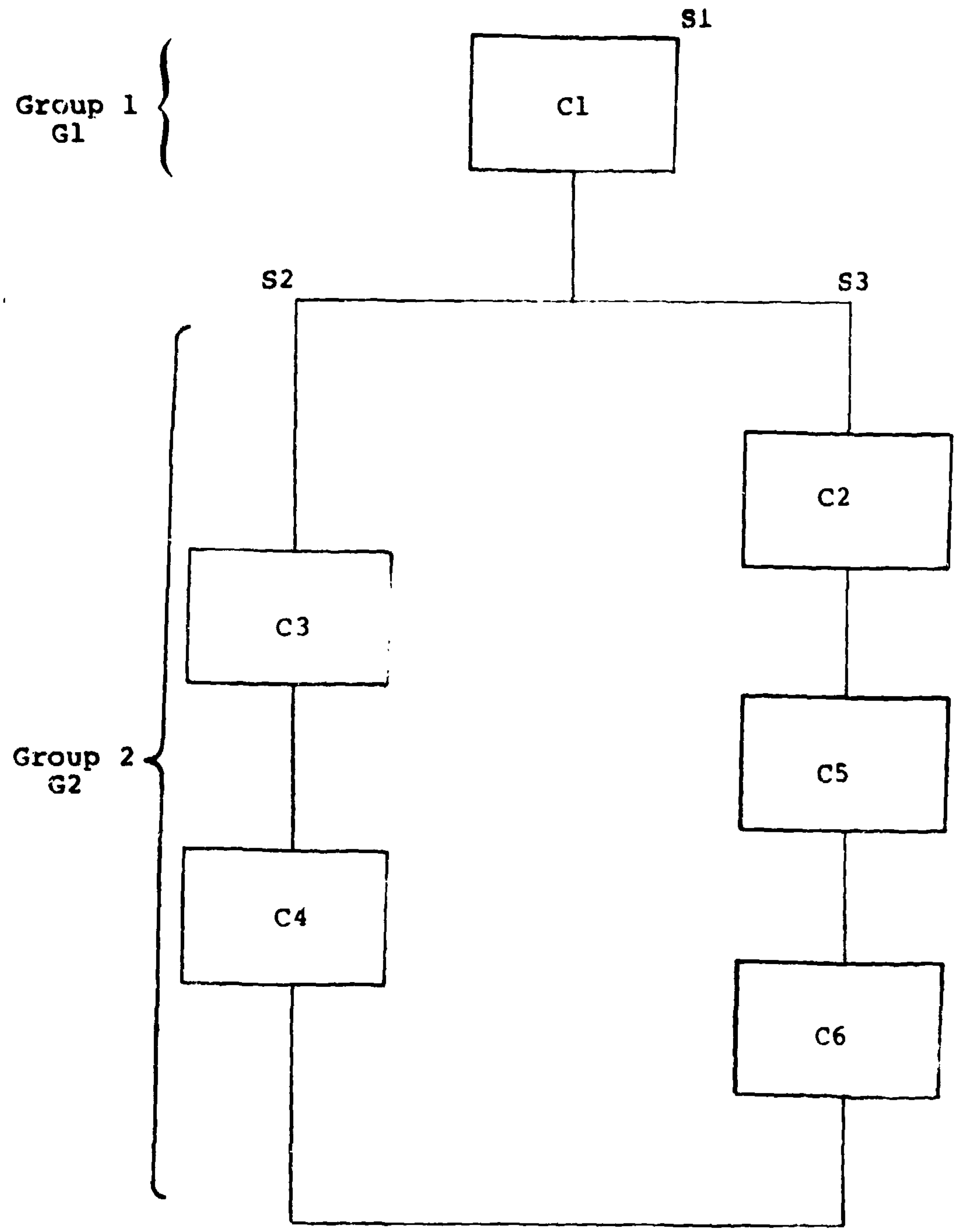


Figure 2-10. Fault Tree for the Target in Figure 2-9.

In order to completely describe the vulnerability of a target, the following definitions are made:

$Ncomps_j$ = number of components in the jth subgroup

$Nsub_k$ = number of subgroups constituting the kth group

Ngroup = number of groups into which the entire target is divided

$Nreq_k$ = number of subgroups required to be killed in the kth group in order to score a kill for the entire group

In this example, each subgroup is a singly vulnerable collection of components and computation of a subgroups's kill probability makes use of Equation 2-45:

$$Pks_j = 1 - \prod_{i=1}^{Ncomps_j} (1 - Pkc_i) \tag{2-45}$$

where

Pks_j = kill probability for subgroup j
Pkc_i = kill probability for component i

Similarly the total target in this example is a singly vulnerable collection of groups, and the total target kill probability is computed with a similar equation

$$Pk = 1 - \prod_{n=1}^{Ngroup} (1 - Pkg_n) \tag{2-46}$$

where

Pk = probability of kill for the total target
Pkg_n = kill probability for group n

Mathematical statements like Equations 2-45 and 2-46 are evaluated in Subroutine FALTRE to determine the Pk for any group of singly vulnerable subgroups.

In this example, group G2 contains redundant or parallel subgroups, and its probability of kill depends on the number of subgroup kills required to cause failure of the entire group. The following paragraphs are used to describe the method used to compute Pk's for parallel or redundant subgroups.

Each subgroup is assumed to exist in only a kill or survive state. The probabilities for each state are:

Pks_n = probability that the nth subgroup is killed

$(1 - Pks_n)$ = probability that the nth subgroup survives

In a probability sample space, the probability content of the total area of the space is unity, i.e. the probability of all possible events occurring is one. Figure 2-11 is a graphical representation of the probability space for a group of two subgroups. The areas constituting Figure 2-11, numbered 1 through 4, represent all combinations of events in the sample space defined by two subgroups with two possible states. Let us now apply a conditional constraint upon the events, specifying interest in accounting for only those events where at least one subgroup is killed. For the example, the probability becomes the sum of the areas 2, 3, and 4

$$P(\text{one kill}) = P(2) + P(3) + P(4) \qquad (2\text{-}47)$$

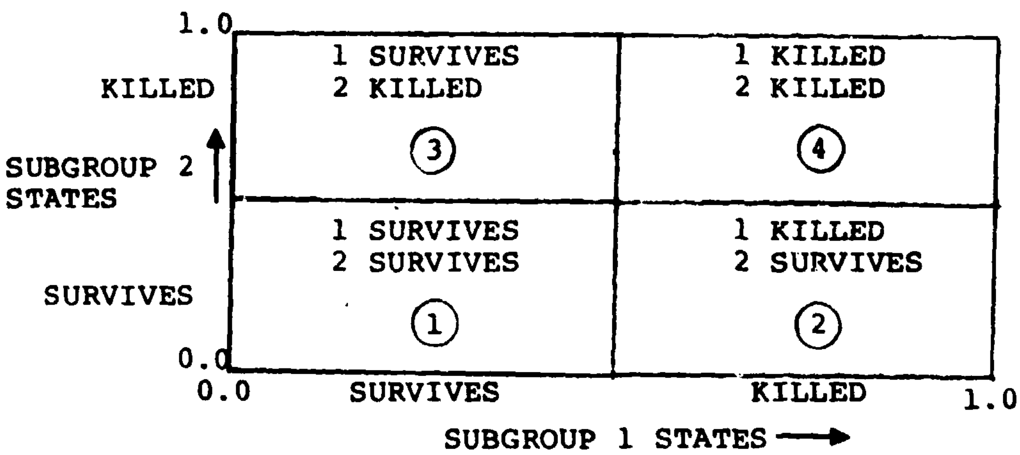


Figure 2-11. Probability Space of Two Subgroups.

or, substituting the appropriate survival and kill probabilities for the summed terms of Equation 2-47,

$$P(\text{one kill}) = (Pks_1)(1-Pks_2) + (1-Pks_1)(Pks_2) + (Pks_1)(Pks_2) \qquad (2\text{-}48$$

Let:

$P(Nreq_k)$ = probability of group kill, given that at least $Nreq_k$ subgroups must be killed

As in the preceding example, the probability of a group kill, $P(Nreq_k)$, is computed as the sum of the exclusive event probabilities, for all events satisfying the specified outcome, i.e. at least $Nreq_k$ subgroup kills. This sum can be implemented by introducing a binary number with $Nsub_K$ binary digits and a function, $B_n(j)$, to select one of these bits.

Define

$B_n(j)$ = binary bit of order n in the binary number representing (j-1), for example $B_1(6)$ is the first order bit of the binary number representing 5 i.e. the right-hand digit of 010<u>1</u> (bits are numbered from right to left).

Now define

$F_n(j)$ = Pks_n, the kill probability of the nth subgroup if $B_n(j)=1$

= $(1-Pks_n)$, the survival probability of the nth subgroup if $B_n(j)=0$

For an exclusive event, A_j, in the sample space of all combinations of subgroup kills and survivals, the probability of the event can be computed using the expression:

$$P(A_j) = \prod_{n=1}^{Nsub_k} F_n(j) \qquad (2\text{-}49)$$

where

A_j = one exclusive event consisting of a unique combination of kills and survivals for subgroups $1, 2, \ldots Nsub_k$

$P(A_j)$ = probability of event A_j

Since only two states, kill or survive, are allowed for each subgroup, the number of possible combinations of subgroup states in the sample space can be computed using

$$M = 2^{Nsub_k} \tag{2-50}$$

where

M = total number of possible combinations of subgroup states in the sample space.

The sum of all events in the sample space is one.

$$1 = \sum_{j=1}^{M} P(A_j) \tag{2-51}$$

$$1 = \sum_{j=1}^{2^{Nsub_k}} \prod_{n=1}^{Nsub_k} F_n(j) \tag{2-52}$$

By introducing another term to include only the desired events from the sample space, the group kill probability can be computed:

$$P(Nreq_k) = \sum_{j=1}^{2^{Nsub_k}} I(j) \prod_{n=1}^{Nsub_k} F_n(j) \tag{2-53}$$

where

$I(j)$ = 1 if at least $Nreq_k$ terms of the product, $\Pi F_n(j)$, are kill probabilities; i.e. if $\sum_{n=1}^{Nsub_k} B_n(j) \geq Nreq_k$

= 0 if at least 1 + $Nsub_k$ - $Nreq_k$ terms of the product $\Pi F_n(j)$ are survival probabilities; i.e. if $\sum_{n=1}^{Nsub_k} B_n(j) < Nreq_k$

Equation 2-53 is the formulation mechanized in Subroutine MVHART.

LIST OF ABBREVIATIONS AND SYMBOLS

This subsection contains a complete list of symbols used in the Mathematical Model. The list is arranged alphabetically with capital letters preceding lower case letters and Greek letters at the end. The list is divided into four columns with the symbols printed in the left column and their definitions printed in the third column. If a mathematical symbol has an equivalent FORTRAN variable name in the program source code, the FORTRAN name is printed in the second column. The fourth column is used to indicate the units of the value for the symbol when any apply.

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
A_j	---	One exclusive event consisting of a unique combination of kills and survivals for subgroups 1,2,...$Nsub_k$; $1 \leq j \leq 2\ Nsub_k$	ND*
A_{1a}	AIMAZ	Azimuth look-angle of the line from the laser to the aim point; $0.0 \leq A_{1a} \leq 2\pi$	radians
A_{1c}	COMPZA	Azimuth look-angle of the line from the laser to the component; $0.0 \leq A_{1c} \leq 2\pi$	radians
A_z	---	Azimuth angle of the line from the weapon to the target in the General Coordinate System	radians
A_z'	AZDOT	Azimuth slew rate of the laser weapon	radians/second
a_1	---	Constant used in the Hastings approximation; =0.0705230784	ND
a_2	---	Constant used in the Hastings approximation; =0.0422820123	ND
a_3	---	Constant used in the Hastings approximation; =0.0092705272	ND
a_4	---	Constant used in the Hastings approximation; =0.0001520143	ND

*Nondimensional

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
a_5	---	Constant used in the Hastings approximation; =0.0002765672	ND
a_6	---	Constant used in the Hastings approximation; =0.0000430638	ND
$B_n(j)$	---	The binary bit of order n in the binary number representing (j-1), for example $B_1(6)$ is the first order bit of the binary number representing 5 i.e. the right hand bit of 010$\underline{1}$ (Bits are numbered from right to left)	ND
b_1	---	Constant used in the modified Hastings approximation; =0.0498673469	ND
b_2	---	Constant used in the modified Hastings approximation =0.0211410061	ND
b_3	---	Constant used in the modified Hastings approximation; =0.0032776263	ND
b_4	---	Constant used in the modified Hastings approximation; =0.0000380036	ND
b_5	---	Constant used in the modified Hastings approximation; =0.0000488906	ND

LIST OF ABBREVIATIONS AND SYMBOLS
(MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
b_6	---	Constant used in the modified Hastings approximation; =0.000005383	ND
c_a	COMP(I,ICOMP), I=1,2,3	Vector locating the component in the Aircraft Coordinate System	meters
c_x	D(1)	x-component of the vector from the laser location to the component in the Aircraft Coordinate System	meters
c_y	D(2)	y-component of the vector from the laser location to the component in the Aircraft Coordinate System	meters
c_z	D(3)	z-component of the vector from the laser location to the component in the Aircraft Coordinate System	meters
D	---	Dive transformation matrix	ND
E_1	---	Elevation angle of the line from the weapon to the target	radians
E_1'	ELDOT	Elevation slew rate for the laser weapon	radians/second
E_{1a}	AIMEL	Elevation look-angle of the line from the laser to the aim point	radians

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
E_{lc}	COMPEL	Elevation look-angle of the line from the laser to the component	radians
$F_n(j)$	PKM(N)	Pks_n, the kill probability of the nth subgroup if $B_n(j)=1$; otherwise $(1-Pks_n)$, the survival probability of the nth subgroup if $B_n(J)=0$	ND
G_{ca}	---	Vector from the laser location to the component in the Aircraft Coordinate System with vector components (c_x c_y, c_z)	meters
G_{ta}	---	Vector in the Aircraft Coordinate System equivalent to G_{tg}	meters
G_{tg}	GUNTAR(I), I=1,2,3	Vector from the laser location to the target center in the General Coordinate System	meters
g_y	---	Width of the component presented area	radians
g_z	---	Height of the component presented area	radians
H	---	Heading transformation matrix	ND

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
$I(j)$	---	=1 if at least $Nreq_k$ terms of the product, $\Pi F_n(j)$, are kill probabilities, i.e. if $\sum_{n=1}^{Nsub_k} B_n(j) \geq Nreq_k$ =0 if at least $1+Nsub_k-Nreq_k$ terms of the product, $\Pi F_n(j)$, are survival probabilities, i.e. if $\sum_{n=1}^{Nsub_k} B_n(j) < Nreq_k$	ND
M	---	Number of possible combinations of subgroup states in the sample space; $M=2^{Nsub_k}$	ND
$Ncomps_j$	LSYS	Number of components in the jth subgroup	ND
Ngroup	LSYS	Number of groups into which the entire target is divided	ND
$Nreq_k$	LREQ	Number of subgroups required to be killed in the kth group in order to score a kill for the entire group	ND
$Nsub_k$	LSYS	Number of subroups constituting the kth group	ND
$P(A_j)$	---	Probability of event A_j	ND

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
P_H	PHIT	Probability of hitting a rectangular component offset from the aim point	ND
P_{Hy}	PHITY	Probability of hit within the y directional limits of the component presented area	ND
Pk	---	Probability of kill for the total target	ND
Pkc_i	---	Kill probability for component i	ND
Pkg_n	---	Kill probability for group n	ND
Pks_j	---	Kill probability for subgroup j	ND
$P(Nreq_k)$	---	Probability of group kill, given that at least $Nreq_k$ subgroups must be killed	ND
P(one kill)	---	Probability of at least one kill in a group consisting of two subgroups	ND
$P(y_1+\frac{g_y}{2})$	---	Integral of the normal probability density function evaluated from $-\infty$ to $(y_1+g_y/2)$	ND
$P(y_1-\frac{g_y}{2})$	---	Integral of the normal probability density function evaluated from $-\infty$ to $(y_1-g_y/2)$	ND
R	---	Roll transformation matrix	ND

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
s_s	S	Fraction of the distance between the smoke corridor first and second end points at which the point of intersection with the laser to aircraft vector lies	ND
s_w	---	Fraction of the horizontal distance between the laser and aircraft at which the point of intersection with the smoke corridor lies	ND
T	TRANS	Transformation matrix between two coordinate systems; product of heading, dive, and roll transformation matrices	ND
v_{xf}	---	x-component of a vector in the Flight Path Coordinate System	ND
v_{xg}	---	x-component of a vector in the General Coordinate System equivalent to (v_{xf}, v_{yf}, v_{zf})	ND
v_{yf}	---	y-component of a vector in the Flight Path Coordinate System	ND
v_{yg}	---	y-component of a vector in the General Coordinate System equivalent to (v_{xf}, v_{yf}, v_{zf})	ND
v_{zf}	---	z-component of a vector in the Flight Path Coordinate System	ND

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
v_{zg}	---	z-component of a vector in the General Coordinate System equivalent to (v_{xf}, v_{yf}, v_{zf})	ND
x_a	TARGET(1)	x-coordinate of the aircraft location in the General Coordinate System	meters
x_f	XIN	x-coordinate of the aircraft location in the Flight Path Coordinate System	meters
x_{fr}	XFP	x-coordinate of the reference point in the Flight Path Coordinate System	meters
x_g	OTAPE(2,I)	x-coordinate of the location in the General Coordinate System equivalent to (x_f, y_f, z_f)	meters
x_{gr}	XG	x-coordinate of the reference point in the General Coordinate System	meters
x_i	XY(1)	x-coordinate of the point of intersection of the smoke corridor and weapon-to-aircraft line	meters
x_{s1}	SMOKX(1)	x-coordinate of the smoke corridor first end point	meters
x_{s2}	SMOKX(2)	x-coordinate of the smoke corridor second end point	meters

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
x_w	GUN(1)	x-coordinate of the weapon location in the General Coordinate System	meters
(x,y,z)	GUNTAR	Vector from the laser location to the target in the General Coordinate System	meters
(x',y',z')	(TXDOT,TYDOT, TZDOT)	Rate of change in the aircraft position vector; the aircraft velocity vector	meters/ second
$(x,y,z)_1$	---	Vector in the coordinate system with axes X_1, Y_1, and Z_1	ND
$(x,y,z)_2$	---	Vector in the coordinate system with axes X_2, Y_2, and Z_2; equivalent to $(x,y,z)_1$	ND
$(x,y,z)_3$	---	Vector in the coordinate system with axes X_3, Y_3, and Z_3; equivalent to $(x,y,z)_2$	ND
$(x,y,z)_4$	---	Vector in the coordinate system with axes X_4, Y_4, and Z_4; equivalent to $(x,y,z)_3$	ND
y_a	TARGET(2)	y-coordinate of the aircraft location in the General Coordinate System	meters

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
y_f	YIN	y-coordinate of the aircraft location in the Flight Path Coordinate System	meters
y_{fr}	YFP	y-coordinate of the reference point in the Flight Path Coordinate System	meters
y_g	OTAPE(3,I)	y-coordinate of the location in the General Coordinate System equivalent to (x_f, y_f, z_f)	meters
y_{gr}	YG	y-coordinate of the reference point in the General Coordinate System	meters
y_i	XY(2)	y-coordinate of the point of intersection of the smoke corridor and weapon-to-aircraft line	meters
y_{s1}	SMOKY(1)	y-coordinate of the smoke corridor first end point	meters
y_{s2}	SMOKY(2)	y-coordinate of the smoke corridor second end point	meters
y_w	GUN(2)	y-coordinate of the weapon location in the General Coordinate System	meters
y_1	COMPE(2)	y-coordinate of the component centroid in the Encounter Coordinate System, measured in radians from the laser location	radians

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
z_f	OTAPE(4,I)	z-coordinate of the aircraft location in the Flight Path Coordinate System	meters
z_g	OTAPE(4,I)	z-coordinate of the location in the General Coordinate System equivalent to (x_f, y_f, z_f)	meters
z_{gr}	ZG	z-coordinate of the reference point in the General Coordinate System	meters
z_1	COMPE(3)	z-coordinate of the component centroid in the Encounter Coordinate System measured in radians from the laser location	radians
θ	---	Rotation angle for the XZ-plane after rotation through Ψ, when converting from the Aircraft to the Encounter Coordinate Systems	radians
π	---	3.14159265	radians
σ_{ay}	SIGMA(IAIM,1)	Standard deviation of the error in locating and tracking the aim point in the direction of the Y-axis of the Encounter Coordinate System	radians

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
σ_{az}	SIGMA(IAM,2)	Standard deviation of the error in locating and tracking the aim point in the direction of the Z-axis of the Encounter Coordinate System	radians
σ_{jy}	YJITTR	Standard deviation due to jitter of the beam in the direction of the Y-axis of the Encounter Coordinate System	radians
σ_{jz}	ZJITTR	Standard deviation due to jitter of the beam in the direction of the Z-axis of the Encounter Coordinate System	radians
σ_{y}	SIGY	Total standard deviation in the direction of the Y-axis of the Encounter Coordinate System	radians
σ_{z}	SIGZ	Total standard deviation in the direction of the Z-axis of the Encounter Coordinate System	radians
$\Phi(x)$	---	Integral evaluated by the Hastings approximation	ND
ψ	---	Rotation angle for the XY-plane in a transformation between coordinate systems	radians

LIST OF ABBREVIATIONS AND SYMBOLS (MATHEMATICAL MODEL)

Abbreviation or symbol	Equivalent in Simulation Model	Definition	Units
ψ_f	OTAPE(13,I)	Aircraft heading angle in the Flight Path Coordinate System	radians
ψ_1	PSI	Rotation angle from the X-axis of the Flight Path Coordinate System to the X-axis of the General Coordinate System (a positive rotation is counterclockwise when viewed from above; i.e. the positive Z-axis)	radians

SECTION III

INPUT

There are two input files required when executing the ASALT-I Model. The first input file, called the data deck, is read from Logical Unit #5 and consists of formatted records or cards. The second input file, read from Logical Unit #10, contains the aircraft flight path data on a binary tape generated by executing the Engagement Model. This section is used to describe these input files by presenting the order of the records on both files, and listing definitions for all input parameters. This information is primarily in tabular form so that this section may be used frequently as a quick reference source while preparing the program input.

FILE5 - INPUT DATA DECK

The input data deck consists of 17 different card types arranged in the order shown in Figure 3-1. Following the figure, a set of data card description forms is used to present the details of each card type including parameter definitions, formats, units, and locations of each field on the card. The card contents and card ID number printed on the top rows of each data card description form correspond to a card contents and ID number in Figure 3-1. The columns of each data card description form are used to list the units, definition, format, and card column location for each input parameter. Cards 1 through 10 are read during execution of Subroutine READY and contain parameters which describe the laser characteristics or select various program options. Cards 11 through 17 contain parameters describing the target aircraft and are read during execution of Subroutines ACIN and MVINPT.

The components of the aircraft can be arranged in a variety of fault tree structures by the parameters on the final group of cards, which use the Card 17 format. These cards contain alphanumeric data which are read and interpreted by executing Subroutine MVINPT, and enable a user of the ASALT program to use an English-like description to define fault trees for as many as three aircraft kill categories. Subroutine EKOMUL is executed after Subroutine MVINPT to print the fault trees as part of the program output

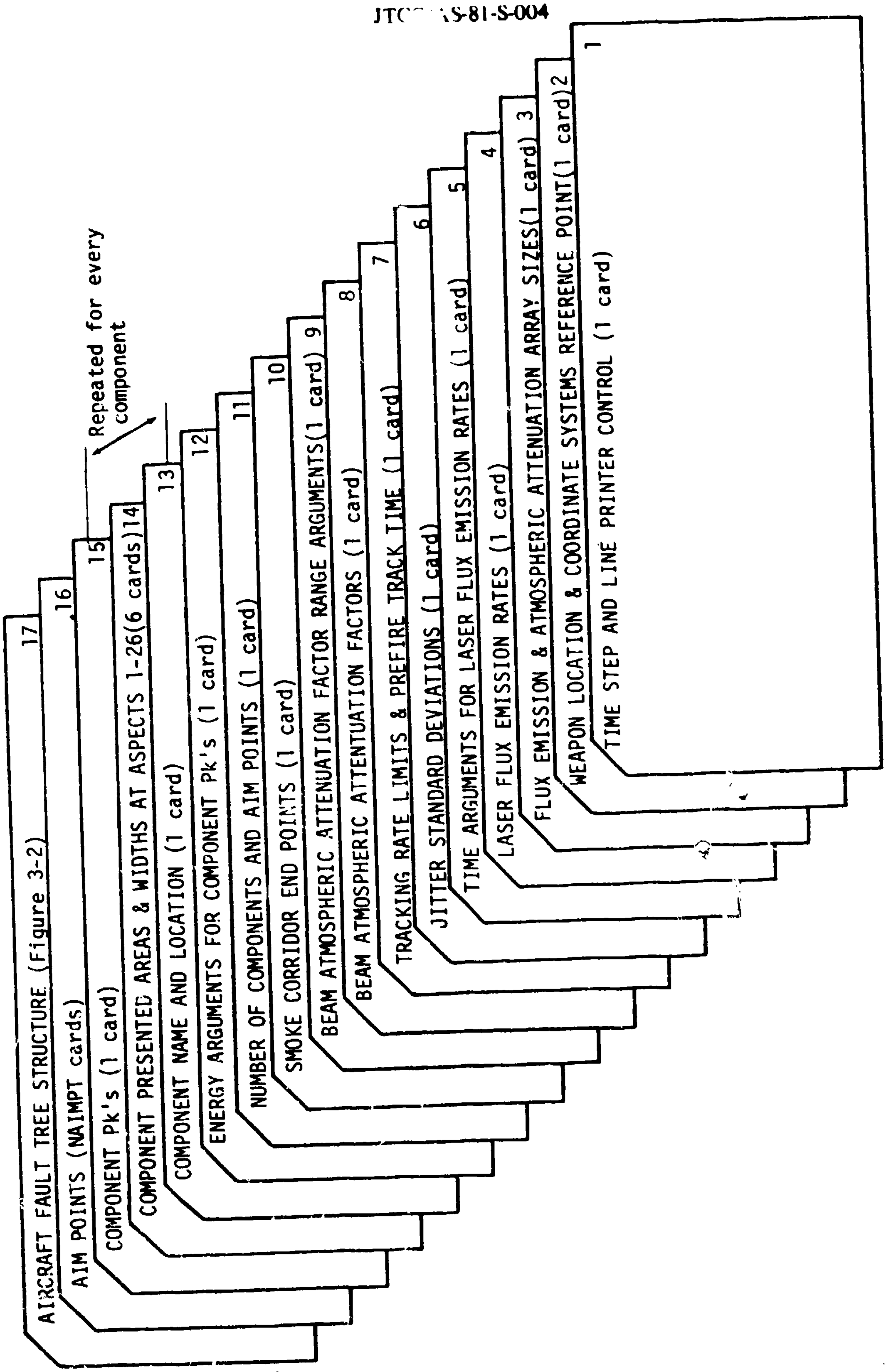


Figure 3-1. Data Deck Setup.

CARD ID NUMBER: 1

CARD CONTENTS: Time Step and Line Printer Control

WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	TDELT	seconds	Time interval between each iteration of the program computations	E8.0	1-8
2	IPRINT	---	Number of time intervals (equal to TDELT) between each line of line printer output; if IPRINT=0, only the final damage summary is printed	I8	9-16
3	LINLIM	---	Number of lines printed on each line printer page; a heading is printed at the top of each new line printer page by counting lines of output and comparing with this number	I8	17-24

CARD ID NUMBER: 2					
CARD CONTENTS: Weapon Location & Coordinate Systems Reference Point					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	GUN(1)	meters	x-coordinate of the weapon location in the General Coordinate System	E8.0	1-8
2	GUN(2)	meters	y-coordinate of the weapon location in the General Coordinate System	E8.0	9-16
3	GUN(3)	meters	z-coordinate of the weapon location in the General Coordinate System	E8.0	17-24
4	XFP	meters	x-coordinate of the reference point in the Flight Path Coordinate System	E8.0	25-32
5	YFP	meters	y-coordinate of the reference point in the Flight Path Coordinate System	E8.0	33-40
6	XG	meters	x-coordinate of the reference point in the General Coordinate System	E8.0	41-48
7	YG	meters	y-coordinate of the reference point in the General Coordinate System	E8.0	49-56
8	ZG	meters	z-coordinate of the reference point in the General Coordinate System	E8.0	57-64
NOTE: The reference point has a z-coordinate equal to 0.0 in the Flight Path Coordinate System.					

CARD ID NUMBER: 2 (concluded)					
CARD CONTENTS: Weapon Location & Coordinate Systems Reference Point					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
9	PSI	degrees	Rotation angle about the Flight Path Coordinate System Z-axis, from the Flight Path Coordinate System to the General Coordinate System. PSI is positive in the counterclockwise direction when viewed from above.	E8.0	65-72

NOTE: The reference point may be any point in the XY-plane of the Flight Path Coordinate System. It is selected by the user and is needed for transforming coordinates from the Flight Path Coordinate System to the General Coordinate System.

CARD ID NUMBER: 3

CARD CONTENTS: Flux Emission & Atmospheric Attenuation Array Sizes

WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	NFLUX	---	Number of elements in the laser flux emission array, read from Card 4; $1 \leq NFLUX \leq 10$	I8	1-8
2	NATN	---	Number of elements in the atmospheric attenuation factor array, read from Card 8; $1 \leq NATN \leq 10$	I8	9-16

CARD ID NUMBER: 4

CARD CONTENTS: Laser Flux Emission Rates

WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	FLUX(1)	watts/ cm^2	Rate of laser flux emission from time 0.0 to time FLTIME(1)	E8.0	1-8
2	FLUX(2)	watts/ cm^2	Rate of laser flux emission at time FLTIME(2)	E8.0	9-16
3	FLUX(3)	watts/ cm^2	Rate of laser flux emission at time FLTIME(3)	E8.0	17-24
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
	FLUX(NFLUX)	watts/ cm^2	Rate of laser flux emission at time FLTIME(NFLUX) and at all times greater than that	E8.0	

NOTE: Laser flux emission rate is linearly interpolated at times between two FLTIME array entries.

CARD ID NUMBER: 5					
CARD CONTENTS: Time Arguments for Laser Flux Emission Rates					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	FLTIME(1)	seconds	Time corresponding to laser flux emission rate FLUX(1) NOTE: If NFLUX=1, FLTIME(1) must be greater than all times in the aircraft flight path file	E8.0	1-8
2	FLTIME(2)	seconds	Time corresponding to laser flux emission rate FLUX(2)	E8.0	9-16
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
	FLTIME(NFLUX)	seconds	Time corresponding to laser flux emission rate FLUX(NFLUX)	E8.0	

CARD ID NUMBER: 6					
CARD CONTENTS: Jitter Standard Deviations					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	YJITTR	mils	Standard deviation due to jitter of the beam in the direction of the Y-axis of the Encounter Coordinate System	E8.0	1-8
2	ZJITTR	mils	Standard deviation due to jitter of the beam in the direction of the Z-axis of the Encounter Coordinate System	E8.0	9-16

CARD ID NUMBER: 7

CARD CONTENTS: Tracking Rate Limits & Prefire Track Time

WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	SLEWAZ	degrees/second	Maximum azimuth slewing rate for the laser weapon	E8.0	1-8
2	SLEWEL	degrees/second	Maximum elevation slewing rate for the laser weapon	E8.0	9-16
3	TRKTIM	seconds	Prefire track time, minimum tracking time necessary before the laser can fire	E8.0	17-24

CARD ID NUMBER: 8

CARD CONTENTS: Beam Atmospheric Attenuation Factors

WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	ATTEN(1)	---	Beam attenuation factor due to propagation through the atmosphere at range RATTEN(1) and at all ranges less than RATTEN(1)	E8.0	1-8
2	ATTEN(2)	---	Beam attenuation factor due to propagation through the atmosphere at range RATTEN(2)	E8.0	9-16
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
	ATTEN(NATN)	---	Beam attenuation factor due to propagation through the atmosphere at range RATTEN(NATN) and at all ranges greater than RATTEN(NATN)	E8.0	

CARD ID NUMBER: 9

CARD CONTENTS: Beam Atmospheric Attenuation Factor Range Arguments

WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	RATTEN(1)	meters	Range corresponding to attenuation factor ATTEN(1) NOTE: If NATN=1, RATTEN(1) must be greater than all possible weapon-to-aircraft ranges for the run	E8.0	1-8
2	RATTEN(2)	meters	Range corresponding to attenuation factor ATTEN(2)	E8.0	9-16
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
	RATTEN(NATN)	meters	Range corresponding to attenuation factor ATTEN(NATN)	E8.0	

CARD ID NUMBER: 10

CARD CONTENTS: Smoke Corridor End Points

WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	SMOKX(1)	meters	x-coordinate in the General Coordinate System of the smoke corridor's first end point	E8.0	1-8
2	SMOKY(1)	meters	y-coordinate in the General Coordinate System of the smoke corridor's first end point	E8.0	9-16
3	SMOKX(2)	meters	x-coordinate in the General Coordinate System of the smoke corridor's second end point	E8.0	17-24
4	SMOKY(2)	meters	y-coordinate in the General Coordinate System of the smoke corridor's second end point	E8.0	25-32
5	SMATN	---	Beam intensity attenuation due to propagation through the smoke corridor	E8.0	33-40

NOTE: If SMOKX(1)=SMOKX(2) and SMOKY(1)=SMOKY(2) then no corridor is modeled.

CARD ID NUMBER: 11					
CARD CONTENTS: Number of Components and Aim Points					
WORD	**VARIABLE**	**UNITS**	**DEFINITION**	**FORMAT**	**COLUMN**
1	NCOMP	---	Number of components in the target model; $1 \leq NCOMP \leq 100$	I8	1-8
2	NAIMPT	---	Number of aim points on the target; $1 \leq NAIMPT \leq 10$	I8	9-16
3	ITRACE	---	Fault tree trace option: $\neq 1$, omit extra output for fault trees; $= 1$, print extra data used in interpreting the fault tree structure cards.	I8	17-24

CARD ID NUMBER: 12

CARD CONTENTS: Energy Arguments for the Component Pk's

WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	ENERGY(1)	kilo-joules/ cm^2	Amount of accumulated energy necessary to cause kill probabilities, PK(1,I), for Ith component	E8.0	1-8
2	ENERGY(2)	kilo-joules/ cm^2	Amount of accumulated energy necessary to cause kill probabilities, PK(2,I), for Ith component	E8.0	9-16
3	ENERGY(3)	kilo-joules/ cm^2	Amount of accumulated energy necessary to cause kill probabilities, PK(3,I), for Ith component	E8.0	17-24
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
10	ENERGY(10)	kilo-joules/ cm^2	Amount of accumulated energy necessary to cause kill probabilities, PK(10,I), for Ith component	E8.0	73-80

CARD ID NUMBER: 13					
CARD CONTENTS: Component Name and Location					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	NAM(I)	---	Eight character alphanumeric name for the Ith component. The left-most character must be other than a blank. Do not use period (.), equal sign (=), or slash (/) in the field.	A8	1-8
2	COMP(1,I)	meters	x-coordinate of the Ith component in the Aircraft Coordinate System	E8.0	9-16
3	COMP(2,I)	meters	y-coordinate of the Ith component in the Aircraft Coordinate System	E8.0	17-24
4	COMP(3,I)	meters	z-coordinate of the Ith component in the Aircraft Coordinate System	E8.0	25-32

CARD ID NUMBER: 14					
CARD CONTENTS: Component Presented Areas & Widths at Aspects 1-26					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	AP(I,1)	$meters^2$	Presented area of the Ith component when viewed from aspect 1	E8.0	1-8
2	WIDTH(I,1)	meters	Width of the Ith component when viewed from aspect 1	E8.0	9-16
3	AP(I,2)	$meters^2$	Presented area of the Ith component when viewed from aspect 2	E8.0	17-24
4	WIDTH(I,2)	meters	Width of the Ith component when viewed from aspect 2	E8.0	25-32
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
51	AP(I,26)	$meters^2$	Presented area of the Ith component when viewed from aspect 26	E8.0	
52	WIDTH(I,26)	meters	Width of the Ith component when viewed from aspect 26	E8.0	

NOTE: 1) See Table 2-1 for a definition of the 26 aspect angles.
2) Six cards in this format are required to enter 26 presented areas and widths for each component as follows:

CARD ID NUMBER: 14 (concluded)					
CARD CONTENTS: Component Presented Areas & Widths at Aspects 1-26					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN

NOTE (Concluded): the 1st card contains data for aspects 1-5, the 2nd card contains data for aspects 6-10, the 3rd card contains data for aspects 11-15, the 4th card contains data for aspects 16-20, the 5th card contains data for aspects 21-25, the 6th card contains data for aspect 26.

CARD ID NUMBER: 15					
CARD CONTENTS: Component PK's					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	PK(1,I)	---	Pk for the Ith component resulting from accumulated energy, ENERGY(1), and all lesser amounts of energy accumulation NOTE: In most cases PK(1,I) and ENERGY(1) should have values equal to 0.0	E8.0	1-8
2	PK(2,I)	---	Pk for the Ith component resulting from accumulated energy, ENERGY(2)	E8.0	9-16
3	PK(3,I)	---	Pk for the Ith component resulting from accumulated energy, ENERGY(3)	E8.0	17-24
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
10	PK(10,I)	---	Pk for the Ith component resulting from accumulated energy, ENERGY(10), and all greater amounts of energy accumulation	E8.0	73-80

CARD ID NUMBER: 16					
CARD CONTENTS: Aim Points					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	AIM(1,I)	meters	x-coordinate of the Ith aim point in the Aircraft Coordinate System	E8.0	1-8
2	AI[illegible],I)	meters	y-coordinate of the Ith aim point in the Aircraft Coordiante System	E8.0	9-16
3	AIM(3,I)	meters	z-coordinate of the Ith aim point in the Aircraft Coordinate System	E8.0	17-24
4	SIGMA(I,1)	mils	Standard deviation of the error in locating and tracking the Ith aim point in the direction of the Y-axis of the Encounter Coordinate System	E8.0	25-32
5	SIGMA(I,2)	mils	Standard deviation of the error in locating and tracking the Ith aim point in the direction of the Z-axis of the Encounter Coordinate System	E8.0	33-40
6	AZLIM(I,1)	degrees	First azimuth look-angle boundary of the envelope for hitting the Ith aim point	E8.0	41-48

CARD ID NUMBER: 16 (Concluded)					
CARD CONTENTS: Aim Points					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
7	AZLIM(I,2)	degrees	Second azimuth look-angle boundary of the envelope for hitting the Ith aim point	E8.0	49-56
8	ELLIM(I,1)	degrees	First elevation look-angle boundary of the envelope for hitting the Ith aim point	E8.0	57-64
9	ELLIM(I,2)	degrees	Second elevation look-angle boundary of the envelope for hitting the Ith aim point	E8.0	65-72

NOTE: This card is repeated for every aim point

CARD ID NUMBER: 17					
CARD CONTENTS: Aircraft Fault Tree Structure					
WORD	VARIABLE	UNITS	DEFINITION	FORMAT	COLUMN
1	ICARD(1)	---	Eighty alphanumeric characters used to define a fault tree structure for a group or subgroup in the aircraft. See Figure 3-2 and Table 3-1 for a description of the English-like text used on these cards.	A1	1
2	ICARD(2)	---		A1	2
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
80	ICARD(80)	---		A1	80

The order of the cards in the fault tree description section of the input deck is depicted in Figure 3-2. The fault tree description for each kill category requires one Kill Category Card, followed by one Group Definition Card, followed by any necessary Subgroup Definition Cards, and finally the End Cards. If a user wants to define a second or third fault tree for a different kill category, the same sequence of cards is repeated. The total number of kill categories must not exceed three. Finally, one Blank Card is necessary to indicate the end of all fault tree descriptions. Figure 3-3 is an example listing of a fault tree input description for two kill categories. The fault trees produced from this input are shown in Section IV.

The rules and examples in Table 3-1 are a summary of the most important rules for assembling the fault tree descriptions. Most of the examples are taken directly from the sample input listed in Figure 3-3. The left-most characters in these examples are always in column 1 of the input records. Subroutine MVINPT is not currently elaborate enough to detect every possible input error. The best method for a user to validate the Aircraft Fault Tree Structure input cards should include both a search for error messages printed by Subroutine MVINPT, and comparing the fault trees printed by executing Subroutine EKOMUL with the fault trees the user intended to create.

One important difference between this method of defining fault trees and the method used in the COVART program is that the ASALT-I program requires every component in a fault tree to be listed in the fault tree description. If a component is omitted from a fault tree description, then it is not included in the fault tree.

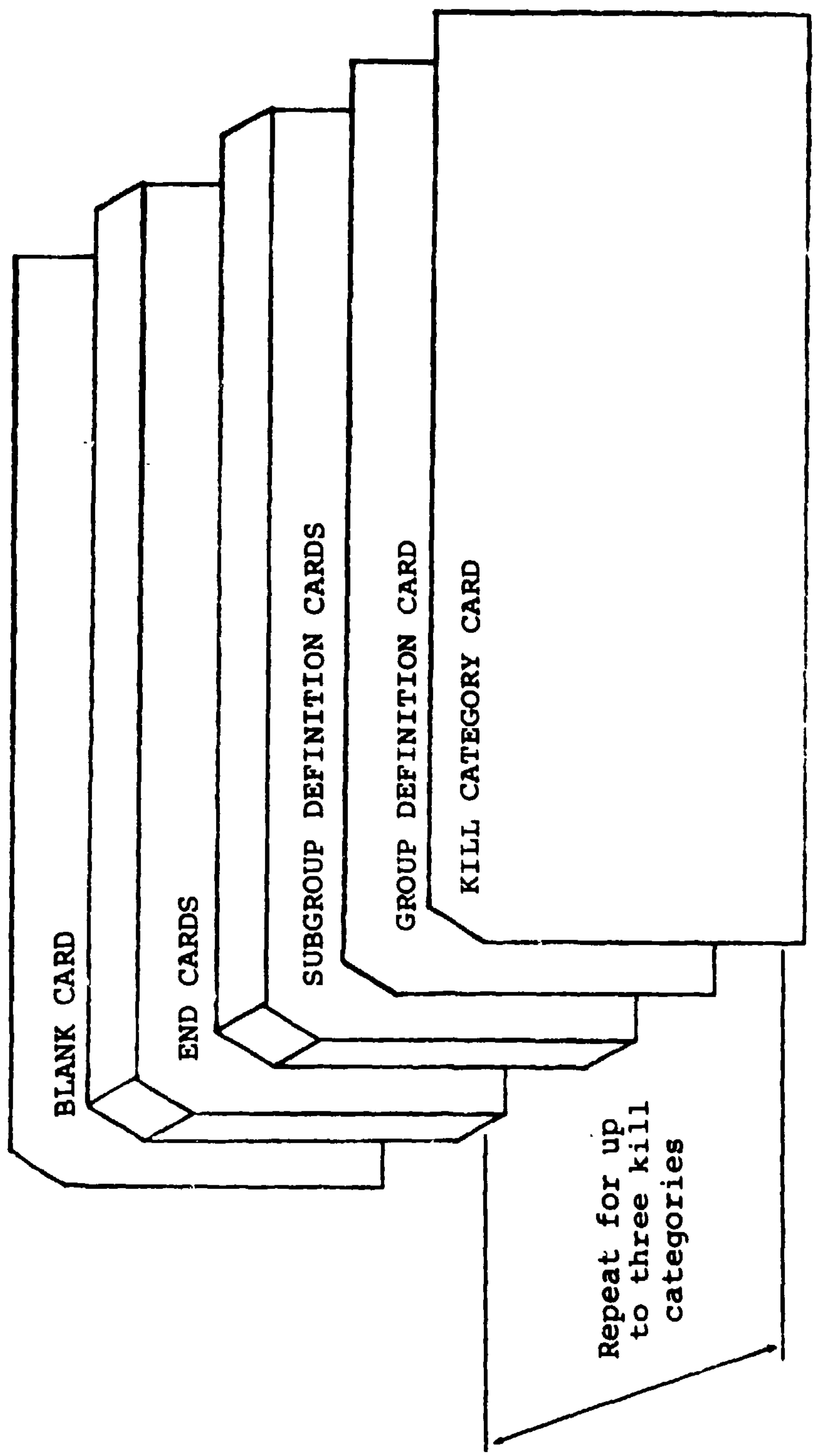


Figure 3-2. Ordering for Aircraft Fault Tree Structure Cards.

```
KILL CATEGORY 1          (ATTRITION)
*GATTRN=FLT CNTL.OR.AFT LNGN
*SFLT CNTL=ELEC CTR.AND.MECH CTR/2/2
*SELEC CTR=ELEC LNK.OR.BAT GENR
*SELEC LNK=ELEC 1.AND.ELEC 2.AND.ELEC 3.AND.ELEC 4/3/4
*SELEC 1=FCES 1.OR.STABS 1
*SELEC 2=FCES 2.OR.STABS 2
*SELEC 3=FCES 3.OR.STABS 3
*SELEC 4=FCES 4.OR.STABS 4
*SBAT GENR=BATTERYS.AND.GENRATRS/2/2
*SBATTERYS=L BATTRY.AND.R BATTRY/2/2
*SGENRATRS=L GENR.AND.R GENR/2/2
*SMECH CTR=LON LK F.OR.LON CABL.OR.LON LK A.OR.LAT LK F.OR.LAT CABL.OR.C4
SC4=LAT LK A
*SAFT LNGN=AFT UL.AND.AFT UR.AND.AFT SL.AND.AFT SR.AND.AFT RL.AND.AFT RR/3/6/
END
END OF ATTRITION GROUP
KILL CATEGORY 2          (MISSION ABORT)
*GM ABORT  = FLT CNTL.OR.AFT LNGN
*SFLT CNTL = FLE CNTR.AND.MECH CTR /1/2
*SELE CNTR = FC CMPS.OR.ELEC LNK
*SFC CMPS  = L FCCOMP.AND.R FCCOMP /1/2
*SELEC LNK = ELEC 1.OR.ELEC 3
*SELEC 1   = FCES 1.AND.FCES 2.AND.FCES 3.AND.FCES 4 /3/4
*SELEC 3   = STABS 1.AND.STABS 2.AND.STABS 3.AND.STABS 4 /3/4
*SMECH CTR = LON LK'S .OR. CABLES .OR. LAT LK'S
*SLON LK'S = LON LK A.AND.LON LK F /2/2
*SCABLES   = LON CABL.AND.LAT CABL.AND.TENS REG /1/3
*SLAT LK'S = LAT LK A.AND.LAT LK F /2/2
*SAFT LNGN = AFT UL.AND.AFT UR.AND.AFT SL.AND.AFT SR.AND.AFT RL.AND.AFT RR/4/6/
END
END OF MISSION ABORT GROUP
```

Figure 3-3. Example Fault Tree Input.

Table 3-1. Rules for Assembling the Fault Tree Input

1. The first card for each kill category fault tree must define a kill category between 1 and 3 in column 15 of the card.

   Correct Example

   ```
   KILL CATEGORY 1          (ATTRITION)
   ```

   Incorrect Example

   ```
   KILL CATEGORY1           (ATTRITION)
   ```

   Incorrect Example

   ```
   KILL CATEGORY 4          (ATTRITION)
   ```

2. The second card for each kill category must be a Group Definition Card. This card contains the letter "G" in column 1 or the characters "*G" in columns 1 and 2. No other card in the kill category fault tree description may be a Group Definition Card.

   Correct Example

   ```
   KILL CATEGORY 1          (ATTRITION)
   *GATTRN=FLT CNTL.OR.AFT ENGN
   *SFLT CNTL=ELEC CTR.AND.MECH CTR/2/2
   ```

   Incorrect Example

   ```
   KILL CATEGORY 1          (ATTRITION)
   *SFLT CNTL=ELEC CTR.AND.MECH CTR/2/2
   *GATTRN=FLT CNTL.OR.AFT ENGN
   ```

3. The fault tree description for each kill category must conclude with a pair of End Cards. The characters "END" in columns 1-3 cause any card to be interpreted as one of the End Cards. The set of End Cards may include an optional card between the two End Cards. This optional card has no effect in the ASALT-I program, but is included to keep the input descriptions compatible with those used for the COVART program.

Table 3-1. Rules for Assembling the Fault Tree Input (Continued)

Correct Example

```
*SGENRATRS=L GENR.AND.R GENR/2/2
END
END OF ATTRITION GROUP
KILL CATEGORY 2            (MISSION ABORT)
```

Correct Example

```
*SGENRATRS=L GENR.AND.R GENR/2/2
END
OPTIONAL
END OF ATTRITION GROUP
KILL CATEGORY 2            (MISSION ABORT)
```

Incorrect Example

```
*SGENRATRS=L GENR.AND.R GENR/2/2
END OF ATTRITION GROUP
KILL CATEGORY 2            (MISSION ABORT)
```

4. A blank space in column 1 of any card in the fault tree descriptions causes the entire card to be interpreted as the Blank Card which indicates the end of all fault tree input. The set of End Cards for the last kill category fault tree description must be followed by the Blank Card.

Corect Example

```
*SLAT LK'S = LAT LK A.AND.LAT LK F /2/2
END
END OF MISSION ABORT GROUP
  ANY CARD WITH A BLANK IN COLUMN 1 CONCLUDES ALL FAULT TREE INPUT
```

Incorrect Example

```
*SLAT LK'S = LAT LK A.AND.LAT LK F /2/2
 END   (ALL OTHER CARDS MUST START IN COLUMN 1)
END OF MISSION ABORT GROUP
  ANY CARD WITH A BLANK IN COLUMN 1 CONCLUDES ALL FAULT TREE INPUT
```

Table 3-1. Rules for Assembling the Fault Tree Input (Continued)

---

5. Column 1 of each Group or Subgroup Definition Card must contain an asterisk (*), letter G, letter S, or letter C. If an asterisk is used in column 1, then either letter G, letter S, or letter C must appear in column 2. There is no difference in the ASALT-_ program when using the letters C or S. Both letters are allowed so that the input is compatible with COVART input.

   Correct Example

```
*GATTRN=FLT CNTL.OR.AFT LNGN
*SFLT CNTL=ELEC CTR.AND.MECH CTR/2/2
SELEC CTR=ELEC LNK.OR.BAT GENR
*SELEC LNK=ELEC 1.AND.ELEC 2.AND.ELEC 3.AND.ELEC 4/4/4
CELEC 1=FCES 1.OR.STARS 1
*CELEC 2=FCES 2.OR.STARS 2
```

   Incorrect Example

```
**GATTRN=FLT CNTL.OR.AFT LNGN
*FLT CNTL=ELEC CTR.AND.MECH CTR/2/2
ELEC CTR=ELEC LNK.OR.BAT GENR
-ELEC LNK=ELEC 1.AND.ELEC 2.AND.ELEC 3.AND.ELEC 4/4/4
 CELEC 1=FCES 1.OR.STARS 1
+CELEC 2=FCES 2.OR.STARS 2
```

6. Each Group or Subgroup Definition Card must contain in order:

   a. the charcters *G, *S, *C, G, S, or C starting in column 1,

   b. a defined name which may be:

      i) a group name used only on a Group Definition Card

      ii) or a name used in the structure definition of a preceding card

   c. an equal sign (=)

   d. the structure definition for the defined name which may be:

      i) a name field

Table 3-1. Rules for Assembling the Fault Tree Input (Continued)

---

ii) a set of name fields separated by connectors (.AND. or .OR.)

Correct Example

```
*GATTRN=FLT CNTL.OR.AFT ENGN
*SFLT CNTL=ELEC CTR.AND.MECH CTR/2/2
*SELEC CTR=ELEC LNK
```

Incorrect Example

```
*GATTRN FLT CNTL.OR.AFT ENGN
*SELEC CTR=ELEC LNK
*SFLT CNTL=ELEC CTR     MECH CTR/2/2
```

7. The kill probabilities for a subgroup are printed in the damage summary at the end of an ASALT run only if:

   a. the subgroup name is the defined name on a Subgroup Definition Card with an asterisk (*) in column 1,

   b. and the subgroup name begins on column 3 of that Subgroup Definition Card.

   Example

```
*GATTRN=FLT CNTL.OR.AFT ENGN
*SFLT CNTL=ELEC CTR.AND.MECH CTR/2/2
*SELEC CTR=ELEC LNK.OR.BAT GENR
SELEC LNK=ELEC 1.AND.ELEC 2.AND.ELEC 3.AND.ELEC 4/4/4
```

   In this example, kill probabilities would be printed in the damage summary for Subgroup FLT CNTL but not for Subgroups ELEC CTR or ELEC LNK.

8. No name field including any embedded blanks may exceed eight characters in length. Note that embedded blanks are part of the name field.

   Correct Example

```
*SBAT GENR=BATTERYS.AND.GENRATRS/2/2
*SBATTERYS=L BATTRY.AND.R BATTRY/2/2
```

Table 3-7. Rules for Assembling the Fault Tree Input (Continued)

Incorrect Example

```
*SBAT GENR=BATTERYS.AND.GENERATORS/2/2
*SBATTERYS=LEFT BATTRY.AND.RIGHT BATTRY/2/2
```

9. An equal sign (=), period (.), or slash (/) are not allowed in any name field. Users of the COVART Program must also exclude the symbols plus (+) and minus (-) from name fields.

Correct Example

```
*SELEC 3=FCES 3.OR.STABS 3
```

Incorrect Example

```
*SELEC=3=FCES.3.OR.STABS/3
```

10. The structure definition for a singly vulnerable group must use the connector .OR. indicating that any one subgroup failure is sufficient to cause failure of the whole group.

Correct Example

```
*SELEC 1=FCES 1.OR.STABS 1
*SELEC 2=FCES 2.OR.STABS 2
```

Incorrect Example

```
*SELEC 1=FCES 1.AND.STABS 1
*SELEC 2=FCES 2.OR.STABS 2/1/2
```

11. The structure definition for a redundant group must use the connector .AND. and conclude with a redundancy specification in the form /M/N indicating M subgroup failures are required to cause failure of the entire group comprised of N subgroups. The value of M must be less than or equal to the number (N) of subgroup names on the right side of the equal sign on the card.

Correct Example

```
*SBAT GENR=BATTERYS.AND.GENRATRS/2/2
*SBATTERYS=L BATTRY.AND.R BATTRY/2/2
*SGENRATRS=L GENR.AND.R GENR/2/2
```

Incorrect Example

```
*SBAT GENR=BATTERYS.OR.GENRATRS/2/2
*SBATTERYS=L BATTRY.AND.R BATTRY
*SGENRATRS=L GENR.AND.R GENR/3/2
```

Table 3-1. Rules for Assembling the Fault Tree Input (Continued)

---

12. No more than eight subgroups can comprise one redundant group (defined on one card using .AND. connectors).

    Correct Example

    ```
    *SRGROUP=C1.AND.C2.AND.C3.AND.C4.AND.C5.AND.C6.AND.C7.AND.C8/5/8
    ```

    Incorrect Example

    ```
    *SRGROUP=C1.AND.C2.AND.C3.AND.C4.AND.C5.AND.C6.AND.C7.AND.C8.AND.C9/5/9
    ```

13. Do not mix .AND. and .OR. connectors on the same card.

    Correct Example

    ```
    *SELEC 3    = STABS 1.AND.STABS 2.AND.STABS 3.AND.STABS 4 /3/4
    *SMECH CTR = LON LK'S .OR. CABLES .OR. LAT LK'S
    ```

    Incorrect Example

    ```
    *SELEC 3    = STABS 1.AND.STABS 2.OR.STABS 3.AND.STABS 4 /3/4
    *SMECH CTR = LON LK'S .AND. CABLES .OR. LAT LK'S
    ```

14. If no connectors are used on a card, then no blanks are allowed between the equal sign and the name field that follows it.

    Correct Example

    ```
    CC4=LAT LK A
    ```

    Incorrect Example

    ```
    CC4= LAT LK A
    ```

15. If a card contains a connector, then either use a blank between the equal sign and the first name field in the structure definition, or place the left period of the first connector (.AND. or .OR.) with less than 10 columns between it and the equal sign.

    Correct Example

    ```
    *SMECH CTR =LON LK'S .OR. CABLES .OR. LAT LK'S
    *SLON LK'S = LON LK A.AND.LON LK F /2/2
    ```

    Incorrect Example

    ```
    *SMECH CTR =LON LK'S  .OR. CABLES .OR. LAT LK'S
    *SLON LK'S =LON LK A  .AND.LON LK F /2/2
    ```

Table 3-1. Rules for Assembling the Fault Tree Input (Concluded)

---

16. No characters past column 80 of the input file are read.

17. Except on the Group Definition Card, do not use a name on the left side of the equal sign unless it has appeared in the structure definition (right side of the equal sign) on a preceding card for the current kill category fault tree descripton.

    Correct Example

    ```
    *GATTRN=FLT CNTL.OR.AFT ENGN
    *SFLT CNTL=ELEC CTR.AND.MECH CTR/2/2
    *SELEC CTR=ELEC LNK.OR.BAT GENR
    *SELEC LNK=ELEC 1.AND.ELEC 2.AND.ELEC 3.AND.ELEC 4/3/4
    ```

    Incorrect Example

    ```
    *SELEC LNK=ELEC 1.AND.ELEC 2.AND.ELEC 3.AND.ELEC 4/3/4
    *SELEC CTR=ELEC LNK.OR.BAT GENR
    *SFLT CNTL=ELEC CTR.AND.MECH CTR/2/2
    *GATTRN=FLT CNTL.OR.AFT ENGN
    ```

18. The entire fault tree description must not contain any undefined names. A name is defined by either using it on the left side of an equal sign on a Subgroup Definition Card, or by being a component name on one of the Component Name and Location Cards (Card 13) in the input deck.

19. Only components and subgroups listed in the fault tree description are included in the fault tree. There are no default components or structures.

FILE10 - BINARY INPUT FLIGHT PATH

The second input file for this program is a binary file read from Logical Unit #10. It contains data describing the aircraft at each time step of the flight path as well as an indicator showing which intervals of the flight path can be engaged by the ground weapon. This file is produced by executing the Engagement Model and consists of two types of records described in Figures 3-4 and 3-5. The top two rows of these figures represent a tape divided into numbered records, with the length of each record listed in the second row. The bottom part of the figures is used to list the units and definitions for each FORTRAN variable whose value is written on a tape record. The first record, described in Figure 3-4, contains an alphanumeric title which is used to identify the flight path file. The second and all subsequent records are in the format described in Figure 3-5. This record contains data describing the aircraft at six consecutive time increments of the flight path with 16 parameters defined in the figure.

RECORD NUMBER	1	2	3	...	LAST
NUMBER OF WORDS	20	96	96	...	96

Record Number 1		Title Record	
WORD	PARAMETER	UNITS	DEFINITION
1	TITLE(1)	---	80 character alphanumeric title describing the flight path file contents
2	TITLE(2)	---	
3	TITLE(3)	---	
.	.	.	
.	.	.	
.	.	.	
20	TITLE(20)	---	

FIGURE 3-4. FILE10 Flight Path File, Record 1 (Page 1 of 1)

RECORD NUMBER	1	2	3		LAST
NUMBER OF WORDS	20	96	96		96

Record Number 2			
WORD	PARAMETER	UNITS	DEFINITION
1	OTAPE(1,1)	seconds	Time at which the next 15 words of data are pertinent
2	OTAPE(2,1)	meters	x-coordinate of the aircraft in the Flight Path Coordinate System at time OTAPE(1,1)
3	OTAPE(3,1)	meters	y-coordinate of the aircraft in the Flight Path Coordinate System at time OTAPE(1,1)
4	OTAPE(4,1)	meters	z-coordinate of the aircraft in the Flight Path Coordinate System at time OTAPE(1,1)
5	OTAPE(5,1)	m/sec	x-component of the aircraft velocity vector
6	OTAPE(6,1)	m/sec	y-component of the aircraft velocity vector
7	OTAPE(7,1)	m/sec	z-component of the aircraft velocity vector
8	OTAPE(8,1)	m/sec^2	x-component of the aircraft acceleration vector
9	OTAPE(9,1)	m/sec^2	y-component of the aircraft acceleration vector

FIGURE 3-6. FILE10 Flight Path File, Record 2 (Page 1 of 4)

RECORD NUMBER	1	2	3	- - -	LAST
NUMBER OF WORDS	20	96	96	- - -	96

Record Number 2

WORD	PARAMETER	UNITS	DEFINITION
10	OTAPE(10,1)	m/sec^2	z-component of the aircraft acceleration vector;
11	OTAPE(11,1)	m/sec	Aircraft speed; the magnitude of OTAPE(11,1) is the aircraft speed at time OTAPE(1,1): =0.0, indicates the end of the flight path file; >0.0, indicates the aircraft is able to be engaged <0.0, indicates the aircraft cannot be engaged NOTE: The Flight Path File used as input for the Engagement Model has all values of OTAPE(11,I) $\geq$ 0.0
12	OTAPE(12,1)	---	Normal load factor on the aircraft
13	OTAPE(13,1)	radians	Aircraft azimuth angle; heading angle of the flight path in the Flight Path Coordinate System at time OTAPE(1,1)
14	OTAPE(14,1)	radians	Aircraft dive angle; angle between the flight path and the horizontal XY-plane of the Flight Path Coordinate System; positive value indicates decreasing altitude

FIGURE 3-6. FILE10 Flight Path File, Record 2 (Page 2 of 4)

RECORD NUMBER	1	2	3	...	LAST
NUMBER OF WORDS	20	96	96	...	96

Record Number 2			
WORD	PARAMETER	UNITS	DEFINITION
15	OTAPE(15,1)	radians	Aircraft roll angle; amount of aircraft rotation about the longitudinal axis of the fuselage
16	OTAPE(16,1)	radians	Aircraft angle of attack
17	OTAPE(1,2)	seconds	Second flight path time; the values of OTAPE(2,2), OTAPE(3,2), . . . OTAPE(16,2) describe the aircraft at time OTAPE(1,2)
.	.	.	.
.	.	.	.
.	.	.	.
33	OTAPE(1,3)	seconds	Third flight path time; the values of OTAPE(2,3), OTAPE(3,3), . . . OTAPE(16,3) describe the aircraft at time OTAPE(1,3)
.	.	.	.
.	.	.	.
.	.	.	.
81	OTAPE(1,6)	seconds	Sixth flight path time; the values of OTAPE(2,6), OTAPE(3,6), . . . OTAPE(16,6) describe the aircraft at time OTAPE(1,6)

FIGURE 3-5. FILE10 Flight Path File, Record 2 (Page 3 of 4)

RECORD NUMBER	1	2	3	...	LAST
NUMBER OF WORDS	20	96	96		96

Record Number 2

WORD	PARAMETER	UNITS	DEFINITION
.	.	.	.
.	.	.	.
.	.	.	.
96	OTAPE(16,6)	radians	Aircraft angle of attack at time OTAPE(1,6) NOTE: All remaining records on the Flight Path File are in the same format as Record 2, each containing flight path data for the next six time increments. The last record has the value of OTAPE(11,I) equal to 0.0 to indicate the end of the flight path.

FIGURE 3-5. FILE10 Flight Path File, Record 2 (Page 4 of 4)

SECTION IV

OUTPUT

Two output files are produced by executing the ASALT-I Model. The line printer output is written on Logical Unit #6 and contains a description of the input parameters as well as the simulation results in a readable form. The first subsection below describes the line printer output, FILE6, by showing examples and outlining the options available for the various parts of this output. The second output file, FILE11, is a binary sequential file written on Logical Unit #11. It contains values for the amount of laser energy that reaches the target during each time increment in the simulation. The second subsection is used to define the parameters whose values are written on FILE11 and describe their order so that an analyst could use a post processor to interpret and perform a more detailed analysis with these data.

FILE6 - LINE PRINTER OUTPUT

The line printer output can be divided into three parts: a description of the input parameters; a time history of the laser and aircraft encounter; and a damage summary. The three following subsections are used to describe these three parts and include an example of each.

Description of the Input Parameters

The first section of line printer output is a description of the input parameters. This information is always printed and provides the user with a good description of the conditions being evaluated. This output is generated by executing WRITE statements in Subroutines READY, ACIN, and EROMUL. An example of this section of output is shown in Figure 4-1. The title of the computer model appears at the top of the first page. The first subsection lists the flight path file name from the first record of the file, and the data used to convert points from the Flight Path Coordinate System to the General Coordinate System (coordinate systems are defined in Section II). The next subsection contains values defining the laser weapon system including: its location in the General Coordinate System; the tracking error caused by jitter; the emission rates as a function of time; the slewing rate limits; and the minimum prefire tracking time. A description of the atmospheric conditions is listed next. This includes the attenuation factors as a function of range, as well as the smoke corridor location and attenuation factor. If no smoke corridor is modeled by the input values on Card 10, the two lines describing the smoke

```
* * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                   *
* ASSESSMENT OF SURVIVABILITY AGAINST LASER THREATS *
*                                                   *
* * * * * * * * * * * * * * * * * * * * * * * * * * *
```

AIRCRAFT FLIGHT PATHS
FLIGHT PATH FILE = SAMPLE FLIGHT PATH FILE FOR ASALT-I DOCUMENTATION
TRANSFORMATION = (X,Y,Z) = (-1000., -500., 0.) IF FLIGHT PATH COORDINATE SYSTEM IS EQUAL TO
(X,Y,Z) = (0., 0., 0.) IN GENERAL COORDINATE SYSTEM WITH .0 DEGREES ROTATION ANGLE

LASER WEAPONS
LOCATION = (X,Y,Z) = (-1500., 7000., 2952.)

AIM POINT STANDARD DEVIATION IN THE ENCOUNTER PLANE DUE TO JITTER SIGMA-Y = 2.00 MILS SIGMA-Z = 1.00 MILS

FLUX EMISSION IN KILOWATTS/SQ.CM.	3.50	4.00	4.20	4.50	5.00	5.50	6.00	6.50	8.00
AT TIMES (IN SECONDS)	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0

TRACKING = MAXIMUM SLEW RATES IN DEGREES PER SECOND AZIMUTH = 45.00 ELEVATION = 10.00
MINIMUM TRACKING TIME = 1.00 SECONDS

ATMOSPHERES

ATTENUATION FACTORS	1.000	.980	.960	.930	.850	.780	.721	.650	.600
AT RANGES (METERS)	1000.	2000.	4000.	6000.	8000.	10000.	15000.	20000.	40000.

SMOKE CONDITION = PNC. COORDINATES (0., 3000.) TO (-5000., 0.) IN THE XY-PLANE
ATTENUATION FACTOR THROUGH SMOKE = .10000

AIRCRAFT COMPONENTS

LOCATION IN AIRCRAFT C.S. / DAMAGING ENERGY LEVELS IN KILOJOULES/SQ.CM.

					.00	.02	.15	1.00	3.00	6.00	7.00	8.00	10.00	20.00
1 IR S NPG	2.50	.00	.50	PK =	.00	.15	.20	.25	.33	.40	.60	.70	.80	.90
2 AFT AL	-2.00	.10	.10	PK =	.00	.15	.20	.25	.40	.55	.66	.85	.50	.76
3 AFT ML	-4.00	.10	.00	PK =	.01	.12	.10	.20	.40	.60	.70	.80	.96	.95
4 AFT UM	-4.00	-.10	.00	PK =	.00	.12	.10	.20	.40	.61	.76	.80	.90	.95
5 FCES 1	2.00	.00	.50	PK =	.00	.15	.20	.25	.33	.46	.66	.76	.86	.96
6 STANS 1	-2.00	.00	.10	PK =	.00	.15	.20	.25	.30	.35	.40	.45	.56	.70
7 FCES 2	2.00	-.10	.50	PK =	.00	.15	.20	.25	.33	.41	.56	.70	.86	.90
8 STANS 2	-2.00	-.10	.10	PK =	.00	.15	.20	.25	.30	.35	.40	.45	.50	.70
9 FCES 3	2.00	.10	.50	PK =	.00	.15	.20	.25	.33	.40	.60	.70	.80	.90
10 STANS 3	-2.00	.10	.10	PK =	.00	.15	.20	.25	.30	.35	.40	.45	.50	.70
11 FCES 4	1.00	.00	.50	PK =	.00	.15	.20	.25	.33	.46	.66	.76	.80	.96
12 STANS 4	-3.00	.10	.10	PK =	.00	.15	.20	.25	.30	.35	.40	.45	.54	.70
13 AFT SL	-5.00	.10	.00	PK =	.00	.12	.10	.20	.40	.64	.70	.80	.96	.95
14 AFT AW	-2.00	-.10	.10	PK =	.00	.15	.20	.25	.30	.35	.40	.45	.54	.70
15 AFT SM	-5.00	-.10	.00	PK =	.00	.12	.10	.20	.40	.60	.70	.80	.96	.95
16 L FCCUMP	.00	2.00	.00	PK =	.00	.02	.13	.20	.40	.56	.60	.70	.86	.96
17 R FCCUMP	.00	-2.00	.00	PK =	.00	.12	.23	.30	.40	.50	.66	.76	.80	.96
18 L MATINY	2.00	.30	-.10	PK =	.00	.02	.13	.20	.40	.50	.60	.70	.86	.90
19 R MATINY	2.00	-.30	-.10	PK =	.00	.12	.23	.30	.40	.56	.60	.70	.86	.90
20 L HEMM	2.00	2.00	.00	PK =	.00	.02	.13	.20	.40	.50	.60	.76	.80	.90
21 R HEMM	2.00	-2.00	.00	PK =	.00	.12	.23	.30	.40	.50	.60	.70	.80	.90
22 LC. CAML	.10	-2.00	.00	PK =	.00	.12	.23	.30	.41	.56	.60	.70	.90	.90
23 LON LN A	.00	-2.00	.00	PK =	.00	.12	.23	.30	.40	.50	.66	.76	.86	.96
24 LON LN F	.00	-1.00	.00	PK =	.00	.12	.23	.30	.40	.50	.61	.76	.80	.90
25 LAT CAML	.10	2.00	.00	PK =	.00	.12	.23	.30	.40	.51	.66	.76	.86	.90
26 LAT LN A	.00	2.00	.00	PK =	.00	.12	.23	.30	.40	.50	.60	.70	.80	.96
27 LAT LN F	.00	1.00	.00	PK =	.00	.12	.23	.30	.40	.56	.60	.76	.81	.90

Figure 4-1. Example Output - Description of Input Parameters (Page 1 of 5).

AIM POINTS IN AIRCRAFT	LOCATION IN AIRCRAFT COOR.			SIGMA-Y (MILS)	SIGMA-Z (MILS)	ENGAGEABLE ENVELOPE TO HIT THE AIM POINT IN DEG. AZIMUTH	ELEVATION
1	5.00	.00	.00	1.00	.50	90.0 TO 270.0	90.0 TO 175.0
2	1.00	.00	-.20	2.00	1.00	45.0 TO 315.0	.0 TO [illegible]
3	-1.00	.00	-.20	3.00	2.00	270.0 TO 180.0	.0 TO [illegible]
4	.00	.00	1.00	2.00	2.00	.0 TO 360.0	110.0 TO 140.0
5	.00	2.00	.10	1.00	1.00	180.0 TO 360.0	90.0 TO 180.0
6	.00	2.00	-.10	1.00	1.00	180.0 TO 360.0	.0 TO 90.0
7	.00	-2.00	.10	4.00	2.00	.0 TO 180.0	90.0 TO 180.0
8	.00	-2.00	-.10	4.00	2.00	.0 TO 180.0	.0 TO 90.0

Figure 4-1. Example Output - Description of Input Parameters (Page 2 of 5).

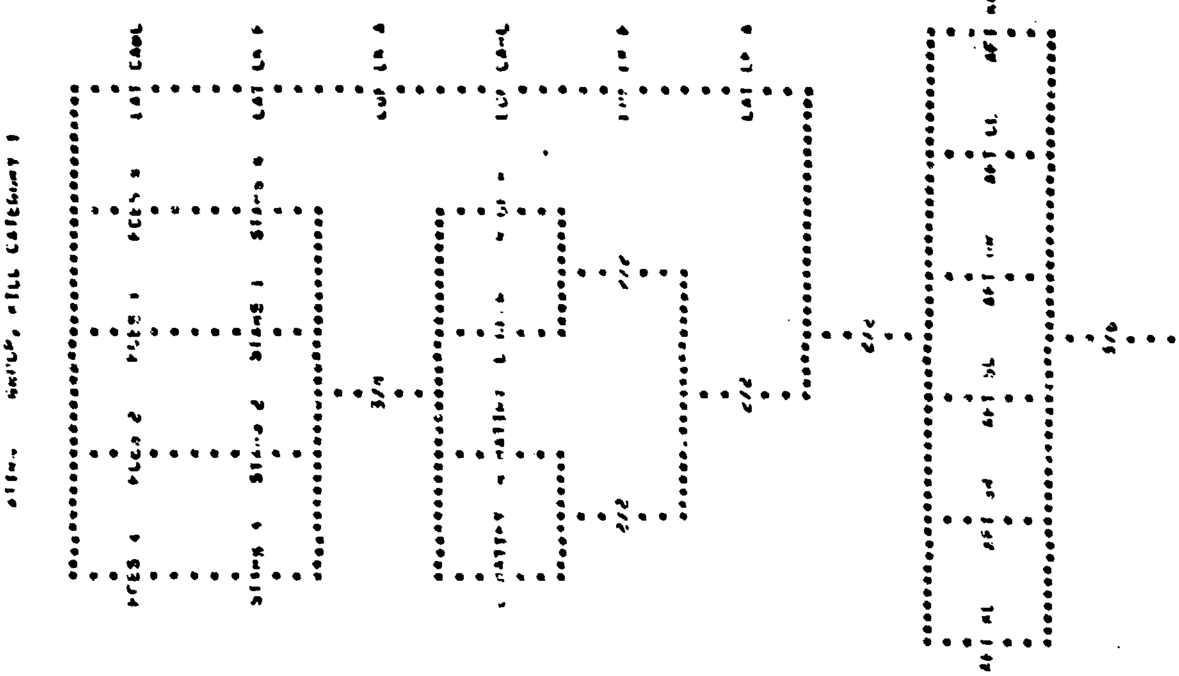


Figure 4-1. Example Output - Description of Input Parameters (Page 3 of 5).

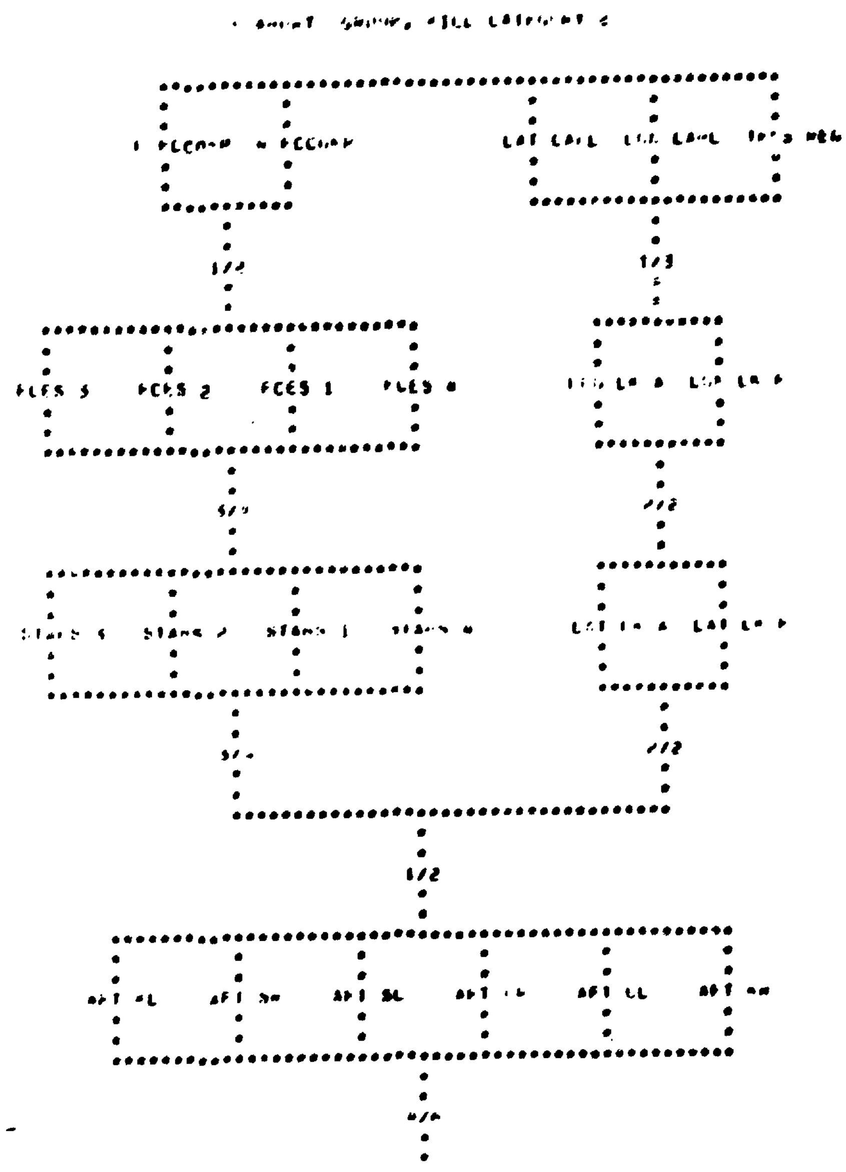


Figure 4-1. Example Output - Description of Input Parameters (Page 4 of 5).

```
TIME STEPS:
        BETWEEN COMPUTATIONS   =   .[illegible] SECOND
        BETWEEN PRINTOUT LINES =  1.[illegible] SECOND
```

Figure 4-1. Example Output - Description of Input Parameters (Page 5 of 5).

corridor are omitted. The aircraft components are described in the next subsection. The component number, name, and location in the Aircraft Coordinate System are listed in the left five columns, and its probabilities of kill at ten levels of accumulated energy are listed in the right ten columns. The aim points expressed in the Aircraft Coordinate System are listed in the next subsection. The errors associated with locating and tracking each aim point are printed in mils in the columns labeled SIGMA-Y and SIGMA-Z. Additionally, the limits on the look-angle envelope for firing at each aim point are listed in degrees.

The next two pages of Figure 4-1 are example fault tree diagrams which are printed by executing Subroutine EKOMUL and are used to depict the interdependence of the aircraft components. These fault trees were generated using the fault tree input shown in Figure 3-3. A user of the ASALT-I program may define as many as three fault tree structures for different kill categories. Each fault tree in this section of output is labeled at the top with an eight character group name followed by the word "GROUP" and a number to identify its kill category. The number at the top of the fault tree is the same kill category number which appears in later sections of the program output. The first fault tree in Figure 4-1 has the group name, ATTRN, and is kill category number 1. The second fault tree has the group name, M ABORT, and is kill category number 2.

A set of component names in one vertical line on the fault tree is a series (singly vulnerable subgroup) in which the failure of any one component is sufficient to cause failure of the entire subgroup. Redundant components which comprise a multiply vulnerable subgroup are represented by parallel vertical lines on the fault tree. The redundancy code is printed at the bottom of each set of vertical lines. In the first example fault tree of Figure 4-1, the subgroup at the bottom of the fault tree contains six components: AFT KL, AFT SR, AFT SL, AFT UR, AFT UL, and AFT KR. Its redundancy specification is 3/6 which means failure of the subgroup requires the failure of three or more components in that subgroup. Users of the ASALT-I program may create very elaborate fault trees using many levels of subgroups as exemplified in Figure 4-1. If the fault tree trace option is selected by the user on Card 11 of the input deck, the printing of each fault tree is preceded by several extra parameters used in interpreting the fault tree structure cards.

The final page of Figure 4-1 is an example of the subsection used to display the time steps for the run. These values are specified by the user on Card 1 of the input deck and are used to control the simulated time between computation iterations and between lines in the time trace output. If no time trace output is requested, the line labeled TIME BETWEEN PRINTOUT LINES is omitted.

Time History of the Laser and Aircraft Encounter

Figure 4-2 is an example of this section of line printer output. These data are printed by executing Subroutines HEADER and OUTPUT, and are omitted if the value, 0, is specified on Card 1 for the parameter IPRINT. The top three lines in Figure 4-2 are a heading printed at the top of each line printer page. The four left columns list the time and location of the aircraft at that instant in the simulation. The slant range in meters between the laser weapon and the aircraft is listed in the fifth column. The interval between consecutive lines in this section of output is determined by the input parameter values on Card 1. The column labeled STATUS may contain five possible entries as shown in Figure 4-2. The "NOT ENGAGE" status occurs for all aircraft locations which cannot be engaged by the laser weapon system. The "NOT ENGAGE" status is determined by execution of the Engagement Model and is detected through the parameter values on the Flight Path input file. The status "TRACKING" occurs when the aircraft can be engaged but the minimum prefire tracking time is not yet fulfilled. If the slewing rate required for the laser system to track the aircraft exceeds the user specified maximum, then the status column will contain the label "TRACK ERR". If none of these conditions occur, the laser system fires at the aircraft and the status column contains the label "ENGAGE", unless the smoke corridor is between the laser and aircraft. When this occurs the status is labeled "SMOKE". Whenever the status is either "SMOKE" or "ENGAGE", the probability of kill values for each kill category of the total aircraft are printed in the right hand columns. One Pk value is printed for each aim point, and each value represents the total target Pk for the kill category, which results from one laser system attempting to fire at one aim point since the beginning of the simulation.

Damage Summary

The damage summary is the last section of line printer output produced by executing the ASALT-I Model. It is printed by executing Subroutine SUMMRY and displays the values for the total target probability of kill for each kill category as well as subgroup and component Pk's for each aim point. An example of this section of output is shown in Figure 4-3, where the matrix of numbers on the right side are the Pk values. Each column corresponds to a different aim point. The total aircraft probabilities of kill for each kill category are listed in the top lines and are labeled with the kill category number and name on the left side. The subgroup Pk's are listed next with each line identified by its

TIME	AIRCRAFT LOCATION X	Y	Z	SLANT RANGE	STATUS	KILL CATEGORY	TOTAL AIRCRAFT PK FROM A LASER FIRED AT EACH AIM POINT 1	2	3	4	5	6	7	8	9	10
50.00	411.	2631.	3747.	1258.	NOT ENGAGE											
50.50	412.	2631.	3746.	1258.	NOT ENGAGE											
51.00	413.	2632.	3746.	1258.	NOT ENGAGE											
51.50	414.	2632.	3746.	1257.	TRACKING											
52.00	415.	2633.	3746.	1257.	TRACKING											
52.50	416.	2635.	3745.	1256.	TRACK ENG											
53.00	417.	2634.	3745.	1256.	TRACK ENG											
53.50	418.	2634.	3745.	1256.	TRACK ENG											
54.00	419.	2635.	3744.	1255.	TRACKING											
54.50	420.	2635.	3744.	1255.	TRACKING											
55.00	421.	2636.	3744.	1255.	ENGAGE	1	.06	.08	[illegible]	[illegible]	.08	.09	.08	.07	.08	.08
						2	.08	.06	.09	.16	.07	.06	.07	.08	.07	.08
						3	.09	.07	.08	.10	[illegible]	.09	.08	.07	.10	.07
55.50	422.	2636.	3745.	1254.	ENGAGE	1	.08	.08	.08	.07	.08	.09	.08	.07	.08	.09
						2	.09	.08	.09	.18	.07	.06	.07	.08	.07	.08
						3	.06	.07	.08	.10	.08	.09	.10	.07	.08	.07
56.00	423.	2637.	3745.	1254.	ENGAGE	1	.09	.09	.19	.08	.13	.10	.09	.08	.09	.10
						2	.10	.19	.11	.09	.08	.07	.08	.09	.08	.09
						3	.10	.08	.09	.10	.06	.10	.07	.08	.09	.08
56.50	424.	2637.	3743.	1254.	ENGAGE	1	.09	.09	.19	.18	.06	.10	.09	.08	.09	.10
						2	.10	.09	.10	.19	[illegible]	.07	.08	.09	.08	.09
						3	.10	.08	.09	.10	.09	.10	.07	.08	.09	.08
57.00	425.	2638.	3743.	1253.	ENGAGE	1	.10	.10	.10	.19	.10	.11	.10	.09	.10	.11
						2	.11	.10	.11	.10	.15	.08	.09	.10	.09	.10
						3	.11	.09	.10	.11	.11	.11	.06	.09	.10	.09
57.50	426.	2639.	3742.	1253.	ENGAGE	1	.10	.10	.10	.09	.10	.11	.10	.09	.10	.11
						2	.11	.10	.11	.10	.09	.08	.09	.10	.09	.10
						3	.11	.09	.10	.11	.10	.11	.08	.09	.10	.09
58.00	427.	2639.	3742.	1253.	ENGAGE	1	.11	.11	.11	.11	.11	.12	.11	.10	.11	.12
						2	.12	.11	.12	.11	.10	.09	.10	.11	.10	.11
						3	.12	.10	.11	.12	.11	.12	.09	.10	.11	.10
58.50	428.	2639.	3742.	1252.	ENGAGE	1	.11	.11	.11	.10	.11	.12	.11	.10	.11	.12
						2	.12	.11	.12	.11	.10	.09	.10	.11	.10	.11
						3	.12	.10	.11	.12	.11	.12	.09	.11	.11	.10
59.00	429.	2640.	3741.	1252.	ENGAGE	1	.12	.12	.12	.11	.12	.13	.12	.11	.12	.13
						2	.13	.12	.13	.12	.11	.10	.11	.12	.11	.12
						3	.13	.11	.12	.13	.12	.13	.10	.11	.12	.11
59.50	430.	2640.	3741.	1252.	ENGAGE	1	.12	.12	.12	.11	.12	.13	.12	.11	.12	.13
						2	.13	.12	.13	.12	.11	.10	.11	.12	.11	.12
						3	.13	.11	.12	.13	.12	.13	.10	.11	.12	.11

Figure 4-2. Example Output - Time Trace of the Encounter.

DAMAGE SUMMARY AT EACH AIM POINTS

			AIM POINTS 1	2	3	4	5	6	7	8
TOTAL AIRCRAFT PK'S										
KILL CATEGORY 1	ATTRN	GROUP	.14	.11	.00	.06	.15	.14	.00	.00
KILL CATEGORY 2	K ARCNT	GROUP	.67	.83	.60	.03	.66	.88	.06	.00
SUBGROUP PK'S										
KILL CATEGORY 1		AFT FAGP	.08	.08	.00	.00	.05	.08	.00	.00
		MECH CTR	.09	.86	.79	.05	.08	.90	.09	.06
		GENRATNS	.13	.08	.03	.00	.10	.12	.00	.06
		BATTERYS	.13	.09	.05	.00	.12	.13	.00	.00
		RAT GEAR	.02	.01	.00	.00	.01	.02	.00	.00
		ELEC 4	.31	.29	.07	.00	.30	.31	.00	.00
		ELEC 3	.31	.29	.07	.00	.30	.31	.00	.00
		ELEC 2	.31	.29	.07	.00	.30	.31	.00	.00
		ELEC 1	.31	.29	.07	.00	.30	.31	.00	.00
		ELEC LNR	.09	.08	.00	.00	.09	.09	.00	.00
		ELEC CTR	.11	.08	.06	.00	.19	.10	.00	.00
		FLT CNTL	.09	.07	.00	.00	.08	.69	.00	.00
KILL CATEGORY 2		AFT LNGR	.01	.00	.00	.00	.01	.01	.00	.00
		LAT LN'S	.09	.08	.04	.00	.09	.10	.00	.00
		CABLES	.61	.57	.57	.02	.59	.61	.08	.00
		LON LN'S	.09	.08	.08	.00	.09	.10	.00	.00
		MECH CTR	.88	.85	.62	.02	.88	.88	.00	.00
		ELEC 4	.01	.01	.00	.00	.01	.02	.00	.00
		ELEC 1	.02	.01	.00	.00	.02	.02	.00	.00
		ELEC LNR	.03	.03	.00	.00	.03	.03	.00	.00
		FL CEFS	.59	.51	.51	.01	.58	.60	.00	.00
		FLT CNTR	.66	.92	.51	.01	.59	.61	.00	.00
		FLT CNTL	.57	.84	.60	.03	.88	.88	.00	.00
COMPNT KILL PK'S										
	1	TRIM REG	.19	.16	.03	.00	.16	.17	.00	.00
	2	AFT RL	.15	.14	.05	.00	.15	.14	.00	.00
	3	AFT LL	.15	.14	.05	.00	.16	.16	.00	.00
	4	AFT UR	.15	.14	.05	.00	.16	.16	.00	.00
	5	FCES 1	.16	.14	.03	.00	.18	.17	.00	.00
	6	STABS 1	.15	.16	.05	.00	.15	.16	.00	.00
	7	FCES 2	.15	.16	.03	.00	.18	.17	.00	.00
	8	STABS 2	.15	.16	.05	.00	.15	.16	.00	.00
	9	FCES 3	.18	.16	.03	.00	.18	.17	.00	.00
	10	STABS 3	.15	.16	.05	.00	.15	.16	.00	.00
	11	FCES 4	.18	.16	.03	.00	.13	.17	.00	.00
	12	STABS 4	.15	.16	.05	.00	.15	.16	.00	.00
	13	AFT SL	.16	.18	.05	.00	.16	.16	.00	.00
	14	AFT RW	.15	.16	.05	.00	.15	.16	.00	.00
	15	AFT BW	.18	.14	.05	.00	.16	.16	.00	.00
	16	L FCCOMP	.40	.42	.14	.00	.46	.42	.00	.00
	17	R FCCOMP	.41	.28	.20	.01	.29	.31	.00	.00
	18	L BATTRY	.43	.32	.14	.00	.39	.41	.00	.00
	19	R BATTRY	.32	.26	.14	.01	.33	.31	.00	.00
	20	L GEAR	.41	.29	.14	.04	.35	.40	.00	.00
	21	R GEAR	.42	.28	.19	.01	.30	.31	.01	.00
	22	LON CABL	.31	.26	.20	.01	.30	.31	.00	.00
	23	LON LK A	.31	.28	.20	.01	.29	.31	.00	.00
	24	LON LK F	.31	.29	.20	.01	.30	.31	.00	.00
	25	LAT CABL	.36	.28	.26	.01	.30	.32	.00	.00
	26	LAT LK A	.30	.20	.20	.01	.30	.32	.00	.00
	27	LAT LK F	.36	.20	.20	.01	.30	.31	.00	.00

Figure 4-3. Example Output - Damage Summary.

subgroup name. Only subgroups which were defined on a Subgroup Definition Card with an asterisk in column 1 are included in this section of output (See Table 3-1, Rule 7). The component Pk's are listed last and are identified by component number and name.

FILE 11 - INCREMENTAL ENERGY FILE

The second output file produced by executing the ASALT-I Model is a binary sequential file written on Logical Unit #11. The first record on this file contains values for the time step, number of aim points, number of components, as well as locations of the aim points and components in the Aircraft Coordinate System. The remainder of the file contains one record for every time step used by the model in simulating the encounter between the laser weapon system and the aircraft. Each of these records contains values for the current simulation time and the amount of laser energy reaching each component during the time increment. Figure 4-4 and 4-5 contain record descriptor forms which are used to show the order of the values on each record and their definitions. All records following the second record are in the same format as Record 2.

Record Number	1	2	---	Last
Number of Words	3+3*NAIMPT +3*NCOMP	1+NAIMPT* NCOMP	---	1+NAIMPT *NCOMP

Record Number: 1

WORD	PARAMETER	UNITS	DEFINITION
1	TDELT	seconds	Time interval between each iteration of the program computations; one record is written on this file for each iteration
2	NAIMPT	---	Number of aim points on the target
3	AIM(1,1)	meters	x-coordinate of the first aim point in the Aircraft Coordinate System
4	AIM(2,1,)	meters	y-coordinate of the first aim point in the Aircraft Coordinate System
5	AIM(3,1)	meters	z-coordinate of the first aim point in the Aircraft Coordinate System
6	AIM(1,2)	meters	x-coordinate of the second aim point in the Aircraft Coordinate System
.	.	.	.
.	.	.	.
.	.	.	.
3*NAIMPT	AIM(1, NAIMPT)	meters	x-coordinate of the last aim point in the Aircraft Coordinate System
1+ 3*NAIMPT	AIM(2, NAIMPT)	meters	y-coordinate of the last aim point in the Aircraft Coordinate System
2+ 3*NAIMPT	AIM(3, NAIMPT)	meters	z-coordinate of the last aim point in the Aircraft Coordinate System
3+ 3*NAIMPT	NCOMP	---	Number of components in the target model

Figure 4-4. FILE11 - Incremental Energy File, Record 1.

Record Number	1	2	Last
Number of Words	3+3*NAIMPT +3*NCOMP	1+NAIMPT* NCOMP	1+NAIMPT *NCOMP

Record Number: 1 (Concluded)

WORD	PARAMETER	UNITS	DEFINITION
4+ 3* NAIMPT	COMP(1,1)	meters	x-coordinate of the first component in the Aircraft Coordinate System
5+ 3* NAIMPT	COMP(2,1)	meters	y-coordinate of the first component in the Aircraft Coordinate System
6+ 3* NAIMPT	COMP(3,1)	meters	z-coordinate of the first component in the Aircraft Coordinate System
.	.	.	.
.	.	.	.
.	.	.	.
1+ 3* NAIMPT+3* NCOMP	COMP(1, NCOMP)	meters	x-coordinate of the last component in the Aircraft Coordinate System
2+ 3* NAIMPT+3* NCOMP	COMP(2, NCOMP)	meters	y-coordinate of the last component in the Aircraft Coordinate System
3+ 3* NAIMPT+3* NCOMP	COMP(3, NCOMP)	meters	z-coordinate of the last component in the Aircraft Coordinate System

Figure 4-4. FILE11 - Incremental Energy File, Record 1.

Record Number	1	2		Last
Number of Words	3+3*NAIMPT +3*NCOMP	1+NAIMPT* NCOMP		1+NAIMPT*NCOMP

Record Number: 2

WORD	PARAMETER	UNITS	DEFINITION
1	TIME	seconds	Current time in the simulation model; the end time of the time slice for the following energy values.
2	ENGYAD(1,1)	kilo-watts/ cm^2	Amount of laser energy reaching component 1 from a laser aimed at aim point 1 during the time step.
3	ENGYAD(1,2)	kilo-watts/ cm^2	Amount of laser energy reaching component 1 from a laser aimed at aim point 2 during the time step.
.	.	.	.
.	.	.	.
.	.	.	.
NAIMPT+1	ENGYAD(1, NAIMPT)	kilo-watts/ cm^2	Amount of laser energy reaching component 1 from a laser aimed at aim point NAIMPT during the time step.
NAIMPT+2	ENGYAD(2,1)	kilo-watts/ cm^2	Amount of laser energy reaching component 2 from a laser aimed at aim point 1 during the time step.
.	.	.	.
.	.	.	.
.	.	.	.
(1+NAIMPT* NCOMP	ENGYAD(NCOMP, NAIMPT)	kilo-watts/ cm^2	Amount of laser energy reaching component NCOMP from a laser aimed at aim point NAIMPT during the time step.

Figure 4-5. FILE11 - Incremental Energy File, Record 2.

APPENDIX A

SOURCE LISTING

This appendix contains a source listing (pages A-2 through A-52) with comment cards for the ASALT-I Model. Program ASALT is listed first, followed by all subroutines in alphabetical order. Each subprogram listing begins on a new page.

```
C***
C       -- ASSESSMENT OF SURVIVABILITY AGAINST LASER THREATS --
C      THIS PROGRAM ACCEPTS OUTPUT FROM THE ENGAGEMENT MODEL AND
C      PROVIDES A MEASURE OF AN AIRCRAFT'S SURVIVABILITY AGAINST A
C      LASER THREAT
C***
       COMMON /AIMPTS/ NAIMPT, AIM(3,10), SIGMA(10,2),
      *                AZLIM(10,2), ELLIM(10,2)
       COMMON /AIRCFT/ NCOMP, COMP(5,100), AP(100,20), WIDTH(100,20),
      *                ENGTHN(100,10)
       COMMON /ING/ LU, NAM(297), IU(2200), LOCNM(297), JENUS(3),
      *           NGROUP
       CHARACTER*6 NAM
       COMMON /LASER/  GUN(3), GUNTAN(3), FFLUX, FLUX(10), FLTIME(10),
      *                FLUXEM, FLUXGN
       COMMON /RECALL/ IOLDST, NSHOTS
       COMMON /STATUS/ ISTAT
C***
C      STATUS DEFINITION TABLE
C       = 0, END OF FLIGHT PATH
C       = 1, CANNOT ENGAGE
C       = 2, INSUFFICIENT TRACK TIME
C       = 3, SLEW RATE LIMIT EXCEEDED
C       = 4, FIRING THROUGH SMOKE
C       = 5, FIRING
C***
       COMMON /TAPE10/ DTAPE(16,N), TLAST(16), TTAP, TARGET(3),
      *                TXDOT, TYDOT, TZDOT, TXDDOT, TYDDOT, TZDDOT,
      *                TSPEED, TLOAT, TAZ, TELEV, THULL, TAA,
      *                IHI, ILO, LEAST
       COMMON /TRACK/  TRKTIM, SLEWAZ, SLEWEL, TRSTRT, YJITTR, ZJITTR
       COMMON /TRANSF/ XEP, YEP, XG, YG, ZG, PSI, CP, SP, GTOAC(3,3),
      *                ACTOG(3,4), AFTOE(3,4), ETOAC(3,3)
       COMMON /TSTEPS/ TDELT, IPRINT, NOUT
       LOGICAL CANHIT
C***
C      INITIALIZE
C***
       FLUXEM = 0.0
       FLUXGN = 0.0
       IOLDST = 0
       NSHOTS = 0
       IHI = 2
       ILO = 1
C***
C      READ THE DATA DECK AND PERFORM PRELIMINARY COMPUTATIONS AND
C      PRINT RUN PARAMETERS
C***
       CALL READY
       RANGE = DIST(TARGET, GUN)
       GO TO 900
C***
C      GET AIRCRAFT POSITION DATA FOR CURRENT TIME
C***
   100 TIME = TIME + TDELT
```

```
      CALL POINTS
      IF (ISTAT .EQ. 0) GO TO 990
      RANGE = DISS(TARGET, GUN)
C***
C     DROP TO END OF TIME LOOP IF CANNOT ENGAGE
C***
      IF (ISTAT .EQ. 1) GO TO 900
C***
C     TRACK MODULE, IF TRACK TIME AND SLEW RATE TESTS ARE SATISFIED THEN
C     BEGIN AIM POINT AND COMPONENT LOOPS.
C***
      CALL VPSV(GUNTAR, TARGET, -1.0, GUN)
      CALL TRACK1
      IF (ISTAT .LE. 3) GO TO 900
      CALL MATRIX(GTOAC, ACTOG, TAZ, TDIVE, TROLL)
C***
C     BEAM PROPAGATION MODULE
C***
      THETA = ATAN2( GUNTAR(1), GUNTAR(2) )
      FLUXLM = COMPUT( FLUX, FLTIME, (TIME-TRSTRT-TRKTIM), NFLUX )
      CALL PROPAT(RANGE, THETA)
C***
C     DAMAGE MODULE, EVALUATE FOR EACH COMPONENT AT EACH AIM POINT
C***
      DO 800 IAIM = 1, NAIMPT
      CALL CONANG(AIMAZ, AIMEL, GUNTAR, AIM(1,IAIM))
      IF ( .NOT. CANHIT(AIMAZ, AIMEL, IAIM) ) GO TO 800
      CALL ARRANM(SIGY, SIGZ, AIMRGE, AIMAZ, AIMEL, IAIM)
      DO 700 ICOMP = 1, NCOMP
      EXPTIM = TOFLT * PHIT(ICOMP, IAIM, SIGY, SIGZ, AIMRGE)
      CALL DAMAGE(ICOMP, IAIM, EXPTIM, FLUXON)
  700 CONTINUE
C***
C     COMPUTE AIRCRAFT PK USING THE FAULT TREE DESCRIPTIONS
C***
      CALL FALTRE(IAIM)
  800 CONTINUE
C***
C     END OF TIME LOOP, UPDATE STATISTICS AND PRINT INTERIM RESULTS
C***
  900 CALL UPDATE( TRSTRT, TIME )
      CALL OUTPUT(NAIMPT, NCOMP, RANGE)
      GO TO 100
C***
C     END OF FLIGHT PATH, PRINT SUMMARY AND STOP
C***
  990 CALL SUMMRY(NAIMPT, NCOMP)
      STOP
      END
```

```
      SUBROUTINE ACIN
C***
C     THIS SUBROUTINE IS USED TO READ THE AIRCRAFT COMPONENT
C     SPECIFICATIONS AND AIM POINTS.  ALL COORDINATES ARE IN THE
C     AIRCRAFT COORDINATE SYSTEM.
C***
      COMMON /AIMPTS/ NAIMPT, AIM(3,10), SIGMA(10,2),
     *                AZLIM(10,2), ELLIM(10,2)
      COMMON /AIRCFT/ NCOMP, COMP(3,100), AP(100,26), AWIDTH(100,26),
     *                ENGYON(100,10)
      COMMON /DAMAGE/ ENERGY(10), PK(10,100), FRSTAD(100,10)
      COMMON /IND/ LU, NAM(297), NCI(2200), LCOMP(297), JENDS(3),
     *             NGROUP
      CHARACTER*8 NAM
      COMMON /TSTEPS/ TDELT, IPRINT, NUMHIT
      DATA DTOR, RADMIL/0.01745329, 0.9817477E-03/
C***
C     READ THE NUMBER OF COMPONENTS, AIM POINTS, AND ENERGY ARGUMENTS
C     FOR THE PK TABLE
C***
      READ (5,810) NCOMP, NAIMPT, ITABLE
      READ (5,800) (ENERGY(J), J=1,10)
      WRITE (6,910) (ENERGY(J), J=1,10)
      DO 10 I = 1,NCOMP
C***
C     COMPONENT LOCATIONS
C***
      READ (5,811) NAM(I), (COMP(J,I), J=1,3)
C***
C     PRESENTED AREAS AND WIDTHS (IN ENCOUNTER PLANE) AT 26 ASPECTS
C***
      READ (5,800) (AP(I,J), AWIDTH(I,J), J=1,26)
C***
C     PK VERSUS ENERGY TABLE
C***
      READ (5,800) (PK(J,I), J=1,10)
C***
C     INITIALIZE THE ENGYON ARRAY AND PRINT COMPONENT LOCATIONS AND PK
C***
      DO 8 J = 1,10
      ENGYON(I,J) = 0.0
    8 CONTINUE
      WRITE (6,911) I, NAM(I), (COMP(J,I), J=1,3), (PK(J,I),J=1,10)
   10 CONTINUE
      LU = NCOMP
C***
C     READ AIM POINTS, AIM POINT SIGMAS, AND AIM POINT ENVELOPES
C                             (IN MILS)               (IN DEGREES)
C
C***
      WRITE (6,900)
      DO 70 I = 1,NAIMPT
      READ (5,800) AIM(1,I), AIM(2,I), AIM(3,I), SIGMA(I,1), ELLIM(I,2)
     *       SIGMA(I,2), AZLIM(I,1), AZLIM(I,2), ELLIM(I,1),
      WRITE (6,910) I, (AIM(J,I), J=1,3), SIGMA(I,1), SIGMA(I,2),
     *                AZLIM(I,1), AZLIM(I,2), ELLIM(I,1), ELLIM(I,2)
      DO 60 K = 1,2
```

```
      SIGMA(I,K) = SIGMA(I,K) + RANGIL
      AZLIM(I,K) = AZLIM(I,K) + [illegible]
      ELLIM(I,K) = ELLIM(I,K) + [illegible]
   60 CONTINUE
   70 CONTINUE
C***
C     INITIALIZE ENGYAD ARRAY AND WRITE FIRST RECORD ON DEVICE 11
C***
      DO 80 J = 1,10
      DO 80 I = 1,100
      ENGYAD(I,J) = 0.0
   80 CONTINUE
      WRITE (11) IDELT, NAIMPT, ((AIM(I,J), I=1,3), J=1,NAIMPT),
     *                  NCOMP, ((COMP(I,J), I=1,3), J=1,NCOMP)
      CALL MVIMPT(ITRACE)
      RETURN
C***
C     FORMATS
C***
  110 FORMAT (21H0AIRCRAFT COMPONENTS, 50X,
              43H0DAMAGING ENERGY LEVELS IN KILOJOULES/SQ.CM. /
     *        17X, 25HLOCATION IN AIRCRAFT C.S., 7X, 10F8.2)
     *        17X, 25HLOCATION IN AIRCRAFT C.S., 3F8.2, 7H    RR =, 10F8.2)
  211 FORMAT (1X, I5, 1X, A8, 4X, 3F8.2, 7H    RR =, 10F8.2)
  300 FORMAT (28H0AIM POINTS IN AIRCRAFT, 11X, 15HSIGMA=Y SIGMA=Z, 5X,
              40H0COURSE-ANGLE ENVELOPE TO HIT THE AIM POINT /
     *        7X, 25HLOCATION IN AIRCRAFT C.S.,    (MILS)  (MILS),
     *        7X, 25HLOCATION IN AIRCRAFT C.S.    IN DEG.)
     *        8X, 7HAZIMUTH, 8X, 14HELEVATION, 2(3X, F5.1, 4H TO , F5.1))
  310 FORMAT (I6, 2X, 3F8.2, 5X, 2F8.2, 1X, 2(3X, F5.1, 4H TO , F5.1))
  400 FORMAT (10E8.0)
  410 FORMAT (10I8)
  411 FORMAT (A8, 3E8.0)
      END
```

```
      SUBROUTINE APSEUP(SIGY, SIGZ, AIMRNG, AIMAZ, AIMEL, IAIM)
      COMMON /AIMPTS/ NAIMPT, AIM(3,10), SIGMA(10,2),
     *                AZLIM(10,2), ELLIM(10,2)
      COMMON /LASER/  GUN(3), GUNTAR(3), NFLUX, FLUX(10), FLTIME(10),
     *                FLUXEP, FLUXLA
      COMMON /TRACK/  INRTIM, SLEWAZ, SLEWEL, TRKINT, YJITTR, ZJITTR
      COMMON /TRANSF/ XFP, YFP, XG, YG, ZG, PSI, CP, SP, GTOAC(3,3),
     *                ACTOG(3,3), ACTOE(3,3), ETOAC(3,3)
      DIMENSION A(3)
C***
C     COMPUTE THE MATRIX FOR TRANSFORMATION FROM THE AIRCRAFT TO THE
C     ENCOUNTER COORDINATE SYSTEM
C***
      CALL MATRIX(ACTOE, ETOAC, (AIMAZ-3.14159265), (1.57079633-AIMEL),
     *            0.0)
C***
C     COMPUTE TOTAL STANDARD DEVIATIONS IN THE ENCOUNTER C.S. IN RADIANS
C***
      SIGY = SQRT( SIGMA(IAIM,1)**2 + YJITTR**2)
      SIGZ = SQRT( SIGMA(IAIM,2)**2 + ZJITTR**2)
C***
C     COMPUTE RANGE TO THE AIM POINT
C***
      CALL VXMAT(A, AIM(1,IAIM), ACTOG)
      CALL VMSV(A, A, 1.0, GUNTAR)
      AIMRGE = VECMAG(A)
      RETURN
      END
```

```
      SUBROUTINE DAMAGE(ICOMP, IAIM, EXPTIM, FLUXON)
      COMMON /DAMAGE/ ENERGY(10), PK(10,100), ENGYAD(100,10)
      COMMON /AIRCFT/ NCOMP, COMP(3,100), AP(100,20), WIDTH(100,20),
     *                ENGYON(100,10)
      COMMON /ALLPKS/ PKV(20, 3, 10)
C***
C     ACCUMULATE EXPECTED ENERGY ON THE COMPONENT WHEN FIRING
C     AT THE CURRENT AIM POINT AND DETERMINE PK
C***
      ENGYAD(ICOMP,IAIM) = EXPTIM * FLUXON
      ENGYON(ICOMP, IAIM) = ENGYON(ICOMP, IAIM) + ENGYAD(ICOMP,IAIM)
C***
C     COMPUTE COMPONENT PK - A FUNCTION OF ACCUMULATED ENERGY
C     ASSUME SAME COMPONENT PK IN ALL KILL CATEGORIES
C***
      PKV(ICOMP, 1, IAIM) = COMPUT( PK(1,ICOMP), ENERGY,
     *                      ENGYON(ICOMP,IAIM), 10)
      PKV(ICOMP, 2, IAIM) = PKV(ICOMP, 1, IAIM)
      PKV(ICOMP, 3, IAIM) = PKV(ICOMP, 1, IAIM)
      RETURN
      END
```

```
      SUBROUTINE PRFMUL(L1,L2,ITRACE)
      CHARACTER*8 NAM
      CHARACTER*8 BLANK
      INTEGER ACN, DAN
      DIMENSION ACN(L2),LOC(L2),DAN(L2),ICH(130)
      DIMENSION IDIG(9)
      CHARACTER*1 IDIG, ICH
      COMMON /IMG/ LU, NAM(297), FUL(2200), LOGRP(297), JENDS(5),
     *             NGROUP
      CHARACTER*1 IHL, IV, IH, ISL
      DATA IDIG/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
      DATA BLANK,IHL,IV,IH,ISL/8H        ,1H ,1H.,1H.,1H//
C
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
C         THIS SUBROUTINE CONSTRUCTS A PRINTER-PLOT OF THE USERS
C         FAULT DIAGRAM, ACCORDING TO ITS INTERPRETATION STORED IN
C         THE FUL ARRAY.
C
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C***
C         FIRST INITIALIZE ACN, DAN, AND LOC
C***
      L21 = L2-1
      M1 = L21
   10 IT = FUL(M1)
      NENT1 = MOD(IABS(IT),10)+1
      LOC(M1)=0
      M11 = M1-NENT1
C***
C         POINT TO SUBHEAD
C***
      ACN(M1) = M11
      DAN(M1) = 0
      DO 30 M2 = 1,NENT1
      M3 = M1-M2
      MENT = 1
      IF(FUL(M3).LE.LU) GO TO 20
C***
C         NOT A PRIMITIVE
C***
      MENT = -1
C***
C         PRIMITIVE
C***
20    ACN(M3) = MENT
      DAN(M3) = MENT
   30 LOC(M3) = 0
      IF(M11.LE.L1) GO TO 40
      M1 = M11-1
      GO TO 10
C***
C         ACCUMULATE ACN AND DAN
C***
   40 M1 = L21
   50 IT = FUL(M1)
```

```
      NENT = MOD(IABS(IT),10)
      M11 = ACW(M1)
      JMI = 1
      IF(IT.LT.0) JMI = 0
      JMD = 1
      IF(IT.GT.0) JMD = 0
      DO 130 M2=1,NENT
      JM = 1
      JN = 1
      M3 = M1+M2
      IF(MUL(M3).LE.LN) GO TO 110
C***
C        NOT A PRIMITIVE ... FIND SIZE
C***
      N1 = L21
   60 N11 = ACR(N1)
       IF(MUL(N11).NE.MUL(M3)) GO TO 70
       IF(ACR(N11).LE.0) GO TO 80
      GO TO 100
   70 IF(N11.LE.1) GO TO 80
      N1 = N11-1
      GO TO 60
C***
C        ERROR ... OUT OF ORDER
C***
   80 WRITE (6,90) MUL(M3)
      WRITE (6,231)
       DO 13 M=1,L2
   13  WRITE (6,232) M,MUL(M),ACR(M),LWR(M),LUP(M)
      STOP
  100 JM = DWN(N11)
      JN = ACR(N11)
   90 FORMAT (20H   OUT OF ORDER ...  ,I6)
      DWN(M3) = JM
      ACR(M3) = JN
  110 IF(IT.LT.0) GO TO 120
C***
C        PARALLEL ... ACCUMULATE WIDTH
C***
      JMD = JMD+JN
      JMI=MAX0(JMI,JM)
      GO TO 130
  120 JMI = JMI+JM
      JMD=MAX0(JMD,JN)
C***
C        SERIES ... ACCUMULATE HEIGHT
C***
  130 CONTINUE
      ACR(M11) = JMD
      DWN(M11) = JMI
      IF(IT.GT.0) DWN(M11) = DWN(M11)+1
       IF(M11.LE.L1) GO TO 140
      M1 = M11-1
      GO TO 50
C***
C        NOW HAVE WHOLE GROUP SIZE ... JMD ENTRIES WIDE,JMI ENTRIES
C
```

```
C        TALL
C***
  140 IF(JMD.LE.13) GO TO 155
      WRITE (6,150)
  150 FORMAT (30H    TOO MANY ENTRIES FOR GROUP PICTURE )
      RETURN
C***
C        CONSTRUCT PICTURE
C***
  155 LOC(L1)=99000
      LMUTMOM(L1)=0*0
      DO 230 M1 = L1,L21
      IF(NAN(M1).NE.0) GO TO 230
      M11 = ACN(M1)
      IT = MUL(M1)
      MENT = MOD(IABS(IT),10)
      IF(LOC(M11).NE.0) GO TO 190
C***
C        MUST LOCATE PREVIOUS REFERENCE TO THIS BLOCK
C***
      N1=M11-1
  160 N11 = ACN(N1)
      NM1 = N1-1
      DO 170 K=N11,NM1
      IF(MUL(K).EQ.MUL(M11)) GO TO 180
  170 CONTINUE
      IF(N11.LE.1) GO TO 174
      N1 = N11-1
      GO TO 160
  174 WRITE (6,175) MUL(M11)
  175 FORMAT (20H    CANNOT POSITION, I5)
       STOP
  180 LOC(M11) = LOC(K)
C***
C        HAVE LOCATION OF COLLECTION ... NOW POSITION ENTRIES
C***
  190 ICC=LOC(M11)/1000
      ILL=LOC(M11)-ICC*1000-6
      LEFT = ICC-ACN(M11)*5
      M12 = M11+1
      M13 = M1-1
      IF(IT.LT.0) GO TO 210
C***
C        PARALLEL COLLECTION
C***
      DO 200 N=M12,M13
       LWID = ACN(N)*10
      LOC(N)=LEFT+LWID/2*1000+ILL+6
  200 LEFT = LEFT+LWID
       GO TO 230
C***
C        SERIES COLLECTION
C***
  210 DO 220 N=M12,M13
      LM1 = NAN(N)*6
      LOC(N)=ICC*1000+ILL+6
```

```
  220 ILL = ILL+LMI
  230 CONTINUE
C***
C        FOR TRACE OPTION- PRINT NUL, ACN, LAN, AND LOC ARRAYS
C***
      IF(ITRACE.NE.1) GO TO 239
      WRITE (6,231)
  231 FORMAT(30H)    N    NUL   ACN   LAN       LOC ,//)
      DO 23 N=L1,L2
   23 WRITE (6,232) N,NUL(N),ACN(N),LAN(N),LOC(N)
  232 FORMAT(1X,4I5,I10)
  239 CONTINUE
C***
C        NOW PRINT PAGE
C***
      IG = NUL(L1)
      KC = NUL(L2)
      WRITE(6,240) NAM(IG),KC
  240 FORMAT (1H1,50X,A8,21H GROUP, KILL CATEGORY ,I2//)
      LINE = 3
  250 LINE = LINE+1
      IF(LINE.GT.LBOT) GO TO 410
C***
C        ANY ENTRIES ON THIS LINE ?
C***
      DO 290 N=L1,L21
       IF(DAN(N).EQ.0) GO TO 290
       IF(NUL(N).GT.LQ) GO TO 290
      IC=LOC(N)/1000
      IL=LOC(N)-IC*1000
      IF(IL.NE.LINE-4) GO TO 290
C***
C        HERES ONE ... PRINT IT
C***
      IN = NUL(N)
       IF(IL.EQ.IC/10*10) GO TO 270
C***
C        WORD CENTERS ON COLUMN X5 ... STARTS AT X0
C***
      NBL = IC/10
      WRITE (6,260) (BLANK,K=1,NBL),NAM(IN)
  260 FORMAT(1H+,13(2X,A8))
      GO TO 290
C***
C        WORD CENTERS ON COLUMN X0 ... STARTS AT X0-5
C***
  270 NBL = IC/10-1
      WRITE (6,280) (BLANK,K=1,NBL),NAM(IN)
  280 FORMAT(1H+,5X,13(2X,A8))
  290 CONTINUE
C***
C        FIRST INITIALIZE - ICH ARRAY
C***
      DO 300 I=1,130
  300 ICH(I) = THL
C***
```

```
C        CHECK FOR ITEM LEADERS FIRST
C***
      DO 310 N=1,L21
      IF(ORN(N).EQ.0) GO TO 310
      IF(MUL(N).GT.0) GO TO 310
      IC=LOC(N)/1000
      IL=LOC(N)-IC*1000
      IF(IL.GT.LINE)GO TO 310
      IF(IL.LT.LINE-5) GO TO 310
      IF(IL.EQ.LINE-3) GO TO 310
C***
C        NEED VERTICAL AT IC
C***
      ICH(IC) = IV
  310 CONTINUE
C***
C        NOW CHECK FOR HORIZONTAL CONNECTORS
C***
      DO 350 N=1,L21
      IF(ORN(N).NE.0) GO TO 350
      IF(MUL(N).LT.0) GO TO 350
C***
C        HAVE A PARALLEL COLLECTION ... DOES IT AFFECT THIS LINE ?
C***
      N1 = ACN(N)
      IT=MOD(LOC(N1),1000)
      IF(IT.EQ.LINE) GO TO 320
      IM=IT+ORN(N1)*6-6
      IF(IM.NE.LINE) GO TO 350
C***
C        ADD HORIZONTAL TO LINE
C***
  320 I1 = 150
      I2 = 1
      N11 = N1+1
      N12 = N-1
      DO 330 I=N11,N12
      IC=LOC(I)/1000
      IF(IC.LT.I1) I1=IC
  330 IF(IC.GT.I2) I2 = IC
      DO 340 I=I1,I2
  340 ICH(I) = IH
  350 CONTINUE
C***
C        EXTEND VERTICAL TAILS TO COMPLETE COLLECTIONS, ADD IPAR
C***
      DO 390 N=1,L21
      IF(ORN(N).NE.0) GO TO 390
      N1 = ACN(N)
      IC=LOC(N1)/1000
      IT=LOC(N1)-IC*1000
      IM=IT+ORN(N1)*6-6
      IF(MUL(N).LT.0) GO TO 390
      IF(LINE.NE.IM+6) GO TO 355
C***
C        HAVE PARALLEL ... THIS IS THE IPAR LINE
```

```
C***
      ILFT = MUL(N)/10
      IRT = MUL(N)-ILFT*10
      LU=IC-2
      WRITE (6,360) (IML,K=1,IC),IDIG(ILFT),ISL,IDIG(IRT)
  360 FORMAT(1H+,130A1)
  355 CONTINUE
      IF(LINE.EQ.(N+1)) GO TO 390
      IF(LINE.GT.IN.AND.LINE.LT.IC+N) ICH(IC)=IV
      IF(LINE.GT.IN) GO TO 390
      N11 = N1+1
      N12 = N-1
      DO 380 I=N11,N12
      IC1=LOC(I)/1000
      IT1=LOC(I)-IC1*1000
      IN1 = IT1+OWN(I)*6
      IF(LINE.LT.IN1) GO TO 380
C***
C        NOT IPAR ... ADD TAIL SEGMENT
C***
      ICH(IC1) = IV
  380 CONTINUE
  390 CONTINUE
      WRITE (6,400) ICH
  400 FORMAT(1X,130A1)
      GO TO 250
C***
C        DONE WITH PICTURE ... RETURN
C***
  410 RETURN
      END
```

```
      SUBROUTINE FALTRS(IAIM)
      COMMON /ING/ LO, NAM(297), MUL(2000), LIGRP(297), JENDS(5),
     *             NGROUP
      CHARACTER*8 NAM
      COMMON /AILPKS/ PMV(297, 5, 10)
      DIMENSION PKK(8)
C***
C     EVALUATE TOTAL AIRCRAFT PK USING FAULT TREE DEFINED IN MUL ARRAY
C***
      DO 500 IGRP = 1,NGROUP
      L = JENDS(IGRP)
      KILLG = MUL(L)
      N1 = 1
      IF (IGRP .GT. 1) N1 = JENDS(IGRP-1) + 1
  100 L = L - 1
      MVR = MUL(L)
      IF (MVR .LT. 0) GO TO 300
C***
C     PARALLEL SUBGROUP
C***
      LREQ = MVR/10
      LSYS = MOD( MVR, 10)
      DO 200 ISYS = 1,LSYS
      L = L - 1
      PKK(ISYS) = PMV( MUL(L), KILLG, IAIM)
  200 CONTINUE
C***
      IF (LSYS .GE. 8) GO TO 220
      LSYS1 = LSYS + 1
      DO 210 ISYS = LSYS1, 8
      PKK(ISYS) = 0.0
  210 CONTINUE
  220 CALL PVMART( LREQ, PKK, PGROUP, LSYS)
      L = L - 1
      PMV( MUL(L), KILLG, IAIM) = PGROUP
      GO TO 450
C***
C     SERIES SUBGROUP
C***
  300 PP = 1.0
      MVR = -MVR
      DO 400 ISYS = 1, MVR
      L = L - 1
      P2 = PMV( MUL(L), KILLG, IAIM)
      IF (P2 .LE. 0.0) GO TO 400
      PP = PP * (1.0-P2)
  400 CONTINUE
      L = L - 1
      PMV( MUL(L), KILLG, IAIM) = 1.0 - PP
  450 IF (L .GT. N1) GO TO 100
  500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GETNAM(ICARD,N1,N2,NAME,NT,NF)
      CHARACTER*1 ICARD,NAME
      CHARACTER*1 IBL,IEQ,IPER,ISL
      DIMENSION ICARD (80),NAME(8)
      DATA IBL,IEQ,IPER,ISL/1H ,1H=,1H.,1H//
C
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
C          THIS SUBROUTINE DECODES THE NEXT FIELD OF THE ALPHA INPUT
C          CARD CONTAINING MULTIPLY VULNERABLE GROUP DESCRIPTIONS.
C
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
      NN = 0
      ND = 0
      N = N1-1
      NT = 0
      DO 5 M=1,8
    5 NAME(M) = IBL
   10 N = N+1
      IF(N-N1.GT.8) RETURN
      IF(N.GT.80) RETURN
      IF(ICARD(N).NE.IBL) GO TO 20
C***
C          TRAILING BLANK
C***
      IF(ND.EQ.0) N1 = N+1
      IF(ND.NE.0) NN=NN+1
      N2 = N
      GO TO 10
C***
C          NOT A BLANK ... SEPARATOR ?
C***
   20 IF(ICARD(N).NE.IEQ) GO TO 30
      IF(ND.NE.0) RETURN
C***
C          LEADING EQUAL SIGN ... THIS IS A NAME COMING
C***
      NF=1
      NT = 1
      N1=N+1
      GO TO 10
C***
C          NOT AN EQUAL SIGN ... PERIOD ?
C***
   30 IF(ICARD(N).NE.IPER) GO TO 50
C***
C          LEADING PERIOD ?
C***
      IF(ND.NE.0) GO TO 40
C***
C          YES ... LEADING PERIOD ... MUST BE CONNECTIVE
C***
      NT = 2
      N1=N+1
      GO TO 10
C***
```

```
C          TRAILING PERIOD ... DONE FOR NOW
C***
   40 N2=N-1
      IF(NT.EQ.2) N2=N
      RETURN
C***
C          NOT AN EQUAL SIGN OR A PERIOD ... SLASH ?
C***
   50 IF(ICARD(N).NE.ISL) GO TO 60
      IF(NO.NE.0) RETURN
C***
C          LEADING SLASH ... REDUNDANCY SPECIFICATION COMING
C***
      NAME(1) = ICARD(N+1)
      NAME(2) = ICARD(N+2)
      NAME(3) = ICARD(N+3)
      N1 = N+1
      N2 = N+3
      NT = 3
      RETURN
C***
C        NO CONNECTIVE ... LOAD DATA CHARACTER
C***
   60 NO=1
      IF(NT.EQ.0) NT=1
      NN = NN+1
      N2 = N
      NAME (NN) = ICARD(N)
      GO TO 10
      END
```

```
      SUBROUTINE HEADER
      COMMON /AIMPTS/ NAIMPT, AIM(3,10), SIGMA(10,2),
     *                AZLIM(10,2), ELLIM(10,2)
C***
C     CAUSE A LINE PRINTER PAGE EJECT AND PRINT COLUMN HEADINGS
C     FOR THE NEW PAGE OF OUTPUT
C***
      WRITE (6,100) (I, I=1,NAIMPT)
      WRITE (6,101)
  100 FORMAT(1H1,11X, 17HAIRCRAFT LOCATION,6X, 6HSLANT,15X, 6HRILL,6X,
     *       58HTOTAL AIRCRAFT OR SHIP A LASER FIRED AT EACH AIM POINT /
     *       7H   TIME, 5X, 1HX, 7X, 1HY, 7X, 1HZ, 6X, 5HRANGE, 4X,
     *       17HSTATUS   CATEGORY, I4, 9I7)
  101 FORMAT(1X, 128(1H=))
      RETURN
      END
```

```
      SUBROUTINE INT26( PAREA, WIDE, COMPAZ, COMPEL, ICOMP )
      COMMON /AIRCFT/ NCOMP, COMP(3,100), AP(100,26), WIDTH(100,26),
     *                ENGVUN(100,10)
      DIMENSION F(2), IN(2)
      DATA QTRPI/0.78539816/
C***
C     INTERPOLATE THE COMPONENT PRESENTED AREA AND WIDTH (Y DIRECTION
C     IN THE ENCOUNTER C.S.), AT LOOK ANGLES COMPAZ AND COMPEL, FROM
C     THE AP AND WIDTH ARRAYS.  THESE ARRAYS CONTAIN DATA AT 26 LOOK-
C     ANGLES ARRANGED AS FOLLOWS:
C
C   INDEX LOOK-AZIMUTH LOOK-ELEVATION  INDEX LOOK-AZIMUTH LOOK-ELEVATION
C     1         0             0         14       180            90
C     2         0            45         15       225            90
C     3        45            45         16       270            90
C     4        90            45         17       315            90
C     5       135            45         18         0           135
C     6       180            45         19        45           135
C     7       225            45         20        90           135
C     8       270            45         21       135           135
C     9       315            45         22       180           135
C    10         0            90         23       225           135
C    11        45            90         24       270           135
C    12        90            90         25       315           135
C    13       135            90         26         0           180
C***
C     STATEMENT FUNCTION USED IN THE 2-DIMENSIONAL INTERPOLATION
C***
      STA(A1,A2,FM) = A1 + FM*(A2-A1)
C***
      IF (COMPEL .LT. 3.14159265) GO TO 5
      PAREA = AP(ICOMP,26)
      WIDE  = WIDTH(ICOMP,26)
      RETURN
    5 F(1) = COMPAZ / QTRPI
      F(2) = COMPEL / QTRPI
      DO 10 I = 1,2
      IN(I) = IFIX(F(I))
      F(I) = F(I) - FLOAT(IN(I))
      IN(I) = IN(I) + 1
   10 CONTINUE
      INDEX1 = 1
      INDEX2 = 1
      IF (IN(2) .EQ. 1) GO TO 20
      INDEX1 = IN(1) + IN(2)*8 - 15
      INDEX2 = INDEX1 + 1
      IF (IN(1) .EQ. 8) INDEX2 = INDEX1 - 7
   20 INDEX3 = 26
      INDEX4 = 26
      IF (IN(2) .EQ. 4) GO TO 30
      INDEX3 = IN(1) + IN(2)*8 - 7
      INDEX4 = INDEX3 + 1
      IF (IN(1) .EQ. 8) INDEX4 = INDEX3 - 7
   30 RES1 = STA( AP(ICOMP,INDEX1), AP(ICOMP,INDEX2), F(1))
      RES2 = STA( AP(ICOMP,INDEX3), AP(ICOMP,INDEX4), F(1))
      PAREA = STA( RES1, RES2, F(2) )
```

```
RES1 = STAT( WIDTH(ICOMP,INDEX1), WIDTH(ICOMP,INDEX2), F(1))
RES2 = STAT( WIDTH(ICOMP,INDEX3), WIDTH(ICOMP,INDEX4), F(1))
VALUE = STAT( RES1, RES2, F(2) )
RETURN
END
```

```
      SUBROUTINE LOKANG(AZLOK, ELLOK, GUNTAR, V)
      DIMENSION GUNTAR(3), V(3), D(3)
      COMMON /TRANSF/ XFP, YFP, XG, YG, ZG, PSI, CP, SP, GTUAC(3,3),
     *                ACTUG(3,3), ACTUL(3,3), FTUAC(3,3)
      DATA EPS / 0.000001 /
C***
C     COMPUTE LOOK ANGLES (AZLOK AND ELLOK) IN THE AIRCRAFT
C     COORDINATE SYSTEM, FOR THE VECTOR FROM THE GUN TO THE END OF
C     VECTOR V (IN THE AIRCRAFT C.S.).  GUNTAR IS THE VECTOR FROM THE
C     GUN TO THE TARGET.
C***
C     TRANSFORM VECTOR GUNTAR INTO THE AIRCRAFT C.S.
C***
      CALL VXMAT(D, GUNTAR, GTUAC)
C***
C     ADD VECTOR V
C***
      CALL VPSV(D, D, 1.0, V)
C***
C     COMPUTE LOOK ANGLES
C***
      AZLOK = 0.0
      IF (ABS(D(1)).LT.EPS .AND. ABS(D(2)).LT.EPS) GO TO 20
      AZLOK = ATAN2(D(2),D(1))
      IF (AZLOK .LT. 0.0) AZLOK = AZLOK + 6.2831853
   20 ELLOK = 1.57079633 - ATAN2( D(3), SQRT(D(1)**2 + D(2)**2) )
      RETURN
      END
```

```
      SUBROUTINE MATCH (NAME,N,NO,M)
      CHARACTER*8  NAM
      CHARACTER*1 NAME,NAME
      COMMON /TNG/ LN, NAM(297), ITL(2200), IGRP(297), JENDS(3),
     *              NGROUP
      DIMENSION NAME(8),NAME(8)
C
C  *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
C
C          THIS SUBROUTINE LOOKS FOR A CORRESPONDENCE BETWEEN
C          THE NAME ARRAY (NA1) AND AN ENTRY IN THE NAM ARRAY (AN).
C
C  *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
      M = 0
      DO 100 K=1,NNO
      READ(NAM(K),20) (NAME(I),I=1,N)
   20 FORMAT (8A1)
      DO 10 L=1,N
      IF(NAME(L).NE.NAME(L)) GO TO 100
   10 CONTINUE
      M = K
      RETURN
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MATRIX(TRANS, TRANSI, YAW, DIVE, ROLL)
      DIMENSION TRANS(3,3), TRANSI(3,3)
C***
C     COMPUTE TRANSFORMATION MATRIX, TRANS, AND ITS INVERSE, TRANSI,
C     BETWEEN TWO COORDINATE SYSTEMS.  ANGLES YAW, DIVE, AND ROLL
C     RELATE THE TWO SYSTEMS, MEASURING FROM THE OLD SYSTEM TO THE
C     NEW SYSTEM:
C      YAW - ROTATES THE XY-PLANE SO THE X-AXIS MOVES TOWARD THE Y-AXIS
C      DIVE- ROTATES THE XZ-PLANE SO THE Z-AXIS MOVES TOWARD THE X-AXIS
C      ROLL- ROTATES THE YZ-PLANE SO THE Y-AXIS MOVES TOWARD THE Z-AXIS
C***
      CY = COS(YAW)
      SY = SIN(YAW)
      CD = COS(DIVE)
      SD = SIN(DIVE)
      CR = COS(ROLL)
      SR = SIN(ROLL)
C***
C     COMPUTE MATRIX ELEMENTS, ROTATION ORDER IS YAW, DIVE, AND THEN
C     ROLL.
C***
      TRANS(1,1) = CY*CD
      TRANS(2,1) = SY*CD
      TRANS(3,1) = -SD
      TRANS(1,2) = CY*SD*SR - SY*CR
      TRANS(2,2) = SY*SD*SR + CY*CR
      TRANS(3,2) = CD*SR
      TRANS(1,3) = CY*SD*CR + SY*SR
      TRANS(2,3) = SY*SD*CR - CY*SR
      TRANS(3,3) = CD*CR
C***
C     COMPUTE INVERSE MATRIX
C***
      DO 10 I = 1,3
      DO 10 J = 1,3
      TRANSI(J,I) = TRANS(I,J)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MVHART(NWPNS, PKM, PKG, NSYS)
      DIMENSION PKM(8)
      NUM = 2**NWPN
      PKG = 0.0
      GO TO (10, 20, 30, 40, 50, 60, 70, 80), NSYS
   10 M7 = 1
   20 M6 = 1
   30 M5 = 1
   40 M4 = 1
   50 M3 = 1
   60 M2 = 1
   70 M1 = 1
      GO TO (150, 140, 130, 120, 110, 100, 90), NSYS
   80 M1 = 2
   90 M2 = 2
  100 M3 = 2
  110 M4 = 2
  120 M5 = 2
  130 M6 = 2
  140 M7 = 2
  150 M8 = 2
C***
C     PERFORM CONDITIONAL PK COMPUTATIONS
C***
      DO 160 N1 = 1,M1
      PKM(8) = 1.0 - PKM(8)
      DO 160 N2 = 1,M2
      PKM(7) = 1.0 - PKM(7)
      DO 160 N3 = 1,M3
      PKM(6) = 1.0 - PKM(6)
      DO 160 N4 = 1,M4
      PKM(5) = 1.0 - PKM(5)
      DO 160 N5 = 1,M5
      PKM(4) = 1.0 - PKM(4)
      DO 160 N6 = 1,M6
      PKM(3) = 1.0 - PKM(3)
      DO 160 N7 = 1,M7
      PKM(2) = 1.0 - PKM(2)
      DO 160 N8 = 1,M8
      PKM(1) = 1.0 - PKM(1)
      IF ( N1+N2+N3+N4+N5+N6+N7+N8 .LT. NUM ) GO TO 160
      PKG = PKG + PKM(1)*PKM(2)*PKM(3)*PKM(4)*PKM(5)*PKM(6)*
     *            PKM(7)*PKM(8)
  160 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MVINPT(ITRACE)
      CHARACTER*8 NAM,NNNAM
      DIMENSION NUL(9)
      CHARACTER*1 ICARD,NAME
      CHARACTER*1 I1,I2,I3,ISTAR,IG,IS,IC,INTGR
      CHARACTER*1 IE,IN,ID,IBL,IA,JEQ
      DIMENSION NAME(8),IWORK(50)
      DIMENSION ICARD(81),NLU(15),INTGR(9)
      COMMON /ING/ LU, NAM(297), NUL(2200), LCGRP(297), JENDS(3),
     *             NGROUP
      DATA I1,I2,I3/1H1,1H2,1H3/
      DATA ISTAR,IG,IS,IC/1H*,1HG,1HS,1HC/
      DATA INTGR/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
      DATA IE,IN,ID/1HE,1HN,1HD/
      DATA IBL/1H /
      DATA IA/1HA/
      DATA JEQ /1H= /
C
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
C         THIS ROUTINE, TOGETHER WITH THE SUBROUTINES IT CALLS,
C         IS USED TO READ THE REVISED ENGLISH-LIKE FORM OF MULTIPLY
C         VULNERABLE INPUT DATA, AND CREATES THE ARRAYS NECESSARY FOR
C         COMPUTATIONS INVOLVING THE MULTIPLY VULNERABLE COMPONENTS.
C
C         FINALLY, SUBROUTINE ERROUT PRINTS A DIAGRAM OF EACH MULTIPLY
C         VULNERABLE GROUP FOR USER VERIFICATION AND DOCUMENTATION.
C
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
      LLU = LU
      LN = 0
      NEND=1
      ITERM=0
      NGROUP = 1
      NWORK = 0
      IBEGIN = 1
      IQUIT = 0
      ICARD(81)=IBL
   20 READ (5,25) (ICARD(I),I=1,80)
   25 FORMAT (80A1)
C***
C         FOR TRACE OPTION, PRINT A COPY OF THE CARD
C***
      IF(ITRACE.EQ.1) WRITE (6,26) ICARD
   26 FORMAT (1X,81A1)
C***
C         END OF MULTIPLE VULNERABILITY INPUT INDICATED BY BLANK CARD
C***
      ICTYPE = 0
      IF(ICARD(1).EQ.IBL) GO TO 125
C***
C         END OF A GROUP FAULT TREE INDICATED BY END CARD
C***
      IF(ICARD(1).EQ.IE.AND.ICARD(2).EQ.IN.AND.
     *                          ICARD(3).EQ.ID) GO TO 120
      IF(IBEGIN.NE.1) GO TO 30
```

```
C***
C         EXPECTING KILL CATEGORY IN COLUMN 15
C***
       KILLG = 0
       IF(ICARD(15).EQ.I1) KILLG = 1
       IF(ICARD(15).EQ.I2) KILLG = 2
       IF(ICARD(15).EQ.I3) KILLG = 3
       IBEGIN = 0
       GO TO 20
C***
C         HAVE ANOTHER CARD ... WHAT DOES IT SAY ?
C***
   30 NNA = 0
       IPRINT = -1
       IEU=0
       N1 = 1
       IF(ICARD(1).NE.ISTAR) GO TO 35
       N1 = 2
       IPRINT = 1
   35 IF(ICARD(N1).NE.IG) GO TO 40
C***
C         GROUP DEFINITION
C***
       ITYPE = 0
       GO TO 60
   40 IF(ICARD(N1).NE.IS) GO TO 45
C***
C         SYSTEM DEFINITION
C***
       ITYPE = 1
       GO TO 60
   45 IF(ICARD(N1).NE.IC) GO TO 50
C***
C         SUBSYSTEM DEFINITION
C***
       ITYPE = 2
       GO TO 60
C***
C         BAD DATA CARD ... PRINT ERROR AND SET IQUIT
C***
   50 WRITE (6,55) N1,ICARD
   55 FORMAT (29H    MV INPUT ERROR ... COLUMN ,I3,2X,80A1)
       IQUIT = 1
       GO TO 61
C***
C         DECODE ONE FIELD AT A TIME
C***
   60 N1 = N1+1
   61 CALL GETNAM(ICARD,N1,N2,NAME,NT,IEU)
       GO TO (65,70,85,110),NT
C***
C         NAME ... IS IT NEW ?
C***
   65 CALL MATCH(NAME,LLU,M)
       IF(M.EQ.0) GO TO 67
C***
```

```
C         OLD NAME ... STORE LO
C***
      IF (N1.LE.5) LORNP(M)=IPRINT+ITYPE
      NNA = NNA+1
      NLU(NNA) = M
      IF(IEU.EQ.1) GO TO 661
      IF(M.LE.LO) GO TO 66
C***
C         DELETE FROM UNDEFINED LIST
C***
      DO 655 I=1,NNORN
      IF(INORN(I).EQ.M) GO TO 656
  655 CONTINUE
      GO TO 66
  656 DO 657 J=I,NNORN
  657 INORN(J) = INORN(J+1)
      NNORN = NNORN-1
   66 CONTINUE
      N1 = N2+1
      GO TO 61
  661 IF (ICARD(N2+1).EQ.IML.AND.ICARD(N1-1).EQ.JEW) GO TO 662
      GO TO 66
  662 IPAN=-1
      GO TO 111
C***
C         NEW NAME ... MAKE ROOM
C***
   67 LLN = LLN+1
      NNA = NNA+1
      NLU(NNA) = LLN
      IF(IEU.EQ.0) GO TO 68
      NNORN = NNORN+1
C***
C         ADD TO UNDEFINED LIST
C***
      INORN(NNORN) = LLN
   68 CONTINUE
      WRITE(NAM(LLN),60) (NAME(I),I=1,6)
   60 FORMAT (6A1)
      IF (N1.LE.5) LORNP(LLN)=IPRINT+ITYPE
      IF (IEU.EQ.1) GO TO 691
      N1=N2+1
      IF(ITYPE.EQ.0) GO TO 61
C***
C         THIS CARD DEFINES AN UNREFERENCED NAME ... ERROR IN ORDERING
C***
      WRITE (6,69) ICARD
   69 FORMAT (36H DEFINITION OF UNREFERENCED NAME*  ,80A1)
      IQUIT = 1
      GO TO 61
  691 IF (ICARD(N2+1).EQ.IML.AND.ICARD(N1-1).EQ.JEW) GO TO 692
      N1=N2+1
      GO TO 61
  692 IPAN=-1
      GO TO 111
C***
```

```
C        CONNECTIVE ... WHAT KIND ?
C***
   70 IF(ICARD(N1).EQ.IA) GO TO 75
C***
C        NOT AND ... ASSUME OR
C***
      IF(ICTYPE.NE.1) GO TO 71
      WRITE (6,55) N1,ICARD
      IQUIT = 1
   71 ICTYPE = -1
      GO TO 80
C***
C        AND ...
C***
   75 IF(ICTYPE.NE.-1) GO TO 76
      WRITE (6,55) N1,ICARD
      IQUIT = 1
   76 ICTYPE = 1
   80 N1 = N2+1
      GO TO 61
C***
C        PARALLEL REDUNDANCY SPECIFICATION ... DECODE
C***
   85 IF(ICARD(N1).NE.ICARD(N1+2)) GO TO 90
C***
C        COMPLETE REDUNDANCY ... MUST GET NNN OUT OF NNN
C***
      NNN=NNA-1
      IPAR=NNN*10+NNN
      GO TO 110
C***
C        DECODE NUMBER REQUIRED
C***
   90 DO 95 N=1,NNA
      IF(ICARD(N1).EQ.INTGR(N)) GO TO 100
   95 CONTINUE
C***
C        ERROR ... PRINT AND SET IQUIT FLAG
C***
      WRITE (6,55) N1,ICARD
      N = NNA
      IQUIT = 1
C***
C        PACK IPAR FOR INCOMPLETE REDUNDANCY SPECIFICATION
C***
  100 IPAR=N*10+NNA-1
C***
C        FINISHED CARD ... UPDATE MUL ARRAY
C***
  110 IF(ICTYPE.LT.0) IPAR = -NNA+1
  111 IT=NLU(NNA)
      NLU(NNA)=NLU(1)
      NLU(1)=IT
  115 LM = LM+1
      JM=NLU(NNA)
      MUL(LM) = JM
```

```
      IF (JM.GT.LU) GO TO 119
      JLO=LOGRP(JM)
      IF (JLO.NE.0) GO TO 117
      LOGRP(JM)=NGROUP*101
      GO TO 119
  117 JLOX=MOD(JLO,100)
      JLOM=JLO/100
      IF (JLOX.LT.NGROUP) JLOX=NGROUP
      IF (JLOM.GT.NGROUP) JLOM=NGROUP
      LOGRP(JM)=JLOM*100+JLOX
  119 CONTINUE
      NNA = NNA-1
      IF(NNA.GT.0) GO TO 115
      LM = LM+1
      MUL(LM) = IPAR
      GO TO 20
C***
C         END OF GROUP ENTER KILL CATEGORY ... STORE LOCATION IN JENDS
C***
  120 LM = LM+1
      MUL(LM) = KILLG
      JENDS(NGROUP) = LM
      NGROUP = NGROUP+1
      INEGIN=1
 1209 READ (5,25)(ICARD(I),I=1,NC)
      IF (ITRACE.EQ.1) WRITE (6,26) ICARD
      IF (ICARD(1).EQ.IE.AND.ICARD(2).EQ.IN.AND.ICARD(3).EQ.ID) GO TO 20
      N1=1
      IERR=0
      ITEMS=0
 1210 CALL GETNAM (ICARD,N1,N2,NAME,II,IEQ)
      GO TO (1211,1212,1212,1214),II
 1211 IF (IEQ.EQ.1) GO TO 1212
C***
C      OUTPUT LABEL, ONLY USED FOR TRACE OUTPUT
C***
      WRITE(NNNAM,68) (NAME(I),I=1,N)
      N1=N2+1
      GO TO 1210
 1212 CONTINUE
      CALL MATCH (NAME,LLO,M)
      IF (M.NE.0) GO TO 1213
C***
C      ERROR .... UNDEFINED LU ....
C***
      WRITE (6,95) N1,ICARD
      N1=N2+1
      IERR=1
      GO TO 1210
C***
C      STORE LU IN MUL + OR - TERM
C***
 1213 IF (N1.EQ.3) M=-M
      ITEMS=ITEMS+1
      NEND=NEND+1
      N1=N2+1
```

```
      MUL(MEND)=M
      GO TO 1210
C***
C     FINISH GROUP
C***
 1214 MEND=MEND+1
      MUL(MEND)=ITEMS
      MEND=MEND+1
C     MUL(MEND)=NNAME
      MEND=MEND+1
      MUL(MEND)=KILLG
      GO TO 1209
C***
C         END OF MULTIPLE VULNERABILITY DATA
C***
  125 NGROUP = NGROUP-1
      MUL(1)=MEND
C***
C         ANY ERRORS ?
C***
      IF(NWORK.LE.0) GO TO 160
      DO 150 N=1,NWORK
      IF(IWORK(N).EQ.0) GO TO 150
      IQUIT = 1
C***
C         NEGATIVE IWORK MEANS UNDEFINED LO
C***
      M = IWORK(N)
      WRITE (6,130) NAM(M)
  130 FORMAT (25H   UNDEFINED M.V. ENTRY= ,A8)
  150 CONTINUE
  160 IF(IQUIT.EQ.0) GO TO 170
      WRITE (6,155)
  155 FORMAT (47H    FATAL M.V.ERROR(S) ... EXECUTION TERMINATED )
      IERR=1
  170 CONTINUE
C***
C         FOR TRACE OPTION ONLY .. MAY BE DELETED
C***
      IF(ITRACE.NE.1) GO TO 199
      WRITE (6,175)
  175 FORMAT (//,11H NAM ARRAY ,//,19H   LO     NAME(LO) /)
      DO 185  I=1,LLO
      WRITE (6,180) I,NAM(I)
  180 FORMAT(I5,5X,A8)
  185 CONTINUE
  195 FORMAT (I4,I10)
      WRITE (6,196)
  196 FORMAT (//12H JENDS ARRAY /)
      WRITE (6,195) (JENDS(I),I=1,NGROUP)
      WRITE (6,197)
  197 FORMAT (//12H LOGRP ARRAY /)
      WRITE (6,195) (I,LOGRP(I),I=1,LLO)
  198 FORMAT (//10H MUL ARRAY  /)
      IF(MEND.EQ.1) GO TO 199
      WRITE (6,198)
```

```
      K=NEUM(I)
  202 NULL = NUL(K)
C 202 WRITE (6,195) K,NUL(K)
      WRITE (6,195) K,NULL
      K=K-1
C     WRITE (6,203) K,NUL(K)
      WRITE (6,203) K, KNNAM
      K=K-1
      NULL =   NUL(K)
C     WRITE (6,195) K,NUL(K)
      WRITE (6,195) K,NULL
      NUM= NUL(K)
      K=K-1
      DO 201 I=1,NUM
      NULL =   NUL(K)
C     WRITE(IOP1,195) K,NUL(K)
      WRITE (6,195) K,NULL
      K=K-1
  201 CONTINUE
      IF(K.GT.1) GO TO 202
      NULL =   NUL(K)
C     WRITE (6,195) K,NUL(K)
      WRITE (6,195) K,NULL
  203 FORMAT (I8,2X,A8)
  199 J2=0
      IF (NGROUP.LE.0) GO TO 300
      DO 200 J=1,NGROUP
      J1 = J2+1
      J2 = JENDS(J)
  200 CALL ERONUL(J1,J2,ITRACE)
  300 IF (ITERR .LE. 0) RETURN
C***
C     ERROR DETECTED IN INPUT
C***
      WRITE (6,301)
  301 FORMAT (1X, 41H*** ERROR IN MULTIPLE, VULNERABILITY INPUT)
      STOP
      END
```

```
      SUBROUTINE OUTPUT(NAIMPT, ICORR, RANGE)
      COMMON /ALLPRS/ PRV(297, 3, 10)
      COMMON /DAMAGE/ ENERGY(10), PR(10,100), ENGYAU(100,10)
      COMMON /ING/ LD, NAM(297), FIL(2200), LOFRP(297), JENDS(3),
     *             NGROUP
      CHARACTER*8 NAM
      COMMON /PAGE/ LINE, LINLIM
      COMMON /STATUS/ ISTAT
C***
C     STATUS DEFINITION TABLE
C      = 0, END OF FLIGHT PATH
C      = 1, CANNOT ENGAGE
C      = 2, INSUFFICIENT TRACK TIME
C      = 3, SLEW RATE LIMIT EXCEEDED
C      = 4, FIRING THROUGH SMOKE
C      = 5, FIRING
C***
      COMMON /TAPE10/ UTAPE(16,6), ILAST(16), TIME, TX, TY, TZ,
     *                TXDOT, TYDOT, TZDOT, TXDDOT, TYDDOT, TZDDOT,
     *                TSPEED, TLEAD, TAZ, TDIVE, TROLL, TAA,
     *                IHI, ILO, NEXIST
      COMMON /TSTEPS/ TDELT, IPRINT, KOUNT
C***
C     OUTPUT SUBROUTINE, PRINT A LINE OF DATA EVERY IPRINT TIME STEPS
C***
      IF (IPRINT .EQ. 0) GO TO 200
      KOUNT = KOUNT + 1
      IF (KOUNT .LT. IPRINT) GO TO 200
      KOUNT = 0
      LINE = LINE + 1
      IF (LINE .LE. LINLIM) GO TO 100
      LINE = 1
      CALL HEADER
  100 GO TO (110, 120, 130, 140, 150), ISTAT
C***
C     CANNOT ENGAGE
C***
  110 WRITE (6,111) TIME, TX, TY, TZ, RANGE
  111 FORMAT (1X, F6.2, 3F8.0, F9.0, 12H  NOT ENGAGE )
      GO TO 200
C***
C     TRACKING
C***
  120 WRITE (6,121) TIME, TX, TY, TZ, RANGE
  121 FORMAT (1X, F6.2, 3F8.0, F9.0, 11H   TRACKING )
      GO TO 200
C***
C     SLEW RATE LIMIT EXCEEDED
C***
  130 WRITE (6,131) TIME, TX, TY, TZ, RANGE
  131 FORMAT (1X, F6.2, 3F8.0, F9.0, 12H   TRACK ERR )
      GO TO 200
C***
C     FIRING THROUGH SMOKE
C***
  140 N1 = NUL(1)
```

```
      DO 146 I = 1,NGROUP
      KILLG = MUL( JENOS(I) )
      IF (I .GE. 2) GO TO 142
      WRITE (6,141) TIME, TX, TY, TZ, RANGE, KILLG,
     *                                  (PKV(NI,KILLG,J), J=1,NAIMPT)
  141 FORMAT (1X, F6.2, 3F8.0, F9.0, 4X, 5HSURF, I4, 2X, 10F7.2)
      GO TO 145
  142 LINE = LINE + 1
      IF (LINE .LE. LINLIM) GO TO 144
      LINE = 1
      CALL HEADER
  144 WRITE (6,199) KILLG, (PKV(NI, KILLG, J), J=1,NAIMPT)
  145 NI = MUL( JENOS(I) + 1 )
  146 CONTINUE
      GO TO 200
C***
C     FIRING
C***
  150 NI = MUL(1)
      DO 156 I = 1,NGROUP
      KILLG = MUL( JENOS(I) )
      IF (I .GE. 2) GO TO 152
      WRITE (6,151) TIME, TX, TY, TZ, RANGE, KILLG,
     *                                  (PKV(NI,KILLG,J), J=1,NAIMPT)
  151 FORMAT (1X, F6.2, 3F8.0, F9.0, 4X, 6HENGAGE, I7, 2X, 10F7.2)
      GO TO 155
  152 LINE = LINE + 1
      IF (LINE .LE. LINLIM) GO TO 154
      LINE = 1
      CALL HEADER
  154 WRITE (6,199) KILLG, (PKV(NI), KILLG, J), J=1,NAIMPT)
  155 NI = MUL( JENOS(I) + 1 )
  156 CONTINUE
  199 FORMAT (50X, I3, 2X, 10F7.2)
C***
C     WRITE ENERGY ADDED ON BINARY TAPE FOR A POST PROCESSOR
C     AND ZERO THE ARRAY FOR THE NEXT TIME ITERATION
C***
  200 CONTINUE
      WRITE (11) TIME, ((ENGYAD(I,J), J=1,NAIMPT), I=1,NCOMP)
      DO 220 I = 1,NCOMP
      DO 210 J = 1,NAIMPT
      ENGYAD(I,J) = 0.0
  210 CONTINUE
  220 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE POINTS
C***
C     INTERPOLATE AND TRANSFORM THE NEW AIRCRAFT POSITION DATA AT
C     THE NEW TIME
C***
      COMMON /STATUS/ ISTAT
      COMMON /TAPEIO/ TAPE(16,7), T(16), IHI, ILO, NEXTST
      DIMENSION OTAPE(16,6), TLAST(16)
      EQUIVALENCE (OTAPE(1,1), TAPE(1,1)), (TLAST(1), TAPE(1,7))
C***
C     CHECK POINTERS SO THAT CURRENT TIME, T(1), IS WITHIN TAPE DATA
C***
      IF (T(1) .LE. TAPE(1,IHI)) GO TO 100
C***
C     ADVANCE POINTERS TO NEXT TAPE DATA PAIR
C***
      IHI = IHI + 1
      ILO = IHI - 1
      IF (ILO .LE. 5) GO TO 50
C***
C     NEED TO READ A NEW RECORD, FIRST SAVE DATA FOR THE LAST TIME
C     FROM THE OLD RECORD
C***
      DO 20 I = 1,16
      TLAST(I) = OTAPE(I,6)
   20 CONTINUE
      IHI = 1
      ILO = 7
      CALL READIO(OTAPE)
C***
C     CHECK FOR END OF FLIGHT PATH
C***
   50 IF (TAPE(11,IHI) .EQ. 0.0) GO TO 900
C***
C     STATUS EITHER REMAINS UNCHANGED, OR BECOMES CANNOT ENGAGE (=1)
C***
      ISTAT = MAX(ISTAT, NEXTST)
      IF (NEXTST .EQ. 1) ISTAT = 1
C***
C     DETERMINE CAN OR CANNOT ENGAGE STATUS AT NEXT FLIGHT PATH POINT
C***
      NEXTST = 2
      IF (TAPE(11,IHI) .LT. 0.0) NEXTST = 1
      TAPE(11,IHI) = ABS(TAPE(11,IHI))
C***
C     INTERPOLATE AIRCRAFT DATA AT TIME T(1)
C***
  100 FRAC = (T(1) - TAPE(1,ILO)) / (TAPE(1,IHI) - TAPE(1,ILO))
      DO 110 I = 2,16
      T(I) = TAPE(I,ILO) + FRAC * (TAPE(I,IHI)-TAPE(I,ILO))
  110 CONTINUE
      RETURN
C***
C     END OF FLIGHT PATH DETECTED
C***
  900 ISTAT = 0
      RETURN
      END
```

```
      SUBROUTINE PROPAT(RANGE, THETA)
      COMMON /LASER/  GUN(3), GUNTAN(3), PFLUX, FLUX(10), FLTIME(10),
     *                FLUXEM, FLUXGN
      COMMON /PROPAG/ ATTEN(10), RATTEN(10), SMATN, SMOKX(2), SMOKY(2),
     *                SMKFLD(2), RAIT, ISMTST
      COMMON /STATUS/ ISTAT
C***
C     STATUS DEFINITION TABLE
C      = 0, END OF FLIGHT PATH
C      = 1, CANNOT ENGAGE
C      = 2, INSUFFICIENT TRACK TIME
C      = 3, SLEW RATE LIMIT EXCEEDED
C      = 4, FIRING THROUGH SMOKE
C      = 5, FIRING
C***
      DIMENSION XY(2)
C***
C     COMPUTE LASER ATMOSPHERIC ATTENUATION FACTOR
C***
      ATN = COMPUT( ATTEN, RATTEN, RANGE, RAIT )
C***
C     TEST FOR ADDITIONAL SMOKE CORRIDOR INTERFERENCE
C***
      IF (ISMTST .EQ. 0) GO TO 80
      IF (ISMTST .EQ. 2) GO TO 50
      IF (SMKFLD(1) .LE. THETA  .AND. THETA .LE. SMKFLD(2) ) GO TO 90
   80 ISTAT = 5
      GO TO 100
   50 IF (THETA .LE. SMKFLD(1)  .OR. SMKFLD(2) .LE. THETA ) GO TO 90
      ISTAT = 5
      GO TO 100
C***
C     LASER BEAM INTERSECTS CORRIDOR, COMPARE RANGES TO DETERMINE IF
C     AIRCRAFT IS BETWEEN SMOKE AND WEAPON
C***
   90 DENOM = (SMOKY(2)-SMOKY(1)) * GUNTAN(1) - (SMOKX(2)-SMOKX(1)) *
     *           GUNTAN(2)
      IF ( ABS(DENOM) .GT. 0.000001 ) GO TO 92
      WRITE (6,91)
   91 FORMAT (41H0***ERROR, SMOKE CORRIDOR PARALLEL TO LOS)
      STOP
   92 S = (GUNTAN(2)*(SMOKX(1)-GUN(1)) + GUNTAN(1)*(GUN(2)-SMOKY(1))) /
     *      DENOM
      XY(1) = SMOKX(1) + S*(SMOKX(2)-SMOKX(1))
      XY(2) = SMOKY(1) + S*(SMOKY(2)-SMOKY(1))
      IF (RANGE .LE. DIS2(XY, GUN) ) GO TO 80
C***
C     SMOKE BETWEEN WEAPON AND AIRCRAFT, ADJUST TRANSMISSION FACTOR
C***
      ISTAT = 4
      ATN = ATN * SMATN
C***
C     DECREASE FLUX ON TARGET BY THE ATTENUATION FACTOR, ATN
C***
  100 FLUXGN = FLUXEM * ATN
      RETURN
      END
```

```
      SUBROUTINE READY
      DIMENSION TITLE(20)
      COMMON /LASER/  GUN(3), GUNTAR(3), NFLUX, FLUX(10), FLTIME(10),
     *                FLUXPP, FLUXPK
      COMMON /PAGE/ LINE, LINLIM
      COMMON /PROPAG/ ATTEN(10), RATTEN(10), SGAIN, SMORX(2), SMORY(2),
     *                SPREID(2), GAIN, ISWIST
      COMMON /STATUS/ ISTAT
      COMMON /TAPE10/ UTAPE(16,8), ILAST(16), TIME, TX, TY, TZ,
     *                TXDOT, TYDOT, TZDOT, TXDPNT, TYDOUT, TZDOUT,
     *                TSPEED, TLOAD, TAZ, TDIVE, THULL, TAA,
     *                IHI, ILO, NEWTST
      COMMON /TRACK/  TRATIO, SLEWAZ, SLEWEL, TRSTRT, YJITTR, ZJITTR
      COMMON /TRANSF/ XFP, YFP, XG, YG, ZG, PSI, CP, SP, GTOAC(3,3),
     *                ACTOG(3,3), ACTLF(3,3), LFTAC(3,3)
      COMMON /TSTEPS/ TDELT, IPRINT, KOUNT
      DATA DTOR, RADMIL/0.0174532925, 0.98174777E-03/
C***
C     PRINT MODEL HEADING; READ THE TIME STEP AND LINE PRINTER CONTROLS
C***
      WRITE (6,500)
      READ (5,420) TDELT, IPRINT, LINLIM
C***
C     READ AND PRINT AIRCRAFT PARAMETERS
C***
      READ (10) TITLE
      WRITE (6,510) TITLE
      READ (5,400) (GUN(I), I=1,3), XFP, YFP, XG, YG, ZG, PSI
      WRITE (6,520) XFP, YFP, 0.0, XG, YG, ZG, PSI
C***
C     CONVERT ROTATION ANGLE TO RADIANS AND COMPUTE SINE AND COSINE
C***
      PSI = PSI * DTOR
      CP = COS(PSI)
      SP = SIN(PSI)
C***
C     PRINT AND INITIALIZE LASER WEAPON PARAMETERS
C***
      WRITE (6,540) (GUN(I), I=1,3)
      READ (5,410) NFLUX, GAIN
      READ (5,400) (FLUX(I), I = 1,NFLUX)
      READ (5,400) (FLTIME(I), I = 1,NFLUX)
C***
C     CONVERT FROM WATTS/SQ.CM. TO KILOWATTS/SQ.CM.
C***
      DO 20 I = 1,NFLUX
      FLUX(I) = FLUX(I) * 0.001
   20 CONTINUE
      READ (5,400) YJITTR, ZJITTR
      WRITE (6,560) YJITTR, ZJITTR
C***
C     CONVERT FROM MILS TO RADIANS
C***
      YJITTR = YJITTR * RADMIL
      ZJITTR = ZJITTR * RADMIL
      WRITE (6,570) (FLUX(I), I = 1,NFLUX)
```

```
      WRITE (6,580) (FLTIME(I), I = 1,NFLUN)
C***
C     READ AND PRINT THE LASER TRACKING LIMITS
C***
      READ (5,800) SLEWAZ, SLEWEL, TRATIM
      WRITE (6,590) SLEWAZ, SLEWEL, TRATIM
      SLEWAZ = SLEWAZ * DTOR
      SLEWEL = SLEWEL * DTOR
C***
C     ATMOSPHERIC ATTENUATION FACTORS
C***
      READ (5,800) (ATTEN(I), I = 1,NATN)
      READ (5,800) (WATTEN(I), I = 1,NATN)
      WRITE (6,600) (ATTEN(I), I = 1,NATN)
      WRITE (6,610) (WATTEN(I), I = 1,NATN)
C***
C     SMOKE CORRIDOR, NO CORRIDOR MODELED IF COORDINATES ARE EQUAL
C***
      READ (5,800) (SMOKX(I), SMOKY(I), I = 1,2), SMATN
      IF (SMOKX(1) .EQ. SMOKX(2)  .AND. SMOKY(1) .EQ. SMOKY(2) )
     *    GO TO 46
      WRITE (6,620) (SMOKX(I), SMOKY(I), I = 1,2), SMATN
      DO 40 I = 1,2
      SMKFLD(I) = ATAN2( SMOKX(I) - GUN(1), SMOKY(I) - GUN(2) )
   40 CONTINUE
      IF (SMKFLD(1) .LE. SMKFLD(2)) GO TO 45
      TEMP = SMKFLD(1)
      SMKFLD(1) = SMKFLD(2)
      SMKFLD(2) = TEMP
   45 ISMIST = 1
      IF ( (SMKFLD(2) - SMKFLD(1)) .GT. 3.141592654) ISMIST = 2
      GO TO 50
   46 ISMIST = 0
C***
C     READ AIRCRAFT TARGET MODEL
C***
   50 CALL ACIN
C***
C     PREPARE THE I/O AND COMPUTATION TIME STEPS
C***
      WRITE (6,630) TDELT
      IF (IPRINT .EQ. 0) GO TO 60
      TEMP = FLOAT(IPRINT) * TDELT
      WRITE (6,640) TEMP
   60 KOUNT = IPRINT
C***
C     ASSIGN INITIAL AIRCRAFT POSITION DATA
C***
      CALL READIO(OTAPE)
      TIME = OTAPE(1,1)
      TX = OTAPE(2,1)
      TY = OTAPE(3,1)
      TZ = OTAPE(4,1)
      TXDOT = OTAPE(5,1)
      TYDOT = OTAPE(6,1)
      TZDOT = OTAPE(7,1)
```

```
      TXOUT = OTAPE(8,1)
      TYOUT = OTAPE(9,1)
      TZOUT = OTAPE(10,1)
      ISTAT = 2
      IF (OTAPE(11,1) .EQ. 0.0) ISTAT = 0
      IF (OTAPE(11,1) .LT. 0.0) ISTAT = 1
      OTAPE(11,1) = ABS( OTAPE(11,1) )
      TSPEED = OTAPE(11,1)
      NEXIST = 2
      IF (OTAPE(11,2) .LT. 0.0) NEXIST = 1
      ILOAD = OTAPE(12,1)
      TAZ = OTAPE(13,1)
      TDIVE = OTAPE(14,1)
      TROLL = OTAPE(15,1)
      TAA = OTAPE(16,1)
C***
C     SET UP PAGE HEADING CONTROL
C***
      LINE = 0
      IF (IPRINT .EQ. 0) GO TO 80
      LINLIM = LINLIM-5
      CALL HEADER
   80 RETURN
C***
C     INPUT FORMATS
C***
  400 FORMAT (10E8.0)
  410 FORMAT (10I8)
  420 FORMAT (E8.0, 2I8)
C***
C     OUTPUT FORMATS
C***
  500 FORMAT (39X,  27(2H* ) / 39X, 1H*, 51X, 1H* / 39X,
     *       51H* ASSESSMENT OF SURVIVABILITY AGAINST LASER THREATS,
     *       2H * / 39X, 1H*, 51X, 1H* / 39X, 27(2H* ) )
  510 FORMAT ( / 22H0AIRCRAFT FLIGHT PATHS /
     *       10X, 19HFLIGHT PATH FILE = , 20A4)
  520 FORMAT (10X, 28HTRANSFORMATION = (X,Y,Z) = (,
     *       2(F8.0,1H,), F8.0, 27H) IN FLIGHT PATH COORDINATE,
     *       19H SYSTEM IS EQUAL TO / 27X, 11H(X,Y,Z) = (,
     *       2(F8.0,1H,), F8.0, 25H) IN GENERAL COORDINATE,
     *       12H SYSTEM WITH, F7.1, 23H DEGREES ROTATION ANGLE)
  560 FORMAT (14H0LASER WEAPONS / 10X, 22HLOCATION = (X,Y,Z) = (,
     *       2(F8.0, 1H,), F8.0, 1H) )
  568 FORMAT (1H0, 13X, 55HAIM POINT STANDARD DEVIATION IN THE,
     *       41H ENCOUNTER PLANE DUE TO JITTER  SIGMA-Y =, F7.2,
     *       17H MILS   SIGMA-Z =, F7.2, 5H MILS)
  570 FORMAT (1H0, 13X, 33HFLUX EMISSION IN KILOWATTS/SQ.CM., 10F8.2)
  580 FORMAT (26X, 21HAT TIMES (IN SECONDS), 10(F7.1, 1X) )
  590 FORMAT (1H0, 13X, 32HTRACKING - MAXIMUM SLEW RATES IN,
     *       30H DEGREES PER SECOND  AZIMUTH =, F7.2,
     *       12H ELEVATION =, F7.2, / 25X, 23HMINIMUM TRACKING TIME =,
     *       F6.2, 8H SECONDS )
  600 FORMAT (12H0ATMOSPHERE / 12X, 19HATTENUATION FACTORS, 10F8.3)
  610 FORMAT (13X, 18HAT RANGES (METERS), 10F8.0)
  620 FORMAT (1H0, 11X, 35HSMOKE CORRIDOR - FROM COORDINATES (, F8.0,
```

```
     *         1H,, F8.0, 6H) TO (, F8.0, 1H,, F8.0, 17H) IN THE XY-PLANE/
     *         20X, 34HATTENUATION FACTOR THROUGH SMOKE =, F8.5)
 630 FORMAT (12H0TIME STEPS: / 12X, 24HBETWEEN COMPUTATIONS   =, F8.4,
     *         7H SECOND )
 640 FORMAT ( 12X, 24HBETWEEN PRINTOUT LINES =, F8.4, 7H SECOND )
     END
```

```
      SUBROUTINE READIN(UTAPE)
      DIMENSION UTAPE(16,6)
      COMMON /TRANSF/ XFP, YFP, XG, YG, ZG, PSI, CP, SP, GTOAC(3,3),
     *                ACTOG(3,3), ACTOE(3,3), ETOAC(3,3)
C***
C     READ AND TRANSFORM AND/OR TRANSLATE APPROPRIATE PARAMETERS
C     FROM THE NEXT RECORD OF TAPE 10, FROM THE FLIGHT PATH
C     COORDINATE SYSTEM TO THE GENERAL COORDINATE SYSTEM
C***
      READ (10,END=99) UTAPE
C***
C     PERFORM NECESSARY TRANSLATION FOR ALL 6 AREA TIME DATA STEPS
C***
      DO 30 I = 1,6
      XIN = UTAPE(2,I)
      YIN = UTAPE(3,I)
      UTAPE(2,I) = XG + (XIN-XFP)*CP + (YIN-YFP)*SP
      UTAPE(3,I) = YG + (YIN-YFP)*CP - (XIN-XFP)*SP
      UTAPE(4,I) = ZG + UTAPE(4,I)
      XD = UTAPE(5,I)
      YD = UTAPE(6,I)
      UTAPE(5,I) = XD*CP + YD*SP
      UTAPE(6,I) = YD*CP - XD*SP
      XD = UTAPE(8,I)
      YD = UTAPE(9,I)
      UTAPE(8,I) = XD*CP + YD*SP
      UTAPE(9,I) = YD*CP - XD*SP
      UTAPE(13,I) = UTAPE(13,I) - PSI
   30 CONTINUE
      RETURN
C***
C     END OF FLIGHT PATH FILE, ASSUME END OF FLIGHT PATH
C***
   99 UTAPE(11,1) = 0.0
      RETURN
      END
```

```
      SUBROUTINE SUMMRY(NAIMPT, NSHOTS)
      COMMON /AIRCFT/ NCOMP, COMP(3,100), AP(100,26), WIDTH(100,26),
     *                 ENGVOR(100,10)
      COMMON /ALLPKS/ PKV(297, 5, 10)
      COMMON /TREE/ LW, NAM(297), MUL(2200), LOGNP(297), JENDS(5),
     *               NGROUP
      CHARACTER*8 NAM
C***
C     PRINT THE SUMMARY FOR THE LASER WEAPON ASSESSMENT
C***
      WRITE (6,100) NSHOTS
      WRITE (6,110) (I, I=1,NAIMPT)
      WRITE (6,120)
C***
C     PRINT DAMAGE SUMMARY BY GROUP/KILL CATEGORY
C***
      WRITE (6,200)
      KPOINT = MUL(1)
      DO 20 IGRP = 1,NGROUP
      KILLG = MUL( JENDS(IGRP) )
      WRITE (6,210) KILLG, NAM(KPOINT), (PKV(KPOINT,KILLG,J),
     *                                          J = 1,NAIMPT)
      KPOINT = MUL( JENDS(IGRP) + 1 )
   20 CONTINUE
C***
C     PRINT SUBGROUP PK'S
C     ONLY SUBGROUP NAMES WITH AN ASTERISK ON THE FAULT TREE
C     STRUCTURE CARDS ARE INCLUDED HERE
C***
      WRITE (6,300)
      MLAST = 1
      DO 38 IGRP = 1,NGROUP
C***
C     TRAVERSE THE FAULT TREE STRUCTURE IN ARRAY MUL, AND PICK
C     OUT THE DESIRED SUBGROUP NAMES
C***
      MPOINT = JENDS(IGRP)
      KILLG  = MUL(MPOINT)
      KPRINT = 0
   30 MPOINT = MPOINT - 1
      IF (MUL(MPOINT) .LT. 0) GO TO 32
C***
C     PARALLEL SUBGROUP -- EXTRACT RIGHT DIGIT FROM MUL
C***
      NELEMT = MOD( MUL(MPOINT), 10)
      GO TO 34
C***
C     SERIES SUBGROUP
C***
   32 NELEMT = -MUL(MPOINT)
C***
C     ADJUST MPOINT TO POINT TO SUBGROUP POINTER IN MUL
C***
   34 MPOINT = MPOINT - NELEMT - 1
      ISUM = MUL(MPOINT)
C***
```

```
C     NEGATIVE LOGRP INDICATES SUBGROUP NAME IS TO BE OMITTED IN SUMMARY
C***
      IF (LOGRP(ISUB) .LE. 0) GO TO 36
      IF (KPRINT .EQ. 1) GO TO 35
      WRITE (6,310) KILLG, NAM(ISUB), (PKV(ISUB,KILLG,J), J=1,NAIMPT)
      KPRINT = 1
      GO TO 36
   35 WRITE (6,320) NAM(ISUB), (PKV(ISUB,KILLG,J), J=1,NAIMPT)
   36 IF (MPOINT .GT. MLAST) GO TO 30
C***
C     END OF SUBGROUPS FOR THE KILL CATEGORY GROUP
C***
      MLAST = JENDS(IGRP) + 1
   38 CONTINUE
C***
C     PRINT COMPONENT PK'S
C***
      WRITE (6,400)
      DO 40 ICOMP = 1,NCOMP
      WRITE (6,410) ICOMP, NAM(ICOMP), (PKV(ICOMP,1,J), J=1,NAIMPT)
   40 CONTINUE
      RETURN
C***
C     FORMATS
C***
  100 FORMAT (1H0, 5X, 26H*** END OF FLIGHT PATH *** /
     *          10X, 19HTOTAL LASER SHOTS = , I5 )
  110 FORMAT (34H1DAMAGE SUMMARY AT EACH AIM POINTS, 85X,
     *          10HAIM POINTS / 48X, 10I7 )
  120 FORMAT (48X, 1HI, 88(1H-) )
  200 FORMAT (10X, 19HTOTAL AIRCRAFT PK'S, 19X, 1HI )
  210 FORMAT (13X, 13HKILL CATEGORY, I3, 3X, A8, 6H GROUP, 3H  I,
     *          F5.2, 9F7.2)
  300 FORMAT (48X, 1HI /
     *          19X, 13HSUBGROUP PK'S, 16X, 1HI)
  310 FORMAT (22X, 13HKILL CATEGORY, I2, 1X, A8, 3H  I, F5.2, 9F7.2)
  320 FORMAT (38X, A8, 3H  I, F5.2, 9F7.2)
  400 FORMAT (48X, 1HI /
     *          28X, 14HCOMPONENT PK'S, 6X, 1HI)
  410 FORMAT (30X, I3, 5X, A8, 3H  I, F5.2, 9F7.2)
      END
```

```
      SUBROUTINE TRACK1
      COMMON /LASER/  GUN(3), GUNTAR(3), GFLUX, FLUX(10), FLTIME(10),
     *                FLUXEM, FLUXEP
      COMMON /STATUS/ ISTAT
      COMMON /TAPE10/ CTAPE(16,6), ILAST(16), TIME, TX, TY, TZ,
     *                TXDOT, TYDOT, TZDOT, TXDDOT, TYDDOT, TZDDOT,
     *                TSPEED, TLEVEL, TAZ, TDIVE, TROLL, TAA,
     *                IHI, ILO, IEXIST
      COMMON /TRACK/  TRKTIM, SLEWAZ, SLEWEL, TRSTRT, YJITTR, ZJITTR
C***
C     CHECK MINIMUM TRACK TIME BEFORE FIRING
C***
      IF ( (TIME-TRSTRT) .GE. TRKTIM ) GO TO 10
      ISTAT = 2
      RETURN
C***
C     CHECK AZIMUTH AND ELEVATION TRACKING RATES
C***
   10 G2 = GUNTAR(1)**2 + GUNTAR(2)**2
      R2 = G2 + GUNTAR(3)**2
      AZDOT = ( GUNTAR(1)*TYDOT - TXDOT*GUNTAR(2) ) / G2
      IF (AZDOT .GT. SLEWAZ) GO TO 20
      ELDOT = (TZDOT - (GUNTAR(3) * ( GUNTAR(1)*TXDOT + GUNTAR(2)*TYDOT
     *         + GUNTAR(3)*TZDOT ) / R2)) / SQRT(G2)
      IF (ELDOT .LE. SLEWEL) GO TO 30
C***
C     SLEW RATE EXCEEDED, ASSIGN STATUS 3
C***
   20 ISTAT = 3
      RETURN
C***
C     ABLE TO TRACK, SET STATUS EQUAL TO 4 AND RETURN
C***
   30 ISTAT = 4
      RETURN
      END
```

```
      SUBROUTINE UPDATE( TKSTRT, TIME )
C***
C     UPDATA STATISTICS DEPENDING ON CURRENT AND OLD STATII
C***
      COMMON /RECALL/ IOLDST, NSHOTS
      COMMON /STATUS/ ISTAT
C***
C     STATUS DEFINITION TABLE
C      = 0, END OF FLIGHT PATH
C      = 1, CANNOT ENGAGE
C      = 2, INSUFFICIENT TRACK TIME
C      = 3, SLEW RATE LIMIT EXCEEDED
C      = 4, FIRING THROUGH SMOKE
C      = 5, FIRING
C***
C     RESET BEGINNING TRACK TIME IF STATUS = 1 OR 3
C***
      IF (ISTAT.EQ.1 .OR. ISTAT.EQ.3) TKSTRT = TIME
      IF (ISTAT .EQ. IOLDST) RETURN
C***
C     TEST FOR NEW LASER SHOT
C***
      IF (IOLDST .LE. 3  .AND.  ISTAT .GE. 4) NSHOTS = NSHOTS+1
      IOLDST = ISTAT
      RETURN
      END
```

```
      SUBROUTINE VPSV(ANS, A, S, B)
      DIMENSION ANS(3), A(3), B(3)
C***
C     THIS SUBROUTINE IS USED TO COMPUTE THE RESULTANT VECTOR, ANS,
C     BY ADDING VECTOR A TO THE PRODUCT OF SCALER S AND VECTOR B
C***
      DO 10 I = 1,3
      ANS(I) = A(I) + S*B(I)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VXMAT(ANS, V, T)
      DIMENSION ANS(3), V(3), T(3,3)
C***
C     TRANSFORM VECTOR, V, TO ANOTHER COORDINATE SYSTEM DEFINED BY
C     TRANSFORMATION MATRIX, T.
C***
      DO 20 I = 1,3
      ANS(I) = 0.0
      DO 10 J = 1,3
      ANS(I) = ANS(I) + V(J)*T(J,I)
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
```

```
      LOGICAL FUNCTION CANHIT(AZ, EL, IAIM)
      COMMON /AIMPTS/ NAIMPT, AIM(3,10), SIGMA(10,2),
     *                AZLIM(10,2), ELLIM(10,2)
C***
C     RETURN THE VALUE TRUE IF ANGLES AZ AND EL ARE INSIDE THE
C     ENVELOPE ANGULAR LIMITS FOR THE AIM POINT
C***
      IF (AZLIM(IAIM,1) .LE. AZLIM(IAIM,2)) GO TO 10
      CANHIT = ( (AZLIM(IAIM,1) .LE. AZ  .OR. AZLIM(IAIM,2) .GE. AZ)
     *       .AND. ELLIM(IAIM,1) .LE. EL .AND. ELLIM(IAIM,2) .GE. EL)
      GO TO 20
   10 CANHIT = ( AZLIM(IAIM,1) .LE. AZ .AND. AZLIM(IAIM,2) .GE. AZ
     *       .AND. ELLIM(IAIM,1) .LE. EL .AND. ELLIM(IAIM,2) .GE. EL)
   20 CONTINUE
      RETURN
      END
```

```
      FUNCTION COMPUT(F, X, ANG, NUM)
      DIMENSION F(10), X(10)
C***
C     INTERPOLATE FROM FUNCTION F AT VALUE ANG IN DOMAIN X
C***
      IF (ANG .GT. X(1)) GO TO 5
      COMPUT = F(1)
      RETURN
    5 I1 = 1
      DO 10 I2 = 2,NUM
      IF (ANG .LE. X(I2) ) GO TO 20
      I1 = I2
   10 CONTINUE
      COMPUT = F(NUM)
      RETURN
   20 COMPUT = F(I1) + (ANG-X(I1))/(X(I2)-X(I1)) * (F(I2)-F(I1))
      RETURN
      END
```

```
      FUNCTION DFN(X)
C***
C     FROM HASTINGS APPROXIMATIONS FOR DIGITAL COMPUTERS, PAGE 187
C     THE COEFFICIENTS ARE CONVERTED BY DIVISION BY SQRT(2) **I
C***
      F = 0.0
      AX = ABS(X)
      IF (AX .GE. 5.0) GO TO 5
      F = (((((.5383E-5*AX + .488906E-4)*AX + .380036E-4)*AX
     *     + .0032776263)*AX + .0211410061)*AX + .0498673469)*AX
     *     + 1.0
      F = 0.5 / ((F**4)**2)
    5 IF (X .GE. 0.0) F = 1.0 - F
      DFN = F
      RETURN
      END
```

```
      FUNCTION DIS2(V1, V2)
      DIMENSION V1(2), V2(2)
C***
C     COMPUTE THE DISTANCE BETWEEN TWO POINTS, V1 AND V2, IN THE
C     SAME TWO DIMENSIONAL PLANE
C***
      DIS2 = 0.0
      DO 10 I = 1,2
      DIS2 = DIS2 + ( V1(I)-V2(I) )**2
   10 CONTINUE
      DIS2 = SQRT( DIS2 )
      RETURN
      END
```

```
      FUNCTION DISS(V1, V2)
      DIMENSION V1(3), V2(3)
C***
C     COMPUTE THE DISTANCE BETWEEN TWO POINTS, V1 AND V2, IN THE
C     SAME THREE DIMENSIONAL COORDINATE SYSTEM
C***
      DISS = 0.0
      DO 10 I = 1,3
      DISS = DISS + ( V1(I)-V2(I) )**2
   10 CONTINUE
      DISS = SQRT( DISS )
      RETURN
      END
```

```
      FUNCTION PHIT( ICOMP, IAIP, SIGY, SIGZ, AIMRGE)
      DIMENSION ATOC(3), COMPE(3)
C***
C     THIS FUNCTION COMPUTES THE PROBABILITY OF HITTING A COMPONENT
C     OFFSET FROM THE AIM POINT WITH GAUSSIAN AIMING DEVIATIONS
C***
      COMMON /AIMPT3/ NAIMPT, AIP(3,10), SIGMA(10,2),
     *                AZLIM(10,2), ELLIM(10,2)
      COMMON /AIRCFT/ NCOMP, COMP(3,100), AP(100,26), WIDTH(100,26),
     *                ENGYOL(100,10)
      COMMON /LASER/  GUN(3), GUNTAR(3), GFLUX, FLUX(10), FLTIME(10),
     *                FLUXFF, FLUXCN
      COMMON /TRANSF/ XPP, YPP, XG, YG, ZG, PSI, CP, SP, GTOAC(3,3),
     *                ACTOG(3,3), ACTOE(3,3), ETOAC(3,3)
C***
C     COMPUTE LOOK-ANGLES TO THE COMPONENT
C***
      CALL LOKANG(COMPAZ, COMPEL, GUNTAR, COMP(1,ICOMP))
C***
C     INTERPOLATE THE COMPONENT PRESENTED AREA AND WIDTH AT THIS ASPECT
C     AND RETURN IF ZERO PRESENTED AREA
C***
      CALL INT2A(PAREA, WIDE, COMPAZ, COMPEL, ICOMP)
      PHIT = 0.0
      IF (PAREA .LT. 0.1E-06) RETURN
C***
C     TRANSLATE AND ROTATE VECTOR FROM-AIM-POINT-TO-COMPONENT INTO THE
C     ENCOUNTER COORDINATE SYSTEM
C***
      CALL VPSV(ATOC, COMP(1,ICOMP), -1.0, AIP(1,IAIP))
      CALL VXMAT(COMPE, ATOC, ACTOE)
C***
C     COMPUTE RANGE FROM LASER TO COMPONENT, AND
C     CONVERT COMPE(2) AND COMPE(3) TO RADIANS FROM THE LASER LOCATION,
C     (AIMRGE,0,0) IN THIS COORDINATE SCOMPE(2)STEM
C***
      COMPRG = SQRT( (COMPE(1)-AIMRGE)**2 + COMPE(2)**2 + COMPE(3)**2 )
      COMPE(2) = ATAN2(COMPE(2), (AIMRGE-COMPE(1)))
      COMPE(3) = ATAN2(COMPE(3), (AIMRGE-COMPE(1)))
C***
C     COMPUTE INTEGRATION LIMITS OF RECTANGLE AROUND COMPE(2) AND
C***
      YDELT = ATAN2( (WIDE/2.0), COMPRG )
      ZDELT = ATAN2( (PAREA/(2.0*WIDE)), COMPRG )
C***
C     COMPUTE PHIT USING HASTINGS APPROXIMATION
C***
      PHITY = DFN( (COMPE(2)+YDELT)/SIGY ) -
     *        DFN( (COMPE(2)-YDELT)/SIGY )
      PHITZ = DFN( (COMPE(3)+ZDELT)/SIGZ ) -
     *        DFN( (COMPE(3)-ZDELT)/SIGZ )
      PHIT = PHITY * PHITZ
      RETURN
      END
```

```
      FUNCTION VECMAG(V)
      DIMENSION V(3)
C***
C     COMPUTE THE MAGNITUDE OF VECTOR V
C***
      SUM = 0.0
      DO 10 I = 1,3
      SUM = SUM + V(I)**2
   10 CONTINUE
      VECMAG = SQRT(SUM)
      RETURN
      END
```

DISTRIBUTION LIST

Aeronautical Systems Division (AFSC)
Wright-Patterson AFB, OH 45433
Attn: ASD/XRM (F. Campanile)
Attn: ASD/ENFTV
Attn: ASD/XRS (L. R. Roesner)
Attn: ASD/ENFTV (D. J. Wallick) (2 copies)

Air Force Wright Aeronautical Laboratories
Wright-Patterson AFB, OH 45433
Attn: AFWAL/FIESD
Attn: AFWAL/FIESD (CDIC) (2 copies)
Attn: AFWAL/FIBEB (D. B. Paul)
Attn: AFWAL/FIER (R. H. Walker)
Attn: AFWAL/POSH (T. A. Hogan)
Attn: AFWAL/XRXI (J. T. Van Kuren)

Applied Technology Laboratory
Army Research & Technology Laboratories (AVRADCOM)
Fort Eustis, VA 23604
Attn: DAVDL-ATL-ASW (A. Royal)
Attn: DAVDL-ATL-ASV (E. V. Merritt)
Attn: DAVDL-ATL-ASV (C. M. Pedriani)

Armament Division
Eglin AFB, FL 32542
Attn: AD/XRSP (Dr. S. W. Turner)
Attn: AD/ENS (C. Varnado)

Army Aviation Research & Development Command
4300 Goodfellow Blvd.
St. Louis, MO 63120
Attn: DRDAV-EGL (H. Lee)
Attn: DRCPM-ASE-TM (COL E. Robinson)

Army Electronics Research & Development Command (ERADCOM)
Electronic Warfare Laboratory
Fort Monmouth, NJ 07703
Attn: DELEW-P (J. Schwartz, EW Protection Division)

Army Foreign Science and Technology Center
220 Seventh St., NE
Charlottesville, VA 22901
Attn: DRXST-CA2 (Crosby)

Army Materials and Mechanics Research Center
Watertown, MA 02172
Attn: DRXMR-SE (J. Adachi)

Army Materiel Systems Analysis Activity
Aberdeen Proving Ground, MD 21005
Attn: DRXSY-AAS (W. B. Paris)

Ballistic Research Laboratory
Aberdeen Proving Ground, MD 21005
Attn: DRDAR-BLV (R. Miller)

David W. Taylor Naval Ship R&D Center
Carderock Laboratory
Bethesda, MD 20084
Attn: Code 522.1, Library

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314
Attn: DTIC-TCA (2 copies)

Federal Aviation Administration Technical Center
Atlantic City, NJ 08405
Attn: ACT-330 (L. J. Garodz)

Foreign Technology Division (AFSC)
Wright-Patterson AFB, OH 45433
Attn: FTD/NIIS (5 copies)

Joint Cruise Missiles Project Office
Washington, DC 20360
Attn: JCM-OOY (V. Roske)

Joint Strategic Target Planning Staff
Offutt AFB, NB 68113
Attn: SAC/NRI (STINFO Library)

Naval Air Development Center
Warminster, PA 18974
Attn: Code 60B (E. J. Boscola)
Attn: Code 60B1 (Dr. G. T. Chisum)
Attn: Code 2012 (M. C. Mitchell)

Naval Air Systems Command
Washington, DC 20361
Attn: AIR-5164J (2 copies)
Attn: AIR-312B (E. A. Lichtman)
Attn: AIR-31A (R. Schmidt)

Naval Sea Systems Command
Washington, DC 20362
Attn: SEA 05R23 (T. Johnson)

Naval Weapons Center
China Lake, CA 93555
Attn: Code 3383 (C. Driussi)
Attn: Code 3381 (C. Padgett) (2 copies)

Naval Weapons Evaluation Facility
Kirtland AFB, NM 87117
Attn: Code 70, Nuclear Survivability

Pacific Missile Test Center
Point Mugu, CA 93042
Attn: Code 6862 (Naval Air Station, Technical Library) (6 copies)

Acurex Aerotherm
Aerospace Systems Division
485 Clyde Ave.
Mountain View, CA 94042
Attn: J. Dodson, M/S 8-8800

Armament Systems, Inc.
838 N. Eglin Parkway, Suite 421
Ft. Walton Beach, FL 32549
Attn: G. A. Brown

Battelle Columbus Laboratories
505 King Ave.
Columbus, OH 43201
Attn: Robert Broderson

The BDM Corp.
1801 Randolph Rd, SE.
Albuquerque International
Albuquerque, NM 87106
Attn: A. J. Holten
Attn: Tech. Resourse Center

The BDM Corp.
2650 Yale S. E.
Albuquerque, NM 87106
Attn: R. Mollo, M/S 2A2, Bldg 2

The BDM Corp.
First National Plaza
Suite 1820
Dayton, OH 45402
Attn: G. J. Valentino

Bell Helicopter - Textron
P. O. Box 482
Fort Worth, TX 76101
Attn: D. Lairdon MD 11

Boeing Vertol Company
A Division of the Boeing Company
Boeing Center
P. O. Box 16858
Philadelphia, PA 19142
Attn: N. Caravasos, M/S P32-18

The Boeing Company
P. O. Box 3707
Seattle, WA 98124
Attn: K. F. Brettmann, M/S 4E-64 (2 copies)
Attn: J. H. Howard, M/S 4E-64

Boeing Military Airplane Company
A Division of the Boeing Company
3801 S. Oliver
Wichita, KS 67210
Attn: G. Heinrich, M/S K16-14
Attn: Classified Control Station 47 for A. L. Weller, M/S K62-55
Attn: R. Gerhardt, K75-25
Attn: J. Wolf, M/S K75-25

Booz, Allen & Hamilton, Inc.
4330 East-West Highway
Bethesda, MD 20814
Attn: W. Herman

Calspan Corp.
P. O. Box 400
Buffalo, NY 14221
Attn: Library (V. M. Young)

COMARCO, inc.
1417 No. Norma St.
Ridgecrest, CA 93555
Attn: G. Russell

Denver Research Institute
University of Denver, University Park
Denver, CO 80208
Attn: J. Thompson

Douglas A/C Co.
Lakewood Blvd.
Long Beach, CA 90846
Attn: Myron Potter (C1-E00-595)

Emerson Electric Co.
8100 Florissant
St. Louis, MO 63136
Attn: Library, M/S 4372

E-Systems Inc.
P. O. Box 1056
Greenville, TX 75401
Attn. L. Phelps, CBN 38

Fairchild Republic Company
Conklin Street
Farmingdale, NY 11735
Attn: Tech. Info. Center (G. A. Mauter)

Falcon Research and Development Co.
One American Drive
Buffalo, NY 14225
Attn: Secruity Officer

Ford Aerospace and Communications Corp.
Ford Road, P. O. Box A
Newport Beach, CA 92663
Attn: Tech. Info. Center

General Electric Co.
Aricraft Engine Group
Mail Drop H-6
Cincinnati, OH 45215
Attn: G. E. Varney

Goodyear Aerospace Corp.
Arizona Division - Library D/154
P. O. Box 85
Litchfield Park, AZ 85340
Attn: G. Potter, D/948

Grumman Aerospace Corp.
South Oyster Bay Rd.
Bethpage, NY 11714
Attn: J. P. Archey, Jr., Dept. 662, M/S C42-05
Attn: J. Hartung, Dept. 661, M/S C27-05
Attn: Technical Information Center, Plant 35 LO1-35 (T. Wilkins)

Hughes Helicopters, Inc.
Division of Summa Corp.
Centinela & Teal Sts.
Culver City, CA 90230
Attn: Library, 6/A69 (D. K. Goss)

Kentron, Inc.
696 Fairmount Ave.
Towson, MD 21204
Attn: G. J. Mitchell

Lockheed-California Co.
A Division of Lockheed Aircraft Corp.
2555 Hollywood Way
P. O. Box 551
Burbank, CA 91520
Attn: D. Tuttle, D/77-76, Bldg. 90-1, Plant B-1

Los Alamos Technical Associates, Inc.
P. O. Box 410
Los Alamos, NM 87544
Attn: Library

McDonnell Douglas Astronautics Co.
5301 Bolsa Ave.
Huntington Beach, CA 92647
Attn: B. L. Cooper, A3-218, M/S 13-3

McDonnell Douglas Corp.
Douglas Aircraft Company
3855 Lakewood Blvd.
Long Beach, CA 90846
Attn: M. Potter, C1-E00-595

McDonnell Douglas Corp.
McDonnell Aircraft Company
P. O. Box 516
St. Louis, Mo 63166
Attn: R. D. Detrich, Dept. 022
Attn: Dr. T. Ender, 313/33/6/605
Attn: J. Robinson, Dept E 457, Bldg 101/MEZ/Post S 53

D. W. Mowrer
221 W. Stanton Road
Quarryville, PA 17566

Northrop Corp.
Aircraft Division
One Northrop Avenue
Hawthorne, CA 90250
Attn: K. Christensen, N3895/94
Attn: J. E. Reynolds, 3360/82

Northrop Corp.
Ventura Division
1515 Rancho Conejo Blvd.
P. O. Box 2500
Newbury Park, CA 91320
Attn: Tech. Info. Center (M. Raine)

The Perkin-Elmer Corporation
100 Wooster Hts. Rd.
Danbury CT 06810
Attn: J. H. Beardsley, M/S 967

Rockwell International Corporation
6633 Canoga Avenue
Canoga Park, CA 91304
Attn: M. D. Black, Jr.

Rockwell International
North American Aircraft Operations
4300 E. Fifth Avenue
P. O. Box 1259
Columbus, OH 43216
Attn: Technical Information Center (N. Brake)
Attn. R. H. Marshall, 871 021 B/6

Rockwell International Corporation
Missile Systems Division
1800 Satellite
Duluth, GA 30136
Attn: R. A. Jones (D362/628 NC-21)

Rockwell International Corporation
P. O. Box 92098
Los Angeles, CA 90009
Attn: W. J. Darling, D116, 011-GB02
Attn: S. C. Mellin, Dept. 115, GD-12
Attn: R. L. Moonan, GC-02
Attn: Tech. Info. Center, GA-06
Attn: A. S. Musicman, GD10

Science Applications, Inc.
505 Marquette NW
Albuquerque, NM 87102
Attn: Library (ALL: 01-18)

Survice Engineering Co.
P. O. Box 693
Bel Air, MD 21014
Attn: Security Officer

Sikorsky Aircraft Division
United Technologies Corporation
North Main Street
Stratford, CT 06002
Attn: D. P. Bartz, Chief Survivability

Southwest Research Institute
6220 Culebra Road
P. O. Drawer 28510
San Antonio, TX 78284
Attn: P. H. Zabel, Div. 02

Sverdrup Technology, Inc.
AEDC Group
Arnold AFS, TN 37389
Attn: Library, M/S 100
Attn: J. A. Reed, M/S 900 (ETF-ED2)

Teledyne Ryan Aeronautical
P. O. Box 80311
San Diego, CA 92138
Attn: Technical Information Services (W. E. Ebner)

Texas Instruments, Inc.
P. O. Box 226015, M/S 3177
Dallas, TX 75266
Attn: Dr. G. L. Smith M/S 3177

Uniroyal, Inc.
312 N. Hill Street
Mishawaka, IN 46544
Attn: R. L. Anderson
Attn: J. Kulesia

United Technologies Corporation
United Technologies Research Center
Silver Lane, Post 10
East Hartford, CT 06108
Attn: UTC Library (M. E. Donnelly)

Vought Corporation
P. O. Box 225907
Dallas, TX 75265
Attn: H. C. Wispell M/S 220-48