DRSAR/SA/R-11

METHODOLOGY FOR COMPUTER-GENERATION OF LINES OF CONSTANT BURST-KILL PROBABILITIES (FOOTPRINTS) FOR GUN AIR DEFENSE SYSTEMS (ISO-PK)

STUART OLSON

ROBERT WILLIAMS

MARCH 1976

US ARMY ARMAMENT COMMAND

SYSTEMS ANALYSIS DIRECTORATE

ROCK ISLAND, ILLINOIS 61201

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the Army Materiel Development and Readiness Command unless so stated.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DRSAR/SA/R-11	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) METHODOLOGY FOR COMPUTER-GENERATION OF LINES OF CONSTANT BURST-KILL PROBABILITIES (FOOTPRINTS) FOR GUN AIR DEFENSE SYSTEMS (ISO-PK)		5. TYPE OF REPORT & PERIOD COVERED Report - Final 6. PERFORMING ORG. REPORT NUMBER DRSAR/SA/R-11
7. AUTHOR(s) Stuart Olson Robert Williams		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Armament Command Systems Analysis Directorate (ATTN: DRSAR-SA) Rock Island, IL 61201		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS OMA
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament Command Systems Analysis Directorate (ATTN: DRSAR-SA) Rock Island, IL 61201		12. REPORT DATE March 1976 13. NUMBER OF PAGES 99
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Air defense
Gun systems
Kill probability
Sensitivity analysis
Computer program
FORTRAN
Error Budget

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

An important tool for analyzing the relative effectiveness of gun air defense systems is a plot of lines of constant (ISO) probability of kill (PK) in the volume of space surrounding the planned deployment of the system; hence, the name ISO-PK. These plots are usually presented as two-dimensional sections taken at selected places in the volume. The lines of constant kill probability sometimes form eliptical traces and, perhaps, for this reason, they are commonly referred to as "footprints". This report presents the computer program and explains the algorithm which calculates the burst-kill probability and automati-

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) cally plots the footprints on a computer line printer. Also included in this report is an analysis of weapon pointing errors which includes the effects of lead angle generation. This model, which may be termed an error budget model, was originally developed for the Air Defense Evaluation Board (ADEB). Instructions for use of the program and a sample problem are presented.

CONTENTS

LIST OF FIGURES

INTRODUCTION

An important tool for analyzing the relative effectiveness of gun air defense systems is a plot of lines of constant (ISO) probability of kill (PK) in the volume of space surrounding the planned deployment of the system; hence, the name ISO-PK. These plots are usually presented as two-dimensional sections taken at selected places in the volume. The lines of constant kill probability sometimes form eliptical traces and, perhaps, for this reason, they are commonly referred to as "footprints". This report presents the computer program and explains the algorithm which calculates the burst-kill probability and automatically plots the footprints on a computer line printer. Also included in this report is an analysis of weapon pointing errors which includes the effects of lead angle generation. This model, which may be termed an error budget model, was originally developed for the Air Defense Evaluation Board (ADEB)[1].

The expression for burst-kill probability represents a considerable improvement in methodology for performing sensitivity analyses of performance parameters of certain gun air defense weapon systems. This expression is combined into a computer program that produces plots on a line printer, thus greatly reducing the amount of manual effort needed to perform footprint studies. The program has been used most extensively for sensitivity analyses.

Sections of the report describe, in order, the operation of the program called "ISO-PK", the error analysis which determines the weapon pointing error, and the input data requirements for the ISO-PK program. The program logic diagram, program FORTRAN source listing, and a sample problem

[1] WECOM ADEB Task Force, Analysis of Air Defense Gun Systems, Technical Note RE-TR-70-191, Research and Engineering Directorate, HQ, US Army Weapons Command, Rock Island, IL, October 1970.

are in the final sections.

FOOTPRINT GEOMETRY

Consider a coordinate system with an air defense gun system at the origin of the axes as shown in Figure 1. The azimuth angle, a, is measured counterclockwise from +X in the X-Y plane. The elevation angle, e, is measured upward from the X-Y plane to the line from the origin to P (this line is called the slant range, r).

Let the X-Y plane represent the ground plane and z represent altitude. The target is assumed to be at a point in space with velocity components $(\dot{x}, 0, 0)$ (the dot notation here refers to the first time derivative of the variable). It is of particular interest to know how certain performance characteristics of the gun affect its overall performance in tracking and firing at a given target at points along straight lines parallel to the X-axis. These lines may be at various altitudes or at various values of Y (crossing ranges).

The air space around the gun is divided into a 3-dimensional grid so that only a discrete number of points P need to be considered. The user may select any plane parallel to the X-Y plane ($Z \geq 0$) or any plane parallel to the Z-X plane ($Y \geq 0$), see Figures 2 and 3. Given the volume of space to be considered in terms of limits on X, Y, Z, the ISO-PK program establishes a grid in two quadrants of the selected plane. The grid contains 100 increments horizontally and 50 increments vertically, corresponding roughly to the size of one page of computer paper on a printer with spacing increments of 1/10" by 1/6", respectively.

In the plane there are 51 x 101 = 5151 points denoted by P_{ij}; $i=1,2,\cdots, 101$; $j=1,2,\cdots, 51$. If the plane under consideration is parallel to the

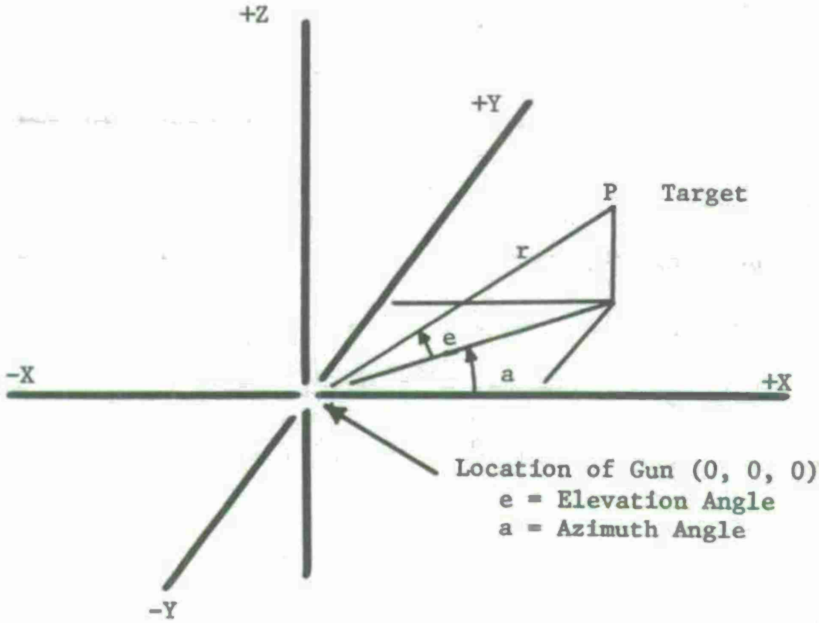


Figure 1. Gun-Target Geometry - Isometric View

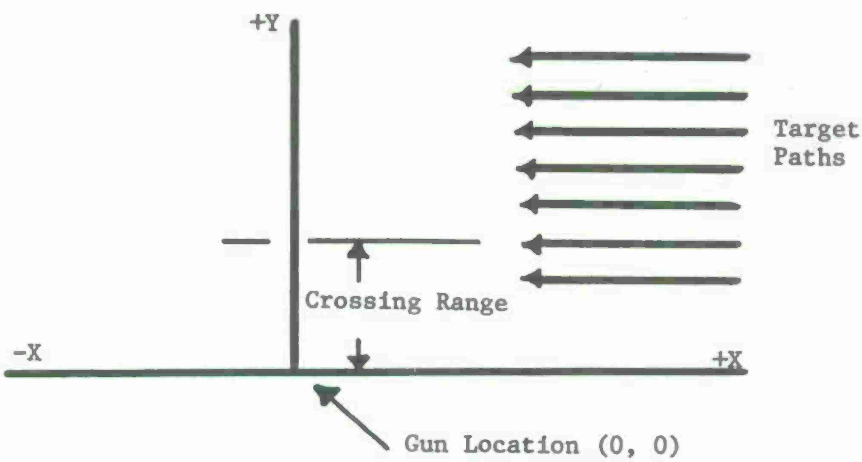


Figure 2. Gun-Target Geometry - X-Y Plane

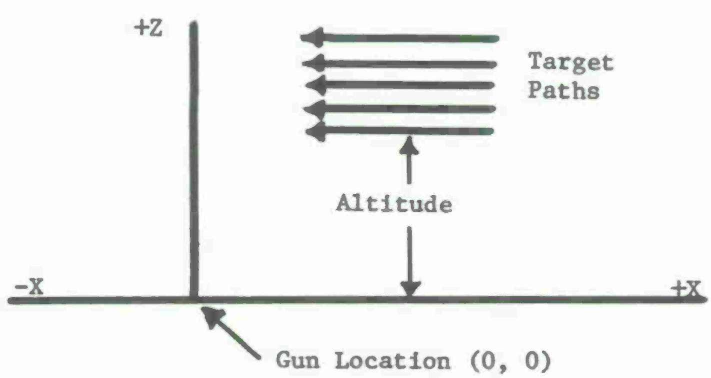


Figure 3. Gun-Target Geometry - X-Z Plane

X-Y plane ($Z \geq 0$), the Y position refers to the crossing range, (see Figure 2). If the plane is parallel to the Z-X plane ($Y \geq 0$), the Z position refers to the altitude (see Figure 3).

The ISO-PK program computes the variables x, y, z, a, e, r, $\dot{a}$, $\dot{e}$, $\dot{r}$, at each point P_{ij}. The gun system errors are determined as a function of these variables from gun system performance data which is provided as input. The resulting gun system errors are used to compute the burst-probability of kill (BPK) from the analytic formula in reference 2. The BPK is calculated about the point of predicted intercept of the target and the projectile. However, the value of the BPK is affixed to the point corresponding to the present position of the target (P_{ij}). Certain values of BPK (e.g., 0.05, 0.1, 0.5) and corresponding characters (e.g., H, I, K) are specified in the program to correspond to the constant BPK lines printed for the footprints. A character corresponding to the numerical value of the BPK is printed whenever the computed value of BPK at a point P_{ij} equals or crosses-over one of the specified values. The lines of constant-kill probability then appear as traces of printer characters in the plane of the paper. The matrix of computed BPK is saved during the horizontal raster scan and is scanned again in vertical order to fill in the contour lines and eliminate any apparent discontinuities in the printed lines when the lines have infinite slope.

Two separate computations are performed at each point P_{ij} and the results of both are superimposed on the computer plot. The first computation performed is the BPK determination as previously discussed. The

[2] Banash, Robert C., An Analytic Procedure for the Computation of Burst-Kill Probabilities in Air Defense. Technical Note SY-TN9-70, Systems Analysis Directorate, HQ, US Army Weapons Command, Rock Island, IL, October 1970.

second is a determination of possible dead zones caused by gun system constraints. The azimuth and elevation rates of the gun are computed and compared with maximum values allowed as specified in the input data. If either are greater than the maximums allowed, the letter A (azimuth) or E (elevation) will be printed corresponding to the type of constraint violated (if both are violated, only an A will appear). The gun elevation is also calculated and compared with the maximum elevation allowed. If the limit is exceeded, an E is printed. The program also prints the boundary formed by the maximum effective range of the gun. The range which is used in this comparison is the range at predicted intercept and not the slant range at time of burst. The character R is printed to signify violation of this constraint. Next, at each point P_{ij} a hypothetical situation concerning total system reaction time is evaluated. The letter T is printed if the target could, at its present velocity, move outside the range of the weapon if it was first detected at that point. This provides a means of making a comparative evaluation of a system's ability to initiate an engagement against a target suddenly appearing within the lethal volume of the system. In any particular problem, the analyst may also insert additional constraint conditions into this part of the program.

ERROR ANALYSIS

The BPK subroutine used in the ISO-PK program requires the means and standard deviations of the errors associated with the gun system. These errors are divided into two classes. One is the within-burst errors. These are the errors associated with the projectile (e.g., the residual dispersion and muzzle velocity variation). The other class of errors is the burst-to-burst errors due to inaccurate gun pointing.

The gun-pointing errors can be further subdivided into two categories. One is the error in the gun position due to errors in sensing the present position of the target. It is assumed that a gun system will measure the present position of the target and then, based upon these measurements, determine the lead angle necessary to achieve a hit on the target, given a certain open fire time. If these measurements are in error, the system will produce an incorrect lead angle (a gun pointing error).

The other source of gun pointing error is the gun servo mechanism. This error is due to the inaccuracies which are usually present in the servo system which positions the gun at a generated lead angle. Figure 4 shows a schematic of a gun air defense system with the various error sources noted (the "Ballistic Errors" are the within-burst errors). The program will accept the mean (bias), the standard deviation, and the covariances of the random components of each of the error sources.

An analysis of the gun pointing errors was reported in reference 2. This analysis assumed that sensor errors produced gun pointing errors of equal magnitude. Also, it assumed there was no lead angle generation. Consequently, the time of sensing, the time of fire, and the time of predicted intercept were all the same (no time of flight). A modification to this analysis is presented here which includes the time of flight and the effects of lead angle generation and approximates the lead angle error due to sensor error. The azimuth and elevation errors produced from this analysis is combined with the gun servo errors. These combined errors are then projected into a two dimensional plane at the target at the time of predicted intercept. Note that the target position at predicted intercept is not the same as the present position of the target.

---

[2] Ibid.

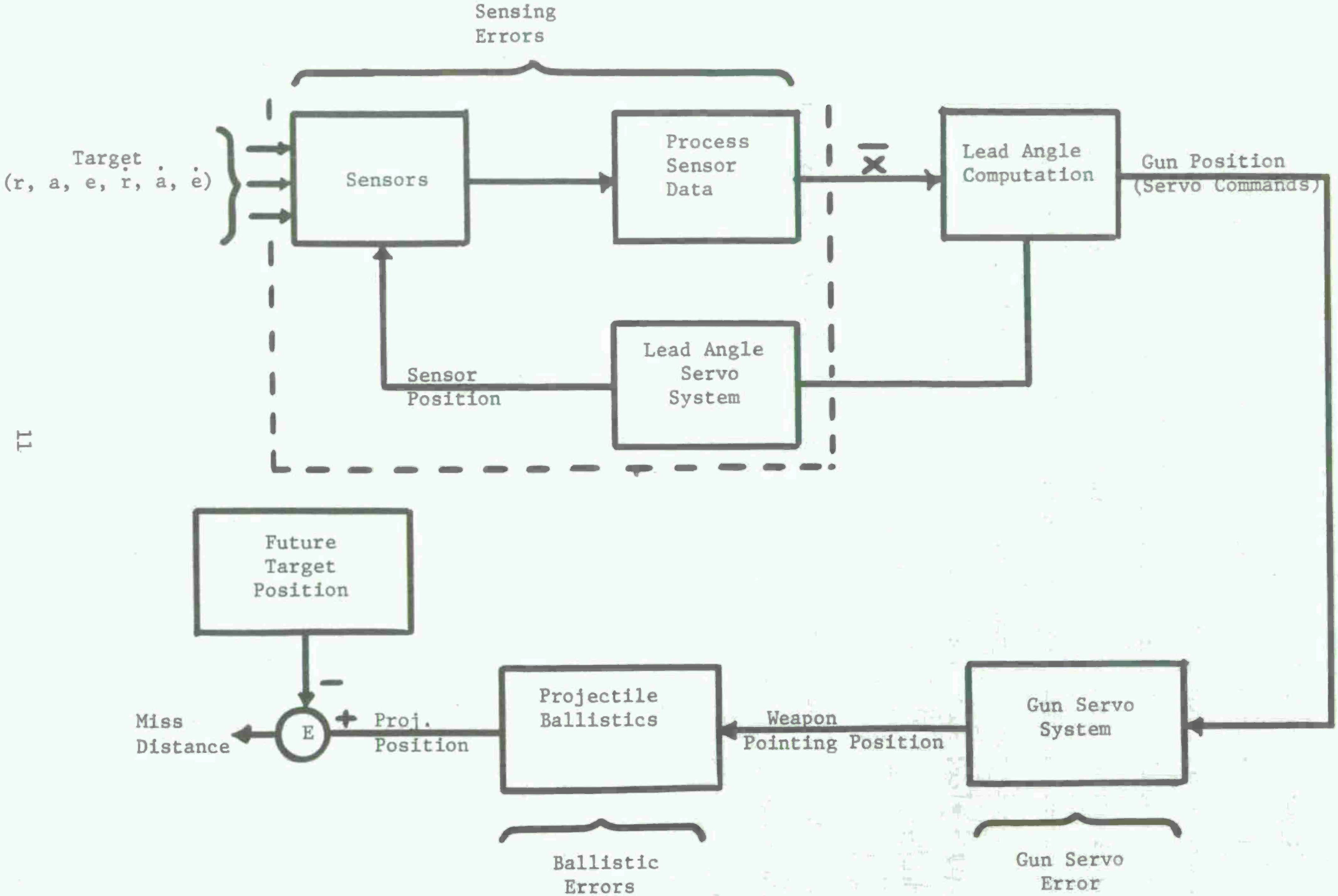


Figure 4. Schematic of the Basic Components of a Gun Air Defense System

Sensor Errors

To determine the gun pointing error due to sensor errors, a first-order error analysis is performed on the equations defining the position of the gun. The independent variables in this analysis are the measurements of the present position of the target $(r, a, e, \dot{r}, \dot{a}, \dot{e})$, and the dependent variables are the azimuth and the elevation of the gun.

Define the gun barrel position by the vector $\bar{Z}$, ($Z(1)$ = azimuth, $Z(2)$ = elevation), and the vector of functions which define $\bar{Z}$ to be $f(\bar{X})$. This is

$$\bar{Z} = f(\bar{X})$$

Where $\bar{X}$ is the vector of measurements taken at the present position of the target

$$\bar{X} = (r, a, e, \dot{r}, \dot{a}, \dot{e})^T \quad .$$

The functions which define the gun position are derived in the Appendix.

The first-order approximation of the change in the dependent variables $(\Delta\bar{Z})$ resulting from a small change in the independent variable $(\Delta\bar{X})$, about a nominal point $\bar{X} = \bar{X}_0$ can be expressed as

$$\Delta\bar{Z} = \left.\frac{\partial f(\bar{X})}{\partial\bar{X}}\right|_{X_0} \Delta\bar{X} \quad .$$

The small change $\Delta\bar{X}$ may be interpreted to be a random error associated with the measurements of the present position of the target. That is, $\Delta\bar{X}$ is a random variable with mean $\bar{M}_x$ and covariance matrix $COV(\bar{X})$. Therefore, $\Delta\bar{Z}$ is also a random variable describing the errors in the gun pointing position. Define the mean value of $\Delta\bar{Z}$ to be $\bar{M}_Z$ and its covariance matrix to be $COV(\bar{Z})$.

Let

$$F = \frac{\partial f(\bar{X})}{\partial \bar{X}} \bigg|_{\bar{X}_0} ,$$

which is the Jacobian matrix evaluated at the nominal value $\bar{X}_0$. Then the mean value of $\Delta\bar{Z}$ is

$$E\,[\Delta Z] = F\,E[\Delta \bar{X}]$$

or

$$\bar{M}_Z = F\,\bar{M}_X \quad .$$

$E[\cdot]$ denotes expected value of the random variable. Also, the covariance matrix of the errors $\Delta\bar{Z}$ is given by

$$COV(\bar{Z}) = F\,COV(\bar{X})\,F^T \quad .$$

The partial derivatives necessary to define F in FORTRAN notation are listed in the section titled FORTRAN Program Source Listing, page 56.

The statistics of the measurement errors ($\bar{M}_X$ and $COV(\bar{X})$) are assumed to be constant (in time). However, provisions have been made in the ISO-PK program for a user-supplied subroutine which can modify these errors as a function of the engagement parameters $(a, e, r, \dot{a}, \dot{e}, \dot{r})$. This subroutine is called MESIG.

Gun Servo Errors

The gun servo errors will be considered next. Define the mean servo error to be the two-vector $\bar{M}_s$ and the (2 x 2) covariance matrix for the servo errors to be $COV(\bar{S})$. The first row in $\bar{M}_s$ and $COV(\bar{S})$ corresponds to azimuth while the second row corresponds to elevation error. The mean values of the gun servo errors are assumed to be constant for a gun system. These bias type errors may be likened to a bore sight error. The covariance matrix is assumed to be a function of the gun rates. It is also assumed that the azimuth and elevation errors are independent of one another, so the off-diagonal terms of the matrix $COV(\bar{S})$ are zero. The standard deviations of the errors on the azimuth and elevation directions are input to the ISO-PK program as a table of values which are a function of gun rates. A corresponding table of gun rates is also input to the program. In the program, the appropriate standard deviations are determined by looking up the rates in the table and linearly interpolating between the tabular values. They are then squared to obtain the variances. It is expected that detailed system performance models of the servos of future gun systems will be developed and represented by these tables for evaluations with the ISO-PK program.

Combining Gun System Errors

The errors from the gun servo system are assumed to add to the gun pointing errors resulting from sensor errors. Therefore, since these two error sources are assumed to be independent, their statistics can be added to obtain the mean and covariance of the total gun pointing error.

The gun pointing errors produced by the above analysis are in angular units. However, the errors required by the BPK subroutine are in units of length, in a coordinate system centered at the center-of-vulnerability of the target. This coordinate system is perpendicular to the line from the gun to the target at time of predicted intercept. The conversion in the elevation direction, from angular units to meters, is achieved by multiplying the elevation error by the range (R) to the target at predicted intercept. In the azimuth direction, the conversion is obtained by multiplying by RCOS(E), where E is the elevation of the target at predicted intercept. The cosine term is needed to convert the azimuth error in the ground plane to an azimuth-like (transverse) error in the elevated coordinate system.

The statistics describing the errors, in meters, in the elevated coordinate system are as follows:

Among-Burst Error:

Mean:

$$\begin{bmatrix} M_A \\ \\ M_E \end{bmatrix} = (\bar{M}_Z + \bar{M}_S) \cdot \begin{bmatrix} R\ COS(E) \\ \\ R \end{bmatrix}$$

Covariance:

$$\begin{bmatrix} \sigma_A^2 & COV(A,E) \\ \\ COV(A,E) & \sigma_E^2 \end{bmatrix} = \begin{bmatrix} R\ COS(E) & 0 \\ \\ 0 & R \end{bmatrix} (COV(\bar{Z}) + COV(\bar{S})) \begin{bmatrix} R\ COS(E) & 0 \\ \\ 0 & R \end{bmatrix}^T$$

These combined statistics are those required to describe the among-burst errors. The BPK subroutine uses the correlation coefficient instead of the covariance term COV(A,E). This correlation coefficient is defined as:

$$\rho_{A,E} = \frac{COV(A,E)}{\sigma_A \sigma_E} .$$

Within-Burst Errors

A brief derivation of the within-burst errors is presented here. A similar, more complete derivation can be found in reference 3. The angular residual dispersion error and range variation error of the projectile must be projected into the elevated coordinate system centered at the target. Define the errors as follows:

Δ_A - azimuth-like (transverse) angular dispersion

Δ_E - elevation angular dispersion

Δ_R - projectile range error

These errors are assumed to be independent, normally distributed with zero means and variances of σ_a^2, σ_e^2, σ_r^2, respectively. The variance of the projectile range error is due to the muzzle velocity variation.

$$\sigma_r^2 = \left(\frac{R}{V_m}\right)^2 \sigma_V^2$$

Where

V_m = muzzle velocity of the projectile

σ_V^2 = variance of the muzzle velocity

Since there is an error in the range coordinate, the time to equal range of the target and projectile must be determined. This time can be approximated by the following if $V_S \gg \dot{R}$.

$$\Delta t = -\frac{\Delta R}{(-\dot{R} + V_S)} \approx \frac{\Delta R}{V_S} \quad ,$$

[3] Banash, Robert C., Notes of the University of Michigan Analytic Gun Model, Technical Note SY-TN3-70, Systems Analysis Directorate, HQ, US Army Weapons Command, Rock Island, IL, April 1970.

where

$\dot{R}$ = range rate of the target

V_S = velocity of the projectile at range R

The velocity components of the target in the elevated coordinate system are $(\dot{A}RCOS(E), \dot{R}, \dot{E}R)$. Where $\dot{A}$ and $\dot{E}$ are the azimuth and elevation rates of change of the target at the time of predicted intercept. The velocity components of the projectile are $(0, V_S, 0)$. Therefore, the relative velocity components between the target and projectile are $-(\dot{A}RCOS(E), -V_S, \dot{E}R)$. The errors, in meters, at the time when the projectile and target are at equal range are

$$\Delta A' = R\ \Delta A - \Delta t\ \dot{A}RCOS(E)$$

$$= R\ \Delta A + \frac{\Delta R}{V_S}\ \dot{A}RCOS(E)$$

and

$$\Delta E' = R\ \Delta E + \frac{\Delta R}{V_S}\ \dot{E}R$$

$\Delta A'$ is the error in the traverse direction and $\Delta E'$ is the error in the elevation direction. Since these quantities are sums of independent, normal random variables, they have a bivariate normal distribution with the following parameters.

Within-Burst Errors:

Means:

$E\ [\Delta A'] = 0$

$E\ [\Delta E'] = 0$

Variances:

$$\sigma^2_{A'} = R^2\sigma^2_{\dot{A}} + \left(\frac{R\cos(E)\dot{A}}{V_S}\right)^2 \sigma^2_r$$

$$\sigma^2_{E'} = R^2\sigma^2_{\dot{E}} + \left(\frac{R\dot{E}}{V_S}\right)^2 \sigma^2_r$$

Correlation coefficient:

$$\rho_{A', E'} = \frac{COV(\Delta A', \Delta E')}{\sigma_{A'}\sigma_{E'}}$$

$$= \frac{1}{\sigma_{A'}\sigma_{E'}} COV((R\ \Delta A + \frac{R\cos(E)\dot{A}}{V_S}\ \Delta R), (R\ \Delta E + \frac{R\dot{E}}{V_S}\ \Delta R))$$

$$= \frac{1}{\sigma_{A'}\sigma_{E'}} \frac{\dot{A}\dot{E}\ R^2 \cos(E)}{V_S^2} \sigma^2_r$$

At this point all the necessary inputs are presented for the computation of burst-probability of kill (BPK) as presented in the form in reference 2, starting at page 6. The following relates the notation used in this report to that of reference 2:

For Within-Burst Errors:

$$M_1 \equiv E[\Delta A']$$

$$M_2 \equiv E[\Delta E']$$

$$\sigma^2_1 \equiv \sigma^2_{A'}$$

$$\sigma^2_2 \equiv \sigma^2_{E'}$$

$$\rho_\omega \equiv \rho_{A',E'}$$

---

[2] Loc. Cit.

For Among-Burst Errors:

$$M_3 \equiv M_A$$

$$M_4 \equiv M_E$$

$$\sigma_3^2 \equiv \sigma_A^2$$

$$\sigma_4^2 \equiv \sigma_E^2$$

$$\rho \equiv \rho_{A,E}$$

Before presenting the equation used for BPK from references 2, 3 or 4, the basic assumption limiting the validity of this equation is reiterated:

a. Target vulnerability area is small compared to the within-burst dispersion.

b. Rounds can be grouped within bursts such that the burst centers can be considered independently distributed while the rounds are distributed independently about the burst centers.

The ' notation used in the following equations means transpose of the indicated matrix.

Under these assumptions, the equation for BPK is given as:

$$BPK = \sum_{j=1}^{n} \binom{n}{j} (-1)^{j+1} \left(\frac{A_V}{2\pi\sigma_1\sigma_2}\right)^j \left| I+jA \right|^{-1/2}$$

$$\cdot \exp\{- \frac{j}{2} v' A[I-j(I+jA)^{-1} A] v\}$$

where n is the number of rounds in the burst and

---

[2]Loc. Cit.

[3]Loc. Cit.

[4]Banash, Robert C., A Markov Chain Approach to Modeling Tracking Error In Air Defense Gun Evaluation, Technical Report No. (GADES 3-1), Artillery and Air Defense Weapons Directorate, Rock Island Weapons Laboratory, Rock Island, IL, September 1971.

$$A = K'P' \begin{bmatrix} \sigma_2^2 & \rho_\omega \sigma_1 \sigma_2 \\ \rho_\omega \sigma_1 \sigma_2 & \sigma_2^2 \end{bmatrix} PK$$

P is the matrix of characteristic vectors of

$$V = \begin{bmatrix} \sigma_3^2 & \rho \sigma_3 \sigma_4 \\ \rho \sigma_3 \sigma_4 & \sigma_4^2 \end{bmatrix} ,$$

$$P'VP = \begin{bmatrix} \frac{1}{\alpha_1} & 0 \\ 0 & \frac{1}{\alpha_2} \end{bmatrix} \equiv K^2 .$$

Finally,

$$v = K^{-1}P \begin{bmatrix} M_3 \\ M_4 \end{bmatrix} .$$

A_V is the target vulnerable area.

INPUT DATA DEFINITION

The input data is divided into three sections: (1) data to control the planes, (2) gun system data, at which the plots are to be made and (3) target data. The program is set up such that the data for each category is read from a different numbered data set. Each category is explained with regard to data content, meaning, and format, as follows:

Control of Selected Planes.

This data controls the size and location of the planes on which the footprint is printed. It also determines the velocity (magnitude and direction) of the target. Each card is described in the order in which it is read. In the following discussion, the paragraphs are headed by the list of variables whose values are read. Detailed description of the variables and how they relate to the program are also presented. This format is also used in the next two sections -- Gun Systems Data and Target Data.

NVEL, NCR, NALT, PLOTXY, PLOTXZ
(3I5,2L5)

NVEL - The number of different velocities which will appear on a later card. One plot will be made for each velocity.

NCR - The number of different crossing ranges at which XZ plots are to be made. The actual values of the crossing ranges appears on a later card. Also, this data only has meaning when an XZ plot is to be made.

NALT - The number of different altitudes at which X-Y plots are to be made. Again the actual values of the altitudes appear on a later data card, and this card is of concern only when X-Y plots are to be made.

PLOTXY - This logical constant should be set equal to T (true) if X-Y plots (at various altitudes) are to be made. If no X-Y plots are required, it should be set equal to F (false).

PLOTXZ - This logical constant should be set equal to T (true) if XZ plots (at various crossing ranges) are to be made. If no X-Z plots are required, it should be set equal to F (false).

XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
(6F10.0)

XMIN - The combination of all the data on this card sets the size of the plots which are to be made. XMIN contains the minimum value of the X-axis.

XMAX - The maximum value of the X-axis.

YMIN - If an X-Y plot is to be made, the minimum value of Y (i.e., the minimum crossing range) is contained in the variable YMIN.

YMAX - The maximum value of the Y-axis. The combination of XMIN, XMAX, YMIN, and YMAX describes the size of the X-Y plane.

ZMIN - If an X-Z plot is to be made, the minimum value of the Z-axis (minimum altitude) is contained in this variable.

ZMAX - The maximum value of the Z-axis. The combination of XMIN, XMAX, ZMIN, and ZMAX describes the size of the X-Z plane.

VEL (I), I = 1, NVEL
(7F10.0)

The VEL array contains the actual values of the different velocities for which plots are to be made. The number of velocities which will be read is equal to NVEL.

CRANGE(I), I = 1, NCR
(7F10.0)

The CRANGE array contains the actual values of the different crossing ranges for which X-Z plots are to be made. The number of values which will be read equals NCR.

ALT(I), I = 1, NALT

> The ALT array contains the actual values of the different altitudes at which X-Y plots are to be made. The number of values which will be read is equal to NALT.

Altogether, there will be (NVEL*(NCR + NALT)) plots made if both PLOTXY and PLOTXZ are initialized to T (true). If only PLOTXY is equal to true, then there will be (NVEL*NALT) plots. Where, if only PLOTXZ is equal to true, there will be (NVEL*NCR) plots. The size of all plots is determined by the parameters on the second card.

Gun Systems Data

The data contained in this data set describe the parameters related to the gun system. This includes the gun constraints as well as the description of the random errors of the system. The same format will be used in this discussion as was used in the previous section in describing the data, card by card.

TITLE
(20A4)

> This is an 80-character verbal title which appears on all the plots. It may contain such identifying information as the name of the system being investigated or some particular configuration of a system. The actual content of the information on this card is optional but must be limited to 80 characters.

CALIB
(I5)

> The value of variable CALIB is used to determine from which numbered data set the target data is to be read. In doing this, 10 is added to the value of CALIB to determine the data set number.

SIGVO, SIGXD, SIGYD, VO
(4F10.0)

The standard deviation of the muzzle velocity of the projectile (σ_{v_o}). The units are meters/sec.

SIGXD - The standard deviation of the residual dispersion of the projectiles in the traverse direction (σ_A). This variable should be in units of radians.

SIGYD - The standard deviation of the residual dispersion of the projectiles in the elevation direction (σ_E). This variable should be in units of radians.

VO - The mean value of the muzzle velocity of the projectile in meters per second.

EDMAX, ADMAX, TREAC, RNDS, RMAX, ELMAX
(6F10.0)

EDMAX - The maximum elevation rate which can be achieved by the gun in radians per second.

ADMAX - The maximum azimuth rate which can be achieved by the gun in radians per second.

TREAC - The reaction time for the gun system in seconds. This reaction time is defined as the time from initial detection of a target to the time when the system can begin firing.

RNDS - The number of rounds in a burst from the gun.

RMAX - The maximum effective range of the projectile in meters. This is usually determined by the range at which the velocity of the projectile is at such a low level that it can no longer inflict damage to the target.

ELMAX - The maximum elevation angle in radians, to which the gun can elevate.

R, TF
(2F10.0)

These two variables are used to determine the ballistic drag coefficient of the projectile. R is a range in meters and TF is the time of flight in seconds to that range. These two parameters can be taken from anywhere on the range - time of flight curve of a projectile, especially if the projectile obeys the "3/2 law" for the drag force. If the projectile doesn't follow this law, a good approximation is usually achieved at a range which is 2/3 the maximum range.

DYNAM, DYNEM, MAXAZ, MAXEL
(2F10.0,2I5)

DYNAM - The mean gun servo error in the azimuth direction. This error is usually considered to be a bore sight error since servo mechanisms are usually not designed with a mean error in them. The units are radians.

DYNEM - The mean gun servo error in the elevation direction. This is usually the result of a bore sight error. The units are radians.

MAXAZ - The number of entries in the azimuth rate - standard deviation tables. (MAXAZ $\leq$ 10)

MAXEL - The number of entries in the elevation rate - standard deviation tables. (MAXEL $\leq$ 10)

ELDOT(I), I = 1, MAXEL
(7F10.0)

The ELDOT array contains the elevation rate entries in the elevation rate vs. standard deviation tables. The units are radians per second and the number of entries should equal the value of MAXEL.

DYEL(I), I = 1, MAXEL
(7F10.0)

The DYEL array contains the standard deviations (in radians) in the elevation direction of the gun servo system error. The quantities of this array must correspond in order to the elevation rates in the ELDOT array. The combination of these two arrays (ELDOT - DYEL) describes a curve relating the standard deviation of the gun servo error in the elevation direction to the gun elevation rate. A linear interpolation between the quantities in these arrays is used to achieve the standard deviation of the elevation error at any particular value of elevation rate.

AZDOT(I), I = MAXAZ
(7F10.0)

The AZDOT array contains the gun azimuth rate entries in the azimuth rate vs. standard deviation tables. The units are radians per second and the number of entries should equal the value of MAXAZ.

DYAZ(I), I = MAXAZ
(7F10.0)

The DYAZ contains the standard deviations (in radians) in the azimuth direction of the gun servo system error. As was the case with the DYEL array, the entries in the DYAZ array should correspond to the rates in the AZDOT array, and a linear interpolation is used to find the standard deviation at any particular value of azimuth rate.

XM (I)
(6F10.0)

The XM array contains the mean value of the measurement errors. Earlier in the text of this report these quantities were described with the symbol $\overline{M}_x$. The units are radians - meters - seconds. The order is: 1) range, 2) azimuth, 3) elevation, 4) range rate, 5) azimuth rate, 6) elevation rate.

SIG (I, J)
(6F10.0)

The SIG matrix contains the standard deviations and correlation coefficients of the measurement errors. The terms on the diagonal of this 6 x 6 symmetric matrix are the standard deviations of the errors. The off diagonal terms are the correlation coefficients. The matrix is read in one row at a time (i.e., one row per card) following the same order as the mean errors. This is, range first, azimuth second, etc. The units are radians - meters - seconds. To illustrate the order of the cards, the information on the first three cards is as follows:

1st card - σ_r $\rho_{r,a}$ $\rho_{r,e}$ $\rho_{r,\dot{r}}$ $\rho_{r,\dot{a}}$ $\rho_{r,\dot{e}}$

2nd card - $\rho_{r,a}$ σ_a $\rho_{a,e}$ $\rho_{a,\dot{r}}$ $\rho_{a,\dot{a}}$ $\rho_{a,\dot{e}}$

3rd card - $\rho_{r,e}$ $\rho_{a,e}$ σ_e $\rho_{e,\dot{r}}$ $\rho_{e,\dot{a}}$ $\rho_{e,\dot{e}}$

etc.

Target Data

The target is described as a combination parallelepipeds which represent the vulnerable components which make up the target. There may be as many as seven components which make up the target. Three values of each of the six faces of the components are input to the program. The three values of each area correspond to the vulnerable area presented as a function of projectile striking velocity. Presently, the program is set up so that the first area for each face is for striking velocity of 500 ft/sec and the second area is for 1000 ft/sec. The third value is not currently being used; however, space must be allowed for it on the data cards. As was mentioned in the previous section, the value of the variable CALIB is used to determine the numbered data set on which this target data is found. A card by card description of the data is given below.

EMAX
(I5)

The number of vulnerable components from which the target is made should be read into the variable EMAX.

S(I, J, K)
(6F10.5)

The S array contains the values of the areas of the faces of the vulnerable components in units of square meters. Because of the format by which these quantities are read, three cards are required to define each of the (EMAX) components. The first three quantities on the first card are the three areas of the front side of the component, while the second three are the areas of the rear of the component. On the second card are the areas of the port and starboard sides of the component (in that order). And on the last card the areas of the bottom and top sides of the component are given (in that order).

Next page is blank.

LOGIC DIAGRAM

A schematic diagram of the program and subprogram logic is presented in Figure 5.

Next page is blank.

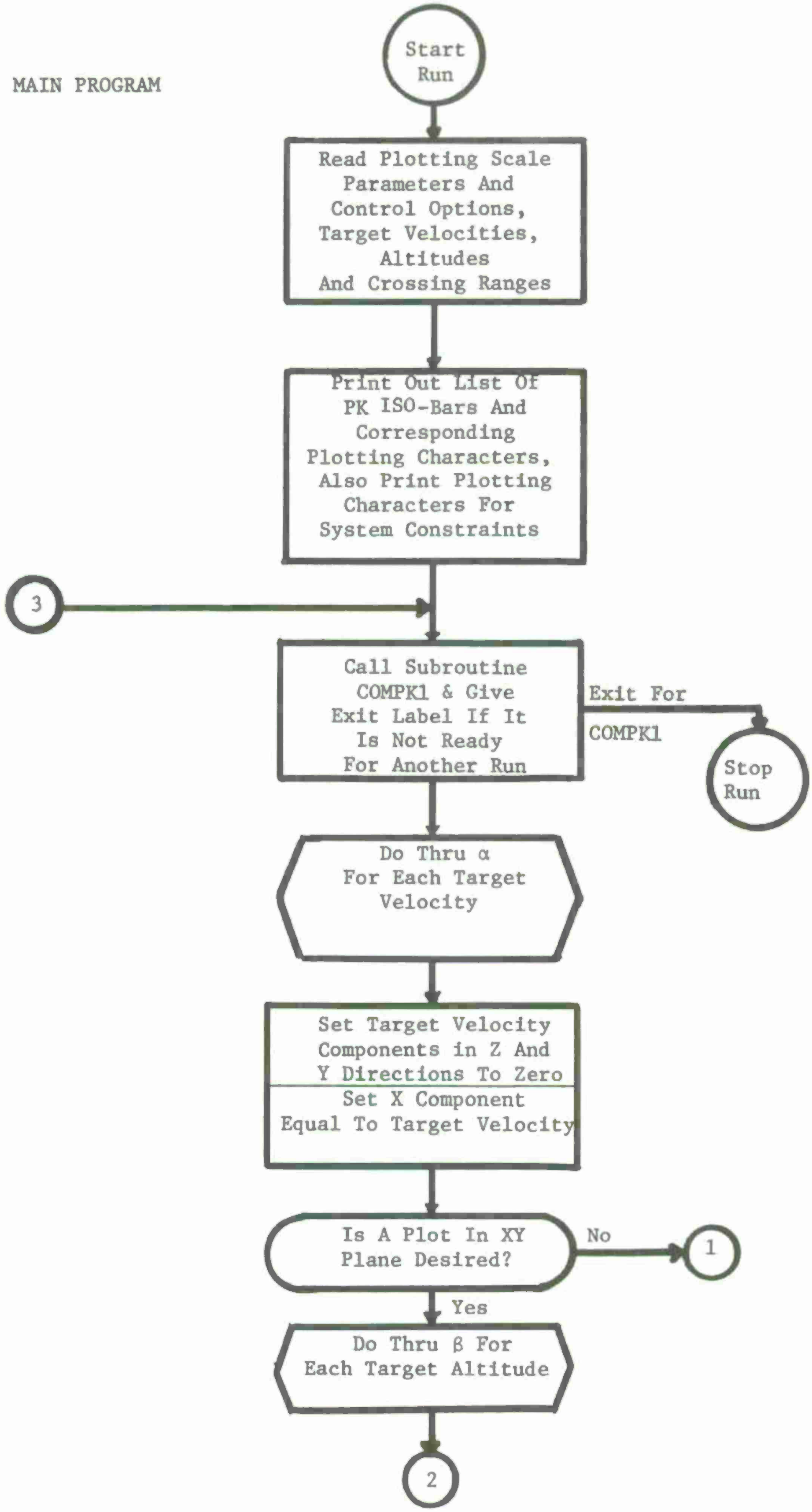


Figure 5. Logic Diagram (1 of 8)

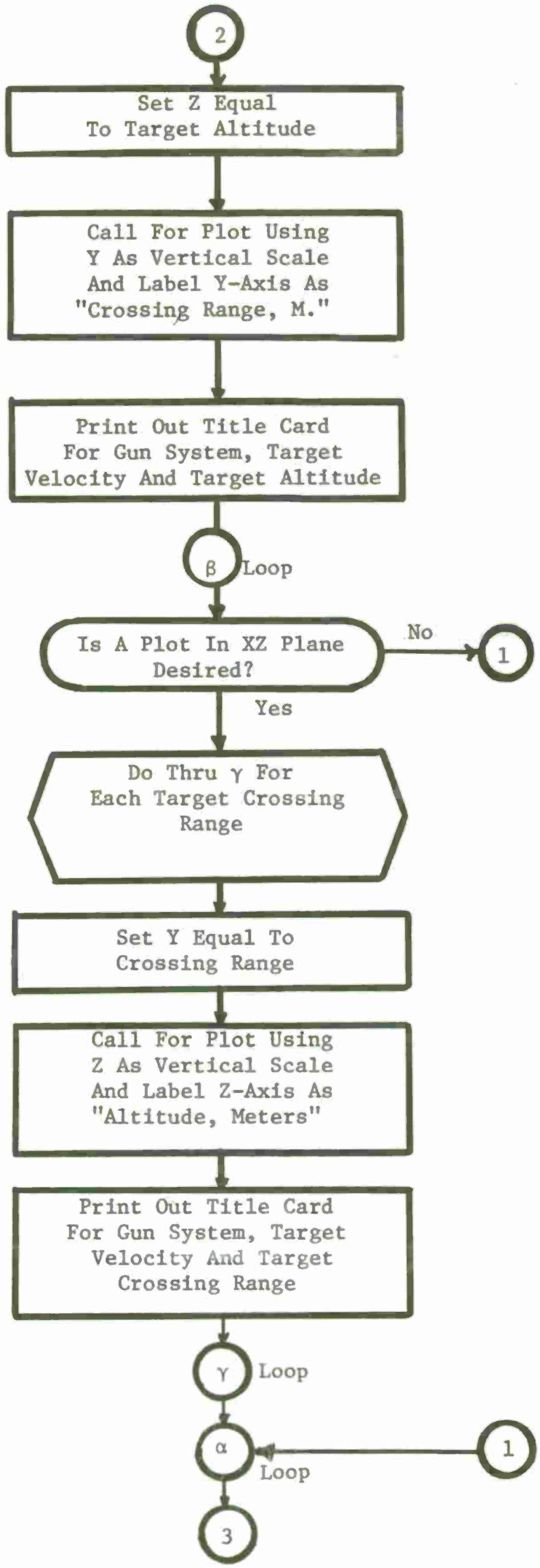


Figure 5. Logic Diagram (2 of 8)

SUBROUTINE PLOT

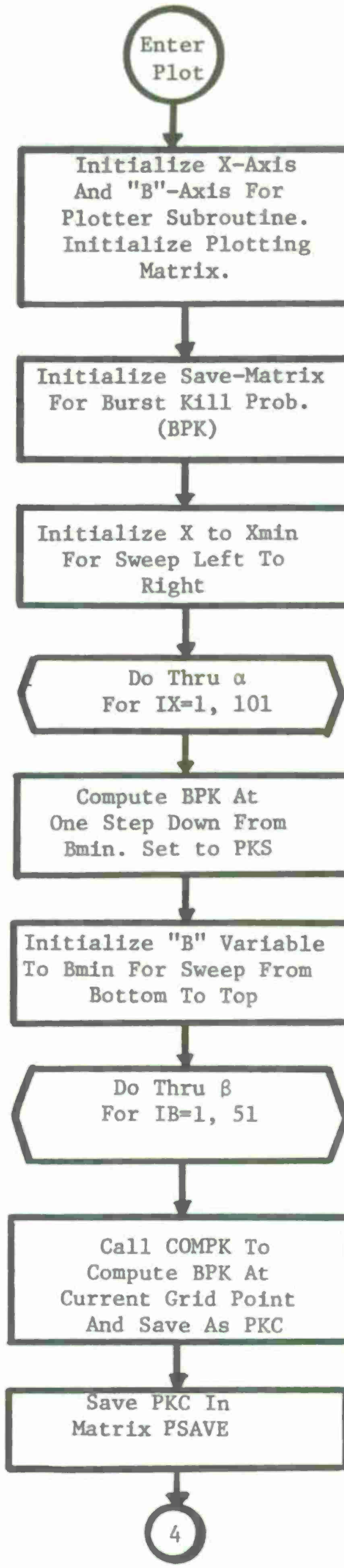


Figure 5. Logic Diagram (3 of 8)

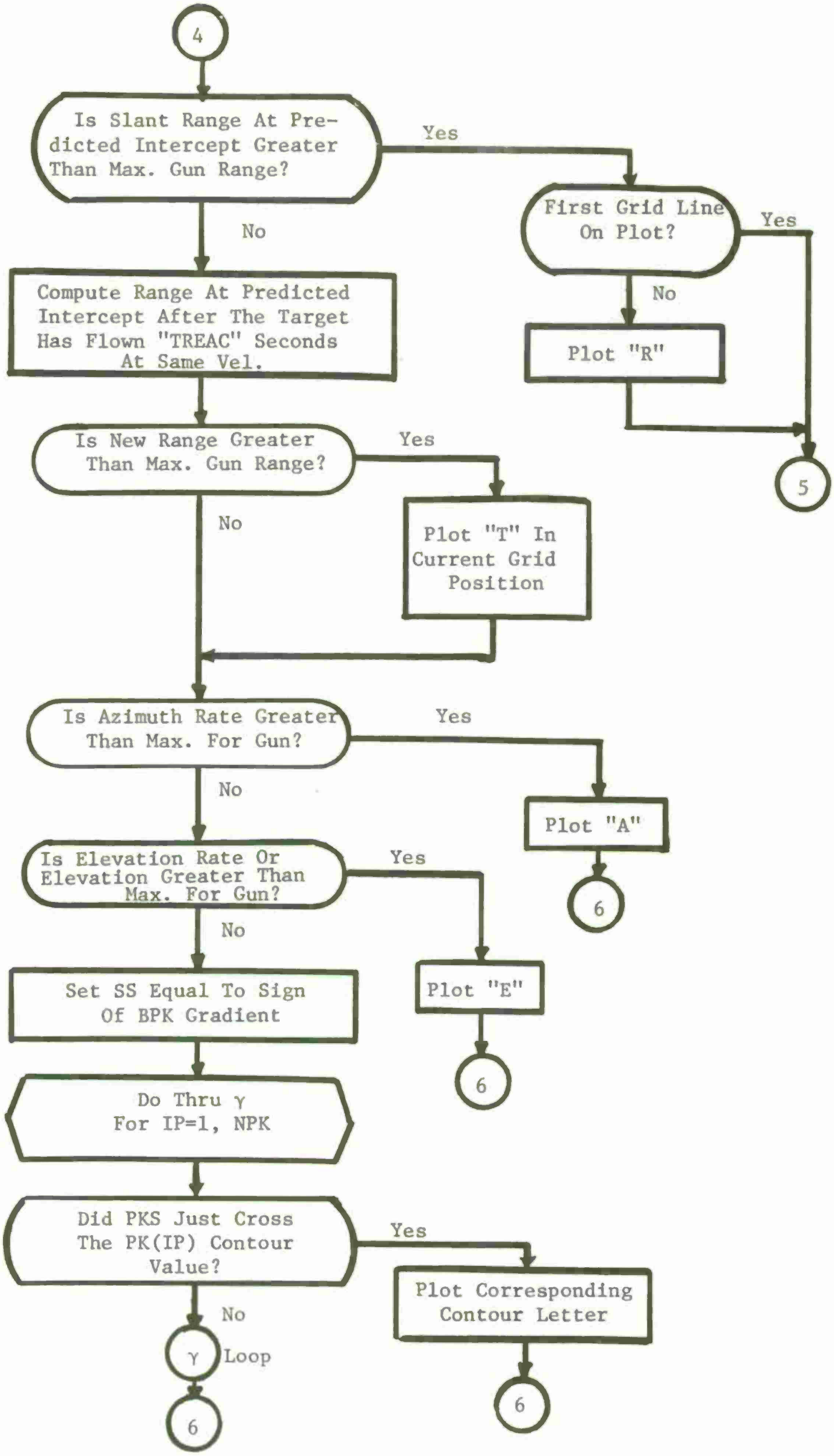


Figure 5. Logic Diagram (4 of 8)

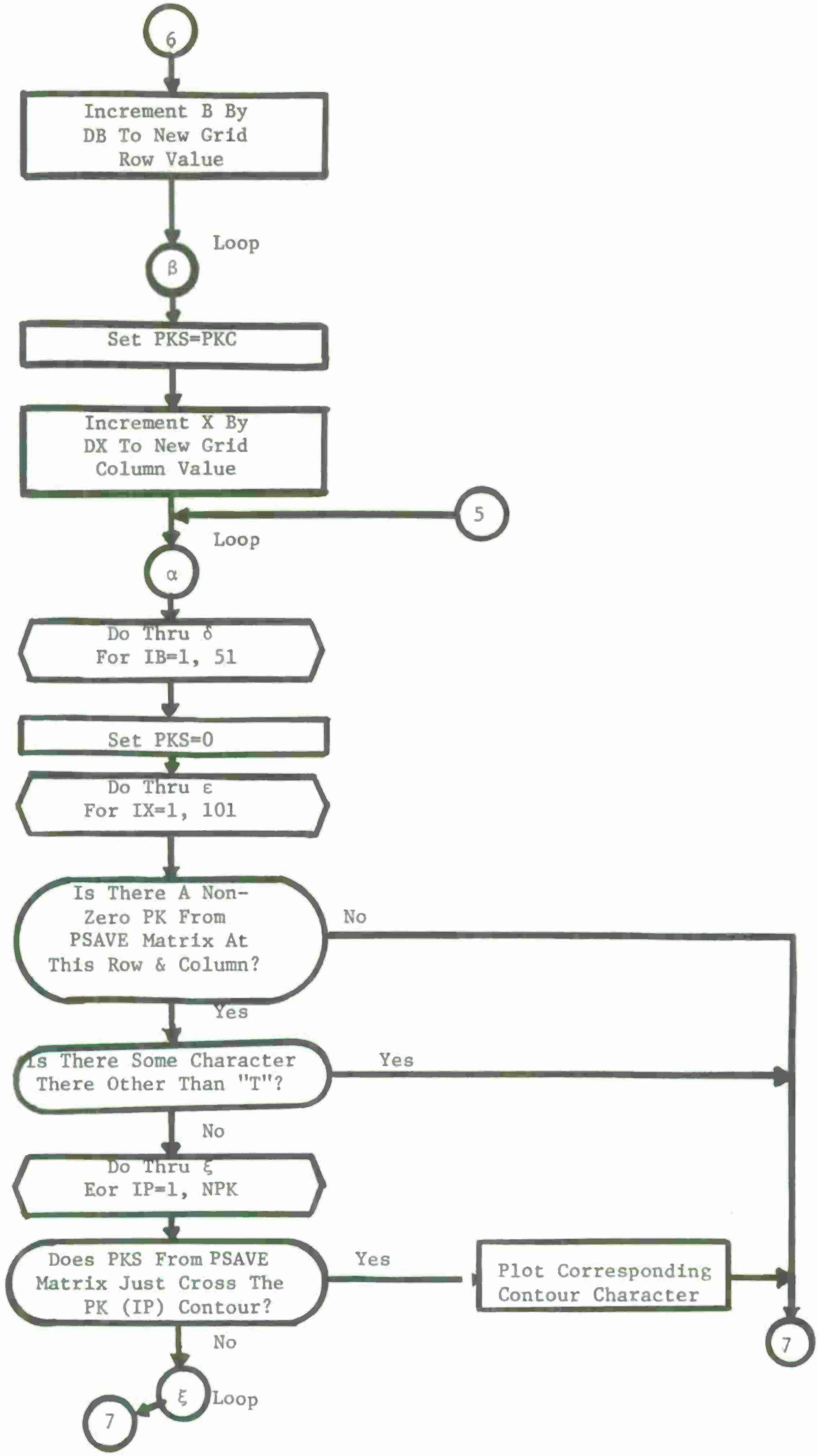


Figure 5. Logic Diagram (5 of 8)

SUBROUTINE COMPK1

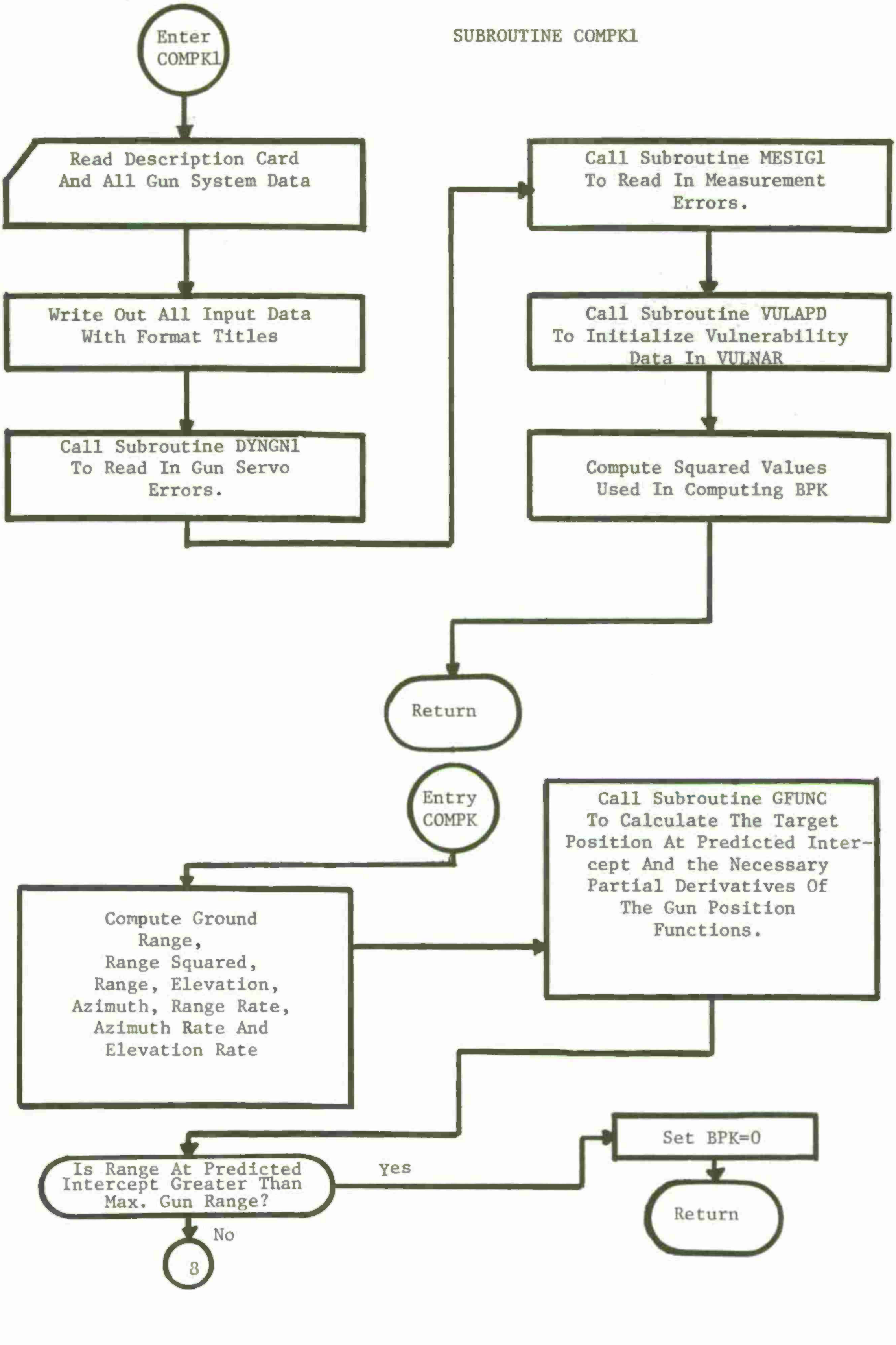


Figure 5. Logic Diagram (6 of 8)

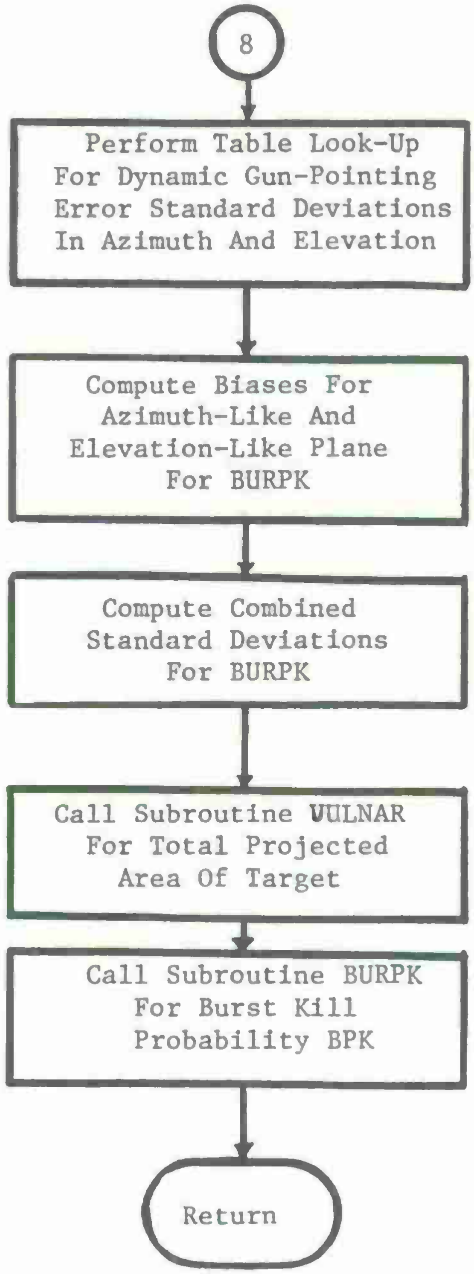


Figure 5. Logic Diagram (7 of 8)

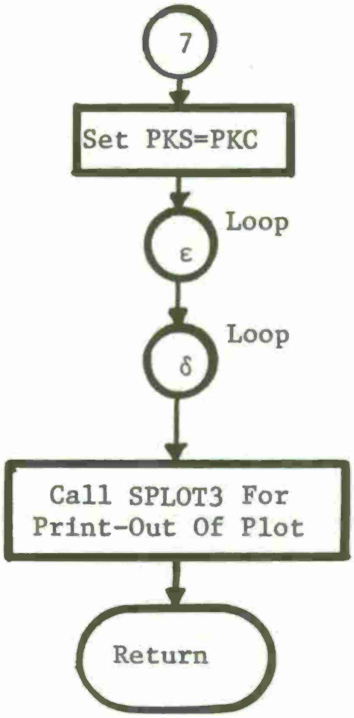


Figure 5. Logic Diagram (8 of 8)

FORTRAN PROGRAM SOURCE LISTING

This section provides a computer printout of the FORTRAN statements making up the ISO-PK program.

Next page is blank.

```
C     ***** U N I M A P -- A I R   D E F E N S E   F O O T P R I N T S
C     THIS PROGRAM DEVELOPS LINES OF CONSTANT BURST-KILL PROBABILITY FOR
C     AIR DEFENSE GUN SYSTEMS. THE MODEL USED IS AN EXPANDED VERSION OF
C     THE UNIVERSITY OF MICHIGAN ANALYTIC MODEL DEVELOPED IN THE AFAADS
C     GUN STUDY
C     THE MAIN PROGRAM ACTS AS A DRIVER AND CONTROL FOR THE MODULES THAT
C     PLOT AND COMPUTE.
C     THE PROGRAM READS PLOTTING CONTROL INFORMATION AND OPTIONS
C     AND PRINTS TABLES OF EQUIVALENTS BETWEEN PROBABILITIES AND
C     PLOTTING CHARACTERS.
C***********************************************************************
C     COORDINATE SYSTEM FOR UNIMAP
C            +Z|
C              |
C              |              +Y
C              |               /
C              |              /
C              |             /
C              |            /
C              |           /
C              |          /           *<---- TARGET POSITION
C              |         /          /  |
C              |        /         /    |
C              |       /        /      |
C              |      /       /        |
C              |     /      /          |
C              |    /     /            |
C              |   /----/-------------*
C              |  /   /              /
C              | /  /               /
C   -X         |/ /                /                            +X
C     ---------+------------------/------------------------------
C            /|GUN
C           / |         AZIMUTH ANGLE MEASURED FROM +X IN XY PLANE.
C          /  |
C         /   |         ELEVATION ANGLE MEASURED FROM XY PLANE TO
C        /    |           SLANT RANGE LINE FROM GUN TO TARGET.
C       /     |
C      /      |
C     /       |
C   -Y/       |
C    /         |
C***********************************************************************
C***********************************************************************
C
C                       CONSIDER THE XY PLANE:
C                               +Y
C                                |    <-------------<<<
C                                |    <-------------<<<
C                                |    <-------------<<<
C                                |    <-------------<<<
C                                |    <-------------<<<
C                                |    <-------------<<<
```

```
          C                             |     <-------------<<<
          C                             |     <-------------<<<   TARGET
          C                             |     <-------------<<<   PATHS
          C                             |     <-------------<<<
          C                             |     <-------------<<<
          C                             |     <-------------<<<
          C                             |     <-------------<<<
          C                             |     <-------------<<<
          C                             |     <-------------<<<
          C                             |     <-------------<<<
          C                             |     <-------------<<<
          C                             |     <-------------<<<
          C   --------------------------+--------------------------------
          C   -X                      (0,0)                             +Y
          C                            GUN
          C     THE SCENARIO IN THE XZ PLANE IS SIMILAR BUT WITH Z BEING THE
          C       VERTICAL AXIS.
          C
          C*****************************************************************
          C
0001            COMMON X,Y(1),Z(1),XDOT,YDOT,ZDOT,XDD,YDD,ZDD,EGDOT,AGDOT,RSQ,VP,
               . TF,RNDS,VO,RMSQ,TREAC,EDMAX,ADMAX,XMIN,XMAX,TITLE(20),CALIB
0002            COMMON /$PLOT/ PK,CHAR,MCHAR,NPK
0003            INTEGER CALIB
          C
          C   CALIB=1:  20MM HE PROJ., KK KILL.
          C   CALIB=2:  30MM HE PROJ., KK KILL.
          C   CALIB=3:  40MM HE PROJ., KK KILL.
          C   CALIB=4:  57MM HE PROJ., KK KILL.
          C   CALIB=5:  8 LB WARHEAD, KK KILL.
          C   CALIB=6:  20MM HE PROJ., CLEAN AIRCRAFT, A KILL.
          C   CALIB=7:  20MM HE PROJ., AIRCRAFT WITH WING TANKS, A KILL.
          C   CALIB=8:  30MM HE PROJ., CLEAN AIRCRAFT, A KILL.
          C   CALIB=9:  30MM HE PROJ., AIRCRAFT WITH WING TANKS, A KILL.
          C   CALIB=10:  40MM HE PROJ., CLEAN AIRCRAFT, A KILL.
          C   CALIB=11:  40MM HE PROJ., AIRCRAFT WITH WING TANKS, A KILL.
          C   CALIB=12:  57MM HE PROJ., CLEAN AIRCRAFT, A KILL.
          C   CALIB=13:  57MM HE PROJ., AIRCRAFT WITH WING TANKS, A KILL.
          C   CALIB=14:  TOTAL PROJECTED AREA OF A FOREIGN AIRCRAFT (FOR HIT PROB. CALC.).
          C   CALIB=15:  GLAADS TARGET: 4 SQ M FRONT & BACK; 24 SQ M SIDES, TOP & BOTTOM.
          C
0004            REAL VEL(10),ALT(10),CRANGE(10),PK(10)
0005            INTEGER *2 CHAR(10),MCHAR(4)
0006            LOGICAL PLOTXY,PLOTXZ
0007            READ (5,11,END=12) NVEL,NCR,NALT,PLOTXY,PLOTXZ,
               . XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX
0008         11 FORMAT (3I5,2L5/(6F10.0))
0009            READ (5,2,END=12) (VEL(I),I=1,NVEL)
0010            READ (5,2,END=12) (CRANGE(I),I=1,NCR)
0011            READ (5,2,END=12) (ALT(I),I=1,NALT)
0012          2 FORMAT (7F10.0)
0013         13 WRITE (1,1)
0014          1 FORMAT ('1G A D E S   I S O - P K   C O N T O U R   P R O G R A M
```

```
                    .---   M A R C H   1 9 7 3')
                C      CALL SUBROUTINE TO READ PARTICULAR GUN SYSTEM DATA.
0015                   CALL COMPK1(&12)
0016                   WRITE (1,8) (I,PK(I),CHAR(I),I=1,NPK)
0017                 8 FORMAT ('1'//,7X,'CONTOUR',8X,'PK',7X,'CHARACTER'/
                      .(' ',I10,F15.4,9XA1))
0018                   WRITE (1,10) MCHAR
0019                10 FORMAT ('0',1X,'MASK',19X,'CHARACTER'/' EL. RATE OR MAX EL',T30,A1
                      ./' AZ. RATE',T30,A1/' INSUF. TIME',T30,A1/' OUT OF RANGE',T30,A1)
                C      START DO LOOPS FOR PLOTTING WITH PARAMETERS OF ALTITUDE,
                C      TARGET VELOCITY AND CROSSING RANGE.
0020                   DO 3 IVEL=1,NVEL
0021                   XDOT=VEL(IVEL)
0022                   YDOT=0.0
0023                   ZDOT=0.0
0024                   XDD=0.
0025                   YDD=0.
0026                   ZDD=0.
                C      TEST IF NO "GROUND PLOT" DESIRED.
0027                   IF (.NOT.PLOTXY) GO TO 24
                C      PLOT FOR "NALT" TARGET ALTITUDES.
0028                   DO 25 IALT=1,NALT
0029                   Z(1)=ALT(IALT)
0030                   CALL PLOT(YMIN,YMAX,Y,16HCROSSING RNG., M)
0031                25 WRITE (1,7) TITLE,XDOT,Z(1)
0032                 7 FORMAT ('0',20A4/
                      .' TARGET VELOCITY=',F8.3,'M/S',5X'TARGET ALTITUDE=',F9.3,'M')
                C      TEST IF NO "SKY PLOT" DESIRED.
0033                24 IF (.NOT.PLOTXZ) GO TO 3
                C      PLOT IN VERTICAL PLANE WITH "NCR" TARGET CROSSING RANGES.
0034                   DO 27 ICR=1,NCR
0035                   Y(1)=CRANGE(ICR)
0036                   CALL PLOT(ZMIN,ZMAX,Z,16HALTITUDE, METERS)
0037                27 WRITE (1,26) TITLE,XDOT,Y(1)
0038                26 FORMAT ('0',20A4/
                      .' TARGET VELOCITY=',F8.3,'M/S',5X'CROSSING RANGE=',F9.3,'M')
0039                 3 CONTINUE
0040                   GO TO 13
0041                12 CALL EXIT
0042                   END
```

```
              C
              C    ======================================================================
 0001                BLOCK DATA
              C    ======================================================================
              C
              C      INITIALIZE PLOTTING CHARACTERS AND PK CONTOUR VALUES.
 0002                COMMON /$PLOT/ PK,CHAR,MCHAR,NPK
 0003                INTEGER NPK/10/
 0004                REAL PK(10)/0.005,0.01,0.05,0.10,0.25,0.5,0.75,0.9,0.95,0.99/
 0005                INTEGER *2 CHAR(10)/'F','G','H','I','J','K','L','M','N','O'/,
                   . MCHAR(4)/'E','A','T','R'/
 0006                END
```

```
      C
      C    ==========================================================
            SUBROUTINE BURPK (XM3,XM4,S1 ,S2 ,S3 ,S4 ,R,XN,AV,BKPS)
      C    ==========================================================
      C
      C
      C     THIS SUBROUTINE COMPUTES BURST KILL PROBABILITY, BPK, USING THE
      C     ANALYTIC MODEL STARTED BY THE UNIVERSITY OF MICHIGAN AND
      C     COMPLETED BY THIS COMMAND.
      C     REFERENCE: "NOTES ON THE UNIVERSITY OF MICHIGAN ANALYTIC GUN
      C     MODEL", TECH NOTE SY-TN3-70, ROBERT C. BANASH, SYSTEMS ANALYSIS
      C     DIRECTORATE, HQ, USA WECOM, APRIL, 1970.
      C
            IMPLICIT REAL*8(A-H,O-Z)
            REAL XM3,XM4,S1,S2,S3,S4,R,XN,AV,BKPS,X(2)
            REAL    SQRT,AMIN1,ABS
                      REAL*8 IIA1,IIA2,IIA3,IIAA1,IIAA2,IIAA3
            DIMENSION ALF(2),P(2,2),BKPJ(300)
            IF( ABS(R).LE.0.05) GO TO 2
            DO 1 J=1,2
            ALF(J) =.5*( S3+S4+(3-2*J)* SQRT((S3+S4)**2-4.*(1.-R**2)*S3*S4))
            X(J) =R* SQRT (S3*S4)/(ALF(J)-S3)
            P(1,J) = SQRT(1./(1./X(J)**2+1.))
            P(2,J) = P(1,J)/X(J)
          1 ALF(J) = 1/ALF(J)
            GO TO 3
          2 ALF(1) = 1./S3
            ALF(2) = 1./S4
            P(1,1) = 1.
            P(1,2) = 0.
            P(2,1) = 0.
            P(2,2) = 1.
          3 BKP = 0.0
            N = XN
            K = 1
            FX = 0.
            C=1.0
            FJM0=1.0
            FNP0=N+1
            PJ=AV/(6.283185*SQRT(S1*S2))
            IF(PJ.GT.1.000) GO TO 100
            IF (XM3.EQ.0..AND.XM4.EQ.0.) K = 2
            A1N= (P(1,1)**2/S1  + P(2,1)**2/S2  ) / ALF(1)
            A2N=(P(1,1)*P(1,2)/S1+P(2,2)*P(2,1)/S2)/DSQRT(ALF(1)*ALF(2))
            A3N= (P(1,2)**2/S1  + P(2,2)**2/S2  )/ALF(2)
            A2N2=A2N*A2N
            DO 29 J=1,N
            NN=J
            FJ = J
            FJ1=1.0/FJ
            DENDIA=DSQRT((A1N+FJ1)*(A3N+FJ1)-A2N2)
            GO TO (4,5),K
          4 A1 = A1N*FJ
```

```
          A2 = A2N*FJ
          A3 = A3N*FJ
          DIA = (A1+1.)*(A3+1.)-A2*A2
          IIA1 = (A3+1.)/ DIA
          IIA2 = -A2/DIA
          IIA3 = (A1+1.)/DIA
          IIAA1 = IIA1*A1+IIA2*A2
          IIAA2 = IIA1*A2+IIA2*A3
          IIAA3 = IIA2*A2+IIA3*A3
          AP1 = A1*(1.-IIAA1)-A2*IIAA2
          AP2 = -A1*IIAA2+A2*(1.-IIAA3)
          AP3 = -A2*IIAA2+A3*(1.-IIAA3)
          V1=DSQRT(ALF(1))*(P(1,1)*XM3+P(2,1)*XM4)
          V2=DSQRT(ALF(2))*(P(1,2)*XM3+P(2,2)*XM4)
          EX =-(V1*V1*AP1+2*V2*V1*AP2+V2*V2*AP3)*0.5
          IF(DABS(EX).GT.170.0D0) EX=DSIGN(170.0D0,EX)
        5 C=-((FNP0-FJ)/FJ)*PJ*(FJM0/FJ)*C
          BKPJ(J)=C*DEXP(EX)/DENDIA
          FJM0=FJ
          IF (DABS(BKPJ(J)).LE.1.0D-10) GO TO 11
       29 CONTINUE
       11 CONTINUE
          DO 12 J=1,NN
          BKP=BKP-BKPJ(NN-J+1)
       12 CONTINUE
          BKPS = DMIN1(1.0D0,BKP)
          RETURN
      100 CONTINUE
          BKPS=1.0
          RETURN
          END
```

```
                C
                C    ==========================================================
 0001                FUNCTION TLU1(X,Y,N,XO)
                C    ==========================================================
                C
                C      THIS SUBROUTINE PERFORMS TABLE LOOK-UP AND LINEAR INTERPOLATION
                C      ON SINGLY-SUBSCRIPTED ARRAYS.
                C
 0002                DIMENSION X(N),Y(N)
 0003                IF (N.LE.1) GO TO 1
 0004                IF (XO.LE.X(1)) GO TO 1
 0005                DO 2 I=2,N
 0006                IF (XO.LE.X(I)) GO TO 3
 0007              2 CONTINUE
 0008                I=N
 0009              3 TLU1=Y(I-1)+(XO-X(I-1))*(Y(I)-Y(I-1))/(X(I)-X(I-1))
 0010                RETURN
 0011              1 TLU1=Y(1)
 0012                RETURN
 0013                END
```

```
      C
      C     ============================================================
              SUBROUTINE DYNGN1
      C     ============================================================
      C
              COMMON DUMY(9),EGDOT,AGDOT
              COMMON/DYNERR/ DYNAM,DYNEM,DYNAV,DYNEV
              DIMENSION DYAZ(10),AZDOT(10),DYEL(10),ELDOT(10)
      C      READ IN DYNAMIC GUN POINTING POSN. ERROR INFORMATION
      C       READ IN TABLES FOR GUN POINTING ERRORS IN AZ. AND EL.
      C
              READ(2,3) DYNAM,DYNEM,MAXAZ,MAXEL
            3 FORMAT(2F10.0,2I5)
              READ (2,2) (ELDOT(I),I=1,MAXEL)
              READ (2,2) (DYEL(I),I=1,MAXEL)
              READ (2,2) (AZDOT(I),I=1,MAXAZ)
              READ (2,2) (DYAZ(I),I=1,MAXAZ)
            2 FORMAT (7F10.0)
      C
      C        WRITE OUT ALL INPUT DATA
      C
              WRITE (1,20) DYNEM,(I,ELDOT(I),DYEL(I),I=1,MAXEL)
           20 FORMAT ('0DYNAMIC GUN-POINTING ERROR FUNCTION - ELEVATION',
             .5X'BIAS, RAD.=',F7.4/
             .' POINT',2X13HEL. RATE, R/S5X23HSTD DEV OF EL. ERROR, R/
             .(' ',I5,F15.3,F20.4))
              WRITE (1,21) DYNAM,(I,AZDOT(I),DYAZ(I),I=1,MAXAZ)
           21 FORMAT ('0DYNAMIC GUN-POINTING ERROR FUNCTION - AZIMUTH',
             .7X'BIAS, RAD.=',F7.4/
             .' POINT',2X13HAZ. RATE, R/S5X23HSTD DEV OF AZ. ERROR, R/
             .(' ',I5,F15.3,F20.4))
              RETURN
              ENTRY DYNGUN
      C      CALCULATE MEAN AND VARIANCE OF DYNAMIC POINTING POSN.
      C
      C       PERFORM TABLE-LOOK-UP TO OBTAIN STANDARD DEVIATIONS OF GUN-
      C       POINTING ERRORS IN AZIMUTH AND ELEVATION.
      C
              SIGDYA=TLU1(AZDOT,DYAZ,MAXAZ,ABS(AGDOT))
              DYNAV = SIGDYA**2
              SIGDYE=TLU1(ELDOT,DYEL,MAXEL,ABS(EGDOT))
              DYNEV=SIGDYE**2
              RETURN
              END
```

```
      C
      C    ============================================================
            SUBROUTINE TFLT(TF,*)
      C    ============================================================
      C
            COMMON/ZZTFLT/ COF(4)
            DIMENSION R(4)
      C
      C    THIS ROUTINE SOLVES THE ANALYTICAL TIME OF FLIGHT EQUATION
      C    THE COEFFICIENTS OF THE FOURTH ORDER TIME OF
      C    FLIGHT EQUATION ARE STORED IN COF(I)
      C    SOLVE THE FOURTH ORDER FOR ITS ROOTS
            CALL QRTIC(COF,R,NRE)
      C
      C    FIND THE SMALLEST REAL ROOT
      C
            TF = 1.0E06
      C
      C    ARE THERE FOUR REAL ROOTS ???
      C
            IF(NRE .EQ. 4) GO TO 5
      C
      C    ARE THERE NO REAL ROOTS ???
      C
            IF(NRE .EQ. 0) GO TO 14
      C
      C    THERE ARE ONLY TWO REAL ROOTS
      C
            GO TO 9
        5   IF(R(4)) 7,7,6
        6   TF = AMIN1(TF,R(4))
        7   IF(R(3)) 9,9,8
        8   TF = AMIN1(TF,R(3))
        9   IF(R(2)) 11,11,10
       10   TF = AMIN1(TF,R(2))
       11   IF(R(1)) 13,13,12
       12   TF = AMIN1(TF,R(1))
       13   IF(TF .GE. 1.0E06) GO TO 14
            RETURN
      C
      C    THER ARE NO REAL ROOTS OR THERE ARE NO POSITIVE REAL ROOTS
      C
       14   RETURN 1
            END
```

```
      C
      C    ============================================================
            SUBROUTINE QRTIC(C,R,NRE)
      C    ============================================================
      C
      C     SOLVES A POLYNOMIAL EQUATION OF THE TYPE
      C       X**4 + C(1)*X**3 + C(2)*X**2 + C(3)*X + C(4) = 0
      C     THE COEFFICIENT OF X**4 IS ASSUMED TO BE 1
      C     R CONTAINS THE ROOTS
      C     NRE CONTAINS THE NUMBER OF REAL ROOTS
      C     IF THERE ARE TWO REAL ROOTS THEY WILL BE IN R(1) AND R(2)
      C       WITH THE COMPLEX ROOTS R(3) +- R(4)*I
      C     IF THERE ARE NO REAL ROOTS, THE COMPLEX ROOTS ARE
      C       R(1) +- R(2)*I AND R(3) +- R(4)*I
      C
            DIMENSION C(4),R(4),CP(3),Y(3)
            C1SQ=C(1)**2
            CP(1)=-C(2)
            CP(2)=C(1)*C(3)-4.*C(4)
            CP(3)=(4.*C(2)-C1SQ)*C(4)-C(3)**2
            CALL CUBIC(CP,Y,NRE)
            A=C1SQ/4.-C(2)+Y(1)
            B=.5*C(1)*Y(1)-C(3)
            D=.25*Y(1)**2-C(4)
            IF(A.GT.0.) GO TO 10
            E=0.
            GO TO 20
         10 E=SQRT(A)
         20 IF(D.GT.0.) GO TO 30
            F=0.
            GO TO 50
         30 F=SIGN(SQRT(D),B)
         50 NRE=0
            REAL=-.25*C(1)+.5*E
            DSCR=REAL**2-.5*Y(1)+F
            RAD=SQRT(ABS(DSCR))
            IF(DSCR.LT.0.) GO TO 60
            NRE=2
            R(1)=REAL+RAD
            R(2)=REAL-RAD
            GO TO 65
         60 R(3)=REAL
            R(4)=RAD
         65 REAL=REAL-E
            DSCR=REAL**2-.5*Y(1)-F
            RAD=SQRT(ABS(DSCR))
            IF(DSCR.LT.0.) GO TO 80
            NRE=NRE+2
            R(NRE)=E
            R(NRE)=REAL-RAD
            R(NRE-1)=REAL+RAD
            RETURN
         80 R(NRE+1)=REAL
```

```
      R(NRE+2)=RAD
      RETURN
      END
```

```
C
C    =============================================================
        SUBROUTINE CUBIC(C,R,NRE)
C    =============================================================
C
C       SOLVES A POLYNOMIAL EQUATION OF THE TYPE
C         X**3 + C(1)*X**2 + C(2)*X + C (3) = 0
C       THE COEFFICIENT OF X**3 IS ASSUMED TO BE 1
C       R CONTAINS THE ROOTS
C       NRE CONTAINS THE NUMBER OF REAL ROOTS
C       IF THERE IS ONE REAL ROOT IT WILL BE R(1),
C       WITH THE COMPLEX ROOTS R(2) +-R(3)*I
C
        DIMENSION C(3),R(3)
        C1SQ=C(1)**2
        P=C(2)-C1SQ/3.
        Q=C(3)-C(1)*(C(2)/3.-2.*C1SQ/27.)
        DEL=4.*P**3+27.*Q**2
        T=C(1)/3.
        IF(DEL.LT.0.) GO TO 10
        SQ=SQRT(DEL/108.)
        HQ=.5*Q
        A=-HQ+SQ
        B=-HQ-SQ
        CRTA=SIGN(ABS(A)**(1.0/3.0),A)
        CRTB=SIGN(ABS(B)**(1.0/3.0),B)
        Y=CRTA+CRTB
        R(1)=Y-T
        R(2)=-.5*Y-T
        R(3)=.866025404*(CRTA-CRTB)
        NRE=1
        RETURN
     10 PHI3=ATAN2(SQRT(-DEL/27.),-Q)/3.
        CON=2.*SQRT(-P/3.)
        R(1)=CON*COS(PHI3)-T
        R(2)=-CON*COS(1.047198-PHI3)-T
        R(3)=-CON*COS(1.047198+PHI3)-T
        NRE=3
        RETURN
        END
```

```
              C
              C    ===========================================================
 0001                SUBROUTINE COMPK1(*)
              C    ===========================================================
              C
              C      THIS SUBROUTINE MANAGES ALL GUN SYSTEM-ORIENTED DATA AND
              C      COMPUTATIONS. THE INITIAL ENTRY POINT READS AND PRINTS SYSTEM
              C      DATA.
 0002                COMMON X,Y,Z,XDOT,YDOT,ZDOT,XDD,YDD,ZDD,EGDOT,AGDOT,RSQ,VP,
                   . TF,RNDS,VO,RMSQ,TREAC,EDMAX,ADMAX,XMIN,XMAX,TITLE(20),CALIB,
                   .ELMAX,ELGUN
 0003                COMMON/ENGPAR/P(9)
 0004                COMMON/DYNERR/DYNAM,DYNEM,DYNAV,DYNEV
 0005                COMMON/BALIST/BARK,BARK2
 0006                COMMON/COVMAT/COV(2,2)
 0007                COMMON/MEANS/ XMEANS(6),AM1,AM2
 0008                REAL POS(6),AVE(7),C(9)
 0009                EQUIVALENCE (X,POS(1)),(X,C(1))
 0010                INTEGER CALIB
              C      READ DESCRIPTION CARD, CALIBER, GUN RESIDUAL DISPERSION PARAMETERS
              C       , MUZZLE VELOCITY, MAX. TRACKING RATES, REACTION TIME,
              C       ROUNDS-PER-BURST AND MAX RANGE
 0011                READ (2,19,END=12) TITLE,CALIB,SIGVO,SIGXD,SIGYD,VO,EDMAX,ADMAX,
                   . TREAC,RNDS,RMAX,ELMAX
 0012             19 FORMAT (20A4/I5/4F10.0/6F10.0)
 0013                WRITE (1,14) TITLE,CALIB,VO,SIGVO,SIGXD,SIGYD,ELMAX,EDMAX,ADMAX,
                   .TREAC,RNDS,RMAX
 0014             14 FORMAT ('0',20A4/'0CALIBER TYPE',T54,I5/
                   .' MUZZLE VELOCITY, M/S',T54,F8.2/
                   .' STD. DEV. OF MUZZLE VELOCITY, M/S',T56,F8.4/
                   .' STD. DEV. OF X-COMP. OF RES. GUN DISP., RAD.',T57,F9.6/
                   .' STD. DEV. OF Y-COMP. OF RES. GUN DISP., RAD.',T57,F9.6/
                   .' MAXIMUM ELEVATION, RAD.',T56,F8.4/
                   .' MAXIMUM ELEVATION RATE, RAD./SEC.',T56,F6.2/
                   .' MAXIMUM AZIMUTH RATE, RAD./SEC.',T56,F6.2/
                   .' AVERAGE SYSTEM REACTION TIME, SEC.',T56,F6.2/
                   .' ROUNDS PER BURST',T56,F4.0/
                   .' MAXIMUM EFFECTIVE RANGE, METERS',T54,F6.0)
              C      READ IN RANGE AND TIME OF FLIGHT USED TO COMPUTE THE BALLISTIC
              C      COEFFICIENT KBAR
              C
 0015                READ (2,20) R,TF
 0016             20 FORMAT(2F10.0)
 0017                BARK=VO/R-1.0/TF
 0018                BARK2=BARK*BARK
              C      WRITE OUT ALL INPUT DATA.
 0019                WRITE(1,22) R,TF,BARK
 0020             22 FORMAT(' RANGE, M. =',F9.4,10X,' TIME OF FLIGHT, SECS. =',F8.4,
                   .10X,' BALLISTIC COEFFICIENT,K-BAR =',F8.4)
 0021                CALL DYNGN1
 0022                CALL MESIG1
              C      CALL INITIALIZATION PORTION OF VULNERABLE-AREA SUBROUTINE.
 0023                CALL VULAPD(CALIB)
```

```
             C     COMPUTE AUXILIARY CONSTANTS USED FOR BURST KILL PROBABILITY.
 0024                RMSQ=RMAX**2
 0025                SIGVO2=SIGVO**2
 0026                SXGD2=SIGAD**2
 0027                SYGD2=SIGYD**2
 0028                RETURN
 0029                ENTRY COMPK(BPK)
             C       THIS ENTRY POINT COMPUTES BURST KILL PROBABILITY, BPK, GIVEN
             C       THE TARGET POSITION.
             C     THE SINGULAR POINT AT THE ORIGIN IS NOT CONSIDERED
 0030                IF(X .NE. 0.0 .OR. Y .NE. 0.0) GO TO 100
 0031                AGDOT = 0.0
 0032                EGDOT = 0.0
 0033                ELGUN = 0.0
 0034                GO TO 1
 0035           100  CONTINUE
 0036                CALL CTOP(C,P)
             C     CALCULATE THE PARTIAL DERIVATIVES
 0037                CALL GFUNC(AG1,EG1,2,&1)
 0038                ELGUN = EG1
 0039                VP=VO/(1.0+BARK*TF)**2
 0040                SQRSQ = SQRT(RSQ)
             C     MODIFY ANY MEANS AND/OR VAR. WHICH ARE FUNCTIONS OF THE
             C     ENGAGEMENT PARAMETERS
 0041                CALL MESIG
             C     PERFORM THE MATRIX MUTIPLICATION NECESSARY TO
             C     CALCULATE THE MEANS AND VARIANCES
 0042                CALL MATMUL
             C     ADD DYNAMIC POINTING ERROR TO THE MEANS AND VAR.
             C     AND CONVERT TO METER MEASUREMENTS IN THE SLANT PLANE
 0043                CALL DYNGUN
 0044                AM1 = (AM1 + DYNAM)*SQRSQ*COS(ELGUN)
 0045                AM2 = (AM2 + DYNEM)*SQRSQ
 0046                S3 = (COV(1,1)+DYNAV)*RSQ*(COS(ELGUN))**2
 0047                S4 = (COV(2,2) + DYNEV)*RSQ
             C     THE CORRELATION COEFFICIENT
 0048                RHO = RSQ*(COS(ELGUN))*COV(1,2)/SQRT(S3*S4)
             C     CALCULATE THE VAR. DUE TO BALISTIC DISPERSION ABOUT THE
             C     FUTURE POSITION OF THE TARGET
 0049                XSAVE = X
 0050                X = X + XDOT*TF
 0051                CALL CTOP(C,P)
 0052                ADOT = P(5)
 0053                EDOT = P(6)
 0054                RVV=RSQ/(VO*VP)
 0055                S1 = ( ADOT*RVV*COS(ELGUN))**2*SIGVO2 + RSQ*SXGD2
 0056                S2=( EDOT*RVV)**2*SIGVO2+RSQ*SYGD2
             C       COMPUTE INDUCED-CORRELATION TERMS - NOTE: RHOB IS NOT NOW USED BUT
             C       IS INCLUDED FOR FUTURE EXPANSION OF BPK.
 0057                RHOB = ADOT*EDOT*COS(ELGUN)*SIGVO2*RVV**2/SQRT(S1*S2)
             C       CALL FOR COMPUTATION OF TARGET VULNERABLE AREA GIVEN TARGET
             C       COORDINATES IN THE "POS" VECTOR.
 0058                CALL VULNAR(POS,VP,AVE)
```

```
0059              X = XSAVE
            C     CALL FOR COMPUTATION OF BURST KILL PROBABILITY, BPK, GIVEN
            C     COMBINED BIAS AND STANDARD DEVIATION TERMS.
0060              CALL BURPK(AM1,AM2,S1,S2,S3,S4,RHO,RNDS,AVE(1),BPK)
0061              RETURN
0062            1 BPK=0.0
0063              RETURN
0064           12 RETURN 1
0065              END
```

```
              C
              C     ==========================================================
 0001                 SUBROUTINE GFUNC(AG,EG,I,*)
              C     ==========================================================
              C
              C     THIS SUBROUTINE CALCULATES THE PARTIAL DERIVATIVES OF THE GUN POSITION
              C     WITH RESPECT TO THE MEASURED QUANTITIES, AND THE AZIMUTH AND
              C     ELEVATION RATES OF THE GUN.
 0002                 COMMON/ZZTFLT/COF(4)
 0003                 COMMON DUMY(9),EGD,AGD,RSQ,DUMY2(1),TFS,DUMY3,VO
 0004                 COMMON/PARIS/F(2,6)
 0005                 COMMON/BALIST/BARK,BARK2
 0006                 COMMON/ENGPAR/R,A,E,RD,AD,ED,RDD,ADD,EDD
 0007                 R2=R*R
 0008                 RD2=RD*RD
 0009                 CE=COS(E)
 0010                 SE=SIN(E)
 0011                 VT2=R2*(AD*AD*CE*CE+ED*ED)+RD2
 0012                 VT=SQRT(VT2)
              C       TIME OF FLIGHT EQUATION LINEAR PREDICTION ASSUMED
              C       EQUATION ASSUMES THE PROJECTILE FOLLOWS THE 3/2 DRAG LAW
 0013                 AA=BARK2*VT2
 0014                 BB=2.0*(R*RD*BARK2+VT2*BARK)
 0015                 CC=R2*BARK2+4.0*R*RD*BARK+VT2-VO*VO
 0016                 DD=2.0*(R2*BARK+R*RD)
 0017                 EE=R2
 0018                 COF(1)=BB/AA
 0019                 COF(2)=CC/AA
 0020                 COF(3)=DD/AA
 0021                 COF(4)=EE/AA
 0022                 CALL TFLT(TF,&1000)
 0023                 TF2=TF*TF
 0024                 RF2=VT2*TF2+R2+2.0*R*RD*TF
 0025                 RF=SQRT(RF2)
 0026                 IF(I .EQ. 3) GO TO 500
              C      CALCULATE THE TERMS FOR THE PARTIAL DERIVATIVES
 0027                 Y=R*CE
 0028                 XD=Y*AD
 0029                 YD=RD*CE-R*ED*SE
 0030                 DEMON1=Y+YD*TF
 0031                 XNUM1=XD*TF
 0032                 XLAMA=ATAN2(-XNUM1,DEMON1)
 0033                 AG=A+XLAMA
 0034                 Z=R*SE
 0035                 ZD=R*ED*CE+RD*SE
 0036                 XNUM2=Z+ZD*TF
 0037                 T3=XNUM2/RF
 0038                 EG=ARSIN(T3)
 0039                 TFS = TF
 0040                 RSQ = RF2
 0041                 T1=-XNUM1/DEMON1
 0042                 T2=1.0/(1.0+T1*T1)
 0043                 PAXD=-T2*TF/DEMON1
```

```
      PAY= T2*T1/DEMON1
      PAYD=PAY*TF
      PATF=-T2*XD*Y/(DEMON1**2)
      PXDR=AD*CE
      PXDAD=Y
      PXDF=-R*AD*SE
      PYR=CE
      PYDR=-ED*SE
      PYDED=-R*SE
      PYE=PYDED
      PYDRD=PYR
      PYDE=-ZD
      T4=1.0/SQRT(1.0-T3*T3)
      PEZ=T4/RF
      PEZD=PEZ*TF
      PETF=PEZ*ZD
      PERF=-PEZ*T3
      PZDE=YD
      PZDR=ED*CE
      PZDRD=SE
      PZDED=Y
      PZR=PZDRD
      PZE=PZDED
      PVTR=R*(AD*AD*CE*CE+ED*ED)/VT
      PVTRD=RD/VT
      PVTAD=R2*AD*CE*CE/VT
      PVTF=-R2*AD*AD*CE*SE/VT
      PVTED=R2*ED/VT
      PRFVT=VT*TF2/RF
      PRFTF=(VT2*TF+R*RD)/RF
      PRFR=(R+TF*RD)/RF
      PRFRD=TF*R/RF
      PAAVT=2.0*BARK2*VT
      PBBR=2.0*RD*BARK2
      PBBRD=2.0*R*BARK2
      PBBVT=4.0*VT*BARK
      PCCR=PBBRD+4.0*RD*BARK
      PCCRD=4.0*R*BARK
      PCCVT=2.0*VT
      PDDR=PCCRD+2.0*RD
      PDDRD=2.0*R
      PEER=PDDRD
      TF4=TF**4
      TF3=TF**3
      DEMON3=4.0*AA*TF3+3.0*BB*TF2+2.0*CC*TF+DD
      PTFR=-(TF3*PBBR+TF2*PCCR+TF*PDDR+PEER)/DEMON3
      PTFRD=-(TF3*PBBRD+TF2*PCCRD+TF*PDDRD)/DEMON3
      PTFVT=-(TF4*PAAVT+TF3*PBBVT+TF2*PCCVT)/DEMON3
      PTFR=PTFR+PTFVT*PVTR
      PTFRD=PTFRD+PTFVT*PVTRD
      PTFAD=PTFVT*PVTAD
      PTFED=PTFVT*PVTED
      PTFE=PTFVT*PVTE
```

```
              PRFR=PRFR+PRFVT*PVTR+PRFTF*PTFR
              PRFRD=PRFRD+PRFVT*PVTRD+PRFTF*PTFRD
              PRFAD=PRFVT*PVTAD+PRFTF*PTFAD
              PRFED=PRFVT*PVTED+PRFTF*PTFED
              PRFE=PRFVT*PVTE+PRFTF*PTFE
      C   CALCULATE THE PARTIAL DERIVATIVES USING THE CHAIN RULE.
              F(1,1) = PAXD*PXDR+PATF*PTFR+PAY*PYR+PAYD*PYDR
              F(1,2) = 1.0
              F(1,3) = PAXD*PXDE+PATF*PTFE+PAY*PYE+PAYD*PYDE
              F(1,4) = PATF*PTFRD + PAYD*PYDRD
              F(1,5) = PAXD*PXDAD + PATF*PTFAD
              F(1,6) = PATF*PTFED + PAYD*PYDED
              F(2,1) = PEZD*PZDR + PETF*PTFR + PERF*PRFR + PEZ*PZR
              F(2,2) = 0.0
              F(2,3) = PEZD*PZDE+PETF*PTFE+PERF*PRFE+PEZ*PZE
              F(2,4) = PEZD*PZDRD+PETF*PTFRD+PERF*PRFRD
              F(2,5) = PETF*PTFAD + PERF*PRFAD
              F(2,6) = PEZD*PZDED + PETF*PTFED + PERF*PRFED
      C    CALCULATE THE RATES OF THE GUN IN AZIMUTH AND ELEV.
              DTFDT = PTFR*RD+PTFRD*RDD+PTFAD*ADD+PTFED*EDD+PTFE*ED
              AGD=AD-(DEMON1*XD*DTFDT-XNUM1*(YD+YD*DTFDT))/(DEMON1**2+XNUM1**2)
              DRFDT = PRFR*RD+PRFRD*RDD+PRFAD*ADD+PRFED*EDD+PRFE*ED
              EGD=T4*(RF*ZD*(1.+DTFDT)-XNUM2*DRFDT )/RF/RF
              RETURN
      C       NO FIRE CONTROL SOLUTION
       1000   RSQ = 1.0E12
              RETURN1
        500   RSQ = RF2
              RETURN1
              END
```

```
                C
                C   ============================================================
0001                 SUBROUTINE CTOP (C,P)
                C   ============================================================
                C
                C    COMPUTE THE POLAR COORDINATES FOR A POINT
                C    LOCATED BY CARTESIAN COORDINATES (C(I))
0002                 DIMENSION C(9),P(9)
0003                 SS = C(1)*C(1) + C(2)*C(2)
0004                 P(1) = SQRT(SS + C(3)*C(3))
0005                 S = SQRT(SS)
0006                 SDOT=(C(1)*C(4)+C(2)*C(5))/S
0007                 P(2) = ATAN2(C(2),C(1))
0008                 IF(P(2) .LT. 0.0 .AND. C(1) .LT.0.0) P(2) = P(2) + 6.28318
0009                 P(3) = ATAN2(C(3),S)
0010                 P(4) = (C(1)*C(4) + C(2)*C(5) + C(3)*C(6))/P(1)
0011                 P(5)=( C(1)*C(5)-C(4)*C(2))/SS
0012                 P(6) = (C(6) -C(3)*P(4)/P(1))/S
0013                 P(7)=(C(4)*C(4)+C(5)*C(5)+C(6)*C(6)+C(1)*C(7)+C(2)*C(8)+C(3)*C(9) 000
                    1 -P(4)*P(4))/P(1)
0014                 P(8) =  (C(8)*C(1) - C(2)*C(7) -2.0*S*SDOT*P(5))/SS
0015                 P(9)=(C(9)-(P(1)*(P(7)*C(3)+P(4)*C(6))-P(4)*P(4)*C(3))/
                    1 (P(1)*P(1))-P(6)*SDOT)/S
0016                 RETURN
0017                 END
```

```
          C
          C    ==========================================================
0001                SUBROUTINE PLOT(BMIN,BMAX,B,LABEL)
          C    ==========================================================
          C
          C         THIS SUBROUTINE PLOTS THE PK CONTOURS AND SYSTEM CONSTRAINTS IN
          C         THE PLANE WITH X AS THE HORIZONTAL AXIS AND B AS THE VERTICAL
          C         AXIS. IF BMIN BMAX AND B HAVE THE VALUES OF THE RESPECTIVE
          C         VARIABLES OF Y, THE PLOT WILL BE A "GROUND PLOT". IF THE DUMMIES
          C         BMIN, BMAX AND B CONTAIN THE ASSOCIATED VALUES FOR Z, THE PLOT
          C         WILL BE IN THE VERTICAL PLANE ("SKY PLOT").
          C
          C         THE PROGRAM WORKS BY INCREMENTING THE DUMMY VARIABLE B BETWEEN
          C         BMIN AND BMAX DURING THE PROCESS OF SWEEPING THE PLOT. THE
          C         VARIABLE IN THE CALLING LIST FOR B WILL BE CHANGED ACCORDINGLY,
          C         THUS ALLOWING FOR THE FREEDOM OF CHOICE TO EITHER SWEEP Y OR Z
          C         MERELY BY PLACING THE VARIABLE NAME IN THE LIST.
          C
0002                COMMON X,Y,Z,XDOT,YDOT,ZDOT,XDD,YDD,ZDD,EGDOT,AGDOT,RSQ,VP,
               . TF,RNDS,VO,RMSQ,TREAC,EDMAX,ADMAX,XMIN,XMAX,TITLE(20),CALIB,
               .ELMAX,ELGUN
0003                COMMON /SPLOT/ PK,CHAR,MCHAR,NPK
0004                COMMON/ENGPAR/P(9)
0005                EQUIVALENCE (C(1),X)
0006                DIMENSION C(9)
0007                REAL AAA(15),PSAVE(101,51),PK(10),LABEL(4),B(1)
0008                INTEGER *2 CHAR(10),MCHAR(4),IPLOT(101,51),BLANK/' '/
0009                INTEGER CALIB
          C
          C         PERFORM INITIALIZATION FOR THE PLOTTING SUBROUTINE "SPLOT".
          C
0010                AAA(1)=XMIN
0011                AAA(2)=XMAX
0012                AAA(3)=BMIN
0013                AAA(4)=BMAX
0014                AAA(5)=-3.0
0015                AAA(6)=-3.0
0016                CALL SPLOT0(IPLOT)
0017                CALL SPLOT1(AAA)
          C
          C         GET THE INCREMENTS FOR THE AXES FROM SPLOT.
          C
0018                DX=AAA(7)
0019                DB=AAA(8)
          C
          C         STORE ZEROS  INTO A MATRIX USED TO STORE PK AT EACH POINT ON GRID.
          C
0020                DO 12 IX=1,101
0021                DO 12 IB=1,51
0022             12 PSAVE(IX,IB)=0.0
          C
          C         INITIALIZE X FOR SWEEP FROM LEFT TO RIGHT.
          C
```

```
                X=XMIN
                DO 1 IX=1,101
      C
      C         INITIALIZE A "PAST" VALUE FOR PK.
      C
                B(1)=BMIN-DB
                CALL COMPK(PKS)
      C
      C         INITIALIZE THE B VARIABLE FOR SWEEP FROM BOTTOM TO TOP.
      C
                B(1)=BMIN
                DO 2 IB=1,51
      C
      C         COMPUTE PK AT THIS POINT AND SAVE.
      C
                CALL COMPK(PKC)
                PSAVE(IX,IB)=PKC
      C
      C         TEST IF TARGET IS OUTSIDE MAXIMIUM EFFECTIVE GUN RANGE.
      C
                IF (RSQ.LE.RMSQ) GO TO 3
      C
      C         STORE AN "R" IF FIRST TIME MAX. RANGE EXCEEDED.
      C
                IF (IB.EQ.1) GO TO 1
                IPLOT(IX,IB)=MCHAR(4)
                GO TO 1
      C
      C         COMPUTE RANGE AT WHICH TARGET WOULD BE IF IT CONTINUED ON ITS
      C         PRESENT COURSE  FOR A TIME EQUAL TO THE AVERAGE SYSTEM REACTION
      C         TIME, TREAC.
      C
              3 XS = X
                X = X + XDOT*TREAC
                IF ( X .EQ. 0.0 .AND. Y .EQ. 0.0 ) GO TO 1000
                CALL CTOP(C,P)
                CALL GFUNC(ADUM,EDUM,3,&1000)
           1000 X = XS
                RNSQ = RSQ
      C
      C         TEST IF NEW RANGE IS BEYOND MAX. SYSTEM RANGE AND PLOT A "T".
      C
                IF (RNSQ.LE.RMSQ) GO TO 4
                IPLOT(IX,IB)=MCHAR(3)
      C
      C         TEST IF EITHER MAX. AZIMUTH RATE OR MAX. ELEVATION OR
      C         ELEVATION RATE ARE EXCEEDED
      C
              4 IF (ABS(AGDOT).LE.ADMAX) GO TO 5
                IPLOT(IX,IB)=MCHAR(2)
                GO TO 8
              5 IF(ABS(ELGUN) .GE. ELMAX) GO TO 2000
                IF (ABS(EGDOT).LE.EDMAX) GO TO 6
```

```
 0049          2000 IPLOT(IX,IB)=MCHAR(1)
 0050               GO TO 8
               C
               C    SET SIGN OF PK GRADIENT TO SS FOR TEST.
               C
 0051             6 SS=SIGN(1.0,PKC-PKS)
               C
               C    SCAN THE PK VECTOR TO SEE IF PKC HAS CROSSED ONE.
               C
 0052               DO 7 IP=1,NPK
 0053               PKD=PK(IP)
 0054               IF (.NOT.(SS*(PKD-PKS).GE.0.0 .AND. SS*(PKD-PKC).LE.0.0)) GO TO 7
               C
               C    IF PKC-PKC STRADDLES PK(IP), PLOT THE APPROPRIATE CHARACTER.
               C
 0055               IPLOT(IX,IB)=CHAR(IP)
 0056               GO TO 8
 0057             7 CONTINUE
               C
               C    INCREMENT B UP BY ONE LINE.
               C
 0058             8 B(1)=B(1)+DB
               C
               C    MAKE PKS EQUAL TO PKC FOR NEXT PASS.
               C
 0059             2 PKS=PKC
               C
               C    INCREMENT X BY ONE TO THE RIGHT.
               C
 0060             1 X=X+DX
               C
               C    THE SCANNING OF THE GRID FROM BOTTOM TO TOP AND LEFT TO RIGHT IS
               C    COMPLETE. NOW SCAN FROM LEFT TO RIGHT AND BOTTOM TO TOP TO FILL
               C    IN THE LINES AS NEEDED.
               C
 0061               DO 9 IB=1,51
 0062               PKS=0.0
 0063               DO 10 IX=1,101
 0064               PKC=PSAVE(IX,IB)
 0065               IF (PKC.EQ.0.0 .OR. PKS.EQ.0.0) GO TO 10
 0066               IF (.NOT.(IPLOT(IX,IB).EQ.BLANK .OR. IPLOT(IX,IB).EQ.MCHAR(3)))
                   .  GO TO 10
 0067               S=SIGN(1.0,PKC-PKS)
 0068               DO 11 IP=1,NPK
 0069               PKD=PK(IP)
 0070               IF (.NOT.(S*(PKD-PKS).GE.0.0 .AND. S*(PKD-PKC).LE.0.0)) GO TO 11
 0071               IPLOT(IX,IB)=CHAR(IP)
 0072               GO TO 10
 0073            11 CONTINUE
 0074            10 PKS=PKC
 0075             9 CONTINUE
               C
               C    PRINT OUT THE "FOOTPRINT"
```

```
              C
 0076               CALL SPLOT3(IPLOT,AAA,20HISO-PK CONTOUR MAP  ,LABEL,
                   , 16HDOWN RANGE, M,  ,1)
 0077               RETURN
 0078               END
```

```
          C
          C    =========================================================
0001             SUBROUTINE SPLOT0(IPLOT)
          C    =========================================================
          C
          C      SPLOT0 INITIALIZES THE 'PICTURE MATRIX' BY CLEARING IT TO BLANKS. 00000020
          C      THE PICTURE MATRIX HAS 101 COLUMNS CORRESPONDING TO THE 101       00000030
          C      HORIZONTAL HAMMER POSITIONS ON THE PRINTER. THE PICTURE MATRIX HAS00000040
          C      51 ROWS CORRESPONDING TO THE VERTICAL SPACING ON THE PRINTER.     00000050
          C
0002             INTEGER *2 IPLOT(101,51),BLANK/' '/                               00000060
0003             DO 1 IY=1,51                                                      00000070
0004             DO 1 IX=1,101                                                     00000080
0005           1 IPLOT(IX,IY)=BLANK                                                00000090
0006             RETURN                                                            00000100
          C
0007             ENTRY SPLOT1(AAA)                                                 00000130
          C
          C      SPLOT1 INITIALIZES THE WORKING PORTION OF THE USER'S AAA-VECTOR.  00000120
          C
0008             REAL AAA(15),EPSLON/5.0E-4/                                       00000140
          C
          C      USER MUST PROVIDE A VECTOR OF LENGTH 15 *4 WORDS.                 00000150
          C      USER PROVIDES PARAMETERS IN FIRST 6 WORDS. THE LAST 9 ARE         00000160
          C      USED BY THE PROGRAM FOR STORAGE OF INTERMEDIATE RESULTS.          00000170
          C       AAA(1)=XMIN
          C       AAA(2)=XMAX
          C       AAA(3)=YMIN
          C       AAA(4)=YMAX
          C       AAA(5)=IXPWR
          C       AAA(6)=IYPWR
          C
0009             AAA(7)=(AAA(2)-AAA(1))/100.                                       00000240
0010             AAA(8)=(AAA(4)-AAA(3))/50.                                        00000250
0011             AAA(15)=0.                                                        00000260
0012             RETURN                                                            00000270
          C
0013             ENTRY SPLOT2(IPLOT,CHAR,AAA,XV,YV)                                00000320
          C
          C      SPLOT2 DETERMINES THE PROPER POSITION IN THE GIVEN PICURE
          C      MATRIX OF THE PAIR (XV,YV), USING THE SCALING INFORMATION
          C      GIVEN IN THE AAA-VECTOR.
          C      IPLOT IS THE PICTURE MATRIX THE (XV,YV) POINT IS TO BE PLOTTED ON.00000330
          C      CHAR IS THE PLOTTING CHARACTER TO BE USED. (INTEGER*2)
          C      AAA IS THE PLOTTING-PARAMETER VECTOR FOR THE FUNCTION BEING
          C      PLOTTED (SEE ABOVE).
          C      XV AND YV ARE THE COORDINATES OF THE POINT TO BE PLOTTED BY SPLOT200000370
          C
0014             INTEGER *2 CHAR                                                   00000390
          C      INCREMENT A POINT COUNTER.
0015             NP=AAA(15)                                                        00000410
0016             NP=NP+1                                                           00000420
0017             AAA(15)=NP                                                        00000430
```

```
C      GET THE DELTA-X AND DELTA-Y FOR THIS FUNCTION.
       DX=AAA(7)
       DY=AAA(8)
C      TEST IF PAST INITIAL START-UP PERIOD OF 3 POINTS.
       IF (NP.GT.3) GO TO 61
C      SAVE THE NEW POINTS.
       AAA(NP+8)=XV
       AAA(NP+11)=YV
C      IF WE NOW HAVE 3 POINTS, START THE PLOT.
       IF (NP.EQ.3) GO TO 60
       RETURN
C      SAVE NEW POINTS AND PUSH OUT OLDEST ONE.
    61 L1=2
       L2=3
       AAA(9)=AAA(10)
       AAA(10)=AAA(11)
       AAA(11)=XV
       AAA(12)=AAA(13)
       AAA(13)=AAA(14)
       AAA(14)=YV
       GO TO 63
    60 L1=1
       L2=2
C      COMPUTE A DIRECTION-SENSITIVE DELTA-X.
    63 DYT=ABS(AAA(L2+11)-AAA(L1+11))
       CORR=DX*DY/AMAX1(DY,DYT)
       CORR=AMAX1(CORR,0.01*DX)
       DIRECT=SIGN(CORR,AAA(L2+8)-AAA(L1+8))
C      INITIALIZE AN X-SCANNER TO TRAVEL BETWEEN POINTS L1 AND L2.
       XSCAN=AAA(L1+8)
       IF (L1.EQ.2) XSCAN=XSCAN+DIRECT
C      COMPUTE AND TEST VALUES USED BY NEWTON INTERPOLATION FORMULA.
       DEN1=AAA(10)-AAA(9)
       DEN2=AAA(11)-AAA(10)
       DEN3=AAA(11)-AAA(9)
       IF (ABS(DEN1).LT.EPSLON*ABS(AAA(10))) GO TO 64
       D1=(AAA(13)-AAA(12))/DEN1
       GO TO 65
    64 D1=0.
    65 IF (ABS(DEN2).LT.EPSLON*ABS(AAA(10))) GO TO 66
       D2=(AAA(14)-AAA(13))/DEN2
       GO TO 67
    66 D2=0.
    67 IF (ABS(DEN3).LT.EPSLON*ABS(AAA(10))) GO TO 68
       DD1=(D2-D1)/DEN3
       GO TO 69
    68 DD1=0.
C      USE NEWTON'S FORMULA TO COMPUTE A Y-VALUE CORRESPONDING TO XSCAN.
    69 IX=1.5+(XSCAN-AAA(1))/DX
C      TEST IF IT IS OFF SCALE.
       IF (IX.LT.1 .OR. IX.GT.101) RETURN
       YSCAN=AAA(12)+(XSCAN-AAA(9))*(D1+(XSCAN-AAA(10))*DD1)
C      COMPUTE CORRESPONDING VERTICAL SPACE POSITION ON PRINTER.
```

```
0060           IY=1.5+(YSCAN-AAA(3))/DY                                      00000970
          C    TEST IF OFF SCALE AND CORRECT.
0061           IF (IY.LT.1) IY=1                                             00000990
0062           IF (IY.GT.51) IY=51                                           00001000
          C    STORE A PLOTTING CHARACTER IN THE DESIRED POSITION.
0063           IPLOT(IX,IY)=CHAR                                             00001020
          C    INCREMENT X-SCANNER AND TEST IF PAST POINT L2.                00001070
0064           XSCAN=XSCAN+DIRECT                                            00001080
0065           IF (DIRECT*(XSCAN-AAA(L2+8)).LE.0.0) GO TO 69                 00001090
          C    IF PAST POINT L2 AND NOT IN INITIAL PERIOD, RETURN.           00001100
0066           IF (L2.EQ.3) RETURN                                           00001110
          C    JUST PLOTTED BETWEEN THE FIRST 2 POINTS, NOW PLOT BETWEEN THE 00001120
          C    2-ND AND 3-RD POINTS.
0067           L1=2                                                          00001140
0068           L2=3                                                          00001150
0069           GO TO 63                                                      00001170
          C
0070           ENTRY SPLOT3(IPLOT,AAA,TITLE,VLAB,HLAB,O)                     00001210
          C
          C    SPLOT3 PRINTS OUT THE PICTURE MATRIX USING THE LABELS AND     00001190
          C    THE SCALING INFORMATION GIVEN.
          C    TITLE IS A 20-CHARACTER HOLLERITH FIELD GIVEN BY THE USER TO LABEL00001220
          C    THE TOP OF THE PICTURE
          C    VLAB IS A 16-CHARACTER HOLLERITH FIELD GIVEN BY THE USER TO LABEL 00001240
          C    THE VERTICAL SCALE.
          C    HLAB IS A 16-CHARACTER HOLLERITH FIELD GIVEN BY THE USER TO LABEL 00001260
          C    THE HORIZONTAL SCALE.
          C    O IS THE OUTPUT DATA SET.
          C
0071           REAL TITLE(5),VLAB(4),HLAB(4),XLAB(11)                        00001290
0072           INTEGER O                                                     00001300
0073           WRITE (O,2) TITLE                                             00001310
0074         2 FORMAT ('1',50X5A4)
0075           BIGDY=10.0**IFIX(AAA(6))                                      00001330
0076           YLAB=AAA(4)*BIGDY                                             00001340
0077           BIGDY=AAA(8)*BIGDY                                            00001350
0078           IP=0                                                          00001360
0079           DO 4 J=1,51                                                   00001370
0080           IY=52-J                                                       00001380
0081           IP=IP+1                                                       00001390
0082           IF (J.NE.23) GO TO 26                                         00001400
0083           WRITE (O,21) VLAB,(IPLOT(IX,IY),IX=1,101)                     00001410
0084           GO TO 31                                                      00001420
0085        26 IF (J.NE.24) GO TO 27                                         00001430
0086           L=AAA(6)                                                      00001440
0087           WRITE (O,22) L,(IPLOT(IX,IY),IX=1,101)                        00001450
0088           GO TO 31                                                      00001460
0089        27 IF (IP)5,6,6                                                  00001470
0090         6 IP=-5                                                         00001480
0091           WRITE (O,7) YLAB,(IPLOT(IX,IY),IX=1,101)                      00001490
0092         7 FORMAT (' ',F16.2,2H *101A1)
0093           GO TO 31                                                      00001510
0094         5 WRITE (O,8) (IPLOT(IX,IY),IX=1,101)                           00001520
```

```
0095            8 FORMAT ('  ',17X'|',101A1)
0096           31 IF (MOD(J,5).EQ.1) GO TO 28                                        00001531
0097              WRITE (0,29)                                                       00001532
0098          29  FORMAT ('+',118X,'|')
0099              GO TO 4                                                            00001534
0100           28 WRITE (0,30)                                                       00001535
0101          30  FORMAT('+',19X, 20(5H    +))
0102              IF( J.EQ. 1 .OR. J .EQ. 51) WRITE(0,50)
0103          50  FORMAT('+',18X,'-',20(5H---- ))
0104            4 YLAB=YLAB-BIGDY                                                    00001540
0105              BIGDX=10.0**IFIX(AAA(5))                                           00001550
0106              XLAB(1)=AAA(1)*BIGDX                                               00001560
0107              TENDX=10.*AAA(7)*BIGDX                                             00001570
0108              DO 9 J=1,10                                                        00001580
0109            9 XLAB(J+1)=XLAB(J)+TENDX                                            00001590
0110              L=AAA(5)                                                           00001600
0111              WRITE (0,10) XLAB,HLAB,L                                           00001610
0112          10  FORMAT(' ',18X,'|',10('....*....|')/' ',9X,11F10.2,/,' ',50X4A4,2X
             1 14H MULT. BY 10**I2)                                                  00001630
0113              RETURN                                                             00001640
0114           21 FORMAT (' ',4A4,1X'|',101A1)
0115           22 FORMAT (' ',14H MULT. BY 10**I2,1X'|',101A1)
0116              END
```

```
      C
      C   =========================================================
            SUBROUTINE MATMUL
      C   =========================================================
      C
      C    THIS SUBROUTINE PERFORMS THE MATRIX MULTIPLICATIOS NECESSARY
      C    TO CALCULATE THE MEANS AND VAR. OF THE GUN ERROR
            COMMON/COVMAT/ COV(2,2)
            COMMON/PARTS/ F(2,6)
            COMMON/ SIGMA/ SIG(6,6)
            COMMON/MEANS/ XM(6),AM(2)
            DIMENSION TEMP(2,6)
            DO 10 I=1,2
            AM(I)=0.
            DO 10 J=1,6
      C    CALCULATE THE MEANS
            AM(I) = AM(I) + F(I,J)*XM(J)
            SUM = 0.0
            DO 5 K=1,6
        5   SUM = SUM + F(I,K)*SIG(K,J)
       10   TEMP(I,J) = SUM
            DO 20 I=1,2
            DO 20 J=I,2
            SUM = 0.0
            DO 15 K=1,6
       15   SUM = SUM + TEMP(I,K)*F(J,K)
      C    CALCULATE THE VAR.
            COV(I,J) = SUM
       20   COV(J,I) = SUM
            RETURN
            END
```

```
                C
                C    ==========================================================
0001                   SUBROUTINE MESIG1
                C    ==========================================================
                C
0002                   COMMON/MEANS/ XM(6),AMDUM(2)
0003                   COMMON/SIGMA/ SIG(6,6)
                C    READ THE GUN SENSOR ERROR DATA
                C    THE ORDER OF THE VARIBALES IN THE SENSOR MEAN MEASUREMENT
                C    ERROR VECTOR (XM(I)) IS :
                C       * RANGE
                C       * AZIMUTH
                C       * ELEVATION
                C       * RANGE RATE
                C       * AZIMUTH RATE
                C       * ELEVATION RATE
0004                   M = 2
0005                   READ(M,1000) (XM(I),I=1,6)
                C    THE INPUT SIG(I,J) ARRAY CONTAINS THE STANDARD
                C    DEVIATIONS IN THE DIAGONAL TERMS AND THE CORRELATION
                C    COEFFICIENTS IN THE OFF DIAGONAL TERMS.  THE ORDER OF THE ROWS OF
                C    THE MATRIX (AND THE COLS.) CORRESPOND TO THE FOLLOWING ORDER OF THE
                C    ENGAGEMENT PARAMETERS:
                C          1 ==> RANGE
                C          2 ==> AZIMUTH
                C          3 ==> ELEVATION
                C          4 ==> RANGE RATE
                C          5 ==> AZ. RATE
                C          6 ==> ELEV. RATE
                C    THE SIG MATRIX WILL BE A SYMETRICAL MATRIX AND IT
                C    IS READ ONE ROW AT A TIME.  THE INPUT DATA SHOULD BE SET
                C    UP WITH ONE ROW OF THE MATRIX ON A CARD.
0006                   READ(M,1000) ((SIG(J,I),I=1,6),J=1,6)
0007              1000 FORMAT(6F10.0)
0008                   WRITE (1,100)
0009               100 FORMAT(' SENSOR ERRORS'/' MEAN MEASURENT ERRORS XM(I)'/)
0010                   WRITE (1,200) (XM(I),I=1,6)
0011               200 FORMAT(6X,'RANGE,M',T30,F8.5/6X,'AZIMUTH,RAD',T30,F8.5/
                      .6X,'ELEVATION,RAD',T30,F8.5/6X,'RANGE RATE,M/S',T30,F8.5/
                      .6X,'AZ. RATE,RAD/S',T30,F8.5/6X,'EL. RATE,RAD/S',T30,F8.5/)
0012                   WRITE (1,300)
0013               300 FORMAT(' MATRIX OF STANDARD DEVIATIONS AND CORRELATION COEFFICIENT
                      .S'/
                      .' THE UNITS ARE THE SAME AS FOR THE MEANS'//
                      .16X,'RANGE',6X,'AZIMUTH',7X,'ELEV.',7X,'R. RATE',6X,'AZ. RATE',
                      .5X,'EL. RATE'/ )
0014                   WRITE (1,400) ((SIG(J,I),I=1,6),J=1,6)
0015               400 FORMAT(' RANGE',6X,F8.2,6X,F8.3,4(5X,F8.3)/' AZIMUTH ',6(4X,F9.4)/
                      .' ELEVATION',3X,F9.4,5(4X,F9.4)/' R. RATE',6(5X,F8.3)/
                      .' AZ. RATE',4X,F9.4,5(4X,F9.4)/' EL. RATE',4 ,F9.4,5(4X,F9.4))
                C    CHANGE THE STD. DEV. INTO VAR.
0016                   DO 5 I=1,5
0017                   K = I + 1
```

```
      DO 5 J =K,6
      SIG(I,J) = SIG(I,J)*SIG(I,I)*SIG(J,J)
      SIG(J,I) = SIG(I,J)
    5 CONTINUE
      DO 6 I=1,6
    6 SIG(I,I) = SIG(I,I)**2
      RETURN
      END
```

```
              C
              C    ==========================================================
0001                 SUBROUTINE  VULAPD(CALIB)
              C    ==========================================================
              C
              C      THIS SUBROUTINE COMPUTES TARGET VULNERABLE AREA FOR POINT-CONTACT
              C      PROJECTILES.
              C      VULNERABLE AREA IS A FUNCTION OF TARGET TYPE, SHELL CALIBER,
              C      STRIKING VELOCITY AND STRIKING DIRECTION.
              C      THIS ENTRY POINT SERVES TO SELECT THE DATA FOR THE SHELL CALIBER
              C      BY "CALIB".
              C
0002                 IMPLICIT INTEGER (E)                                            00000200
              C
              C    CALIB=1:  20MM HE PROJ., KK KILL.
              C    CALIB=2:  30MM HE PROJ., KK KILL.
              C    CALIB=3:  40MM HE PROJ., KK KILL.
              C    CALIB=4:  57MM HE PROJ., KK KILL.
              C    CALIB=5:  8 LB WARHEAD, KK KILL.
              C    CALIB=6:  20MM HE PROJ., CLEAN AIRCRAFT, A KILL.
              C    CALIB=7:  20MM HE PROJ., AIRCRAFT WITH WING TANKS, A KILL.
              C    CALIB=8:  30MM HE PROJ., CLEAN AIRCRAFT, A KILL.
              C    CALIB=9:  30MM HE PROJ., AIRCRAFT WITH WING TANKS, A KILL.
              C    CALIB=10:  40MM HE PROJ., CLEAN AIRCRAFT, A KILL.
              C    CALIB=11:  40MM HE PROJ., AIRCRAFT WITH WING TANKS, A KILL.
              C    CALIB=12:  57MM HE PROJ., CLEAN AIRCRAFT, A KILL.
              C    CALIB=13:  57MM HE PROJ., AIRCRAFT WITH WING TANKS, A KILL.
              C    CALIB=14:  TOTAL PROJECTED AREA OF A FOREIGN AIRCRAFT (FOR HIT PROB. CALC.).
              C    CALIB=15:  GLAADS TARGET: 4 SQ M FRONT & BACK; 24 SQ M SIDES, TOP & BOTTOM.
              C
0003                 INTEGER CALIB,SYSIN
0004                 DIMENSION        TV(3),SV(3),V(3),VR(3),RSULT(3,3),FR(3),B(3),
                    1S(7,6,7),AVR(10)
              C
              C      COMPUTE DATA SET NUMBER AND READ DATA FROM DISK.
              C
0005                 SYSIN =10+CALIB
0006                 REWIND SYSIN
0007                 READ (SYSIN,801) EMAX,(((S(J,K,L),J=1,3),K=1,6),L=1,EMAX)
0008             801 FORMAT(I5/(6F10.5))
              C
              C      EMAX EQUALS THE NUMBER OF VULNERABLE COMPONENTS
              C      K=1: FRONT; K=2: REAR; K=3: PORT; K=4: STARBOARD; K=5: BOTTOM;
              C      K=6: TOP.
              C      THE AREAS IN THE "S" MATRIX ARE IN SQ METERS
              C      CORRESPONDING VELOCITIES FOR VARIOUS LEVELS OF J ARE IN METERS/S
              C
0009                 RETURN
              C
0010                 ENTRY VULNAR(POS,VP,AV)
0011                 REAL POS(6),AV(7)
0012                 DIMENSION  IN(3),IM(3)
0013                 R = SQRT (POS(1)*POS(1)+POS(2)*POS(2)+POS(3)*POS(3) )
```

```
              GP= SQRT (POS(1)*POS(1)+POS(2)*POS(2))
              CE = GP/R
              SE = POS(3)/R
              SA = POS(2)/GP
              CA = POS(1)/GP
              SV(1) = VP*CE*CA
              SV(2) = VP*CE*SA
              SV(3) = VP*SE
              VR(1) = SV(1)-POS(4)
              VR(2) = SV(2)
              VR(3) = SV(3)
              RT = SQRT( VR(1)*VR(1)+VR(2)*VR(2)+VR(3)*VR(3) )
              DO 14 I = 1,3
           13 B(I) = ABS (VR(I)/RT)
              AVR(I) = ABS( VR(I) )
           14 IM(I) = 2*I-1-( SIGN(1.0,VR(I) )-1.)/2.
              DO 30 E = 1,EMAX
              AV(E) = 0.
              DO 30 I = 1,3
        C
        C     THIS EQUATION PERFORMS LINEAR INTERPLOATION ON VULNERABLE AREAS
        C     S(2,IM(I),E) AND S(1,IM(I),E) FOR STRIKING VELOCITIES OF V2 AND V1
        C     RESPECTIVELY.  V2=304.8 M/S (1000 FT/SEC); V1=152.4 M/S (500 FT/SEC).
        C
              ATP=(S(2,IM(I),E)-S(1,IM(I),E))*(AVR(I)-152.4)/152.4+S(1,IM(I),E)
              ATP = AMAX1(ATP,0.)
              ATP = AMIN1(ATP,S(2,IM(I),E))
        C
        C     NOTE THAT LINEAR INTERPLOLATION IS USED TO COMPUTE VULNERABLE
        C     AREAS. WE DO NOT ALLOW AN AREA SMALLER THAN THE SMALLEST VALUE
        C     GIVEN OR A VALUE LARGER THAN THE LARGEST VALUE GIVEN.            00005980
        C
           30 AV(E)=B(I)*ATP+AV(E)
              RETURN
              END
```

SAMPLE PROBLEM

The following sample problem illustrates the use of the ISO-PK program for the analysis of a hypothetical air defense gun system being used against a hypothetical target aircraft.

Next page is blank.

```
//FT01F001 DD SYSOUT=A,DCB=RECFM=UA  OUTPUT DATA SET FOR FOOTPRINTS
//DATA DD *
    3    4    3    T    F
-2000.    4000.    0.        5000.     0.        5000.                 FPC/2
-128.61111-180.05555-231.50000
250.      450.     850.      1200.                                     FPC/4
250.      450.     800.                                                FPC/5
$STOP
//FT02F001 DD *  GUN DATA SET
 ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
    1
      12.       .003      .003    1014.
      .60       .60       0.0       30.0      1750.0     1.48
     1500.     2.834
0.0030    0.0035        4    4
0.        0.060     0.300     0.600
     .004      .0047     .0058     .0072
0.        0.060     0.300     0.600
     .004      .0047     .0058     .0072
      -5.       0.        0.        0.        0.        0.
5.0
          .0025
                    .0025
                              5.0
                                        .0025
                                                  .0025
//FT11F001 DD *
    1                                                                  2022110
.17       .33       .33       .06       .12       .12                  2022120
1.27      2.54      2.54      1.27      2.54      2.54                 2022130
1.11      2.23      2.23      .8        1.6       1.6                  2022140
```

ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2

CALIBER TYPE	1
MUZZLE VELOCITY, M/S	1014.00
STD. DEV. OF MUZZLE VELOCITY, M/S	12.0000
STD. DEV. OF X-COMP. OF RES. GUN DISP., RAD.	0.003000
STD. DEV. OF Y-COMP. OF RES. GUN DISP., RAD.	0.003000
MAXIMUM ELEVATION, RAD.	1.4800
MAXIMUM ELEVATION RATE, RAD./SEC.	0.60
MAXIMUM AZIMUTH RATE, RAD./SEC.	0.60
AVERAGE SYSTEM REACTION TIME, SEC.	0.0
ROUNDS PER BURST	30.
MAXIMUM EFFECTIVE RANGE, METERS	1750.

RANGE, M. =1500.0000 TIME OF FLIGHT, SECS. = 2.8340 BALLISTIC COEFFICIENT,K-BAR = 0.3231

DYNAMIC GUN-POINTING ERROR FUNCTION - ELEVATION BIAS, RAD.= 0.0035

POINT	EL. RATE, R/S	STD DEV OF EL. ERROR, R
1	0.0	0.0040
2	0.060	0.0047
3	0.300	0.0058
4	0.600	0.0072

DYNAMIC GUN-POINTING ERROR FUNCTION - AZIMUTH BIAS, RAD.= 0.0030

POINT	AZ. RATE, R/S	STD DEV OF AZ. ERROR, R
1	0.0	0.0040
2	0.060	0.0047
3	0.300	0.0058
4	0.600	0.0072

SENSOR ERRORS

MEAN MEASURENT ERRORS XM(I)

RANGE,M	-5.00000
AZIMUTH,RAD	0.0
ELEVATION,RAD	0.0
RANGE RATE,M/S	0.0
AZ. RATE,RAD/S	0.0
EL. RATE,RAD/S	0.0

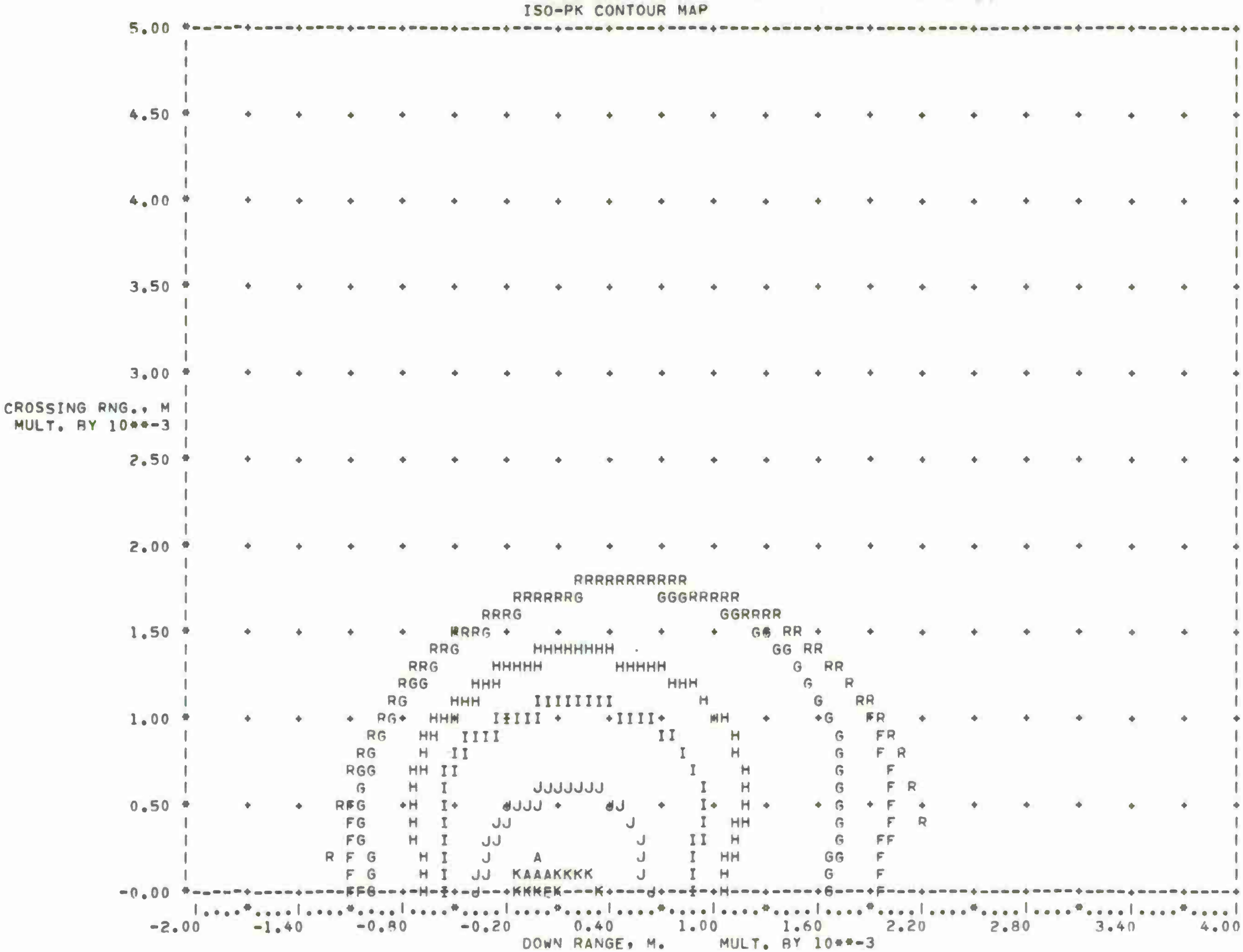
MATRIX OF STANDARD DEVIATIONS AND CORRELATION COEFFICIENTS

THE UNITS ARE THE SAME AS FOR THE MEANS

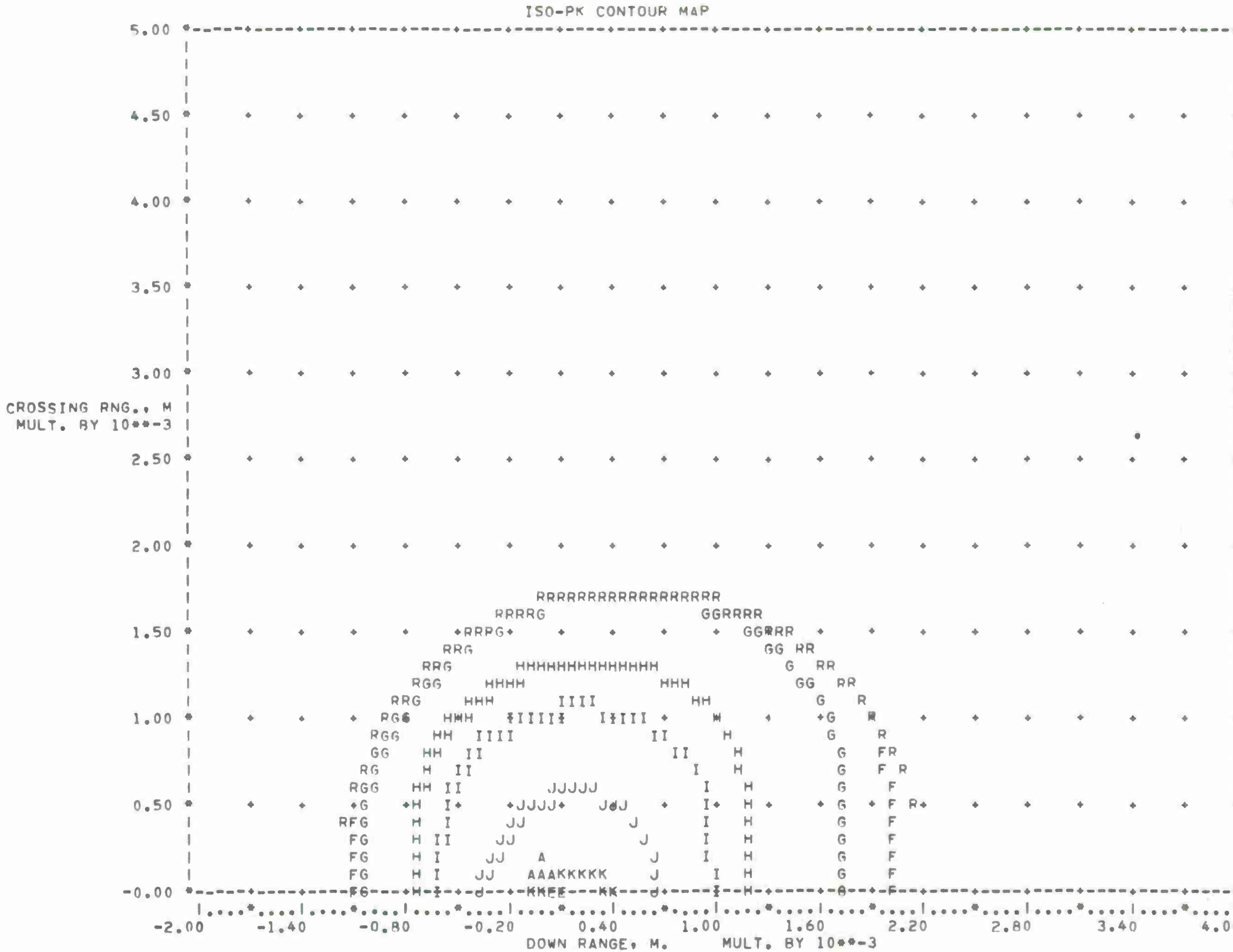
	RANGE	AZIMUTH	ELEV.	R. RATE	AZ. RATE	EL. RATE
RANGE	5.00	0.0	0.0	0.0	0.0	0.0
AZIMUTH	0.0	0.0025	0.0	0.0	0.0	0.0
ELEVATION	0.0	0.0	0.0025	0.0	0.0	0.0
R. RATE	0.0	0.0	0.0	5.000	0.0	0.0
AZ. RATE	0.0	0.0	0.0	0.0	0.0025	0.0
EL. RATE	0.0	0.0	0.0	0.0	0.0	0.0025

CONTOUR	PK	CHARACTER
1	0.0050	F
2	0.0100	G
3	0.0500	H
4	0.1000	I
5	0.2500	J
6	0.5000	K
7	0.7500	L
8	0.9000	M
9	0.9500	N
10	0.9900	O

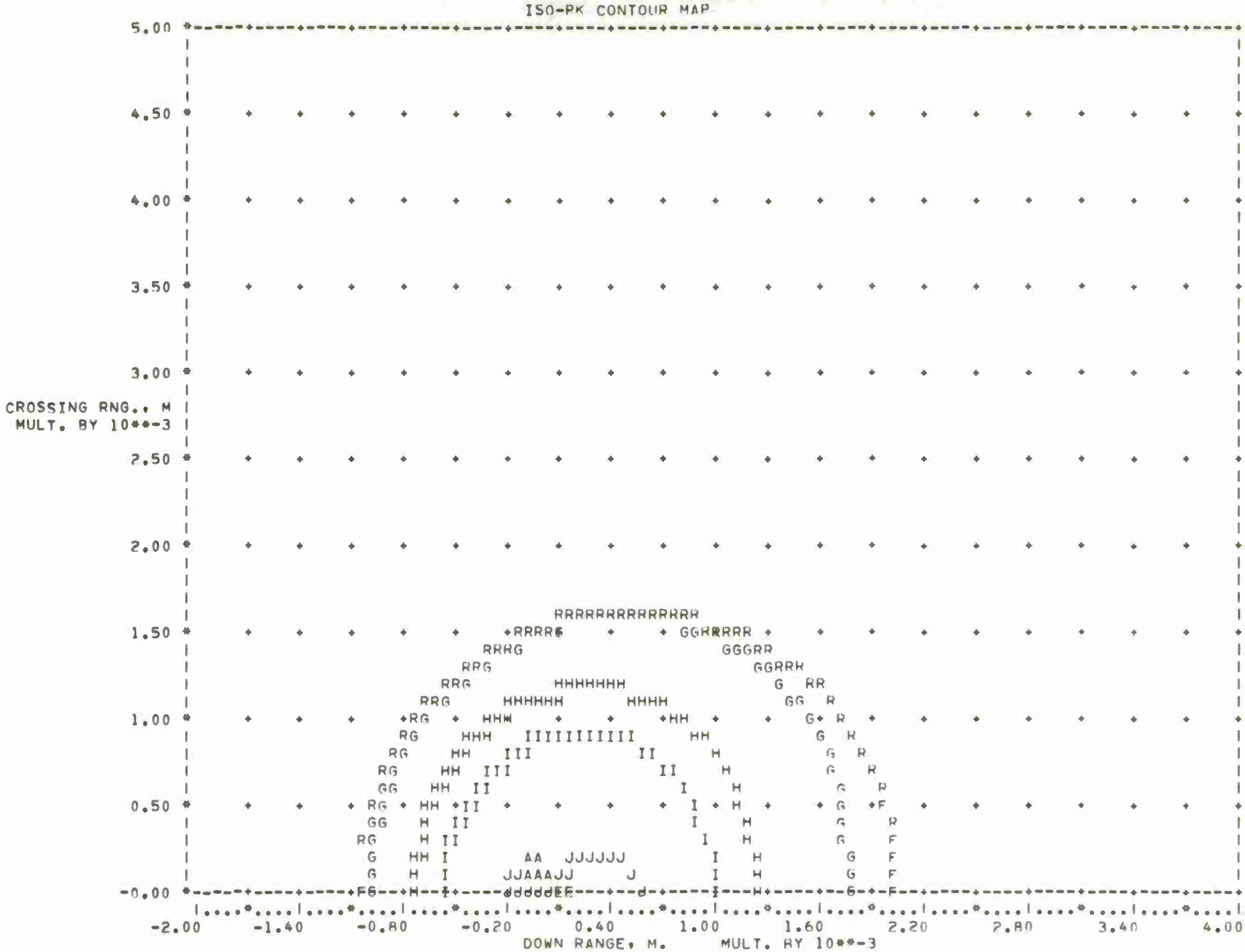
MASK	CHARACTER
EL. RATE OR MAX EL	F
AZ. RATE	A
INSUF. TIME	T
OUT OF RANGE	R



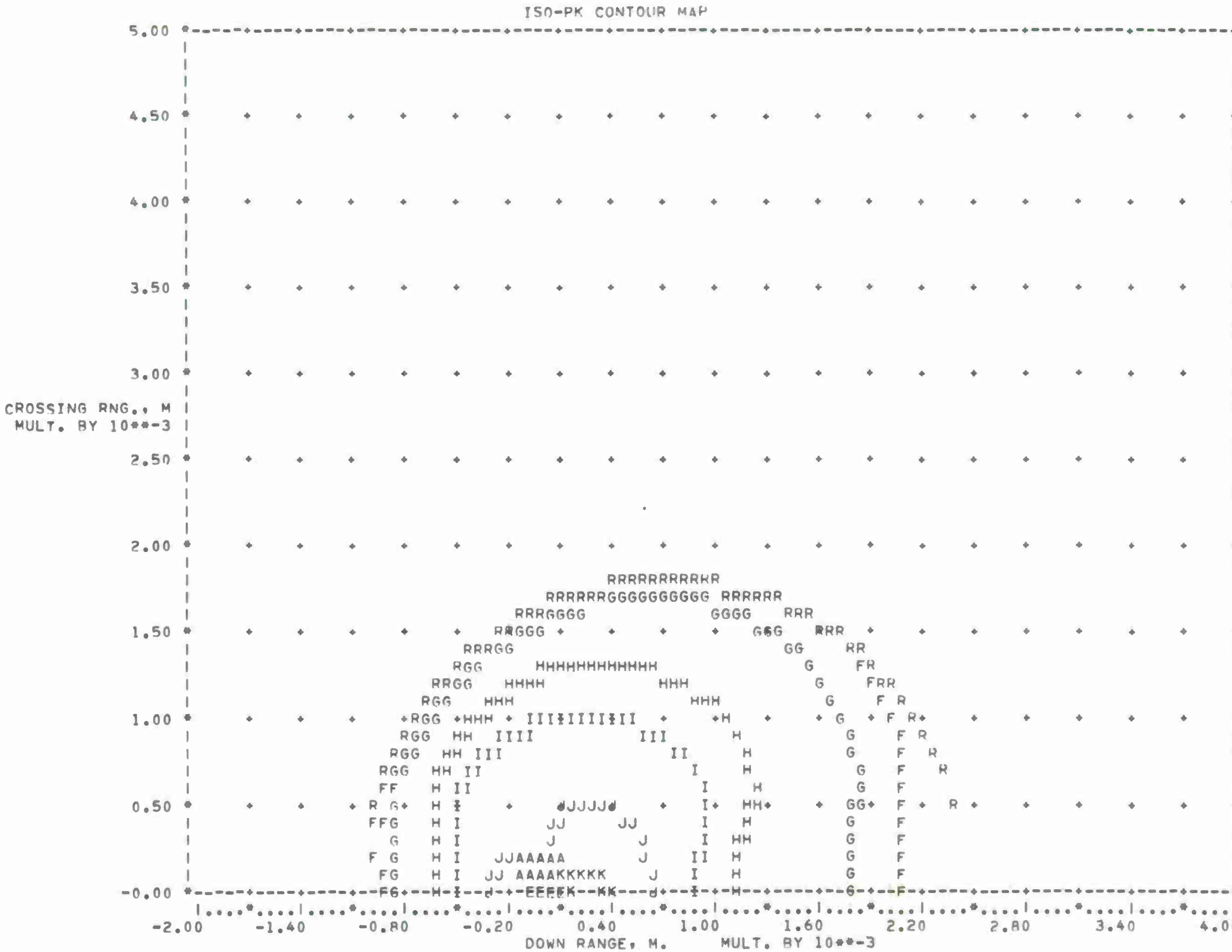
ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
TARGET VELOCITY=-128.611M/S TARGET ALTITUDE= 250.000M



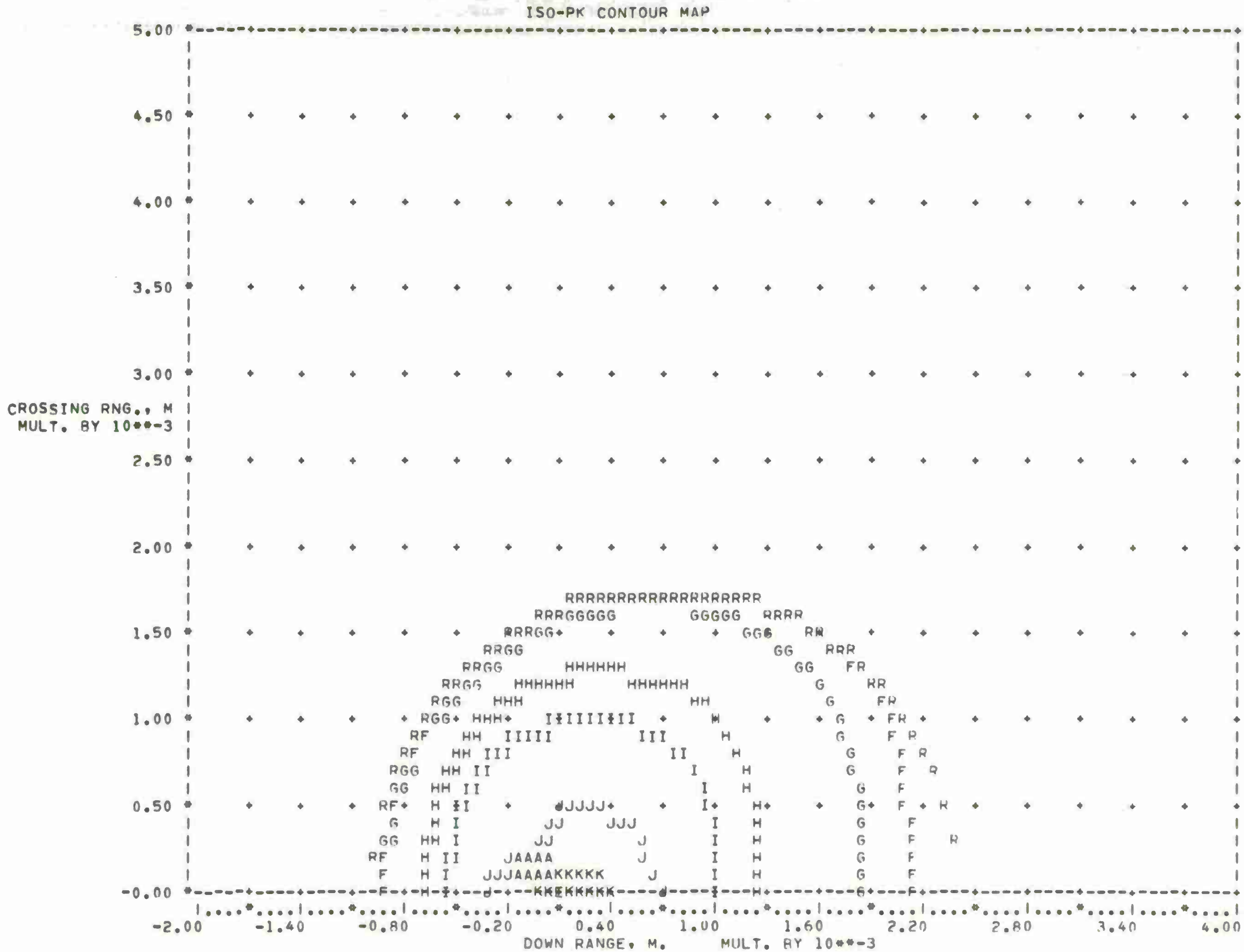
ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
TARGET VELOCITY=-128.611M/S TARGET ALTITUDE= 450.000M



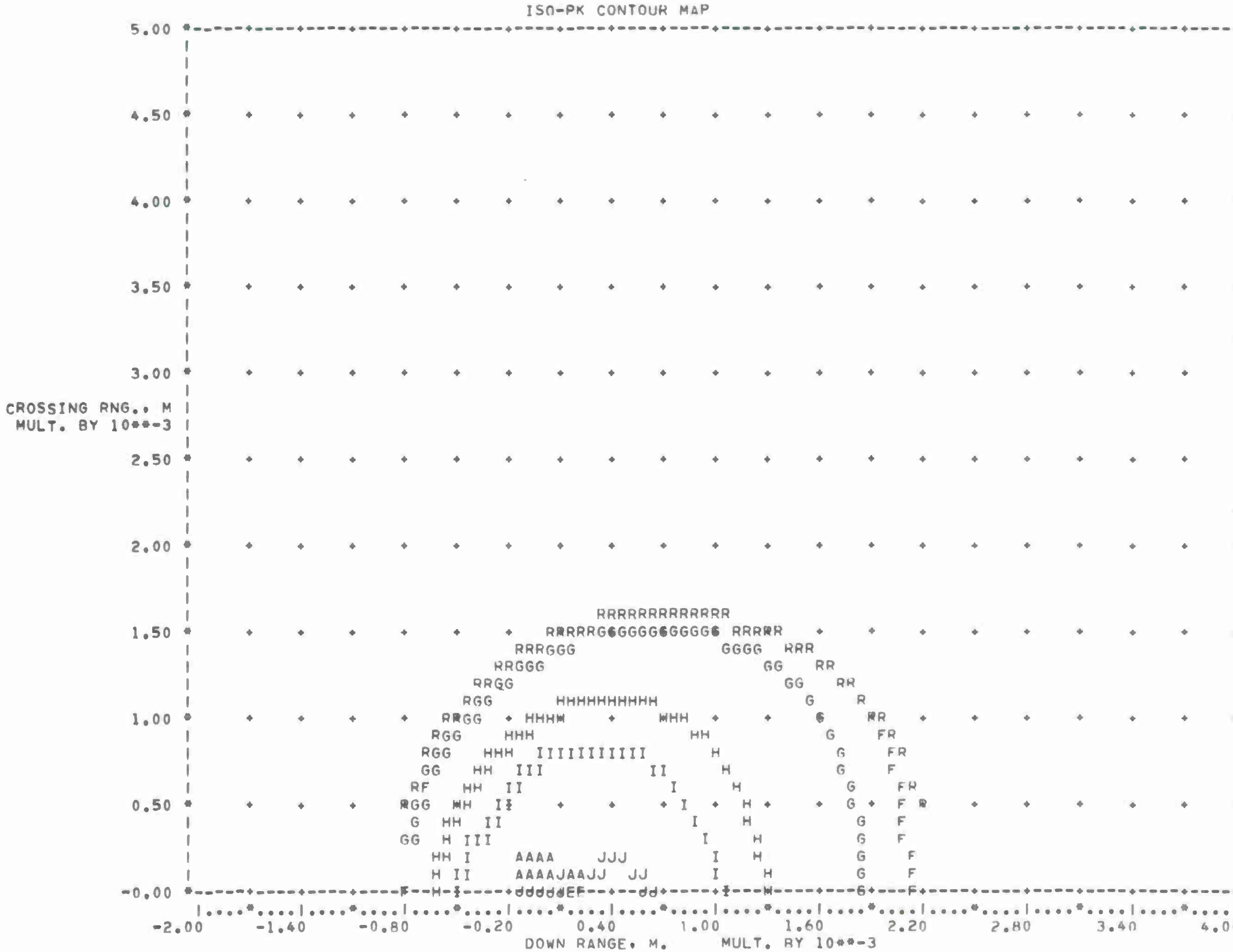
ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
TARGET VELOCITY=-128.611M/S   TARGET ALTITUDE= 800.000M



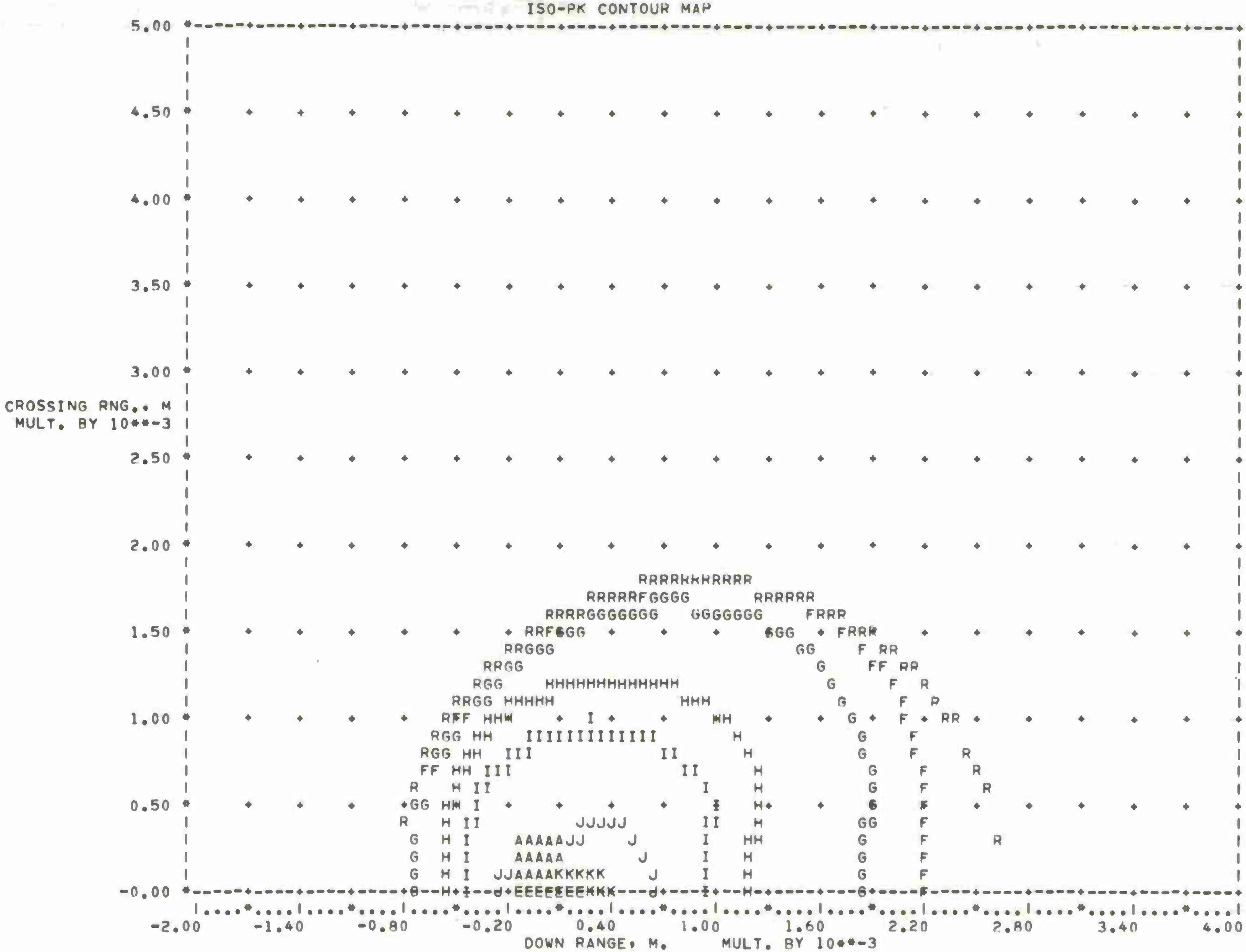
ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
TARGET VELOCITY=-180.056M/S TARGET ALTITUDE= 250.000M



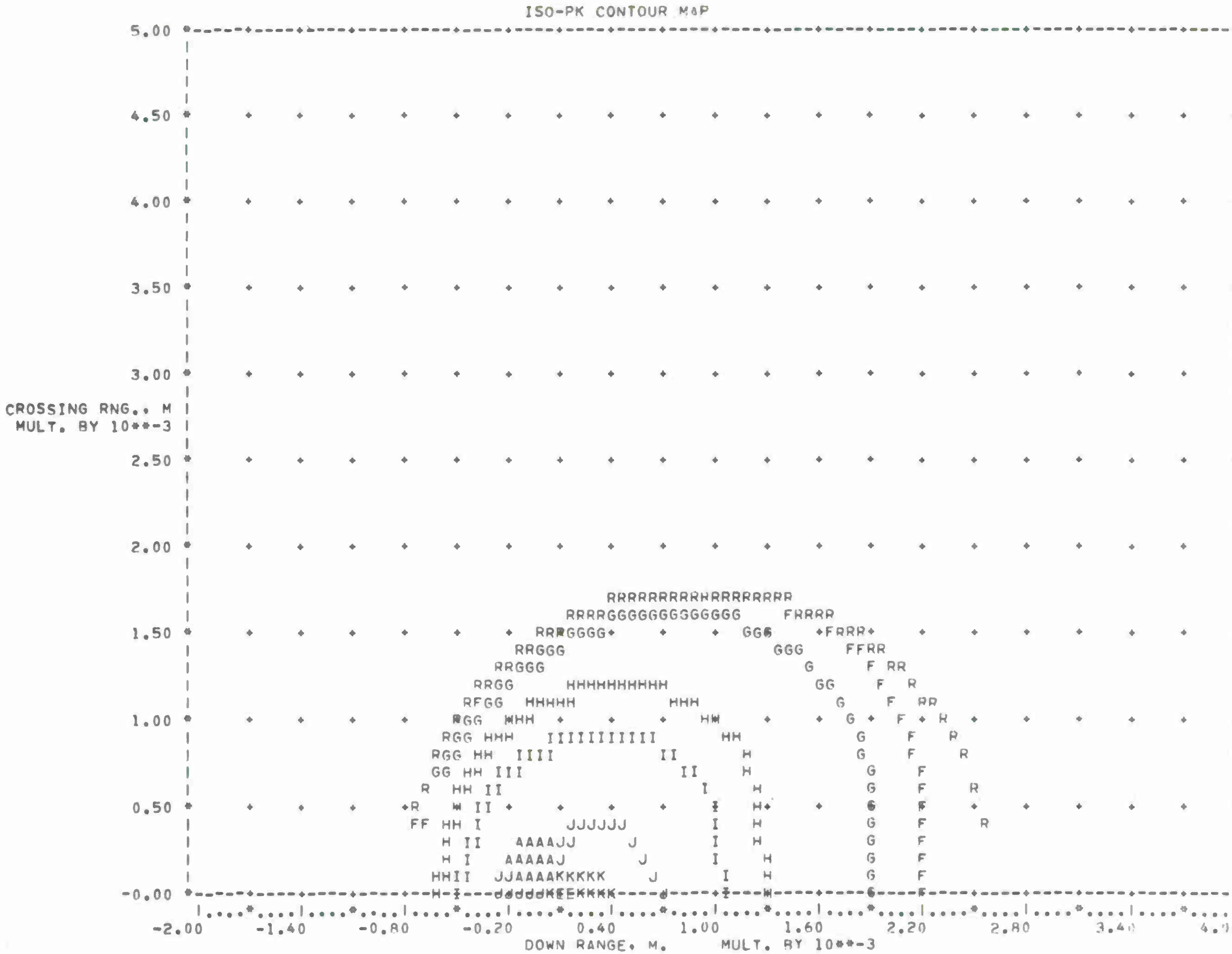
ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI.1/2
TARGET VELOCITY=-180.056M/S   TARGET ALTITUDE= 450.000M



ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
TARGET VELOCITY=-180.056M/S   TARGET ALTITUDE= 800.000M



ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
TARGET VELOCITY=-231.500M/S   TARGET ALTITUDE= 250.000M



ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
TARGET VELOCITY=-231.500M/S TARGET ALTITUDE= 450.000M

ISO-PK SENSITIVITY ANALYSIS:EXPERIMENTI,1/2
TARGET VELOCITY=-231.500M/S   TARGET ALTITUDE= 800.000M

REFERENCES

1. WECOM ADEB Task Force, Analysis of the Air Defense Gun Systems, Technical Note RE-TR-70-191, Research and Engineering Directorate, HQ, US Army Weapons Command, Rock Island, IL, October 1970.

2. Banash, Robert C., An Analytic Procedure for the Computation of Burst-Kill Probabilities in Air Defense, Technical Note SY-TN9-70, Systems Analysis Directorate, HQ, US Army Weapons Command, Rock Island, IL, October 1970.

3. Banash, Robert C., Notes of the University of Michigan Analytic Gun Model, Technical Note SY-TN3-70, Systems Analysis Directorate, HQ, US Army Weapons Command, Rock Island, IL, April 1970.

4. Banash, Robert C., A Markov Chain Approach to Modeling Tracking Error in Air Defense Gun Evaluation, Technical Report No. (GADES 3-1), Artillery and Air Defense Weapons Directorate, Rock Island Weapons Laboratory, Rock Island, IL, September 1971.

Next page is blank.

APPENDIX

DERIVATION OF THE GUN POSITION EQUATION

Next page is blank.

INTRODUCTION

The expressions used to define the position of the gun necessary to score a hit on the target are derived in this appendix. The equations are based upon linear prediction methods and are, therefore, correct only if the target is flying at a constant speed. The expressions must, of necessity, be in terms of the measured quantities (a, e, r, $\dot{a}$, $\dot{e}$, $\dot{r}$) (see Figure 1, p. 7) at the present position of the target; for only then can the necessary partial derivations be taken.

Azimuth Angle of the Gun

The azimuth position of the gun (and target) is a rather artificial quantity since there usually is no reference on the gun from which it can be measured. However, it is a required mathematical necessity. It will be shown that the actual measurement of an azimuth angle is not required in determining the lead angle of the gun. The position of the gun can be expressed as:

$$A_G = a + \lambda_A$$

where

A_G = azimuth angle of the gun

λ_A = azimuth lead angle measured in the ground plane

Mathematically, A_G and "a" are both measures from the same arbitrary reference.

The lead angle λ_A is defined: Consider the projection of the path of the target into the ground plane as shown in Figure A-1. The $\overline{XY}$ plane defines the arbitrary coordinate system in which the azimuth angle, a, is measured. The xy plane is not arbitrary; however, it is set up so that the y-axis coincides with the reference position on the gun from which the lead angle is measured. This axis also lines up with the measured position of the target (T(t)). The velocity vector v to signify the path of the target, and the position of the target at the time of predicted intercept (T(t+Δt)) is shown in Figure A-1.

In the xy plane and in terms of the measured quantities,

$$\dot{x} = r\dot{a} \cos(e)$$

$$y_o = r \cos(e)$$

$$\dot{y} = \dot{r} \cos(e) - r\dot{e} \sin(e)$$

The lead angle λ_A can be expressed as

$$\lambda_A = \tan^{-1} \left[\frac{-x(t + \Delta t)}{y(t + \Delta t}\right] = \tan^{-1}\left[\frac{-\dot{x}\ \Delta\ t}{y_o + \dot{y}\ \Delta\ t}\right]$$

Δt is time of flight of the projectile to the point of intercept.

It is more convenient to leave the expression for the lead angle in this form. Note that all of the terms in this equation (except Δt) are functions of the measured quantities. Δt will be treated later, and will be shown to be a function of the measured quantities.

The azimuth angle of the gun is defined as

$$A_G = a + \tan^{-1} \left[\frac{-\dot{x}\Delta t}{y_0 + y\Delta t}\right]$$

The various partial derivatives of this expression with respect to the measured quantities can be obtained by application of the chain rule for partial differentation. Note that the lead angle calculation is independent of the azimuth angle measurement, but that the gun position can be wrong if there is a measurement error involved with the azimuth angle a. This error can be likened to a tracking error in the azimuth direction.

Elevation Angle of the Gun

The elevation angle of the gun is shown in Figure A-2. The ω-axis in this figure coincides with the one shown in Figure A-1. The expression for the elevation of the gun (E_G) is given by

$$E_G = \sin^{-1}\left[\frac{z(t + \Delta t)}{R_f}\right]$$

Where

$z(t + \Delta t)$ = the altitude of the target at time of predicted intercept,

R_f = the range to the target at time of predicted intercept

In terms of the measured quantities:

$$z(t + \Delta t) = z_0 + \dot{z}\Delta t$$

Where

$$z_0 = r \sin(e)$$

$$\dot{z} = r\dot{e} \cos(e) + \dot{r} \sin(e)$$

and (referring to Figure A-1)

$$R_f^2 = (\dot{x}\Delta t)^2 + (y_0 + \dot{y}\Delta t)^2 + (z_0 + \dot{z}\Delta t)^2$$

$$= (\dot{x}^2 + \dot{y}^2 + \dot{z}^2)\ \Delta t^2 + 2\ r\dot{r}\ \Delta t + r^2$$

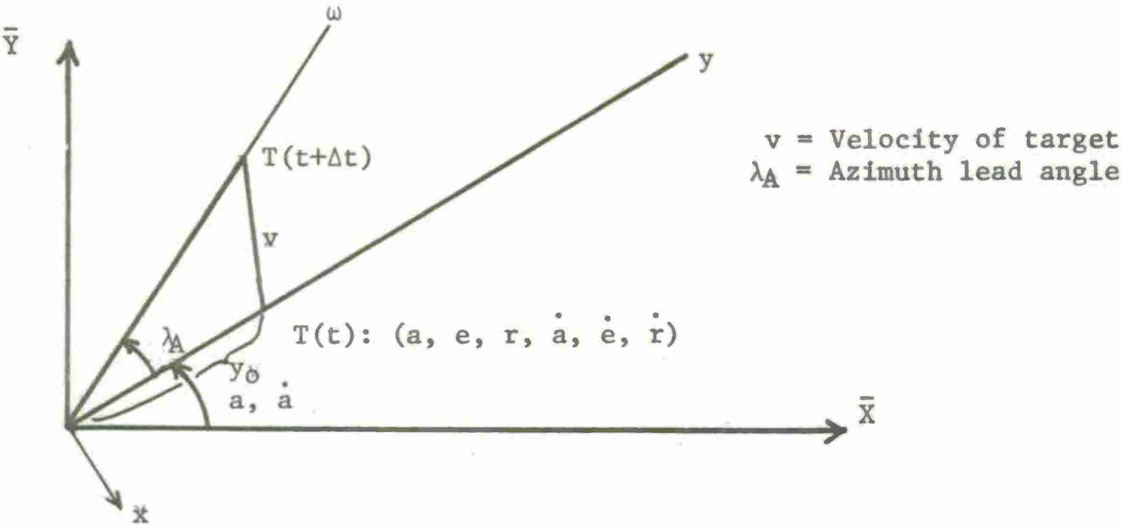


Figure A-1. Ground Plane Projection Of An Aircraft Flying At A Constant Speed.

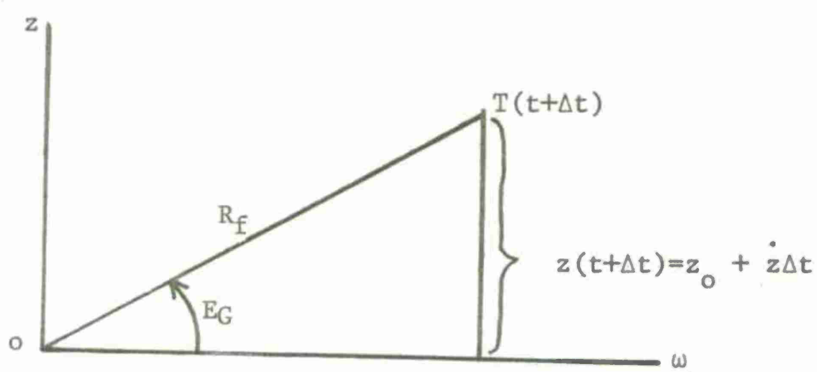


Figure A-2. Elevation View Of The Target At Time Of Predicted Intercept

The terms defining E_G can be expressed in terms of the measured quantities; therefore, the necessary partial derivatives can be obtained using the chain rule. Note that the actual measurement of the present elevation angle of the target is required here to calculate the elevation angle of the gun.

Time of Flight

The time of flight of the projectile to the point of predicted intercept must also be expressed in terms of the six measured quantities. This is required because the partial derivatives of this quantity are necessary to complete the total partial derivatives of the expressions for A_G and E_G.

This derivation is based upon the two assumptions:

1. Linear prediction (i.e., the target is not accelerating)
2. The projectile obeys the following equation

$$\Delta t = \frac{R_f}{V_m - \bar{K}R_f}$$

The derivation proceeds as follows:

Solving the R_f from the equation for Δt

$$R_f = \frac{\Delta t \; V_m}{1 + \bar{K}\Delta t} \quad .$$

Referring to Figure A-1

$$R_f^2 = V_t^2 \, \Delta t^2 + 2r\dot{r} \, \Delta t + r^2$$

and equating to obtain

$$\frac{V_m^2 \, \Delta t^2}{(1 + \bar{K} \, \Delta t)^2} = V_t^2 \, \Delta t^2 + 2r\dot{r} \, \Delta t + r^2$$

Carrying out the indicated multiplications and combining coefficients of powers of Δt yields

$$\Delta t^4(\bar{K}^2 V_t^2) + (2\, r\dot{r}\, \bar{K}^2 + 2V_t^2\bar{K})\Delta t^3 + (r^2\bar{K}^2 + 4\, r\dot{r}\, \bar{K} + V_t^2 - V_m^2)\, \Delta t^2$$

$$+ (2\, r^2\bar{K} + 2\, r\dot{r})\, \Delta t + r^2 = 0$$

where

$\bar{K}$ = ballistic coefficient of the projectile

V_t = velocity of the target

V_m = muzzle velocity of the projectile

The only term in this quartic equation which must be expressed in terms of the measured quantities is V_t, the velocity of the target. This can be expressed as

$$V_t^2 = \dot{x}^2 + \dot{y}^2 + \dot{z}^2$$

$$= (r\dot{a}\cos(e))^2 - r^2\dot{e}^2) + \dot{r}^2 \quad .$$

This completes the derivation of the equations defining the position of the gun in terms of the measured quantities. The partial derivatives of the expressions for A_G and E_G can be obtained by application of the chain rule for partial differentation. Similarly, the partial derivative of Δt with respect to the measured quantities can be obtained and used in the expressions for E_G and A_G.

Angular Rates of Guns

The angular rates of the gun in the azimuth and elevation directions are required so checks in the ISO-PK program can be made against maximum rates allowed. These rates can be obtained from the expression derived earlier in this appendix.

The partial differential equations necessary to obtain F on page 13 are contained in the section titled FORTRAN Source Program Listing on pages 56, 57, and 58 of the listing in subroutine GFUNC. The derivation of these partials is straight forward but very tedious. It was felt that inclusion of the derivation of these partials would be of little interest to readers or users of this report. Therefore, only the final form of the partials in FORTRAN notation is included. The terms are given FORTRAN names which make a mnemonic interpretation of the quantity easy. For example, PAXD stands for the partial of azimuth with respect to X-dot or $\frac{\partial A}{\partial \dot{x}}$.

DISTRIBUTION LIST

No. of Copies	
1	Office of the Under Secretary of the Army ATTN: SAUS-OR Washington, DC 20310
1	HQDA (DAMA-WSM/MAJ Reed) Washington, DC 20310
1	Commander US Army Development and Readiness Command ATTN: DRCPA-S 5001 Eisenhower Avenue Alexandria, VA 22333
	Commander US Army Armament Command
15	ATTN: DRSAR-SAS
1	DRSAR-SAA
1	DRSAR-SAL
1	DRSAR-SAM
2	DRSAR-RDG
1	DRSAR-ASV
	Rock Island, IL 61201
1	Commander US Army Test and Evaluation Command ATTN: DRSTE-SY Aberdeen Proving Ground, MD 21005
1	Commander US Army Electronics Command ATTN: DRSEL-SA Fort Monmouth, NJ 07703
1	Commander US Army Missile Command ATTN: DRSMI-CS Redstone Arsenal, AL 35809
1	Commander US Army Tank Automotive Command ATTN: DRSTA-V Warren, MI 48090
1	Commander HQ, US Army Aviation Command ATTN: DRSAV-D P.O. Box 209, Main Office St. Louis, MO 64502

DISTRIBUTION LIST (Cont'd)

No. of Copies	
1	Commander US Army Troop Support Command ATTN: DRSTS-G St. Louis, MO 63120
1	Commander US Army Materiel Systems Analysis Activity ATTN: DRXSY-AD Aberdeen Proving Ground, MD 21005
	Commander Rock Island Arsenal
6	ATTN: SARRI-LR
2	SARRI-LPL Rock Island, IL 61201
3	Commandant US Army Air Defense School Fort Bliss, TX 79916
1	Project Manager for Army Radar-Directed Gun Air Defense System US Army Armamant Command ATTN: DRCPM-ARGADS Rock Island, IL 61201
1	Commander Frankford Arsenal ATTN: SARFA-FCW-N4200 Philadelphia, PA 19137
1	Commander US Army Aviation Center Fort Rucker, AL 36362
1	Office of Project Manager for XM1 Tank Systems ATTN: DRCPM-GCM-SI Warren, MI 48090
12	Defense Documentation Center Cameron Station Alexandria, VA 22314

Next page is blank.