UNANNOUNCEL

REPORT D

AD-A956 318

Form Approved
OMB No. 0704-0188

①

1. AGENCY USE ONLY (Leave bla

21 Sept 77

E AND DATES COVERED
Memorandum, Sept. 1977

4. TITLE AND SUBTITLE
CSPAM: A Program For Evaluating Buoy Field Detection Performance

5. FUNDING NUMBERS
C N00014-77-C-0385

6. AUTHOR(S)
McCABE, Bernard J.

DTIC
ELECTE
MAR 1 8 1992
S C D

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Daniel H. Wagner, Associates, Inc.
Station Square Two
Paoli, PA 19301

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
COMPATWINGSPAC

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unrestricted

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The purpose of this memorandum is to describe a revised version of the computer program SPAM, which will be called CSPAM. It incorporates a more detailed model for spatial correlation.

14. SUBJECT TERMS
Detection modelling, spatial correlation, cumulation detection probability

15. NUMBER OF PAGES
47 (incl. Appendix)

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	Contract	PR	Project
G	Grant	TA	Task
PE	Program Element	WU	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

DANIEL H. WAGNER, PRES.
HENRY R. RICHARDSON, SR. V. PRES.
DAVID C. BOSSARD, V. PRES.
LAWRENCE D. STONE, V. PRES.
BARRY BELKIN, V. PRES.
LESLIE K. ARNOLD, SR. ASSOC.
BERNARD J. McCABE, SR. ASSOC.
THOMAS L. CORWIN, SR. ASSOC.
STANLEY J. BENKOSKI, ASSOC.
WILLIAM J. BROWNING, ASSOC.
BRUCE E. SCRANTON, ASSOC.
WILLIAM H. BARKER, ASSOC.
FRANK P. ENGEL, ASSOC.
SCOTT S. BROWN, ASSOC.
ROBERT P. BUEMI, ASSOC.
LARRY K. GRAVES, ASSOC.
JAMES R. WEISINGER, ASSOC.

DANIEL H. WAGNER, ASSOCIATES
CONSULTANTS
OPERATIONS RESEARCH MATHEMATICS
STATION SQUARE ONE
PAOLI, PENNSYLVANIA 19301

NIAGARA 4-3400
AREA CODE 215
(620)

September 21, 1977

MEMORANDUM REPORT

To: COMPATWINGSPAC
Attn: LCDR Philip Harvey

Contract No. N00014-77-C-0385

From: B. J. McCabe

Subject: CSPAM - A Program For Evaluating Buoy Field Detection Performance

The purpose of this memorandum is to describe a revised version of the computer program SPAM, which will be called CSPAM, which attempts to account for the effects of range-dependent buoy to buoy correlation. In SPAM the detection capability of a field of buoys is arrived at by averaging what it would be if the buoys were statistically independent with what it would be if they were completely dependent. The weighting does not depend on the spacing of the buoys. In CSPAM a more elaborate detection model is used so that the correlation between a pair of buoys is monotone decreasing as the range between the two buoys varies from 0 to ∞. The anticipated use of CSPAM is as a tool to study optimal buoy field patterns and spacing.

In the sections to follow we describe the mathematical model for detection used in CSPAM and compare it to that in SPAM, give some sample runs of

CSPAM to illustrate the effects of varying the correlation function, give an abbreviated programmer's guide, and provide a listing of CSPAM.

The inputs to CSPAM, as currently configured, are identical in form and import to those for SPAM, with one exception: the parameter which determines the correlation function. Moreover, all of SPAM's subroutines are incorporated in CSPAM. For these reasons no separate user's guide for CSPAM is given - the reader should consult reference [a] for a SPAM user's guide as well as reference [b] for additional background material on SPAM and its detection model.

A comprehensive assessment of the impact of treating correlation by means of this more elaborate model has not yet been made. Such analysis will be forthcoming. Some sample runs are included but they pertain to only one environment and one target scenario.

The abbreviated programmer's guide has a twofold purpose. Firstly, the routines in CSPAM that are different from those in SPAM are explained. Secondly, there are portions of CSPAM that are currently configured in an arbitrary way, awaiting further discussion and deliberation. Attention is directed to these problem areas and the parts of the program where suitable changes can be made are pointed out. This memorandum is partly intended as an impetus for discussion of these provisional aspects of CSPAM.

Underlying Analysis

In this section we discuss the problem of modeling correlated sensors, and how this problem is addressed in CSPAM.

Correlated sensors. In modeling acoustic detection it is generally assumed that the occurrence of a detection is determined by a function

$$SE(t) + \alpha(t), \quad t \geq 0$$

where

$$SE(t) = FOM - N_w(R(t))$$

is the expected signal excess at time t and α is a stochastic process representing the uncertainties and fluctuations in the various components of figure of merit (FOM) and propagation loss (N_w). R(t) is the distance between the sensor and the target at time t.

Suppose there are n sensors and a single target and n corresponding signal excess histories SE_i and fluctuation processes α_i, $i = 1, \ldots, n$. Suppose that cumulative detection probability is given by a threshold crossing* model

$$cdp_i(t) = Pr\{SE_i(s) + \alpha_i(s) \geq 0 \quad \text{for some } s \leq t\}.$$

If the processes α_i, $i = 1, \ldots, n$, were all mutually independent, then cdp for the field of sensors as a whole would be an easy computation,

$$cdp_I(t) = 1 - \prod_{i=1}^{n} (1 - cdp_i(t)),$$

or if the fluctuation processes were identical, $\alpha_1 = \ldots = \alpha_n = \alpha$, we would have an easy calculation

---

*The detection model in SPAM and CSPAM is slightly different from this but the gist of this discussion is relevant.

$$cdp_D(t) = \Pr\{\alpha(s) + \max_{1 \le i \le n} SE_i(s) \ge 0 \quad \text{for some } s \le t\}.$$

However, it is reasonable to believe that the processes α_i, $i = 1,\ldots,n$, have correlation between 0 and 1, and that the amount of correlation between two sensors should decrease with the distance between them. The SPAM model recognizes correlation by taking a weighted average of cdp_I and cdp_D but the weighting is independent of the spacing among the n sensors.

<u>Model for range dependent correlation</u>. Suppose that each process α_i is a sum of processes each of which is a model for the fluctuations contributed by the components of the sonar equation: target radiated noise (L_S), background noise (L_E), recognition differential (N_{RD}), and propagation loss (N_W).

We make the following assumptions which are rough approximations and subject to debate.

Fluctuations in L_S are felt equally at all sensors so we assume that the component of α_i due to L_S is identical for each i. This neglects the aspect dependence of radiated noise.

For buoys, background noise is dominated by ambient noise which at most frequencies of interest is due to shipping. Thus we assume that the component of α_i due to L_E is correlated with the corresponding component of α_j, for all $i \neq j$, and the correlation structure is given by a model such as that of reference [c], and, in particular, is range dependent. More about this in a later subsection.

The correlation in N_{RD} from buoy to buoy is determined by the performance of the same operator examining n lofargrams. If the operator is more or less perceptive, diligent or attentive in examining one lofargram then he will tend to perform similarly at each other display. Thus we assume that the component of α_i due to N_{RD} is positively correlated with the corresponding component of α_j and that this amount of correlation is the same for any pair i, j.

The effects due to propagation loss are more difficult to describe compactly. There is no doubt some correlation between any pair of sensors. The correlation depends not only on the distance between the sensors but also on the position of the target relative to the sensor pair, and thus the amount of correlation is also <u>time dependent</u> since the target is moving. These effects could be addressed but only at the expense of great added complexity and computer time, so we have for the present decided to ignore these effects and hypothesize an <u>independent</u> component for N_W.

To summarize, we have four major contributors to uncertainty in SE. Three of them will be modeled fairly adequately, and it is hoped that the deficiency in the modeling of the other term will not be critical, especially since we only require that the sum of these four terms be reasonably accurate in its portrayal of buoy to buoy correlation.

Our mathematical model for each α_i is then of this form*:

$$\alpha_i = \xi + \xi_i + \eta_i^1 + \eta_i^2, \quad i = 1, \ldots, n,$$

where:

ξ is $N(0, \sigma_1)$ and independent of all ξ_i, η_i^1, η_i^2,

ξ_i is $N(0, \sigma_2)$, independent of ξ_j for $j \neq i$, and independent of all η_j^1, η_j^2,

η_i^1 is $N(0, \sigma_3)$, independent of all η_j^2, and has correlation $\rho \geq 0$ with each η_j^1, $j \neq i$, and

η_i^2 is $N(0, \sigma_4)$ and has correlation $C(r_{ij})$ with η_j^2 where r_{ij} is the distance between sensors i and j, and C is the ambient noise correlation function to be discussed below.

---

*$N(\mu, \sigma)$ denotes a random variable with a Gaussian distribution, mean μ and standard deviation σ.

The variance of each α_i is thus $\sigma^2 = \sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^4$.

Note that we have modeled these components as random variables, not as time varying stochastic processes. This is consistent with the methodology of SPAM to which we are adhering. A time varying model is more realistic. It allows for detections past CPA, for example, which are precluded in this kind of model. Unfortunately, for this application it would lead to an unreasonably long running time for the program, and it is therefore a luxury that cannot be afforded.

As a comment on the mathematical model it might be mentioned that if a time varying model were being used and if running time were not a problem, a reasonable parametric model for propagation loss correlation could be easily constructed. For example, one might hypothesize that correlation between two buoys is of the form $\kappa R_1(t) \cos (\theta(t)/2)/R_2(t)$ where R_1 and R_2 are the ranges from the target to the two buoys, R_1 being the smaller, and θ is the angular separation of the two buoys as viewed from the target. It can be checked that this formula gives sensible values in all limiting cases and the parameter κ can be used to tune the results to any available data.

Finally, we note that choices must be made for the parameters σ_1, σ_2, σ_3, σ_4 and ρ. These choices are difficult because there probably is not any suitable body of data from which to estimate them. For lack of any better information the program is currently configures with $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 4$ db and $\rho = .5$ and these values were used in the test runs included in the next section. These particular values imply that the correlation between α_i and α_j is between .375

and .625, these limiting values being reached when $r_{ij} = +\infty$ and $r_{ij} = +0$, respectively. This is a fairly narrow window and it will be seen that the result is to make cdp not extremely sensitive to the shape of the correlation function C. If σ_4 were higher, the other σ_i lower and ρ lower, then the window would be larger and the results more sensitive to C. In other words the choices for these parameters are quite important and need to be discussed.

The CTW model. A thorough description of the Cumulative Time Window (CTW) model is in reference [b] but we give a summary account here for purposes of our discussion. In this subsection we discuss the model as applied in SPAM; the next subsection describes the revision in CSPAM.

There are three inputs required in SPAM: t_0, t_1, and t_2. To put it briefly, t_0 is a minimum integration time, t_2 is a maximum integration time (opportunities at a fixed signal excess which are of longer duration that t_2 buy no additional detection probability) and t_1 is the time step used in the simulation.

Since SPAM takes a simple weighted average of independent and completely dependent buoys it suffices to define $\text{cdp}_i(j)$, the probability that buoy i makes a detection by time jt_1.

Let

$$c_{ij} = \Pr\{SE_i(jt_1) + X \geq 0\}$$

where X is $N(0,\sigma)$; c_{ij} is called the instantaneous probability of detection (ipd). Let $A(j) = \{j - \left[\frac{t_2}{t_1}\right], \ldots, j-1, j\}$.

Then we define

$$\mathrm{cdp}_i(j) = \begin{cases} 1 - \prod_{k \in A(j)} (1-c_{ik})^{t_1/t_0} & \text{if this quantity is} \geq \mathrm{cdp}_i(j-1), \\ \mathrm{cdp}_i(j-1) & \text{otherwise.} \end{cases}$$

To initialize, assume $\mathrm{cdp}_i(0) = 0$ and $c_{ik} = 0$ for any $k < 0$.

<u>Incorporation of the general model into the framework of SPAM.</u> Just as in SPAM, an initial position for the target, its course, and its speed are drawn from the appropriate distributions, for each replication. SPAM at this point computes cdp_i analytically for each buoy i, and arrives at cdp for the field by combining cdp_I and cdp_D as described above. Each replication produces a cdp curve and these are all averaged to give the final estimated cdp.

In CSPAM we cannot compute cdp for the buoy field analytically because of the complicated correlation structure of the α_i's. Recalling that

$$\alpha_i = \xi + \xi_i + \eta_i^1 + \eta_i^2$$

we see that one way to proceed would be to simulate values of ξ, η_i^1, η_i^2, $i=1,\ldots,n$ and compute cdp analytically from this point since the detection events $SE_i + \alpha_i - N_W(R_i) > 0$ are independent when conditioned on values of ξ, η_i^1, η_i^2. This is basically how CSPAM works.

As it turns out, because of the special nature of the correlation structure of the η_i^1, it is actually not necessary to simulate them. The following gimmick allows us to simulate two processes instead of three, which saves labor and reduces variance.

Since the correlation of η_i^1 and η_j^1 is ρ for all $i \neq j$, we may represent η_i^1 in the form

$$\eta_i^1 = \sqrt{\rho}\ \sigma_3 N_0 + \sqrt{1-\rho}\ \sigma_3 N_i$$

where N_0, $N_1, \ldots, N_n$ are N(0, 1) and independent. This is a sort of decomposition of each η_i^1 into identical and independent parts. Now these parts can be grouped with ξ and ξ_i respectively to give the following representation:

$$\begin{aligned} \alpha_i &= \xi + \xi_i + \eta_i^1 + \eta_i^2 \\ &= (\xi + \sqrt{\rho}\ \sigma_3 N_0) + (\xi_i + \sqrt{1-\rho}\ \sigma_3 N_i) + \eta_i^2 . \end{aligned}$$

Thus there is actually no need to simulate the η_i^1's. Their role can be subsumed into two other terms if we make the transformations

$$\sigma_1 \rightarrow \sqrt{\sigma_1^2 + \rho\ \sigma_3^2} \quad \text{and}$$

$$\sigma_2 \rightarrow \sqrt{\sigma_2^2 + (1-\rho)\ \sigma_3^2}\ .$$

In addition to saving computer time this reduces the variance of our results since a ratio $(\sigma_2^2 + (1-\rho)^2\ \sigma_3^2)/\sigma^2$ of our work on the α_i's is now being done analytically instead of just σ_2^2/σ^2.

Let $\bar{\xi}$ denote the adjusted process $\xi + \sqrt{\rho}\ \sigma_3 N_0$ and $\bar{\xi}_i$ the adjusted process $\xi_i + \sqrt{1-\rho}\ \sigma_3 N_i$. Then

$$\alpha_i = \bar{\xi} + \bar{\xi}_i + \eta_i^2 .$$

In CSPAM we make draws for $\bar{\xi}$ and η_i^2, $i = 1, \ldots, n$, and compute the conditional idp

$$c_{ij}(\bar{\xi}, \eta_i^2) = \Pr\{SE_i(jt_1) + \bar{\xi} + \bar{\xi}_i + \eta_i^2 \geq 0\}.$$

Then cdp_i, conditioned on the choices of $\bar{\xi}$, η_i^2, is computed using the c_{ij}, just as in SPAM, and finally cdp conditioned on $\bar{\xi}$, η_i^2, $i = 1, \ldots, n$, is computed as

$$cdp(j) = 1 - \prod_{i=1}^{n} (1 - cdp_i(j)).$$

Just as in SPAM, these cdp functions, one for each replication, are then averaged.

The method for generating the η_i^2's with the correct joint distribution will now be described.

Let D be an artitrary n x n covariance matrix. Let η be an n-dimensional vector of independent N(0, 1) random variables. Since D is non-negative definite there exists an n x n matrix B such that $BB^T = D$. Then $B\eta$ is an n-dimensional vector of Gaussian random variables whose i and j components have as their covariance the (i, j) element of D.

CSPAM uses this technique to generate sample values of η_i^2, $i = 1, \ldots, n$. For the η_i^2, D is a matrix with $\sigma_4^2 C(r_{ij})$ in the (i, j) position.

CSPAM uses the Cholesky algorithm to compute the matrix B such that $BB^T = D$. This is discussed in reference [c].

<u>The range correlation function.</u> A required input to CSPAM is the function C(r) which gives the correlation in ambient noise received at two sensors

separated by a distance r. In principle any correlation function C can be used whether it is derived from data or a mathematical model such as in reference [c]. The current version of CSPAM, however, assumes the C(r) is of the form $\exp(-\beta r)$. The user inputs a value of the range R* at which he feels correlation in ambient noise is equal to .5. The program then computes $\beta = (\ln 2)/R^*$.

There are several reasons for using this simple parametric form at this time. The main one is that CSPAM is intended as a research tool which will enable us to decide on optimal patterns and their spacings and these will depend mainly on the rate of decrease of the function C. Having R* as an input gives us a convenient way to vary this rate of decrease while holding other tactical variables fixed.

A related reason is that if C were to be estimated via the technique of reference [c] which depends on the propagation loss curve, it has been our experience that the shape of C is extremely sensitive to the fairly minor oscillations of N_W. In most tactical problems, however, we do not know N_W well enough to predict all of its oscillations and minor wiggles so that our conclusions may be based on spurious effects.

Eventually a technique such as in reference [c] may be used to characterize the C that pertains to a certain type of environment by averaging the results over several representative propagation losses for that environment but for now we recommend using the exponential correlation function to study the effects of variation in R*.

Sample Results

Nine sample runs were make, six with a 5-6-5 distributed field and three with a BRUSHTAC. In each case all inputs are fixed except figure of merit and R*. The fixed inputs are shown in Table 1. Figures 1 to 9 show the resulting cdp functions and Table 2 summarizes the total cdp results. These runs can be used to compare results with SPAM to verify that the two programs are giving compatible answers.

We note from Table 2 that the results as a function of R* do not vary spectacularly and that this is a result of the relatively narrow (.375, .625) correlation window mentioned earlier. If the values $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 4$ db, $\rho = .5$, that were used are valid, one begins to suspect, pending more extensive runs, that the effects of range dependent ambient noise correlation are minor. However, these input values must be subjected to scrutiny.

TABLE 1

Fixed Inputs to Nine Sample Runs

```
ENTER FREQUENCY INDEX (1, 2 OR 3): 2
ENTER SOURCE-RECEIVER GEOMETRY (1=D-D, 2=S-S, 3=D-S) : 1
ENTER .5 CORRELATION RANGE (MILES): 100
ENTER THE FIGURE OF MERIT: 74
ENTER YOUR PATTERN TYPE.  THE CHOICES ARE:
          DF44      DF56      DF88      BTAC
          CIRC      ARC       SPEC
ENTER PATTERN TYPE: DF56
ENTER NUMBER OF BUOYS MONITORED: 16
ENTER MONITORING TIME (MINUTES): 180
ENTER THE TRUE PATTERN BEARING: 90
SPECIFY PATTERN SPACING AS FRACTION OF CZ? (Y OR N): N
ENTER THE PATTERN SPACING (MILES): 18
ENTER E/B TO SPECIFY AN ELLIPSE/BEARING BOX: E
ENTER SEMI-MAJOR AXIS LENGTH: 25
ENTER SEMI-MINOR AXIS LENGTH: 15
ENTER TRUE BEARING, RANGE FROM SPA TO PATTERN CENTER: .0,0
ENTER THE TRUE BEARING OF THE TPA: 90
ENTER ON-STATION TIME IN HOURS: 3
ENTER A/C GROUND SPEED IN KNOTS: 250
ENTER THE TIME LATE IN HOURS: 0
SPECIFY TARGET MOTION CATEGORIES (Y OR N): N
ENTER THE TARGET COURSE: 0
ENTER THE COURSE STANDARD DEVIATION: -180
ENTER THE LOWER SOA LIMIT: 3
ENTER THE UPPER SOA LIMIT: 5
OUTPUT BUOY COORDINATES (Y OR N): N
ENTER A COMMENT TO IDENTIFY YOUR RUN:
ENTER THE NUMBER OF REPETITIONS: 100
ENTER THE RF RANGE: 80
PLOT PROPLOSS CURVE (Y OR N): N
ENTER RANDOM NUMBER SEED: ,".55555
ENTER SONAR EQ. STD. DEV.(1=6 DB, 2=8 DB, 3=10 DB, 4=12 DB): 2
```

TABLE 2

Summary of cdp Results

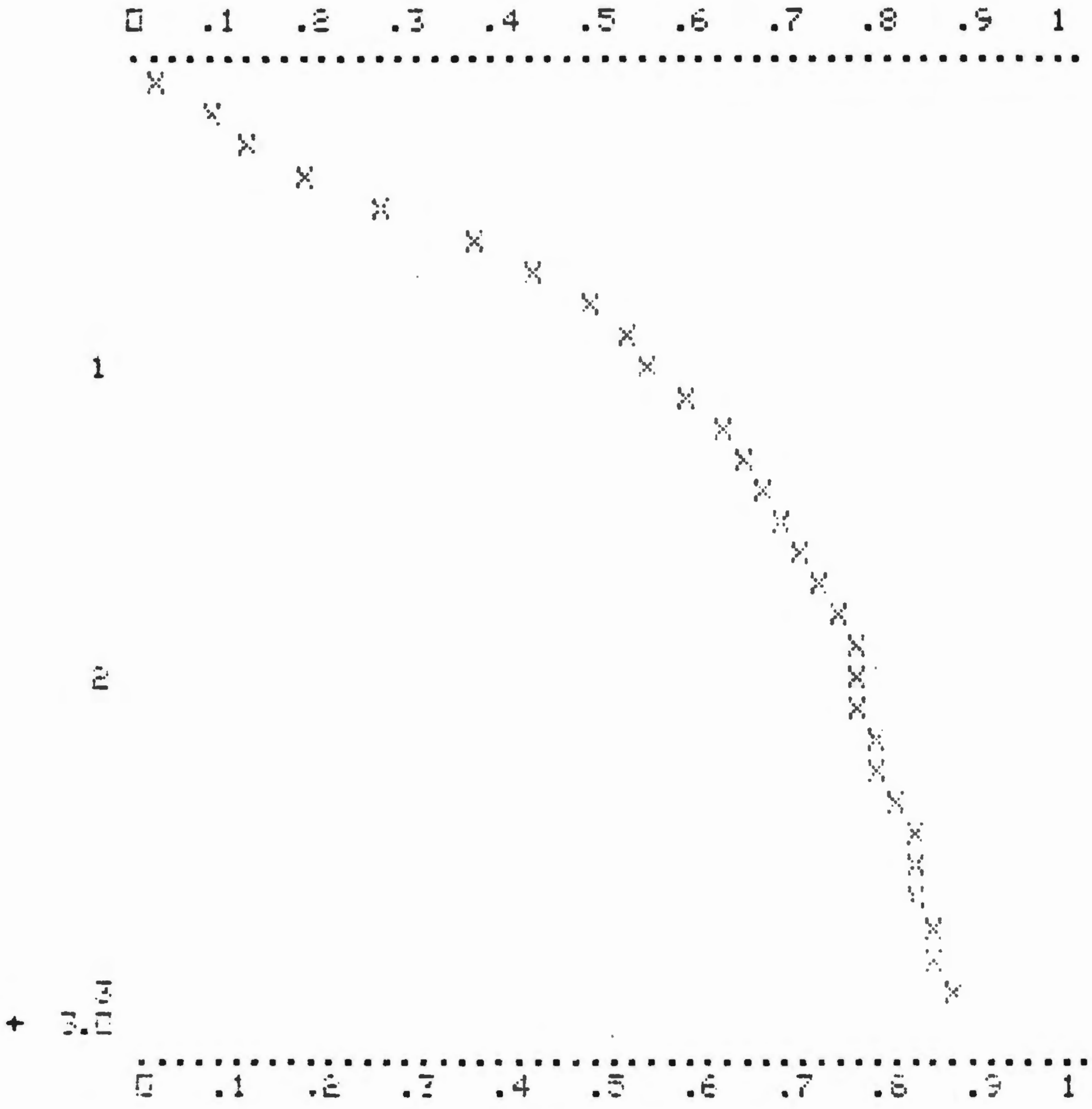
R* =	5 nm	18nm	100nm
DF 5-6-5, FOM=74 db	.86	.85	.79
DF 5-6-5, FOM=70 db	.67	.64	.56
BRUSHTAC, FOM=70 db	.74	.70	.66

FIGURE 1

CDP Results for Case 1

5-6-5 Distributed Field

FOM = 74 R_{50} = 5nm



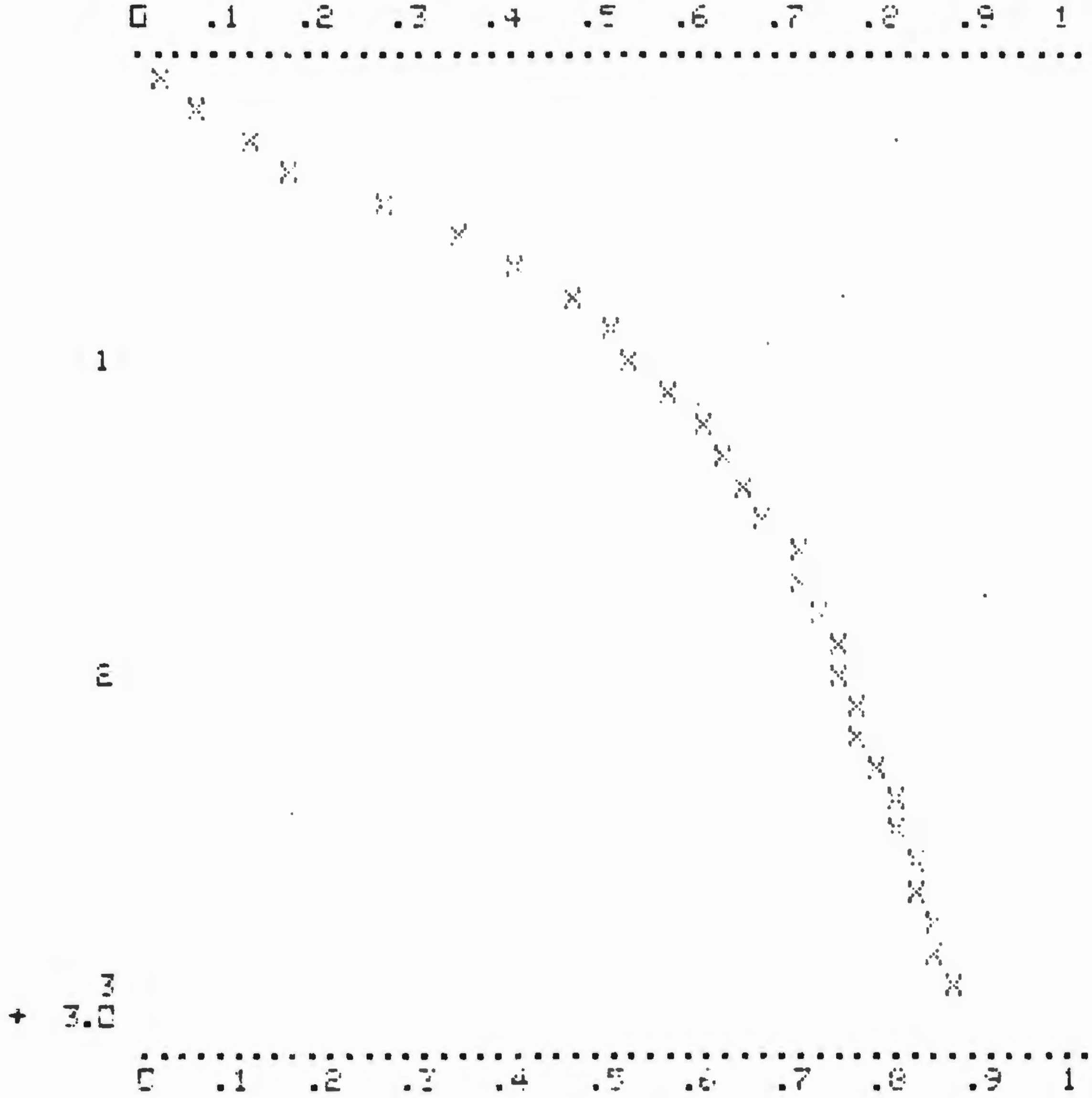
CUMULATIVE PROBABILITY OF DETECTION= 0.86

FIGURE 2

CDP Results for Case 2

5-6-5 Distributed Field

FOM = 74 R_{50} = 18nm



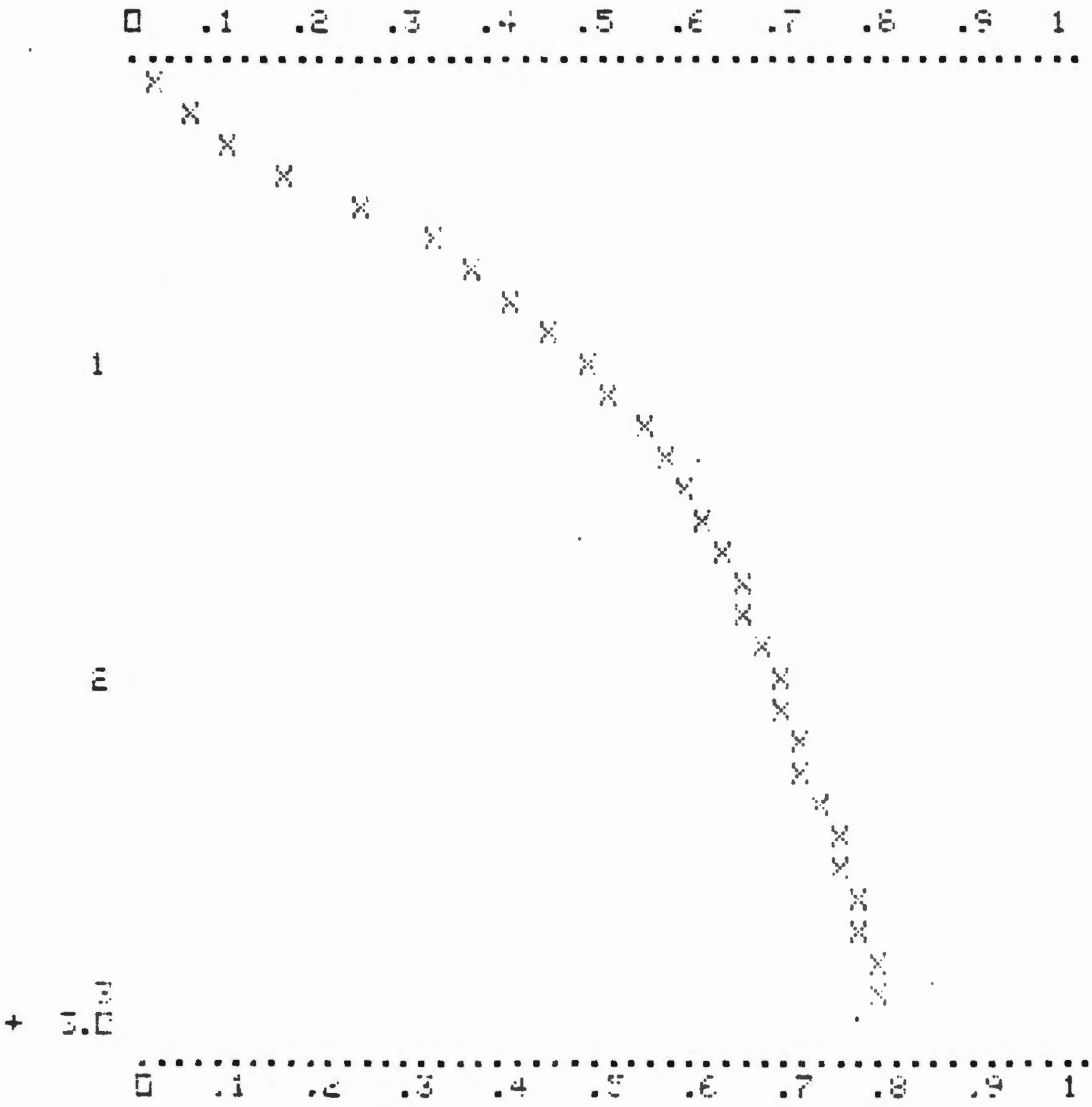
CUMULATIVE PROBABILITY OF DETECTION= 0.85

FIGURE 3

CDP Results for Case 3

5-6-5 Distributed Field

FOM = 74 R_{50} = 100 nm



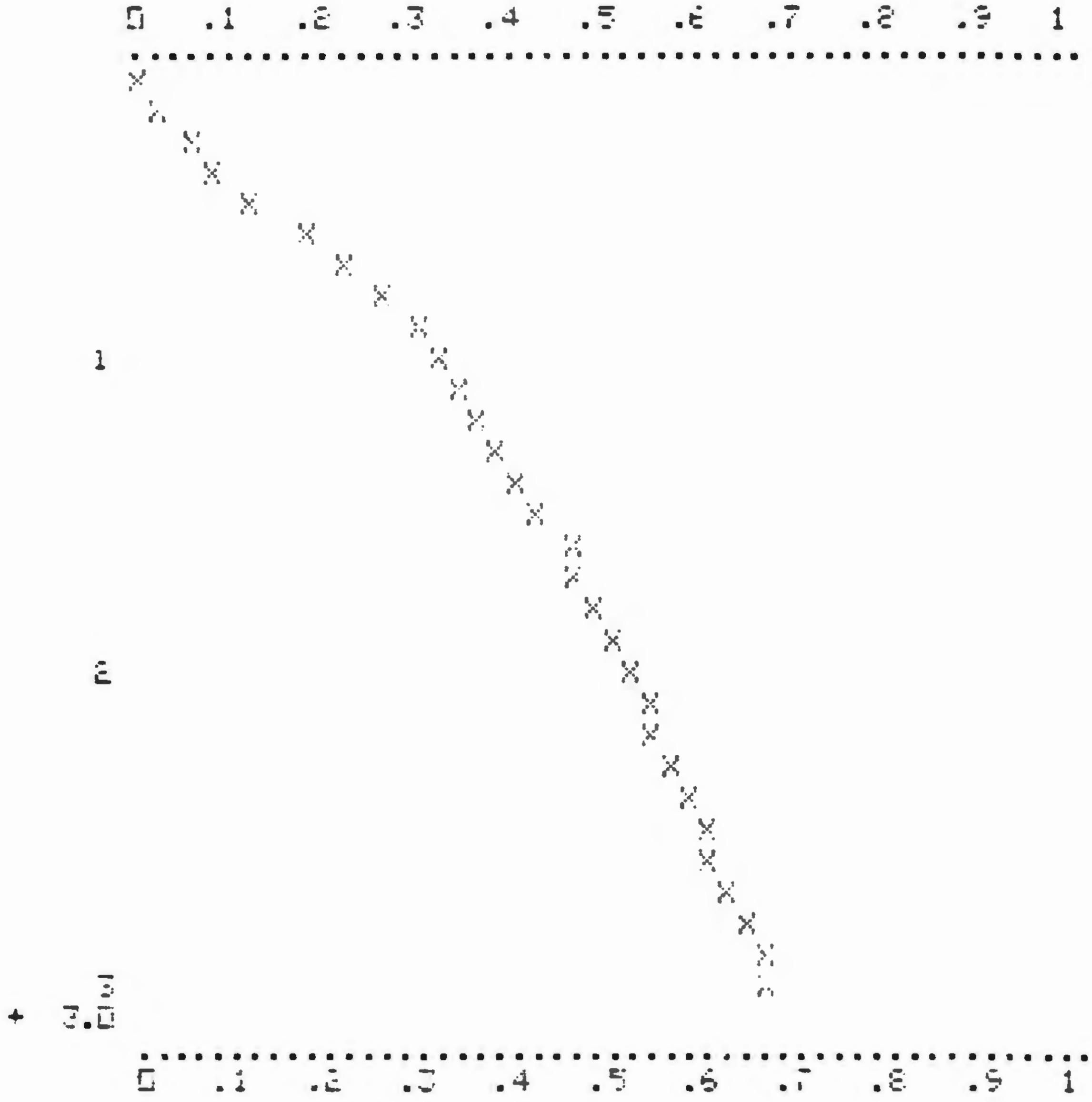
```
CUMULATIVE PROBABILITY OF DETECTION=   0.79
```

FIGURE 4

CDP Results for Case 4

5-6-5 Distributed Field

FOM = 70 R_{50} = 5nm



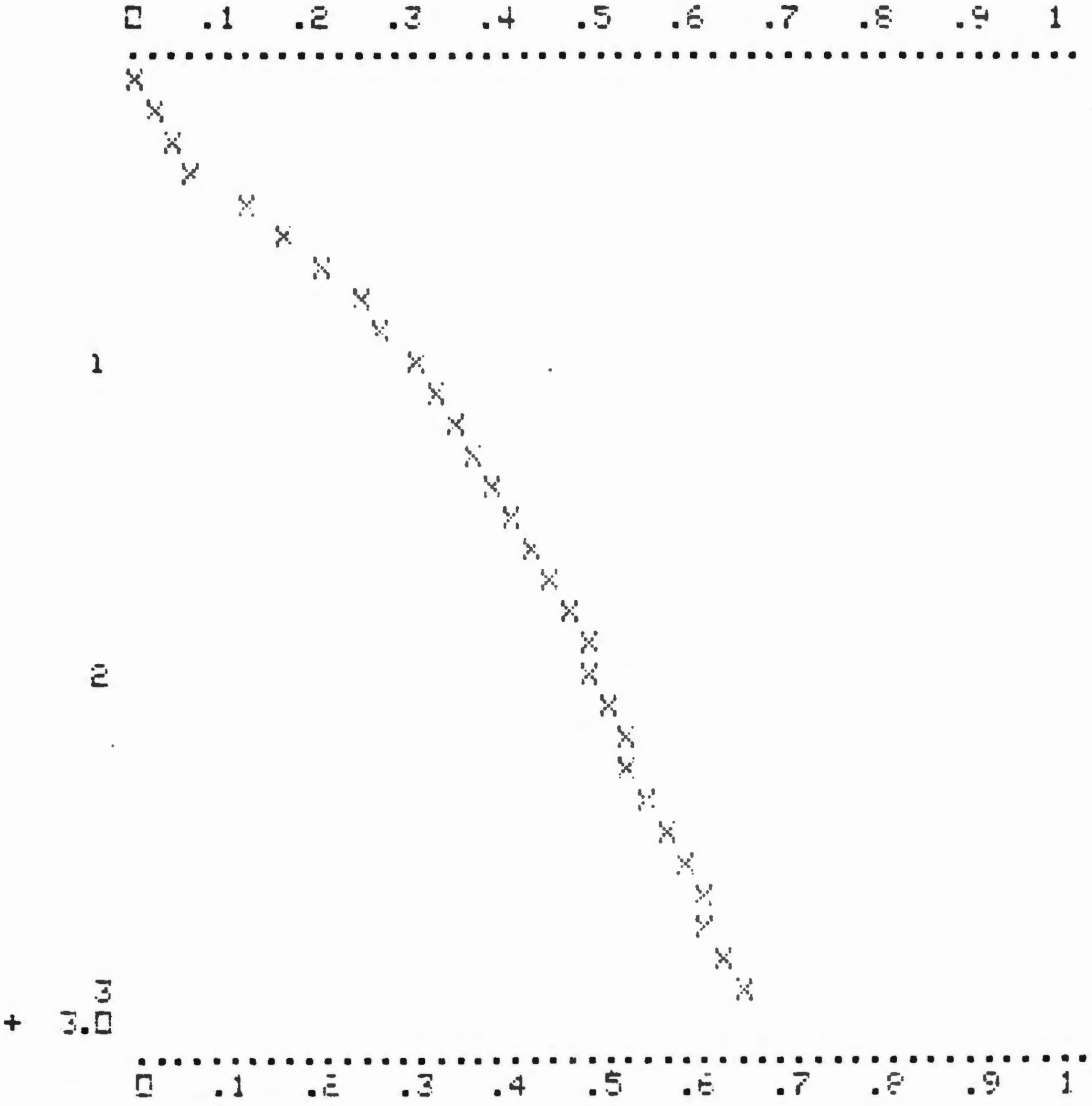
```
CUMULATIVE PROBABILITY OF DETECTION=   0.67
```

FIGURE 5

CDP Results for Case 5

5-6-5 Distributed Field

FOM = 70 R_{50} = 18nm



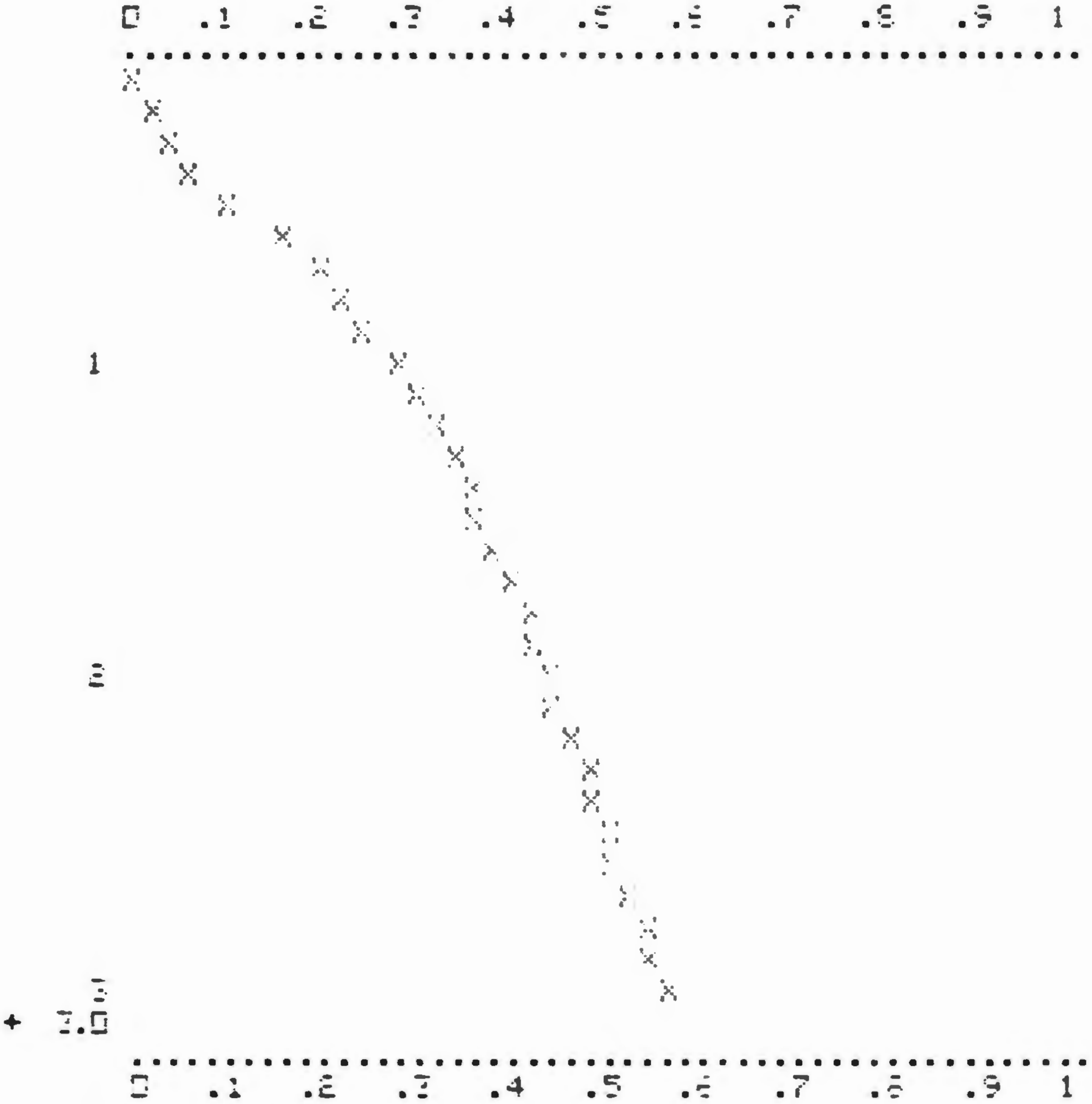
```
CUMULATIVE PROBABILITY OF DETECTION=   0.64
```

FIGURE 6

CDP Results for Case 6

5-6-5 Distributed Field

FOM = 70 R_{50} = 100nm



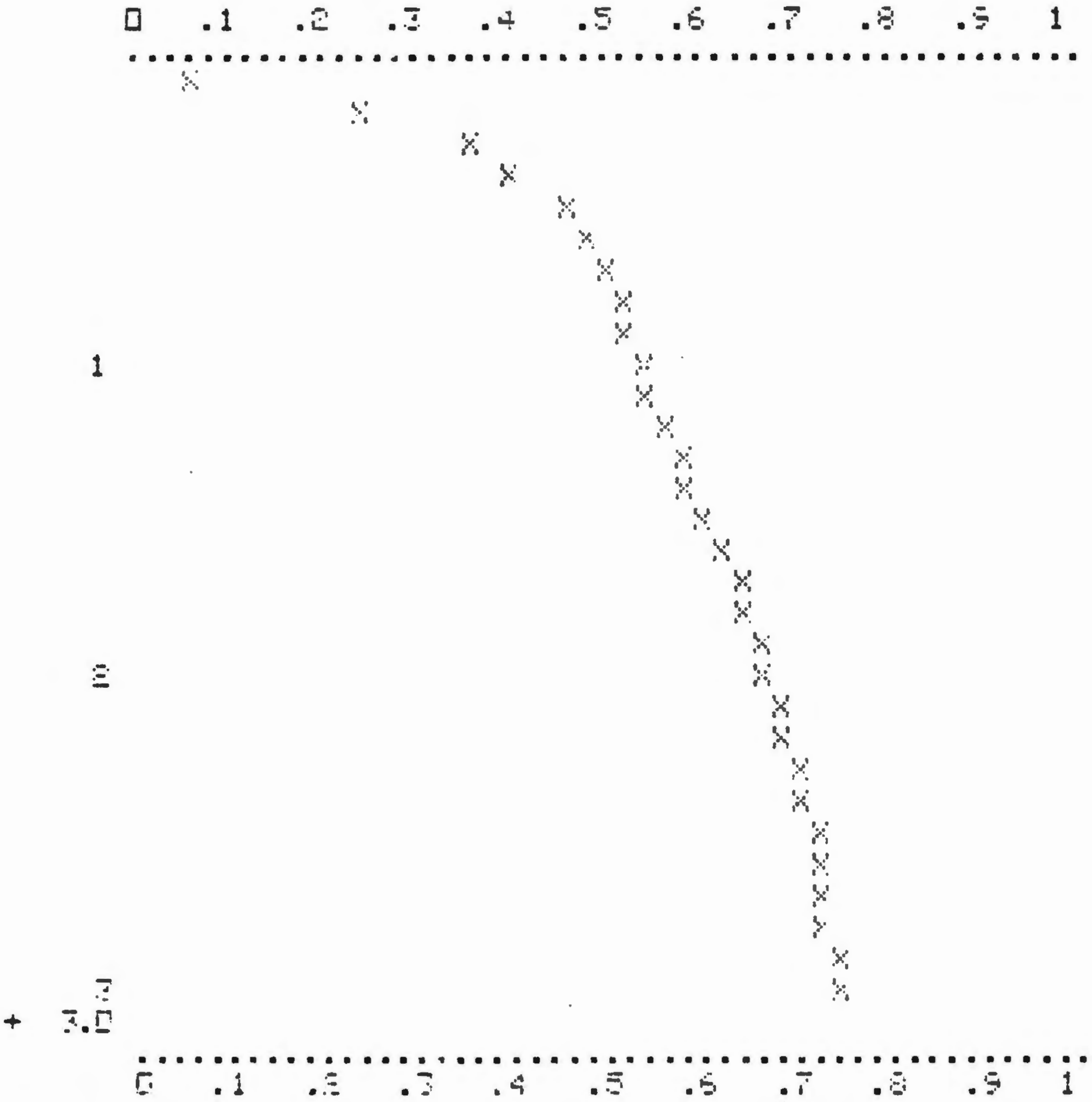
CUMULATIVE PROBABILITY OF DETECTION= 0.56

FIGURE 7

CDP Results for Case 7

BRUSHTAC

FOM = 70 R_{50} = 5nm



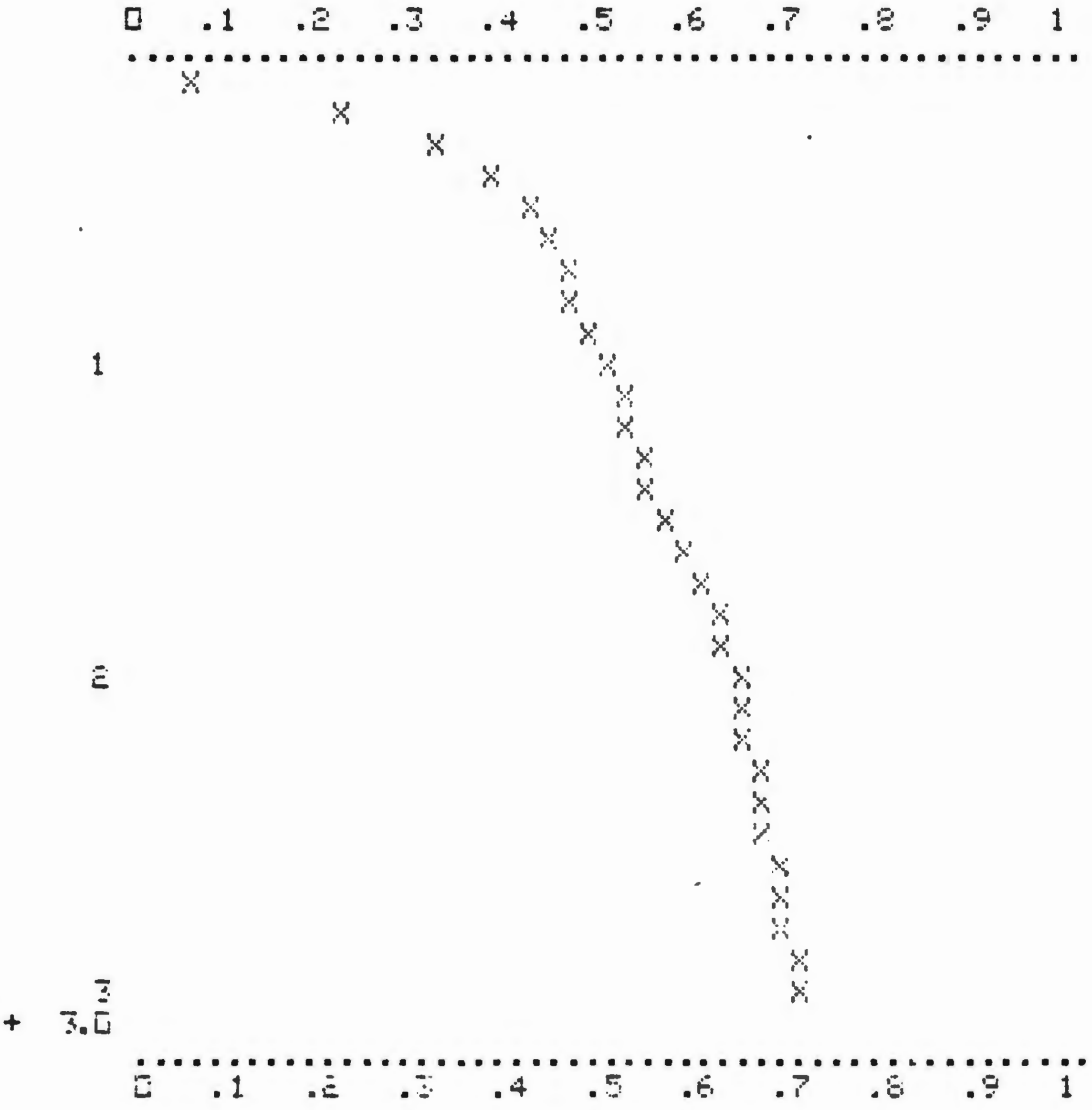
```
CUMULATIVE PROBABILITY OF DETECTION=   0.74
```

FIGURE 8

CDP Results for Case 8

BRUSHTAC

FOM = 70 R_{50} = 18nm



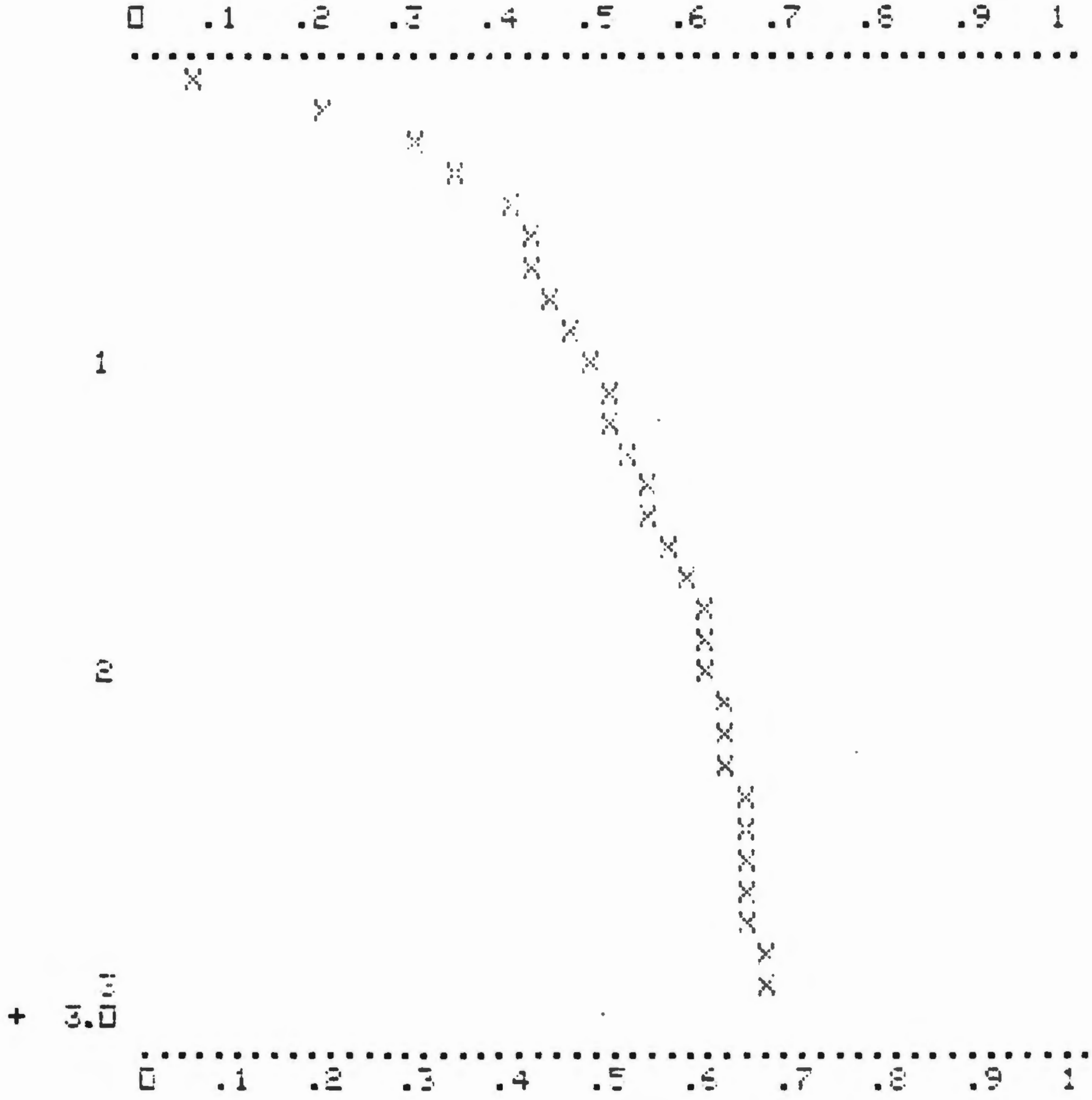
CUMULATIVE PROBABILITY OF DETECTION= 0.70

FIGURE 9

CDP Results for Case 9

BRUSHTAC

FOM = 70 R_{50} = 100nm



CUMULATIVE PROBABILITY OF DETECTION= 0.66

Programmer's Guide

The variables σ_1, σ_2, σ_3, σ_4, and ρ are set at the top of the program. They are called, respectively, SIG1, SIG2, SIG3, SIG4, and ALPHA. Currently, the σ_i's are set equal to $\sigma/2$ where σ is an input standard deviation, and ρ is set equal to .5. As mentioned above we may want to try other values of these variables in which case they may be changed directly in the code or a new set of enter statements may be programmed.

The correlation function is used on only one line in the program, in the CHOLES subroutine at the point where ETA(I, J) is defined. Thus an arbitrary correlation function could be accommodated easily by writing a routine to input it manually and changing this line in CHOLES.

In SPAM idp's are computed using a table that has range as its input. This is not possible in CSPAM because idp depends not only on range but on the conditioning values of $\bar{\xi}$ and η_i^2. Thus idp, called BW(IBUOY, INDEX), is computed as a function of FOM + $\bar{\xi}$ + η_i^2 - N_W using the array TABLE defined at the top of the program. Note that TABLE evaluates the cumulative Gaussian distribution raised to the t_1/t_o power. This saves doing an exponentiation in the innermost loop.

Bernard J. McCabe

Bernard J. McCabe

BJM/dd

References

[a] Search Pattern Assessment Model (SPAM), COMPATWINGSPAC report by G. A. Marin, April 14, 1976, Unclassified.

[b] ASW Detection Simulation and the Cumulative Time Window Model, forthcoming CNA memorandum, by G. A. Marin, Unclassified.

[c] Ambient Noise Effects in the Modeling of Detections by a Field of Sensors, Daniel H. Wagner, Associates report by B. J. McCabe, November 5, 1976, Unclassified.

APPENDIX

LISTING OF CSPAM

```
C********************************
C*
C*     PROGRAM XSPAM
C*     IMPROVED VERSION OF SPAM
C*
C*     BJM/BMM/JSR      JUNE, 1977
C*
C*     DANIEL H. WAGNER, ASSOCIATES
C*     PAOLI, PA
C*
C********************************
      DIMENSION BW(32,100),LM(32),DELTA(32),BS(8),
     *          ETA1(32,32),ETA2(32,32),PROB(220),PROPLO(220),
     *          RAN1(32),RAN2(32),JF(8),KG(8),
     *          BR1(32),BR2(32),PREV(32),TABLE(321),PLMAP(220),
     *          NCT(100),ICTRY(100),CTMAX(100),PCAT(100),THRSH(100),
     *          COMENT(15),BX(32),BY(32),BXI(32),BYI(32),DISTAB(32)
      NAMELIST/INFO/ISEED,NREPS,IFREQ,IGEO,RF,OSTH,NOPROP,T0,T1,T2,IGO,
     *   CORR50,BETA,DIMX,DIMY,COURSE,SPD,SDDIMX,SDDIMY,SDCRSE,SDSPD,
     *   TIMLAT,FOM,NPAT,NCRS,NSOA,SDEV,NB,NM,ACFTGS,TIMMON,PATSPC
      DATA YES/3HY  /
C
C
      CALL WPRNOS
C
      READ (10,5002) COMENT
      READ (10,5003) ISEED,NREPS,IFREQ,IGEO,RF,OSTH,NOPROP,T0,T1,T2,
     *                IGO,BETA,CORR50
      READ (10,5004) DIMX,DIMY,COURSE,SPD,SDDIMX,SDDIMY,SDCRSE,SDSPD,
     *                 TIMLAT,FOM,NPAT,SDEV
      READ (10,5005) NCRS,CRSINT,CRSWND,NSOA,SOAINT,SOAWND
      IF (NCAT.EQ.0) GO TO 25
      READ (10,5006) (PCAT(I),I=1,NCAT)
      READ (10,5006) (THRSH(I),I=1,NCAT)
   25 READ (10,5006)
      READ (10,5007) NB,NM,ACFTGS,TIMMON,PATSPC
      READ (10,5008) (BXI(I),I=1,NB)
      READ (10,5008) (BYI(I),I=1,NB)
      DO 28 I=1,NB
      BX(I)=BXI(I)
      BY(I)=BYI(I)
   28 CONTINUE
C
      WRITE (6,5000) (COMENT(I),I=1,15)
C
C
C
C    LIST BUOY POSITIONS
C
   30 WRITE (6,5019)
         DO 40 I=1,NB
   40    WRITE (6,5020) I,BX(I),BY(I)
      WRITE (6,5001)
C
C
C    SET-UP PROGRAM CONSTANTS
C
         DO 50 I=1,220
```

```
      M=34359738367
      CALL RANSET(M.ISEED)
      WRITE (6,INI
C
      MAXTIM=T2/T1
      IF (IGO.NE.7) GO TO 52
      DISTAB(1)=0.
         DO 51 I=2,NB
           DISTAB(I)=DISTAB(I-1)+SQRT((BY(I)-BY(I-1))**2 +
     *          (BX(I)-BX(I-1))**2)
   51    CONTINUE
      DMAX=DISTAB(NB)
      GO TO 53
   52 DMAX=(NB-1)*PATSPC
   53 TRAT=T1/T0
      NCAT = NCRS * NSOA
      VAR=SDEV/2.
      ISTART=TIMLAT*60.+0.5
      ISTOP=(TIMLAT+OSTH)*60.+0.5
      ISTEP=T1*60.+0.5
      ALPHA=0.5
      MAX=0
      PMAX=0.
C
         DO 55 I = 1,100
           NCT(I) = 0
           ICTRY(I) = 0
           CTMAX(I) = 0.
55       CONTINUE
C
C
C   SET-UP TABLE CONTAINING (1-NORMAL INTEGRAL) VALUES
C   WITH LIMITS OF [-15,5]
         DO 60 I=1,321
   60    TABLE(I)=ERFC(((I-1.)/16.)-15.)**TRAT
C
C
C   INPUT PROPLOSS DATA
C
      M=0
  100 READ (9,3000,END=110) (IX,JF(I),KG(I),BS(I),I=1,8)
         DO 105 I=1,8
           IF (JF(I).NE.IFREQ.OR.KG(I).NE.IGEO) GO TO 105
           M=M+1
           IF (M.GT.220) GO TO 110
           PROPLO(M)=BS(I)
  105    CONTINUE
      GO TO 100
  110 IF (NOPROP.EQ.0) GO TO 115
C
C
C   OUTPUT PROPLOSS CURVE
C
      WRITE (6,5022)
      PMAX = 0.0
      DO 113 I = 1,220
      IF (PROPLO(I) .GT. PMAX) PMAX = PROPLO(I)
113   CONTINUE
         DO 114 I = 1,220
           PLMAP(I) = PROPLO(I) / PMAX
114      CONTINUE
      WRITE (6,5023) PMAX
      CALL WSHOW(PLMAP,220,4)
```

```
  115 CALL CHOLES(NB,IFREQ,ALPHA,BETA,BX,BY,ETA1,ETA2)
C
C
C   BEGIN MONTE CARLO REPITITIONS LOOP
C
         DO 500 IREP=1,NREPS
C
C
C   SELECT CATEGORY FOR THIS REP BY A WEIGHTED RANDOM
C
           IF (NCRS .EQ. 0) GO TO 149
           CTST = URAND(0.)
           PCTSUM = 0.0
              DO 116 ICAT = 1,NCAT
                PCTSUM = PCTSUM + PCAT(ICAT)
                IF (CTST .LE. PCTSUM) GO TO 117
116           CONTINUE
117        ICOL = MOD(ICAT-1, NSOA) + 1
           ICTRY(ICAT) = ICTRY(ICAT) + 1
           RICAT = FLOAT(ICAT)
           ROWD = (RICAT - 1.) / FLOAT(NSOA)
           IROW = INT(ROWD) + 1
C
C
C   DETERMINE CHOSEN CATEGORY CHARACTERISTICS - -
C        SPEED, COURSE AND STANDARD DEVIATIONS OF EACH
C
           COURSE = CRSINT + (IROW-1)*ABS(CRSWND)*2.
           SPD = SOAINT + (ICOL-1)*ABS(CRSWND)*2.
           IF (SOAWND) 120,120,121
120        SDSPD = SOAWND
           GO TO 125
121        SDSPD = 0.5 * SOAWND
125        IF (CRSWND) 126,126,127
126        SDCRSE = CRSWND
           GO TO 130
C
C
C   CONVERT THESE CHARACTERISTICS TO RADIANS
C
127        SDCRSE = 0.5 * CRSWND
130        COURSE = 0.0174532925 * COURSE
           SDCRSE = 0.0174532925 * SDCRSE
149        CONTINUE
C
C
C   SELECT RANDOM STARTING LOCATION, COURSE AND SPEED
C   FROM USER-SELECTED TPA.
C
         CALL GENATE(DIMX,SDDIMX,RDIMX)
         CALL GENATE(DIMY,SDDIMY,RDIMY)
         CALL GENATE(COURSE,SDCRSE,RCRSE)
         CALL GENATE(SPD,SDSPD,RSPD)
         IF (RSPD.LT.0.1) RSPD=0.1
C
C
C   CONVERT CHOSEN COURSE AND SPEED TO MOTION IN
C   RECTANGULAR COORDINATES.
C
         DX=RSPD * SIN(RCRSE)
         DY=RSPD * COS(RCRSE)
         PSIC=VAR*ANRMRN(0.)
         RDIMX = RDIMY + TIMLAT * DY
```

```
C
C   COMPUTE CORRELATION QUANTITIES - -
C        PSIC, BR1 AND BR2.
C
            DO 150 I=1,NB
              BR1(I)=0.
              BR2(I)=0.
              RAN1(I)=ANRMRN(0.)*VAR
              RAN2(I)=ANRMRN(0.)*VAR
              PREV(I)=0.
                 DO 150 J=1,MAXTIM
  150       BW(I,J)=1.
            DO 170 I = 1, NB
                 DO 160 J = 1, NB
                   BR1(I) = BR1(I) + ETA1(I,J) * RAN1(J)
                   BR2(I) = BR2(I) + ETA2(I,J) * RAN2(J)
  160       CONTINUE
         DELTA(I) = FOM + BR1(I) + BR2(I) + PSIC
  170       CONTINUE
C
C
C   RESET TIME COUNTERS - -
C        INDEX, JTIM AND TIMMON.
C
            INDEX = 0
            JTIM = TIMMON
            MON = 0
C
C
C   FOR THIS REP, SELECT RANDOM DETECTION PROBABILITY USING
C   THRESHOLD FOR THE SELECTED CATEGORY.
C
            PTRN = URAND(0.0)
            IF (NCRS.GT.0) PTRN=URAND(0.0)*(1.-THRSH(ICAT))+THRSH(ICAT)
C
C        BEGIN THE TIME LOOP
C
            DO 400 ITIM=ISTART,ISTOP,ISTEP
C
C
C   COMPUTE PRESENT BUOY CONFIGURATION UP UNTIL
C   RELATIVE ELAPSED TIME, JDEX.
C
            JDEX=(ITIM-ISTART)/ISTEP+1
            ITMIN=JDEX
            IF (MAXTIM .LT. JDEX) ITMIN = MAXTIM
            TIME=JDEX*T1
C
            CALL MONITR(BX,BY,LM,NM,NB,ISTEP,JTIM,TIMMON,TIME,ACFTGS,
     *                        PATSPC,RF,DMAX,MON,IGO,DISTAB)
C
C
C   COMPUTE TARGET LOCATION
C
            TX=RDIMX+ITIM*DX
            TY=RDIMY+ITIM*DY
            PROD=1.
C
C
C   UPDATE TIME WINDOW INDEX
C
            INDEX=INDEX+1
```

```
C     BEGIN THE BUOY LOOP
C
               DO 300 IBUOY=1,NB
C
C
C   IF BUOY IS OUT OF RF RANGE OF AIRCRAFT,
C   OR NOT PRESENTLY BEING MONITORED,
C   USE PREVIOUS DETECTION PROBABILITY FOR THIS BUOY.
C
               PR=PREV(IBUOY)
               IF (LM(IBUOY).NE.0) GO TO 180
               BW(IBUOY,INDEX) = 1.0
               GO TO 300
C
C
C   CONVERT RANGE FROM MILES TO KILOYARDS FOR
C   PROPLOSS TABLE LOOK-UP.
C
  180          RANGE=SQRT((TX-BX(IBUOY))*(TX-BX(IBUOY))+
     *                (TY-BY(IBUOY))*(TY-BY(IBUOY)))
               IRANGE = RANGE + RANGE
               IF (IRANGE.GT.220) IRANGE=220
C
C
C   COMPUTE NEW TIME WINDOW ELEMENT
C
               DEL=(DELTA(IBUOY)-PROPLO(IRANGE))/VAR
               IDEL=(DEL+15.)*16.+1.5
               IF (IDEL.LT.1) IDEL=1
               IF (IDEL.GT.321) IDEL=321
               BW(IBUOY,INDEX)=TABLE(IDEL)
C
C
C   COMPUTE PRODUCT OVER ENTIRE TIME WINDOW
C
               PR=1.
                  DO 200 I=1,ITMIN
  200             PR=PR*BW(IBUOY,I)
               PR=1.-PR
               IF (PREV(IBUOY).GT.PR) PR = PREV(IBUOY)
               PREV(IBUOY)=PR
C
  300          PROD=PROD*(1.-PR)
            PROD=1.-PROD
            IF (NCRS .EQ. 0) GO TO 400
C
C
C  COUNT REPS WITH SUCCESSFUL DETECTIONS
C
            CTMAX(ICAT) = AMAX1(CTMAX(ICAT),PROD)
  400       PROB(JDEX)=PROB(JDEX)+PROD
         IF (PROD .LT. PTRN) NCT(ICAT) = NCT(ICAT) + 1
  500    CONTINUE
C
C       END OF LOOPS
C       RENORMALIZE PROB OVER TIME
C
         DO 600 I=1,JDEX
         PROB(I)=PROB(I)/NREPS
         IF (PROB(I).LE.PMAX) GO TO 600
         MAX=I
         PMAX=PROB(I)
```

```
C   PRODUCE PLOT [illegible] DETECTION PROBABILITY VS. TIME
C
      CALL WSHOW(PROB,MAX,20)
      WRITE (6,5021) PMAX
C
C          OUTPUT THE CATEGORY RESULTS
C
      IF (NCRS .EQ. 0) GO TO 699
      WRITE (6,5030)
      IWR = 0
C
C
C   CATEGORY DESCRIPTIONS
C
      HD2 = CRSINT - ABS(CRSWND)
         DO 601 IROW = 1,NCRS
           HD1 = HD2
           HD2 = HD1 + 2.*ABS(CRSWND)
           SOA2 = SOAINT - ABS(SOAWND)
              DO 601 ICOL = 1,NSOA
                IWR = IWR + 1
                SOA1 = SOA2
                SOA2 = SOA1 + 2.*ABS(SOAWND)
                WRITE (6,5031) IWR, HD1, HD2, SOA1, SOA2
  601    CONTINUE
      NCAT = NCRS * NSOA
C
C
C   RENORMALIZE CATEGORY DETECTION PROBABILITIES
C
      PSTDNM = 0.
         DO 602 IWR = 1,NCAT
           RMISS = FLOAT(NCT(IWR))
           RTRY = FLOAT(ICTRY(IWR))
           IF (RTRY .EQ. 0.) GO TO 602
           PSTDNM = PSTDNM + (RMISS/RTRY)*PCAT(IWR)
  602    CONTINUE
C
C
C   COMPUTE NEW THRESHOLDS AND POSTERIOR DISTRIBUTION
C   FOR EACH CATEGORY.
C
      WRITE (6,5032)
C
         DO 610 IWR = 1,NCAT
           RMISS = FLOAT(NCT(IWR))
           RTRY = FLOAT(ICTRY(IWR))
           PSTR = 0.
           IF(RMISS .EQ. 0.) GO TO 608
           CTMAX(IWR) = CTMAX(IWR) / RMISS
           CTMAX(IWR) = CTMAX(IWR)*(1.-THRSH(IWR)) + THRSH(IWR)
           IF (RTRY .EQ. 0.) GO TO 608
           PSTR = ((RMISS/RTRY)*PCAT(IWR))/PSTDNM
  608    WRITE (6,5033) IWR,PCAT(IWR),NCT(IWR),ICTRY(IWR),PSTR
  610    CONTINUE
C
      WRITE (6,5034) (THRSH(I), I=1,NCAT)
      WRITE (6,5035) (CTMAX(I), I=1,NCAT)
C
C
C   INPUT PATTERN MODIFICATION PARAMETERS
C
```

```
 699  WRITE (6,5001)
      WRITE (6,5010)
      READ (5,5025) ANSWER
      IF (ANSWER .NE. YES  ) STOP
  700 WRITE (6,5011)
      READ (5,5026,ERR=700) CK
         DO 821 I = 1,NB
           BX(I) = BXI(I) * CK
           BY(I) = BYI(I) * CK
  821    CONTINUE
      PATSPC = PATSPC * CK
  701 WRITE (6,5012)
      READ (5,5026,ERR=701) ROT
  702 WRITE (6,5013)
      READ (5,5027,ERR=702) TRANX, TRANY
      CALL ROTATE (NB, BX, BY, ROT, TRANX, TRANY)
      WRITE (6,5028)
      GO TO 30
C
C
 3000 FORMAT (8(I3,2I1,F5.1))
 5000 FORMAT (//20A2//)
 5001 FORMAT (//)
 5002 FORMAT (1X,15A4)
 5003 FORMAT ('X',2I9,2I1,2F8.1,I2,3F8.2,I2,2F10.6)
 5004 FORMAT (10F7.1,2I5)
 5005 FORMAT (2(I5,2F10.5))
 5006 FORMAT (16F5.4)
 5007 FORMAT (2I3,2F7.1,F6.2)
 5008 FORMAT (16F5.1)
 5010 FORMAT (' MODIFY PATTERN AND RE-RUN (Y OR N): ')
 5011 FORMAT (' SPACING MULTIPLICATION FACTOR: ')
 5012 FORMAT (' PATTERN ROTATION (DEG.): ')
 5013 FORMAT (' PATTERN TRANSLATION IN MILES (X,Y): ')
 5019 FORMAT (//' BUOY POSITIONS:')
 5020 FORMAT ('   BUOY',I3,2F10.2)
 5021 FORMAT (/' CUMULATIVE PROBABILITY OF DETECTION=',F7.2)
 5022 FORMAT (/' PROBABILITY OF DETECTION VS. DISTANCE'//)
 5023 FORMAT (' MAXIMUM CORRESPONDS TO PROPLOSS OF',F8.2)
 5025 FORMAT (A1)
 5026 FORMAT (F10.0)
 5027 FORMAT (2F10.0)
 5028 FORMAT (//' RE-RUN BEGINS.'/)
 5030 FORMAT (//30X,' DESCRIPTION OF CATEGORIES'/1X,'CATEGORY',
     1  10X,'HEADING INTERVAL',10X,'SOA INTERVAL')
 5031 FORMAT (2X,I4,15X,F5.0,' - ',F5.0,11X,F5.0,' - ',F5.0)
 5032 FORMAT (1X,'CATEGORY',10X,'PRIOR',10X,'MISSES',10X,'TRIES',
     1      10X,'POSTERIOR')
 5033 FORMAT (2X,I4,11X,F7.5,10X,I5,10X,I5,11X,F7.5)
 5034 FORMAT (10X,'OLD THRESHOLD VALUES = ',12F10.5)
 5035 FORMAT (10X,'NEW THRESHOLD VALUES = ',12F10.5)
      END
```

```
      SUBROUTINE WPRNOS
C
      DIMENSION ICD2(4),CD2(2),CD3(10),ICD3(2),ICD6(2),
     1CD6(2),BX(32),BY(32)
      DIMENSION ICD4(2),CD4(4),PCAT(100)
      DIMENSION COMENT(15),PNAME(7)
      CALL OBEY('EQUATE 10 INPUT',4)
      DATA PNAME/4HDF44,4HDF56,4HDF88,4HBTAC,4HCIRC,3HARC,4HSPEC/
      DATA YES/1HY/,BEARNG/1HB/
   80 CONTINUE
      WRITE(6,300)
  300 FORMAT(' ENTER THE FREQUENCY INDEX FOR YOUR PROPLOSS. ')
      READ(5,*,ERR=80) ICD2(3)
   81 CONTINUE
      WRITE(6,301)
  301 FORMAT(' ENTER THE SOURCE-RECEIVER GEOMETRY INDEX. ')
      READ(5,*,ERR=81) ICD2(4)
 8133 CONTINUE
      WRITE (6,3333)
 3333 FORMAT(' ENTER .5 CORRELATION RANGE (MILES). ')
      READ (5,*,ERR=8133) CORR50
      BETA=ALOG(2.)/CORR50
   82 CONTINUE
      WRITE(6,302)
  302 FORMAT(' ENTER THE FIGURE OF MERIT.')
      READ(5,*,ERR=82) CD3(10)
   83 CONTINUE
      WRITE(6,303)
  303 FORMAT(' ENTER YOUR PATTERN TYPE.  YOU HAVE',
     1' THE FOLLOWING CHOICES:')
      WRITE(6,304) (PNAME(I),I=1,7)
      READ(5,200,ERR=83)PATYPE
  304 FORMAT(7(20X,A4/))
      WRITE(6,888)PATYPE
  888 FORMAT(' PATYPE - ',A4)
      DO 195 IGO = 1,7
      IF (PNAME(IGO).EQ.PATYPE) GO TO 307
  195 CONTINUE
      WRITE (6,210)
  210 FORMAT(' INVALID PATTERN TYPE.')
      GO TO 83
  307 CONTINUE
      NB=16
      ICD6(1)=16
      ICD6(2)=16
      IF(PATYPE.EQ.PNAME(5))GO TO 126
      IF(PATYPE.NE.PNAME(7)) GO TO 60
   97 CONTINUE
      WRITE(6,305)
  305 FORMAT(' ENTER THE NUMBER OF BUOYS TO BE DROPPED',
     1' 32 IS THE MAXIMUM ALLOWABLE.')
      READ(5,*,ERR=97)ICD6(1)
      NB=ICD6(1)
      ICD6(2)=ICD6(1)
      DO 151 IBUOY=1,NB
  130 CONTINUE
      WRITE (6,150) IBUOY
  150 FORMAT (' ENTER X,Y COORDINATES OF BUOY',I4)
```

```
  151 CONTINUE
      GO TO 61
   60 CONTINUE
      ICD6(1)=16
      NB=16
      ICD6(2)=16
  103 FORMAT(' ENTER THE TRUE PATTERN BEARING      ')
  104 FORMAT(' ENTER PATTERN SPACING      ')
  106 FORMAT(' ENTER PATTERN RADIUS.      ')
  107 FORMAT(' DO YOU WANT TO SPECIFY PATTERN SPACING AS A ',
     1'FRACTION OF CZ?')
  112 FORMAT(' ENTER THE CZ WIDTH. ')
  108 FORMAT(' ENTER THE FRACTION OF CZ.   X.X   ')
  111 FORMAT(' ENTER THE  CZ RANGE.      ')
  200 FORMAT(A4)
      IWRFLG=-1
      ICZFLG=0
      DO 101 ITY=1,3
      IF(PATYPE.EQ.PNAME(ITY))IWRFLG=1
  101 CONTINUE
   70 CONTINUE
      WRITE(6,103)
      READ(5,*,ERR=70)PBEAR
      WRITE(6,107)
      READ(5,201)ANSWER
  201 FORMAT(A1)
      IF(ANSWER.EQ.YES)ICZFLG=IWRFLG
      IF(ICZFLG)50,20,10
  121 CONTINUE
   50 WRITE(6,112)
      READ(5,*,ERR=121)CZR
  122 CONTINUE
      WRITE(6,108)
      READ(5,*,ERR=122)FRAC
      PATSPC=FRAC*CZR
      GO TO 30
  123 CONTINUE
   10 WRITE(6,111)
      READ(5,*,ERR=123)CZR
  124 CONTINUE
      WRITE(6,108)
      READ(5,*,ERR=124)FRAC
      PATSPC=FRAC*CZR
      GO TO 30
  125 CONTINUE
   20 WRITE(6,104)
      READ(5,*,ERR=125)PATSPC
   30 CONTINUE
      IF(PATYPE.NE.PNAME(5).AND.PATYPE.NE.PNAME(6))GO TO 61
  126 CONTINUE
      WRITE(6,106)
      READ(5,*,ERR=126) RADIUS
      PATSPC=.3901806*RADIUS
   61 CONTINUE
      WRITE(6,306)
  306 FORMAT(' ENTER E/B TO SPECIFY AN ELLIPSE/BEARING BOX.')
      READ(5,201,ERR=61)TYPE
      IF(TYPE.EQ.BEARNG)GO TO 308
   84 CONTINUE
      WRITE(6,309)
  309 FORMAT(' ENTER SEMI-MAJOR AXIS LENGTH.')
      READ(5,*,ERR=84)SMAL
```

```
  310 FORMAT(' ENTER SEMI-MINOR AXIS LENGTH.')
      READ(5,*,ERR=85)SMIL
      CD3(5)=SMAL/2.0
      CD3(6)=SMIL/2.0
      GO TO 311
   86 CONTINUE
  308 WRITE(6,312)
  312 FORMAT(' ENTER BEARING BOX LENGTH.')
      READ(5,*,ERR=86)SMAL
      CD3(5)=-SMAL/2.0
   87 CONTINUE
      WRITE(6,313)
  313 FORMAT(' ENTER BEARING BOX HALF WIDTH.')
      READ(5,*,ERR=87)CD3(6)
      CD3(6)=CD3(6)/2.0
  311 CONTINUE
      IF(PATYPE.EQ.PNAME(7))GO TO 344
      IF(PATYPE.NE.PNAME(5).AND.PATYPE.NE.PNAME(6))GO TO 342
  344 BPAT=0.
      RPAT=0.
      TBEAR=90.
      GO TO 89
  342 WRITE(6,314)
  314 FORMAT(' ENTER TRUE BEARING,RANGE FROM SPA CENTER TO PATTERN',
     1' CENTER.')
      READ(5,*,ERR=311)BPAT,RPAT
      BPAT=3.1415927*BPAT/180.
      PATX=RPAT*SIN(BPAT)
      PATY=RPAT*COS(BPAT)
   88 CONTINUE
      WRITE(6,315)
  315 FORMAT(' ENTER THE TRUE BEARING OF THE SPA.')
      READ(5,*,ERR=88)TBEAR
   89 CONTINUE
      WRITE(6,316)
  316 FORMAT(' ENTER ON-STATION TIME IN HOURS.')
      READ(5,*,ERR=89)CD2(2)
   90 CONTINUE
      WRITE(6,317)
  317 FORMAT(' ENTER A/C GROUND SPEED IN KNOTS.')
      READ(5,*,ERR=90)CD6(1)
   91 CONTINUE
      WRITE(6,318)
  318 FORMAT(' ENTER TIME LATE.')
      READ(5,*,ERR=91)CD3(9)
   92 CONTINUE
      WRITE(6,319)
  319 FORMAT(' DO YOU WISH TO SPECIFY TARGET MOTION CATEGORIES?')
      READ(5,201,ERR=92)ANSWER
      IF(ANSWER.NE.YES)GO TO 93
   94 CONTINUE
      WRITE(6,321)
  321 FORMAT(' ENTER THE NUMBER OF COURSE CATEGORIES',
     1' (NO MORE THAN 10).')
  324 FORMAT(' ENTER THE NUMBER OF SOA CATEGORIES',
     1' (NO MORE THAN 10).')
      READ(5,*,ERR=94)ICD4(1)
   95 CONTINUE
      WRITE(6,322)
  322 FORMAT(' ENTER THE CENTER COURSE OF THE FIRST COURSE WINDOW.')
```

```
   96 CONTINUE
      WRITE(6,323)
  323 FORMAT(' ENTER THE HALF WIDTH OF THE COURSE WINDOWS.')
      READ(5,*,ERR=96)CD4(2)
  131 CONTINUE
      WRITE(6,324)
      READ(5,*,ERR=131)ICD4(2)
  132 CONTINUE
      WRITE(6,325)
  325 FORMAT(' ENTER THE CENTER SOA OF THE FIRST SOA WINDOW.')
      READ(5,*,ERR=132)CD4(3)
  133 CONTINUE
      WRITE(6,326)
  326 FORMAT(' ENTER THE HALF WIDTH OF THE SOA WINDOWS.')
      READ(5,*,ERR=133)CD4(4)
      NCAT=ICD4(1)*ICD4(2)
      WRITE(6,331)
  331 FORMAT(20X,' DESCRIPTION OF CATEGORIES')
      WRITE(6,332)
  332 FORMAT(2X,' CATEGORY',10X,'HEADING-INTERVAL',13X,
     1'SOA INTERVAL')
      NCRS=ICD4(1)
      NSOA=ICD4(2)
      HD2=CD4(1)-CD4(2)
      IWR=0
      DO 330 IROW=1,NCRS
      HD1=HD2
      HD2=HD1+2.0*CD4(2)
      SOA2=CD4(3)-CD4(4)
      DO 431 ICOL=1,NSOA
      IWR=IWR+1
      SOA1=SOA2
      SOA2=SOA1+2.0*CD4(4)
      WRITE(6,432)IWR,HD1,HD2,SOA1,SOA2
  432 FORMAT(4X,I4,15X,F5.0,' - ',F5.0,11X,F5.0,' - ',F5.0)
  431 CONTINUE
  330 CONTINUE
      WRITE(6,328)
  328 FORMAT(' ENTER THE CATEGORY PROBABILITIES.  THEY',
     1' SHOULD TOTAL 1.')
      DO 329 I=1,NCAT
  134 CONTINUE
      WRITE(6,333)I
  333 FORMAT(' CATEGORY ',I3)
      READ(5,*,ERR=134) PCAT(I)
  329 CONTINUE
      GO TO 135
   93 CONTINUE
      CD3(3)=0.
      CD3(7)=-180.
      WRITE(6,334)
  334 FORMAT(' ENTER THE LOWER SOA LIMIT.')
      READ(5,*,ERR=93)SLOW
  136 CONTINUE
      WRITE(6,335)
  335 FORMAT(' ENTER THE UPPER SOA LIMIT.')
      READ(5,*,ERR=136)SUP
      CD3(8)=(SUP-SLOW)/2.0
      CD3(4)=SLOW+CD3(8)
      CD3(8)=-CD3(8)
  135 CONTINUE
```

```
  336 FORMAT(' DO YOU WANT OUTPUT OF BUOY COORDINATES?')
      READ(5,201,ERR=135)ANSWER
      CALL PATGEN(PATYPE,PATX,PATY,PBEAR,PATSPC,
     1RADIUS,NB,BX,BY,CD3(9),CD4(3),CZR,TBEAR,CD4(1))
      IF(ANSWER.NE.YES)GO TO 137
      WRITE(6,337)  (BX(IWR),IWR=1,NB)
      WRITE(6,338)  (BY(IWR),IWR=1,NB)
  337 FORMAT('0 BX = '/4(2X,F8.2))
  338 FORMAT('0 BY = '/4(2X,F8.2))
  137 CONTINUE
      WRITE(6,339)
  339 FORMAT(' ENTER A COMENT TO IDENTIFY YOUR RUN.')
      READ(5,340)  (COMENT(I),I=1,15)
  340 FORMAT(15A4)
      WRITE(10,341)  (COMENT(I),I=1,15)
  341 FORMAT(1X,15A4)
  140 CONTINUE
      WRITE(6,806)
  806 FORMAT(' ENTER THE NUMBER OF REPETITIONS.')
      READ(5,*,ERR=140)ICD2(2)
      CD2(1)=90.
      CD3(1)=0.
      CD3(2)=0.
      CD6(2)=180.
      NOPROP=0
      T0=.25
      T1H=.05
      T2=.50
      ICD2(1)=1357931
      WRITE (10,800) ICD2,CD2,NOPROP,T0,T1H,T2,IGO,BETA,CORR50
  800 FORMAT('X',2I9,2I1,2F8.1,I2,3F8.2,I2,2F10.6)
      ICD3(1)=1
 8012 WRITE (6,8010)
 8010 FORMAT(' ENTER SONAR EQ. STANDARD DEVIATION.')
      READ (5,*,ERR=8012) ICD3(2)
      WRITE(10,801) CD3,ICD3
  801 FORMAT(10F7.2,2I5)
      WRITE(10,802) ICD4(1),(CD4(IWR),IWR=1,2),ICD4(2),
     1(CD4(IWR),IWR=3,4)
  802 FORMAT(2(I5,2F10.5))
      IF(ICD4(1).EQ.0)GO TO 138
      WRITE(10,803)  (PCAT(I),I=1,NCAT)
  803 FORMAT(16F5.3)
      WRITE(10,804)
  804 FORMAT('            ')
  138 CONTINUE
      WRITE(10,804)
      WRITE (10,805) ICD6,CD6,PATSPC
  805 FORMAT(2I3,2F7.1,F6.2)
      WRITE (10,807)  (BX(IB),IB=1,NB)
  807 FORMAT(16F5.1)
      WRITE (10,807)  (BY(IB),IB=1,NB)
      WRITE (6,152)
  152 FORMAT (20X,'THE INPUT FILE HAS BEEN CREATED  ')
      ENDFILE 10
      REWIND 10
      RETURN
      END
```

```
      SUBROUTINE GENATE (X,Y,Z)
      IF(Y)1,4,2
C  UNIFORM DISTRIBUTION OVER (X+Y,X-Y)

    1 R=URAND(DUMMY)
      Z=X - Y*(2.*R-1.)
      RETURN
C  NORMAL DISTRIBUTION N(X,Y)
    2 Z=ANRMRN(DUMMY)
      Z=X+Z*Y
      RETURN
C  OPERATOR SPECIFIED VALUE
    4 Z=X
      RETURN
      END
```

```
      SUBROUTINE RANORM(R)
      R=ANRMRN(R)
      RETURN
      END
```

```
C****************************
C*
C*      SHOW:      PRODUCES PLOTTED OUTPUT
C*
C*            ARRAY 'PROB' IS PLOTTED FROM 1 TO 'MAX'
C*
C****************************
      SUBROUTINE WSHOW(PROB,MAX,ID)
      DIMENSION PROB(220),LOSS(51)
      DATA IB,IX/1H ,1HX/
         DO 200 I=1,51
  200    LOSS(I)=IB
  350 WRITE (6,2)
      WRITE (6,5)
      J=0
         DO 120 I=2,MAX,2
           VAL=50.*PROB(I)+.5
           K = INT(VAL) + 1
           LOSS(K)=IX
           J=J+1
           IF (J .NE. 10) GO TO 130
           J=I/ID
           WRITE (6,3) J,LOSS
           J=0
           GO TO 120
  130      WRITE (6,4) LOSS
  120      LOSS(K)=IB
      Q=MAX
      R=ID
      Q=Q/R
      WRITE (6,6) Q
      WRITE (6,5)
      WRITE (6,2)
      RETURN
    2 FORMAT(7X,'0   .1   .2   .3   .4   .5   .6   .7   .8   .9   1')
    3 FORMAT(1X,I5,1X,51A1)
    4 FORMAT(7X,51A1)
    5 FORMAT(7X,51('.'))
    6 FORMAT('+',F5.1)
      END
```

```
      REAL FUNCTION URAND(SAME)
      COMMON/MIRNG/RAN(10),GEN(10),NWRD,BASE,MOD,FBASE,FMOD
      DIMENSION SUM(10)
      INTEGER RAN,GEN,BASE,CARRY,SUM,PROD,HPROD
      DO 30 IS=1,NWRD
   30 SUM(IS)=0
      DO 1 IG=1,NWRD
      N2=NWRD-IG+1
      DO 1 IR=1,N2
      IS=IR+IG-1
      PROD=RAN(IR)*GEN(IG)
      HPROD=PROD/BASE
      LPROD=PROD-HPROD*BASE
      SUM(IS)=SUM(IS)+LPROD
      IF(IS.LT.NWRD) SUM(IS+1)=SUM(IS+1)+HPROD
    1 CONTINUE
      N2=NWRD-1
      DO 5 IS=1,N2
      CARRY=SUM(IS)/BASE
      SUM(IS)=SUM(IS)-CARRY*BASE
      SUM(IS+1)=SUM(IS+1)+CARRY
    5 CONTINUE
      SUM(NWRD)=SUM(NWRD)-MOD*(SUM(NWRD)/MOD)
      DO 20 IS=1,NWRD
   20 RAN(IS)=SUM(IS)
      FRAN=SUM(1)
      DO 10 IS=2,NWRD
   10 FRAN=FRAN/FBASE+SUM(IS)
      FRAN=FRAN/FMOD
      URAND=FRAN
      RETURN
      END
```

```
      SUBROUTINE ROTATE(NB,BX,BY,ROT,TRANX,TRANY)
C*****************************************************
C* ANGLES IN THIS SUBROUTINE ARE MEASURED COUNTER-*
C* CLOCKWISE FROM THE POSITIVE X AXIS IN CONTRAST *
C* TO "TRUE" BEARINGS THAT ARE USED IN THE OTHER  *
C* SUBROUTINES.                                   *
C*****************************************************
      DIMENSION BX(32),BY(32)
      PI=3.1415926
      IF(ROT.EQ.0.0) GO TO 200
      DO 100 I=1,NB
      R=SQRT(BX(I)*BX(I)+BY(I)*BY(I))
      IF(BX(I).NE.0.)GO TO 120
      IF(BY(I).GE.0.)A=PI/2.
      IF(BY(I).LT.0.)A=-PI/2.
      GO TO 110
  120 A=ATAN(BY(I)/BX(I))
      IF(BX(I).LT.0.)A=PI+A
  110 A=A-(PI*ROT/180.)
      BX(I)=R*COS(A)
      BY(I)=R*SIN(A)
  100 CONTINUE
  200 DO 300 I=1,NB
      BX(I)=BX(I)+TRANX
      BY(I)=BY(I)+TRANY
  300 CONTINUE
      RETURN
      END
```

```
SUBROUTINE RANUMB(U)
COMMON/RANNO/NORAND
%NORAND IS THE COUNT OF RANDOM NUMBER PICKS
NORAND=NORAND+ 1
U=URAND(U)
RETURN
END
```

```
      FUNCTION EXPRN(DUMMY)
      I=0
  100 X=URAND(X)
  105 Y=URAND(Y)
      IF(X.LT.Y) GO TO 120
  110 X=URAND(X)
      IF(X.LT.Y) GO TO 105
  115 I=I+1
      GO TO 100
  120 EXPRN=X+I
      RETURN
      END
```

```
      SUBROUTINE RANSET (MAXINT,NSTRT)
      COMMON/MIRNG/RAN(10),GEN(10),NWRD,BASE,MOD,FBASE,FMOD
      INTEGER RAN,GEN,BASE,CARRY,REM
      MAXI=MAXINT/4
      IB=0
      BASE=1
   99 IF(BASE.GT.MAXI) GO TO 100
      BASE=BASE*4
      IB=IB+1
      GO TO 99
  100 BASE=2**IB
      FBASE=BASE
      NWRD=47/IB+1
      REM=47-IB*(NWRD-1)
      MOD=2**REM
      FMOD=MOD
      DO 101 N=1,10
      RAN(N)=0
  101 GEN(N)=0
      GEN(1)=5
      DO 200 I=1,14
      CARRY=0
      DO 190 N=1,NWRD
      GEN(N)=GEN(N)*5+CARRY
      CARRY=0
      IF(GEN(N).LT.BASE) GO TO 190
      CARRY=GEN(N)/BASE
      GEN(N)=GEN(N)-BASE*CARRY
  190 CONTINUE
  200 CONTINUE
      NSTART=NSTRT
      IF(NSTART.LE.0) NSTART=2001
      NSTART=2*(NSTART/2)+1
      DO 300 N=1,NWRD
      NTEMP=NSTART/BASE
      RAN(N)=NSTART-NTEMP*BASE
  300 NSTART=NTEMP
      RETURN
      END
```

```
C****************************
C*
C*    ERFC(X)  RETURNS  1- (INTEGRAL OF NORMAL DENSITY)
C*
C*             OR   1 - PHI(X)
C*
C*      POLYNOMIAL APPROX. FROM HANDBOOK OF MATHEMATICAL FUNCTIONS
C*      SECTION 26.2.18  (1964 EDITION)
C*
C****************************
      FUNCTION ERFC(X)
      DATA C1,C2,C3,C4/.196854,.115194,.000344,.019527/
      X2=ABS(X)
      ERFC=.5/(1.0+C1*X2+C2*X2*X2+C3*X2**3+C4*X2**4)**4
      IF (X.LT.0.) ERFC=1-ERFC
      RETURN
      END
```

```
      FUNCTION ANRMRN(DUMMY)
    1 R=URAND(R)
      IF(R.GT.0.8638) GO TO 10
      ANRMRN=2.*(URAND(X)+URAND(Y)+URAND(Z)-1.5)
      RETURN
   10 IF(R.GT.0.9745) GO TO 20
      ANRMN=1.5*(URAND(X)+URAND(Y)-1.0)
      RETURN
   20 IF(R.GT.0.9973020) GO TO 100
   25 X=6.*URAND(X)-3.0
      Y=0.358*URAND(X)
      XSQ=X*X
      GX=17.49731*EXP(-XSQ*.5)
      AX=ABS(X)
      IF(AX.GT.1.0) GO TO 30
      IF(Y.GT.(GX-17.44392+4.735703**XSQ+2.157875*AX)) GO TO 25
      ANRMRN=X
      RETURN
   30 AX3=2.367851*(3-AX)**2
      IF(AX.GT.1.5) GO TO 40
      IF(Y.GT.(GX-AX3-2.157875*(1.5-AX))) GO TO 25
      ANRMRN=X
      RETURN
   40 IF(Y.GT.(GX-AX3)) GO TO 25
      ANRMRN=X
      RETURN
  100 X=SQRT(9+2*EXPRN(X))
      IF(URAND(X).GT.3/X) GO TO 100
      IF(URAND(X).GT.0.5) X=-X
      ANRMRN=X
      IF(ABS(X).GT.3.) GO TO 1
      RETURN
      END
```

```
C****************************
C*
C*    CHOLES:      PERFORMS CHOLESKY DECOMPOSITION
C*                 TO GET ETA1 AND ETA2 MATRICES
C*
C****************************
      SUBROUTINE CHOLES(NB,IFREQ,ALPHA,BETA,BX,BY,ETA1,ETA2)
      DIMENSION BX(32),BY(32),ETA(32,32),ETA1(32,32),ETA2(32,32)
         DO 100 I=1,NB
            DO 50 J=1,NB
            ETA(I,J)=ALPHA
   50       ETA1(I,J)=0.
         ETA(I,I)=1.
  100    ETA1(I,1)=ETA(I,1)
         DO 251 K=2,NB
         K1=K-1
            DO 150 J=1,K1
  150       ETA1(K,K)=ETA1(K,K)+ETA1(K,J)*ETA1(K,J)
         ETA1(K,K)=SQRT(ETA(K,K)-ETA1(K,K))
         K2=K+1
         IF (K2 .GT. NB) GO TO 251
            DO 250 I=K2,NB
               DO 200 J=1,K1
  200          ETA1(I,K)=ETA1(I,K)+ETA1(I,J)*ETA1(K,J)
  250       ETA1(I,K)=(ETA(I,K)-ETA1(I,K))/ETA1(K,K)
  251    CONTINUE
         DO 350 I=1,NB
            DO 300 J=1,NB
            RANGE=SQRT((BX(I)-BX(J))*(BX(I)-BX(J))+
     *                        (BY(I)-BY(J))*(BY(I)-BY(J)))
            ETA(I,J)=EXP(-BETA*RANGE)
  300       ETA2(I,J)=0.
  350    ETA2(I,1)=ETA(I,1)
         DO 501 K=2,NB
         K1=K-1
            DO 400 J=1,K1
  400       ETA2(K,K)=ETA2(K,K)+ETA2(K,J)*ETA2(K,J)
         ETA2(K,K)=SQRT(ETA(K,K)-ETA2(K,K))
         K2=K+1
         IF (K2 .GT. NB) GO TO 501
            DO 500 I=K2,NB
               DO 450 J=1,K1
  450          ETA2(I,K)=ETA2(I,K)+ETA2(I,J)*ETA2(K,J)
  500       ETA2(I,K)=(ETA(I,K)-ETA2(I,K))/ETA2(K,K)
  501 CONTINUE
      RETURN
      END
```

```
C*********************************
C*
C*     MONITR ROUTINE:  RETURNS LIST OF BUOYS TO BE MONITORED
C*
C*        CALCULATES WHICH BUOYS HAVE BEEN DROPPED, AND ARE IN
C*        RANGE, OR SELECTS SET OF BUOYS TO MONITOR AFTER ALL
C*        HAVE BEEN DROPPED.
C*
C*********************************
      SUBROUTINE MONITR(BX,BY,LM,NM,NB,ISTEP,JTIM,TIMMON,TIME,ACFTGS,
     *                  PATSPC,RF,DMAX,MON,IGO,DISTAB)
      DIMENSION BX(32),BY(32),LM(32),DISTAB(32)
      DATA NL/1/
C
C        CALCULATE DISTANCE TRAVELED BY AIRCRAFT
      DIST=TIME*ACFTGS
C        COMPARE WITH TOTAL DISTANCE TO COMPLETE PATTERN
      IF (DIST.GT.DMAX) GO TO 25
C
C         SPECIAL CASE FOR USER SPECIFIED PATTERN IN WHICH THE
C         PATTERN SPACING MAY NOT BE CONSTANT
      IF (IGO.NE.7) GO TO 15
         DO 5 I=NL,NB
C           COMPARE DIST WITH CUMULATIVE DISTANCE TO BUOY I
           IF (DISTAB(I).GT.DIST) GO TO 10
    5    CONTINUE
   10 NL=I-1
      GO TO 16
C           NORMAL CASE:  BUOY SPACING IS CONSTANT (PATSPC APART)
   15 NL=DIST/PATSPC+1
   16 DIST1=(NL-1)*PATSPC
C           DO LINEAR INTERPOLATION FOR POSITION BETWEEN BUOYS
      RATIO=(DIST-DIST1)/PATSPC
      PX=BX(NL)+RATIO*(BX(NL+1)-BX(NL))
      PY=BY(NL)+RATIO*(BY(NL+1)-BY(NL))
C
C           BEGIN BUOY LOOP, CHECKING RANGE TO EACH BUOY
         DO 20 I=1,NB
         LM(I)=0
         IF (I.GT.NL) GO TO 20
         DX=PX-BX(I)
         DY=PY-BY(I)
         DIST=SQRT(DX*DX+DY*DY)
         IF (DIST.LE.RF) LM(I)=1
   20    CONTINUE
      RETURN
C
C     CASE 2:  ALL BUOYS DROPPED.  SELECT A SET TO MONITOR
C
   25 JTIM = JTIM + ISTEP
      IF (JTIM .LE. TIMMON) RETURN
C    SELECT NEW SET
      DO 30 I = 1, NB
30    LM(I) = 0
      IF (NM .LE. 0) GO TO 50
      DO 40 I = 1, NM
      INDEX = MON + I
      IF (INDEX .GT. NB)  INDEX = INDEX - NB
      LM(INDEX) = 1
40    CONTINUE
      MON = MON + NM
      IF (MON .GT. NB) MON = MON - NB
```

```
      SUBROUTINE PATGEN(PATYPE,PATX,PATY,PBEAR,PATSPC,RADIUS,
     1NB,BX,BY,TIMLAT,SPDI,CZR,TBEAR,CUSI)
      DIMENSION BX(32),BY(32)
      DIMENSION PNAME(7)
      DATA PNAME/4HDF44,4HDF56,4HDF88,4HBTAC,4HCIRC,3HARC,4HSPEC/
C     *********************************************************
C     THIS SUBROUTINE BUILDS A PATTERN OF THE INDICATED TYPE WITH
C     THE FOLLOWING SPECIFICS:
C     CENTER:        PATX,PATY
C     PATTERN BEARING:       PBEAR
C     TPA BEARING:      TBEAR
C     SPACING:       PATSPC
C     RADIUS:        RADIUS (FOR CIRCULAR OR ARC PATTERNS)
C     NOTE THAT THE PATTERN IS BUILT SO THAT IT IS PERPENDICULAR
C     TO THE HEADING PATHD THAT IS CALCULATED BELOW.
C     ****************************************************************
      IGO=0
      DO 72 I=1,7
      IF(PATYPE.EQ.PNAME(I))IGO=I
   72 CONTINUE
      PI=3.1415927
C     (XP,YP) = FINAL COORDINATES OF THE PATTERN CENTER
C     PATHD = FINAL PATTERN HEADING RELATIVE TO STANDARD TPA
C     TBPC = TRUE BEARING OF THE PATTERN CENTER
C     THETA = REFERENCE ANGLE TO X-AXIS FOR ANGLE TBPC
C     TR = TPA ROTATION REQUIRED TO REACH STANDARD POSITION
C     DIST = DISTANCE BETWEEN PATTERN CENTER AND TPA CENTER
      HSPC=PATSPC*SIN(PI/3.)
      DIST=SQRT(PATX*PATX+PATY*PATY)
      IF(PATX.NE.0.) GO TO 70
      IF(PATY.GE.0.) TBPC=0.
      IF(PATY.LT.0.) TBPC=180.
      GO TO 71
   70 THETA = ATAN(PATY/PATX)
      IF (PATX.GT.0.) TBPC = (0.5)*PI - THETA
      IF (PATX.LT.0.) TBPC = 1.5*PI - THETA
   71 IF (TBEAR.LE.180.) TR=90. - TBEAR
      IF (TBEAR.GT.180.) TR = 270. - TBEAR
      CUSI=CUSI+TR
      PATHD = PBEAR - 90. + TR
      IF(IGO.EQ.5.OR.IGO.EQ.6) PATHD = PATHD + 90.
      TR = (TR*PI)/180.
      XP=DIST*SIN(TBPC+TR)
      YP=DIST*COS(TBPC+TR)
      IF (IGO.LE.0) GO TO 100
      GO TO 200
  100 WRITE(6,101)
  101 FORMAT(' ',10X,'ERROR...UNIDENTIFIABLE PATTERN TYPE.
     1A 5-6-5 DISTRIBUTED FIELD PATTERN HAS BEEN ASSUMED.')
      IGO=2
  200 GO TO (10,20,30,40,50,60,300),IGO
C     CODE FOR DF44 PATTERN. IGO=1
   10 DO 11 I=1,4
      BX(I)=-1.75*PATSPC + (I-1)*PATSPC
      BX(I+8)=BX(I)
      BY(I)=-1.5*HSPC
      BY(I+8)=.5*HSPC
```

```
      J=9-I
      BX(J)=BX(I)+.5*PATSPC
      BX(J+8)=BX(J)
      BY(J)= -.5*HSPC
   11 BY(J+8) =1.5*HSPC
      GO TO 300
C     CODE FOR DF56 PATTERN.  IGO=2
   20 DO 21 I=1,5
      BX(I)= -2.*PATSPC + (I-1)*PATSPC
      BX(I+11)=BX(I)
      BY(I)= - HSPC
   21 BY(I+11) = HSPC
      DO 22 I=1,6
      BX(12-I)=-2.5*PATSPC + (I-1)*PATSPC
   22 BY(12-I)=0.
      GO TO 300
C     CODE FOR DF88 PATTERN.  IGO=3
   30 DO 31 I=1,8
      BX(I)= -3.75*PATSPC + (I-1)*PATSPC
      BX(17-I)=BX(I) + .5*PATSPC
      BY(I)= -.5*HSPC
   31 BY(I+8) = .5*HSPC
      GO TO 300
C     CODE FOR BTAC PATTERN.  IGO=4
   40 DO 41 I=1,16
      BX(I)= -7.5*PATSPC + (I-1)*PATSPC
   41 BY(I)=0.
      GO TO 300
C     CODE FOR CIRCULAR PATTERN.  IGO=5
   50 IF (RADIUS.GT.0) GO TO 51
      RADIUS = TIMLAT*SPDI-CZR
      IF (RADIUS.LE.0.) RADIUS=TIMLAT*SPDI + CZR
      WRITE(6,102) RADIUS
  102 FORMAT(' ',10X,'ERROR...  CIRCULAR OR ARC PATTERN HAS NON-POSITIVE
     1RADIUS.  A RADIUS OF ',F10.5,' HAS BEEN USED.')
   51 THETA=(2.*PI)/16.
      DO 52 I=1,16
      ANGLE=(I-8.5)*THETA
      BX(I)=RADIUS*COS(ANGLE)
   52 BY(I)=RADIUS*SIN(ANGLE)
      GO TO 300
C     CODE FOR ARCP.  IGO=6
   60 IF(RADIUS.GT.0.) GO TO 61
      RADIUS=TIMLAT*SPDI-CZR
      IF(RADIUS.LE.0.)RADIUS=CZR+TIMLAT*SPDI
      WRITE(6,102)RADIUS
   61 THETA=2*ARSIN((.5*PATSPC)/RADIUS)
      DO 62 I=1,16
      ANGLE=(I-8.5)*THETA
      BX(I)=RADIUS*SIN(ANGLE)
   62 BY(I)=RADIUS*COS(ANGLE)
  300 CALL ROTATE(16,BX,BY,PATHD,XP,YP)
      RETURN
      END
```