A COMPUTER SIMULATION OF ASW INTERACTIONS

by

Charles Donald Griffin

United States
Naval Postgraduate School

THESIS

A COMPUTER SIMULATION OF ASW INTERACTIONS

by

Charles Donald Griffin, Jr.

Thesis Advisor: A. F. Andrus

March 1971

Approved for public release; distribution unlimited.

A Computer Simulation of ASW Interactions

by

Charles Donald Griffin, Jr.
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1962

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1971

ABSTRACT

The mcdel presented in this thesis is a computer simulation model of ASW interactions between a formation of high value group ships, protected by some screening ships, and some penetrating submarines. The model is designed for use as an aid in improving the ability of a proposed screening tactic in the detection of a penetrating submarine. A systematic procedure to improve a screen's effectiveness against a known submarine threat is demonstrated, and an example problem is worked using this procedure.

TABLE OF CONTENTS

I. INTRODUCTION

In the present era of a continuing and growing Sino-Soviet submarine threat, the problem of effective anti-submarine warfare (ASW) to protect a high value group of shipping is of considerable interest to the Naval Officer. In recent years a great deal of effort has been expended in an attempt to develop an optimal solution to the ASW problem; but, due primarily to the complexities of the environment and the resulting inability to quantitize parameters, no such optimal solution has been achieved. It is not the purpose of this paper to provide another solution technique for the ASW problem, but rather to provide a computer simulation model of an ASW action and to suggest one possible use for the model.

The model presented in this paper is designed for use as an aid in the design of ASW screens by providing measures of effectiveness of a screen in the detection of penetrating submarines, the number of attacks made by penetrating submarines, and the number of successful attacks made by a penetrating submarine. Using the computer simulation model, the effectiveness of a simple ASW screen against penetrating submarines is maximized within the constraint limitations provided.

II. DESCRIPTION

The model is a computer simulation of the ASW environment, utilizing an event store procedure. The model is written in FORTRAN IV for the IBM 360/67 of the U. S. Naval Postgraduate School. A copy of the program is contained in Appendix I.

Before beginning the model description, it is desirable to define a few terms commonly used in the model.

A replication is defined as one complete ASW action. The action begins with the initial courses, speeds, and locations set by the user for the ships and submarine and proceeds until termination of the action.

A run is a collection of one or more replications, each of which has the same initial values of initial courses, speeds, and locations of ships and submarine. The variation among replications in any run are caused by the randomly generated occurances within each replication.

A high value group is a group of one or more ships to be protected by escort ships from submarine attack.

A screen is defined as a barrier formed by a group of escort ships, distributed about the high value group, whose purpose is to protect the high value group from submarine attack. In both actuality and the model there are two general screen types, the random patrol area

screen and the fixed station screen. In the random patrol area screen, screening ships are assigned a circular patrol area with a specified radius and a center that is fixed at a relative range and bearing from the high value group. The screening ships randomly patrol this area at some speed greater than that of the high value group. In the fixed station screen, the screening ships are assigned a fixed station relative to the high value group and maintain this station throughout the replication.

CPA is defined as the closest point of approach of a submarine to a screening ship.

Detection is defined as occurring when a penetrating submarine reaches a point within the detection range of a screening ship.

A formation is defined as the group of ships that make up both the high value group and its screen. The center of a formation is referred to as ZZ and may or may not be occupied by a ship.

The model is designed to simulate an ASW environment. The scenario consists of a formation, proceeding along a predetermined track and being attacked by submarines attempting to penetrate the screen in specific places. The entire ASW action takes place sufficiently far in the first quadrant of an X-Y grid to ensure positive locations at all times in the model. The ability of the individual screen ship to detect the penetrating submarines is determined by a detection range that is uniquely determined for each screen ship in each replication. If a random patrol area screen is used, the ability

of a screen ship to detect a submarine is also influenced by the location of the screen ships within their patrol area; i.e., at the extremities of the area, the screening ship may be several thousand yards closer or farther away from a penetrating submarine than is the center of the ship's patrol area. To use the model, the user must supply certain data consisting primarily of the initial locations, courses and speeds of all ships, and some curve of detection range vs. probability of detection for the individual ship's sensors. The specific data are placed on control cards in a standard format and a new set of control cards is used in each run. The flow of simulation log is contained in Figure 1.

There are five events in the model. These are:

1. A course change by a screening ship. This event, called whenever a screening ship reaches the edge of its patrol area, generates a new course for the screen ship to continue its random patrolling.

2. A detection by a screening ship. In this event the location of the detecting screen ship, detected submarine, and ZZ are noted at the time of detection.

3. A CPA by a screening ship. At CPA, the time; CPA range; screen ship and submarine identitites; and locations of screen ship, submarine, and ZZ are stored in a vector for reproduction at the end of the run.

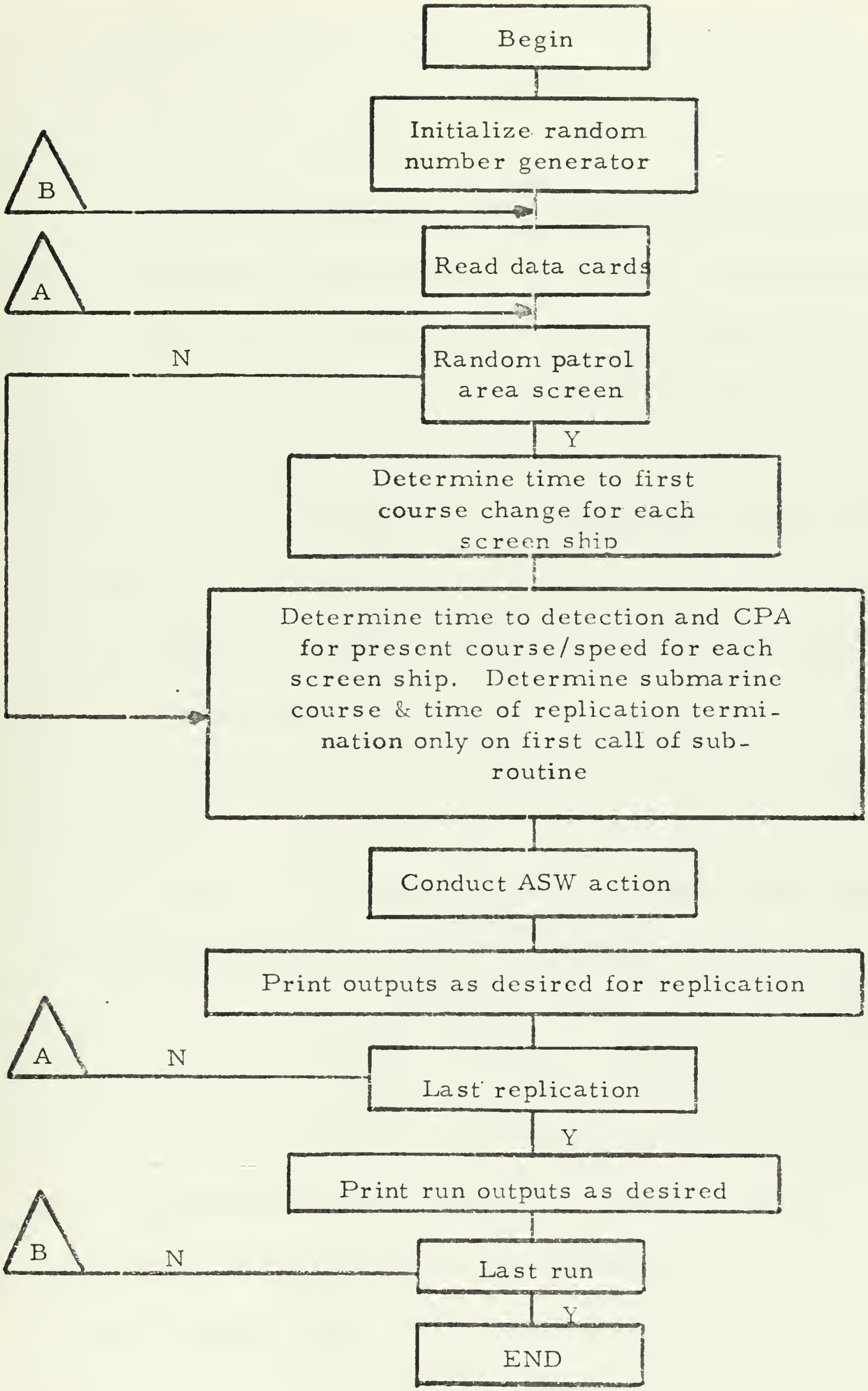
Begin
Initialize random number generator
B
Read data cards
A
N
Random patrol area screen
Y
Determine time to first course change for each screen ship
Determine time to detection and CPA for present course/speed for each screen ship. Determine submarine course & time of replication termination only on first call of subroutine
Conduct ASW action
Print outputs as desired for replication
A
N
Last replication
Y
Print run outputs as desired
B
N
Last run
Y
END

Figure 1: Simulation Logic Flow

4. An attack by the submarine on a high value group ship. The time and location of the submarine and attacked ship for each attack are noted in this event.

5. Termination of the replication. This event locates all screening ships, submarines, and high value group ships at replication termination. The mode of termination, either by successful submarine penetration or by a sufficient number of screen ship detections on the submarine, is indicated.

The measures of effectiveness for a specified formation computed in the model are:

1. The number of attacks on high value group ships made by penetrating submarines.

2. The number of times the penetrating submarines are detected by screen ships.

3. The number of successful penetrations made by the submarine.

These measures may be used to measure the effectiveness of a screen. The expression "effectiveness of the screen" refers to the probability of the screen ships detecting the penetrating submarines and, at the same time, detecting the submarine sufficiently far from ZZ to prevent attacks on the high value group ships. Clearly, a screen may be designed that will maximize the probability of detection of the penetrating submarine; but, with the necessity of minimizing the number of attacks made on high value group ships as an added constraint, the problem of designing an effective screen is complex.

With the measures of effectiveness computed in the model, the user may evaluate the relative effect of changes in parameters; e.g., changing the radius of the random patrol area on the effectiveness of the screen.

It should be noted that the model does not consider the aspect of screen ship attacks on detected submarines. There are two reasons for this. The first reason is that the inclusion of screen ship attacks on detected submarines is not considered germaine to the measurement of screen effectiveness as previously defined. The second reason is that the inclusion of screen ship attacks would greatly complicate the model, adding to both time and size requirements.

III. LIMITATIONS OF THE MODEL

The limitations of the model fall into the areas of dimension limitations and ship maneuvering limitations. A more complete description of these limitations is contained in the following paragraphs.

A. DIMENSION LIMITATIONS

The model is limited to certain maximum numbers of ships and replications in each run to preclude using excessive computer storage. These limitations are:

1.	Screen ships	10
2.	Submarines	10
3.	High value group ships	10
4.	Replications	20

B. MANEUVERING LIMITATIONS

1. Screening ships are constrained at all times to remain within their assigned screening stations. In the event of a detection of a submarine, the screen ships do not attack but remain in their stations.

2. Submarine maneuvering is confined to determination of a course to pass under ZZ.

3. High value group ships do not maneuver but maintain course and speed throughout the replication.

IV. ASSUMPTIONS

In the model certain assumptions have been made to reduce the complexity of the simulation. These assumptions are discussed in the following paragraphs with respect to screening ships, penetrating submarines, and high value group ships. These assumptions are consistent with the philosophy of model design and the measures of effectiveness derived for the model.

A. SCREENING SHIP ASSUMPTIONS

1. Detection ranges are assumed to be Boolien in nature; detection occurs with probability one when the penetrating submarine is within this range, and detection does not occur otherwise.. The detection ranges are generated uniquely for each ship in each replication. Detection ranges are generated at the beginning of each replication using the following procedure: The model user provides some curve of detection probability vs. detection range for each sensor that is associated with the screen ships. A number is randomly drawn from a uniform distribution on [0, 1]; this number is the detection probability for the ship in that replication. The range corresponding to this detection probability for the sensor is the detection range of the ship for that replication.

2. As indicated above, the model is designed making an assumption of a uniform distribution on [0, 1] for detection probability. If the user

feels that detection probabilities are distributed according to some distribution other than uniform, the model subroutine generating screen ship detection ranges must be rewritten by the user to reflect this different distribution.

3. Screening ships remain within their assigned stations at all times in the model.

4. Screen ships maintain a constant speed throughout each run.

5. Upon detection of a penetrating submarine, the detecting ship does not attack the submarine. The detection of the submarine is noted, and the replication continues.

6. Once a screening ship has gained contact on a penetrating submarine, the interactions of that screen ship and that submarine are terminated; i.e., as the replication continues, the screen ship does not redetect the submarine or calculate a new CPA for the submarine with each new random course change of the screen ship in its patrol area.

B. SUBMARINE ASSUMPTIONS

1. The penetrating submarine conducts no evasive maneuvering in the model. This assumption corresponds to actual submarine maneuvering practices. The submarine fire control solution for a penetration yields a course to pass under ZZ. Elaborate submarine maneuvering in a screen penetration is avoided, if possible, as any maneuvering has a detrimental effect on the submarine fire control solution.

2. The penetrating submarine conducts no evasive maneuvering while penetrating the screen, nor does it attack screen ships. The submarine does conduct attacks on high value group ships when arriving within a weapon's release range set by the user for each run. These attacks allow records of the effects of submarine penetrations to be maintained.

C. HIGH VALUE GROUP ASSUMPTIONS

1. The high value group ships do not maneuver in the model but maintain the initial course and speeds assigned by the user for each run.

2. High value group ship number 1 is always located at ZZ. If the formation for the high value group does not have a ship at ZZ, high value group ship number 1 is a dummy at ZZ, and the maximum high value group size is reduced to nine.

V. INPUTS

Inputs to the model are by control cards, one set for each run. The entries on the control cards are all right adjusted; i.e., the entry values set as far to the right as possible in the column width specified. The input cards are ordered sequentially as listed below and follow the source deck directly.

A. CARD NUMBER 1

The entry values of control card 1 are integer values.

Columns	Definition	Range
1 - 4	Number of screen ships in run	1≤N≤10
5 - 8	Number of submarines in run	1≤N≤10
9 -12	Number of high value group ships in run	1≤N≤10
13-16	Number of replications in run	1≤N≤20
17-20	Number of detections of submarine by screen ships to terminate replication	1≤N≤11
21-24	Print screen ship detection range for each replication & average detection range for each ship for the run	0, No; 1, Yes
25-28	Print out CPA & locations of each ship & submarine in replication	0, No; 1, Yes
29-32	Print out locations of screen ship, submarine, & ZZ at time of a penetrating submarine detection by a screen ship	0, No; 1, Yes

Columns	Definition	Range
33-36	Minimum submarine penetration speed	0≤N
37-40	Print out locations of screen ships, submarines, & ZZ at termination of each replication	0, No; 1, Yes
41-44	Print out event calendar for each replication	0, No; 1, Yes
45-48	Print out screening ship information for each replication	0, No; 1, Yes
49-52	Print out high value group ship information for each replication	0, No; 1, Yes
53-56	Print out submarine information for each replication	0, No; 1, Yes
57-60	Submarine weapon release range (100's of yards)	
61-64	Print out locations of high value group ship & submarine at weapon release for each replication	0, No; 1, Yes

B. CARDS FOR SCREEN SHIPS

Cards for screen ships follow sequentially the first control card. These entries and those of following cards must be non-negative and may be integer or floating point. For each screen ship, there is one card with the following format:

Columns	Definition
1 - 8	Screen ship number
9 -16	Initial course (in radians)
17-24	Ship speed
25-32	Initial X-position
33-40	Initial Y-position
41-48	Sensor type
57-64	Patrol station radius
65-72	Initial X-position of center of random patrol area
73-80	Initial Y-position of center of random patrol area

C. CARDS FOR PENETRATING SUBMARINES

Cards for penetrating submarines follow directly after the cards for the screen ships. For each submarine, there is one card with the following format:

Columns	Definition
1 - 8	Submarine number
17-24	Speed ($\geq$ minimum penetration speed)
25-32	Initial X-position
33-40	Initial Y-position

D. CARDS FOR HIGH VALUE GROUP SHIPS

Cards for the high value group ships follow directly after the cards for the penetrating submarines. For each high value group ship, there is one card with the following format:

Columns	Definition
1 - 8	High value group ship number
9 -16	Course
17-24	Speed
25-32	Initial X-position
33-40	Initial Y-position

VI. OUTPUTS

The outputs of the model are of two types, general statistical outputs and specific format outputs. The general statistical outputs are always provided by the model, while the specific format outputs are provided to the user only if they are desired. To have specific format outputs, the user codes control card 1 in the appropriate column with a 1 for each specific format output desired and a 0 for those specific format outputs not desired. Examples of general statistical and specific format outputs are included in Figures 2, 3, and 4.

A. GENERAL STATISTICAL OUTPUTS

1. The number of attacks on high value group ships by submarines for each replication and the total for the run.

2. The number of detections made on the penetrating submarine by screening ships during each replication. The upper limit of this value is set on control card one for each run.

3. The number of successful penetrations to ZZ made by penetrating submarines in each replication and the total for the run.

4. Average detection range of the penetrating submarine from ZZ for each replication; average detection range and its standard deviation of the penetrating submarine from ZZ for each run.

B. SPECIFIC FORMAT OUTPUTS

1. Individual screen ship's generated detection ranges for each replication and the average generated detection range for the run.

2. CPA of each screen ship to a submarine in each replication.

3. Locations of ZZ, screen ship making detection, and the detected submarine at time of detection in each replication.

4. Locations of ZZ, screen ships, and submarines at the termination of each replication.

5. Event calendar for each replication..

6. Screen ship information at termination of each replication.

This information consists of:

a. Screen ship number

b. Course (in radians)

c. Speed

d. Ship X-position

e. Ship Y-position

f. Sensor type

g. Generated detection range

h. Patrol station radius

i. X-position of center of random patrol area

j. Y-position of center of random patrol area

k. Time of next random course change

l. Detection status of ship (1 if ship made detection, 0 otherwise)

m. Time of last position update

7. High value group ship information at termination of each replication. This information consists of:

a. High value group ship number

b. Course (in radians)

c. Speed

d. X-position

e. Y-position

f. Time of last position update

g. Time first submarine passes under ZZ

8. Submarine information at termination of each replication. This information consists of:

a. Submarine number

b. Course (in radians)

c. Speed

d. X-position

e. Y-position

f. Time of last position update

g. Number of times the submarine has been detected by different screen ships. The upper limit of this value is set on control card 1 for each run.

9. Locations of submarine and target ship at weapon release in each replication.

1. Examples of General Statistical Outputs

   a. Output for each replication

```
WEPLAUNCH          DETECTIONS          PENETRATION

     1                  1                    1
```

   b. Output for each run

```
PENETRATIONS     WEPLAUNCH     AVDET      STDDEV

     8               12        6057.6     2386.65

AVERAGE RANGE OF SUBMARINE DETECTION FROM ZZ
IN EACH REPLICATION

7765.18    8809.04    0.0    4286.71    8633.73    6465.39
```

2. Examples of Specific Format Outputs

   a. Individual screen ship's generated detection ranges for each replication & the average generated detection range for the run

```
SHIP   AVRANGE   REP 1   REP 2    REP 3   REP 4    REP 5

 1      3049.9    0.0    1097.8   0.0     3380.6   7427.6
```

   b. CPA of each screen ship to a submarine in a replication

```
SHIP   SUB   TIME   CPA    SHIPXLOC   SHIPYLOC   SUB XLOC

 3.0   1.0    73.   7245.    44238.     68293.      37006.

SUB YLOC    ZZ X LOC    ZZ Y LOC

  67857.      40000.      64319.
```

   c. Location of ZZ, screen ship & detected submarine at time of detection

```
DETECTION TIME 64 BY DD 4 ON SUBMARINE 1 AT
X=36733.00  Y=62409.00
```

Figure 2: Examples of Outputs Available in the Model

d. Locations of ZZ, screen ships, & submarines at replication termination

```
TIME      SCREEN SHIP        X-POSIT     Y-POSIT
 86            1             42180.0     72546.0
TIME    MAIN BODY SHIP       X-POSIT     Y-POSIT
 86            1             40000.0     68652.0
TIME      SUBMARINE          X-POSIT     Y-POSIT
 86            1             39910.0     68724.0
```

e. Event calendar for each replication

```
TIME      EVENT NO.     SHIP    SUB     MAIN BODY
 86           9           1      1          1
```

f. Screen ship information at termination of each replication

```
SHIP   COURSE   SPD   X-COORD   Y-COORD   SONAR
 4      1.62    15.    35434.     72029.    0.
DETRANGE   PATRAD   PATX    PATY    CCTIME   DS
 4691.3     1800.   35830.  71074.    90.     1.
XYTIME
  86.
```

g. High value group ship information at the termination of each replication

```
SHIP   COURSE   SPD   X-COORD   Y-COORD   XYTIME
  1     0.0     10.    40000.    68652.     86.0
EVT9TIME
    86.
```

Figure 3: Examples of Outputs Available in the Model

h. Submarine information at the termination of each replication

```
SHIP    COURSE    SPD    X-COORD    Y-COORD    XYTIME
 1.      1.28     7.0    39910.     68724.      86.0
TIMES DETECTED
         1.
```

i. Locations of submarine & target ship at weapons' release in each replication

```
TIME   SUB NO   SUB X    SUB Y    TARGET NO   TAR X
 74      1      37230.   67927.       1       40000.
TAR Y    LAUNCH RANGE
64654.        4000.
```

Figure 4: Examples of Outputs Available in the Model

VII. MODEL USE

As an example of the use of the ASW simulation model, the problem of detection for a screen of four ships is optimized against a penetrating submarine threat. The use of the term optimization in the context of this section refers to a search for an improved, though not necessarily optimal, solution. The screen is specified to be a random patrol area screen with the four screen ships symmetrically distributed about the formation base course in patrol areas whose centers are equally spaced. This symmetric distribution of screen ships will cover a specific arc in front of the high value group. The screen ships must be located a distance from ZZ that is at least equal to maximum submarine weapons' launch range. The submarine threat is known, with the submarine specified to pass between screen stations. The size of the main body is unspecified. (See Figure 5)

A. PROCEDURE

The procedure used for optimization in this thesis is that of response surface methodology (RSM). This procedure focuses on the class of simulations for which a single output variable exists, and the input and output variables are quantitative and approximately continuous.

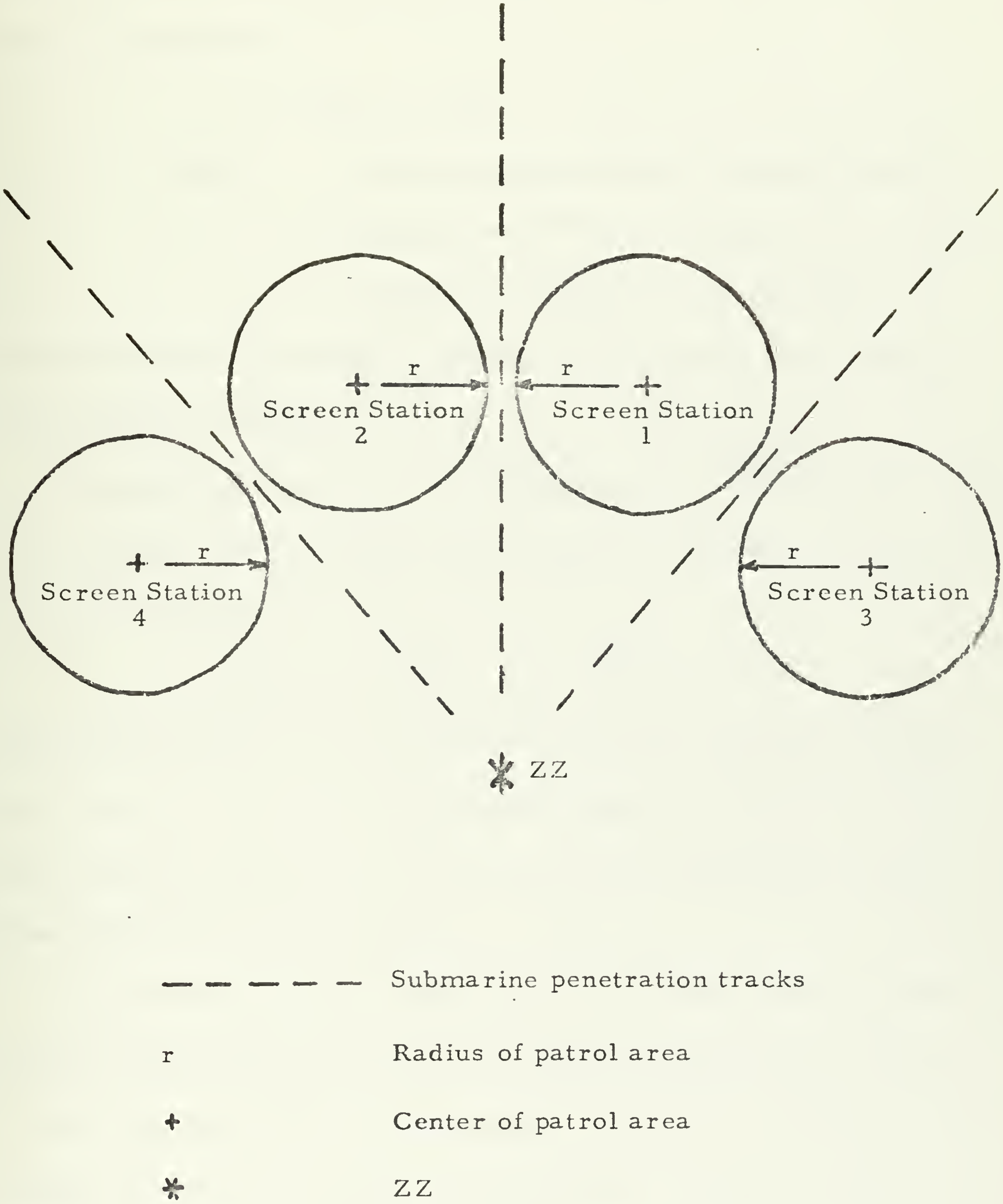

Figure 5: Four Ship Random Patrol Area Screen with Specified Submarine Penetration Tracks

If there exists a functional relationship between the quantitative output variable and the quantitative input variables, this relationship may be expressed as:

$$y = g\,(x_1, \ldots, x_k)$$

where: y denotes the quantitative response variable

x_i denotes the i^{th} input variable

g denotes the functional relationship

and the value of the response variable y lies on some k + 1 dimensional response surface.

Assuming this functional relationship g to be continuous and differentiable, we will assume that a Taylor series expansion of g can be expressed as:

$$y = g(\underline{X}) = a_0 + \sum_{1}^{k} a_i X_i + \sum_{1}^{k}\sum_{1}^{k} a_{ij} X_i X_j + \ldots$$

In the above notation, y is a measure of screen effectiveness determined for a screen by an average of the number of submarine detections over all replications in the runs; and X_i is the value of the i^{th} input parameter for the screen.

The problem using the simulation model is that we do not know the functional relationship g, yet we wish to maximize $g(\underline{X})$ subject to some set constraints; i.e., maximize y subject to maximum and minimum values for X_i. The model is used to estimate the coefficients of the above Taylor series expansion by sampling. That is, the model will be used to obtain values for y for specified values of X_i; and the technique of least squares can be used to estimate the values of the

coefficients of the Taylor series expansion. Having an approximation for the functional form of $g(\underline{X})$, it is possible, from this functional form, to estimate the gradient of g and to use this estimated gradient to determine the direction to move for further sampling in an effort to maximize $g(\underline{X})$.

If the initial samples used to determine the values of the estimators of the coefficients of the Taylor series expansion are taken from a rather restricted area, it is reasonable to make the assumption that g is linear throughout this area. That is, the functional form of $g(\underline{X})$ may be written:

$$y = g(\underline{X}) = a_0 + \sum_{1}^{k} a_i X_i$$

and that the gradient of g is:

$$(a_1, \ldots, a_k).$$

The sampling procedure is to observe at least k + 1 values of y at k + 1 different values of $\underline{X}$ and to estimate the values of $(a_0, a_1, \ldots, a_k)$ by least squares. A large number of simulation runs with different values of the X_i's may be made to determine the values of the a_i's by least squares. By considering only the largest and smallest values of the X_i's within the restricted area, there are 2^k experiments that may be performed without redundancy to determine the value of the a_i's. The minimum number of experiments that will uniquely determine the values of the a_i's in k + 1 dimensional space is k + 1. Due to the high cost of computer time, the author did not consider it necessary to use

more than k + different values of the input variables to determine the value of the a_i's.

To ease the computational complexity of the solution, the user may code each X_i with +1 for its largest value and -1 for its smallest value. By coding the input variables as $\underline{+}1$, the assumed linear function relationship g is transformed to a new linear functional relationship of the form:

$$y = b_0 + \sum_{1}^{k} b_i x_i$$

where

$$x_i = \frac{X_i - \Delta^+}{\Delta^-}$$

$$\Delta^+ = \frac{\text{maximum } X_i + \text{minimum } X_i}{2}$$

$$\Delta^- = \frac{\text{maximum } X_i - \text{minimum } X_i}{2}$$

and

$$b_0 = a_0 + \sum_{1}^{k} a_i \Delta^+$$

$$b_i = a_i \Delta^-$$

By using k + 1 experiments, a unique solution for the estimated values of b_0 through b_k is obtained by the simultaneous solution of k + 1 linear equations.

The estimates of b_o through b_k obtained by sampling are then used as estimates of the gradient of the response surface. By using the gradient as a guide, one expects to proceed to an improved solution

$$y = b_o + w \overset{k}{\underset{1}{}} b_i 2$$

where y denotes the predicted response along the path

$(wb_1, wb_2, \ldots, wb_k)$

where w is a positive constant to indicate we are moving in the direction of increasing values of y. That is, we shall observe a new value of y at the point $(wb_1, wb_2, \ldots wb_k)$ and hope that the observed value of y will be close to the predicted value of y using $\hat{y}$ above. It should be noted that with increasing w, the accuracy of the predicted response may become unreliable unless the unknown functional relationship is, in fact, linear; and, thus, this path should be followed with caution. As long as observed values for y continue to increase along the path, one can continue. If the linear approximation for the unknown functional relationship has seemed satisfactory along the path, the appearance of stationary or decreasing values for y as the path is continued to be followed indicates that the linear approximation and gradient is calculated. The new path indicated by this gradient is followed until either the optimal solution is reached, or a constraint is reached; or a new linear approximation and gradient must be calculated.

B. THE PROBLEM

For the problem being considered, the screen is specified to be a random patrol area screen with the ship located on bearing 300°, 340°, 020°, and 060° relative to ZZ respectively. The screen center of each patrol area is to be no greater than 10,000 yards from ZZ and no less than 4,000 yards from ZZ. The submarine is to penetrate from 320°, 000°, and 040° relative to ZZ. The number of ships in the high value group is unspecified. (See Figure 5)

The model provides for each run the number of successful penetrations made by the submarine. Therefore, using the number of unsuccessful penetrations as a measure of screen effectiveness to optimize using RSM, one must first consider what variables in the model affect this measure of effectiveness. The variables that affect the output parameter of effectiveness y are: ship speed, radius of random patrol area, the range of the center of the random patrol area from ZZ, ship detection range and the location of the ship within its random patrol area. The first three variables, ship speed, radius of patrol area, and the range of the center of the patrol area from ZZ, are variables over which the user has direct control in the model. Ship detection range and the location of the ship within its random patrol area are random within the model and, therefore, not directly controlled.

For this problem, it was decided to limit the number of computer runs to eleven, with ten replications in each run, and to optimize the screen effectiveness using no more than those runs. Screen ship speed was fixed at 15 knots with the speed of the high value group at 10 knots. Thus, there are two variables to consider in optimizing the screen: radius of the patrol area (X_1) and the range of the center of the patrol area from ZZ (X_2).

For sampling, the initial area considered was bounded by the values of 1,000 yards and 3,000 yards for X_1 and 8,000 yards and 10,000 yards for X_2. The coded variables are:

$$x_1 = \frac{X_1 - 2,000}{1,000}$$

$$x_2 = \frac{X_2 - 9,000}{1,000}$$

The initial sample points and their associated responses were:

Sample point	X_1	X_2	x_1	x_2	y
1	1,000	10,000	-1	1	0..533
2	1,000	8,000	-1	-1	0.567
3	3,000	8,000	-1	-1	0.667

The solution of the three simultaneous linear equations yielded the following linear approximation:

$$y = 0.457 + 0.017x_1 - 0.075x_2$$

Since the largest observed value of the response was 0..667, it was decided to choose w such that the predicted response would be in the neighborhood of 0.700. The w used was 41, which corresponded to y = 0.697. We, therefore, sampled y at the point

$$(41\ (0.017)\ ,\ 41\ (-0.075))$$

The data for the next observation was as follows:

Sample point	w	X_1	X_2	x_1	x_2	y	$\hat{y}$
4	41	2,700	5925	w(0.017)	w(-0.075)	0.667	0.697

The value of the observed response for this run was 0.667, a value less than predicted but still not less than the greatest previously observed value. It was decided to continue along this path to see if the solution would improve. Accordingly, a value for w of 49,

corresponding to a predicted response of 0.747, was chosen. We sampled y at the point

$$(49\ (0.017)\ ,\ 49\ (-0..075))..$$

The data for this next observation was as follows:

Sample point	w	X_1	X_2	x_1	x_2	y	$\hat{y}$
5	49	2800	5325	49(0..017)	49(-0.075)	0.767	0.747

By examination of the differences between predicted and response output for the previous runs, it appeared that the linear approximation was not valid. Nevertheless, as long as the response continued to increase, the path indicated by our approximation of the gradient of g appeared to indicate the direction to an improved solution. Continuing along the path, we next selected w = 58, a value corresponding to $\hat{y}$ = 0.800, and sampled at the point

$$(58\ (0.017)\ ,\ 58\ (-0.017)).$$

The data for this next observation was as follows:

Sample point	w	X_1	X_2	x_1	x_2	y	$\hat{y}$
6	58	3000	4650	58(0.017)	58(-0.075)	0.667	0.800

As the value of the response has decreased, it was necessary to return to the point of maximum response and calculate a new linear relationship and gradient. Again, three sample points were used with the previous highest response point as one of these. These sample points and their associated responses were:

Sample point	X_1	X_2	x_1	x_2	y
7	1800	5325	-1	1	0.733
8	2800	5325	1	1	0.733
9	2800	4325	1	-1	0.800

The solution of the simultaneous linear equations associated with these sample points yielded the following linear approximation:

$$y = 0.7665 + 0x_1 - 0.0335x_2.$$

It is of interest to note that the value of the estimator corresponding to the value of the radius of the patrol area is zero. This indicates that the value of the radius is at its optimal value and can, therefore, no longer contribute to an improved solution. A value of $x_1 = 0$ corresponding to $X_1 = 2,300$ yards was, therefore, assigned in the final runs of the model.

We decided to choose w = 30 such that the predicted response would be $\hat{y} = 0.800$. We sampled y at the point

$$(30\ (0),\ 30\ (-0.0335)).$$

The data for this next observation was as follows:

Sample point	X_1	X_2	x_1	x_2	y	$\hat{y}$
10	2300	4820	0	30(-0.0335)	0.800	0.800

It was realized that the predicted response of sample point 10 was not an improvement over the best previous value; but the value of X_2 was approaching the overall minimum constraint of 4,000 yards, and a more cautious step was taken.

Continuing along the path for the final run, a value of w = 74 corresponding to $\hat{y}$ = 0.850 was chosen. We sampled y at the point

$$(74\ (0),\ 74\ (-0.0335)).$$

The data for this final observation was as follows:

Sample point	X_1	X_2	x_1	x_2	y	$\hat{y}$
11	2300	4085	0	74(-0.0335)	0.833	0.850

With the completion of the eleven runs of the model, the optimal screen for the given constraints was found to have values of X_1 and X_2 of 2,300 yards and 4,085 yards respectively. The optimal response for these values of X_1 and X_2 was y = 0.833; this means that the optimized screen can be expected to detect a penetrating submarine prior to the submarine attacking the high value group 83.3% of the time. With additional runs, the response could possibly be improved; but, with only the variable X_2 available to vary and with its value so close to its lower limit of 4,000 yards, the improvement would probably be marginal. A summary of the above steps is contained in Figure 6.

In this sample example, a simple screen that had several constraints was considered. By using RSM, the screen variables were optimized against the threat of submarine penetration. Using the same techniques, the model may be used to optimize more complex screens with additional variables.

Run No.	X_1	X_2	x_1	x_2	$\hat{y}$	y	Comments
1	1,000	10,000	-1	1	--	0.533	Value for path of steepest ascent are:
2	1,000	8,000	-1	-1	--	0.567	(2000+17w, 9000-75w) in x_1, x_2 space
3	3,000	8,000	1	-1	--	0.667	$\hat{y}$=0.457+w(0.005914)
4	2,700	5,925	--	--	0.697	0.667	w = 41
	2,800	5,325	--	--	0.747	0.767	w = 49 point of maximum response
6	3,000	4,650	--	--	0.80	0.667	w = 58
7	1,800	5,325	-1	1	--	0.733	
8	2,800	5,325	1	1	--	0.733	$\hat{y}$=0.7665+w(0.00112225) b =0.0 in this solution
9	2,800	4,325	1	-1	--	0.80	
10	2,300	4,820	--	--	0.8	0.8	w = 30
11	2,300	4,085	--	--	0.85	0.833	w = 74

Figure 6: Summary of the Search for Improved Solution

VIII. CONCLUSIONS AND RECOMMENDATIONS

The model presented in this thesis is a simulation model of ASW interactions between a formation and penetrating submarines. A procedure has been demonstrated in which the model was used to improve the effectiveness of a proposed screening tactic against a known submarine threat.

An attempt to use the model to measure the effectiveness of various screen tactics was examined. The procedure used in this attempt was to determine the average detection range of penetrating submarines from ZZ for each screen and to compare these average ranges by the analysis of variance techniques. This procedure to rank screen effectiveness in order of average detection range from ZZ, with the screen having the greatest average detection range being the most effective, had intuitive appeal. Unfortunately, the standard deviations of the average detection ranges were so great, on the order of two to three times the differences between average detection ranges of the screens, that no statistical interferences could be made with this procedure.

In an effort to use the same general procedure, an attempt was made to compare the effectiveness of proposed screen tactics by comparing the average time until first detection of a penetrating submarine. This procedure also yielded no statistically significant results for

the same reason as the previous proposal. The results of these attempts are not included in this thesis.

The model has not been successfully used to directly compare the effectiveness of various proposed screening tactics; but it may be used as shown in the preceding chapter to maximize the effectiveness of a screen, with certain constraints, against a specific submarine threat.

APPENDIX A. COMPUTER PROGRAM

```
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SET RANDOM NUMBER GENERATOR TO ZERO
C
      CALL URN(0)
C
C     RESET ALL VECTORS FOR NEW RUN
C
   80 CALL INIT
C
C     READ IN CONTROL CARD FOR NEW RUN.
C
   81 READ(5,3499,END=9999)(MSTAT(I),I=1,16)
 3499 FORMAT(16I4)
      WRITE(6,3498)(MSTAT(I),I=1,16)
 3498 FORMAT('0',5X,16I4,//)
C
C     READ IN INITIAL VALUES FOR SCREENING SHIPS, SUBMARINES AND
C     HIGH VALUE GROUP
C
      L=MSTAT(1)
      READ(5,3500)((SV(I,J),J=1,6),SV(I,8),SV(I,9),SV(I,10),I=1,L)
 3500 FORMAT(F8.0,F8.0,F8.0,F8.0,F8.0,F8.0,F16.0,F8.0,F8.0)
      L=MSTAT(2)
      READ(5,3501)(SSV(I,1),(SSV(I,J),J=3,5),I=1,L)
 3501 FORMAT(F8.0,F16.0,F8.0,F8.0)
      L=MSTAT(3)
      READ(5,3502)((TAR(I,J),J=1,5),I=1,L)
 3502 FORMAT(5F8.0)
C
C     SET A COPY OF THE ORIGINAL DATA IN EACH VECTOR INTO ITS
C     CORRESPONDING DUMMY VECTOR. THE DUMMY VECTORS ARE USED TO HOLD
C     THE ORIGINAL COPY OF THE DATA TO RESET THE SHIP, SUBMARINE AND
C     HIGH VALUE GROUP VECTORS TO THE ORIGINAL VALUES FOR A NEW
C     REPLICATION.
C
      L=MSTAT(1)
      DO 1 I=1,L
      DO 1 J=1,15
      DSV(I,J)=SV(I,J)
    1 CONTINUE
      L=MSTAT(2)
      DO 4 I=1,L
      DO 4 J=1,15
```

```
      DSSV(I,J)=SSV(I,J)
    4 CONTINUE
      N=MSTAT(3)
      DO 2 K=1,N
      DO 2 L=1,10
      DTAR(K,L)=TAR(K,L)
    2 CONTINUE
C
C     DETERMINE TIME OF EVENT 1 FOR EACH SCREEN SHIP. EVENT 1 IS THE
C     TIME THAT AN INDIVIDUAL SCREEN SHIP REACHES THE EDGE OF ITS
C     RANDOM PATROL AREA AND A NEW COURSE MUST BE GENERATED AT RANDOM
C     THAT WILL CONTINUE THE RANDOM PATROL WITHIN THE ASSIGNED PATROL
C     AREA. STORE THE TIME FOR THE NEXT EVENT 1 IN THE EVENT CALENDAR.
C
      L=MSTAT(4)
      DO 3 M=1,L
      DO 82 I=1,10
      IF(SV(I,1).EQ.0.0)GO TO 82
      IF(SV(I,8).EQ.0.0)GO TO 82
      MTIME=(SV(I,8)-1.0)/((SV(I,3)-TAR(1,3))*(2000.0/60.0))
      SV(I,11)=FLOAT(MTIME)
      CALL SNE(MTIME,1,I,1,0)
   82 CONTINUE
C
C     CALL SUBROUTINE TO DETERMINE TIME SUBMARINE PASSES UNDER ZZ,
C     COURSE OF THE SUBMARINE AND TIME SUBMARINE ARRIVES WITHIN
C     FIRING RANGE OF A HIGH VALUE GROUP SHIP. ENTER GENERATED EVENTS
C     IN EVENT CALENDAR- EVENT 9 TIME OF PASSING UNDER ZZ, EVENT 4 TIME
C     OF SUBMARINE ARRIVING AT FIRING POINT TO ATTACK A HIGH VALUE
C     GROUP SHIP.
C
      CALL FIND(M,0,0,0,1)
C
C     TAKE NEXT EVENT FROM THE EVENT CALENDAR AND INDEX THE EVENT
C     CALENDAR
C
 4001 CALL TNE(ET,EN,EA1,EA2,EA3,&4002)
      GO TO (41,42,43,44,45,46,47,48,49),EN
   41 CALL EVENT1(ET,EN,EA1,EA2,EA3,M)
      GO TO 4001
   42 CALL EVENT2(ET,EA1,EA2,EA3,M,&4009)
      GO TO 4001
   43 CALL EVENT3(ET,EA1,EA2,EA3)
      GO TO 4001
   44 CALL EVENT4(ET,EA1,EA2,EA3)
      GO TO 4001
   45 CALL EVENT5(ET,EA1,EA2,EA3)
      GO TO 4001
```

```
   46 CALL EVENT6(ET,EA1,EA2,EA3)
      GO TO 4001
   47 CALL EVENT7(ET,EA1,EA2,EA3)
      GO TO 4001
   48 CALL EVENT8(ET,EA1,EA2,EA3)
      GO TO 4001
   49 CALL EVENT9(ET,EA1,EA2,EA3)
C
C     WRITE FOR EACH REPLICATION
C
      WRITE(6,8002)
 8002 FORMAT('0',4X,'WEPLAUNCH        DETECTIONS         PENETRATIONS',/)
      WRITE(6,8003)DESKIL,SUBKIL,MBKIL
 8003 FORMAT('0',I9,10X,I9,10X,I9)
      GO TO 4001
C
C     WRITE ONLY WHEN THE SUBMARINE SUCCESSFULLY PENETRATES THE SCREEN
C     AND PASSES UNDER ZZ.
C
 4002 WRITE(6,4003)
 4003 FORMAT('0',4X,'EVENT CALENDAR EXHAUSTED, PROBLEM COMPLETED',/)
C
C     WRITE THE FOLLOWING ONLY IF THE USER DESIRES. THE VALUES TO BE
C     PRINTED OUT ARE INDICATED ON THE CONTROL CARD BY CODING A 1 FOR
C     THOSE VALUES TO BE PRINTED AND A 0 FOR THOSE VALUES NOT DESIRED.
C
 4009 IF(MSTAT(11).EQ.1)CALL SNE(ET,99,EA1,EA2,EA3)
      IF(MSTAT(12).EQ.1)GO TO 4004
   13 IF(MSTAT(13).EQ.1)GO TO 4005
   14 IF(MSTAT(14).EQ.1)GO TO 4006
   15 IF(MSTAT(7).EQ.1)GO TO 4010
      GO TO 12
 4004 WRITE(6,4012)
      NN=MSTAT(1)
      WRITE(6,4007)((SV(I,J),J=1,13),I=1,NN)
      GO TO 13
 4005 WRITE(6,4013)
      NN=MSTAT(2)
      WRITE(6,4007)((SSV(I,J),J=1,7),I=1,NN)
      GO TO 14
 4006 WRITE(6,4014)
      NN=MSTAT(3)
      WRITE(6,4008)((TAR(I,J),J=1,7),I=1,NN)
      GO TO 15
 4010 LL=MSTAT(1)
      MM=MSTAT(2)
      WRITE(6,4015)
 4015 FORMAT('0',4X,'SHIP  SUB  TIME     CPA     SHIPXLOC  SHIPYLOC  SUB
```

```
     1XLOC   SUB YLOC   ZZ X LOC   ZZ Y LOC',//)
      WRITE(6,4011)(((CPA(I,J,K),K=1,10),J=1,MM),I=1,LL)
      GO TO 12
   12 CALL REPLIC
    3 CONTINUE
      IF(MSTAT(6).EQ.1)CALL SONDAT
      CALL FIGURE
      GO TO 80
 9999 STOP
 4007 FORMAT('0',5X,F3.0,2X,F7.3,2X,F3.0,2X,F7.0,2X,F7.0,2X,F4.0,2X,F8.1
     1,2X,F5.0,2X,F7.0,2X,F7.0,2X,F4.0,2X,F3.0,2X,F4.0,2X,//)
 4008 FORMAT('0',5X,F3.0,2X,F7.3,2X,F3.0,2X,F7.0,2X,F7.0,2X,F4.0,2X,F8.1
     1,2X,F5.0,2X,F7.0,2X,F7.0,//)
 4011 FORMAT('0',4X,F4.1,1X,F4.1,2X,F4.0,2X,F8.0,2X,F8.0,2X,F8.0,2X,F8.0
     1,2X,F8.0,2X,F8.0,2X,F8.0,//)
 4012 FORMAT('0',4X,'SHIP    COURSE   SPD   X-COORD   Y-COORD SONAR   DETRANG
     1E   PATRAD    PATX       PATY CCTIME DS   XYTIME',/)
 4013 FORMAT('0',4X,'SHIP    COURSE   SPD   X-COORD   Y-COORD XYTIME TIMES D
     1ETECTED',/)
 4014 FORMAT('0',4X,'SHIP    COURSE   SPD   X-COORD   Y-COORD XYTIME EVT9TIM
     1E',/)
      END


      SUBROUTINE INIT
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SET ALL VECTORS AND VARIABLES FOR A NEW RUN.
C
      PI=3.14159
      DO 1 I=1,10
      DO 1 J=1,15
      SV(I,J)=0.0
      SSV(I,J)=0.0
      DSV(I,J)=0.0
      DSSV(I,J)=0.0
    1 CONTINUE
      DO 2 L=1,10
      DO 2 K=1,10
      TAR(L,K)=0.0
      DTAR(L,K)=0.0
      SONAR(L,K)=0.0
    2 CONTINUE
      DO 3 M=1,200
```

```
      DO 3 N=2,5
      ECAL(M,1)=999999999
      DECAL(M,1)=999999999
      ECAL(M,N)=0
      DECAL(M,N)=0
    3 CONTINUE
      DESKIL=0
      SUBKIL=0
      MBKIL=0
      KT=0
      MAXEV=200
      DO 4 I=1,5
      TEMP(I)=0
    4 CONTINUE
      DO 5 I=1,10
      SONAV(I)=0.0
    5 CONTINUE
      DO 6 I=1,20
      MSTAT(I)=0
    6 CONTINUE
      DO 7 I=1,10
      DO 7 J=1,10
      DO 7 K=1,10
      CPA(I,J,K)=99999.99
    7 CONTINUE
      MBKILL=0
      MTARK=0
      DO 8 I=1,20
      DO 8 J=1,10
      DO 8 K=1,10
      DSTAT(I,J,K)=0.0
    8 CONTINUE
      DO 9 I=1,20
      DO 9 J=1,2
      STAT(I,J)=0.0
    9 CONTINUE
      R=1.0
      RETURN
      END


      SUBROUTINE FIND(M,ET,EA1,EA2,EA3)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20.2)
C
```

```
C     R IS USED AS AN INDEX FOR EACH REPLICATION. ON THE FIRST CALL OF
C     THIS SUBROUTINE THE SUBMARINE COURSE AND TIME TO ZZ IS
C     CALCULATED.
C     THIS SECTION ALSO CALCULATES THE TIME OF SUBMARINE ATTACKS ON
C     HIGH VALUE GROUP SHIPS.
C     ON THE FIRST CALL OF THIS SUBROUTINE, UNIQUE DETECTION RANGES ARE
C     GENERATED FOR EACH SCREEN SHIP.
C
      IF(R.GT.1.0)GO TO 10
      N=MSTAT(2)
      DO 1003 J=1,N
      IF(SSV(J,3).LT.FLOAT(MSTAT(9)))SSV(J,3)=FLOAT(MSTAT(9))
      DX=(TAR(1,4)-SSV(J,4))
      DY=(TAR(1,5)-SSV(J,5))
      IF(DX.LT.0.0.AND.DY.LT.0.0)GO TO 2
      IF(DX.GE.0.0.AND.DY.LT.0.0)GO TO 3
      IF(DX.GE.0.0.AND.DY.GE.0.0)GO TO 4
      IF(DX.LT.0.0.AND.DY.GE.0.0)GO TO 5
    2 RANG=SQRT((DX**2)+(DY**2))
      IF(DY.EQ.0.0)GO TO 1004
      ALPH=ATAN(ABS(DX/DY))
   22 PSI=((TAR(1,3)/SSV(J,3))*SIN(ALPH))
      IF(PSI.GT.1.0)GO TO 23
      GAMMA=ARSIN(PSI)
      SSV(J,2)=(PI+(ALPH+GAMMA))
C
C     CALCULATE TIME OF REPLICATION TERMINATION
C
      LTIME=(RANG/(((SSV(J,3)*ABS(COS(GAMMA)))+ABS(TAR(1,3)*ABS(COS(ALPH
     1))))*(2000.0/60.0)))
      CALL SNE(LTIME,9,1,J,1)
      IF(J.NE.1)GO TO 601
      TAR(1,7)=(FLOAT(LTIME))
  602 LTIM6=((RANG-FLOAT(100*MSTAT(15)))/(((SSV(J,3)*ABS(COS(GAMMA)))+AB
     1S(TAR(1,3)*ABS(COS(ALPH))))*(2000.0/60.0)))
      CALL SNE(LTIM6,4,EA1,J,1)
      GO TO 1008
    3 RANG=SQRT((DX**2)+(DY**2))
      ALPH=ATAN(ABS(DX/DY))
   20 PSI=((TAR(1,3)/SSV(J,3))*SIN(ALPH))
      IF(PSI.GT.1.0)GO TO 21
      GAMMA=ARSIN(PSI)
      SSV(J,2)=(PI-(ALPH+GAMMA))
C
C     CALCULATE TIME OF REPLICATION TERMINATION
C
      LTIME=(RANG/(((SSV(J,3)*ABS(COS(GAMMA)))+ABS(TAR(1,3)*ABS(COS(ALPH
     1))))*(2000.0/60.0)))
```

```
      CALL SNE(LTIME,9,1,J,1)
      IF(J.NE.1)GO TO 603
      TAR(1,7)=(FLOAT(LTIME))
  604 LTIM6=((RANG-FLOAT(100*MSTAT(15)))/(((SSV(J,3)*ABS(COS(GAMMA)))+AB
     1S(TAR(1,3)*ABS(COS(ALPH))))*(2000.0/60.0)))
      CALL SNE(LTIM6,4,EA1,J,1)
      GO TO 1008
   21 SSV(J,3)=SSV(J,3)+2.0
      GO TO 20
   23 SSV(J,3)=SSV(J,3)+2.0
      GO TO 22
 1004 ALPH=(PI/2.0)
      GO TO 22
    4 RANG=SQRT((DX**2)+(DY**2))
      IF(FLOAT(MSTAT(9)).LT.TAR(1,3).AND.SSV(J,3).LT.TAR(1,3))SSV(J,3)=T
     1AR(1,3)+2.0
      IF(DY.EQ.0.0)GO TO 24
      ALPH=ATAN(ABS(DX/DY))
   25 THETA=(PI-ALPH)
      A=(2.0*TAR(1,3)*COS(THETA))
      C=(4.0*((SSV(J,3)**2)-(TAR(1,3)**2)))
      B=(0.5*(A+SQRT((A**2)+C)))
      SSV(J,2)=ARCOS(((TAR(1,3)**2)+(SSV(J,3)**2)-(B**2))/(2.0*TAR(1,3)*
     1(SSV(J,3))))
C
C     CALCULATE TIME OF REPLICATION TERMINATION
C
      LTIME=(RANG/(B*(2000.0/60.0)))+ET
      CALL SNE(LTIME,9,EA1,J,1)
      IF(J.NE.1)GO TO 605
      TAR(1,7)=(FLOAT(LTIME))
  606 AB=(RANG-(FLOAT(100*MSTAT(15))))
      IF(AB.LT.0.0)GO TO 26
      LTIM6=(AB/(B*(2000.0/60.0)))+ET
      CALL SNE(LTIM6,4,EA1,J,1)
      GO TO 1008
   24 ALPH=((3.0*PI)/2.0)
      GO TO 25
    5 RANG=SQRT((DX**2)+(DY**2))
      IF(FLOAT(MSTAT(9)).LT.TAR(1,3).AND.SSV(J,3).LT.TAR(1,3))SSV(J,3)=T
     1AR(1,3)+2.0
      IF(DY.EQ.0.0)GO TO 28
      ALPH=ATAN(ABS(DX/DY))
   27 THETA=(PI-ALPH)
      A=(2.0*TAR(1,3)*COS(THETA))
      C=(4.0*((SSV(J,3)**2)-(TAR(1,3)**2)))
      B=(0.5*(A+SQRT((A**2)+C)))
      SSV(J,2)=((2.0*PI)-ARCOS(((TAR(1,3)**2)+(SSV(J,3)**2)-(B**2))/(2.0
```

```
     1*(TAR(1,3))*(SSV(J,3)))))
C
C     CALCULATE TIME OF REPLICATION TERMINATION
C
      LTIME=(RANG/(B*(2000.0/60.0)))+ET
      CALL SNE(LTIME,9,EA1,J,1)
      IF(J.NE.1)GO TO 607
      TAR(1,7)=(FLOAT(LTIME))
  608 AB=(RANG-(FLOAT(100*MSTAT(15))))
      IF(AB.LT.0.0)GO TO 26
      LTIM6=(AB/(B*(2000.0/60.0)))+ET
      CALL SNE(LTIM6,4,EA1,J,1)
      GO TO 1008
   28 ALPH=(PI/2.0)
      GO TO 27
   26 WRITE(6,1022)
 1022 FORMAT('0',10X,'WITHIN WEAPONS LAUNCH RANGE OF ZZ AT ZERO TIME',/)
      GO TO 1008
  601 IF(FLOAT(LTIME).LT.TAR(1,7))TAR(1,7)=FLOAT(LTIME)
      GO TO 602
  603 IF(FLOAT(LTIME).LT.TAR(1,7))TAR(1,7)=FLOAT(LTIME)
      GO TO 604
  605 IF(FLOAT(LTIME).LT.TAR(1,7))TAR(1,7)=FLOAT(LTIME)
      GO TO 606
  607 IF(FLOAT(LTIME).LT.TAR(1,7))TAR(1,7)=FLOAT(LTIME)
      GO TO 608
 1008 CONTINUE
C
C     CALCULATE TIME OF WEAPONS LAUNCH AGAINST THOSE MAIN BODY SHIPS
C     THAT ARE WITHIN WEAPONS LAUNCH RANGE OF THE SUBMARINE DURING ITS
C     PENETRATION.
C
      IF(MSTAT(3).EQ.1)GO TO 1003
      WRITE(6,1011)(MSTAT(I),I=1,20)
 1011 FORMAT(20I4)
      RDX=((TAR(1,3)*SIN(TAR(1,2)))-(SSV(J,3)*SIN(SSV(J,2))))
      RDY=((TAR(1,3)*COS(TAR(1,2)))-(SSV(J,3)*COS(SSV(J,2))))
      IF(RDX.GE.0.0.AND.RDY.GE.0.0)GO TO 401
      IF(RDX.LT.0.0.AND.RDY.GE.0.0)GO TO 402
      IF(RDX.LT.0.0.AND.RDY.LT.0.0)GO TO 403
      IF(RDX.GE.0.0.AND.RDY.LT.0.0)GO TO 404
  401 IF(RDY.EQ.0.0)GO TO 405
      RELCSE=ATAN(RDX/RDY)
      GO TO 407
  405 RELCSE=(PI/2.0)
      GO TO 407
  402 IF(RDY.EQ.0.0)GO TO 406
      RELCSE=((2.0*PI)-ATAN(ABS(RDX/RDY)))
```

```
      GO TO 407
  406 RELCSE=((3.0*PI)/2.0)
      GO TO 407
  403 RELCSE=(PI+ATAN(ABS(RDX/RDY)))
      GO TO 407
  404 RELCSE=(PI-ATAN(ABS(RDX/RDY)))
      GO TO 407
  407 RELSPD=SQRT((RDX**2)+(RDY**2))
      LL=MSTAT(3)
      DO 1009 I=2,LL
      DX=(TAR(I,4)-SSV(J,4))
      DY=(TAR(I,5)-SSV(J,5))
      RANGM=SQRT((DX**2)+(DY**2))
      IF(DX.GE.0.0.AND.DY.GE.0.0)GO  TO 408
      IF(DX.LT.0.0.AND.DY.GE.0.0)GO  TO 409
      IF(DX.LT.0.0.AND.DY.LT.0.0)GO  TO 410
      IF(DX.GE.0.0.AND.DY.LT.0.0)GO  TO 411
  408 IF(DY.EQ.0.0)GO  TO 412
      ALPH1=ATAN(DX/DY)
  413 TURN1=(ALPH1+(PI/2.0))
      TURN2=(ALPH1+((3.0*PI)/2.0))
      IF(RELCSE.GE.TURN2.OR.RELCSE.LE.TURN1)GO  TO 1003
      GAMMA2=ABS((ALPH1+PI)-RELCSE)
      POINT=RANGM*SIN(GAMMA2)
      IF((FLOAT(100*MSTAT(15)))-POINT)1003,414,414
  412 ALPH1=(PI/2.0)
      GO  TO  413
  414 AB=RANGM*COS(GAMMA2)
      BC=SQRT(((FLOAT(100*MSTAT(15)))**2)-(POINT**2))
      LTIM6=((AB-BC)/(RELSPD*(2000.0/60.0)))+ET
      CALL  SNE(LTIM6,4,0,J,I)
      GO  TO  1009
  409 IF(DY.EQ.0.0)GO  TO  415
      ALPH1=((2.0*PI)-ATAN(ABS(DX/DY)))
  416 TURN1=(ALPH1-((3.0*PI)/2.0))
      TURN2=(ALPH1-(PI/2.0))
      IF(RELCSE.LE.TURN1.OR.RELCSE.GE.TURN2)GO  TO 1003
      GAMMA2=ABS((ALPH1-PI)-RELCSE)
      POINT=RANGM*SIN(GAMMA2)
      IF((FLOAT(100*MSTAT(15)))-POINT)1003,417,417
  415 ALPH1=((3.0*PI)/2.0)
      GO  TO  416
  417 AB=RANGM*COS(GAMMA2)
      BC=SQRT(((FLOAT(100*MSTAT(15)))**2)-(POINT**2))
      LTIM6=((AB-BC)/(RELSPD*(2000.0/60.0)))+ET
      CALL  SNE(LTIM6,4,0,J,I)
      GO  TO  1009
  410 ALPH1=(PI+ATAN(ABS(DX/DY)))
```

```
      TURN1=(ALPH1+(PI/2.0))
      TURN2=(ALPH1-(PI/2.0))
      IF(RELCSE.LE.TURN1.AND.RELCSE.GE.TURN2)GO TO 1003
      GAMMA2=ABS((ALPH1-PI)-RELCSE)
      POINT=RANGM*SIN(GAMMA2)
      IF((FLOAT(100*MSTAT(15)))-POINT)1003,418,418
  418 AB=RANGM*COS(GAMMA2)
      BC=SQRT(((FLOAT(100*MSTAT(15)))**2)-(POINT**2))
      LTIM6=((AB-BC)/(RELSPD*(2000.0/60.0)))+ET
      CALL SNE(LTIM6,4,0,J,I)
      GO TO 1009
  411 ALPH1=(PI-ATAN(ABS(DX/DY)))
      TURN1=(ALPH1+(PI/2.0))
      TURN2=(ALPH1-(PI/2.0))
      IF(RELCSE.LE.TURN1.AND.RELCSE.GE.TURN2)GO TO 1003
      GAMMA2=ABS((ALPH1+PI)-RELCSE)
      POINT=RANGM*SIN(GAMMA2)
      IF((FLOAT(100*MSTAT(15)))-POINT)1003,419,419
  419 AB=RANGM*COS(GAMMA2)
      BC=SQRT(((FLOAT(100*MSTAT(15)))**2)-(POINT**2))
      LTIM6=((AB-BC)/(RELSPD*(2000.0/60.0)))+ET
      CALL SNE(LTIM6,4,0,J,I)
      GO TO 1009
 1009 CONTINUE
 1003 CONTINUE
C
C     THIS SECTION IS REACHED ONLY IN THE FIRST CALL OF THE SUBROUTINE
C
      NN=MSTAT(1)
      MM=MSTAT(2)
      DO 1C05 K=1,NN
      DO 1005 L=1,MM
      IF(SV(K,1).EQ.0.0.OR.SSV(L,1).EQ.0.0)GO TO 1005
C
C     CALL SUBROUTINE TO UNIQUELY DETERMINE DETECTION RANGE FOR EACH
C     SCREEN SHIP
C
      CALL DETECT(K,M)
      IF(SV(K,8).EQ.0.0)SV(K,11)=TAR(1,7)
C
C     DETERMINE WHEN ON PRESENT COURSE AND SPEED DETECTION OR CPA WILL
C     OCCUR.
C
      DX1=SV(K,4)-SSV(L,4)
      DY1=SV(K,5)-SSV(L,5)
      RANG1=SQRT((DX1**2)+(DY1**2))
      IF(DX1.LT.0.0.AND.DY1.LT.0.0)GO TO 6
      IF(DX1.GE.0.0.AND.DY1.LT.0.0)GO TO 7
```

```
      IF(DX1.GE.0.0.AND.DY1.GE.0.0)GO TO 8
      IF(DX1.LT.0.0.AND.DY1.GE.0.0)GO TO 9
    6 BETA=ATAN(DX1/DY1)
      GO TO 43
    7 BETA=((2.0*PI)-(ATAN(ABS(DX1/DY1))))
      GO TO 43
    8 IF(DY1.EQ.0.0)GO TO 46
      BETA=(PI+ATAN(ABS(DX1/DY1)))
      GO TO 43
    9 IF(DY1.EQ.0.0)GO TO 47
      BETA=(PI-ATAN(ABS(DX1/DY1)))
      GO TO 43
   46 BETA=((3.0*PI)/2.0)
      GO TO 43
   47 BETA=(PI/2.0)
      GO TO 43
   43 RDX=SV(K,3)*SIN(SV(K,2))-SSV(L,3)*SIN(SSV(L,2))
      RDY=SV(K,3)*COS(SV(K,2))-SSV(L,3)*COS(SSV(L,2))
      IF(RDY.EQ.0.0)GO TO 44
      THETA=ATAN(ABS(RDX/RDY))
   45 ANG1=BETA+(PI/2.0)
      ANG2=BETA-(PI/2.0)
      IF(ANG1.GE.(2.0*PI))ANG1=ANG1-(2.0*PI)
      IF(ANG2.LT.0.0)ANG2=ANG2+(2.0*PI)
      RELSPD=SQRT((RDX**2)+(RDY**2))
      IF(RDX.GE.0.0.AND.RDY.GE.0.0)GO TO 31
      IF(RDX.LT.0.0.AND.RDY.GE.0.0)GO TO 32
      IF(RDX.LT.0.0.AND.RDY.LT.0.0)GO TO 33
      IF(RDX.GE.0.0.AND.RDY.LT.0.0)GO TO 34
   31 RELCSE=THETA
      IF(BETA.GE.PI.AND.BETA.LE.((3.0*PI)/2.0))GO TO 80
      IF(BETA.GT.0.0.AND.BETA.LT.(PI/2.0))GO TO 35
      IF(BETA.GE.((3.0*PI)/2.0).AND.BETA.LE.(2.0*PI).AND.RELCSE.LT.ANG1)
     1GO TO 35
      IF(BETA.GT.(PI/2.0).AND.BETA.LT.PI.AND.RELCSE.GT.ANG2)GO TO 35
      GO TO 80
   32 RELCSE=(2.0*PI)-THETA
      IF(BETA.GE.(PI/2.0).AND.BETA.LE.PI)GO TO 80
      IF(BETA.GT.((3.0*PI)/2.0).AND.BETA.LT.(2.0*PI))GO TO 35
      IF(BETA.GT.PI.AND.BETA.LE.((3.0*PI)/2.0).AND.RELCSE.LT.ANG1)GO TO
     135
      IF(BETA.GE.0.0.AND.BETA.LT.(PI/2.0).AND.RELCSE.GT.ANG2)GO TO 35
      GO TO 80
   33 RELCSE=PI+THETA
      IF(BETA.GE.0.0.AND.BETA.LE.(PI/2.0))GO TO 80
      IF(BETA.GE.PI.AND.BETA.LE.((3.0*PI)/2.0))GO TO 35
      IF(BETA.GT.(PI/2.0).AND.BETA.LT.PI.AND.RELCSE.LT.ANG1)GO TO 35
      IF(BETA.GT.((3.0*PI)/2.0).AND.BETA.LT.0.0.AND.RELCSE.GT.ANG2)GO TO
```

```
     1 35
       GO TO 80
    34 RELCSE=PI-THETA
       IF(BETA.GE.((3.0*PI)/2.0).AND.BETA.LE.0.0)GO TO 80
       IF(BETA.GE.(PI/2.0).AND.BETA.LE.PI)GO TO 35
       IF(BETA.GT.0.0.AND.BETA.LT.(PI/2.0).AND.RELCSE.LT.ANG1)GO TO 35
       IF(BETA.GT.PI.AND.BETA.LT.((3.0*PI)/2.0).AND.RELCSE.GT.ANG2)GO TO
     135
       GO TO 80
    35 IF(BETA-RELCSE)36,37,38
    36 EPSIL=(2.0*PI)+(BETA-RELCSE)
       GO TO 39
    37 EPSIL=0.0
       GO TO 39
    38 EPSIL=(BETA-RELCSE)
       GO TO 39
    39 CPA(K,L,9)=ABS(RANG1*SIN(EPSIL))
       A=ABS(RANG1*COS(EPSIL))
       LTIM1=ABS(A/((2000.0/60.0)*RELSPD))+ET
       IF(LTIM1.GT.IFIX(SV(K,11)))GO TO 300
   301 IF(CPA(K,L,9)-SV(K,7))50,50,51
    50 C=SQRT((SV(K,7)**2)-(CPA(K,L,9)**2))
       LTIM2=ABS((A-C)/((2000.0/60.0)*RELSPD))+ET
       IF(LTIM2.LT.IFIX(SV(K,11)))CALL SNE(LTIM2,2,K,L,0)
    51 IF(MSTAT(7).EQ.0)GO TO 85
       IF(LTIM1.LT.IFIX(SV(K,11)))CALL SNE(LTIM1,3,K,L,0)
       GO TO 85
    80 CONTINUE
    85 R=R+1.0
  1005 CONTINUE
       GO TO 1006
    10 CONTINUE
C
C      THIS SECTION IS REACHED ON ALL SUBROUTINE CALLS AFTER THE FIRST
C
       IF(EA1.EQ.0.OR.EA2.EQ.0)GO TO 1006
       IF(SV(EA1,1).EQ.0.0.OR.SSV(EA2,1).EQ.0.0)GO TO 1006
       IF(SSV(EA2,3).LT.FLOAT(MSTAT(9)))SSV(EA2,3)=FLOAT(MSTAT(9))
C
C      DETERMINE TIME OF DETECTION OR CPA FOR THE SCREEN SHIP ON PRESENT
C      COURSE AND SPEED
C
C
       DX1=SV(EA1,4)-SSV(EA2,4)
       DY1=SV(EA1,5)-SSV(EA2,5)
       RANG1=SQRT((DX1**2)+(DY1**2))
       IF(DX1.LT.0.0.AND.DY1.LT.0.0)GO TO 103
       IF(DX1.GE.0.0.AND.DY1.LT.0.0)GO TO 104
```

```
      IF(DX1.GE.0.0.AND.DY1.GE.0.0)GO TO 105
      IF(DX1.LT.0.0.AND.DY1.GE.0.0)GO TO 106
  103 BETA=ATAN(DX1/DY1)
      GO TO 107
  104 BETA=((2.0*PI)-(ATAN(ABS(DX1/DY1))))
      GO TO 107
  105 IF(DY1.EQ.0.0)GO TO 125
      BETA=ATAN(DX1/DY1)
      GO TO 107
  125 BETA=((3.0*PI)/2.0)
      GO TO 107
  106 IF(DY1.EQ.0.0)GO TO 126
      BETA=ATAN(DX1/DY1)
      GO TO 107
  126 BETA=(PI/2.0)
      GO TO 107
  107 RDX=SV(EA1,3)*SIN(SV(EA1,2))-SSV(EA2,3)*SIN(SSV(EA2,2))
      RDY=SV(EA1,3)*COS(SV(EA1,2))-SSV(EA2,3)*COS(SSV(EA2,2))
      IF(RDY.EQ.0.0)GO TO 108
      GO TO 109
  108 THETA=(PI/2.0)
      GO TO 110
  109 THETA=ATAN(ABS(RDX/RDY))
  110 ANG1=BETA+(PI/2.0)
      ANG2=BETA-(PI/2.0)
      IF(ANG1.GE.(2.0*PI))ANG1=ANG1-(2.0*PI)
      IF(ANG2.LT.0.0)ANG2=ANG2+(2.0*PI)
      RELSPD=SQRT((RDX**2)+(RDY**2))
      IF(RDX.GE.0.0.AND.RDY.GE.0.0)GO TO 111
      IF(RDX.LT.0.0.AND.RDY.GE.0.0)GO TO 112
      IF(RDX.LT.0.0.AND.RDY.LT.0.0)GO TO 113
      IF(RDX.GE.0.0.AND.RDY.LT.0.0)GO TO 114
  111 RELCSE=THETA
      IF(BETA.GE.PI.AND.BETA.LE.((3.0*PI)/2.0))GO TO 115
      IF(BETA.GT.0.0.AND.BETA.LT.(PI/2.0))GO TO 116
      IF(BETA.GE.((3.0*PI)/2.0).AND.BETA.LE.(2.0*PI).AND.RELCSE.LT.ANG1)
     1GO TO 116
      IF(BETA.GT.(PI/2.0).AND.BETA.LT.PI.AND.RELCSE.GT.ANG2)GO TO 116
      GO TO 115
  112 RELCSE=(2.0*PI)-THETA
      IF(BETA.GE.(PI/2.0).AND.BETA.LE.PI)GO TO 115
      IF(BETA.GT.((3.0*PI)/2.0).AND.BETA.LT.(2.0*PI))GO TO 116
      IF(BETA.GT.PI.AND.BETA.LE.((3.0*PI)/2.0).AND.RELCSE.LT.ANG1)GO TO
     1116
      IF(BETA.GE.0.0.AND.BETA.LT.(PI/2.0).AND.RELCSE.GT.ANG2)GO TO 116
      GO TO 115
  113 RELCSE=PI+THETA
      IF(BETA.GE.0.0.AND.BETA.LE.(PI/2.0))GO TO 115
```

```
      IF(BETA.GE.PI.AND.BETA.LE.((3.0*PI)/2.0))GO TO 116
      IF(BETA.GT.(PI/2.0).AND.BETA.LT.PI.AND.RELCSE.LT.ANG1)GO TO 116
      IF(BETA.GT.PI.AND.BETA.LT.((3.0*PI)/2.0).AND.RELCSE.GT.ANG2)GO TO
     1116
      GO TO 115
  114 RELCSE=PI-THETA
      IF(BETA.GE.((3.0*PI)/2.0).AND.BETA.LE.0.0)GO TO 115
      IF(BETA.GE.(PI/2.0).AND.BETA.LE.PI)GO TO 116
      IF(BETA.GT.0.0.AND.BETA.LT.(PI/2.0).AND.RELCSE.LT.ANG1)GO TO 116
      IF(BETA.GT.PI.AND.BETA.LT.((3.0*PI)/2.0).AND.RELCSE.GT.ANG2)GO TO
     1116
      GO TO 115
  116 IF(BETA-RELCSE)117,118,119
  117 EPSIL=(2.0*PI)+(BETA-RELCSE)
      GO TO 120
  118 EPSIL=0.0
      GO TO 120
  119 EPSIL=(BETA-RELCSE)
      GO TO 120
  120 CPA(EA1,EA2,9)=ABS(RANG1*SIN(EPSIL))
      A=ABS(RANG1*COS(EPSIL))
      LTIM1=ABS(A/((2000.0/60.0)*RELSPD))+ET
      IF(LTIM1.GT.IFIX(SV(EA1,11)))GO TO 302
  303 IF(CPA(EA1,EA2,9)-SV(EA1,7))121,121,122
  121 C=SQRT((SV(EA1,7)**2)-(CPA(EA1,EA2,9)**2))
      LTIM2=ABS((A-C)/((2000.0/60.0)*RELSPD))+ET
      IF(LTIM2.LT.IFIX(SV(EA1,11)))CALL SNE(LTIM2,2,EA1,EA2,0)
  122 IF(LTIM1.LT.IFIX(SV(EA1,11)))CALL SNE(LTIM1,3,EA1,EA2,0)
      GO TO 123
  124 CONTINUE
      GO TO 123
  115 CONTINUE
  123 R=R+1.0
 1006 CONTINUE
      RETURN
  300 IF(MSTAT(7).EQ.0)GO TO 301
      CPA(K,L,1)=FLOAT(K)
      CPA(K,L,2)=FLOAT(L)
      CPA(K,L,3)=SV(K,13)
      CPA(K,L,4)=RANG1
      CPA(K,L,5)=SV(K,4)
      CPA(K,L,6)=SV(K,5)
      CPA(K,L,7)=SSV(L,4)
      CPA(K,L,8)=SSV(L,5)
      CPA(K,L,9)=TAR(1,4)
      CPA(K,L,10)=TAR(1,5)
      GO TO 301
  302 IF(MSTAT(7).EQ.0)GO TO 303
```

```
      CPA(EA1,EA2,1)=FLOAT(EA1)
      CPA(EA1,EA2,2)=FLOAT(EA2)
      CPA(EA1,EA2,3)=SV(EA1,13)
      CPA(EA1,EA2,4)=RANG1
      CPA(EA1,EA2,5)=SV(EA1,4)
      CPA(EA1,EA2,6)=SV(EA1,5)
      CPA(EA1,EA2,7)=SSV(EA2,4)
      CPA(EA1,EA2,8)=SSV(EA2,5)
      CPA(EA1,EA2,9)=TAR(1,4)
      CPA(EA1,EA2,10)=TAR(1,5)
      GO TO 303
   44 THETA=(PI/2.0)
      GO TO 45
 1010 FORMAT(10F10.4)
      END


      SUBROUTINE TNE(ET,EN,EA1,EA2,EA3,*)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SUBROUTINE PROVIDES THE NEXT SEQUENTIAL EVENT FROM THE EVENT
C     CALENDAR.
C     AFTER PROVIDING THE EVENT TO THE PROGRAM, THE EVENT IS DELETED
C     FROM THE EVENT CALENDAR BY THE INSERTION OF A VERY HIGH TIME OF
C     OCCURRANCE.
C     EACH TIME THE SUBROUTINE IS CALLED, THE EVENT CALENDAR IS
C     SEQUENTIALLY ORDERED IN ORDER OF INCREASING TIME.
C
      CALL RPLSRT(EN)
      IF(ECAL(1,1).GT.200)RETURN 1
      ET=ECAL(1,1)
      EN=ECAL(1,2)
      EA1=ECAL(1,3)
      EA2=ECAL(1,4)
      EA3=ECAL(1,5)
      ECAL(1,1)=999999999
      RETURN
      END
```

```
      SUBROUTINE DETECT(N,M)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SUBROUTINE PROVIDES A UNIQUE DETECTION RANGE FOR EACH SCREEN SHIP
C     AT THE BEGINNING OF EACH REPLICATION. THE CURVES USED ARE
C     PROVIDED BY THE USER FOR EACH SENSOR USED BY THE SCREEN SHIPS.
C     THE GENERATED VALUES OF DETECTION RANGE FOR EACH SCREEN SHIP ARE
C     STORED IN THE SONAR VECTOR FOR USE IN DETERMINING AVERAGE
C     GENERATED DETECTION RANGE FOR EACH REPLICATION.
C
      X=URN(1)
      IF(X.GT.0.5)GO TO 1
      IF(X.LE.0.5.AND.X.GT.0.3)GO TO 2
      IF(X.LE.0.3.AND.X.GT.0.15)GO TO 3
      Y=15000.0-(66666.6666*X)
      SV(N,7)=Y
      SONAR(N,M)=Y
  100 RETURN
    1 Y=0.0
      SV(N,7)=0.0
      SONAR(N,M)=0.0
      GO TO 100
    2 Y=2000.0-(10000.0*(X-0.3))
      SV(N,7)=Y
      SONAR(N,M)=Y
      GO TO 100
    3 Y=5000.0-(20000.0*(X-0.15))
      SV(N,7)=Y
      SONAR(N,M)=Y
      GO TO 100
      END


      SUBROUTINE POSIT(ET,EN,EA1,EA2,EA3,*)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SUBROUTINE LOCATES SHIPS IN X/Y COORDINATES AT ANY SPECIFIC TIME.
C
C
C     LOCATES ZZ (HIGH VALUE GROUP SHIP 1)
```

```
C
      TIME5=(FLOAT(ET)-TAR(1,6))
      A5=TAR(1,4)
      B5=TAR(1,5)
      C5=((TIME5*2000.0)/60.0)*(TAR(1,3))*(SIN(TAR(1,2)))
      D5=((TIME5*2000.0)/60.0)*(TAR(1,3))*(COS(TAR(1,2)))
      LOCX5=(A5+C5)
      LOCY5=(B5+D5)
      TAR(1,4)=LOCX5
      TAR(1,5)=LOCY5
      TAR(1,6)=FLOAT(ET)
      IF(EN.EQ.9)GO TO 1100
      IF(EA1.NE.0)GO TO 10
      IF(EA1.EQ.0)GO TO 1200
 1100 CONTINUE
C
C     LOCATES ALL SCREEN SHIPS AND SCREEN SHIP PATROL AREA CENTER
C
      L=MSTAT(1)
      DO 1017 I=1,L
 1018 TIME1=(FLOAT(ET)-SV(I,13))
      IF(TIME1.LT.0.0)TIME1=0.0
      A1=SV(I,4)
      B1=SV(I,5)
      C1=((TIME1*2000.0)/60.0)*(SV(I,3))*(SIN(SV(I,2)))
      D1=((TIME1*2000.0)/60.0)*(SV(I,3))*(COS(SV(I,2)))
      LOCX1=(A1+C1)
      LOCY1=(B1+D1)
      SV(I,4)=LOCX1
      SV(I,5)=LOCY1
      SV(I,13)=FLOAT(ET)
      AA1=(SV(I,9))
      BB1=(SV(I,10))
      CC1=((TIME1*2000.0)/60.0)*(TAR(1,3))*(SIN(TAR(1,2)))
      DD1=((TIME1*2000.0)/60.0)*(TAR(1,3))*(COS(TAR(1,2)))
      LOCX4=(AA1+CC1)
      LOCY4=(BB1+DD1)
      SV(I,9)=LOCX4
      SV(I,10)=LOCY4
 1017 CONTINUE
      IF(EN.EQ.9)GO TO 1023
      IF(EA2.NE.0)GO TO 1200
 1023 L=MSTAT(2)
      DO 1020 J=1,L
C
C     LOCATES ALL SUBMARINES
C
 1019 TIME2=(FLOAT(ET)-SSV(J,6))
```

```
      IF(TIME2.LT.0.0)TIME2=0.0
      A2=SSV(J,4)
      B2=SSV(J,5)
      C2=((TIME2*2000.0)/60.0)*(SSV(J,3))*(SIN(SSV(J,2)))
      D2=((TIME2*2000.0)/60.0)*(SSV(J,3))*(COS(SSV(J,2)))
      LOCX2=(A2+C2)
      LOCY2=(B2+D2)
      SSV(J,4)=LOCX2
      SSV(J,5)=LOCY2
      SSV(J,6)=FLOAT(ET)
 1020 CONTINUE
      IF(EN.EQ.9)GO TO 1024
      IF(EA3.NE.0)GO TO 1400
 1024 N=MSTAT(3)
      IF(N.LT.2)GO TO 1021
C
C     LOCATES ALL HIGH VALUE GROUP SHIPS EXCEPT HIGH VALUE GROUP SHIP 1
C     WHICH HAS BEEN LOCATED EARLIER IN THE SUBROUTINE.
C
      DO 1021 L=2,N
 1022 TIME3=(FLOAT(ET)-TAR(L,6))
      IF(TIME3.LT.0.0)TIME3=0.0
      A3=TAR(L,4)
      B3=TAR(L,5)
      C3=((TIME3*2000.0)/60.0)*(TAR(L,3))*(SIN(TAR(L,2)))
      D3=((TIME3*2000.0)/60.0)*(TAR(L,3))*(COS(TAR(L,2)))
      LOCX3=(A3+C3)
      LOCY3=(B3+D3)
      TAR(L,4)=LOCX3
      TAR(L,5)=LOCY3
      TAR(L,6)=FLOAT(ET)
 1021 CONTINUE
      GO TO 1400
   10 CONTINUE
C
C     LOCATES A SPECIFIC SCREEN SHIP AND CENTER OF ITS RATROL AREA.
C     THIS SECTION OF THE SUBROUTINE IS USED WHEN A SPECIFIC SCREEN SHIP
C     NUMBER IS PASSED TO THE SUBROUTINE.
C
      TIME1=(FLOAT(ET)-SV(EA1,13))
      A1=SV(EA1,4)
      B1=SV(EA1,5)
      C1=((TIME1*2000.0)/60.0)*(SV(EA1,3))*(SIN(SV(EA1,2)))
      D1=((TIME1*2000.0)/60.0)*(SV(EA1,3))*(COS(SV(EA1,2)))
      LOCX1=(A1+C1)
      LOCY1=(B1+D1)
      SV(EA1,4)=LOCX1
      SV(EA1,5)=LOCY1
```

```
      SV(EA1,13)=FLOAT(ET)
      AA1=SV(EA1,9)
      BB1=SV(EA1,10)
      CC1=((TIME1*2000.0)/60.0)*(TAR(1,3))*(SIN(TAR(1,2)))
      DD1=((TIME1*2000.0)/60.0)*(TAR(1,3))*(COS(TAR(1,2)))
      LOCX4=(AA1+CC1)
      LOCY4=(BB1+DD1)
      SV(EA1,9)=LOCX4
      SV(EA1,10)=LOCY4
 1200 CONTINUE
C
C     LOCATES A SPECIFIC SUBMARINE. THIS SECTION OF THE SUBROUTINE IS
C     USED ONLY WHEN A SPECIFIC SUBMARINE NUMBER IS PASSED TO THE
C     SUBROUTINE.
C
      IF(EA2.EQ.0)GO TO 1300
      TIME2=(FLOAT(ET)-SSV(EA2,6))
      A2=SSV(EA2,4)
      B2=SSV(EA2,5)
      C2=((TIME2*2000.0)/60.0)*(SSV(EA2,3))*(SIN(SSV(EA2,2)))
      D2=((TIME2*2000.0)/60.0)*(SSV(EA2,3))*(COS(SSV(EA2,2)))
      LOCX2=(A2+C2)
      LOCY2=(B2+D2)
      SSV(EA2,4)=LOCX2
      SSV(EA2,5)=LOCY2
      SSV(EA2,6)=FLOAT(ET)
 1300 CONTINUE
C
C     LOCATES A SPECIFIC HIGH VALUE GROUP SHIP. THIS SECTION OF THE
C     SUBROUTINE IS USED ONLY WHEN A SPECIFIC HIGH VALUE GROUP SHIP
C     NUMBER IS PASSED TO THE SUBROUTINE.
C
      IF(EA3.EQ.0)GO TO 1400
      TIME3=(FLOAT(ET)-TAR(EA3,6))
      A3=TAR(EA3,4)
      B3=TAR(EA3,5)
      C3=((TIME3*2000.0)/60.0)*(TAR(EA3,3))*(SIN(TAR(EA3,2)))
      D3=((TIME3*2000.0)/60.0)*(TAR(EA3,3))*(COS(TAR(EA3,2)))
      LOCX3=(A3+C3)
      LOCY3=(B3+D3)
      TAR(EA3,4)=LOCX3
      TAR(EA3,5)=LOCY3
      TAR(EA3,6)=FLOAT(ET)
 1400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SNE(ET,EN,EA1,EA2,EA3,*)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SUBROUTINE USED TO STORE NEW EVENTS IN THE EVENT CALENDAR IN
C     SEQUENTIAL ORDER OF INCREASING TIME OF OCCURRANCE
C
      CALL RPLSRT(EN)
      IF(EN.EQ.99)GO TO 2003
      I=1
    1 IF(ECAL(I,2).NE.EN)GO TO 2
      IF (ECAL(I,3).EQ.EA1.AND.ECAL(I,4).EQ.EA2.AND.ECAL(I,5).EQ.EA3)GO
     1TO 3
    2 IF(ECAL(I,1).GE.999999999)GO TO 3
      I=I+1
      IF(I.LE.MAXEV)GO TO 1
      WRITE(6,2001)
 2001 FORMAT(' ','WARNING EVENT CALANDER FILLED')
      RETURN 1
    3 ECAL(I,1)=ET
      ECAL(I,2)=EN
      ECAL(I,3)=EA1
      ECAL(I,4)=EA2
      ECAL(I,5)=EA3
      KT=KT+1
      DECAL(KT,1)=ET
      DECAL(KT,2)=EN
      DECAL(KT,3)=EA1
      DECAL(KT,4)=EA2
      DECAL(KT,5)=EA3
 2005 RETURN
 2003 WRITE(6,2006)
 2006 FORMAT('0',4X,'EVENT CALENDER',//,5X,'TIME',5X,'EVENT NO.',5X,'SHI
     1P',5X,'SUB',6X,'MAIN BODY',//)
      WRITE(6,2004)((DECAL(I,J),J=1,5),I=1,KT)
 2004 FORMAT('0',4X,I4,8X,I3,8X,I3,6X,I2,10X,I2)
      GO TO 2005
      END


      SUBROUTINE RPLSRT(EN)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
```

```
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     EVENT ORDERS THE EVENT CALENDAR IN SEQUENTIAL ORDER OF INCREASING
C     TIME.
C
      IF(EN.EQ.99)GO TO 200
      NPASS=MAXEV-1
      DO 17 I=1,NPASS
      NSTOP=MAXEV-I
      DO 21 J=1,NSTOP
      IF(ECAL(J,1).LE.ECAL(J+1,1))GO TO 21
      IF(ECAL(J,1).GT.ECAL(J+1,1)) GO TO 20
      IF(ECAL(J,1).EQ.ECAL(J+1,1).AND.ECAL(J+1,1).EQ.ECAL(J+2,1).AND.
     1ECAL(J+2,1).EQ.999999999)GO TO 21
      IF(ECAL(I,1).EQ.ECAL(I+1,1).AND.ECAL(I+1,1).EQ.999999999)GO TO 18
   20 DO 22 K=1,5
      TEMP(K)=ECAL(J,K)
      ECAL(J,K)=ECAL(J+1,K)
      ECAL(J+1,K)=TEMP(K)
   22 CONTINUE
   21 CONTINUE
   17 CONTINUE
      GO TO 18
  200 NPASS=KT-1
      DO 300 I=1,NPASS
      NSTOP=KT-I
      DO 301 J=1,NSTOP
      IF(DECAL(J,1).LE.DECAL(J+1,1))GO TO 301
      IF(DECAL(J,1).GT.DECAL(J+1,1))GO TO 302
  302 DO 303 K=1,5
      TEMP(K)=DECAL(J,K)
      DECAL(J,K)=DECAL(J+1,K)
      DECAL(J+1,K)=TEMP(K)
  303 CONTINUE
  301 CONTINUE
  300 CONTINUE
      GO TO 18
   18 RETURN
      END


      SUBROUTINE EVENT1(ET,EN,EA1,EA2,EA3,M)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
```

```
C     EVENT GENERATES NEW COURSES FOR SCREEN SHIPS THAT WILL KEEP THEM
C     WITHIN THEIR RANDOM PATROL AREAS. EVENT ALSO GENERATES THE TIME
C     OF THE NEXT COURSE CHANGE AND STORES THIS TIME AND SHIP NUMBER
C     IN THE EVENT CALENDAR.
C
      CALL POSIT(ET,1,EA1,EA2,EA3)
      A=TAR(1,3)
      B=SV(EA1,3)
      C=((B**2)-(A**2))
      AAA1=(SV(EA1,4)-SV(EA1,9))
      BBB1=(SV(EA1,5)-SV(EA1,10))
      SCRLOC=SQRT((AAA1**2)+(BBB1**2))
      IF(SCRLOC.GT.SV(EA1,8))GO TO 100
  102 IF(BBB1.EQ.0.0)GO TO 1
      XI=ATAN(ABS(AAA1/BBB1))
      GO TO 2
    1 XI=(PI/2.0)
      GO TO 2
    2 IF(AAA1.GE.0.0.AND.BBB1.GE.0.0)GO TO 3
      IF(AAA1.LT.0.0.AND.BBB1.GE.0.0)GO TO 4
      IF(AAA1.LT.0.0.AND.BBB1.LT.0.0)GO TO 5
      IF(AAA1.GE.0.0.AND.BBB1.LT.0.0)GO TO 6
    3 T1=(XI+((3.0*PI)/2.0))
      T2=(XI+(PI/2.0))
      XIP=XI
    7 Q=(2.0*PI)*URN(1)
      IF(Q.LT.T1.AND.Q.GT.T2)GO TO 20
      GO TO 7
   20 IF(Q.GE.PI)GO TO 500
      GAMMA=(PI-Q)
      A1=(2.0*A*COS(GAMMA))
      RS=(0.5*(A1+SQRT((A1**2)+(4.0*C))))
      DESCSE=ARCOS(((A**2)+(B**2)-(RS**2))/(2.0*A*B))
      SV(EA1,2)=DESCSE
      GO TO 501
  500 GAMMA=(Q-PI)
      A1=(2.0*A*COS(GAMMA))
      RS=(0.5*(A1+SQRT((A1**2)+(4.0*C))))
      DESCSE=((2.0*PI)-(ARCOS(((A**2)+(B**2)-(RS**2))/(2.0*A*B))))
      SV(EA1,2)=DESCSE
      GO TO 501
  501 IF(Q.GE.(XIP+PI))GO TO 21
      B3=(XIP-(Q-PI))
   22 F=(SV(EA1,8)-1.0)
      G=(2.0*SCRLOC*COS(B3))
      H=((F**2)-(SCRLOC**2))
      DRM=(0.5*(G+SQRT((G**2)+(4.0*H))))
      LTIME4=((DRM*60.0)/(RS*2000.0))
```

```
      LTIME5=LTIME4+ET
      SV(EA1,11)=LTIME5
C
C     SUBROUTINE FIND IS CALLED TO GENERATE TIME OF EITHER DETECTION OR
C     CPA FOR THE SCREEN SHIP ONLY IF THE SCREEN SHIP HAS NOT PREVIOUSLY
C     MADE A DETECTION ON THE SUBMARINE.
C
      IF(SV(EA1,12).LT.1.0)CALL FIND(M,ET,EA1,EA2,EA3)
      IF(LTIME5.GT.IFIX(TAR(1,7)))GO TO 90
C
C     STORE TIME OF NEXT COURSE CHANGE
C
      CALL SNE(LTIME5,1,EA1,EA2,EA3)
      GO TO 90
   21 B3=(Q-(XIP+PI))
      GO TO 22
    4 XIP=(2.0*PI)-XI
      T1=(XIP-PI)
      T2=(XIP-((3.0*PI)/2.0))
    8 Q=(2.0*PI)*URN(1)
      IF(Q.LT.T1.AND.Q.GT.T2)GO TO 30
      GO TO 8
   30 IF(Q.GE.PI)GO TO 600
      GAMMA=(PI-Q)
      A1=(2.0*A*COS(GAMMA))
      RS=(0.5*(A1+SQRT((A1**2)+(4.0*C))))
      DESCSE=ARCOS(((A**2)+(B**2)-(RS**2))/(2.0*A*B))
      SV(EA1,2)=DESCSE
      GO TO 601
  600 GAMMA=(Q-PI)
      A1=(2.0*A*COS(GAMMA))
      RS=(0.5*(A1+SQRT((A1**2)+(4.0*C))))
      DESCSE=((2.0*PI)-(ARCOS(((A**2)+(B**2)-(RS**2))/(2.0*A*B))))
      SV(EA1,2)=DESCSE
      GO TO 601
  601 IF(Q.GT.(XIP-PI))GO TO 602
      B3=((XIP-PI)-Q)
  603 F=(SV(EA1,8)-1.0)
      G=(2.0*SCRLOC*COS(B3))
      H=((F**2)-(SCRLOC**2))
      DRM=(0.5*(G+SQRT((G**2)+(4.0*H))))
      LTIME4=((DRM*60.0)/(RS*2000.0))
      LTIME5=LTIME4+ET
      SV(EA1,11)=LTIME5
C
C     SUBROUTINE FIND IS CALLED TO GENERATE TIME OF EITHER DETECTION OR
C     CPA FOR THE SCREEN SHIP ONLY IF THE SCREEN SHIP HAS NOT PREVIOUSLY
C     MADE A DETECTION ON THE SUBMARINE.
```

```
C
      IF(SV(EA1,12).LT.1.0)CALL FIND(M,ET,EA1,EA2,EA3)
      IF(LTIME5.GT.IFIX(TAR(1,7)))GO TO 90
C
C     STORE TIME OF NEXT COURSE CHANGE
C
      CALL SNE(LTIME5,1,EA1,EA2,EA3)
      GO TO 90
  602 B3=(Q-(XIP-PI))
      GO TO 603
    5 XIP=(XI+PI)
      T1=(XIP-(PI/2.0))
      T2=(XIP+(PI/2.0))
    9 Q=(2.0*PI)*URN(1)
      IF(Q.GT.T2.OR.Q.LT.T1)GO TO 40
      GO TO 9
   40 IF(Q.GE.PI)GO TO 700
      GAMMA=(PI-Q)
      A1=(2.0*A*COS(GAMMA))
      RS=(0.5*(A1+SQRT((A1**2)+(4.0*C))))
      DESCSE=ARCOS(((A**2)+(B**2)-(RS**2))/(2.0*A*B))
      SV(EA1,2)=DESCSE
      GO TO 701
  700 GAMMA=(Q-PI)
      A1=(2.0*A*COS(GAMMA))
      RS=(0.5*(A1+SQRT((A1**2)+(4.0*C))))
      DESCSE=((2.0*PI)-(ARCOS(((A**2)+(B**2)-(RS**2))/(2.0*A*B))))
      SV(EA1,2)=DESCSE
      GO TO 701
  701 IF(Q.GT.(XIP-PI))GO TO 702
      B3=(Q-(XIP-PI))
  703 F=(SV(EA1,8)-1.0)
      G=(2.0*SCRLOC*COS(B3))
      H=((F**2)-(SCRLOC**2))
      DRM=(0.5*(G+SQRT((G**2)+(4.0*H))))
      LTIME4=((DRM*60.0)/(RS*2000.0))
      LTIME5=LTIME4+ET
      SV(EA1,11)=LTIME5
C
C     SUBROUTINE FIND IS CALLED TO GENERATE TIME OF EITHER DETECTION OR
C     CPA FOR THE SCREEN SHIP ONLY IF THE SCREEN SHIP HAS NOT PREVIOUSLY
C     MADE A DETECTION ON THE SUBMARINE.
C
      IF(SV(EA1,12).LT.1.0)CALL FIND(M,ET,EA1,EA2,EA3)
      IF(LTIME5.GT.IFIX(TAR(1,7)))GO TO 90
C
C     STORE TIME OF NEXT COURSE CHANGE
C
```

```
      CALL SNE(LTIME5,1,EA1,EA2,EA3)
      GO TO 90
  702 B3=((XIP-PI)-Q)
      GO TO 703
    6 XIP=(PI-XI)
      T1=(XIP-(PI/2.0))
      T2=(XIP+(PI/2.0))
   10 Q=(2.0*PI)*URN(1)
      IF(Q.GT.T2.OR.Q.LT.T1)GO TO 50
      GO TO 10
   50 IF(Q.GE.PI)GO TO 800
      GAMMA=(PI-Q)
      A1=(2.0*A*COS(GAMMA))
      RS=(0.5*(A1+SQRT((A1**2)+(4.0*C))))
      DESCSE=ARCOS(((A**2)+(B**2)-(RS**2))/(2.0*A*B))
      SV(EA1,2)=DESCSE
      GO TO 801
  800 GAMMA=(Q-PI)
      A1=(2.0*A*COS(GAMMA))
      RS=(0.5*(A1+SQRT((A1**2)+(4.0*C))))
      DESCSE=((2.0*PI)-(ARCOS(((A**2)+(B**2)-(RS**2))/(2.0*A*B))))
      SV(EA1,2)=DESCSE
      GO TO 801
  801 IF(Q.LT.(XIP+PI))GO TO 802
      IF(Q.GT.(XIP+PI))GO TO 803
      IF(Q.LT.T1)B3=(Q+(XIP-(PI/2.0)))
      GO TO 804
  802 B3=(XIP+PI)-Q
      GO TO 804
  803 B3=(Q-(XIP+PI))
      GO TO 804
  804 F=(SV(EA1,8)-1.0)
      G=(2.0*SCRLOC*COS(B3))
      H=((F**2)-(SCRLOC**2))
      DRM=(0.5*(G+SQRT((G**2)+(4.0*H))))
      LTIME4=((DRM*60.0)/(RS*2000.0))
      LTIME5=LTIME4+ET
      SV(EA1,11)=LTIME5
C
C     SUBROUTINE FIND IS CALLED TO GENERATE TIME OF EITHER DETECTION OR
C     CPA FOR THE SCREEN SHIP ONLY IF THE SCREEN SHIP HAS NOT PREVIOUSLY
C     MADE A DETECTION ON THE SUBMARINE.
C
      IF(SV(EA1,12).LT.1.0)CALL FIND(M,ET,EA1,EA2,EA3)
      IF(LTIME5.GT.IFIX(TAR(1,7)))GO TO 90
C
C     STORE TIME OF NEXT COURSE CHANGE
C
```

```
      CALL SNE(LTIME5,1,EA1,EA2,EA3)
      GO TO 90
   90 CONTINUE
      RETURN
  100 WRITE(6,101)EA1,SV(EA1,4),SV(EA1,5),SV(EA1,9),SV(EA1,10),ET,M
  101 FORMAT('0','SHIP',I4,2X,'OUT OF PATROL AREA. LOCATION',F8.0,F8.0,2
     1X,'SCREEN CENTER',F8.0,F8.0,2X,'TIME',I5,2X,'REPLICATION',I5)
      GO TO 102
      END


      SUBROUTINE EVENT2(ET,EA1,EA2,EA3,M,*)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     EVENT LOCATES THE SHIPS AND SUBMARINES AT DETECTION AND CALCULATES
C     THE DISTANCE FROM ZZ OF THE SUBMARINE AT DETECTION. THIS VALUE IS
C     STORED FOR USE IN DETERMINING AVERAGE DETECTION RANGE OF THE
C     SUBMARINE FROM ZZ IN EACH REPLICATION AND IN THE RUN.
C
      CALL POSIT(ET,2,EA1,EA2,EA3,1350)
      DX=TAR(1,4)-SSV(EA2,4)
      DY=TAR(1,5)-SSV(EA2,5)
      DSTAT(M,EA1,EA2)=SQRT((DX**2)+(DY**2))
      IF(MSTAT(8).EQ.0)GO TO 8004
      WRITE(6,8001)ET,EA1,EA2,SV(EA1,4),SV(EA1,5)
 8001 FORMAT('0',4X,'SUBROUTINE EVENT2',//,5X,'DETECTION TIME',2X,I10,2X
     1,'BY DD',I4,2X,'ON SUBMARINE',I4,2X,'AT X=',F10.2,2X,'Y=',F10.2,2X
     2,//)
 8004 SSV(EA2,7)=SSV(EA2,7)+1.0
      SV(EA1,12)=SV(EA1,12)+1.0
      SUBKIL=SUBKIL+1
      IF(SSV(EA2,7).GE.FLOAT(MSTAT(5)))GO TO 8005
      GO TO 8003
 1350 WRITE(6,8002)
 8002 FORMAT('0','TIME USED IN SUBROUTINE POSIT GREATER THAN 200.0 MIN.'
     1,//)
 8003 RETURN
 8005 CALL EVENT9(ET,EA1,EA2,88)
      RETURN 1
      END
```

```
      SUBROUTINE EVENT3(ET,EA1,EA2,EA3)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     EVENT LOCATES SHIPS AND SUBMARINES AT CPA AND STORES THIS VALUE
C     IN THE CPA VECTOR WHERE IT MAY BE PRINTED OUT AT THE END OF EACH
C     REPLICATION IF THE CONTROL CARD IS SO CODED BY THE USER.
C
      CALL POSIT(ET,3,EA1,EA2,EA3,1360)
      IF(EA3.EQ.99)GO TO 8006
      IF(MSTAT(7).EQ.0)GO TO 8004
      DX1=SV(EA1,4)-SSV(EA2,4)
      DY1=SV(EA1,5)-SSV(EA2,5)
      RANG1=SQRT((DX1**2)+(DY1**2))
      IF(CPA(EA1,EA2,4).LT.RANG1)GO TO 8004
      CPA(EA1,EA2,1)=FLOAT(EA1)
      CPA(EA1,EA2,2)=FLOAT(EA2)
      CPA(EA1,EA2,3)=FLOAT(ET)
      CPA(EA1,EA2,4)=RANG1
      CPA(EA1,EA2,5)=SV(EA1,4)
      CPA(EA1,EA2,6)=SV(EA1,5)
      CPA(EA1,EA2,7)=SSV(EA2,4)
      CPA(EA1,EA2,8)=SSV(EA2,5)
      CPA(EA1,EA2,9)=TAR(1,4)
      CPA(EA1,EA2,10)=TAR(1,5)
 8004 RETURN
 8006 NN=MSTAT(1)
      MM=MSTAT(2)
      DO 8007 I=1,NN
      DO 8007 J=1,MM
      DX1=SV(I,4)-SSV(J,4)
      DY1=SV(I,5)-SSV(J,5)
      RANG1=SQRT((DX1**2)+(DY1**2))
      IF(CPA(I,J,4).LT.RANG1)GO TO 8007
      CPA(I,J,1)=FLOAT(I)
      CPA(I,J,2)=FLOAT(J)
      CPA(I,J,3)=FLOAT(ET)
      CPA(I,J,4)=RANG1
      CPA(I,J,5)=SV(I,4)
      CPA(I,J,6)=SV(I,5)
      CPA(I,J,7)=SSV(J,4)
      CPA(I,J,8)=SSV(J,5)
      CPA(I,J,9)=TAR(1,4)
      CPA(I,J,10)=TAR(1,5)
 8007 CONTINUE
```

```
      GO TO 8004
      END

      SUBROUTINE EVENT4(ET,EA1,EA2,EA3)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     EVENT LOCATES SHIPS AND SUBMARINES AT TIME OF SUBMARINE ATTACK.
C     THIS EVENT WILL PRINT OUT THE LOCATIONS OF SHIPS, ZZ AND SUBMARINE
C     AT TIME OF ATTACK IF DESIRED BY THE USER. TO HAVE THE EVENT PRINT
C     OUT, THE USER CODES THE CONTROL CARD WITH A 1 IN COLUMN 64 OF THE
C     CONTROL CARD. IF NO OUTPUT IS DESIRED, A ZERO IS CODED IN THE
C     SAME COLUMN.
C
      MTARK=MTARK+1
      DESKIL=DESKIL+1
      IF(MSTAT(16).EQ.0)GO TO 8001
      MM=(100*MSTAT(15))
      CALL POSIT(ET,4,EA1,EA2,EA3)
      WRITE(6,8002)ET,EA2,SSV(EA2,4),SSV(EA2,5),EA3,TAR(EA3,4),TAR(EA3,5
     1),MM
 8002 FORMAT('0',4X,'TIME',2X,'SUB NO',2X,'SUB X',2X,'SUB Y',2X,'TARGET
     1NO',2X,'TAR X',2X,'TAR Y',2X,'LAUNCH RANGE',//,5X,I4,3X,I3,3X,F6.0
     2,2X,F6.0,5X,I4,3X,F6.0,2X,F6.0,5X,I5,//)
 8001 RETURN
      END

      SUBROUTINE EVENT5(ET,EN,EA1,EA2,EA3)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
      RETURN
      END

      SUBROUTINE EVENT6(ET,EN,EA1,EA2,EA3)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
```

```
      RETURN
      END

      SUBROUTINE EVENT7(ET,EN,EA1,EA2,EA3)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
      RETURN
      END

      SUBROUTINE EVENT8(ET,EN,EA1,EA2,EA3)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
      RETURN
      END

      SUBROUTINE EVENT9(ET,EA1,EA2,EA3)
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     EVENT LOCATES SHIPS AND SUBMARINES AT THE TERMINATION OF A
C     REPLICATION.
C
      CALL POSIT(ET,9,EA1,EA2,EA3,1370)
      IF(EA3.EQ.88)GO TO 8009
      MBKILL=MBKILL+1
      MBKIL=MBKIL+1
 8009 IF(MSTAT(7).EQ.1)GO TO 8005
 8008 IF(EA3.EQ.88)GO TO 8006
 8004 IF(MSTAT(10).EQ.0)GO TO 8003
      MM=MSTAT(1)
      WRITE(6,8002)
 8002 FORMAT('0',4X,'TIME  SCREEN SHIP      X-POSIT   Y-POSIT',/)
      WRITE(6,8001)(SV(I,13),I,SV(I,4),SV(I,5),I=1,MM)
 8001 FORMAT('0',T5,F4.0,T17,I2,T28,F7.1,T38,F7.1,/)
      NN=MSTAT(3)
      WRITE(6,8010)
```

```
 8010 FORMAT('0',4X,'TIME  MAIN BODY SHIP   X-POSIT   Y-POSIT',/)
      WRITE(6,8001)(TAR(I,6),I,TAR(I,4),TAR(I,5),I=1,NN)
      LL=MSTAT(2)
      WRITE(6,8011)
 8011 FORMAT('0',4X,'TIME   SUBMARINE        X-POSIT   Y-POSIT',/)
      WRITE(6,8001)(SSV(I,6),I,SSV(I,4),SSV(I,5),I=1,LL)
 8003 RETURN
 8005 CALL EVENT3(ET,EA1,EA2,99)
      GO TO 8008
 8006 WRITE(6,8007)ET,SSV(EA2,7)
 8007 FORMAT('0',4X,'RUN TERMINATED AT TIME',I5,2X,'BY DETECTION BY',F6.
     10,2X,'SCREEN SHIPS',/)
      GO TO 8003
      END


      SUBROUTINE REPLIC
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SUBROUTINE RESETS SHIP,SUBMARINE, AND MAIN BODY VECTORS TO THE
C     ORIGINAL VALUES  FOR USE IN THE NEXT REPLICATION OF THE RUN
C
      L=MSTAT(1)
      DO 5 I=1,L
      DO 5 J=1,15
      SV(I,J)=DSV(I,J)
    5 CONTINUE
      N=MSTAT(2)
      DO 4 K=1,N
      DO 4 L=1,15
      SSV(K,L)=DSSV(K,L)
    4 CONTINUE
      N=MSTAT(3)
      DO 6 L=1,N
      DO 6 K=1,10
      TAR(L,K)=DTAR(L,K)
    6 CONTINUE
C
C     CPA VECTOR RESET FOR NEW REPLICATION
C
      DO 3 I=1,10
      DO 3 J=1,10
      DO 3 K=1,10
      CPA(I,J,K)=99999.99
```

```
    3 CONTINUE
C
C     EVENT CALENDAR AND DUMMY EVENT CALENDAR RESET FOR NEW REPLICATION
C
      DO 7 M=1,200
      DO 7 N=2,5
      ECAL(M,1)=999999999
      DECAL(M,1)=999999999
      ECAL(M,N)=0
      DECAL(M,N)=0
    7 CONTINUE
C
C     COUNTERS FOR SUBMARINE PENETRATIONS, SCREEN SHIP DETECTIONS AND
C     SUBMARINE ATTACKS ON THE MAIN BODY RESET FOR NEW REPLICATION
C
      SUBKIL=0
      DESKIL=0
      MBKIL=0
      KT=0
      R=1.0
      RETURN
      END


      SUBROUTINE SONDAT
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SUBROUTINE PRINTS SONAR INFORMATION FOR EACH SCREEN SHIP AND THE
C     AVERAGE GENERATED RANGE FOR THE SCREEN SHIP OVER THE REPLICATIONS
C     IN THE RUN.
C
      NN=MSTAT(1)
      MM=MSTAT(4)
      DO 1 I=1,NN
      X=0.0
      SONSUM=0.0
      DO 2 J=1,MM
      SONSUM=SONSUM+SONAR(I,J)
      X=X+1.0
    2 CONTINUE
      SONAV(I)=(SONSUM/X)
      WRITE(6,1002)
      IF(MM.GT.10)GO TO 1003
      WRITE(6,1001)I,SONAV(I),(SONAR(I,J),J=1,MM)
```

```
    1 CONTINUE
      RETURN
 1003 WRITE(6,1004)
      WRITE(6,1005)I,SONAV(I),(SONAR(I,J),J=1,MM)
 1004 FORMAT('0',6X,'SHIP      AVRANGE      REPLICATION RANGES',/)
 1005 FORMAT('0',4X,I6,2X,11(F8.1,2X),//,21X,10(F8.1,2X),/)
      GO TO 1
 1001 FORMAT('0',2X,I6,2X,11(F8.1,2X),//)
 1002 FORMAT('0',3X,' SHIP      AVRANGE   REP1      REP2      REP3      R
     1EP4      REP5      REP6      REP7      REP8      REP9      REP10')
      END


      SUBROUTINE FIGURE
      INTEGER ET,SUBKIL,DESKIL,EN,EA1,EA2,EA3,ECAL,DECAL,TEMP,KT
      COMMON/ASWGAM/SV(10,15),DSV(10,15),SSV(10,15),DSSV(10,15),TAR(10,1
     10),DTAR(10,10),CPA(10,10,10),ECAL(200,5),DESKIL,SUBKIL,MBKIL,PI,MA
     2XEV,TEMP(5),SONAR(10,20),SONAV(10),R,KT,MSTAT(20),DECAL(200,5),MBK
     3ILL,MTARK,DSTAT(20,10,10),STAT(20,2)
C
C     SUBROUTINE PROVIDES AVERAGE RANGE OF SUBMARINE DETECTION FROM
C     ZZ FOR EACH REPLICATION
C
      II=MSTAT(4)
      JJ=MSTAT(1)
      KK=MSTAT(2)
      DSUM=0.0
      Q=0.0
      DSQR=0.0
      DO 1001 L=1,II
      DDSUM=0.0
      T=0.0
      DO 1002 M=1,JJ
      DO 1003 N=1,KK
      IF(DSTAT(L,M,N).EQ.0.0)GO TO 1003
      DDSUM=DDSUM+DSTAT(L,M,N)
      DSUM=DSUM+DSTAT(L,M,N)
      DSQR=DSQR+(DSTAT(L,M,N)**2)
      T=T+1.0
 1003 CONTINUE
 1002 CONTINUE
      IF(T.EQ.0.0)GO TO 1009
      AVSUM=(DDSUM/T)
      AVSQR=(AVSUM**2)
      STAT(L,1)=AVSUM
      STAT(L,2)=AVSQR
      Q=Q+T
      GO TO 1001
```

```
1009 AVSUM=0.0
     AVSQR=0.0
     STAT(L,1)=0.0
     STAT(L,2)=0.0
     GO TO 1001
1001 CONTINUE
     IF(Q.EQ.0.0)GO TO 1006
     DAV=(DSUM/Q)
     IF(Q.EQ.1.0)GO TO 1008
     STD=SQRT((1.0/(Q-1.0))*(DSQR-(Q*(DAV**2))))
1007 WRITE(6,1004)MBKILL,MTARK,DAV,STD
1004 FORMAT('0',5X,'PENETRATIONS     WEPLAUNCH     AVDET    STDDEV',//,10X
    1,I5,10X,I5,4X,F8.1,3X,F8.2,//)
     WRITE(6,1006)
1006 FORMAT('0',4X,'AVERAGE RANGE OF SUBMARINE DETECTION FROM ZZ IN EAC
    1H REPLICATION',//)
     WRITE(6,1005)((STAT(I,J),I=1,II),J=1,2)
1005 FORMAT(10F13.2)
     RETURN
1006 MBKILL=0.0
     DAV=0.0
     STD=0.0
     GO TO 1007
1008 STD=0.0
     GO TO 1007
     END
```

APPENDIX B: LIST OF PROGRAM VARIABLES

This appendix contains the name and definition of the symbols appearing in the computer program. This list contains only those variables using storage space in the program and does not contain those used only for temporary storage.

Name	Definition
CPA(I, J, K)	CPA information for the I^{th} ship on the J^{th} submarine in a replication. The values in this vector are:
CPA(I, J, 1)	Screen ship number
CPA(I, J, 2)	Penetrating submarine number
CPA(I, J, 3)	Time of CPA
CPA(I, J, 4)	CPA range
CPA(I, J, 5)	Screen ship X location at CPA
CPA(I, J, 6)	Screen ship Y location at CPA
CPA(I, J, 7)	Penetrating submarine X location at CPA
CPA(I, J, 8)	Penetrating submarine Y location at CPA
CPA(I, J, 9)	X location of ZZ at CPA
CPA(I, J, 10)	Y location of ZZ at CPA
DECAL (I, J)	Dummy event calendar, a copy of all events entered in the event calendar. The I^{th} entry in DECAL contains:
DECAL(I, 1)	Event time
DECAL(I, 2)	Event number
DECAL(I, 3)	Number of screen ship in event
DECAL(I, 4)	Number of penetrating submarine in event
DECAL(I, 5)	Number of main body ship in event
DESKIL	An index of successful attacks by a penetrating submarine on a main body ship in a replication

Name	Definition
DSSV(I, J)	Dummy submarine vector containing initial information
DSSV(I, 1)	Number of Ith penetrating submarine
DSSV(I, 3)	Speed of submarine
DSSV(I, 4)	Initial X location of submarine
DSSV(I, 5)	Initial Y location of submarine
DSTAT(I, J, K)	Detection range from ZZ in the Ith replication for the Rth submarine by the Jth screen ship
DSV(I, J)	Dummy screen ship vector containing initial information
DSV(I, 1)	Number of Ith screen ship
DSV(I, 2)	Initial screen ship course (in radians)
DSV(I, 3)	Screen ship speed
DSV(I, 4)	Initial screen ship X location
DSV(I, 5)	Initial screen ship Y location
DSV(I, 6)	Screen ship sonar type
DSV(I, 8)	Screen ship patrol area radius
DSV(I, 9)	Initial X location of center of assigned patrol area
DSV(I, 10)	Initial Y location of center of assigned patrol area
DTAR(I, J)	Dummy main body ship vector containing initial information
DTAR(I, 1)	Number of Ith main body ship
DTAR(I, 2)	Initial main body ship course (in radians)
DTAR(I, 3)	Main body ship speed
DTAR(I, 4)	Initial X location of main body ship
DTAR(I, 5)	Initial Y location of main body ship

Name	Definition
EA1	Number of a screen ship
EA2	Number of a penetrating submarine
EA3	Number of a main body ship
ECAL(I, J)	I^{th} entry in event calendar
ECAL(I, 1)	Event time
ECAL(I, 2)	Event number
ECAL(I, 3)	Number of screen ship in event
ECAL(I, 4)	Number of penetrating submarine in event
ECAL(I, 5)	Number of main body ship in event
EN	Event number
ET	Event time
KT	Index of total number of events entered in DECAL(I, J)
MAXEV	Maximum number of events allowed in ECAL(I, J)
MBKIL	Number of successful submarine penetrations in replication
MBKILL	Number of successful submarine penetrations in run
MSTAT(I)	Control vector for each run
MSTAT(1)	Number of screen ships in run
MSTAT(2)	Number of submarines in run
MSTAT(3)	Number of main body ships in run
MSTAT(4)	Number of replications in run
MSTAT(5)	Number of detections of submarine by screen ships to terminate run

Name	Definition
MSTAT(6)	Print screen ship detection range for each replication & average detection range for each screen ship during run
MSTAT(7)	Print out CPA & locations of each ship & submarine in replication
MSTAT(8)	Print out locations of screen ship, submarine, and ZZ at time of detection of a penetrating submarine by a screen ship
MSTAT(9)	Minimum submarine penetration speed
MSTAT(10)	Print out locations of screen ships, submarines, & ZZ at termination of each replication
MSTAT(11)	Print out event calendar for each replication
MSTAT(12)	Print out ship vectors for each replication
MSTAT(13)	Print out main body vectors for each replication
MSTAT(14)	Print out submarine vectors for each replication
MSTAT(15)	Submarine weapon release range (100's of yards)
MSTAT(16)	Print out locations of main body ship & submarine at weapon release for each replication
MTARK	Index of submarine attacks on main body ships
PI	3.14159
R	Index used in subroutine FIND to isolate sections of the subroutine after the first call on the subroutine

Name	Definition
SONAR(I, J)	The generated detection range of the J^{th} screen ship for the I^{th} replication in a run.
SONAV(I)	The average of generated detection ranges of the I^{th} screen ship in a run
SSV(I, J)	Information vector of the I^{th} submarine in a replication
SSV(I, 1)	Number of the I^{th} submarine
SSV(I, 2)	Submarine course (in radians)
SSV(I, 3)	Submarine speed
SSV(I, 4)	Submarine X location
SSV(I, 5)	Submarine Y location
SSV(I, 6)	Time X-Y location fixed
SSV(I, 7)	Number of times detected by screen ships
SUBKIL	Number of times the penetrating submarines detected
SV(I, J)	Information vector of the I^{th} screen ship in a replication
SV(I, 1)	Number of the I^{th} screen ship
SV(I, 2)	Screen ship course (in radians)
SV(I, 3)	Screen ship speed
SV(I, 4)	Screen ship X location
SV(I, 5)	Screen ship Y location
SV(I, 6)	Screen ship sonar type
SV(I, 7)	Screen ship generated detection range
SV(I, 8)	Patrol area radius

Name	Definition
SV(I, 9)	X location of center of assigned patrol area
SV(I, 10)	Y location of center of assigned patrol area
SV(I, 11)	Time of next course change
SV(I, 12)	Number of submarines detected in replication
SV(I, 13)	Time X-Y location fixed
TAR(I, J)	Information vector for the Ith main body ship
TAR(I, 1)	Number of the Ith main body ship
TAR(I, 2)	Main body ship course (in radians)
TAR(I, 3)	Main body ship speed
TAR(I, 4)	Main body ship X location
TAR(I, 5)	Main body ship Y location
TEMP(I)	Used to aid in ordering event calendar as a temporary storage location for an event

BIBLIOGRAPHY

Box. G. E. P., and Hunter, J. S., "The 2^{k-p} Fractional Factorial Designs: Part I," Technometrics, v. 3, p. 311, 1961.

Box, G. E. P., and Hunter, J. S., "The 2^{k-p} Fractional Factorial Designs: Part II," Technometrics, v. 3, p. 449, 1961.

Larson, H. J., Introduction to Probability Theory and Statistical Inference, Wiley, 1968.

Ostle, B., Statistics in Research, 2d ed., Iowa State University Press, 1969.

Smith, D. E., "Optimization within Computer Simulation: An Application of Response Surface Methodology," MORS Conference Report, 1969.

INITIAL DISTRIBUTION LIST

No. Copies

1. Library, Code 0212 2
Naval Postgraduate School
Monterey, California, 93940

2. Asst. Professor Alvin F. Andrus, Code 55 1
Department of Operations Analysis
Naval Postgraduate School
Monterey, California, 93940

3. Department of Operations Analysis (Code 55) 1
Naval Postgraduate School
Monterey, California, 93940

4. LCDR Charles Donald Griffin, Jr., USN 1
1600 South Joyce St., Apt. B-607
Arlington, Virginia, 22202

5. Defense Documentation Center 2
Cameron Station
Alexandria, Virginia 22314

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*
Master's Thesis; March 1971

5. AUTHOR(S) *(First name, middle initial, last name)*

Charles D. Griffin, Jr.

6. REPORT DATE
March 1971

7a. TOTAL NO. OF PAGES
83

7b. NO. OF REFS
5

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

The model presented in this thesis is a computer simulation model of ASW interactions between a formation of high value group ships, protected by some screening ships, and some penetrating submarines. The model is designed for use as an aid in improving the ability of a proposed screening tactic in the detection of a penetrating submarine. A systematic procedure to improve a screen's effectiveness against a known submarine threat is demonstrated, and an example problem is worked using this procedure.

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Simulation						
War Game						
ASW Simulation Model						
Response Surface Methodology (RSM)						

Unclassified
Security Classification