1975-03

A computer simulation for the evaluation of fleet air defense employing monostatic and bistatic search radars

Leonhardt, Richard John; Schlegelmilch, Charles Robert

Monterey, California. Naval Postgraduate School

A COMPUTER SIMULATION FOR THE EVALUATION OF FLEET AIR DEFENSE EMPLOYING MONOSTATIC AND BISTATIC SEARCH RADARS

Richard John Leonhardt

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A COMPUTER SIMULATION FOR THE
EVALUATION OF FLEET AIR DEFENSE EMPLOYING
MONOSTATIC AND BISTATIC SEARCH RADARS

by

Richard John Leonhardt

and

Charles Robert Schlegelmilch

March 1975

Thesis Advisor: A. F. Andrus

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Computer Simulation for the Evaluation of Fleet Air Defense Employing Monostatic and Bistatic Search Radars		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1975
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard John Leonhardt Charles Robert Schlegelmilch		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1975
		13. NUMBER OF PAGES 312
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

Professor A. F. Andrus, Autovon 479-2350

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

Bistatic Radar
War Game
Simulation
Fleet Air Defense

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

A computer simulation model is developed for use in analysing the Navy's Fleet Air Defense problem. The model provides for employment of monostatic and bistatic search radars in a clear or jamming environment. Use of the model is demonstrated by analysing a hypothetical problem.

A Computer Simulation for the Evaluation of Fleet Air Defense Employing Monostatic and Bistatic Search Radars

by

Richard John Leonhardt
Lieutenant Commander, United States Navy
B.S., University of Notre Dame, 1965

and

Charles Robert Schlegelmilch
Lieutenant Commander, United States Navy
B.S., Marquette University, 1961

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1975

ABSTRACT

A computer simulation model is developed for use in analysing the Navy's Fleet Air Defense problem. The model provides for employment of monostatic and bistatic search radars in a clear or jamming environment. Use of the model is demonstrated by analysing a hypothetical problem.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

The purpose of this thesis is to develop a computer simulation model for use in analysing the Navy's Fleet Air Defense problem, and to demonstrate the usefulness of the model with a hypothetical air defense scenario. The model is unique in that it provides for employment of bistatic as well as monostatic search radars. A bistatic radar is simply a radar system in which the radar transmitter and the radar receiver are not collocated. This capability was provided to allow for the investigation of the possible advantages of the use of a bistatic radar receiver in view of the extensive capability for passive detection of active radars.

The model, acronym FADS, for Fleet Air Defense Simulation, is programmed in FORTRAN IV level G, and is designed to run on an IBM 360/65 computer meeting the core requirement of 350 kilobytes.

FADS is structured along the lines of FLOATS IIB, a model currently maintained at the Johns Hopkins University Applied Physics Laboratory [Ref. 6]. The significant differences from FLOATS IIB which have been incorporated in FADS are: bistatic search radar systems, provision for engagement of bombers by surface-to-air missiles (SAM), the addition of a surface-to-surface missile (SSM) capability for the defensive forces, a control logic which requires the defensive forces to hold fire until fired on, the inclusion of a dispersed defensive force disposition, and the

means for constructing up to six distinct target lists for the attacking units.

FADS is suitable for analysing both large scale problems and one-on-one situations in a clear or jamming environment. In general, the model provides for the description of up to 90 enemy raiders of 10 different classes, 40 defensive units of 10 different classes, 15 classes of radars, 10 classes of SAMs, 35 time delay distribution functions, and 18 logic switches are included which vary the defensive play of the simulation. During the computer debugging phase of the model construction, FADS was exercised in all of its various modes. Included among these were clear and jamming environment, single and multi threats, area and point defense, and monostatic and bistatic search radar systems.

The particular problem used to demonstrate the usefulness of the model is an investigation of the relative effectiveness of 100 nautical mile and 50 nautical mile non-fire control dedicated SAM systems deployed on picket type surface units with bistatic search radar capability. This type of SAM system employs active or passive homing throughout its entire flight. The measure of effectiveness for the respective SAM systems is the rate at which enemy air-to-surface and surface-to-surface missiles (ASM/SSM) arrive at their target area during the ASM/SSM impact phase. This is an indication of the level of the enemy attack surviving the picket defenses and will demonstrate the diminishing of the intensity of the air battle which will be

waged in the vicinity of the main force. The primary concern is to determine what effect the different missile ranges will have on breaking up this focus of an enemy ASM/SSM attack.

The analysis was conducted with hypothetical or illustrative values for all systems. Specific performance characteristics for the bistatic radar system and the missile system do not exist. Care was taken to ensure that the hypothetical values were within the range of existing systems and that the proposed systems' parameters were comparable to present system capabilities when appropriate.

Section II of the thesis contains a general description of the flow of events in FADS and a discussion of the principle assumptions in the model. The specific inputs and results of the demonstration problem together with an analysis of the results are contained in Section III. Sections IV and V provide a detailed listing of the inputs required and the outputs received from FADS. Conclusions are given in Section VI. The appendixes contain background information on bistatic radar systems, a verbal description of the simulation program, glossaries, and a FORTRAN IV computer program listing.

II. BACKGROUND AND DISCUSSION

A. GENERAL DESCRIPTION OF THE SIMULATION MODEL

FADS is a large scale store-event simulation of a Task Force size surface missile engagement with an enemy air attack. The model is capable of representing a threat consisting of a coordinated missile attack launched from aircraft, surface units or submarines. Although the model provides for defensive engagement of the enemy missiles and launch platforms with either SAM or SSM, there is no determination of defensive interceptor attrition of enemy systems during the engagement. If interceptors are to be employed, they must be played against the launch platforms prior to the start of the game and the enemy forces reduced to the appropriate level. The Air Battle Analyzer developed by the Applied Physics Laboratory of Johns Hopkins University is a suitable tool to determine these interceptor attritions [Ref. 11].

The simulation determines friendly force equipment availabilities at the start of each iteration. These availabilities are based on a comparison between user inputs of the probability that a given equipment is operational and the value of a uniform (0,1) random number which is drawn by the computer. The next step in the simulation's iterative process is to determine detections and firing events for the friendly forces against the enemy launch platforms. These

SAM/SSM fire events are stored in an events list in a time sequence. The enemy ASM/SSM launch sequence and targeting is computed next. Detections and resultant fire events against the enemy ASM/SSM's are then determined and entered in the appropriate sequence in the events list. Friendly SAM/SSM intercepts and target hits are evaluated followed by enemy ASM/SSM hits and ship kill determinations. The kill and damage events are time interactive so that offensive and defensive force capabilities are decremented as hits occur. The simulation then prints statistics for the iteration and starts the next iteration by re-entering the flow at the point where equipment availabilities are determined. The program stops upon completion of the specified number of iterations and cumulative statistics are provided.

B. ASSUMPTIONS

1. Bistatic Radar System

In the bistatic detection portion of the simulation, the principle assumptions made regarding bistatic radar systems are: a) the target's bistatic radar cross section is equal to its monostatic cross section; b) the target is not located along the transmitter-receiver base line; and c) the radar propagation losses are the same as those experienced in monostatic systems.

a. Radar Cross Section

The approximation for the bistatic radar cross section is made on the basis that, in the limit, the bistatic cross section is equal to the monostatic cross

section except for the case where the scattering angle is equal to 180° [Ref. 10, p. 590]. The scattering angle is the angle between the path from the transmitter to the target and the path from the target to the receiver. It is equal to 180° when the target is on the base line between the transmitter and the receiver. The fact that the monostatic and bistatic cross sections are the same in the limit does not imply that they are everywhere equal. It does imply that the range of the values will be equal and thus on the average allow usage of the monostatic cross section to provide representative values for detection ranges during the iterative process of the computer simulation.

b. Target Location

The assumption concerning the target not being on the base line between the transmitter and the receiver is necessary since range ambiguities result when the scattering angle is equal to 180°. This assumption is not unreasonable since the typical track may cross the base line, but will not originate on it and remain on the base line. A situation in which this could occur would be the case when a raider while on the base line launches a missile at a target which is either the transmitter or the receiver platform.

c. Propagation Losses

The assumption concerning the propagation losses is appropriate in that similar assumptions are routinely made in arriving at detection ranges for monostatic radars. A representative propagation loss for the nominal detection

range is used in conjunction with the signal to noise ratio required for a specified probability of detection to arrive at a minimum reflected power level which must be received in order for target detection to occur.

An additional assumption concerning the bistatic radar system in the simulation model is that the bistatic transmitter is always in an operative status. It is not destroyed by enemy missiles during the attack phase nor is it evaluated as being down or degraded during the equipment status evaluation which is conducted at the start of each iteration. Individual bistatic receivers have an assigned probability of being operational which should also reflect the reliability of their associated transmitter.

2. Jamming

Some major assumptions were made in simulating the jamming environment in order to keep the problem tractable and to remain within reasonable time limits for the computer program. The jamming assumptions will tend to drive the solution to conservative estimates for the defensive forces. The first assumption made is that the jammer and detecting radar have a single gain for their main beam. Thus targets not in the main beam are considered to be in a unique side-lobe which is a specified number of db below the main beam. To determine the level of the jamming power arriving at the detecting radar, it was necessary to approximate the position of the jammer relative to the detecting radar. This was done by determining the jammer's position at the time a

given target was at the midpoint of its track through the radar's detection envelope. In the case of monostatic radar this point was the target's closest point of approach. For bistatic radar the midpoint in the first detection envelope was utilized. The jamming power arriving at the detecting radar, given the geometry of this midpoint position, was then used to calculate the detection range for the target. Additionally, in the computation of bomber detections and "shoot" events, no provision is made for jammers which may have been shot down prior to the time of detection. This assumption is not required in the case of missile detections where detection events for missiles take into consideration reduction in jamming levels caused by previous destruction of jammers.

The model is limited to four frequency bands for frequency separation of individual radars in the jamming environment.

3. Target Tracks

The principle target track assumptions are concerned with minor anomalies from the real world situations. These assumptions do not detract excessively from the simulation results, but do ease computation and simplify programming. A fixed altitude for the raider and missiles, and the lack of course changes on the part of raiders are such simplifying assumptions. In the case where a raider or missile track does not change more than 0.1 nautical mile (nm) in either the x or y direction, the target is assumed to fly

parallel to the non-changing axis. If the target does not move more than 0.1 nm in both the x and y directions, the target is assumed to be stationary. Additionally, raiders who have completed their missile launches continue on their preset course rather than exiting the battle area. Although these targets will be engaged by the defensive forces if higher priority targets are not present, the resulting target hits can be readily "backed out" of the simulation results by the user if desired. On the other hand, this assumption does allow raiders that are providing standoff jamming protection for the missiles to continue to close the defensive forces and thereby increase the effectiveness of their jamming.

Other less general assumptions are covered as they occur in the details of the subroutine descriptions in Appendix B.

III. DEMONSTRATION PROBLEM

This section presents a demonstration of the use of FADS in analysing a hypothetical problem involving bistatic radar systems and surface-to-air missiles (SAM). The problem is to select the maximum range required for the SAM system to be employed by a small surface combatant (SSC). The proposed SAM system is to protect the major units of a naval task force involved in an engagement with enemy missile systems. The SSC will have a bistatic search radar and the SAMs will be active homing missiles requiring no fire control director. As such, the SSC will emit no radiation. The SAM characteristics and performance capabilities have been specified with the exception of the maximum range. Two range capabilities will be considered, a 100 nautical mile missile and a 50 nautical mile missile. The problem thus becomes that of determining the relative effectiveness of the SSC when equipped with the different range missiles. As discussed in the introduction, the measure of effectiveness used to evaluate the missiles is the number of enemy missiles penetrating to the force center per unit of time. That is, the number of missiles arriving at the force center per minute during the time period in which missiles are impacting at the force center.

A. THE SCENARIO

The BLUE force consists of one High Value Target (HVT) and eight small escorts. The BLUE force is maneuvering within range of enemy air forces. The HVT is located at force center and the pickets are equally spaced on a circle one hundred miles from the HVT. The picket force is silent except for one early warning aircraft which carries the transmitter for the bistatic radars. For purposes of this analysis, the aircraft is assumed to be at force center and at 20,000 feet.

The RED force has detected the radar transmissions of the aircraft and the HVT and have launched an attack of twenty-one aircraft accompanied by three jammers. The RED aircraft will attack simultaneously in three sections, separated by forty degrees of bearing. The RED air force has practiced this maneuver to the point that the maximum deviation from simultaneous launch will be fifteen minutes. Each aircraft will take a maximum of three minutes to launch its missiles. The three sections will approach their target from 320, 000, and 040 degrees true. The jammers will carry equipment to cover all radar bands used by the BLUE force. Each section will consist of one heavy and six light attack aircraft.

The RED air force will attempt to penetrate the BLUE defense and sink the HVT. The HVT is considered to be unarmed. In order to target a BLUE unit in this dispersed formation, a RED unit will have to have positive identification of the

unit as hostile. An ECM fix, or visual ID will be adequate. As the BLUE pickets are not radiating, no ECM fix may be obtained. As the pickets are assumed to be smaller than a destroyer escort, and the RED aircraft will fly at a minimum altitude of 33,000 feet, no positive visual identification will be allowed.

The radio and radar silence of the pickets leaves them without the ability to coordinate their defense. No BLUE intercept will be allowed if the intercept point is within two miles of another BLUE unit's assigned position. The BLUE units will use a shoot-look-shoot doctrine.

B. THE INPUTS

1. The Units

a. The BLUE Units

The BLUE units consist of the HVT, eight pickets, and one early warning aircraft. Positions are as indicated in the scenario.

(1) HVT. For the purpose of this analysis the HVT is assumed to be unarmed.

(2) Pickets. The pickets will employ a type 4, bistatic, band 4 radar with an antenna height of 50 feet. They will carry thirty missiles in canisters. The probability of a correct target identification will be .95.

(3) Early Warning Aircraft. The BLUE early warning aircraft will be unarmed. It will carry a type 4 radar and serve as the transmitter for the pickets' bistatic receiver.

b. The RED Units

The three classes of RED units are heavy attack, light attack and jammers.

(1) Heavy Attack. The RED heavy attack aircraft will carry a type 5 radar and have the following characteristics.

WEAPONS: 1 type I ASM
4 type II ASM

SPEED: 415 Knots

JAMMERS: none

RADAR CROSS SECTION: 40 square meters

ALTITUDE: 35,000 feet

(2) Light Attack. The RED light attack aircraft will carry a type 5 radar and have the following characteristics.

WEAPONS: 2 type II ASM

SPEED: 420 Knots

JAMMERS: none

RADAR CROSS SECTION: 13 square meters

ALTITUDE: 33,000 feet

(3) Jammers. The RED jammer will carry no radar. It will have the following characteristics.

WEAPONS: none

SPEED: 417 Knots

JAMMERS: bands 1-4
250 watts/MHZ/band
359° beam width
-20 db side-lobe ratio
jamming target, HVT

RADAR CROSS SECTION: 32 square meters

ALTITUDE: 40,000 feet

2. The Radars

The two radars which the simulation will use to play the simulation are the class 4 and class 5 radars. Class 4 radars are found on all pickets and class 5 on all raiders.

a. Class 4 Radar

TYPE: Bistatic

β: 120 $nm/m^{\frac{1}{2}}$

α: 30 $nm/m^{\frac{1}{2}}$

SIDE LOBE RATIO: -20 db

MAXIMUM INSTRUMENTED RANGE: 200 miles

RADAR FREQUENCY BAND: 4

BEAM WIDTH: 10 degrees

TRANSMITTER UNIT: aircraft overhead HVT

TRANSMITTER ANTENNA HEIGHT: 20,000 feet

b. Class 5 Radar

TYPE: Monostatic

β: 100 $nm/m^{\frac{1}{2}}$

α: 50 $nm/m^{\frac{1}{2}}$

SIDE LOBE RATIO: -20 db

MAXIMUM INSTRUMENTED RANGE: 200 miles

RADAR FREQUENCY BAND: 3

BEAM WIDTH: 3 degrees

3. <u>The Reaction Time Distributions</u>

The reaction time distributions for target processing by the picket are displayed below.

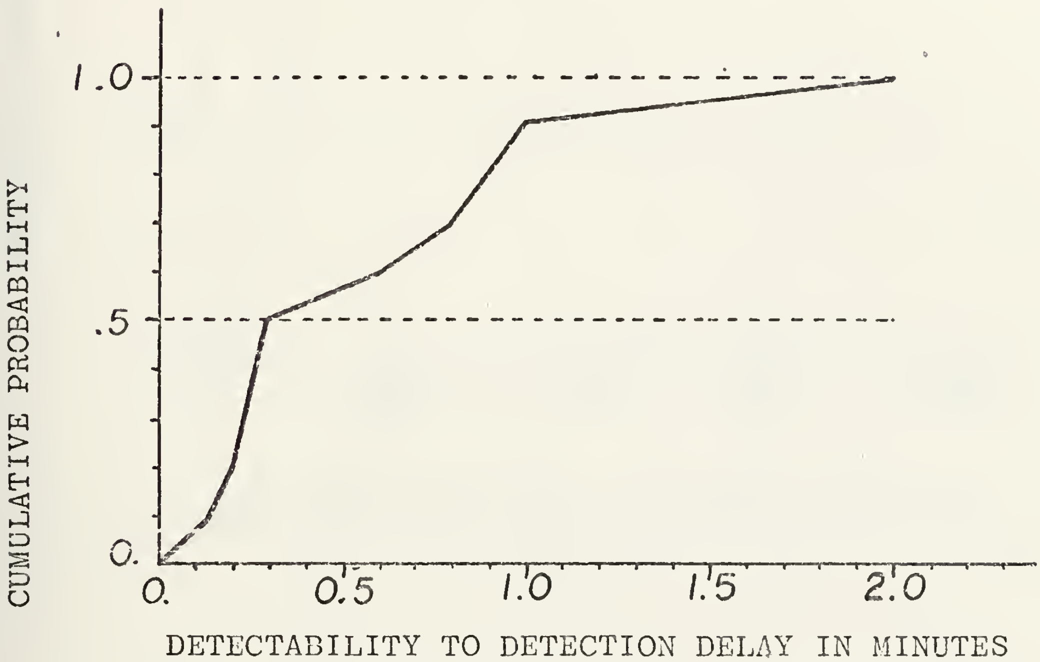


Figure 1

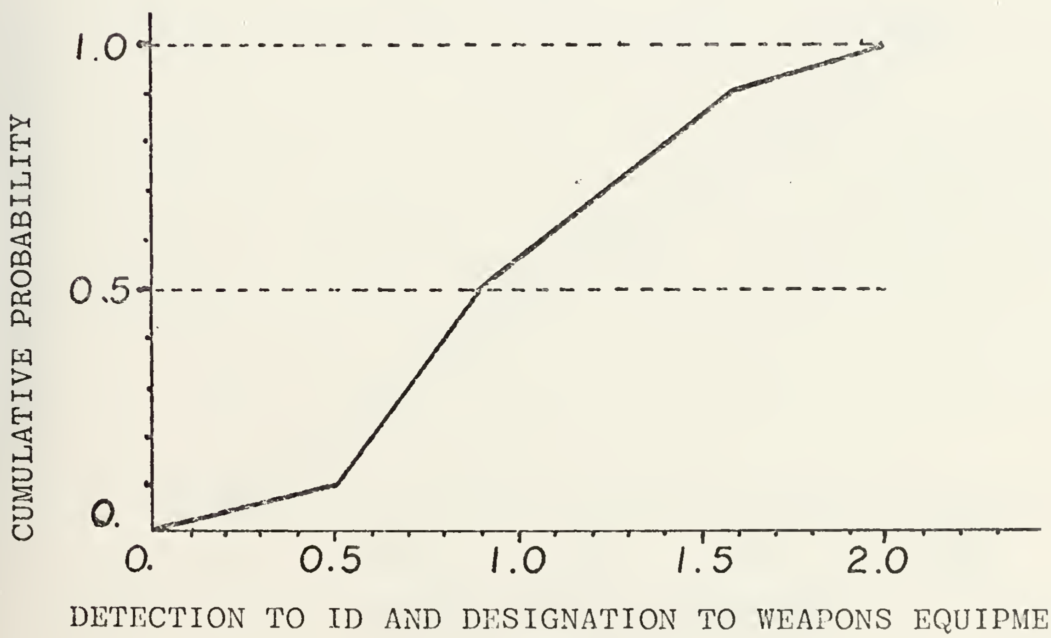
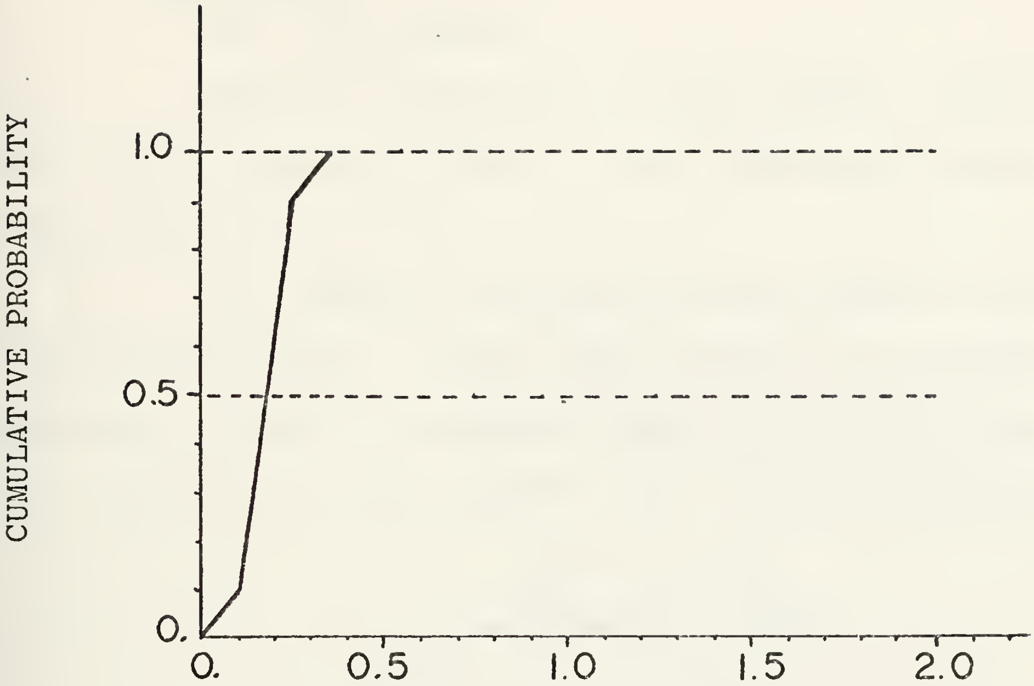
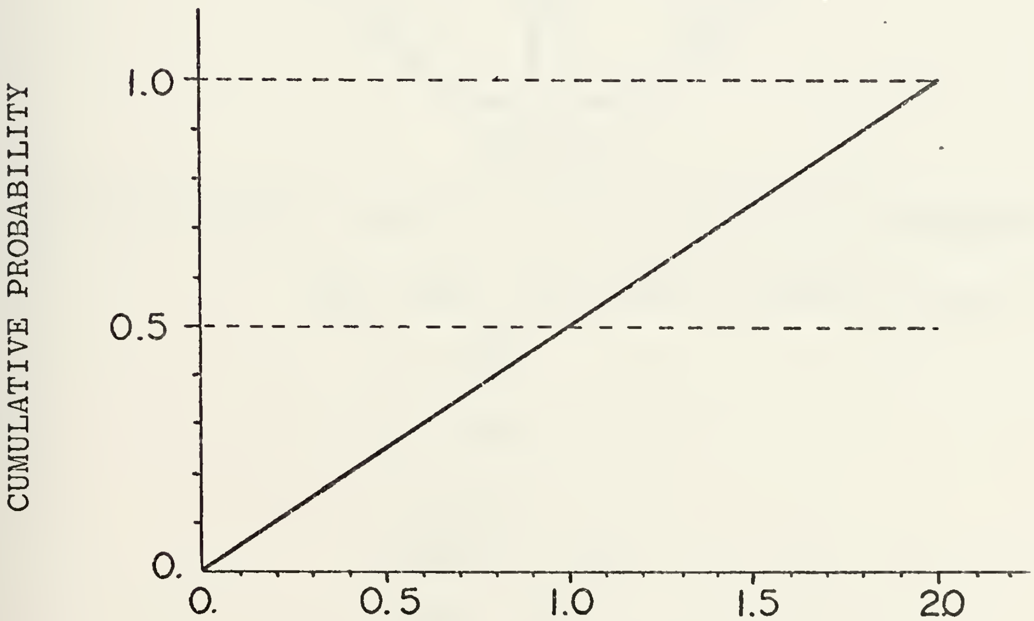


Figure 2



ASSIGNMENT TO FIRING CONSOLE IN MINUTES

Figure 3



FIRING DELAY IN MINUTES

Figure 4

4. The Weapons

a. The BLUE Weapons

There are two types of BLUE weapons. Both are SAMs and are denoted as SAM LR (long range) and SAM SR (short range).

(1) SAM LR. This will be the 100 mile version of the BLUE missile. It will have two sets of intercept parameters. The first is for a large, slow, high flying target. The second is for a small, fast, medium altitude target.

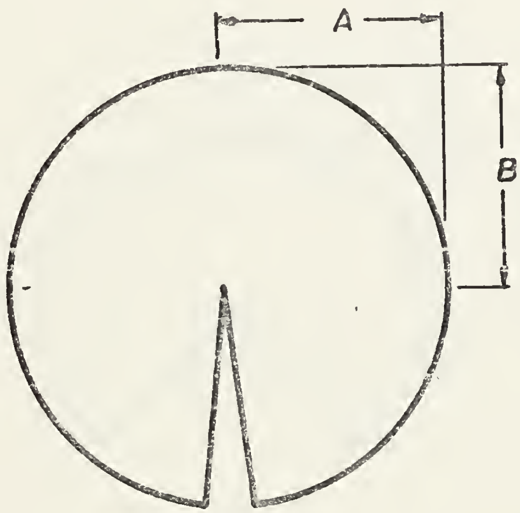


Figure 5

SAM Intercept Contour for High Altitude Engagement

The input value for the SAM LR class are:

A = 100

B = 100

N = 2.0

SLOPE = -20

SEEK = -5

SPEED = 1400 Knots

PROBABILITY KILL FOR SINGLE SHOT (PKSS) = .88

PKSS (CROSSING) = .79

SYSTEM RELIABILITY = .95

LAUNCH RELIABILITY = .99

MAXIMUM ALTITUDE = 59,000 feet

MINIMUM RANGE = 0.5 miles

The intercept parameters A, B, N, SLOPE, and SEEK are defined in the description of subroutine TIMMY in Appendix B.

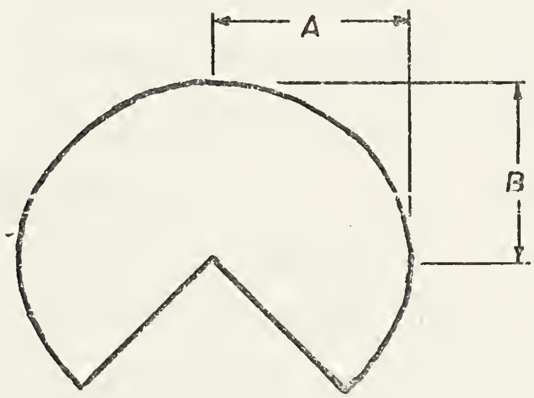


Figure 6

SAM Intercept Contour for the Medium Altitude Engagement

A = 100

B = 90

N = 2.8

SLOPE = -1

All other parameters are the same as high altitude engagement

(2) SAM SR. This will be the 50 mile version of the BLUE missile. The intercept contours will be the same as those for the SAM LR except the down range and cross range values are half the SAM LR values.

b. The RED Weapons

The RED aircraft carry two types of ASMs.

(1) ASM Type I.

SPEED: 750 Knots

ALTITUDE: 20,000 feet

LAUNCH RANGE: 120 miles

LAUNCH RELIABILITY: 0.85

INFLIGHT RELIABILITY: 0.80

RADAR CROSS SECTION: 1.79 square meters

(2) ASM Type II.

SPEED: 900 Knots

ALTITUDE: 15,000 feet

LAUNCH RANGE: 110 miles

LAUNCH RELIABILITY: 0.85

INFLIGHT RELIABILITY: 0.9

RADAR CROSS SECTION: 1.01 square meters

C. RESULTS

A summary of the results for 100 iterations is presented below.

1. ASM Results.

	SAM SR	SAM LR
Available	5100	5100
Destroyed with launch platform	172 (3.4%)	246 (4.8%)
Launched	4928 (96.6%)	4854 (95.2%)
Failed at launch	724 (14.7%)	750 (15.4%)
Failed in flight	429 (8.7%)	418 (8.6%)
Destroyed by SAMs	0	0
Impact on HVT	3775 (76.6%)	3686 (75.9%)

2. SAM Results.

	SAM SR	SAM LR
Available	24000	24000
Launched	3401	6440
Kills	2116 (62.2%)	2398 (37.2%)
ASM	0	0
Raiders	2116	2398
Overkills	605 (17.8%)	2694 (41.8%)
Failures	680 (20.0%)	1348 (20.9%)

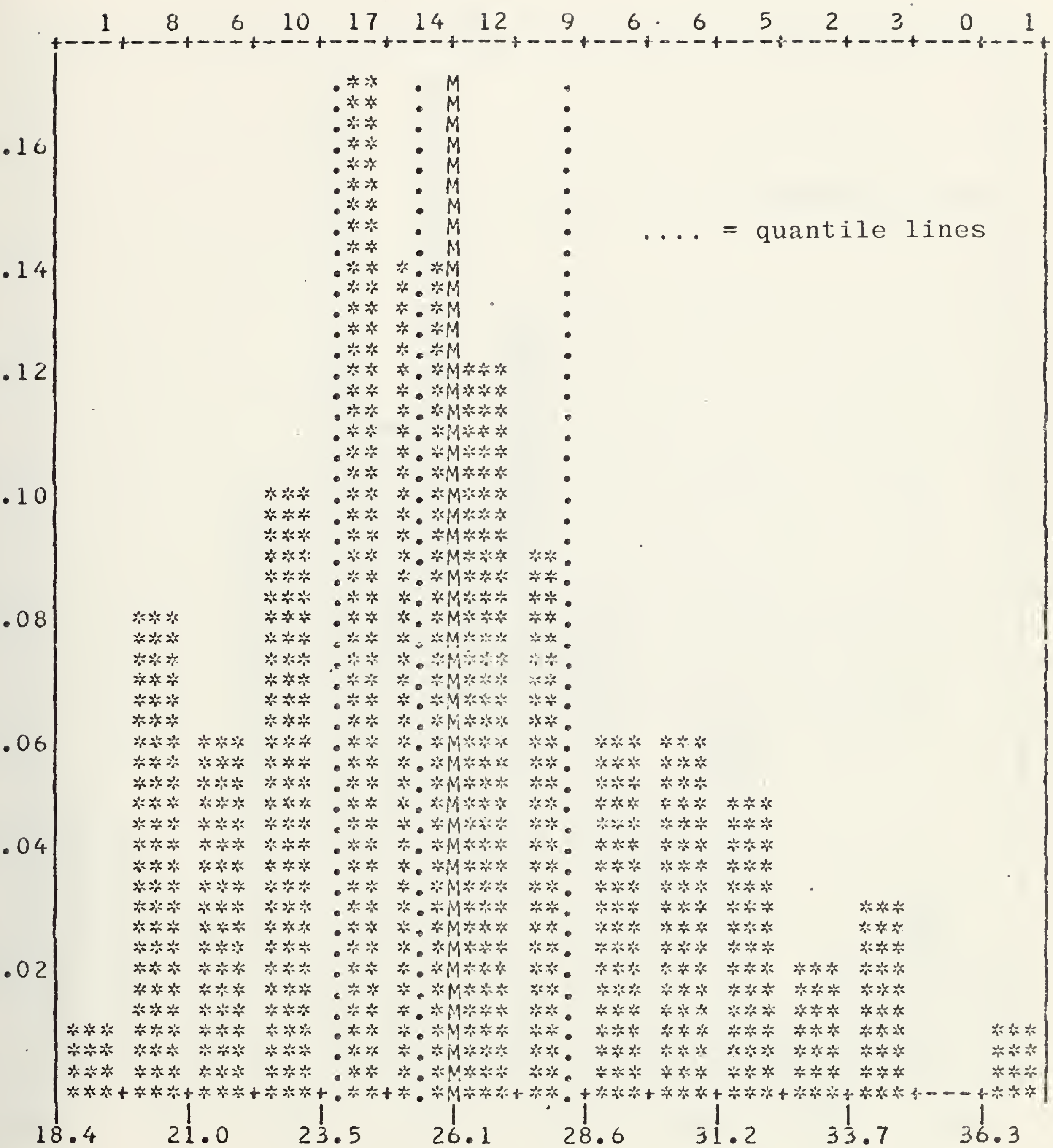
3. ASM Impact Intervals (minutes).

	SAM SR	SAM LR
Mean	1.468	1.432
Variance	0.033	0.037
Skewness	-0.060	0.070
Kurtosis	-0.620	-0.532
Minimum	1.091	0.966
Maximum	1.849	1.862

4. ASM Impacts per Minute.

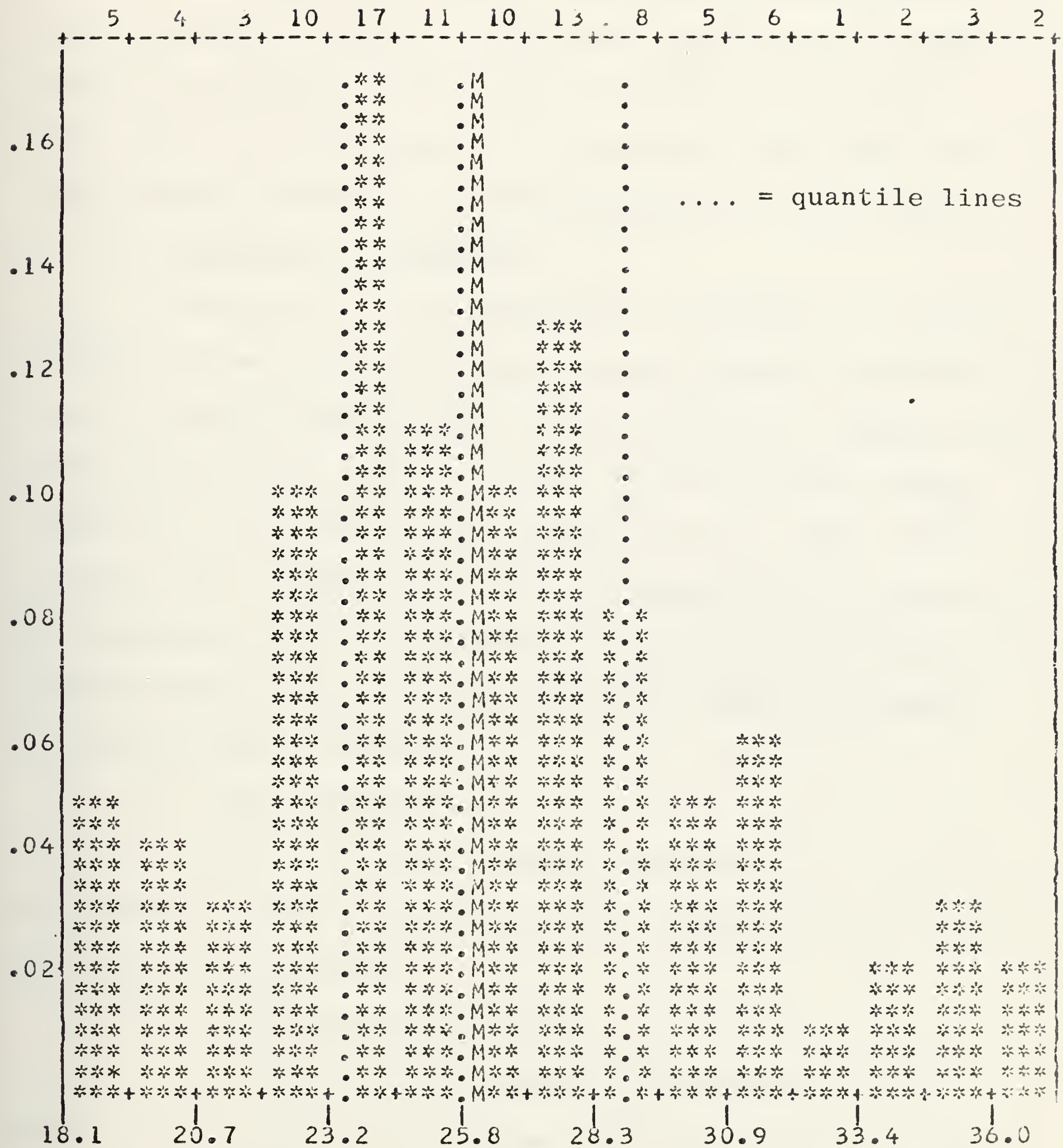
	SAM SR	SAM LR
Mean	26.101	26.192
Variance	14.651	17.541
Skewness	-0.584	0.453
Kurtosis	-0.006	-0.136
Minimum	18.422	18.129
Maximum	37.567	37.246

5. Histograms of the ASM impacts per minute are shown in Figures 7 and 8.



Histogram of the ASM Impacts per Minute, SAM SR

Figure 7



Histogram of the ASM Impacts per Minute, SAM LR

Figure 8

D. ANALYSIS

The analysis of the results consists of three parts. The first part contains a parametric evaluation of the ASM impacts per minute for the two SAM systems. The second part provides a nonparametric evaluation. Part three provides general comments concerning anomalies in the results.

1. Parametric Evaluation

The purpose of the parametric evaluation is to test whether the mean ASM impacts per minute against the SAM SR (u_{50}) is significantly greater than the mean ASM impacts per minute against the SAM LR (u_{100}). With its shorter range capability, the SAM SR system should attain a lower number of kills than the SAM LR system. Consequently the number of ASM impacts on the HVT that occur during the defense by the SAM SR system should be greater than the number of impacts during the SAM LR defense.

a. Null Hypothesis

H_0: $u_{50} \leq u_{100}$; SAM SR effectiveness is equal to or greater than the SAM LR system's as indicated by the mean ASM impacts per minute.

b. Alternative Hypothesis

H_1: $u_{50} > u_{100}$; SAM SR is less effective than SAM LR and therefore has a higher mean ASM impacts per minute.

c. Significance Level

A significance level of 0.05 was used in testing H_0.

d. Statistical Test

The null hypothesis was tested against the alternative hypothesis using a t test. The t statistic for testing equal means is:

$$t = \frac{\bar{x}_{50} - \bar{x}_{100}}{\left[\frac{n_{50}+n_{100}}{n_{50}n_{100}}\right]^{\frac{1}{2}} \left[\frac{n_{50}s_{50}^{2}+n_{100}s_{100}^{2}}{n_{50}+n_{100}-2}\right]^{\frac{1}{2}}}$$

where:

$\bar{x}_{50}$ = sample mean for SAM SR. (26.10)

$\bar{x}_{100}$ = sample mean for SAM LR. (26.19)

s_{50}^2 = sample variance for SAM SR. (14.651)

s_{100}^2 = sample variance for SAM LR. (17.541)

n_{50}= n_{100}= number of samples for SAM SR and SAM LR. (100)

The null hypothesis will be rejected if the computed test statistic t is greater than or equal to the table value for the 0.95 point of t distribution with 198 degrees of freedom. The table value is:

$$t_{.95,198} = 1.96$$

e. Decision

The computed test statistic is t = -0.1578. Since the computed t ratio is less than the tabulated value, the null hypothesis cannot be rejected at the 0.05 level.

f. Conclusion

The effectiveness of the SAM SR system was equal to or greater than the effectiveness of the SAM LR system at the 0.05 level of significance.

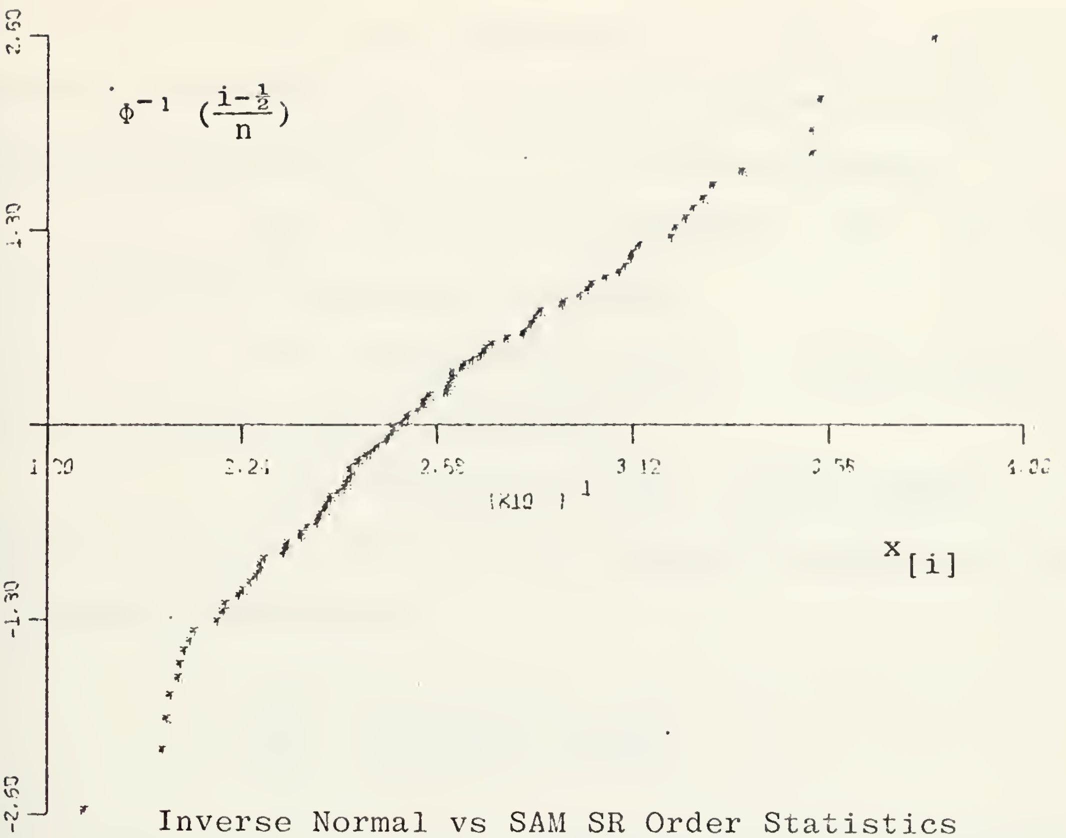
g. Justification for Use of the t Test

In order to use the t test for difference in means, three basic conditions had to be met. These were: i) the sample means were independent; ii) the underlying distributions from which the samples were drawn was normal; and iii) the underlying distributions had equal variances.

(1) Independence. The independence of the sample means was a reasonable assumption since the individual samples were obtained from different simulation runs using different random number streams.

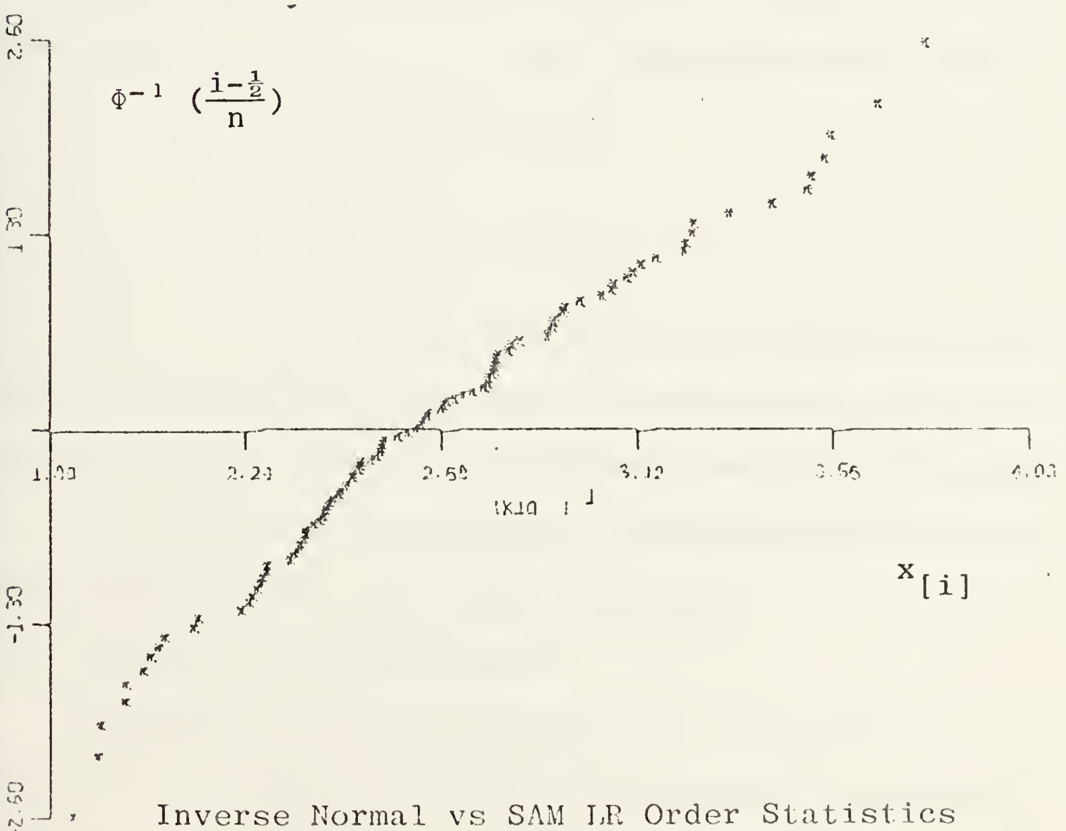
(2) Normality. The normality of the sample mean values could be asserted on the basis of the sample size (100) and the central limit theorem or by demonstrating that the means were computed from samples which came from a normal distribution. The underlying normal distribution can be shown on the basis that both samples were unimodal and symetric, and that the plots of their order statistics against the inverse normal probability scale was linear. The unimodality of the samples is seen in the histograms in figures 7 and 8. Symmetry is evident in both histograms and supported by the fact that the kurtosis of the sample values is close to zero. The order statistic plots, which are clearly linear, are shown in figures 9 and 10.

(3) Equal Variance. The relationship between the variance of the population from which the samples were drawn is tested using the F ratio. This test is applicable since the populations are normally distributed.



Inverse Normal vs SAM SR Order Statistics

Figure 9



Inverse Normal vs SAM LR Order Statistics

Figure 10

(a) Null Hypothesis. $H_0: \sigma^2_{50} = \sigma^2_{100}$; the population variance (σ^2_{50}) of the SAM SR system is equal to the population variance (σ^2_{100}) of the SAM LR system.

(b) Alternative Hypothesis. $H_1: \sigma^2_{50} \neq \sigma^2_{100}$; the population variances are not equal.

(c) Significance Level. A significance level of 0.05 was used in testing H_0.

(d) Statistical Test. H_0 was tested against H_1 using the ratio of the unbiased estimators of the two population variances.

$$F = \frac{s^2_{100}}{s^2_{50}} = \frac{17.541}{14.651} = 1.1973$$

The null hypothesis will be rejected if the computed ration F is greater than the tabulated value of $F_{(1-\alpha/2;\ 99,\ 99)}$ or less than $F_{(\alpha/2;\ 99,\ 99)}$. The tabulated values are:

$$F_{(.975,\ 99,\ 99)} = 1.451$$

$$F_{(.025,\ 99,\ 99)} = 0.684$$

(e) Decision. Since the computed ratio F is not greater than or less than the tabulated values for the F distribution, the null hypothesis cannot be rejected.

(f) Conclusion. The samples were drawn from populations which had equal variances.

2. Nonparametric Evaluation

The purpose of the nonparametric evaluation is to test whether the sample values for the SAM SR system and the SAM LR system are from the same population.

a. Null Hypothesis

H_0: There is no difference in the number of ASM impacts per minute when defending with the SAM LR system or with the SAM SR system.

b. Alternative Hypothesis

H_1: There is a difference.

c. Significance Level

A significance level of 0.05 was used in testing H_0.

d. Statistical Test

The Kolmogorov-Smirnov (K-S) two-sample test was used to test the null hypothesis. The K-S test requires the use of independent samples. Independence was shown in D.1.g.(1) above. The K-S test is conducted using the statistic

$$K = \max \left| F_n(X) - G_m(Y) \right|$$

where $F_n(X)$ is the empirical cumulative distribution function for the first set of sample values and $G_m(Y)$ is the empirical cumulative distribution function for the second.

The null hypothesis will be rejected if there is a large enough deviation between the two sample cumulative distributions. The critical values of D for the level of significance of 0.05 and 100 observations in each sample is 0.19233.

e. Decision

The computed value of K was determined to be 0.0800. Since the computed value of K is less than the tabulated critical value, the null hypothesis cannot be rejected at the 0.05 level. A plot of the empirical cumulative distribution functions for the two sets of sample values is provided in figure 11. This plot clearly shows the close relationship between the two underlying populations.

f. Conclusion

There was no difference in the defensive capabilities of systems in the environment in which they were examined.

3. General Comments

The principal anomalies present in the data include the following:

a) a lack of difference in the SAM LR and SAM SR effectiveness;

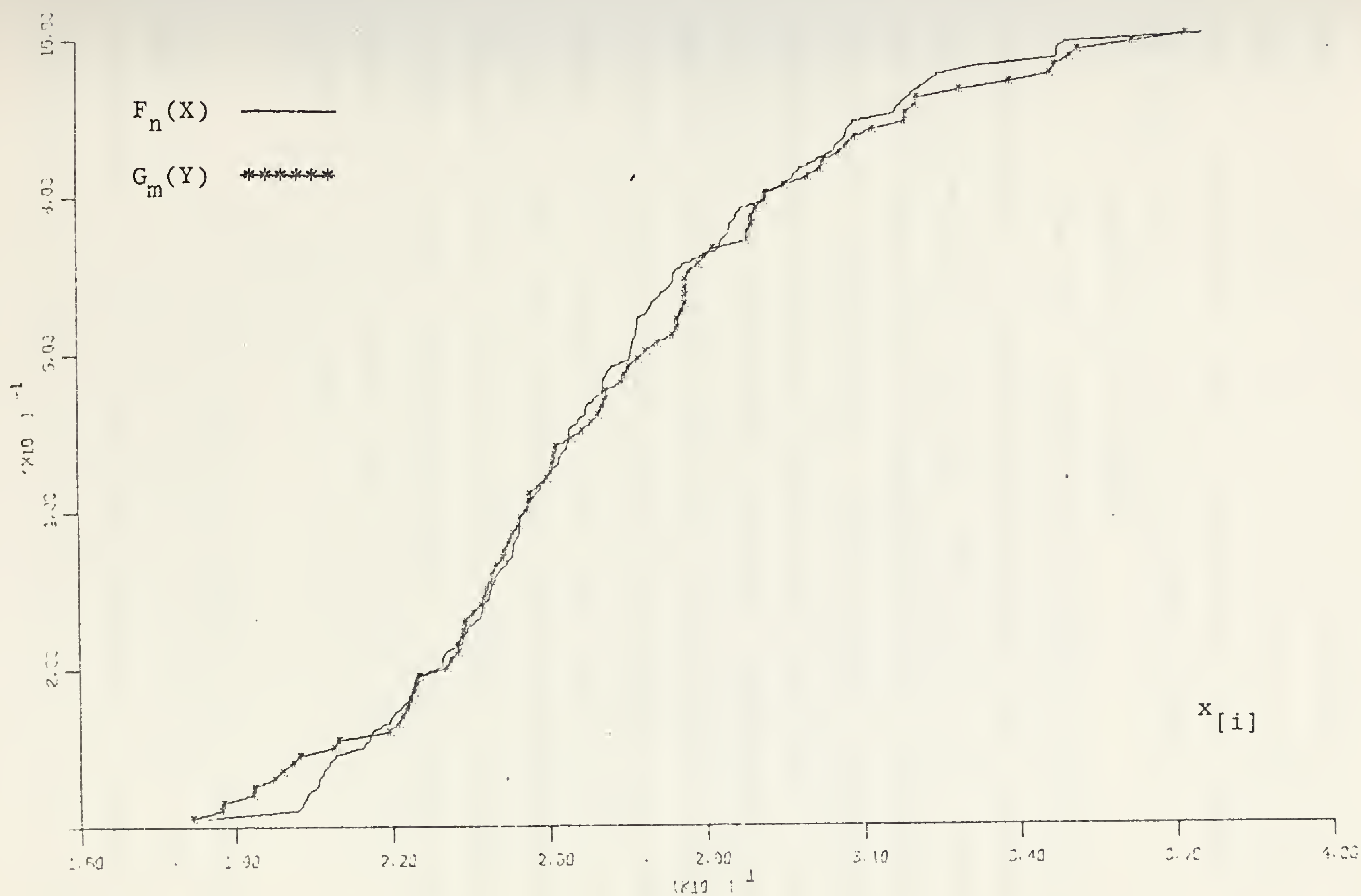
b) no SAM kills of ASMs;

c) a greater number of overkills for the SAM LR;

d) the number of SAM failures; and

e) the short ASM impact interval.

These anomalies do not significantly detract from the conclusions presented in the parametric and nonparametric evaluations.



Plot of Empirical Cumulative Distribution Function for Both SAMs.

Figure 11

a. Lack of Difference in Effectiveness

The most probable reason for the lack of difference in the effectiveness of the two SAM systems was the jamming environment. The reduced detection ranges experienced in jamming will tend to preclude the use of SAMs at the longer ranges available to the SAM LR system.

b. No SAM Kills of ASMs

The lack of SAM kills of ASMs was primarily due to the geometry of the ASM launch positions and the program logic concerning SAM targeting. The program logic requires that the closest target be considered as most threatening and therefore engaged first. In the problem, the ASM launch positions were less than 10 miles from their point of closest approach to the SAM ship positions. Consequently, the ASMs with their greater speed were closer than the raiders to the SAM ships for only the first 30 seconds after their launch. After that period, their speed caused them to be at a greater range from the SAM ship. Thus the SAM ships had a 30 second period in which to detect, acquire, and designate an ASM as the most threatening target. The reaction time distributions for the SAM ships precluded such a short reaction time in most cases.

c. Overkills

The higher number of overkills for the 100 mile SAM was due to the longer flight time of the SAM. This coupled with a lack of coordination resulted in a greater opportunity for another unit to obtain a kill on the target

while a second SAM was in flight. Targets that had been destroyed prior to the scheduled SAM launch time were not engaged.

d. SAM Failures

The number of SAM failures includes SAMs which were fired on the ASMs that impacted on the HVT while the SAM was in flight. These attempted engagements occurred since the SAM units were not given the capability to determine the ASM impact point.

e. Short ASM Impact Interval

The tightness of the ASM impact interval may appear to be pessimistic in relation to a real world threat. However, this should not detract from the comparison of the two SAM systems.

IV. MODEL INPUTS

The inputs required to use FADS fall into three groups. These are the X vector, the target list assignments, and the target lists themselves. The amount of actual input required depends upon the size of the problem to be analyzed.

A. THE X-VECTOR

All X vector inputs are real values and are read into the X vector in the following way. Each input card must contain six numbers. The first number is the integer address in the X vector. This integer should be placed in columns 1-10, right justified. The remaining five numbers are the values to be inserted in the five X vector positions beginning with the position indicated in columns 1-10 of the input card. The format for each card with X vector data is (I10,5F10.3). If there are no data to be read into a position, no input is required, unless the position falls within the scope of another input card. The end of the X vector inputs is signaled by the inclusion of a zero card. This card contains a zero in column 10 and zeros in the five data positions. A brief overview of the organization of the input array is provided below. Details concerning the individual sub-sections are provided at the beginning of each sub-section. Specifics dealing with the input values are discussed in the description of the particular input value and as footnotes to the individual sections.

The following is the organization of the input X vector:

X(1) to X(100), Control switches and program constants

X(101) to X(500), Ship Classes

X(501) to X(900), SAM Classes

X(901) to X(1200), Radar Classes

X(1201) to X(1400), Ship Positions

X(1401) to X(1800), Delay Functions

X(1801) to X(2000), Raid Classes

X(2001) to X(2200), Raid Weapon Classes

X(2201) to X(2740), Initial Raider Positions.

1. Control Switches

These inputs include logic switches for varying the general play of the game. They also provide the means for controlling the output in the form of debug and array printout switches. Several of the control switches are related to inputs for specific systems which appear in later sections and must be consistent with later usage. Other portions of this input section are used to store values for generating statistics and for internal use by the model. These portions should not have inputs since their value will be set as required by the model.

X( )	Inputs
1	Case number
2	Blue force equipment availability switch: 0 = 100% availability 1 = use input availabilities

X( )	Inputs
3	Blue force radar detection range printout in Event 2: 0 = no print 1 = print
4	Fleet coordination of Blue forces: 0 = none 1 = coordinated defense
5	Blue ship out of action after N hits: 0 = yes 1 = no (N is a ship class input)
6	Number of iterations. (Must be at least 1. If X(85) used, must be at least 10 and less than 101.)
7	Horizon/Sector Scan for Blue fire control radar system enabled: 0 = no 1 = yes if equipment capable
8	Blue SAM Sector enable: 0 = don't use sectors 1 = sectors used
9	Print control: 0 = none 1 = selected output tables and intermediate debug printouts. Set to 0 by the model after first iteration.
10	Detectability to detection delay: 0 = use input distribution function 0.XX = a fraction; time for Raider or ASM to move this portion of calculated detection range.
11	ISEED value. Initial seed value used in generation of random numbers. Can be up to a seven digit number.
12	X-array print out switch: 0 = no print 1 = print out X-array as read in, target lists, and target list assignments.
13	Random number counter. Set to 0 by model. Maintains running count of random numbers drawn.

X( )	Inputs
14	Total number of random numbers used. Computed by model.
15	Converted X-array print out switch: 0 = no print 1 = print out X-array as converted by model.
16	Blue first shoot control: 0 = Blue may fire as soon as Red detected. 1 = Blue may only fire after first Red missile launch.
17	Number of jamming raiders: 0 = no jammers 1 = one jammer 2 = two jammers, etc.
18-19	Spares
20	Altitude in feet above which a raider will be considered as being at "high altitude" for purposes of obtaining a visual identification of a non-radiating picket.
21	Probability of "high altitude," from X(20), raider launching ASM at a non-radiating picket.
22	Probability of "low altitude," below X(20) input, raider launching ASM at non-radiating picket.
23	Maximum range (nautical miles) at which a raider is able to obtain a visual identification of a Blue surface unit.
24	Geographical plots: 0 = no plots required 1 = print plot of Blue positions 2 = print plot of raider tracks 3 = print both plots
25-50	Spares
(51-60)	(Debug switches - cleared at end of first iteration a printout will occur if input value is 1.)
51	Ship table.
52	Raid table.
53	SAM table.

X( )	Inputs
54	Random numbers used.
55	Subroutine login/logout.
56	SAM/ASM engagement, ASM impact, and SAM kill.
57	ASM detection data.
58	Closest point of approach and IBAT data.
59	SAM-ASM Intercept data (angles).
60	Store and Get Event Log.
(61-70)	(Statistics generated by the model, ignore for input.)
61	Total ASM's fired for all iterations.
62	Total ASM's failed at launch.
63	Total ASM's killed by SAM's.
64	Total ASM's failed in flight.
65	Total ASM's destroyed with launch platform.
66	Total SAM launches.
67	Total SAM failures.
68	Total SAM overkills.
69	Total ASM successfully launched.
70	Total ASM hits.
71-74	Spares
75	Number of SAM rounds remaining in Fleet. Internal use by the model.
76	Hazard zone radius, nm. If predicted intercept of target is within this range of another ship, then the firing ship must hold fire until the predicted intercept point has moved beyond this range.
77	Breakpoint altitude, ft, used to select proper SAM velocities, ranges, and intercept contours, for "high" or "low" SAM contours.

X( )	Inputs
78	Ship position inputs coordinate system: 0 = Polar (000 True is Y axis). 1 = Cartesian (X = ; Y =).
79-80	Spares
81	Radar horizon factor (default value 1.23).
82	End of game time (default value 120 min).
83	Launch interval for one raid, min.
84	Launch interval for all raids, min.
85	Number of ship for which the display of hit frequency and hit interval is desired. Enter 0 if analysis is not desired or if the number of iterations is less than 10 or greater than 100, in which case the histogram can not be generated by the model.
86	Interval for initial launch of Red SSM's, min.
87	Recognition delay, in minutes, for stationary Red raiders. For use when X(10) is not zero.
88	Number of ships.
89	Number of raiders (including Red aircraft which are only jammers, those without weapons, and Red surface/subsurface units).
90-100	Spares

2. Ship Class Inputs

The different types of BLUE ships in the simulation are specified by the ship class inputs. Included under one listing are all of the ships of a particular class (e.g. DDG, CG, FFG) which possess similar weapons and detection capabilities. A maximum of 10 different classes may be used. Ship class 1 is input into X(101) to X(140); ship class 2 into X(141) to X(180); etc.

X( )	Inputs
101	SAM/SSM battery #1 class[1]
102	Sam/SSM battery #2 class
103	SAM/SSM battery #3 class
104	SAM/SSM battery #4 class
105	Search radar #1 class number[2]
106	Search radar #2 class number
107	Search radar #3 class number
108	Search radar #1 antenna height (ft)
109	Search radar #2 antenna height (ft)
110	Search radar #3 antenna height (ft)
111	Fire control radar #1 antenna height (ft)
112	Fire control radar #2 antenna height (ft)
113	Fire control radar #3 antenna height (ft)
114	Fire control radar #4 antenna height (ft)
115	Coordination capability for this ship class: 0 = none (override) 1 = coordination X(14) if fleet coordination is in use.
116	Probability of correct identification and designation to Weapon Direction Equipment (WDE).
117	Class of RTDF #2 (detection to ID and designation to WDE).

---

[1] Enter 1 if the battery has SAM/SSM's where class inputs are listed in X(501) to X(540); enter 2 if SAM/SSM inputs are in X(541) to X(580); etc.

[2] Enter 1 if the search radar class inputs for the respective search radar are contained in X(901) to X(920); enter 2 if the search radar class inputs are in X(921) to X(940); etc. Note that the Fire Control Radar class is listed with its associated SAM class inputs.

X( )	Inputs
118	Class of RTDF #3 (designation to assign FCS).
119	Class of RTDF #5 (firing delay).
120	Class of RTDF #7 (decisional delay for horizon/ sector scan mode -- used with X(7)).
121	Number of hits required to put ship out of action.
122	Left SAM battery #1 launcher train angle limits (deg. true).
123	Right SAM battery #1 launcher train angle limits (deg. true).
124	Left SAM battery #2 launcher train angle limits (deg. true).
125	Right SAM battery #2 launcher train angle limits (deg. true).
126	Left SAM battery #3 launcher train angle limits (deg. true).
127	Right SAM battery #3 launcher train angle limits (deg. true).
128	Left SAM battery #4 launcher train angle limits (deg. true).
129	Right SAM battery #4 launcher train angle limits (deg. true).
130	Ability to determine target at which an ASM has been fired: 0 = No 1 = Yes
131	Designation for non-radiating picket units using bistatic radar: 0 = No; unit will be targeted using only the normal target list algorithm. 1 = Yes; unit will be targeted using both the normal target list algorithm and the additional algorithm for non-radiating pickets. If some bistatic units are to be used in the main force, a separate ship class must be entered for them.
132-140	Spares

X( )	Inputs
141-180	Ship class 2
181-220	Ship class 3
221-260	Ship class 4
261-300	Ship class 5
301-340	Ship class 6
341-380	Ship class 7
381-420	Ship class 8
421-460	Ship class 9
461-500	Ship class 10

3. SAM/SSM Class Inputs

These inputs specify the BLUE SAM/SSM system capabilities in terms of missile flight characteristics, fire control radar class, salvo firing doctrine, missile reliabilities and kill probabilities. A maximum of 10 different SAM/SSM classes may be input. SAM/SSM class 1 is input into X(501) to X(540); SAM/SSM class 2 into X(541) to X(580); etc.

X( )	Inputs
501	Max range, nm, against "high" altitude target.
502	Average speed, knots, for expected intercept range against "high" altitude target (converted by the model to nm/min).

X( )	Inputs
530	A, nm, "high" altitude target[3]
504	B, nm, "high" target[3]
505	N, for "high" target[3]
506	SLOPE against "high" target[3]
507	SEEK, nm, for "high" target[3]
508	Max range, nm against "low" altitude target
509	Average speed, knots, against "low" altitude target (converted by the model to nm/min).
510	A, nm, against a "low" target
511	B, nm, against a "low" target
512	N, for "low" target
513	SLOPE for "low" target
514	SEEK, nm, for "low" target
515	Minimum range, nm
516	Maximum altitude, ft
517	Number of Fire Control Radar (FCR) channels for FCR Type 1. (0 = missile not requiring a dedicated fire control system.)
518	Class of FCR - Type 1
519	Number of FCR channels for FCR Type 2
520	Class of FCR - Type 2
521	Number of FCR channels for FCR Type 3
522	Class of FCR - Type 3

[3] The definitions of A, B, N, SLOPE, and SEEK are contained in the description of subroutine TIMMY in Appendix B.

X( )	Inputs
523	Horizon sector scan capability for fire control radars: 0 = No (used when X(7) set) 1 = Yes input as XXX = unit digit FCR Type 1 ten digit FCR Type 2 hundred digit FCR Type 3
524	Number of SAM's per salvo
525	Launch cycle time, min
526	Probability that launcher is available (used only when X(2) is set to 1).
527	Magazine size
528	Spare
529	Probability of single-shot kill
530	System reliability
531	Reliability (launch, guidance, and fuse)
532	Terminal guidance time, min (if HAW set to 0).
533	Kill assessment time, min
534-535	Spares
536	Flag for Blue surface to surface missiles: 0 = SAM 1 = SSM
537	Single shot kill probability for a crossing target
538-540	Spares
541-580	SAM Class 2
581-620	Sam Class 3
.	.
.	.
.	.
861-900	SAM Class 10

4. Radar Class Inputs

Search and fire control radars for both BLUE and RED units are specified in this section of the inputs. Search radars classes are related to the ship class and raid class inputs. The fire control radars are used with the appropriate SAM/SSM class inputs although their antenna heights are contained in the ship class inputs. The radars are specified on the basis of their detection capabilities, beam widths, side-lobes, radar frequency, and delay functions. In the case of a bistatic radar, the location and antenna height of the transmitter must also be specified. A maximum of 15 radar classes may be included. Radar class 1 is input into X(901) to X(920); radar class 2 in X(921) to X(940); etc.

X( )	Inputs
901	Radar Type: 0 = Bistatic 1 = Monostatic
902	β, detection range in clear environment, $nm/m^{\frac{1}{2}}$ (see Appendix C).
903	α, detection range with standoff jamming of 1w/mhz, $nm/m^{\frac{1}{2}}$ (see Appendix C).
904	Side-lobe ratio. Number of db that average side-lobe is below main beam. Enter as a negative number. Converted by model to side-lobe coefficient for use in obtaining jamming effects.
905	Maximum instrumented range, nm.
906	Class of reaction time distribution function: Detectability-to-detection for search radars. FCS assigned to acquisition for fire control radars.

X( )	Inputs
907	Radar frequency band index. 1, 2, 3, or 4. Used with jamming to provide for frequency differences in various radars.
908	Class of reaction time distribution function for fire control radars: Detectability to detection delay in horizon/sector scan mode. Blank for search radars.
909	Max Director assignment range for FCR, nm.
910	Spare
911	Radar beam width in degrees. (Converted to one-half beam width in radians by the model.)
912	Spare
913	Probability radar is fully operable. (Used when X(2) set.)
914-916	Spares
917	Ship number of bistatic transmitter. Used only with bistatic radars. If transmitter is not located on a ship, a dummy ship must be input at the appropriate geographical location of the transmitter.
918	Bistatic transmitter antenna height in feet. Used only with bistatic radars.
919-920	Spares
921-940	Radar Class 2
.	.
.	.
.	.
1181-1200	Radar Class 15

5. Ship Position Inputs

BLUE ship positions are input by specifying the class of a ship that is at a given location. The position may be entered as the range and bearing from force center or as X and Y coordinates in nautical miles. The form must be

the same for all ships and is dependent on the value of X(78). The left and right firing bearing for the ships must also be specified if firing sectors are used. If a mix of some ships with firing sectors and others without is used, X(8) should be set to 1 and appropriate bearings entered for all ships. Ships which are not limited should be assigned a left bearing of 000 and a right bearing of 359.99. A maximum of 40 ships may be used. Ship 1 is input into X(1201) to X(1205); ship 2 in X(1206) to X(1210); etc.

X( )	Inputs
1201	Ship class number.
1202	If X(78) = 0, circle in nautical miles = 1, X coordinate in nm
1203	If X(78) = 0, bearing from forces center in degrees true = 1, Y coordinate in nm
1204	Left sector limit, degrees true, use if X(8) = 1.
1205	Right sector limit, degrees true, use if X(8)= 1. In assigning sector limits to ships with no limitations, put 000 in X(1204) and 359.99 in X(1205).
1206-1210	Ship number 2
.	.
.	.
.	.
1396-1400	Ship number 40

6. Reaction Time Distribution Function Inputs

These inputs provide appropriate delays, in minutes, for processing events during the simulation. A detailed description of the functions together with an example is contained in Appendix B under subroutine REACT. There are

seven events which may be specified by up to five different classes of delay functions for each event. The reaction functions 2, 3, 5, and 7 are related to ship class inputs. Functions 1, 4, and 6 are used with radar class inputs.

#1: Detactability to Detection (5 classes)

X( )	Inputs
1401	Delay in minutes for 0.1 cumulative probability.
1402	Delay in minutes for 0.2 cumulative probability.
1403	Delay in minutes for 0.3 cumulative probability.
1404	Delay in minutes for 0.4 cumulative probability.
1405	Delay in minutes for 0.5 cumulative probability.
1406	Delay in minutes for 0.6 cumulative probability.
1407	Delay in minutes for 0.7 cumulative probability.
1408	Delay in minutes for 0.8 cumulative probability.
1409	Delay in minutes for 0.9 cumulative probability.
1410	Delay in minutes for 1.0 cumulative probability.
1411-1420	Class 2 of RTDF1
.	.
.	.
.	.
1441-1450	Class 5 of RTDF1

#2: Detection to ID and Designation to Weapons Direction Equipment (5 classes)

1451-1460	Class 1 of RTDF2
.	.
.	.
.	.
1491-1500	Class 5 of RTDF2

X( )	Inputs
#3:	ID and Designation to Fire Control System (FCS) Assignment (5 classes)
1501-1510	Class 1 of RTDF3
.	.
.	.
.	.
1541-1550	Class 5 of RTDF3
#4:	FCS Assigned to Acquisition (5 classes)
1551-1560	Class 1 of RTDF4
.	.
.	.
.	.
1591-1600	Class 5 of RTDF4
#5:	FCS Acquire to Fire (5 classes)
1601-1610	Class 1 of RTDF5
.	.
.	.
.	.
1641-1650	Class 5 of RTDF5
#6:	Detectability to Detection for Fire Control Radar in Horizon/Sector Scan Mode (5 classes)
1651-1660	Class 1 of RTDF6
.	.
.	.
.	.
1691-1700	Class 5 of RTDF6
#7	Decisional Delay for Horizon/Sector Scan Mode (5 classes)
1701-1710	Class 1 of RTDF7
.	.
.	.
.	.
1741-1750	Class 5 of RTDF7
1751-1800	Spares

7. Raid Class Inputs

Raid classes are used to specify the launch platforms for the RED ASM/SSM and standoff jammers. The launch platforms can be aircraft or surface units. Submarines are required to be on the surface for launch and are input as a surface unit. The raid classes contain the offensive capability of the RED units in terms of weapons carried, radar class, jamming ability, and speed. The inputs also specify the radar cross section of the raider and the number of hits to put it out of action. A maximum of 10 classes may be input. Raid class 1 is contained in X(1801) to X(1820); raid class 2 in X(1821) to X(1840); etc.

X( )	Inputs
1801	Spare
1802	Number of hits to put out of action, always 1 for aircraft. Must be whole number.
1803	Number of first type weapons carried.
1804	Class of first type weapon.
1805	Number of second type weapons carried.
1806	Class of second type of weapon.
1807	Radar class number.
1808	Salvo switch (surface and subsurface only) 0 = 1 missile per salvo 1 = 2 missiles per salvo
1809	Launch cycle time for salvo, minutes for surface/subunits only.
1810	Jamming power, w/MHz on band 1.
1811	Jamming power, w/MHz on band 2.
1812	Jamming power, w/MHz on band 3.

X( )	Inputs
1813	Jamming power, w/MHz on band 4.
1814	Raid speed, knots (converted by the model to nm/min).
1815	Raid radar cross section, meters squared.
1816	Fourth root of raid radar cross section (computed by the model).
1817	Jammer beam width in degrees. (Converted to one-half beam width in radians by the model.)
1818	Jammer side-lobe ratio. Number of db that average side-lobe is below main beam. Enter as negative number. (Converted by model to side-lobe coefficient for use in obtaining jamming effects.)
1819-1820	Spares
1821-1840	Raid Class 2
.	.
.	.
.	.
1981-2000	Raid Class 10

8. Raid Weapon Class Inputs

The raid weapon classes specify the types of ASM/SSM by their flight characteristics, reliabilities, and radar cross section. A total of 20 different classes may be used. Weapon class 1 is input in X(2001) to X(2010); weapon class 2 in X(2011) to X(2020); etc.

X( )	Inputs
2001	ASM/SSM speed, knots (converted by the model to nm/min).
2002	ASM/SSM flight altitude, ft.

X( )	Inputs
2003	Launch range[4]
2004	Reliability (launch and initial guidance).
2005	Inflight reliability (including homing and fusing).
2006	Radar cross-section, meters squared.
2007-2008	Spares
2009	Weapon horizon, nm (computed by model).
2010	Fourth root of radar cross section (computed by model).
2011-2020	Weapon Class 2
.	.
.	.
.	.
2192-2200	Weapon Class 20

9. Raid Position Inputs

These inputs contain the location, course, and altitude of an individual launch platform. If the raider has a jammer, the jammer target is also specified. A maximum of 90 raiders is permitted. Raid 1 is input in X(2201) to X(2206); raid 2 in X(2207) to X(2212); etc.

X( )	Inputs
2201	Raid class number (1, 2, etc.)
2202	Range, nm (from (0,0)).
2203	Bearing, degrees true (from (0,0)).

---

[4] Launch Range is measured from the position (0,0). The user must consider this fact when dealing with dispersed formations and multiple target list. The distance of the intended launch point from force center is the proper input, not the range of the launch point from the primary target.

X( )	Inputs
2204	Altitude, feet (less than 50 feet for surface/ subsurface units).
2205	Course, degrees true.
2206	Jammer's jamming target: 0 = no jam capability N = ship number of jamming target 999 = force center (0,0)
2207-2212	Raid number 2
.	.
.	.
.	.
2734-2740	Raid number 90

B. THE TARGET LIST ASSIGNMENTS

The target list assignments are read in immediately following the zero card indicating the end of the X vector inputs. Each RED raider must be assigned a target list, even if the raider carries no weapons. If the raider carries no weapons the target list information is not used, but the input logic requires that some number be entered. There must be a total of three cards which are to be dimensioned 30I2. The first thirty raiders are entered with card one, the second thirty raiders on two, etc. For example, if the simulation has four RED aircraft, column 2 card 1 would contain the number of the target list for raider number 1; column 4 card 1 that for aircraft number 2; etc. Columns 9 through 60 on the first card, and 1 through 60 on the next two cards could contain any integers desired, usually 00.

C. THE TARGET LISTS

The cards that input the target lists follow immediately after the three cards described above. The values to be read in are stored in the array PLIST. This is a (40,6) array in which the columns are the target lists. The array allows for up to 40 individual targets and six separate listings of targets. The target list is identified by its column number, and it is this number that was entered in NRTGT. The PLIST cards follow immediately after the three input cards for NRTGT mentioned above. The format for the PLIST cards is 10F8.4. The column entries are the total probability of targeting all ships up to and including the target ship in the given row. As an example, in the case of three target lists and five BLUE ships the PLIST array could appear as follows:

	1	2	3	4	5	6
1	.12	0.0	.25	0	0	0
2	.42	.25	.50	0	0	0
3	.62	.50	.50	0	0	0
4	.82	.75	1.0	0	0	0
5	1.0	1.0	1.0	0	0	0
6	0	0	0	0	0	0

The raiders that have been assigned target list 1 would have a probability of .12 of firing a missile at ship 1 each time it fires. It would have a probability of .3 of firing at ship 2, and .20 of firing at ship 3. A raider firing a missile with target list 2 would have a probability of 0.0 of firing at ship 1 and .25 of firing at any of the other four. A raider with list 3 would have a .25 probability of

firing at ship 1, a .25 probability of firing at ship 2, and a .50 probability of firing at ship 4.

The input cards for the PLIST array would be structured as follows. Card one would contain .12 in columns 1-8, .42 in 9-16, .62 in 17-24, .82 in 25-32, 1.0 in 33-40, and zeros throughout the remainder. Cards two, three, and four would contain zeros. Since card five starts list two, it is filled out in the same manner as card one, with, in this case, cards 6, 7, and 8 containing zeros. Since there are only five ships and three lists, the last card required for this run would be card 9. If the full 40 ships and six target lists were being used, a total of twenty-four cards would be required to input the target lists.

V. MODEL OUTPUT

The model produces three types of output. The types are standard, special displays, and debugging. For debugging, there are a number of switches in the control section of the inputs which may be set to provide intermediary values during the iteration. The special displays provide one time outputs of geographical plots, histograms, and selected tables. The standard output occurs at the end of each iteration and contains an abbreviated battle history along with statistical data concerning the engagement.

A. STANDARD OUTPUT

The standard output consists of the ASM table, the SAM-ASM-RAID table, and statistics regarding hits, detections, and opportunities to engage. The contents of the tables are given below. The actual format of this output is provided in the tables of representative output at the end of this section.

1. The ASM Table

Column 1 — ASM/SSM number. This number identifies the ASM/SSM by target number. The order in which the missiles were launched or destroyed can be determined by subtracting 90 from the target number.

Column 2 — Status code. This code indicates the disposition of the missile. The code is as follows:

0 missile failed on launch.

1 missile was in flight.

2 missile was destroyed by SAM. The SAM-ASM-RAID table will indicate which missile ship was responsible.

3 missile failed in flight.

4 missile flight successful, missile impacted on target ship.

5 missile destroyed when raider destroyed.

Column 3 Time of impact. This will be the time the missile impacted, or would have impacted, unless it was lost when raider was destroyed. In this case it will be 0.0.

Column 4 Launch raid.

Column 5 Target ship. Target ship number unless the missile was lost with raider destruction.

Column 6 Launch time. The time of missile launch unless lost when raider destroyed. In this case it is the time of loss.

Column 7 Weapon class. If lost with raid, 0.0.

Column 8 X coordinate at launch.

Column 9 Y coordinate at launch.

2. <u>The SAM-ASM-RAID Table</u>

Column 1 Target number. Targets 1 through 90 are raiders, 91 through 190 are missiles.

Column 2 Launcher. The launcher code AAB, where AA is the ship number and B is the launcher.

Column 3 Channel. The director-channel code AB where A is the director number and B is the channel number.

Column 4 Time of Launch. Time of BLUE missile launch.

Column 5 Kill code. The code is as follows: a negative number, an overkill, the number indicating the number of missiles in the salvo; a zero, a missile failure or system support failure; a positive number, a kill, the value indicating the number of missiles in the salvo.

Column 6 Time channel free. The game time that this launcher-channel combination will be available for re-engagement or reassignment.

3. Statistical Output

The output form of the statistics for each iteration are underlined. The definitions of the terms used follow each form.

The ships took "H" hits: The total number of hits (H) sustained by the BLUE force.

Of "X" ASMs fired, "Y" ("Z" percent) failed at launch.

X: total number of ASM/SSM that survived to launch.

Y: of X, the number that failed on launch.

Z: Y/X.

Of launched ASM, "W" ("P" percent) were killed by SAMs, "Q" ("R" percent) failed in flight.

Launched ASM; from above, X-Y.

W: the number of RED SSM/ASM shot down.

P: W/Launched ASM.

Q: the number of RED SSM/ASM that failed in flight.

R: Q/Launched ASM.

Of total ASMs available "I" ("J" percent) were destroyed when mother killed by SAMs.

Total ASMs: All SSM/ASM carried by engageable raiders or raiders that reach a launch position.

I: The number of RED ASM/SSM lost when raider destroyed.

J: I/Total available.

"D" SAMs were launched of which "E" ("F" percent) failed and "G" ("H" percent) were overkills.

D: Total BLUE SSM/SAM launched.

E: Number of SAM/SSM evaluated as failure at intercept or system support.

F: E/D.

G: Overkills. Overkill occurs when the target does not survive to the SAM intercept point due to being shot down. If the target is a missile and the RED missile reaches its target prior to intercept, this intercept is scored as a failure and not an overkill.

H: G/D.

The ship statistics for each iteration are printed in a table labeled SHIP STATS. The format is as follows:

Column 1 Ship number.

Column 2 Chances to engage. The number of RED targets that will enter this ship's missile envelope.

Column 3 Number of engagements. The number of actual missile intercept evaluations.

Column 4 Number of targets killed.

Column 5 Bombers detectable. Of those targets that will enter the ship's missile envelope, the number that are raiders.

Column 6 Bombers detected. Of the raiders detectable, those detected.

Column 7 ASM's detectable. Those ASM/SSM that will enter the ship's missile envelope.

Column 8 ASM's detected.

The following additional statistics are provided with the SHIP STATS table:

Average detection range: The average detection range of each ship on those targets that entered its missile envelope.

"X" ASMs were fired on by the fleet. X is the number of ASM/SSM that were engaged by one or more BLUE units.

"X" Bombers were fired on by the fleet. X is the number of RED raiders that were engaged by one or more BLUE units.

ASMs fired at. The number of ASM/SSM that were first engaged by each ship.

Bombers fired at. The number of RED raiders that were first engaged by each ship.

A table of the frequency of hits received by ships is printed at the end of the run. This table displays a distribution of the hit frequency over all ships for all runs. It constitutes a histogram of hit frequencies for all BLUE units.

B. SPECIAL DISPLAY SWITCHES

The below listed switches will provide the indicated output if set to one or the specified value. Setting X(2) to 1 will provide the equipment availability status for each iteration. The remaining switches provide a one-time output for a run.

X(2) Indicates equipment availability check desired. Prints out each ship number and equipment status.

X(24) Plot of BLUE unit positions using DISPL if set to 1.

Plot of raider tracks and launch points using RAIDIS if set to 2.

Both plots are provided if set to 3.

X(51) Prints the SHIP table at the end of the first iteration. The table is structured as follows: (see Table IV)

Column 1 Ship number.

Column 2	Ship class number.	
Column 3	Ship's X coordinate.	
Column 4	Ship's Y coordinate.	
Column 5	Hits to sink. The number of RED SSM/ASM missile hits to put out of action.	
Column 6	Class address. The ship class starting index in the X array.	
Column 7	Up/down. 0 = sunk 1 = operating normally.	

X(52) Prints the RAID table at the end of the first iteration in the following form: (see Table IV)

Column 1	RAIDER	The raid number, 1-90.
Column 2	CLASS	The class starting index in the X array.
Column 3	TIMER (R)	The time the raider was at the position indicated in columns 4 and 5.
Column 4	XR	The raider's X position at the time specified in column 3.
Column 5	YR	The raider's Y position at the time specified in column 3.
Column 6	XEOG	The X position of the raider at the time specified as the end of the game.
Column 7	YEOG	The Y position at the end of the game.
Column 8	XL	The raider X position at its first launch time. If the raider has no missiles, or cannot reach a launch position, the end of game position will also be indicated here.
Column 9	YL	The raider Y position at first launch.

Column 10	ALT	The raider altitude in feet. If 50 feet or below, the raider is considered a surface or subsurface unit.
Column 11	HORZ	The raider radar horizon in miles.
Column 12	WEP-CLASS	The class number of the raider's longest range ASM/SSM.
Column 13	WEPS	The total number of weapons carried by the raider.
Column 14	UP/DOWN	The status of the raider at the end of the game. 0 = shot down or sunk 1 = operating normally.

X(53) Prints the SAM table at the end of the first iteration. The table is organized as follows: (see Table V)

Column 1	SAM	The salvo number.
Column 2	SAM CLASS INDEX	The SAM type class number for the SAM/SSM in this BLUE miss-le salvo.
Column 3	SHIP NUMBER	The firing ship.
Column 4	SAM LEFT	The number of SAM/SSM remaining in the battery's magazine after this salvo was fired.
Column 5	CH1 FREE	This is the time that channel 1 for the fire control radar used in this intercept will be free for reengagement or reassignment. If the missile requires director illumination throughout missile flight, or the missile is totally independent of the director, this is an indication of system availability.

Columns 6-10		Channel availability times for systems employing a time sharing system of homing.
Column 11	SALVO	The number of missiles in the salvo.

X(85) If the number of iterations is between 10 and 100, prints a histogram of hit frequency and ASM/SSM arrival interval for the BLUE ship who's number has been entered.

C. THE DEBUGGING SWITCHES

The debugging switches, X-3,9,12,15,54,55,56,57,58,59, 60, are set by assigning a value of 1 to the specified location in the X-array. These switches are subroutine related and will provide the indicated output whenever the program cycles through the appropriate portion of the subroutine. Reference should be made to the description of the subroutines in Appendix B when using the debugging switches.

In the below format, the debug switch which must be set to 1 is listed and is followed by the subroutine name with its associated output. The output is in the form as it appears in the computer printout. The parenthesis indicate the location of the intermediary values.

X-3

SUBE1 Detection range by ship ( ) on raider ( ) with radar ( ) in clear, jamming, and horizon limited environments.

Reaction time delays for raid ( ) by ship ( ).

Bistatic radar gain and lose contact times on raider ( ) by ship ( ).

SUBE2 ASM/SSM failed on launch.

Detection range by ship ( ) on ASM/SSM ( ) with search radar ( ) in clear, jamming, and horizon limited environments.

No detection due to reaction time delay.

Detection range on ASM/SSM by ship ( ) on missile ( ) with system ( ) fire control radar ( ) in clear, jamming, and horizon limited cases. Jamming range of 999. indicates no jamming.

Earliest search radar detection time and earliest fire control radar detection time ( ) ( ).

When missile launch time will exceed the launch interval set by the user, the launch does not take place. Output is: bomber ( ) could not fire as launch time was ( ) and all ASM's had to be launched by ( ).

X-9

SUBE3 EV 3 Ship ( ) target ( ) reason ( ).

EV 3 SAM ( ) battery ( ) channel ( ).

PRIOR PRIOR debug data input channel ( ) output channel ( ) battery number ( ) input target ( ) output target ( ).

TIMMY Intercept data: target, firing ship, firing ship position, intercept position, number of firing zones, left and right bearings of the firing zones, launch bearing of SAM.

Target, firing ship, indicator flag, reason.

X-12

MAIN Original X array as read into the computer.

The target lists as read into the computer.

The target list assignments as read into the computer.

X-15

MAIN	The X-array as converted by the program.

X-54

MAIN	Prints out all random numbers used in the first iteration, in the order they were used.

X-55 Provides the computer time that the program enters and leaves the specified subroutine.

EFFECT	Login, logout
JULIE	Login, logout
EVENT	Login, logout
GETEV	Login, logout
STOREV	Login, logout
REMOVE	Login, logout
TMDASM	Login, logout
TMDBOM	Login, logout
SUBE1	Login, logout
SUBE2	Login, logout
SUBE3	Login, logout
SUBE4	Login, logout
SUBE5	Login, logout
SUBE6	Login, logout
LOCTIM	Login, logout
PRIOR	Login, logout
TIMMY	Login, logout
RCHECK	Login, logout
JAMM	Login, logout
BIDET	Login, logout

X-56

SUBE4	EV 4 ASM ( ) ship ( ) ASM status ( ) hits on ship ( ) time ( ).
SUBE5	EV 5 target number ( ) firing ship ( ) launcher ( ) salvo number ( ) time ( ).

X-57

SUBE1	Detection ranges by ship ( ) on raider ( ) with system ( ) fire control radar ( ) in clear, jamming, and horizon limited cases.
	Raider ( ) surface or submarine unit, ship ( ) has no SSM.
	Detection ranges on raider ( ) by ship ( ) in horizon search mode.
SUBE2	Reaction time delays.
	Detection ranges in horizon search mode of ASM/SSM ( ) by ship ( ) in clear, jamming, and horizon limited cases.
TMDASM	No engagement possible, raid ( ) target ship ( ) battery ( ) missile ship ( ) ASM/SSM ( ).
	No engagement possible, raider ( ) ship ( ).
	Engagement data - time entering and time leaving a given range from ship ( ).

X-58

JULIE	Number of SAM/SSM in each battery.
SUBE2	Prints the NONO table after each raider's launch cycle.
	NONO table codes are contained in the description of SUBE2 in Appendix B.
SUBE5	For salvo ( ) results ( ).

X-59

TIMMY	Crossing target data: target, firing ship, angle between target flight path and line of sight at launch, angle between SAM/SSM flight path and target flight path, angle between SAM/SSM flight path and line of sight at launch, target position at launch, GSINE, range to target at launch.

X-60

JULIE	The events list at start of the iteration.
EVENT	The event retrieved.
STOREV	The event stored.
	The event lost.

THE NUMBER OF SAMS LEFT AT THE END OF ITERATION 1 WAS 126

ASM TABLE
ITERATION 1

ASM NUMBER	STATUS CODE	TIME OF IMPACT	LAUNCH RAID	TARGET SHIP	LAUNCH TIME	WEAPON CLASS	X LAUNCH	Y LAUNCH
91	5.	0.0	2.	0.	15.36	0.	0.0	0.0
92	5.	0.0	2.	0.	15.36	0.	0.0	0.0
93	5.	0.0	2.	0.	15.36	0.	0.0	0.0
94	5.	0.0	2.	0.	15.36	0.	0.0	0.0
95	5.	0.0	1.	0.	16.37	0.	0.0	0.0
96	5.	0.0	1.	0.	16.37	0.	0.0	0.0
97	5.	0.0	1.	0.	16.37	0.	0.0	0.0
98	5.	0.0	1.	0.	16.37	0.	0.0	0.0
99	5.	0.0	5.	0.	17.13	0.	0.0	0.0
100	5.	0.0	5.	0.	17.13	0.	0.0	0.0
101	5.	0.0	5.	0.	17.13	0.	0.0	0.0
102	5.	0.0	5.	0.	17.13	0.	0.0	0.0
103	2.	26.694	3.	1.	20.46	1.	11.58	-99.33
104	0.	27.542	3.	1.	21.86	1.	- 2.59	-91.15
105	2.	27.706	3.	1.	22.11	1.	- 5.11	-89.69
106	2.	28.392	3.	1.	23.07	1.	-14.91	-84.04

THE ORIGINAL ATTACK CONSISTED OF 5 RAIDERS. 1 WERE LEFT AFTER THE ATTACK.

TABLE I

REPRESENTATIVE OUTPUT FOR ASM TABLE

SAM-ASM-RAID TABLE

TGT NO.	LAUNCHER	CHANNEL	TIME OF LAUNCH	KILL CODE	TIME CHANNEL FREE
2.	21.	17.	10.61	1.	15.36
1.	21.	17.	12.15	1.	16.37
5.	31.	17.	12.37	1.	17.13
3.	51.	17.	13.90	0.	18.15
3.	41.	17.	18.15	0.	21.61
103.	41.	17.	21.09	1.	22.85
3.	41.	17.	21.61	1.	23.92
105.	41.	17.	22.76	1.	24.69
106.	41.	17.	23.18	0.	25.49
106.	21.	17.	25.49	1.	27.03

```
STATISTICS FOR ITERATION NO. 1

    THE SHIPS TOOK  0 HITS.
    OF  4 ASMS FIRED, 1 (25.00 PERCENT) FAILED AT LAUNCH,
    OF LAUNCHED ASM, 3 (100.00 PERCENT) WERE KILLED BY SAMS,
                     0 (0.0 PERCENT) FAILED IN FLIGHT.
    OF THE TOTAL ASMS AVAIL,  12 (75.00 PERCENT) WERE DESTROYED WHEN
                                         MOTHER KILLED BY SAMS.
    10 SAMS WERE LAUNCHED OF WHICH  3 (30.00 PERCENT) FAILED AND
                                    0 ( 0.0  PERCENT) WERE OVERKILLS.
    334 RANDOM NUMBERS WERE DRAWN.

SUMMARY FOR 1 ITERATIONS
    OF 4 ASMS FIRED, 1  (25.00 PERCENT) FAILED AT LAUNCH,
    OF LAUNCHED ASM, 3 (100.00 PERCENT) WERE KILLED BY SAMS,
                     0 (  0.0  PERCENT) FAILED IN FLIGHT.
    OF THE TOTAL ASMS AVAIL,  12 (75.00 PERCENT) WERE DESTROYED WHEN
                                         MOTHER KILLED BY SAMS.
    AVERAGE NUMBER OF HITS ON FLEET FOR 1 ITERATIONS IS 0.0
    10 SAMS WERE LAUNCHED OF WHICH  3 (30.00 PERCENT) FAILED AND
                                    0 ( 0.0  PERCENT) WERE OVERKILLS.
```

TABLE II

REPRESENTATIVE OUTPUT FOR SAM-ASM-RAID TABLE AND THE STATISTICS

334 RANDOM NUMBERS WERE DRAWN.

SHIP STATS FOR ITERATION 1

SHIP NO.	CHANCES TO ENGAGE	NUMBER OF ENGAGEMENTS	TARGETS KILLED	BOMBERS DETECTABLE	BOMBERS DETECTED	ASMS DETECTABLE	ASMS DETECTED
1	4	0	0	1	1	3	3
2	6	3	3	3	3	3	3
3	8	1	1	5	5	3	3
4	7	5	3	4	4	3	3
5	7	1	0	4	4	3	3

SHIP	AVE DET RANGE BOMBERS	AVE DET RANGE ASMS
1	142.96	69.38
2	86.26	61.68
3	63.20	23.85
4	77.66	49.21
5	67.44	16.02

3 ASMS WERE FIRED ON BY THE FLEET
4 BOMBERS WERE FIRED ON BY THE FLEET

SHIP	ASMS FIRED AT	BOMBERS FIRED AT
1	0	0
2	0	2
3	0	1
4	3	0
5	0	1

	HITS ON SHIP "I"
1	0
2	0
3	0
4	0
5	0

TABLE III

REPRESENTATIVE OUTPUT FOR SHIP STATS TABLE

SHIP TABLE

SHIP	CALSS	X	Y	HITS TO SINK	CLA-ADR	UP/DOWN
1	1.	0.0	0.0	5.	101.	1.
2	2.	25.00	25.00	1.	141.	1.
3	2.	-25.00	25.00	1.	141.	1.
4	2.	-25.00	-25.00	1.	141.	1.
5	2.	25.00	-25.00	1.	141.	1.

RAID TABLE

RAIDER	CLASS	TIMER(R)	XR	YR	XEOG	YEOG
1	1801.	15.34	15.7	120.9	122.5	-1100.6
2	1801.	14.86	71.3	118.3	-950.0	- 570.6
3	1801.	21.61	32.0	- 67.2	-998.4	483.8
4	1801.	26.17	-51.5	122.1	498.2	1074.2
5	1801.	16.33	-99.3	104.0	774.5	- 739.8

XL	YL	ALT	HORZ	WEP-CLASS	WEPS	UP/DOWN
17.6	98.4	10000.0	174.	1.	4.	0.
34.9	93.7	25306.0	196.	1.	4.	0.
11.6	-99.3	35000.0	230.	1.	4.	0.
498.2	1074.2	10000.0	123.	1.	4.	1.
-67.7	73.6	27000.0	202.	1.	4.	0.

TABLE IV

REPRESENTATIVE OUTPUT FOR SHIP AND RAID TABLES

SAM TABLE

SAM	SAM CLASS INDEX	SHIP NUMBER	SAM LEFT	CH1 FREE
1	1.	2.	29.	15.36
2	1.	2.	28.	16.37
3	1.	3.	29.	17.13
4	1.	5.	29.	18.15
5	1.	4.	29.	21.61
6	1.	4.	28.	22.85
7	1.	4.	27.	23.92
8	1.	4.	26.	24.69
9	1.	4.	25.	25.49
10	1.	2.	27.	27.03

CH2 FR	CH3 FR	CH4 FR	CH5 FR	CH6 FR	SALVO
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.
0.0	0.0	0.0	0.0	0.0	1.

TABLE V

REPRESENTATIVE OUTPUT FOR SAM TABLE

VI. CONCLUSIONS

The use of bistatic search radars is not a new concept. FADS provides a method of evaluating their potential usefulness to the Navy. As demonstrated in the sample problem, FADS also has many applications in weapons systems analysis and the analysis of tactics.

The model was structured along the lines of the FLOATS model in use at the Johns Hopkins University Applied Physics Laboratory. The additional capabilities introduced by FADS should provide a basis for a more realistic view of the present day and future tactical environments. The employment of decoys, dispersed formations, and passive missile traps are but a few of the ways in which FADS might be utilized.

Lastly, the model is ideal for use in an advanced war gaming course. The model is complex, but the simulation was written with ease of reading in mind. With rewriting the core requirements might be reduced, but, in its present form, FADS is easily read and understood, and it could be disected by the interested student with ease.

APPENDIX A

BISTATIC RADAR SYSTEMS

Although the bistatic radar system was described in the introduction as a radar system in which the transmitter and the receiver are not collocated it is necessary to point out that the separation between the transmitter and the receiver should be comparable to the target distance, that is, on the order of several nautical miles as opposed to several feet. The assumed bistatic system is to provide detection and location information similar to that obtained with current search radar systems. It should in essence provide the user with what would be on a plan position indicator (PPI) scope. No provision is assumed for height finding capability. The basic difference between the target geometry for monostatic and the bistatic radar systems is shown in figure A1.

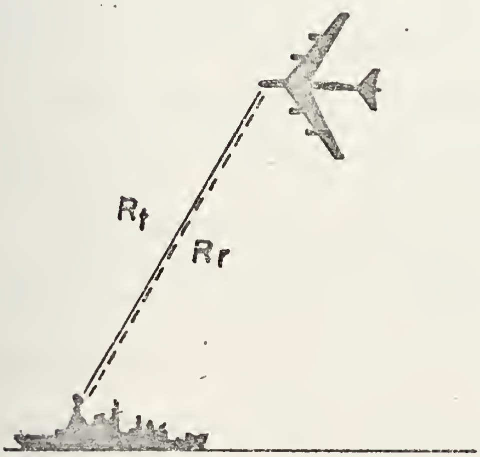


Figure A1a
Monostatic Radar Principle

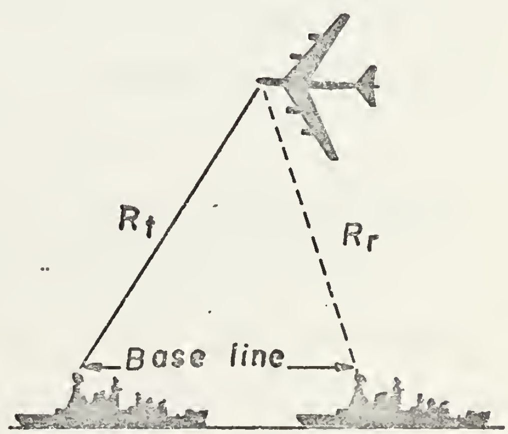


Figure A1b
Bistatic Radar Principle

The bistatic radar concept is not a recent development, although the technology required to bring the system up to the capabilities described above is. The bistatic concept predates the monostatic radar in that the development of the radar duplexer was required in order to convert the original bistatic radar to a monostatic system. Sholnik [Ref. 10, p. 585] states:

> Its (the bistatic radar) principle was known and demonstrated many years before the development of practical monostatic radar. ...the first "radar" observations in both the United States and in Great Britain were made with separated CW transmitters and receivers. These early radars were known as wave-interference equipment but were the same as what would now be called bistatic radar. Taylor and Young of the Naval Research Laboratory first demonstrated bistatic radar for the detection of ships in 1922. Their work was disclosed in a patent issued in 1934. The early experiments with wave-interference (bistatic) radar led to the development of monostatic radar in the late 1930s in both this country and abroad. Further development was put aside after the demonstration of the more versatile monostatic-radar principle. Bistatic radar lay dormant for about fifteen years until it was "reinvented" in the early 1950s and received new interest.

Current interest is conceivably motivated by considerations of electromagnetic counter-measures, defense against anti-radiation missiles, and deception roles.

One method for target location with a bistatic system requires knowledge of the distance from the transmitter to the target, (R_x), and to the receiver (R_r), the horizontal angle between the base line (a line from the receiver to the transmitter) and the target bearing, and the elevation angle from the receiver to the target. However, in the simulation problem it is not necessary to determine a target's

location since the target's position at any time is a function of the input parameters. It is only necessary to determine if the target is detectable and, given that it is, determine where along the target's track an individual radar is able to detect the target.

Detection occurs when the reflected power at the receiver is greater than the detection threshold of the receiver for a given probability of detection. The radar range equation for the bistatic radar is identical to the well known monostatic range equation except that the power received varies inversely with the product of $R_x^2 R_r^2$. Additionally the two-way propagation losses differ since the path of the reflected signal is not the same as that of the transmitted signal. The range equation can be readily written in terms of the constants for the radar system, the variables R_x, R_r and the target's radar cross section.

For comparison purposes both the monostatic and the bistatic detection range equations are given below:

$$\text{Monostatic} \quad P_r = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 L_p^2 L_s R^4}$$

$$\text{Bistatic} \quad P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 L_p(t) L_p(r) L_s R_r^2 R_x^2}$$

where:

R = range to target (monostatic)

R_x = range from transmitter to target (bistatic)

R_r = range from receiver to target (bistatic)

P_t = power of transmitter

P_r = power received at the receiver

G = gain of monostatic antenna

G_t = gain of bistatic transmitter antenna

G_r = gain of bistatic receiver antenna

λ = radar wavelength

σ = target's radar cross section (see below for discussion of bistatic and monostatic approximation)

L_p = one way propagation losses for monostatic

$L_p(t)$ = propagation losses from transmitter to target (bistatic)

$L_p(r)$ = propagation losses from target to receiver (bistatic)

L_s = system losses

By defining P_r as the minimum power required by the receiver in order to obtain a given probability of detection with typical propagation and system losses and by grouping constants of the radar system, the above equations can be simplified to read: $R = \beta\sigma^{\frac{1}{4}}$ and $R_x^2 R_r^2 = \beta^4\sigma$ where β is the range at which a particular radar system can detect a one square meter target in a non-jamming environment. Values of β for current U.S. Navy radars are listed in Ref. 9.

When standoff electronic noise jamming is present, the range equations are defined in a similar manner:

$$\text{Monostatic:} \quad R_{bt}^4 = \frac{P_t G \sigma}{4\pi B_r (S/J)\Sigma JP}$$

$$\text{Bistatic:} \quad R_{bt}^2 R_x^2 = \frac{P_t G_t^{\frac{1}{2}} G_r^{\frac{1}{2}} \sigma}{4\pi B_r (S/J)\Sigma JP}$$

where:

R_{bt} = burn through range monostatic

B_r = receiver band width

S/J = signal-to-jamming ratio (includes propagation losses)

ΣJP = sum of jamming power arriving at the receiver

The jamming level from an individual jammer is obtained by multiplying the power of the jammer by the appropriate coefficient for the geometry of the radar and jammer main to side-lobe ratios and dividing by the square of the distances between the jammer and the radar. The individual jamming levels are then added to obtain the sum of the jamming power arriving at the radar.

As in the case with the non-jamming environment, the above equations can be simplified by defining a term α which contains the constants of the radar system and which is equal to the range at which a one square meter target can be detected in the presence of one watt of jamming power per megahertz. The monostatic range equation in jamming is:

$$R_{bt} = \alpha^{\frac{1}{2}}(\sigma/\Sigma JP)^{\frac{1}{4}}$$

and the bistatic equation is

$$R_x R_r = \alpha(\sigma/\Sigma JP)^{\frac{1}{2}} .$$

In solving the bistatic detection problem in the simulation algorithm, R_x and R_r have only the target's position in common. It is therefore necessary to state the product $R_x R_r$ as a function of the target's position and solve the

detection equation in terms of the time at which the target can be detected. If a constant CHAR is defined to be the product of β^4 times σ for a clear environment and α^2 times σ divided by the sum of the jamming power for the jamming situation, the detection equation then becomes $R_x^2 R_r^2 = \text{CHAR}$. The radar is able to detect a target whenever the range product squared is less than or equal to the constant CHAR. By expressing R_x and R_r in terms of time, the solution of the bistatic equation $R_x^2(t)R_r^2(t) - \text{CHAR} = 0$ provides the times at which detection can occur and then be lost as the target moves along its track relative to the transmitter and the receiver. The below figure is a layout of the track and detection points that is illustrative of the detection equation:

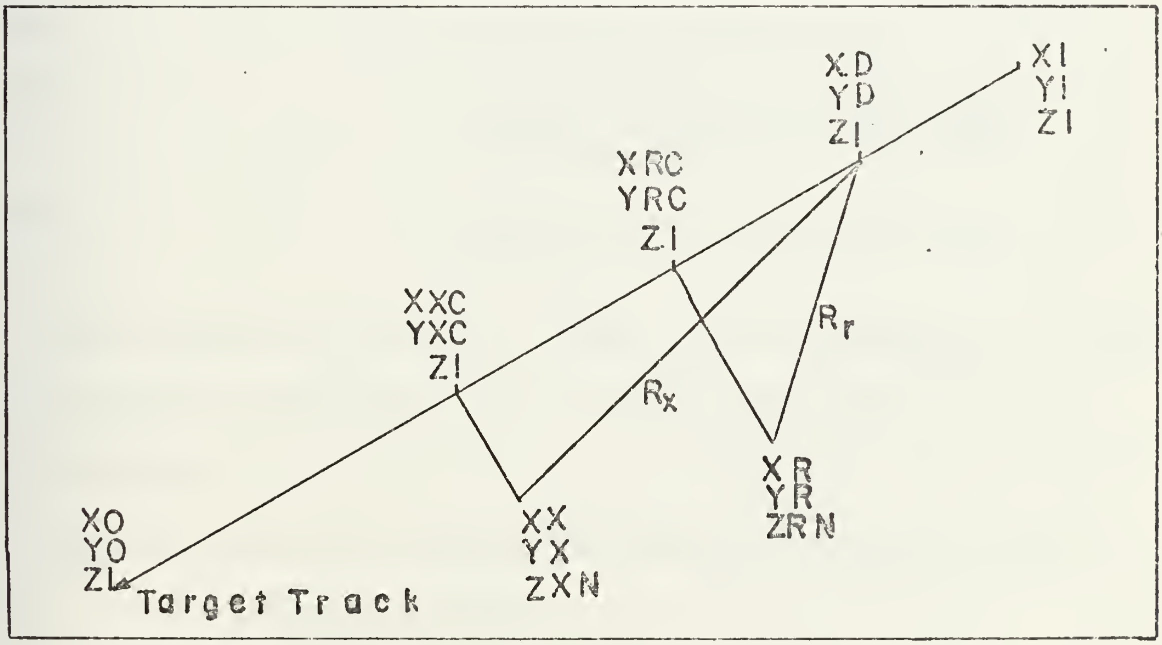


Figure A2
Bistatic Detection Geometry

Labels are as follows:

Receiver	Transmitter	Value
XR	XX	X Coord of position
YR	YX	Y Coord of position
ZRN	ZXN	antenna height in nautical miles
XRC	XXC	X Coord of track CPA
YRC	YXC	Y Coord of track CPA
CPR	CPX	Distance to CPA

Track

X1,Y1	initial position
Z1	altitude in nautical miles
XO,YO	terminal position
V	velocity in nm/min
Vx	velocity in X direction
Vy	velocity in Y direction
XD	X Coord of detection position = (X1+VxT) where T is the time of detection
YD	Y Coord of detection position = (Y1+VyT)

The equation $R_x^2(t)R_r^2(t) - CHAR = 0$ is expressed in terms of the stated positions; time is the independent variable.

For example:

$$R_r^2(t) = (XR-XRC)^2+(YR-YRC)^2+(ZRN-Z1)^2+(XRC-(X1+VxT))^2 +(YRC-(Y1+VyT))^2$$

$$= (Vx^2+Vy^2)T^2-2(Vx(XRC-X1)+Vy(YRC-Y1))T+(CPR)^2 +(XRC-X1)^2+(YRC-Y1)^2$$

$$= AT^2+DT+E$$

In a similar manner $R_x^2(t) = AT^2+BT+C$ where $B = -2(Vx(XXC-X1)+Vy(YXC-Y1))T$ and $C = (CPX)^2+(XXC-X1)^2+(YXC-Y1)^2$.

The range equation in terms of t then reduces to

$$T^4+\left(\frac{B+D}{A}\right)T^3+\left(\frac{C+E}{A}+\frac{BD}{A^2}\right)T^2+\left(\frac{BE+CD}{A^2}\right)T+\frac{CE-CHAR}{A^2}=0$$

The real roots of this fourth order polynomial in ascending order are the times at which the target is first detectable, lost, redetected and then lost again. In the case where there are no real roots, the target is not detectable along its track. If only one pair of real roots exists, the radar has only one detection zone against the target.

The multiple detection zones occur when the target penetrates the contour of the equation $R_x R_r = CHAR^{\frac{1}{2}}$ in more than two locations. These contours plot as ovals of Cassini in the plane containing the base line between the transmitter and the receiver. The contours, shown in Figure A3, represent curves of constant sensitivity for the bistatic radar system and a given target. In the case of low sensitivity, the curves consist of two disjunct ellipses with the transmitter and receiver at the furthest foci. As sensitivity is increased the curves become ovals about the receiver and the transmitter and eventually become an ellipse with the transmitter and receiver at the foci. If this ellipse is rotated about its major axis, a prolate spheroid is generated which defines the detection surface for the bistatic radar.

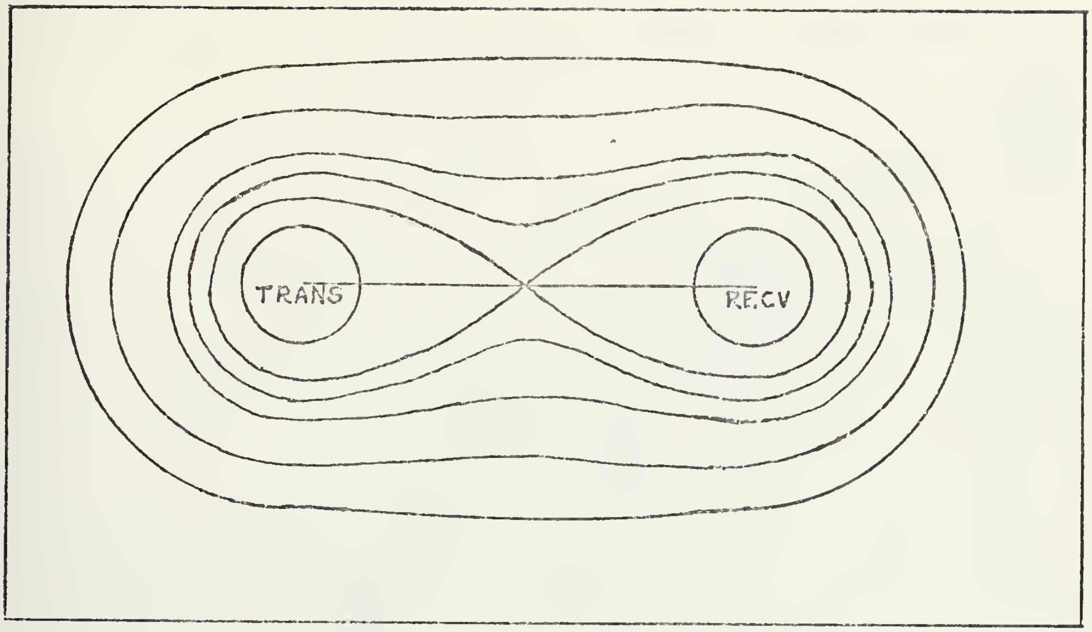


Figure A3

Curves of Constant Sensitivity for Bistatic Radar System

In addition to the requirement that a solution to the radar range equation exists, the target must also be above both the transmitter and the receiver radar horizon for a detection to occur. The radar horizon restrictions can be seen in figure A4, which uses the same track as presented in figure A2, along with an additional axis representing the aircraft position with respect to time.

The positions in figure A4 are defined in figure A2. HORX and HORR are the distances to the transmitter and receiver radar horizons for their respective antenna heights and the target's altitude. TR1 is the time the target first enters the radar horizon of the receiver; TR2 is the time it departs. TX1 and TX2 are similar times for the transmitter. The time that the target is simultaneously above both horizons is the interval (T_1^0, T_2^0) where T_1^0 = max (TR1,TX1) and T_2^0 = min (TR2,TX2).

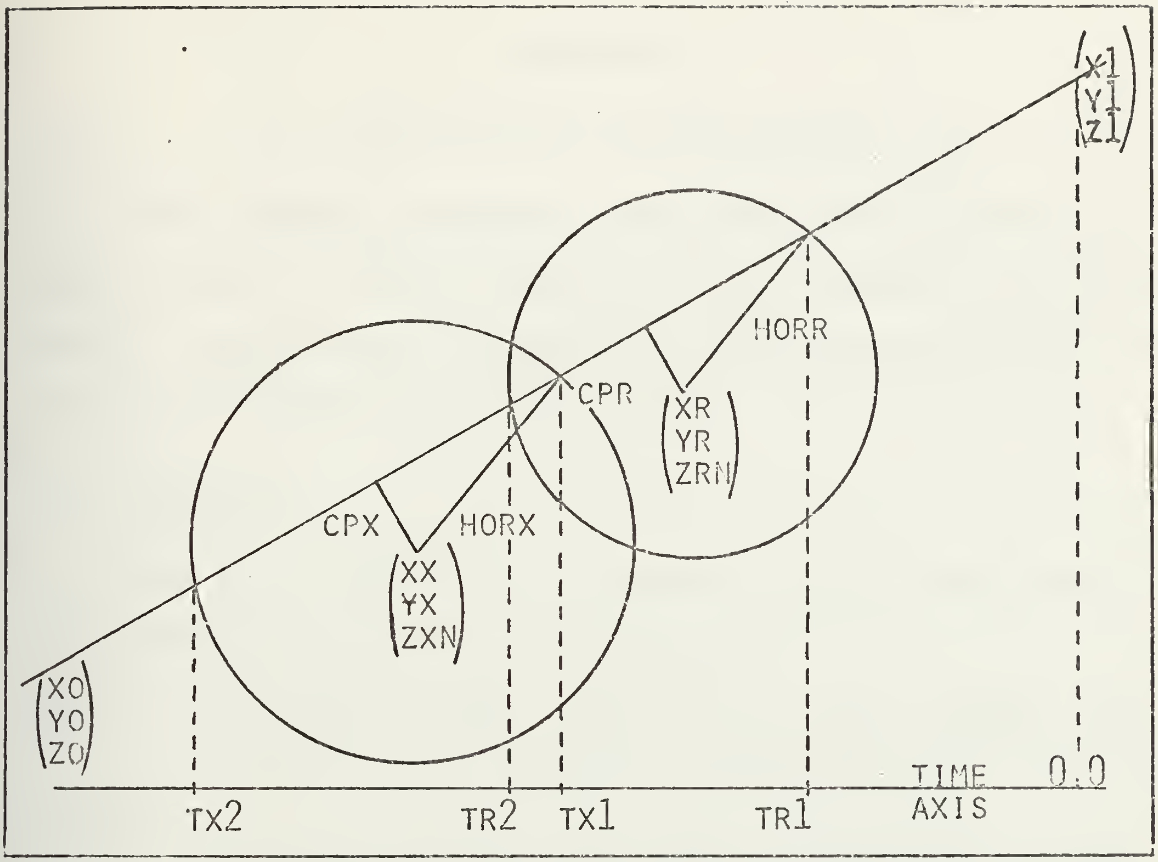


Figure A4
Intersection of the Bistatic Radar Horizons

The intersection of the joint horizon time interval (T_1^0, T_2^0) with the time intervals obtained from the range detection equation provides the time of detection and the time of lost detection for a target.

These times, in conjunction with the ranges which are related to the times, are then used in the simulation to determine "shoot events" for the defensive forces and to accumulate statistics on the detection ranges.

APPENDIX B

DESCRIPTION OF THE PROGRAM

This appendix provides a description of the FADS computer program. The MAIN portion of the program is presented first. The principal subroutines follow with the minor subroutines and functions covered last.

1. OVERVIEW

This paragraph gives an overview of the functions of the individual sections of the program.

MAIN - The MAIN program reads in the input values, compiles statistics and controls the simulation through the specified number of iterations.

EFFECT - Subroutine EFFECT controls the individual iteration by selecting the proper subroutine to play the next event in the events list.

JULIE - Subroutine JULIE initializes or resets values for each iteration.

EVENT - Subroutine EVENT controls the input and retrieval of events from the events list.

REMOVE - Subroutine REMOVE retrieves higher priority targets from the events list.

TARGET - Subroutine TARGET returns all of the "shoot" events which occur at a specified time.

SUBE1 - Subroutine SUBE1 computes detections of the raiders by the BLUE forces and determines "shoot" events against the raiders.

SUBE2 - Subroutine SUBE2 computes the RED ASM/SSM launches and subsequent detections followed by the "shoot" events against the ASM/SSMs.

SUBE3 - Subroutine SUBE3 evaluates the engageability of a target based on the situation at the proposed BLUE missile launch time.

SUBE4 - Subroutine SUBE4 determines ASM/SSM impacts on BLUE ships and evaluates out of action status for them.

SUBE5 - Subroutine SUBE5 evaluates the BLUE missile intercepts.

SUBE6 - Subroutine SUBE6 releases the fire control radar at the time of intercept.

TIMMY - Subroutine TIMMY computes the predicted intercept point and flight time for the BLUE SAM/SSM and determines whether the intercept is within the missile's capability.

JAMM - Subroutine JAMM determines the range at which a monostatic radar can detect a target in the presence of jamming.

BIDET - Subroutine BIDET computes detection ranges and times of detection for bistatic radars in a clear or jamming environment.

DTIMD - Subroutine DTIMD determines the real and imaginary roots of the fourth order polynomial used to solve for the bistatic radar detection times.

CPA - Subroutine CPA computes the coordinates of the closest point of approach of a target to a unit.

REACT - Subroutine REACT computes reaction time delays.

PRIOR - Subroutine PRIOR determines the most threatening target among those that a defensive system may engage at a given time.

RCHECK - Subroutine RCHECK evaluates a prospective target ship to determine if the target is above the raider's radar horizon and within the raider's maximum theoretical radar range.

TMDASM - Subroutine TMDASM determines if and when a RED target will close within a specified range of a BLUE unit.

TMDBOM - Subroutine TMDBOM determines the time when a RED raider will come within a given range of force center.

ZLNKA - Subroutine ZLNKA is a sorting routine.

TJSORT - Subroutine TJSORT is a sorting routine.

LOCTIM - Subroutine LOCTIM computes the raider position at a given time.

ASMTIM - Subroutine ASMTIM computes the ASM/SSM position at a given time.

RANDOM - Subroutine RANDOM generates uniform (0,1) random numbers.

HISTG - Subroutine HISTG provides a histogram and associated statistics of hits, hit intervals, and hits per minute for a specified BLUE unit.

DISPL - Subroutine DISPL provides a geographical plot of the BLUE unit positions.

RADIAN - Function RADIAN converts angles in degrees to angles in radians.

RAIDIS - Subroutine RAIDIS provides a geographical plot of the RED raider tracks and launch points.

SETIMX - Subroutine SETIMX provides the internal computer clock time.

2. MAIN

The simulation is initialized by MAIN which also compiles iteration results for the summaries. The MAIN program begins the first of its three principle functions by zeroing specified arrays. It then proceeds to read in the X array, the raider target list assignments (NRTGT), and the target lists (PLIST). The second principle function is to enter those initial values that will remain constant through all iterations into the various tables. MAIN's third principle function is the accumulation and printing of the statistical results.

The initial step in performing the above functions is the zeroing of the X array. In the next step, data are read into the X array by reading data cards in the format

(I10,5F10.3). The information is first read into an integer, IRT, and the array RXATA(5). From there, using IRT as the initial X address, MAIN fills the next five X array slots sequentially. X array addresses that are to remain 0.0 need not be read in unless they are within the neighborhood of the five addresses contained on an input card. In this case they must be specified 0.0. The sequence of reading data cards and filling the X array will continue until a card is read with the value 0 for IRT and 0.0 for the remainder of the card input. This is the flag for MAIN to proceed to the next step.

The program then obtains from the values previously read into the X array the starting values for ISEED, the seed value for the random number generator, and NI, the number of iterations to be played. End of game time is determined and the X array is printed as it was read in. MAIN now computes the weapon (RED ASM/SSM) horizon as follows:

$$\text{HORZ} = (\text{CURVATURE FACTOR}) \times (\text{CRUISING ALT})^{\frac{1}{2}}$$

HORZ is in miles and ALT is in feet. MAIN next converts all courses and bearings from degrees true to degrees in the cartesian coordinate system using the relationship:

$$\text{DEG(C)} = 450.-\text{DEG(T)}$$

All velocities are converted from knots to nautical miles per minute. Jammer and radar beam widths are converted to half beam widths in radians, and MAIN finds the fourth root of the raider and RED weapon radar cross sections. Next the

target list assignments and target lists are read in, and, if X(12) is set, MAIN prints the X-array, PLIST, and NTRGT as read in.

MAIN next fills out the ship table and the raid table. It counts up the number of BLUE missiles (both SAM and SSM) available, and, if X(15) has been set, MAIN prints the converted X array. If X(55) is set, MAIN calls SETIMX to start the internal timer.

MAIN now calls EFFECT to control the first iteration. Upon completion of the first iteration, MAIN prints the ASM table and the SAM-ASM-RAIDER table. It then computes and prints various statistics for this iteration and computes and prints cumulative statistics for all iterations thus far completed. If X(51) is set, the SHIP table is printed at the end of iteration one. If X(52) is set the RAID table is printed and if X(53) is set the SAM table is printed. If X(54) is set, all the random numbers used in the first iteration are printed, in the order of use.

At the end of iteration one, MAIN sets flags 9, 52, 53, 55, 56, 57, 58, 59, and 60 to 0. MAIN then loops back and calls EFFECT to begin the next iteration. Upon completion of the last iteration, additional statistics are printed and the program stops.

3. SUBROUTINE EFFECT

EFFECT controls each iteration. If X(55) is set for debugging, EFFECT calls subroutine GETIMX on entry and exit

to login and logout. EFFECT next calls JULIE to set the tables and arrays that were altered during the previous iteration back to their proper starting values. The simulation begins when EFFECT calls subroutine GETEV for the first event to be played. GETEV supplies EFFECT with the type of event to be played, NEXTEV, the time of the event, and a code identifying the type of event. If the event type returned by GETEV is 0, the iteration is over and the subroutine exits. If the event type is greater than 6, the program prints an error message and terminates. Otherwise EFFECT calls one of the following subroutines to continue the play:

SUBE1	for a RED raider detection and intercept event.
SUBE2	for a RED missile launch-target-detect-intercept event.
SUBE3	for a BLUE missile firing event.
SUBE4	for a RED ASM/SSM impact event.
SUBE5	for a BLUE missile kill event.
SUBE6	for a "set fire control channel free" event.

Upon return from one of the SUBE's, EFFECT again calls GETEV for the next event to be played. Event types used elsewhere in this manuscript correspond to the SUBE subroutine which plays the event.

4. SUBROUTINE JULIE

Subroutine JULIE initializes or resets those items that are required for each iteration. If X(55) is set, JULIE

calls GETIM upon entry and exit. JULIE is called by EFFECT and calls RANDOM, TMDBOM, STOREV, and EVENT.

EVENT is called to zero the events list, and to enter the end of game event into the events list. The following counters are then set to zero; NASM, NSAM, IFAG, IFBG, ITEMS, IFLASK, and IFIRST. The following arrays are zeroed; JAM, IHIT, ISAG, ISBG, STATS, TGTAV, IGAGE, NONO, JACK, and SAMASM. The RAID table and SHIP table are reset and the SAM table is zeroed. The missile batteries are reloaded and, if X(2) is set, the availability of each BLUE radar and launcher is determined. JULIE does this as follows. The first check is on the search radars; if at least one is operating, JULIE determines which fire control radars are operable. If no search radar is available, the program checks to see whether the horizon search mode for fire control radars is authorized, X(7), and if so, whether the ship in question has radars with this capability. If the ship has a search radar and fire control radar combination, or a search radar combined with active, HOJ, ARM, or some other type missile which does not require a fire control radar, the launch availability is checked. If the search radars are inoperative, the fire control radars are horizon search capable, and at least one is operable, the launchers are checked for availability. If the ship has no means of firing or of detection, it is marked a non-combatant and the next unit is evaluated. If the launcher can be coupled with a search/detection/acquisition system, the ship is marked as a G-ship,

the equipment status is entered in the NADAR array, and the next unit is evaluated.

For each RED raider, JULIE calls STOREV to store a type 1 event at time 0., and, if the raider carries ASMs or SSMs, a type 2 event is stored for launch time as determined by TMDBOM.

Finally, JULIE totals the number of BLUE SAM(SSM) batteries available, sets the RELOAD array to zero, and, if X(60) is set, prints the events list.

5. SUBROUTINE EVENT

Subroutine EVENT is called by JULIE. It enters the end of game time in the events list. If X(55) is set, EVENT calls GETIMX upon entry and exit. The event counter LOVER is set to one and the events list is zeroed. End of game time is set equal to the integer IT using the equation:

$$IT = 1000 \times (T + 0.0005)$$

where T is equal to the value stored in X(82). The end of game event, a type 0 event, is then entered in the events list along with the time IT. EVENT then returns to JULIE.

a. <u>Entry GETEV</u>

GETEV is an entry into Subroutine EVENT. It is called by EFFECT and calls GETIMX at entry and exit if X(55) is set. GETEV removes the first entry in the events array, IVENTS (1,1) and IVENTS (1,2), converts these back into event time, type, and participating units, and sends these three pieces of information back to EFFECT. The counter

LOVER is decremented and the entire events list is moved up one position. If the debugging switch X(60) is set, GETEV prints out TIME, NEXTEV, and IUNIT.

b. Entry STOREV

STOREV is an entry in Subroutine EVENT and its purpose is to place the given event in its proper place in the events list. STOREV is called by JULIE, SUBE1, SUBE2, SUBE3, SUBE5, and PRIOR. If X(55) is set, STOREV calls GETIMX upon entry and exit and prints the login/logout time. The inputs to STOREV are the time of the event, the type of event, and the code for the units involved. STOREV first checks to see whether the time the event is to be played exceeds the allotted game time. If it does, STOREV prints a message giving the time of the last event, type of event, and player code. If LOVER is greater than or equal to 1000, the event is lost and STOREV prints a similar message. If both tests are passed, STOREV combines the event type and player code into one 9 digit integer (ICODE). Time is converted to an integer using the same equation as is in EVENT. STOREV places the event in the events array by time. If the time is not unique, it proceeds to compare the stored values of ICODE to the one to be entered. These are ordered from smallest to largest. When the proper place is found, LOVER is incremented, and the remainder of the array is moved up one address. If no place is found to insert the event, including even just right before the end of game event, an error message is printed and the game stops. If the debug switch X(60) is set, STOREV prints the event stored.

6. SUBROUTINE REMOVE

Subroutine REMOVE is called by PRIOR when PRIOR discovers a more threatening target for the missile system being evaluated. If X(55) is set, REMOVE calls GETIMX upon entry and exit. REMOVE receives the code IUNYT from PRIOR. IUNYT is the (event type/player) combination that is stored in the events array. IUNYT was found by TARGET and evaluated by PRIOR as the next event to be played. REMOVE searches the events array for a matching code group. If it finds the group it removes both the code and the time and then moves the remainder of the array up one, and exits. If REMOVE can not find the code, it prints an error message. REMOVE then prints the events array and exits.

7. SUBROUTINE TARGET

Subroutine TARGET is called by PRIOR. Its purpose is to provide PRIOR with all Type 3 events which occur at the present game time. The input for TARGET is time and the outputs are IUNIT and MORE. If IFIRST is equal to zero, TARGET sets it equal to one. TARGET then converts time to the integer format of EVENT and compares it to the time in the array location indicated by IFIRST. If the times are equal, TARGET checks the event code. If it is a Type 3 event, TARGET increments IFIRST, sets MORE equal to one, and exits with the event code. If the times are not equal, or the code is not Type 3, MORE and IFIRST are set to zero and the subroutine exits. If MORE was set to one, PRIOR will

call again until MORE is set to zero. If X(55) is set, TARGET calls GETIMX each time it is entered or is exited.

8. SUBROUTINE SUBE1

SUBE1 is the RAIDER detection subroutine. It is called by EFFECT and in turn calls subroutines CPA, JAMM, STOREV, REACT, LOCTIM, RANDOM, TMDASM, BIDET, and, if X(55) is set, GETIMX on entry and exit. SUBE1 tracks the raider from its start of the game position to its end of game position and determines which BLUE units, if any, will be able to engage the raider. If it is determined an engagement is possible, detection delays, command and control delays, and equipment availability are checked to determine whether or not to enter a Type 3, or shoot, event in the events list array.

SUBE1 first sets flags NASA, NSURF, and NJ. If X(16) is set, SUBE1 checks the ASM counter to see if the BLUE force has been fired upon. If X(16) is set and NASM is zero, SUBE1 stores a Type 1 event at game time plus 2 minutes and then exits. Otherwise, the raider's altitude is checked to determine whether the raider is an aircraft or surface/sub-surface unit, and then the applicable raider characteristics are drawn from the X-array.

Each BLUE ship is next queried to determine whether it can engage the raider. If the ship has been sunk, or if it is a non-missile ship, the NONO table is set to 7 and the next unit is evaluated. The maximum missile range of the operable systems on the BLUE unit is obtained. The distinction between surface and air targets is made by comparing

target altitude to the fifty-foot ceiling for enemy surface units and the fifty-foot floor for enemy aircraft. A missile system must be designated as a surface to surface, or surface to air system when the system characteristics are input. There is no cross matching of systems to targets. SAM's are fired only at aircraft and ASM's, SSM's only at ships.

TMDASM is called to determine whether the target will enter the BLUE unit's missile envelope. If not, the NONO table is set to 9 and the next unit evaluated. If the target will enter the envelope after the end of game time, the NONO table is set to 8.

When it has been determined that the target can be engaged, the target CPA is obtained and the ship's engagement opportunity counter is incremented.

The detection phase begins with a check of the search radars. Each BLUE unit is allowed up to three. If the radar is operable, the clear detection range, jamming environment detection range, combined radar horizon, and theoretical maximum radar detection range are compared. The minimum of these is the radar's detection range for the target in question. This range is compared to that of the other search radars on board, and the maximum is taken as the ship's detection range.

The between radar comparison is made by converting detection ranges into time and adding in the reaction time delays obtained from REACT.

If there is no search radar operational, SUBE1 checks to see whether the target can be detected in the horizon search mode. If there has been no detection and the BLUE unit is operating a bistatic search radar, the subroutine checks to see whether BIDET indicated two detection zones. If so, the search radar portion of SUBE1 is redone using the second bistatic zone. If there remains no detection, or no radar capable of detection, the NONO table is set to 2 and the next BLUE unit evaluated.

Given a search radar detection, SUBE1 proceeds to evaluate the missile systems for fire control acquisition. This is done by obtaining the maximum fire control radar range from among those radars available, and converting this to a time as was done in the search radar algorithm. If the missile of a system requires no fire control radar, the maximum missile range is used as that system's acquisition range.

Having determined the maximum acquisition range, the probability of correct identification is compared to a random number drawn from RANDOM. If the ID check declares the target non-hostile, the NONO table is set equal to 3 and the next BLUE unit is evaluated. Otherwise, the time the target will be acquired is determined by calling TMDASM.

The reaction times for weapons direction equipment delays, target assignment, and acquisition (if the missile requires a fire control radar) are obtained from REACT and added to the search radar detection time. The last time to shoot is determined using the time the target leaves the maximum fire control range, and the SAM/SSM velocity.

The firing delay is obtained from REACT and added to the greater of the detection time and the acquisition time. The "shoot" time is compared to the last time to fire, and if the shot is still possible, the program stores a shoot, Type 3, event for each director-launcher combination in operation. If the missile requires no director, the number 7 is stored as the channel number, and only one shoot event is stored for that system. If the unit requires a director, no channel number is assigned.

If the delays have precluded a firing, SUBE1 checks to see whether there was a second bistatic detection. If so, it returns to the search radar portion of the algorithm and proceeds as above evaluating the second bistatic detection zone. Otherwise it sets the NONO table equal to 5 and processes the next BLUE unit.

The horizon search portion of SUBE1 is entered as described above. The program checks the NADAR table to see if the directors can operate in the horizon search mode. If so, the program proceeds in the same way as the search algorithm, using the fire control radar parameters. Having determined the detection time, the decisional and reaction delays for horizon scan mode are obtained from REACT. The shoot time is determined to be the detection time plus the acquisition time, plus the maximum of the firing time and decision to fire time. A range check is made by calling TMDASM and if the time to fire is less than the last time to fire, a Type 3 event is stored for each director operable in the horizon search mode.

After all the BLUE units have been processed, SUBE1 exits calling GETIMX if flag X(55) has been set.

9. SUBROUTINE SUBE2

SUBE2 is the ASM/SSM launch and detection subroutine. It is called by EFFECT and in turn calls RCHECK, REACT, STOREV, BIDET, RANDOM, TMDASM, TJSORT, CPA, LOCTIM, JAMM, and if X(55) is set, GETIMX on entry and exit.

The inputs to SUBE2 are launch time and raider number. The subroutine first checks the RAID table to see whether the raider has already been shot down or sunk. If so, the subroutine exits. Next the program checks the flag IFLASK to determine whether any RED ASM/SSM's have been fired. If IFLASK is greater than zero, the subroutine proceeds to check whether the launch time is greater than TMAX and exits if it is. If IFLASK equals zero, SUBE2 sets IFLASK to one and TMAX, the time by which all ASM's must be launched, to present time plus X(84). SUBE2 now retrieves the raider speed and, if the raider is an aircraft, determines the ASM launch interval, sets IFLASK equal to 2, and determines the total number of ASM's on board. If the raider is a surface unit; and no ASM's have been fired, IFLASK is set to 0.

SUBE2 determines which of the two possible missile types has the longest firing range, (firing range is measured from coordinate (0,0)). The missile parameters are drawn from the X vector, and the launching sequence begins.

The launching of ASM's is done by drawing a series of uniform (0,1) random numbers, depending upon how many missiles

are to be launched, and ordering them. The exact launch time of each ASM becomes the present time, plus the next random number times the launching interval. The launching of SSM's from surface units begins with the first launch at time plus some portion of the interval specified in X(86). Subsequent surface launches are at fixed increments, X(1809).

Having determined when, and in the case of surface to surface launches, how many RED missiles are to be launched, the targeting phase beings. LOCTIM is called to insert the raider position at launch into the RAID table. The raider target list is located and a temporary target list is constructed. Each ship on the raider's target list is cycled through RCHECK to determine whether it is afloat, above the radar horizon and within radar range. If all these tests are passed, the ship is entered on the temporary target list by putting the values from the original target list in the next slot of the temporary target list. Once the target list has been formed, a random number, Z, is drawn from RANDOM and the first ship with a cumulative probability of being targeted greater than Z is designated the target. If there is none, the first ship on the list is taken. A check is made to see whether this ship has been designated a non-radiating picket. If not, SUBE2 proceeds to the detection phase. If the target chosen is a non-radiating picket, checks are made to see whether the raider can correctly identify the target. This is doen by determining whether the raider is within X(23) miles of the target. This

measurement is done in the horizontal plane. If the raider is close enough to the target, RANDOM is called for a random number to compare with the probability of correct identification. If the picket fails to qualify as a target, it is eliminated from the target list and the process starts again. If, at any time the situation arises where there are no targets on a raider's target list, launching stops and an event Type 2 is stored for time plus three minutes.

Upon acquiring a target, the ASM/SSM is considered launched. A random number is drawn and compared to the launch reliability. The ASM table is filled out and, if launch was unsuccessful, the NONO table is set equal to 1 and the next missile processed. Upon an unsuccessful launch, a check is made to see whether there is to be a second SSM in the same salvo. If so, the time is not advanced for this launch.

Assuming a successful launch, the detection phase begins. The target ship's position is stored in the RAID table for use in BIDET, but otherwise, the detection sequence follows that outlined in SUBE1.

When all of the first type of missiles have been launched, and if the raider is carrying a second type of missile, SUBE2 decides whether or not the raider has reached its second launch point. If so, the appropriate parameters are extracted and the launching continues; otherwise an event Type 2 is stored for the new launch time and the subroutine exits. When there are no second type missiles, the subroutine exits after the last launch.

The exit sequence consists of checking X(58), (if set the NONO table is printed), and X(55), if set, (the exit clock time is printed). The NONO table codes for SUBE1 and SUBE2 are:

0 System can engage the target.

1 The ASM failed at launch.

2 The ship did not detect this target.

3 The missile or raider was not identified as a target suitable for engagement.

4 The ship can not acquire the target.

5 Reaction delay was too long.

6 The missile will impact prior to coming into max SAM range.

7 Non missile ship.

8 Detection will not occur until after end of game.

9 Can not engage.

10. SUBROUTINE SUBE3

This subroutine evaluates the engageability of the target in light of the current situation at the proposed launch time. SUBE3 is called by EFFECT, and it in turn calls PRIOR, STOREV, CPA, TIMMY, and, if X(55) is set, GETIMX on entry and exit.

The inputs to SUBE3 are event time and a participant code. The code is an integer of the following type, IIIJJKLM, where III is the target number, JJ is the firing ship, K is the missile system, L the director, and M is the channel number. The subroutine breaks down the code and then checks the appropriate portion of the SHIP table to determine whether the firing ship has been sunk, (if X(5) has been set, this check will indicate the ship has not been sunk

even if it has absorbed more hits than are considered necessary to put it out of action). If the ship has expended the magazine of this missile system, the subroutine exits. Having established the credibility of the firing ship, SUBE3 determines whether the raider has been shot down or sunk, and if so, it exits.

SUBE3 now determines the maximum missile range for the system under consideration and compares this with the raider's CPA. If the target will not enter this maximum missile envelope the subroutine exits. PRIOR is called to determine whether this ship/system/director combination has another shoot event at the present game time and, if so, if that target is more threatening. PRIOR logic will consider the closest target as most threatening. If PRIOR returns the same target, the subroutine continues. If not, it checks out the target as above and exits or continues as appropriate.

If the target has not been destroyed, SUBE3 checks the coordination set-up. If both the fleet and the firing ship are coordination capable, the subroutine checks the target availability. If the target is under engagement, the subroutine calls STOREV to store a Type 3 event at the time the target will become available. If the target remains engageable, SUBE3 checks the NONO table. If the value 0 is found the event continues. If not, the subroutine exits.

The subroutine now goes to the channel assignment phase. If the SAM or BLUE SSM requires no director, channel number

7 has already been assigned in SUBE1 or SUBE2, and the flag ICOT is set to 0. If the missile requires a director for part or all of its flight, and a channel has been assigned on a previous run through SUBE3, the subroutine checks to see whether the assigned channel is busy. If it is, ICOT is set to 1; if not, ICOT is set to 0. If the missile has not been assigned a channel previously, the availability of a channel is checked. A free channel, if found, is assigned and ICOT is set to 0; otherwise a search is made to determine which channel will first become available and this one is assigned. If at this point all channels are busy and the director is required for the full SAM/SSM flight, STOREV is called and a Type 3 event stored for the time the director will become available.

If the systems all indicate they are engaged in an intercept and upon searching the SAMASM and SAM tables no record of this can be found, the program terminates after printing an error message.

SUBE3 now proceeds to check whether the firing ship itself is engaging the target. If it is the subroutine exits. If the target is an ASM or SSM, the subroutine checks to see that the target will not impact prior to BLUE missile launch. If it will, the program exits.

At this point, if ICOT equals 1, the channels were all considered busy and the BLUE missile only requires the director for terminal homing. SUBE3 checks the SAMASM listing to determine when the channel will be available and stores a Type 3 event.

If ICOT equals 0, SUBE3 calls TIMMY to determine the intercept position, single shot kill probability, SAM/SSM flight time, and the intercept possibility given the BLUE missile flight and warhead characteristics. If TIMMY returns the possibility flag -1, the intercept is not possible, but may be in the future. SUBE3 stores a Type 3 event for present time plus six seconds. If the indicator is zero, the intercept is not and will not be possible, and the subroutine exits. If the indicator is one, the intercept is now possible and the program continues. If the intercept was possible, but the intercept point was in the hazard zone of another BLUE unit, TIMMY returns the indicator two. In this case, SUBE3 again stores a Type 3 event for present time plus six seconds.

If the SAM/SSM requires the director for only the terminal phase of flight, SUBE3 determines whether the director can be scheduled for the intercept. If not, the subroutine stores a Type 3 event for the time the director will be free, minus flight time, plus terminal homing time, plus 0.06 seconds.

If scheduling is not needed, or the director will be available when needed the missile is considered launched. The SAM-ASM-RAIDER and SAM tables are filled in and missiles on board and in the fleet counters are decremented. If there is coordination, and the firing ship is coordination capable, the target availability is up-dated. The launcher reload time is determined and stored. The channel

availability indicator is set to 1 and a Type 5 event is stored for intercept time.

Prior to exit, if X(9) is set, SUBE3 prints out the firing ship number, the target number, and the reason for exiting the subroutine are printed. The breakdown for the REASON code for SUBE3 follows:

0	Blue Salvo fired.
1	Firing ship sunk or out of action.
2	Target being engaged.
3	Target already destroyed.
4	Firing ship out of missiles.
5	Director required for entire missile flight and is not available.
6	Firing ship already engaging the target.
7	Missile requires director only part time but all channels are busy.
8	NONO table prohibits engagement.
9	Launcher not available.
10	ASM will impact prior to SAM launch.
11	Director cannot be scheduled for the intercept.
12	Director operation in horizon scan mode, busy.
13	Target will not close BLUE missile envelope or TIMMY indicates no intercept.
14	TIMMY indicates target may enter envelope later.
15	TIMMY indicates target intercept point is too close to another BLUE unit.

11. SUBROUTINE SUBE4

SUBE4 determines whether the RED missile failed in flight, and if not, increments certain counters indicating ship hits and possible sinking. SUBE4 is called by EFFECT and it in turn calls RANDOM, and, if X(55) is set, GETIMX on entry and exit.

SUBE4 first breaks down the input code into the RED ASM/SSM number, and the BLUE target ship number. The code enters SUBE4 as the integer RRRBB. It then checks to see whether the ASM/SSM has been shot down, or had its launch negated due to the loss of the launching raider prior to the actual missile launch time. If neither is the case, the probability of successful flight is drawn from the X array and RANDOM is called to produce a random number. If it is determined that a failure occurred, the ASM table is set to 3 and the subroutine exits. Otherwise the target ship is assumed to have been hit and the appropriate counters are incremented. If X(5) is not set, the subroutine compares the number of hits taken to the amount of damage the ship can withstand, and if too many hits have been taken, declares the ship sunk.

12. SUBROUTINE SUBE5

SUBE5 evaluates the BLUE missile intercept and performs checks on other ASM/SSM if the target shot down or sunk was a RED raider. SUBE5 is called by EFFECT and it in turn calls STOREV, REACT, RANDOM, and if X(55) is set, GETIMX on entry and exit.

SUBE5 is called with an input variable of salvo number. Using this, SUBE5 gets the target number, firing ship number, missile system, director, and channel number from the SAM-ASM-RAID table. The director reassignment and reacquisition delays are determined and stored, fleet and individual ship

statistical data are taken, and a check is made to insure that the target is still engageable. If the target has already been sunk or shot down, the SAM-ASM-RAID table is set to -1, indicating an overkill. If the target is an ASM/SSM that failed in flight or has already impacted, the salvo is determined to have been a failure and the SAM-ASM-RAID table is set to zero. If the target is engaged, the probability of BLUE missile successful flight and support is pulled from the X array, and the single shot kill probability is removed from the SAM-ASM-RAID table. RANDOM is called to provide a random number. Success is determined by comparing PK to the random number drawn. This is done by determining PK as follows:

$$PK = PS \times (1-(1-PKSS)^n)$$

Where PS is the probability of successful flight, PKSS is the single shot kill probability, and N is the number of BLUE missiles in the salvo. If the random number is less than PK, the shot was successful. If the shot was unsuccessful and the missile requires a director, the channel is released and Type 3, shoot, event is stored for present game time. If the missile doesn't require a director, just the shoot event is stored, and the subroutine exits.

If the shot was successful and the target was an ASM/SSM, the SAM, SAM-ASM-RAID, ASM, and kill counters are updated, a Type 6, channel free, event is stored, and the subroutine exits. If the target destroyed was an aircraft, or if the number of hits has been sufficient to disable a

surface raider, the following steps are taken prior to the regular table update and exit. A check is made to determine whether the raider has any ASM's or SSM's aboard. If so, the ASM counter is incremented and the appropriate numbers are entered in the ASM table indicating that these missiles were lost with the launching platform. Next the ASM table is checked to locate any missiles launched by the raider just destroyed. When one is found, its launch time is compared to raider kill time, and if the launch time is greater, entries are changed in the ASM table to correspond. Upon completing the search, the program continues to exit where it checks X(55), X(56), and X(58) for debug output.

13. SUBROUTINE SUBE6

SUBE6 releases the fire control radar channel at the time of the event. SUBE6 calls GETIMX upon entry and exit if X(55) is set.

The input value to SUBE6 is an integer code AABCD, where AA is the firing ship number, B is the missile system, C is the director number, and D the channel number. After breaking down the code, SUBE6 sets the appropriate portion of the CHANEL array to zero, and exits.

14. SUBROUTINE TIMMY

TIMMY computes the predicted intercept point and flight time for the BLUE SAM/SSM and determines whether the intercept lies within the intercept contour. TIMMY is called by SUBE3 and TIMMY in turn calls TMDASM, ASMTIM, RADIAN, ZLNKA, LOCTIM, and if X(55) is set, GETIMX on entry and exit.

Inputs to TIMMY are target number, firing ship number, missile system number, present target position, and present game time. Outputs are single shot kill probability, flight time, and an indicator flag which says the intercept is possible, is not and will not be possible, or is not but may be possible in the future.

Using the target number, TIMMY decides whether the target is a raider or an ASM/SSM and determines the end of game position for raiders and target ship position for an ASM/SSM. If X(8) is set, firing sectors have been assigned and TIMMY checks to ensure that the assigned firing sector and the launcher train limits permit a launch. If launch is prohibited, REASON is set to 2, flag IND is set to zero, and the subroutine exits. If the launch is possible, there may be as many as two firing zones. These two occurrences are shown in Figure B1.

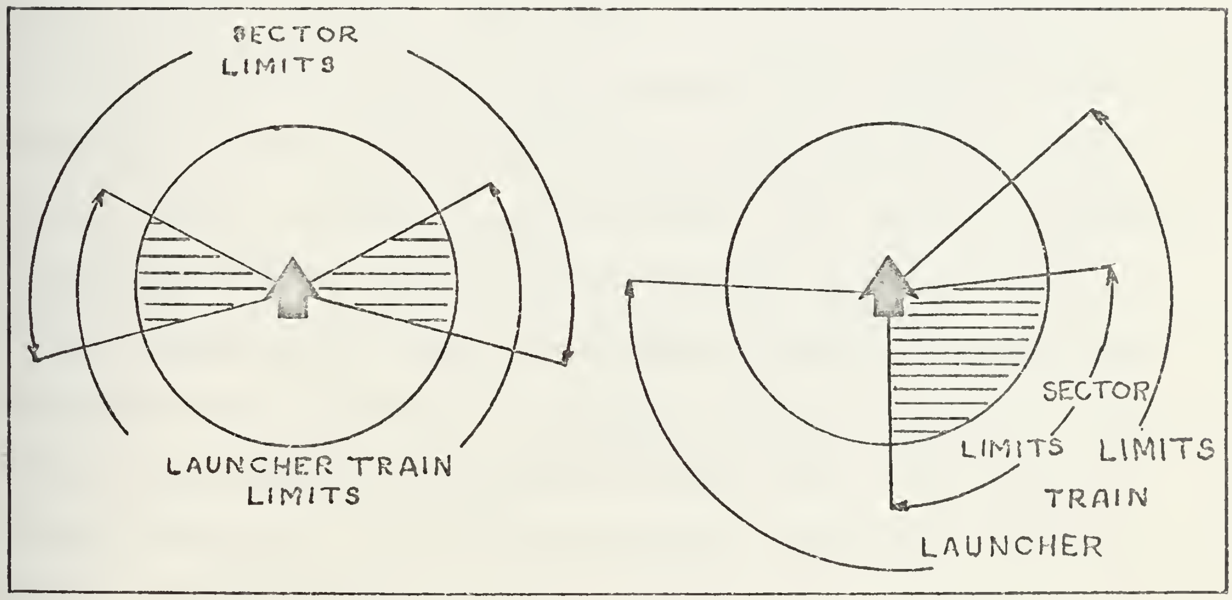


Figure B1
One or Two Firing Zones

Using the results of the ZLNKA reordering of the sector limits and launcher train limits, TIMMY determines whether there are two zones or one, and which of the bearings constitute the actual firing zone, or zones, limits. If no sectors have been assigned to the fleet, or this unit, the firing zone limits coincide with the launcher train limitations.

TIMMY next checks target altitude against the maximum SAM altitude. If the target is too high, REASON is set to eleven and the subroutine exits. Otherwise, the intercept parameters are determined. These intercept parameters are the values entered into the X-array that approximate the actual intercept contour. The intercept parameters are A, B, N, SLOPE, and SEEK. The actual intercept contour depends on target size, speed, and altitude as well as missile parameters. There are usually more than one contour for any given missile system. The actual contours are found in the various technical publications dealing with the specific missile. OP-3594, Performance and Installation Characteristics Surface Missile Weapon Systems, also contains general contour nomograms, and the Applied Physics Laboratory of Johns Hopkins University has developed computer simulations which generate contours. A typical contour, along with the TIMMY approximation, is shown in Figure B2. A is the off-range capability, B is the down-range capability, and N is a degree of curvature.

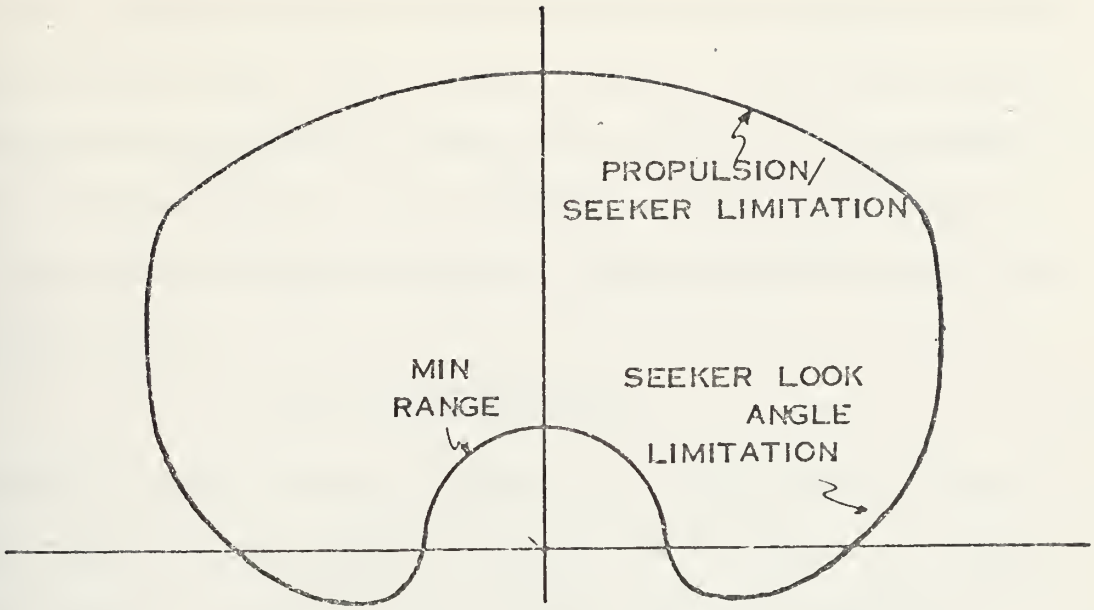


Figure B2a
The Actual Intercept Contour

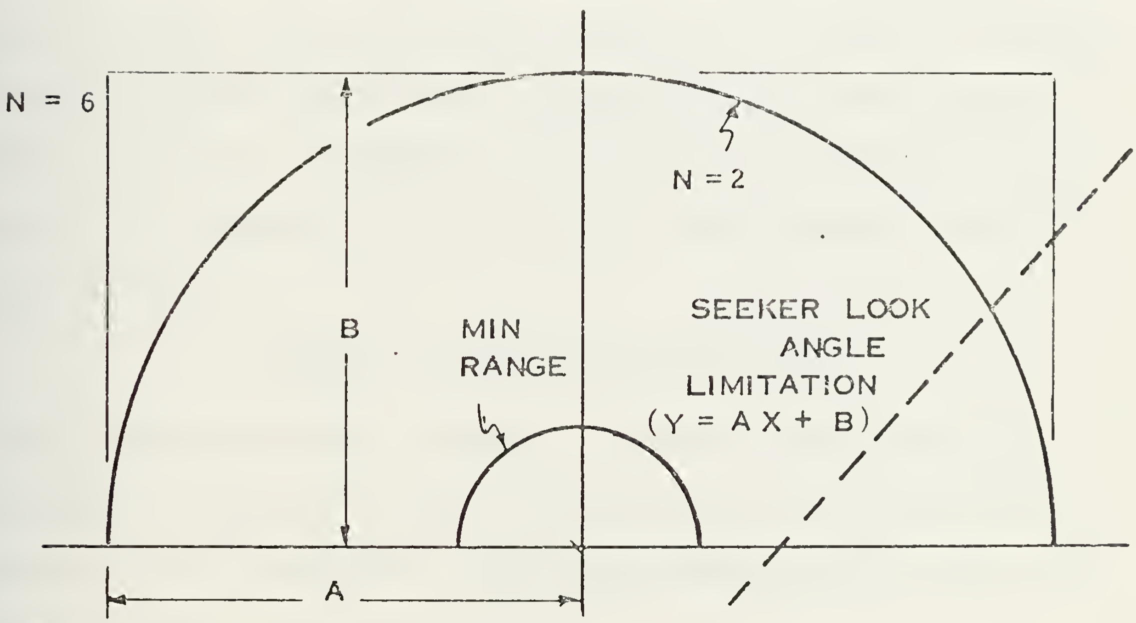


Figure B2b
The Approximation of the Contour

The maximum missile range for the given target altitude is determined, and if the target will not enter this range, as determined by TMDASM, TIMMY exits after setting REASON to seven. If the target is an ASM/SSM and the target ship is the firing ship, target course is found and SAM flight time is determined by:

$$FT = DLT/(SSPD+VEL)$$

where DLT is the distance between the firing ship and the target, SSPD is the speed of the SAM, and VEL is the speed of the target. In this case the PK that is returned to SUBE3 is that for a non-crossing target.

If the target ship is not the firing ship, or the target is a raider, the following angles are determined: ALPHA, the angle between the target flight path and the line of sight at launch, and THETA, the angle in Cartesian coordinates of the present line of sight. The quantity GSINE is determined by:

$$GSINE = SIN(ALPHA)*(VEL/SSPD)$$

If the absolute value of GSINE is greater than one, the intercept is impossible and TIMMY exits after setting REASON equal to ten. Otherwise, the angles GAMMA, BETA, and DELTA are computed.

$$GAMMA = ARCSIN(GSINE)$$

$$BETA = 180° - GAMMA - ALPHA$$

$$DELTA = 180° - BETA$$

GAMMA is the angle between the SAM/SSM flight path and the line of sight, BETA is the angle between the SAM/SSM flight path and the target flight path, and DELTA is the compliment of BETA. If X(59) is set, the values for target number, firing ship number, ALPHA, BETA, GAMMA, present target position, GSINE, and present range to the target are printed. PK for this type of target is determined by DELTA. If DELTA is greater than 45°, the target is declared a crossing target and the appropriate PK is retrieved from the X-array. Otherwise, the probability of kill for a non-crossing target is retrieved. Flight time is determined by:

FT = DET * SIN(ALPHA)/(SSPD * SIN(BETA))

Using present time plus flight time for the input, TIMMY now calls LOCTIM or ASMTIM, as appropriate, to determine the intercept coordinates. The range to the intercept is computed and compared to the minimum missile range. If the range is too short, IND is set to minus one, REASON to four, and the subroutine exits. The launch bearing is calculated and the first of two possible sets of firing zone bearings are compared to determine whether the launch bearing is within the firing bearings. If it is not, TIMMY checks to see if there was a second firing zone, if so it checks this one, otherwise, IND is set to minus one, REASON to three, and TIMMY exits. If the intercept angle is good, the intercept parameter is checked. A, B, and N are combined to form the equation:

$$(X/A)^N + (Y/B)^N = 1$$

A value of two for N results in a circle or an ellipse. A value of six results in a near rectangle. Fractional values for N are permitted.

The "ellipse" is translated so that its center is on the firing ship, and rotated so that the positive Y axis is opposite to the line of flight of the target. The intercept coordinates are substituted into the equation, in absolute value, and if the resulting number is less than or equal to one, the intercept will be within the ellipse. The next check is the seeker limit test. The y intercept value and the absolute value of the x coordinate are entered into the equation:

Z = YI - SLOPE * XI

If Z is greater than or equal to SEEK, the problem continues. If not, TIMMY exits after setting REASON to six. The subroutine now checks the X-vector to determine whether there are any restrictions on how close an intercept can be to another friendly unit. If there are, the hazard zone is constructed. If the intercept falls within the hazard zone, REASON is set to nine, IND is set to two, and TIMMY exits.

If the problem has progressed this far, the intercept is considered possible and the BLUE missile salvo is launched while IND is set to one. If the target is an ASM/SSM and the firing ship is the target, the program exits. If not, the target altitude is figured into the flight time prior to return to SUBE3. If X(9) is set, TIMMY prints out the ship number, target number, IND value, and REASON prior to exit.

Reason codes as used in TIMMY

1 Not used.
2 No firing zones for this system.
3 Firing bearing outside sector.
4 Target too close.
5 Target intercept point outside engagement contour.
6 Target beyond seeker limits.
7 Target beyond range and outbound.
8 Intercept after ASM/SSM impact.
9 Target too close to BLUE unit at intercept.
10 GSINE greater than one.
11 Target too high.

15. SUBROUTINE JAMM

The subroutine JAMM determines the range at which a monostatic radar can detect a given target in a multiple standoff and self-screening jamming environment.

The calling routine, either SUBE1 in the case of raiders, or SUBE2 in the case of missiles, first determines that there are jammers in the problem and that at least one jammer is on the same frequency band as the detecting radar. If both situations obtain, subroutine JAMM is called.

JAMM initially checks to see if the target will close the detecting unit within the minimum of the clear environment, radar horizon, and theoretical range of the radar. If the target will not close within this minimum range, the subroutine is exited with a flag set to show that detection would not occur. In the case where the target will enter this minimum range, JAMM computes the time at which the target will arrive at the closest point of approach (CPA) to

the detecting radar. Appropriate checks are made for stationary targets and targets with only minor changes in their X or Y position components. JAMM next obtains the bearing from the radar to the target at the CPA position for use in selecting the appropriate side-lobe coefficient. Individual radars are then checked to see if they have been shot down prior to the target arriving at CPA (applicable only for missile detections), if they are jamming on the frequency band of the detecting radar, and if they are above the detecting radar's horizon. If all of the conditions are met, the bearing from the jammer to the jammer's jamming target is found in order to determine whether the detecting radar is within the main beam or the side-lobe of the jammer. The jamming power arriving at the radar from the individual jammer is then computed, taking into consideration the power of the jammer, the range from the jammer to the detecting radar, and the side-lobe/main beam relationship between the detecting radar and the individual jammer. The jamming power arriving at the detecting radar is then summed over all of the jammers and used to compute the burn-through range for the radar against the specified target. Prior to exiting the subroutine, the range to the target's CPA to the detecting radar is checked to ensure that it is less than the burn-through range. If the distance to CPA is greater than the burn-through range a flag is set indicating that detection did not occur due to CPA considerations. Otherwise the subroutine returns as the detection range the minimum of the clear, jamming, radar horizon, and theoretical ranges.

16. SUBROUTINE BIDET

This subroutine computes the number of detection zones, ranges, and times of detection for a bistatic radar in either a clear or jamming environment. BIDET is called by SUBE1 and SUBE2, and it in turn calls GETIMX on entry and exit if X(55) is set.

BIDET begins by setting the initial values of aircraft/ASM start position, final position, altitude, and velocity. The transmitter and receiver positions and radar antenna height are established for the bistatic system under consideration. A check for non-moving targets is conducted prior to entering the closest point of approach (CPA) routine. In the case where the target has moved less than 0.1 nautical mile in both the X and Y direction, the target's speed is considered to be zero and the target enters a special loop to determine whether it is detectable at its initial position within the constraints of both radar horizon and radar signal strength.

For the moving target, the CPA of the target's projected track to the transmitter is determined by solving simultaneously the equations of the target's track and the line perpendicular to the target's track and passing through the transmitter location. The CPA to the receiver is solved in a similar manner. If movement in either the X or Y directions is less than 0.1 nautical mile, the CPA is established by setting the CPA coordinate for the small changing axis to the target's original position corresponding coordinate and

the alternate coordinate to the corresponding alternate coordinate of the transmitter or receiver. For example, if the X coordinate of the target's positions changes less than 0.1 nautical mile, the CPA to the receiver is set to the X coordinate of the target's initial position and the Y coordinate of the receiver's position.

The radar horizon for the transmitter and the receiver are then compared to the distance to their respective target track CPAs to determine whether the target enters the radar horizon of both the transmitter and the receiver. If it fails to enter both, the routine is exited with a no detection. Otherwise, the times that the target first appears above and disappears below the transmitter and receiver radar horizons are computed. The time interval that the target is simultaneously above both horizons is determined by taking the maximum of the appearance times and the minimum of the disappearance times. If the target is never simultaneously above both horizons; or the appearance time is greater than maximum game time; or the disappearance time is prior to start of game time, the subroutine is exited with a no detection.

Next, detection times for the target based on radar signal strength are computed, and the intersection of these time intervals with the above joint radar horizon time interval determines the time during which the target is detected. No detection occurs if the intersection of the two time sets (i.e. radar strength detection and radar horizon detection) is the null set.

If a detection has occurred in the clear environment, the subroutine checks to see whether any jammers are in use and if they are on the same frequency band as the bistatic radar. If jammers are in use, the time that the target arrives at its midpoint in the bistatic detection envelope is determined. This time is used in arriving at an average jammer position to compute the jamming power at the radar during the detection phase. This will tend to provide a somewhat conservative estimate of detection ranges in the face of jamming because the jammers are in general going to be nearer to the radar and the range square losses of the jamming power will be reduced. In the case of the stationary target, midpoint time is set to an arbitrary three minutes after start of game time to allow for some movement of the jammers toward the force. The subroutine then adds the jamming power from each jammer. It does this by checking all of the raiders to determine if the individual raider has been shot down prior to the midpoint time, and if the jammer is above the radar horizon and is jamming on the frequency band of the radar. If these conditions are met, the subroutine examines the geometry of the jammer's position and the radar to determine the relationship between the jamming and radar antennas. With this relationship, the appropriate side-lobe/main beam antenna coefficients are used to adjust the jamming level.

The constant in the bistatic range equation is then adjusted to reflect the jamming environment and the computation

previously conducted in the clear environment are repeated to determine whether detection will occur in the jamming situation.

The subroutine exits with the number of detection zones which occurred; the time and range of initial detection in the first zone, the time of last detection in the first zone; and, if applicable, the time and range of first detection in the second zone along with the time contact is lost in the second zone.

17. SUBROUTINE DTIMD

Subroutine DTIMD is a double precision subroutine that determines the real and imaginary roots of the fourth order polynomial which is used to solve for the bistatic radar detection times. The subroutine employs the Newton-Raphson iterative technique with the final iteration on each root performed using the original polynomial rather than the reduced polynomial to avoid accumulated errors in the reduced polynomial. The subroutine was obtained from the Naval Postgraduate School Computer Center library.

18. SUBROUTINE CPA

Subroutine CPA obtains the closest point of approach of a target's track to a reference point by solving the simultaneous equations of the line representing the target's track and the line through the reference point and perpendicular to the target's track. Checks are conducted for target tracks which pass through the reference point,

represent stationary targets, or targets which have only minor changes in either the X or the Y direction.

19. SUBROUTINE REACT

Subroutine REACT allows the use of reaction time delays through the use of probability distribution functions. This subroutine is similar to, if not identical to, one developed for use in FLOATS by APL at the Johns-Hopkins University. The distribution functions may be entered for seven classes of delay. They are:

1. Target detectability to actual detection.
2. Detection to ID and designation to the weapons direction equipment.
3. Director assignment delay.
4. Director acquisition delay.
5. Firing delay.
6. Targēt detectability to detection for horizon scan mode.
7. Decision delay in horizon scan mode.

Within these class areas, up to five separate types may be designated. REACT uses the type and class numbers to determine a delay time which it returns to the calling subroutine. REACT is called by SUBE1, SUBE2, and SUBE5. REACT in turn calls only RANDOM.

The inputs to REACT are the class number and type, and the output is the delay in minutes. The class number is used to locate that portion of the X-array where the delay is stored. The type number is used to localize the address within the area. In employing its Monte Carlo technique, REACT calls RANDOM and converts the returned uniform (0,1)

number to an integer. If this integer is 10, the subroutine sets it equal to 9. If the integer is zero, the RANGE, defined as the difference in time between two adjacent blocks, is set equal to the value in the first block. Otherwise, the address is determined using the class and type numbers and then adding on the integer. The delay consists of two parts. The first part is the time in the address block. The second part is determined by drawing another random number and multiplying it by RANGE. Except in the case where the integer is zero, RANGE is determined by subtracting the time in the address block from that in the next highest block. In the "zero" case, the total delay consists of this fraction of RANGE. In all other cases, the fraction of RANGE is added to the time stored in the address block to obtain the total delay. At this point REACT returns to the calling routine.

As an example of a distribution function, consider the waiting time for an aircraft for a catapult launch. Data is available for one hundred aircraft. The maximum waiting time was seven minutes. The distribution of waiting times was as follows:

length of wait	0-1	1-2	2-3	3-4	4-5	5-6	6-7
frequency	10	40	15	15	5	5	10

Figure B3

An Example of Waiting Time

The plot of length of time waiting against number of occurrences would look like this:

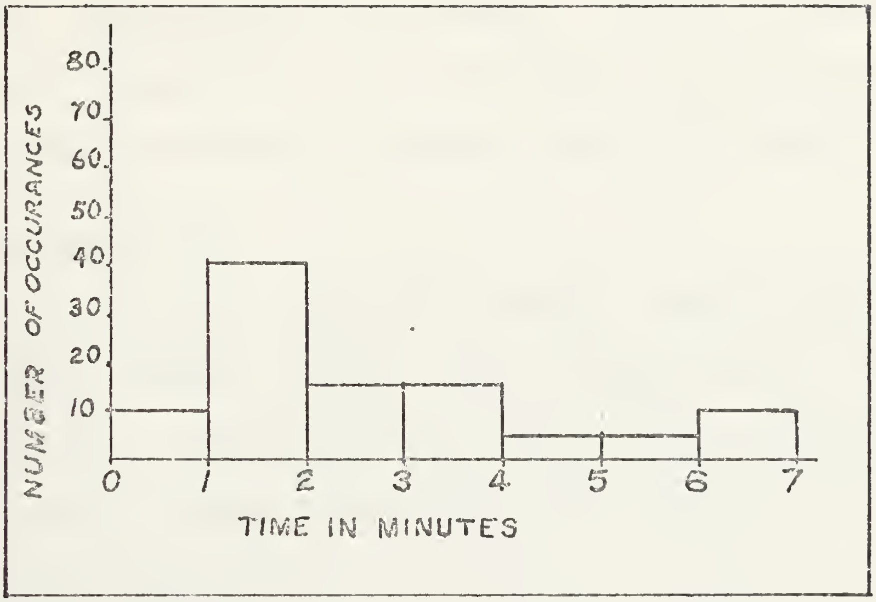


Figure B4
Histogram of Waiting Times

The probability of waiting between 0.0 and 1.0 minutes is 0.1, the probability of waiting between 1.0 and 2.0 minutes is 0.4. When the histogram is transformed into a cumulative probability distribution function, the following results.

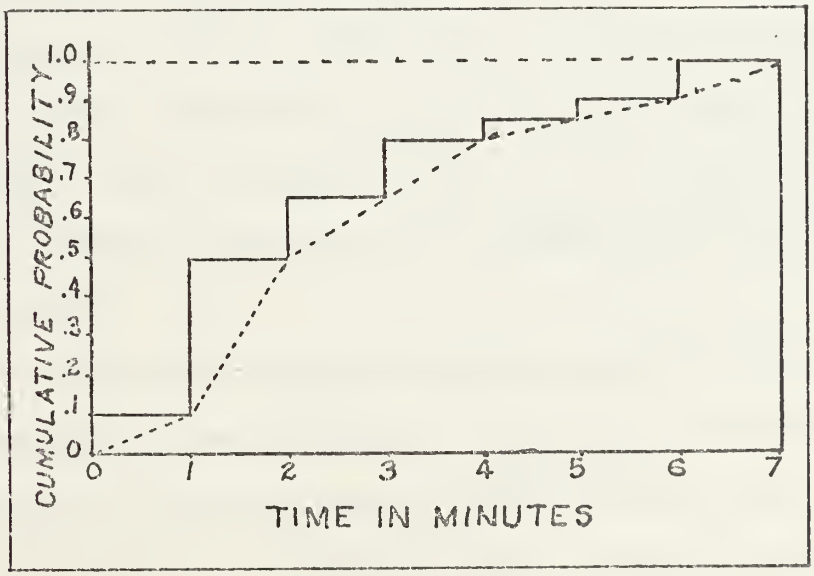


Figure B5
Cumulative Distribution Function of the Waiting Times

This distribution is what is entered in the X-array. The first block would contain the time for .1 cumulative probability. This would be one minute. Block two that for .2 cumulative probability, or 1.25 minutes, and so on up to block ten which would have entered a seven for seven minutes.

20. SUBROUTINE PRIOR

Subroutine PRIOR is called by SUBE3. PRIOR calls REMOVE, STOREV, LOCTIM, ASMTIM, TARGET, and if X(55) is set, GETIMX on entry and exit. The purpose of PRIOR is to determine the most threatening target among those that a system may choose from at present game time.

PRIOR first computes the value CHECK for the target being compared. CHECK is defined as maximum missile system range divided by present target range. TARGET is called to provide PRIOR with a candidate for comparison. The same measure of threat is obtained for the candidate target and stored in CHERRY. If the present value in CHECK is greater than or equal to CHERRY, PRIOR goes to the next candidate, if there is one. If CHERRY is greater than CHECK, the values of target number, channel number, and target position are saved. CHECK is made equal to CHERRY and PRIOR seeks a new comparison.

If the value returned by TARGET in MORE is not zero, there is another candidate to be evaluated. If MORE is zero, there are no more comparisons to make and the final section of PRIOR is entered. The stored value of most threatening target number is compared with the target sent to PRIOR

originally from SUBE3. If they are the same, the subroutine exits. If PRIOR has found a more threatening target, REMOVE is called to take the type 3 event associated with the new target out of the events list. STOREV is then called to enter a type 3 event for the target being intercepted.

The debug switch X(9) is checked at exit and if it is set, old target, new target, missile system (in the code AABC were AA is ship number, B is system number, and C is director number), old channel, and new channel are printed. X(55) is checked and PRIOR returns to SUBE3.

21. SUBROUTINE RCHECK

Subroutine RCHECK evaluates the prospective target ship to determine if the raider can in fact "see" the target. RCHECK is called by SUBE2 and RCHECK calls GETIMX on entry and exit if X(55) is set. The inputs to RCHECK are the raider number and proposed target ship number. The two dimensional distance between the two units is computed. The antenna heights on the target ship are compared and the tallest one is used to determine the ship's radar horizon. If the ship has no radar, a height of 15 feet is entered. The combined radar horizons, maximum theoretical radar range of the raider's radar, and the actual range are compared. If the target is within both the radar horizon and the maximum radar range, the go/nogo flag is set to zero. If either test fails, the flag is set to one and the subroutine exits.

22. SUBROUTINE TMDASM

This subroutine determines if the target, or other unit in question, will enter within a certain input range of a BLUE unit, and if so, at what time will it enter and leave. TMDASM is called by SUBE1, SUBE2, and TIMMY. TMDASM in turn calls GETIMX on entry and exit if X(55) is set.

TMDASM begins by determining the range from the reference ship to the target unit. If this range is less than or equal to the input range, the time of entry is considered to be the input time. If the range to the target unit is greater than the input range, TMDASM determines the slope of a line representing target unit movement. Using this and the present target unit position as a reference point, TMDASM forms the line equation. This equation and the equation for a circle with radius equal to the input range are solved simultaneously to determine the points of intersection. This is done by substituting the line equation into the circle equation and solving the quadratic equation. If the roots are imaginary or equal, the entry is either not made or the length of stay is infinitely small. These cases are rejected as no solution. Upon finding two real roots, the subroutine determines which of the two points the target unit would reach first, and which one it would reach second. If the target unit is already further down its track than the exit point, no entry is indicated. Otherwise the time of leaving the circle is determined and, if the target unit was not already inside the circle, the time of entering.

23. SUBROUTINE TMDBOM

This subroutine determines when the raider in question will reach a given range from the force center, position (0,0). TMDBOM is called by MAIN, and JULIE. TMDBOM in turn calls GETIMX on entry and exit if X(55) is set.

TMDBOM proceeds in the same way as TMDASM except that the reference position, center of the range circle, is always force center instead of a firing ship. TMDBOM only computes the time that the raider will cross a given range. It neglects the departure time.

24. SUBROUTINE ZLNKA

Subroutine ZLNKA is a copy of the subroutine SHSORT, mentioned below, as it existed on 1 February 1975. ZLNKA inputs a vector of real numbers and an ordered vector of integers. ZLNKA reorders the real vector from smallest to largest and reorders the integer vector to the same order as the real vector. ZLNKA is called by TIMMY.

25. SUBROUTINE TJSORT

Subroutine TJSORT is a modification of the subroutine SHSORT which is available from the Naval Postgraduate School Computer library. TJSORT is called by SUBE2, and BIDET. This subroutine inputs a vector of real numbers and rearranges them from smallest to largest.

26. SUBROUTINE LOCTIM

LOCTIM computes the raider position at time T. It is called by MAIN, SUBE1, SUBE2, SUBE3, JAMM, BIDET, TIMMY, and PRIOR. LOCTIM in turn calls GETIMX on entry and exit if X(55) is set.

LOCTIM first determines raider velocity and the time of the raider position stored in the RAID table. The distance the raider travels between the present time and the stored time (DL) is computed. If the velocity is zero, present target position is set to the initial position and the program exits. The distance between the stored position and the end of game position (D) is determined and the ratio DL/D is computed. The X,Y position at the time desired is determined by:

$$XP = XZ - \Delta X * DT$$

$$YP = YZ - \Delta Y * DT$$

where

XP is the X position at the time of interest

YP is the Y position at the time of interest

XZ is the stored X position

YZ is the stored Y position

ΔX is the distance between XZ and the end of game position

ΔY is the distance between YZ and the end of game position

DT is the ratio DL/D

If the input flag IND has been set to 5, the new time and position are stored in the RAID table, otherwise the subroutine exits.

27. SUBROUTINE ASMTIM

ASMTIM computes the ASM/SSM position at time T. It is called by SUBE3, TIMMY, and PRIOR. ASMTIM calls GETIMX on entry and exit if X(55) is set.

The logic of ASMTIM is identical to that of LOCTIM but the definition of variables has been changed.

XZ becomes the ASM/SSM X launch coordinate

YZ becomes the ASM/SSM Y launch coordinate

ΔX becomes the distance between the launch position and the X coordinate of the target ship

ΔY becomes the distance between the launch position and the Y coordinate of the target ship

The ASM/SSM position is now computed in the same manner as in LOCTIM and the subroutine exits. The positions are not saved by ASMTIM.

28. SUBROUTINE RANDOM

Subroutine RANDOM is a generator of Uniform (0.0, 1.0) random numbers. It was programmed by Gerald P. Learmouth, of the W.R. Church Computer Center, Naval Postgraduate School. The following is taken from the library description of the subroutine:

"The basic generator is a Lehmer congruential of the form:

X(N) = A*X(N-1) MOD (P)

where A = 7**5 = 16807 and P = (2**31)-1. The resulting X(N)'s are uniformly distributed random integers between 1 and P. The uniform REAL*4 numbers are formed by right-shifting the integer X(N) and appending a proper floating point exponent to form a number between 0.0 and 1.0."

29. SUBROUTINE HISTG

Subroutine HISTG is found in the subroutine library of the Naval Postgraduate School subroutine library. A copy of HISTG has been included at the end of the program. The subroutine is used if the user wants a statistical display of the number of hits a certain specified BLUE unit will take. Also provided is a statistical display of the time interval over which the RED missiles arrive, and one of the hits per minute. A histogram of the selected parameter is provided in all cases. HISTG is called by MAIN just prior to run completion if X(85) has been set and the conditions detailed in section V.B have been met.

30. SUBROUTINE RAIDIS

This subroutine provides a plot of the raider tracks from their initial position to launch position. The maximum size of the plot is 800 nm from force center. In the case where a raider does not launch weapons, the 30 minute game time position is used as the terminal point of the track. If the launch or 30 minute position is more than 800 nm from force center, a label of "no launch" is displayed at the raider's initial position.

31. SUBROUTINE DISPL

DISPL is similar to RAIDIS except that the BLUE unit positions are plotted and they must be within 200 nm of force center. A 100 nm range circle is included in the display.

32. SUBROUTINE SETIMX

This subroutine sets the internal clock to zero. The entry GETIMX gives the internal clock time when called.

33. FUNCTION RADIAN

The function RADIAN converts degrees to radians. It is called by MAIN, JULIE, and TIMMY. The conversion is done with:

RADIAN = RD * 0.017453293

where RADIAN is the output angle in radians and RD is the input angle in degrees.

APPENDIX C

RADAR DETECTION RANGE INPUTS

This appendix provides the equations used to determine the β and α inputs for the individual radar classes.

1. CLEAR ENVIRONMENT

β is the range at which a one square meter target can be detected in a non-jamming environment. It is a constant of the radar system and in the monostatic case provides the range of detection for the radar by the relation:

Range = $\beta(\sigma)^{\frac{1}{4}}$ where σ is the target's radar cross section.

The general equation for β is:

$$\beta = \left[\left(\frac{P_t}{P_m}\right)\frac{G^2\lambda^2}{(4\pi)^3}\right]^{\frac{1}{4}}$$

where P_t is the power of the transmitter in watts; G is the antenna gain coefficient; λ is the radar wave length in meters; and P_m is the minimum detectable signal of the radar with appropriate consideration for probability of detection and propagation losses. If the above units are used in calculating β, the result will be a detection range in meters which must be converted to nautical miles for use in the simulation. Values of β, in nautical miles for a one square meter target, are reported in Ref. 9 for standard U.S. Navy radar systems.

In the case of the bistatic system, the equation for β becomes

$$\beta = [(\frac{P_t}{P_m}) \frac{G_t G_r \lambda^2}{(4\pi)^3}]^{\frac{1}{4}}$$

where the possibility of different gains of the transmitter and receiver must be considered and the propagation losses in P_m must take into consideration the differences in the transmitter to target path and the target to receiver path. Additionally, in the bistatic situation, the detection range is not only a function of the range from the receiver (R_r) to the target but it is also a factor of the range from the transmitter (R_t) to the target. Thus the bistatic range equation becomes

$$R_t R_r = \beta^2 \sigma^{\frac{1}{2}}$$

The monostatic radar cross section provides a suitable estimate for the bistatic radar cross section [Ref. 10, p. 592].

2. JAMMING ENVIRONMENT

α is the range at which a one square meter target can be detected in a jamming level of one watt/mhz. The burn through range for a monostatic radar in a multiple-standoff jamming environment is:

$$R_{bt} = \alpha^{\frac{1}{2}} (\sigma/SJP)^{\frac{1}{4}}$$

where σ is the target's radar cross section and SJP is the sum of the jamming power arriving at the radar. The jamming power from an individual jammer on the radar's frequency is

a function of the power of the jammer; the side-lobe/main beam relationships between the jammer and the radar; and the square of the range from the jammer to the radar. The equation is:

$$SJP = \sum_i C_i JP_i / R_i^2$$

where C_i is the side-lobe coefficient for jammer i and the radar; JP_i is the jamming power of jammer i on the radar's frequency; and R_i is the range from the jammer to the radar. α is found by the relation:

$$\alpha = \left(\frac{P_t G_t^{\frac{1}{2}} G_r^{\frac{1}{2}}}{4\pi B_n (S/J)}\right)^{\frac{1}{2}}$$

where P_t is the power of the transmitter; G is the gain of the radar (In bistatic systems G is the square root of the product of the transmitter and receiver gains.); B_n is the receiver bandwidth of the radar; and S/J is the signal to jamming ratio for the desired probability of detection and appropriate propagation losses. In the bistatic range equation in a multiple standoff jamming environment, the burn through range is found in a manner similar to the clear environment to be

$$R_{bt} R_t = \alpha(\sigma/SJP)^{\frac{1}{2}}$$

where R_{bt} is the burn through range, R_t is the range from the bistatic transmitter to the target and the remaining factors are as defined above.

APPENDIX D

PROGRAM ARRAY LISTING

This appendix provides an alphabetical listing of the principal arrays used in the computer program. The listing contains the values of the arrays for the indicated array arguments. The arguments for the arrays are specified in the heading of their respective listing. The dimensions of the individual arrays are stated in the COMMON statement at the beginning of each subroutine.

ASMT (I,J)

The ASMT array contains data concerning the RED missiles.

I AMS/SSM launch sequence number

J=1 ASM/SSM status code as follows:

0 = Launch failure
1 = Successful launch
2 = Killed by SAM
3 = Inflight failure
4 = Hit target ship
5 = Launch platform shot down or sunk prior to launch

J=2 Time of impact on target ship

J=3 Raid number of launch platform

J=4 Target ship number

J=5 Time of launch

J=6 Weapon class

J=7 X coordinate of launch position

J=8 Y coordinate of launch position

CHANEL (I,J,K,L)

An array used to keep track of which channels are busy.

I   Ship number

J   System number

K   Director number

L   Channel number

0 = Free
1 = Busy

JACK (I,J)

This array stores the engagement and detection statistics for individual ships.

I   Ship number

J=1 Chances to engage. The number of targets that will enter within the maximum missile range of the ship

J=2 Number of engagements

J=3 Number of targets destroyed

J=4 The number of raiders that will enter the missile envelope.

J=5 The number of raiders that will enter the envelope and are detected

J=6 The number of ASM/SSM that will enter the envelope

J=7 The number of ASM/SSM that will enter the envelope and are detected

NADAR (I,J)

This array indicates the operational status of the onboard BLUE equipment.

I   Ship number

J=1 Search radar number 1

0 = down
1 = up

J=2 Search radar number 2

J=3 Search radar number 3

J=4 Battery one, fire control radar number 1

0 = down
1 = operational normal mode
2 = no director needed for missile
3 = operational in horizon scan mode

J=5 Battery one, fire control radar 2

J=6 Battery one, fire control radar 3

J=7 Battery two, fire control radar 1

J=8 Battery two, fire control radar 2

J=9 Battery two, fire control radar 3

J=10 Battery three, fire control radar 1

J=11 Battery three, fire control radar 2

J=12 Battery three, fire control radar 3

J=13 Battery four, fire control radar 1

J=14 Battery four, fire control radar 2

J=15 Battery four, fire control radar 3

J=16 Launcher for battery one

0 = non operational
1 = operational

J=17 Launcher for battery two

J=18 Launcher for battery three

J=19 Launcher for battery four

RAID (I,J)

This array stores information on the raider's location and weapons.

I	Raider number
J=1	X vector starting index of the raid class
J=2	Time that raid was at position indicated in J=3,4
J=3	X coordinate at time T
J=4	Y coordinate at time T
J=5	X coordinate at end of game
J=6	Y coordinate at end of game
J=7	X coordinate at launch if the raider has weapons and can reach the launch position, otherwise the X coordinate at end of game.
J=8	Y coordinate at launch, same as J=7
J=9	Raider altitude in feet
J=10	Raider's radar horizon
J=11	Weapon class of raider's longest range weapon
J=12	Total number of weapons assigned
J=13	Status 0 = destroyed or sunk N = number of hits the raider may take until destroyed
J=14	X coordinate of raider at end of game or X coordinate of target ship for ASM/SSM launched by raider I
J=15	Y coordinate comparable to J=14
J=16	Time raider will reach second launch point if it is carrying two types of missiles
J=17	Time the raider was destroyed

SAMASM (I,J)

This array contains information covering the BLUE missile launch, target, engagement outcome, and probability of kill.

I	BLUE salvo number
J=1	Target number, 1-90 are raiders, 91-191 are ASM/SSM
J=2	Ship number followed by system number
J=3	Director number followed by channel number
J=4	Time of launch
J=5	Engagement outcome. Negative number indicates an overkill, zero a failure, and a positive number a successful shot.
J=6	Time the channel will be free for reassignment
J=7	Single shot probability of kill

SHIP (I,J)

The SHIP array contains information concerning the BLUE units.

I	Ship number
J=1	Ship class
J=2	X coordinate of ship position
J=3	Y coordinate of ship position
J=4	Number of hits ship can receive until declared out of action. If 0, ship is out of action.
J=5	Ship class starting index
J=6	Missile status code 0 = no missile capability 1 = missile capability

SMAT (I,J)

The SMAT array lists the results of the SAM firings.

I	SAM number
J=1	SAM class index
J=2	Firing ship number
J=3	Number of missiles remaining in the battery firing the SAM
J=4	Time fire control radar channel 1 free
J=5	Time fire control radar channel 2 free
J=6	Time fire control radar channel 3 free
J=7	Time fire control radar channel 4 free
J=8	Time fire control radar channel 5 free
J=9	Time fire control radar channel 6 free
J=10	Number of SAMs in the salvo

STATS (I,J)

The STATS array is used to compile statistics concerning the BLUE units.

I	Ship number
J=1	Sum of the detection ranges on raiders
J=2	Sum of the detection ranges on missiles

APPENDIX E

GLOSSARY FOR PROGRAM VARIABLES

1. MAIN PROGRAM

ALFA	REAL*8 variable used for conversion of degrees to radians.
AMF	Small real constant used for conversion from floating point variables to integer variables.
ASMT	The table containing RED ASM/SSM launch and targeting information.
BWJ	Beam width of the jammer.
BWR	Beam width of the radar receiver.
HISA	Vector used to store data for use with subroutine HISTG.
HISO	Vector used to store data for use with subroutine HISTG.
HISTG	Subroutine in the NPS computer library that provides a histogram display of an input vector.
HISX	Vector used to store data for use with subroutine HISTG.
IBAT	An array used to keep track of the number of missiles remaining in each missile battery of a BLUE unit.
IFAG	Integer constant containing the number of RED missiles engaged by the BLUE fleet.
IFBG	Integer constant containing the number of RED raiders engaged by the BLUE fleet.
IHIT	Vector containing the number of hits received by each BLUE unit.
IIF	The number of RED missiles that failed in flight.
IKS	The number of RED missiles shot down by BLUE SAMs.
ILF	The number of RED missiles that failed when launched.

ILS	The number of RED missiles that were in flight at the end of the iteration.
IMF	The number of RED missiles destroyed when the launching aircraft or ship was destroyed.
IMK	The number of RED missiles destroyed by SAMs over all iterations.
IMM	The number of RED missiles destroyed prior to launch over all iterations.
INSECT	The original seed value used with the random number generator. Used to reproduce the random number chain if X(54) is set to 1.
IRT	The starting address in the X vector for the next five input quantities.
ISAG	An array containing the number of RED missiles first engaged by each BLUE ship.
ISBG	An array containing the number of RED raiders first engaged by each BLUE ship.
ITS	The number of RED missiles that impacted on their target.
JACK	The array containing the BLUE ship detection and engagement counters.
KOUNT	The iteration number.
MARY	The vector containing the data for the hit frequency histogram.
NADAR	The array that stores the BLUE radar and missile system availability for each iteration.
NASM	The number of ASM/SSMs.
ND	Index used to convert true bearings to cartesian bearings.
NI	The number of iterations to be performed.
NMIF	The number of missiles in the BLUE fleet.
NR	The number of armed and unarmed RED raiders.
NRTGT	The vector containing the RED units target list assignment.

NS The number of BLUE ships.

NSAM The number of BLUE missile salvos fired.

PLIST The array containing the target lists.

RAID The table containing the RED raider position and weapon information.

RAT REAL*8 variable used to convert degrees to radians.

RXATA Dummy array used to read in data and fill the X vector.

SAMASM The array containing the BLUE missile RED target engagement information.

SETIMX The subroutine that sets the internal clock if X(55) is set to 1.

SHIP The table containing the BLUE ship position information.

SMAT The table containing the BLUE missile battery engagement data.

STATS The array containing the BLUE unit detection ranges.

T Time.

2. SUBROUTINE EFFECT

IUNIT The code containing the units involved in playing the next event.

NEXTEV The next event type.

TIME The time the next event is to be played.

TIMES The internal clock time used for entry and exit if X(55) is set.

3. SUBROUTINE JULIE

BNG A REAL*8 variable used to convert degrees to radians.

CHANEL The array that keeps track of which BLUE missile channels are busy.

IA The X vector address of the fire control radar class number.

IBAT	The array that keeps track of the number of BLUE missiles remaining in each battery.
IC	The radar class of the fire control radar associated with the number one director of the indicated BLUE missile battery.
IFAG	Integer constant containing the number of RED missiles engaged by the BLUE fleet.
IFBG	Integer constant containing the number of RED raiders engaged by the BLUE fleet.
IFIRST	Flag used for checking a target's priority.
IFLASK	Flag used to set the time by which all RED raiders must reach their launch point in order to launch their missiles.
IFOG	Flag to indicate a) horizon search mode, and b) the system has a launcher that is operable.
IGAGE	A vector that keeps track of which targets have been engaged by BLUE missiles.
IHIT	A vector that keeps track of how many hits each BLUE unit has taken.
IJ	The ship class starting index in the X vector.
IOR	The raid class starting index in the X vector.
IP	Missile system class number.
IS	The ship class starting index in the X vector.
ISAG	Integer constant to determine how many RED missiles the ship was the first to engage.
ISBG	Integer constant to determine how many RED raiders the ship was the first to engage.
IT	Search radar class number.
ITEMS	The number of events that have been retrieved from the events list.
IVENTS	The array containing the events list.
JACK	The array containing the ship detection and engagement statistics.

JAM	The vector that keeps track of the number of jammers operating on a given radar frequency band.
KFOG	Flag to indicate that a search radar is available.
LOVER	The number of events in the events list.
MAMBAT	The number of missile batteries available to the BLUE force.
MISLFT	The array that keeps track of how many missiles a RED raider has left on board.
NADAR	The array that keeps track of the operational mode of the BLUE equipment.
NASM	The number of RED missiles.
NONO	The array that keeps track of engageability of a RED unit or missile with respect to each BLUE unit.
NR	The number of armed and unarmed RED raiders.
NS	The number of BLUE units.
NSAM	The number of BLUE missile salvos.
PROB	The probability that a given piece of equipment is operational.
RAID	The table with the raider location and weapon information.
RELOAD	The array that stores the time the launcher will be reloaded by after a missile firing.
RGE	The launch range of the longest ranged missile carried by each raider.
SHIP	The table containing the ship position information.
SMAT	The table with the BLUE missile system launch information.
STATS	The table of BLUE detection ranges.
TGTAV	The table that tells when a target will be available for engagement by another BLUE unit.

4. SUBROUTINE EVENT

ICODE	The participating units-event type code that is stored in the events list.
ID	The event type.
IT	The time of the event converted to the integer format used to store it in the events list.
ITB	Used to determine where in the events list the next event to be stored should go.
ITEMS	The number of events retrieved.
ITIM	The time as stored in the events list.
ITM	The time as stored in the events list.
IUNIT	The units involved in playing the next event.
IVENTS	The events list.
IVET	The participating units as stored in the events list.
LOVER	The number of events stored in the events list.
NEXTEV	The next event type.
NUMBER	The code for the participating units.
T	The time of the event.
TIME	The time of the event.

5. SUBROUTINE REMOVE

IUNYT	The code stored in the events list of the event to be removed from the events list.
IVENTS	The events list.
LOVER	The number of items in the events list.

6. SUBROUTINE TARGET

IFIRST	The flag that indicates that the counter has been zeroed.
IT	The game time in events list code.

IUNIT The code of the event to be compared by PRIOR.

IVENTS The events list.

MORE The flag indicating that there is an event with the same playing time and of type three.

NPGS The counter that is used as an index in searching the events list.

TYME Present game time.

7. <u>SUBROUTINE SUBE1</u>

ALT The raider's altitude.

AREA The fourth root of the raider's radar cross section.

DELR The detection delay.

FCRH The radar horizon of the fire control radar.

FRG The minimum of the clear environment detection range, the radar horizon, and the radar's theoretical maximum detection range.

HFCSR The radar horizon of the fire control radar.

HORZ The combined radar horizons of the radar and the target.

HRAG The combined radar horizons of the radar and the target.

HRFC The combined radar horizons of the radar and the target.

HSFR The minimum of the clear environment detection range, the radar horizons, and the maximum theoretical detection range.

HT The height of the fire control radar antenna.

IA Used to check the class numbers of the various systems.

IC Missile system class number.

ICH Used to enter the NADAR table.

ICL Fire control radar class number.

ICPA	Flag telling whether the target will come within the maximum detection range.
IFA	Integer to store the radar number.
IFC	Integer to store the battery number.
IFCR	Fire control radar class number.
IF	Radar frequency band.
IR	The raid number.
ISI	The ship class starting index in the X vector.
JACK	The table of engagements and detections.
JAM	The table that keeps track of how many jammers there are on each radar frequency band.
JAMM	The subroutine for determining the detection range of a monostatic radar in a jamming environment.
JCPA	The flag that tells whether or not the target will come within the detection range.
JFOG	The flag that tells whether or not this is the second bistatic detection zone.
KEY	The type of probability distribution associated with this unit's time delays.
KFLAG	Flag that says that this unit has a search radar that is operational.
LCPA	Flag that tells whether or not a target will close to within the detection range.
MFOG	Counter giving the number of second bistatic detection zones to be processed.
MJL	Index for table lookup.
MTY	Missile system class number.
NADAR	The table containing the equipment availability.
NASA	The flag indicating that raider is an aircraft.
NASM	The number of ASM/SSM launched or destroyed.
NB	The battery number.
ND	The number of bistatic detection zones.

NF	The director number.
NFOG	Flag telling whether or not there are any more bistatic detection zones to check.
NJ	The number of jamming raiders.
NONO	The table that contains the reason for non-engageability.
NSR	The search radar index.
NSURF	Flag that says that this ship has a surface to surface missile system.
PROB	Probability of acquisition.
RAGJ	Detection range in a jamming environment.
RAID	The table with raider location and weapon information.
RCPS	Range at CPA squared.
REACT	Subroutine for determining reaction time delays.
RFCJ	Fire control radar detection range in the jamming environment.
RG	The minimum of the clear environment detection range, the radar horizon range, and the theoretical maximum detection range.
RGE	Detection range.
RGF	Jamming environment detection range.
RGFC	Detection range of the fire control radar.
RGI	Detection range in a jamming environment.
RGIA	Storage for second bistatic detection range.
RGIB	Storage for second bistatic detection range.
RGIC	Storage for second bistatic detection range.
RGM	Missile range.
RIP	The maximum detection range, or present target position, whichever is smaller.
RRFC	Clear environment fire control radar detection range.

RRH	Radar horizon of the search radar.
R2	The detection range in the second bistatic detection zone.
SALT	The radar horizon of the raider.
SAMVEL	The BLUE missile's speed.
SHIP	The table with the ship position information.
SIGMA	The raider radar cross section.
SRAG	Missile range.
SRFC	Fire control radar clear environment detection range.
STATS	Ship's detection range storage.
SVEL	Missile speed.
TD	Time raider will close to the given range.
TDES	The designation time delay.
TDET	The time of detection.
TD2	The time of detection in the second bistatic detection zone.
TFIR	The last time to fire.
TFIRE	The firing time delay.
TFR	Time the target will close to fire control acquisition range.
THR	Theoretical radar maximum detection range.
THSCAN	Horizon scan detection delay.
TI	Time the target will close to the given range.
TIME	Present game time.
TIML	Time the target will go beyond the given range.
TIMLA	Storage for lose detection time.
TIMLB	Storage for lose detection time.
TIMLC	Storage for lose detection time.

TIML2 Time of lose detection in the second bistatic detection zone.

TLS Last time to fire.

TRGE Time of detection.

TSHOOT Time of launch.

TSR Total reaction delay.

TTA Storage for detection time in second detection zone.

TTB Storage for detection time in second detection zone.

TTC Storage for detection time in second detection zone.

TWDS Reaction time delay for weapons designation equipment.

VEL Raider speed.

XCP The X coordinate of the CPA.

YCP The Y coordinate of the CPA.

8. SUBROUTINE SUBE2

ALT Altitude of the RED ASM/SSM during flight in feet.

ALTN2 Altitude of the RED ASM/SSM squared in miles.

AREA The fourth root of the radar cross sectional area of the RED missile.

ASMT The table containing RED missile launch and targeting information.

DELR The detection time delay.

DELTAX The distance in the X direction between the target ship and the ASM/SSM launch point.

DELTAY The distance in the Y direction between the target ship and the ASM/SSM launch point.

DIST The three dimensional distance from the ASM/SSM launch point to the target ship. This distance assumes launch at cruising altitude.

HORZ The combined radar horizons of the firing ship and the RED missile.

HRAG — The combined radar horizons of the firing ship and the RED missile.

HT — The BLUE radar antenna height.

IA — Fire control radar class number.

IC — The BLUE missile system class number.

ICL — Fire control radar class number.

ICOP — Constant for X vector entry set by SUBE2.

ICPA — Flag telling whether or not a RED missile will come within the detection range.

IDP — Index counter used to make up the RED raider's temporary target list.

IFA — The director number.

IFC — The battery number.

IFCR — BLUE missile system class number.

IFLAG — Flag indicating RED unit is firing a two missile salvo.

IFLASK — Flag indicating the first air launch has occurred, set TMAX.

IHIT — Table of hits taken by BLUE ships.

IJ — Raider class number.

IJI — Radar frequency band.

IOU — Target ship number.

IPQ — Target hit number.

IR — Raider number.

ISI — Ship class starting index in the X vector.

JACK — Ship detection and engagement table.

JAM — Table that keeps track of the number of jammers on the radar frequency bands.

JAMM — Subroutine to determine monostatic radar detection ranges in a jamming environment.

JFLAG	Flag indicating that there are more second bistatic detection zones to evaluate.
JFOG	Flag indicating that there were two bistatic detection zones.
KATHY	Flag indicating that a second type ASM/SSM is being processed.
KEY	Type of probability distribution to be used in computing the reaction time delay.
KFLAG	Flag indicating BLUE unit has an operational search radar.
KING	Flag for one or two missile salvos.
KQED	Flag indicating whether or not raid could target a particular BLUE unit.
LNUM	The class number of the raider's long range missiles.
MARIE	Number of hits required to put the BLUE unit out of action.
MFOG	Counter indicating the number of second bistatic detection zones.
MISLFT	Array to indicate the number of ASM/SSM left on board a raider.
NADAR	The array that stores BLUE equipment operability.
NASM	The number of ASM/SSMs processed.
ND	The number of bistatic detection zones.
NEXT	Flag indicating the short range RED missiles.
NFOG	Flag indicating there are more detection zones to process.
NJ	The number of jammers.
NMIS	The number of ASM/SSM of this type to be launched.
NONE	The index used to draw random numbers from the random number vector.
NONO	The array specifying the target-firing ship relationship.
NOW	The type missile being processed now.

NR	The number of raiders.
NRTGT	The target list assignments.
NS	The number of BLUE units.
NTOT	The total number of ASM/SSM on board.
PLIST	The target lists.
PROB	The probability of acquisition in horizon scan mode of operation.
QRT	The vector of (0,1) random numbers used to determine missile launch times.
RAGJ	The search radar detection range in a jamming environment.
RAID	The table with the raider location and weapon information.
RCPS	The slant range to CPA.
RELI	The RED missile's inflight reliability.
RFCJ	Detection range of fire control radar in a jamming environment.
RG	The minimum of the clear, jamming, and theoretical detection ranges.
RGE	Detection range.
RGF	Detection range.
RGFC	Fire control radar detection range.
RGI	Detection range.
RGIA	Storage for secondary bistatic detection range.
RGIB	Storage for secondary bistatic detection range.
RGIC	Storage for secondary bistatic detection range.
RGM	BLUE missile range.
RGS	Detection range squared.
RIP	Minimum of present range and detection range.
RRFC	Horizon search clear environment detection range.

RRH	Radar horizon.
RT	Cumulative probability of targeting this target or one previous.
R2	Second bistatic detection range.
SALT	The radar horizon of the RED weapon.
SAMVEL	BLUE missile speed.
SHIP	The table containing ship position information.
SIGMA	The RED weapon radar cross sectional area.
SPD	The RED weapon speed.
SRAG	Search radar detection range in a clear environment.
SRFC	Fire control radar detection range in a clear environment.
SRGE	BLUE ship missile range.
STATS	Array containing the sum of a ship's detection ranges.
SVEL	BLUE missile speed.
TAC	Acquisition time delay.
TD	Time of detection.
TDES	Designation time delay.
TDET	Time of detection.
TD2	Time of detection in the second bistatic detection zone.
TFIR	Last time to fire based on search radar range and maximum range missile.
TFIRE	Firing time delay.
TFR	Time target will come within the fire control detection range.
THR	Maximum theoretical radar range.
THSCAN	Reaction time delay for horizon scan mode of operation.

TI	Time target will come within the search radar detection range.
TIFA	Time target will come within the horizon scan detection range.
TIME	Present game time.
TIMF	Time target will exit the fire control radar detection range.
TIML	Time the target will go beyond the search radar detection range.
TIMLA	Storage for second bistatic detection zone loss of target time.
TIMLB	Storage for second bistatic detection zone loss of target time.
TIMLC	Storage for second bistatic detection zone loss of target time.
TIML2	Time target will go beyond detection range in the second bistatic detection zone.
TINT	The time interval over which all missiles must be launched.
TLS	Last time to fire based on fire control radar range.
TM	Proposed RED missile launch time.
TMAX	Time limit on launch of RED missiles.
TRGE	a) Hazard zone range squared, b) time of detection.
TSHOOT	Time of proposed BLUE missile launch.
TSR	Total reaction time delay.
TTA	Storage for second detection zone detection time.
TTARG	The temporary target list.
TTB	Storage for second detection zone detection time.
TTC	Storage for second detection zone detection time.
TWDS	Delay for weapons direction equipment processing.
TX	Time between RED missile launchings.
VEL	RED missile speed.

XCPA	CPA position in the X direction.
YCPA	CPA position in the Y direction.
Z	(0,1) random number.

9. <u>SUBROUTINE SUBE3</u>

ASMT	The table containing the RED missile launch and targeting information.
CHANEL	The array that indicates which system channels are in use.
EBTJR	The Y coordinate of CPA.
FT	BLUE missile flight time.
IBAT	Array giving the number of BLUE missiles available by battery.
ICOT	Flag indicating channel assigned.
IP	Ship class starting index in the X vector.
IU	Input code giving target, ship, system, and director information.
JAPL	A channel number.
JULIA	RED missile number for table lookup.
K	Firing ship number.
KA	Target ship number.
M	Number of original target.
MFOG	Flag used to check director scheduling.
M1	Target number after searching PRIOR.
NADAR	Array to check launcher operability.
NB	Director number.
NC	Channel number.
NCH	Number of channels associated with this radar.
NC1	Channel number.
NMS	Missile system number.

NSAM	BLUE salvo number.
PROB	Probability of kill.
RAID	Table of raider positions.
REASON	Code for non-engageability.
RELOAD	Table that tells if the launcher is loaded.
RGE	Systems maximum range.
RJL	The X coordinate of CPA.
RLT	Launcher reload time.
SAMASM	The table of BLUE missile and RED target engagement data.
SHIP	Table of ship positions.
SMAT	Table of BLUE missile system data.
TCF	Dummy vector for storing the times the channels will be free for other use.
TFT	Time director first available for channel use.
TF	Time director needed.
TGTAV	Time target available for engagement by another unit.
TIM	Time channel free.
TIME	Present game time.
TL	Time director no longer needed.
TT	Time director needed for homing.
XP	Present target X coordinate.
YP	Present target Y coordinate.

10. <u>SUBROUTINE SUBE4</u>

ID	Used as a class indicator, i.e. ASM type, or as an integer constant, i.e. number of hits to sink the target ship.
IHIT	The table containing the number of hits sustained thus far in the game.

IU The input code containing the RED missile number and the BLUE target ship number.

K The target ship number.

NAM The RED ASM/SSM number for table lookup.

NAS The RED ASM/SSM number in the input code.

PB The probability of successful flight for the RED missile. Also used in determining the input and output time of the subroutine if X(55) is set.

11. SUBROUTINE SUBE5

ASMT The table containing the data on the RED missile engagement.

CHANEL The table that keeps track of which BLUE missile channels are being used.

ID An integer constant used for the starting address in the X-array of the BLUE firing ship's class inputs. Also used as the number of RED ASM/SSMs launched up to the present game time.

IFAG Integer constant that totals up the number of RED missiles engaged by the BLUE fleet.

IFBG Integer constant that totals up the number of RED raiders engaged by the BLUE fleet.

IGAGE Integer constant that keeps track of which targets have been engaged.

IP Radar class of the BLUE fire control radar associated with the engagement.

IQ BLUE missile system class number.

ISAG Integer constant that totals up the number of RED missiles first engaged by this firing ship.

ISBG Integer constant that totals up the number of RED raiders first engaged by this firing ship.

JACK The table that keeps track of the number of missiles and raiders engaged by the firing ship.

JAM The table that totals up the number of jammers operating on a particular radar frequency band.

JFK	Integer used in table lookup.
K	The BLUE firing ship.
KEY	As used in the statement -- CALL REACT (KEY,4,TA) -- where 4 is the type of delay, KEY is the class of the particular distribution within the type, and TA is the delay.
M	The BLUE salvo number.
MISLFT	The table that keeps track of the number of missiles remaining on board the RED raider.
N	The number of missiles in the BLUE salvo.
NA	The RED ASM/SSM number for table lookup.
NASM	The number of RED ASM/SSMs processed thus far.
NB	The BLUE ship's director involved in the engagement.
NC	The channel number of the fire control system engaged in the intercept.
NDU	The starting address of the class associated with the raider involved in the intercept.
NMS	The number of the ship's missile system involved in the intercept.
NT	The target number, 1-90 for a raider, 91-190 for a RED missile.
NTOT	The number of missiles left on board the raider.
PB	The probability that the BLUE missile will receive the proper support during flight.
PROB	The overall probability of a successful intercept.
SAMASM	The table that keeps BLUE salvo information.
SMAT	The table that keeps BLUE system information.
TA	The time delay for target acquisition.
TF	The time delay for firing.
TK	The time delay for kill assessment.

12. <u>SUBROUTINE SUBE6</u>

CHANEL	The table that keeps track of which BLUE missile channels are being used.
K	The firing ship.
M	The input code containing the ship, system, director, and channel number to be released.
NB	The director number.
NC	The channel number.
NMS	The system number.
T	The input and output time for the subroutine, used if X(55) is set.

13. <u>SUBROUTINE TIMMY</u>

A	The cross range factor in the intercept contour.
ALPHA	The angle between the target flight path and the line of sight to the target.
ALT	The target altitude.
ASMT	The table containing the data on the RED missile engagement.
ASMTIM	Subroutine to determine RED missile position at a given time.
B	The downrange factor in the intercept contour.
BETA	The angle between the BLUE missile flight path and the target's direction of movement.
CS	The target course.
DELTA	The reciprocal of BETA.
DELTAX	Used for a) the X distance from the target ship to the RED missile, b) the present distance in the X direction from the firing ship to the target, and c) the distance in the X direction from the firing ship to the intercept point.
DELTAY	Same as DELTAX but in the Y direction.
DIS	Range to the intercept point.

DLT	Range to the intercept point.
DX	Distance in the X direction from the initial target position to the intercept point. Used if intercept point is over the firing ship.
DY	Distance in the Y direction from the initial target position to the intercept point. Used if the intercept point is directly over the firing ship.
EX	The exponent that defines the super ellipse of the intercept contour.
FT	BLUE missile flight time.
FZL	Left hand limit of the firing zone.
FZL1	Left hand limit of the second firing zone if a second exists.
FZR	Right hand limit of the firing zone.
FZR1	Right hand limit of the second firing zone.
GAMMA	Angle between the BLUE missile flight path and the line of sight to the target at launch.
GSINE	The sine of GAMMA.
ID	RED missile type.
IND	Flag indicating the feasibility of the intercept.
IR	RED missile's launching raid number.
JOAN	RED missile's target ship number.
JULIA	RED missile number for table lookup.
K	Firing ship number.
KL	Starting index for the BLUE missile class.
KR	Raider class number.
LOCTIM	Subroutine for determining raider position at a desired time.
M	Target number.
NMS	BLUE missile system number.
NS	The number of ships in the BLUE force.

PHI	The BLUE firing bearing.
PHI1	The angle of axis rotation for intercept contour engagement evaluation.
PI	3.141592654.
PIZ	The hazard range squared.
PK	The single shot kill probability.
PM	The probability of successful launch.
PROB	PK times PM.
RADIAN	A function for converting angles in degrees to angles in radians.
RAID	The table with the raider position and weapon information.
RAT	A REAL*8 variable for use with RADIAN.
REASON	A code to indicate the reason for non-engagement.
RMAX	The maximum range of the BLUE missile.
RMIN	The minimum range of the BLUE missile.
RUTH	Target altitude in miles.
RXATA	Dummy vector use to determine the firing zone.
SEEK	Y intercept of the intercept contour seeker limitation approximation.
SLOPE	The slope of the line that approximates the seeker look limitation.
SSPD	Speed of the BLUE missile.
THETA	The angle, measured in the normal X,Y plane, of the line from the firing ship to the present target position.
TH1	The reciprocal of THETA.
TIME	The time of the proposed missile launch.
TKIL	The time of intercept.
TX	The time the target will enter the maximum missile range for this system.

TXL	The time the target will depart the maximum missile range for this system.
VEL	The speed of the target.
XI	The X coordinate of the intercept.
XIP	The X intercept coordinate after axis rotation and translation.
XIS	The X intercept coordinate after axis translation.
XP	The X coordinate of the present target position.
XT	The end of game X coordinate for the target if it is a raider, or the target ship coordinate if the BLUE missile's target is a RED ASM/SSM.
XW	The initial position of the target in the X direction.
YI	The Y coordinate of the intercept point.
YIP	The Y intercept coordinate after axis rotation and translation.
YIS	The Y intercept coordinate after axis translation.
YP	The Y coordinate of the present target position.
YT	The end of game Y coordinate for the target.
YW	The Y coordinate of the initial target position.
ZLNKA	A sorting subroutine.

14. SUBROUTINE JAMM

AREA	Fourth root of target's radar cross section.
COFF	Jammer's main beam/side-lobe coefficient.
DELX	Target's track change in X direction minus 0.1 nm.
DELY	Target's track change in Y direction minus 0.1 nm.
HJBW	One-half jammer beam width in radians.
HRBW	One-half radar beam width in radians.
ITS	Target ship number.
JJTBR	Bearing from jammer to the jamming target.

JRBRG Bearing from the jammer to the detecting radar.

LJBW Left bearing of the jamming beam.

LTBRG Left bearing of the detecting radar's beam while pointed at the target.

NASM Number of the ASM.

NR Number of raiders.

PJ Power of the jammer on the detecting radar's frequency.

RCOF Detecting radar's side-lobe coefficient.

RCOFF Detecting radar's side-lobe coefficient for specific jammer-target-radar geometry.

RCPS Range to track CPA squared.

RJBRG Detecting radar to jammer bearing.

RJBW Right bearing of jammer beam.

RJHS Detecting radar-jammer horizon squared.

RRJS Range from the detecting radar to the jammer squared.

RTBRG Right bearing of the detecting radar's beam width while pointed at the target.

SUMJP Sum of the jamming power arriving at the detecting radar.

TBRG Target bearing from the detecting radar.

TIMC Time constant used to adjust midpoint time. Base on target's launch time or game time of target's initial position.

TIMI Time target is at its midpoint in detecting radar's detection envelope.

TIMX Time target is at the X coordinate of its midpoint in the detecting radar's detection envelope.

TIMY Time target is at the Y coordinate of its midpoint in the detecting radar's detection envelope.

VD Divisor used in obtaining the X and Y components of the target's velocity.

VX	X component of the target's velocity.
VY	Y component of the target's velocity.
XC	X coordinate of target's CPA to detecting radar.
XCOF	Product of the jammer and detecting radar coefficients based on the jammer-target-radar geometry.
XJ	X coordinate of the jammer when the target is at its midpoint.
XJJ	X coordinate of the jammer at game time 0.
XJJT	Difference in X coordinates of the jammer's position and the jammer's target position.
XJR	Difference in X coordinates of the detecting radar's position and the jammer's position.
XJT	X coordinate of the jammer's target.
XO	X coordinate of the target's terminal position.
XR	X coordinate of the detecting radar's position.
XRT	Difference in Y coordinates of the detecting radar's position and the target's midpoint position.
X1	X coordinate of the target's initial position.
YC	Y coordinate of target's CPA to detecting radar.
YJ	Y coordinate of the jammer when the target is at its midpoint.
YJJ	Y coordinate of the jammer at game time 0.
YJJT	Difference in Y coordinates of the jammer's position and the jammer's target position.
YJR	Difference in Y coordinates of the detecting radar's position and the jammer's position.
YJT	Y coordinate of the jammer's target.
YO	Y coordinate of the target's terminal position.
YR	Y coordinate of the detecting radar's position.
YRT	Difference in Y coordinates of the detecting radar's position and the target's midpoint position.

Y1 Y coordinate of the target's initial position.

Z1 Target altitude in nautical miles.

15. SUBROUTINE BIDET

A Coefficient used in solution of bistatic detection and radar horizon equations. See Appendix A for bistatic equation.

A1 2A.

A2 A squared.

B Similar to A. See Appendix A.

C Similar to A. See Appendix A.

CC Constant used in solution of receiver and transmitter radar horizons equations.

CHAR Constant of bistatic detection equation. See Appendix A.

CHAS Bistatic radar detection range in clear environment against one square meter target.

COF Array used only in calling subroutine DTIMD.

COFF See definition in subroutine JAMM portion of this glossary.

CPRS Range squared from track CPA to bistatic receiver position.

CPXS Range squared from track CPA to bistatic transmitter position.

D Similar to A. See Appendix A.

DELX Same definition as in subroutine CPA.

DELY Same definition as in subroutine CPA.

E Similar to A. See Appendix A.

EE Constant used in the solution of bistatic receiver and transmitter radar horizon equations.

HJBW Defined in subroutine JAMM.

HORRS Receiver's radar horizon range squared.

HORXS	Transmitter's radar horizon range squared.
HRBW	Defined in subroutine JAMM.
IFLAG	Flag set to 1 for stationary targets.
IT	Ship number of bistatic transmitter.
JAM	Array indicating the number of active jammers on each radar frequency band.
JFLAG	Flag set to 1 when the algorithm has completed the jamming section of the detection phase.
JJTBR	Defined in subroutine JAMM.
JRBRG	Defined in subroutine JAMM.
LJBW	Defined in subroutine JAMM.
LTBRG	Defined in subroutine JAMM.
NASM	ASM number.
PJ	Defined in subroutine JAMM.
Q	Track change in Y direction divided by track change in X direction.
RCOF	Defined in subroutine JAMM.
RCOFF	Defined in subroutine JAMM.
RD2	Square of the range from the bistatic receiver to stationary target's position.
RJBRG	Defined in subroutine JAMM.
RJBW	Defined in subroutine JAMM.
RJHS	Defined in subroutine JAMM.
RRAD	Radical part of solution to quadratic equation for time above bistatic receiver's radar horizon.
RRH	Range to receiver's radar horizon.
RRJS	Defined in subroutine JAMM.
RTBRG	Defined in subroutine JAMM.
SUMJP	Defined in subroutine JAMM.

T	An array containing the bistatic detection times.
TBRG	Defined in subroutine JAMM.
TIMC	Time constant based on target's launch time of missile. Otherwise start of game time.
TIME	Array containing the real parts to the solution of the bistatic equation.
TIMEI	Array containing the complex parts to the solution of the bistatic equation.
TIMI	Defined in subroutine JAMM.
TR1	Time target enters receiver's radar horizon range.
TR2	Time target departs receiver's radar horizon range.
TX1	Time target enters transmitter's radar horizon range.
TX2	Time target departs transmitter's radar horizon range.
T1	Time target enters bistatic radar detection envelope.
T1O	Time target is first above both transmitter and receiver's radar horizon.
T2	Time target departs bistatic radar detection envelope.
T2O	Time target is no longer above both the transmitter and receiver's radar horizon.
V	Target's velocity in nautical miles per minute.
VD	Defined in subroutine JAMM.
VX	X component of target velocity.
VY	Y component of target velocity.
XCOF	Defined in subroutine JAMM.
XCOFF	Defined in subroutine JAMM.
XD2	Square of the range from the bistatic transmitter to the stationary target's position.
XJ	Defined in subroutine JAMM.
XJJ	Defined in subroutine JAMM.

XJJT	Defined in subroutine JAMM.
XJR	Defined in subroutine JAMM.
XO	X coordinate of target's terminal position.
XR	X coordinate of bistatic radar receiver position.
XRAD	Similar to RRAD above.
XRC	X coordinate of target track CPA to bistatic receiver.
XRT	Defined in subroutine JAMM.
XX	X coordinate of the bistatic transmitter.
XXC	X coordinate of target track CPA to bistatic transmitter.
X1	X coordinate of target's initial position.
YJ	Defined in subroutine JAMM.
YJJ	Defined in subroutine JAMM.
YJJT	Defined in subroutine JAMM.
YJR	Defined in subroutine JAMM.
YJT	Defined in subroutine JAMM.
YO	Y coordinate of target's terminal position.
YR	Y coordinate of bistatic receiver's position.
YRC	Y coordinate of target track CPA to bistatic receiver.
YRT	Defined in subroutine JAMM.
YX	Y coordinate of the bistatic transmitter position.
YXC	Y coordinate of target track CPA to bistatic transmitter.
Y1	Y coordinate of target's initial position.
ZR	Bistatic receiver antenna height in feet.
ZRN	Bistatic receiver antenna height in nautical miles.
ZX	Bistatic transmitter antenna height in feet.

ZXN Bistatic transmitter antenna height in nautical miles.

Z1 Target altitude in nautical miles.

Z12 Target altitude in nautical miles squared.

16. SUBROUTINE CPA

DELX First used as a test variable to determine if track passes through reference position. Second usage is to indicate track change in X direction.

DELY First used as a test variable to determine if track passes through reference position. Second usage is to indicate track change in Y direction.

Q Track change in Y direction divided by track change in X direction.

XC X coordinate of CPA position.

XI X coordinate of initial track position.

XO X coordinate of terminal track position.

XP X coordinate of reference position.

YC Y coordinate of CPA position.

YI Y coordinate of initial track position.

YO Y coordinate of terminal track position.

YP Y coordinate of reference position.

17. SUBROUTINE RCHECK

DELX The distance along the X axis from the raider position to the potential target position.

DELY The distance along the Y axis from the raider position to the potential target position.

DIS The square of the distance from the raider to the potential target.

HT The height of the highest radar on the potential target ship.

HTR The radar horizon for HT.

ID	The class number of the potential target ship.
IR	The raider number.
IT	The potential target ship number.
K	Flag for go/nogo situation.
RADAR	The combined radar horizons of the raider and the potential target.
RADAR2	RADAR squared.
RR	The maximum theoretical range for the raider's radar.
RR2	RR squared.
RTR	Radar horizon for the raider.
SHIP	The table containing the BLUE ship location information.
XS	The raider's present position along the X axis.
XT	The ship's present position along the X axis.
YS	The raider's present position along the Y axis.
YT	The ship's present position along the Y axis.

18. <u>SUBROUTINE TMDASM</u>

A	Coefficient A in the quadratic equation.
B	Coefficient B in the quadratic equation.
B1	B in Y = mX + b
C	Coefficient C in the quadratic equation.
DELTAX	The distance in the X direction from present position to the end of game position.
DELTAY	The distance in the Y direction from present position to the end of game position.
DELX	The distance in the X direction from the intercept position to the present position.
DELY	The distance in the Y direction from the intercept position to the present position.

DEX	The distance in the X direction from the firing ship to the target at present time.
DEY	The distance in the Y direction from the firing ship to the target at present time.
DIC	The distance in either the X or Y direction from intercept to end of game.
DIF	The distance in either the X or Y direction from present position to end of game position.
D1	Distance from present position to first intercept point, squared.
D2	Distance from present position to second intercept point, squared.
IFLAG	Flag indicating target is already within the indicated range.
IRAID	Raider number of target or launching raid.
ISAMBT	Battery number.
ITS	Target ship number when target is an ASM/SSM.
KFLAG	Flag indicating target moving in a north or south direction.
M	Raider number.
NFS	Firing ship number.
RAID	The raid array.
RIP	The range to the target or to intercept, whichever is smaller.
ROOT	$B^2 - 4AC$.
SHIP	The ship array.
SLM	m in $Y = mX + b$.
SRG	Range of interest.
TIME	Time target crosses range inbound.
TIML	Time target crosses range outbound.
TML	Present game time.
TOAST	$B^2 - 4AC$.

VEL	Target speed.
XI	Inbound crossing point.
YI	Inbound crossing point.

19. SUBROUTINE TMDBOM

A	A in the quadratic equation.
B	B in the quadratic equation.
BA	b in Y = mX + b.
C	C in the quadratic equation.
DELTAX	The distance from present position to the end of game position in the X direction.
DELTAY	The distance from the present position to the end of game position in the Y direction.
DELX	Distance from the intercept point to present position in the X direction.
DELY	Distance in the Y direction from the intercept point to the present position.
DIC	Distance from the intercept point to the end of game position in the X direction.
DIF	Distance from the present position to the end of game position in the X direction.
IR	Raider number.
RAID	Raid array.
RGE	The launch range.
SLM	m in Y = mX + b.
T	Time raider will reach the launch range.
VEL	The raider speed.

20. SUBROUTINE LOCTIM

D	The distance between the raider end of game position and the position stored in the RAID table.
DELTAX	The distance along the X axis from the end of game position to the position stored in the RAID table.
DELTAY	Distance along the Y axis.
DL	Distance traveled in the time interval defined as the difference between the time stored in the RAID table and the time for which the raider position is desired.
K	The raider number.
PTIME	The time stored in the RAID table.
RAID	The table that contains the position and weapon information.
T	The time for which the raider position is desired.
VEL	The raider speed.
XP	The X position at time T.
XZ	The X position stored in the RAID table.
YP	The Y position at time T.
YZ	The Y position stored in the RAID table.

21. SUBROUTINE ASMTIM

ASMT	The table containing the data on the RED missile engagement.
D	The distance between the missile launch point and the impact point.
DL	The distance the missile will travel in the time defined as the difference between the launch time and the time for which the missile position is desired.
ID	The RED missile type.
M	The RED missile target number.
MA	The RED missile number for table lookup.

T	The time for which the ASM/SSM position is desired.
VEL	The missile speed.
XP	The X position at the time T.
YP	The Y position at the time T.

```
// EXEC FORTCLGP,REGION.GO=350K
//SYSIN DD *
```

MAIN

```
      REAL *8RAT,COSI,SINI,RRK,RRS,BETA,ALFA,GAMA,RRC
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      DIMENSION HISO(100)
C
C   AMF          A SMALL CONSTANT USE TO CONVERT REAL NUMBERS TO
C                INTEGERS
C
C   ASMT         THE ARRAY WITH THE ASM LAUNCH, TARGET, ALTITUDE ETC., DATA
C
C   CHANEL       AN ARRAY USED TO KEEP TRACK OF WHAT AEGIS CHANNELS ARE
C                CURRENTLY ENGAGED IN SAM INTERCEPTS
C
C   CHECK        THE INTERNAL MEASUREMENT OF THREAT
C
C   HISA         A VECTOR USED WITH HISTG
C
C   HISX         VECTOR FOR USE WITH HISTG
C
C   IBAT         THE ARRAY THAT CONTAINS THE NUMBER OF MISSILES LEFT IN
C                EACH SAM BATTERY
C
C   IFAG         KEEPS TRACK OF THE NUMBER OF ASM/SSM ENGAGED
C                BY THE FLEET
C
C   IFBG         SAME AS IFAG, BUT FOR RAIDERS
C
C   IFIRST       THE POINTING INDICATOR USED IN TARGET
C
C   IFLASK       THE FLAG INDICATING THE FIRST AIRLAUNCHED RED
C                MISSILE HAS BEEN FIRED
C
C   IGAGE        THE ARRAY THAT KEEPS TRACK OF WHICH TARGETS
C                HAVE BEEN ENGAGED
C
C   IHIT         KEEPS TRACK OF THE HITS ON INDIVIDUAL SHIPS
```

```
C
C   ISAG          VECTOR TO KEEP TRACK OF HOW MANY ASM/SSMS A SHIP
C                 FIRST ENGAGED
C
C   ISBG          SAME AS ISAG, BUT FOR RAIDERS
C
C   ISEED         THE INPUT FOR THE CALL RANDOM RANDOM NUMBER GENERATOR
C
C   ITEMS         THE NUMBER OF EVENTS THAT HAVE BEEN RETRIEVED FROM THE
C                 EVENTS LIST
C
C   IVENTS        THE EVENTS LIST
C
C   JACK          USED TO KEEP STATISTICS OF ENGAGEMENT AND DETECTION
C
C   JAM           A VECTOR TO KEEP TRACK OF THE PRESENCE OF JAMMING ON
C                 EACH OF THE FOUR SIMULATED RADAR FREQUENCY BANDS
C
C   KMART         AN ARRAY USED WITH ZLNKA TO DETERMINE THE REORDERING
C                 OF THE INPUT VECTOR
C
C   KOUNT         THE ITERATION IN PROGRESS
C
C   LOVER         THE NUMBER OF EVENTS CURRENTLY IN THE EVENTS LIST
C
C   MAMBAT        THE NUMBER OF AVAILABLE SAM BATTERIES
C
C   MARY          THE VECTOR THAT KEEPS THE TOATAL HITS STATISTIC
C
C   MISLFT        AN ARRAY TO KEEP TRACK OF ASMS LEFT ON BOARD EACH RAIDER
C
C   NADAR         THE ARRAY CONTAINING SEARCH RADAR, FC RADAR, AND LAUNCHER
C                 UP/DOWN STATUS
C
C   NASM          THE NUMBER OF ASMS LAUNCHED
C
C   NONO          THE ARRAY THAT GIVES THE SHIP VS. ASM/RAIDER  AN OK OR
C                 THE REASON FOR NON ENGAGEMENT
C
C   NPGS          A FLAG USED IN PRIOR
C
C   NR            THE NUMBER OF RED RAIDERS
C
C   NRTGT         A VECTOR LISTING THE TARGET LIST ASSIGNED TO EACH RAIDER
C
C   NS            THE NUMBER OF SHIPS IN THE BLUE FLEET
C
C   NSAM          NUMBER OF SALVOS FIRED
```

```
C
C     PLIST      THE ARRAY CONTAINING THE TARGETING PROBABILITIES
C
C     QRT        A DUMMY ARRAY USED FOR STORING RANDOM NUMBERS
C
C     RAID       THE ARRAY WITH THE RED RAIDER DATA
C
C     REASON     A FAILURE TO SHOOT A SAM CODE
C
C     RELOAD     THE ARRAY THAT TELLS WHETHER A LAUNCHER IS THROUGH
C                WITH ITS RELOAD CYCLE
C
C     RXATA      A DUMMY ARRAY USED TO LOAD INPUT
C
C     SAMASM     THE ARRAY WITH THE PARTICULAR SAM AND ASM ENGAGEMENT DATA
C
C     SMAT       THE ARRAY WITH THE SAM FIRING RESULT
C
C
C     SHIP       THE ARRAY WITH BLUE FORCE INFO
C
C     STATS      THE ARRAY FOR COMPILING SHIP DETECTION AND ENGAGEMENT
C                STATISTICS
C
C     TCF        USED TO REORDER THE CHANNEL AVAILABULITY TIMES
C
C     TGTAV      THE COORDINATION OF FIRE ARRAY
C
C     TMAX       THE TIME BY WHICH ALL AIR ASM LAUNCHES MUST BE COMPLETED
C
C     X          THE INPUT ARRAY HOLDING SCENARIO DATA, BLUE SHIP(AC)
C                DATA AND THE RED RAIDER DATA
C
C
      CALL OVFLOW
C
      AMF = 5.0E-3
C
C
C
      DO 30 I=1,40
C
C
C
      DO 20 J=1,6
      PLIST(I,J) = 0.
   20 CONTINUE
C
```

```
C
C
   30 CONTINUE
C
C          SET ALL   X   ARRAY TO ZERO
C
      DO 40 I=1,2744
      X(I) = 0.
   40 CONTINUE
C
C
      DO 50 I=1,100
      MARY(I) = 0
   50 CONTINUE
C
C          READ IN THE DATA
C
      DO 70 IJ=1,548
      READ (5,1170) IRT,RXATA
      IF (IRT.EQ.0) GO TO 80
      JRT = IRT+4
C
      DO 60 J=IRT,JRT
      X(J) = RXATA(J-IRT+1)
   60 CONTINUE
C
   70 CONTINUE
C
   80 ISEED = IFIX(X(11)+AMF)
      NI = IFIX(X(6)+AMF)
      IF (X(82).EQ.0.) X(82)=120.
      INSECT = ISEED
C
C          PRINT OUT THE X ARRAY AS READ INTO THE COMPUTER
C
      IF (X(12).EQ.0.) GO TO 100
      WRITE (6,1190)
C
      DO 90 I=1,343
      IJ = I+343
      IK = IJ+343
      IL = IK+343
      IM = IL+343
      IN = IM+343
      IO = IN+343
      IP = IO+343
      WRITE (6,1200) I,X(I),IJ,X(IJ),IK,X(IK),IL,X(IL),IM,X(IM),IN,X(IN)
     1,IO,X(IO),IP,X(IP)
```

```
   90 CONTINUE                                                          MAIN1860
C                                                                       MAIN1870
C       COMPUTE WEAPON HORIZON                                          MAIN1880
C                                                                       MAIN1890
  100 IF (X(81).GT.0.) GO TO 110                                        MAIN1900
      X(81) = 1.23                                                      MAIN1910
C                                                                       MAIN1920
  110 DO 120 I=2002,2192,10                                             MAIN1930
      IF (X(I-1).EQ.0.) GO TO 130                                       MAIN1940
      IK = I+7                                                          MAIN1950
      X(IK) = X(81)*SQRT(X(I))                                          MAIN1960
  120 CONTINUE                                                          MAIN1970
C                                                                       MAIN1980
C         CHANGE TRUE ANGLES TO CARTESIAN FOR SHIPS                     MAIN1990
C         SET SAM VELOCITY TO NM/MIN                                    MAIN2000
C                                                                       MAIN2010
  130 IN = IFIX(X(78)+AMF)                                              MAIN2020
C                                                                       MAIN2030
      DO 140 I=1,10                                                     MAIN2040
      ID = 502+(I-1)*40                                                 MAIN2050
      IK = ID+7                                                         MAIN2060
      IF (X(ID-1).EQ.0.) GO TO 150                                      MAIN2070
      X(ID) = X(ID)/60.                                                 MAIN2080
      X(IK) = X(IK)/60.                                                 MAIN2090
  140 CONTINUE                                                          MAIN2100
C                                                                       MAIN2110
C          CONVERT RADAR BEAM WIDTH TO 1/2 BEAM WIDTH IN RADIANS        MAIN2120
C          AND REPLACE SIDELOBE COEFFICIENT IN -DB WITH DIRECTIVE LOSSES MAIN2130
C                                                                       MAIN2140
  150 RAT = .8726646D-2                                                 MAIN2150
C                                                                       MAIN2160
      DO 160 NN=1,15                                                    MAIN2170
      NX = 20*(NN-1)                                                    MAIN2180
      IF (X(903+NX).EQ.0.) GO TO 170                                    MAIN2190
      BWR = DBLE(X(911+NX))*RAT                                         MAIN2200
      X(911+NX) = BWR                                                   MAIN2210
      BWR = X(904+NX)                                                   MAIN2220
      X(904+NX) = 10.**(BWR/10.)                                        MAIN2230
  160 CONTINUE                                                          MAIN2240
C                                                                       MAIN2250
  170 NS = IFIX(X(88)+AMF)                                              MAIN2260
      IF (IN.EQ.1) GO TO 190                                            MAIN2270
C                                                                       MAIN2280
      DO 180 I=1,NS                                                     MAIN2290
      IX = 5*(I-1)+1203                                                 MAIN2300
      X(IX) = 450.-X(IX)                                                MAIN2310
      IF (X(IX).GE.360.) X(IX)=X(IX)-360.                               MAIN2320
  180 CONTINUE                                                          MAIN2330
```

```
C
C        CHANGE RAIDER BEARINGS AND COURSES TO CARTESIAN
C        FIGURE FOURTH ROOT OF RAIDER AND WEAPON CROSS SECTIUN
C        CHANGE KNOTS TO NM/MIN
C         COMPUTE 1/2 JAMMER BEAM WIDTH IN RADIANS
C         REPLACE SIDELOBE COEFFICIENT IN -DB WITH DIRECTIVE LOSSES
C
  190 DO 200 I=1,10
      IX = 20*(I-1)
      IF (X(IX+1815).EQ.0.) GO TO 210
      X(IX+1814) = X(1814+IX)/60.
      RRC = DBLE(X(IX+1815))
      RRS = DSQRT(RRC)
      X(IX+1816) = DSQRT(RRS)
      IF (X(IX+1817).EQ.0.) GO TO 200
      BWJ = DBLE(X(IX+1817))*RAT
      X(IX+1817) = BWJ
      BWJ = X(IX+1818)
      X(IX+1818) = 10.**(BWJ/10.)
  200 CONTINUE
C
C
  210 DO 220 I=1,20
      IX = (I-1)*10
      IT = IX+2001
      IF (X(IT+5).EQ.0.) GO TO 230
      X(IT) = X(IT)/60.
      RRC = DBLE(X(IX+2006))
      RRS = DSQRT(RRC)
      X(IX+2010) = DSQRT(RRS)
  220 CONTINUE
C
  230 NR = IFIX(X(39)+AMF)
C
      DO 240 I=1,NR
      IX = (I-1)*6
      X(IX+2205) = 450.-X(IX+2205)
      IF (X(IX+2205).GE.360.) X(IX+2205)=X(IX+2205)-360.
      X(IX+2203) = 450.-X(IX+2203)
      IF (X(IX+2203).GE.360.) X(IX+2203)=X(IX+2203)-360.
  240 CONTINUE
C
      READ (5,1210) NRTGT
      IF (X(12).EQ.0.) GO TO 260
C
      DO 250 I=1,10
      IA = I+10
      IB = I+20
```

```
      IC = I+30                                                         MAIN2820
      ID = I+40                                                         MAIN2830
      IE = I+50                                                         MAIN2840
      IF = I+60                                                         MAIN2850
      IG = I+70                                                         MAIN2860
      IH = I+80                                                         MAIN2870
      WRITE (6,1220) I,NRTGT(I),IA,NRTGT(IA),IB,NRTGT(IB),IC,NRTGT(IC),IMAIN2880
     1D,NRTGT(ID),IE,NRTGT(IE),IF,NRTGT(IF),IG,NRTGT(IG),IH,NRTGT(IH)   MAIN2890
  250 CONTINUE                                                          MAIN2900
C                                                                       MAIN2910
      WRITE (6,1240)                                                    MAIN2920
  260 READ (5,1230) PLIST                                               MAIN2930
      IF (X(12).EQ.0.) GO TO 280                                        MAIN2940
C                                                                       MAIN2950
      DO 270 I=1,40                                                     MAIN2960
      WRITE (6,1250) I,PLIST(I,1),I,PLIST(I,2),I,PLIST(I,3),I,PLIST(I,4)MAIN2970
     1,I,PLIST(I,5),I,PLIST(I,6)                                        MAIN2980
  270 CONTINUE                                                          MAIN2990
C                                                                       MAIN3000
C                                                                       MAIN3010
  280 DO 340 I=1,NS                                                     MAIN3020
      IX = 5*(I-1)+1204                                                 MAIN3030
      X(IX) = 450.-X(IX)                                                MAIN3040
      IF (X(IX).GE.360.) X(IX)=X(IX)-36C.                               MAIN3050
      X(IX+1) = 450.-X(IX+1)                                            MAIN3060
      IF (X(IX+1).GE.360.) X(IX+1)=X(IX+1)-360.                         MAIN3070
C                                                                       MAIN3080
      DC 290 K=1,19                                                     MAIN3090
      NADAR(I,K) = 0                                                    MAIN3100
  290 CONTINUE                                                          MAIN3110
C                                                                       MAIN3120
C         SET THE SHIP TABLE                                            MAIN3130
C                                                                       MAIN3140
      IJ = 1201+(I-1)*5                                                 MAIN3150
      IK = IJ+1                                                         MAIN3160
      IL = IK+1                                                         MAIN3170
      IM = 101+(X(IJ)-1)*40                                             MAIN3180
      SHIP(I,1) = X(IJ)                                                 MAIN3190
      IF (IN.EQ.0) GO TO 300                                            MAIN3200
      SHIP(I,2) = X(IK)                                                 MAIN3210
      SHIP(I,3) = X(IL)                                                 MAIN3220
      GO TO 310                                                         MAIN3230
  300 RAT = DBLE(X(IL))                                                 MAIN3240
      RD = RADIAN(RAT)                                                  MAIN3250
      SHIP(I,2) = X(IK)*COS(RD)                                         MAIN3260
      SHIP(I,3) = X(IK)*SIN(RD)                                         MAIN3270
  310 IS = IM+20                                                        MAIN3280
      SHIP(I,4) = X(IS)                                                 MAIN3290
```

```
      SHIP(I,5) = FLOAT(IM)                                             MAIN3300
      IT = X(IM)                                                        MAIN3310
C                                                                       MAIN3320
      DO 320 K=1,4                                                      MAIN3330
      IBAT(I,K) = 0                                                     MAIN3340
  320 CONTINUE                                                          MAIN3350
C                                                                       MAIN3360
      IF (IT.GT.0) GO TO 330                                            MAIN3370
      SHIP(I,6) = 0.                                                    MAIN3380
      GO TO 340                                                         MAIN3390
  330 SHIP(I,6) = 1.                                                    MAIN3400
  340 CONTINUF                                                          MAIN3410
C                                                                       MAIN3420
C                                                                       MAIN3430
      DO 360 I=1,10                                                     MAIN3440
C                                                                       MAIN3450
      DO 350 J=1,3                                                      MAIN3460
      ND = 120+(I-1)*40+J*2                                             MAIN3470
      X(ND) = 450.-X(ND)                                                MAIN3480
      IF (X(ND).GE.360.) X(ND)=X(ND)-360.                               MAIN3490
      X(ND+1) = 450.-X(ND+1)                                            MAIN3500
      IF (X(ND+1).GE.360.) X(ND+1)=X(ND+1)-360.                         MAIN3510
  350 CONTINUE                                                          MAIN3520
C                                                                       MAIN3530
  360 CONTINUE                                                          MAIN3540
C                                                                       MAIN3550
C         THE INITIAL RAID TABLE                                        MAIN3560
C                                                                       MAIN3570
      IF (X(55).EQ.0.) GO TO 370                                        MAIN3580
      CALL SETIMX                                                       MAIN3590
C                                                                       MAIN3600
  370 DO 490 I=1,NR                                                     MAIN3610
      IJ = 2201+(I-1)*6                                                 MAIN3620
      IK = IJ+1                                                         MAIN3630
      IL = IJ+2                                                         MAIN3640
      IM = IJ+3                                                         MAIN3650
      IN = IJ+4                                                         MAIN3660
      IT = IFIX(X(IJ)+AMF)                                              MAIN3670
      JC = 1801+(IT-1)*20                                               MAIN3680
      VELR = X(JC+13)                                                   MAIN3690
      RAID(I,1) = FLOAT(JC)                                             MAIN3700
      ALFA = DBLE(X(IL))                                                MAIN3710
      RAT = RADIAN(ALFA)                                                MAIN3720
      COSI = DCOS(RAT)                                                  MAIN3730
      SINI = DSIN(RAT)                                                  MAIN3740
      RAID(I,2) = 0.                                                    MAIN3750
      RAID(I,3) = DBLE(X(IK))*COSI                                      MAIN3760
      RAID(I,4) = DBLE(X(IK))*SINI                                      MAIN3770
```

```
      IU = IFIX(X(JC+3)+AMF)                                          MAIN3780
      RRK = DBLE(X(82)*VELR)                                          MAIN3790
      RRS = DBLE(X(IN))                                               MAIN3800
      RAT = RADIAN(RRS)                                               MAIN3810
      RAID(I,5) = RRK*DCOS(RAT)+RAID(I,3)                             MAIN3820
      RAID(I,6) =  RRK*DSIN(RAT)+RAID(I,4)                            MAIN3830
      IF (IU.GT.0) GO TO 390                                          MAIN3840
  380 RAID(I,7) = RAID(I,5)                                           MAIN3850
      RAID(I,8) = RAID(I,6)                                           MAIN3860
      GO TO 450                                                       MAIN3870
  390 IW = IFIX(X(JC+5)+AMF)                                          MAIN3880
      IX = 2003+(IU-1)*10                                             MAIN3890
      IF (IW.EQ.0) GO TO 400                                          MAIN3900
      IY = 2003+(IW-1)*10                                             MAIN3910
      IF ((X(IX)-X(IY)).LT.0.) GO TO 410                              MAIN3920
      RK = X(IX)                                                      MAIN3930
      RB = X(IY)                                                      MAIN3940
      GO TO 420                                                       MAIN3950
  400 RK = X(IX)                                                      MAIN3960
      GO TO 430                                                       MAIN3970
  410 RK = X(IY)                                                      MAIN3980
      RB = X(IX)                                                      MAIN3990
  420 CALL TMDBOM (I,RB,T)                                            MAIN4000
      RAID(I,16) = T                                                  MAIN4010
  430 CALL TMDBOM (I,RK,T)                                            MAIN4020
      IF (T.GT.X(82)) GO TO 380                                       MAIN4030
      IF (T.EQ.0.) GO TO 440                                          MAIN4040
      CALL LOCTIM (I,T,3,XX,YY)                                       MAIN4050
      RAID(I,7) = XX                                                  MAIN4060
      RAID(I,8) = YY                                                  MAIN4070
      GO TO 450                                                       MAIN4080
  440 RAID(I,7) = RAID(I,3)                                           MAIN4090
      RAID(I,8) = RAID(I,4)                                           MAIN4100
      GO TO 450                                                       MAIN4110
  450 RAID(I,9) = X(IM)                                               MAIN4120
      RAID(I,10) = X(81)*SQRT(X(IM))                                  MAIN4130
      IF (IU.EQ.0) GO TO 470                                          MAIN4140
      IF (IW.EQ.0) GO TO 460                                          MAIN4150
      IF (X(IX).GT.X(IY)) GO TO 460                                   MAIN4160
      RAID(I,11) = X(JC+5)                                            MAIN4170
      GO TO 480                                                       MAIN4180
  460 RAID(I,11) = X(JC+3)                                            MAIN4190
      GO TO 480                                                       MAIN4200
  470 RAID(I,11) = 0.                                                 MAIN4210
  480 RAID(I,12) = X(JC+2)+X(JC+4)                                    MAIN4220
  490 CONTINUE                                                        MAIN4230
C                                                                     MAIN4240
      IF (X(24).EQ.2..OR.X(24).EQ.3.) CALL DISPL                      MAIN4250
```

```
      IF (X(24).EQ.1..OR.X(24).EQ.3.) CALL RAIDIS
C
C      GET THE NUMBER OF SAM IN THE FLEET
C
      NMIF = 0
C
      DO 510 L=1,NS
      IF (SHIP(L,6).EQ.0.) GO TO 510
      IT = IFIX(SHIP(L,5)+AMF)
C
      DO 500 K=1,4
      IF (X(IT-1+K).EQ.0.) GO TO 510
      MT = IFIX(X(IT-1+K)+AMF)
      NT = 527+(MT-1)*40
      NMIF = NMIF+IFIX(X(NT)+AMF)
  500 CONTINUE
C
  510 CONTINUE
C
      X(75) = FLOAT(NMIF)
      WRITE (6,1260) NMIF
      IF (X(2).EQ.1.) GO TO 610
C
      DO 600 I=1,NS
      IS = IFIX(SHIP(I,5)+AMF)
C
      DO 520 J=4,6
      K = J-3
      IF (X(IS+J).EQ.0.) GO TO 530
      NADAR(I,K) = 1
  520 CONTINUE
C
C
  530 DO 590 J=1,4
      IP = IFIX(X(IS+J-1)+AMF)
      IF (IP.EQ.0) GO TO 600
C
      DO 570 L=1,3
      IA = IFIX(X(516+(IP-1)*40+2*L)+AMF)
      IF (IA.GT.0) GO TO 560
      GO TO (540,580,580), L
C
  540 DO 550 LK=1,3
      NADAR(I,(3+(J-1)*3+LK))=2
  550 CONTINUE
C
      GO TO 580
  560 NADAR(I,(3+(J-1)*3+L))=1
```

```
  570 CONTINUE                                                          MAIN4740
C                                                                       MAIN4750
  580 NADAR(I,(15+J)) = 1                                               MAIN4760
  590 CONTINUE                                                          MAIN4770
C                                                                       MAIN4780
  600 CONTINUE                                                          MAIN4790
C                                                                       MAIN4800
  610 IF (X(15).EQ.0.0) GO TO 630                                       MAIN4810
      WRITE (6,1180)                                                    MAIN4820
C                                                                       MAIN4830
      DO 620 I=1,343                                                    MAIN4840
      IJ = I+343                                                        MAIN4850
      IK = IJ+343                                                       MAIN4860
      IL = IK+343                                                       MAIN4870
      IM = IL+343                                                       MAIN4880
      IN = IM+343                                                       MAIN4890
      IO = IN+343                                                       MAIN4900
      IP = IO+343                                                       MAIN4910
      WRITE (6,1200) I,X(I),IJ,X(IJ),IK,X(IK),IL,X(IL),IM,X(IM),IN,X(IN)MAIN4920
     1,IO,X(IO),IP,X(IP)                                                MAIN4930
  620 CONTINUE                                                          MAIN4940
C                                                                       MAIN4950
C         START THE GAME                                                MAIN4960
C                                                                       MAIN4970
  630 DO 990 I=1,NI                                                     MAIN4980
      KOUNT = I                                                         MAIN4990
      CALL EFFECT                                                       MAIN5000
      IJ = IFIX(X(75)+AMF)                                              MAIN5010
      WRITE (6,1270) KOUNT,IJ                                           MAIN5020
      WRITE (6,1280) KOUNT                                              MAIN5030
C                                                                       MAIN5040
C                PRINT THE ASM TABLE                                    MAIN5050
C                                                                       MAIN5060
      WRITE (6,1290)                                                    MAIN5070
C                                                                       MAIN5080
      DO 640 L=1,NASM                                                   MAIN5090
      LA = L+90                                                         MAIN5100
      WRITE (6,1300) LA,(ASMT(L,J),J=1,8)                               MAIN5110
  640 CONTINUE                                                          MAIN5120
C                                                                       MAIN5130
      IM = 0                                                            MAIN5140
C                                                                       MAIN5150
      DO 650 L=1,NR                                                     MAIN5160
      IF (RAID(L,13).GT.0.) IM=IM+1                                     MAIN5170
  650 CONTINUE                                                          MAIN5180
C                                                                       MAIN5190
      WRITE (6,1310) NR,IM                                              MAIN5200
      WRITE (6,1320)                                                    MAIN5210
```

```
      WRITE (6,1330) ((SAMASM(L,J),J=1,6),L=1,NSAM)                      MAIN5220
C                                                                        MAIN5230
C          FORMULATE THE STATISTICS FOR EACH ITERATION                   MAIN5240
C                                                                        MAIN5250
      ILF = 0                                                            MAIN5260
      ILS = 0                                                            MAIN5270
      IKS = 0                                                            MAIN5280
      IIF = 0                                                            MAIN5290
      ITS = 0                                                            MAIN5300
      IMF = 0                                                            MAIN5310
C                                                                        MAIN5320
      DO 720 L=1,NASM                                                    MAIN5330
      IL = IFIX(ASMT(L,1)+AMF)                                           MAIN5340
      IL = IL+1                                                          MAIN5350
      GO TO (660,670,680,690,700,710), IL                                MAIN5360
  660 ILF = ILF+1                                                        MAIN5370
      GO TO 720                                                          MAIN5380
  670 ILS = ILS+1                                                        MAIN5390
      GO TO 720                                                          MAIN5400
  680 IKS = IKS+1                                                        MAIN5410
      GO TO 720                                                          MAIN5420
  690 IIF = IIF+1                                                        MAIN5430
      GO TO 720                                                          MAIN5440
  700 ITS = ITS+1                                                        MAIN5450
      GO TO 720                                                          MAIN5460
  710 IMF = IMF+1                                                        MAIN5470
  720 CONTINUE                                                           MAIN5480
C                                                                        MAIN5490
      X(61) = X(61)+FLOAT(NASM)                                          MAIN5500
      X(62) = X(62)+FLOAT(ILF)                                           MAIN5510
      X(63) = X(63)+FLOAT(IKS)                                           MAIN5520
      X(64) = X(64)+FLOAT(IIF)                                           MAIN5530
      X(65) = X(65)+FLOAT(IMF)                                           MAIN5540
      X(69) = X(69)+FLOAT(ILS)                                           MAIN5550
      X(70) = X(70)+FLOAT(ITS)                                           MAIN5560
      WRITE (6,1340) KOUNT,ITS                                           MAIN5570
      IM = NASM-IMF                                                      MAIN5580
      IF (IM.NE.0) GO TO 730                                             MAIN5590
      P1 = 0.                                                            MAIN5600
      PC2 = 0.                                                           MAIN5610
      PC3 = 0.                                                           MAIN5620
      GO TO 740                                                          MAIN5630
  730 PC1 = FLOAT(IM)                                                    MAIN5640
      P1 = (FLOAT(ILF)/PC1)*100.                                         MAIN5650
      P2 = FLOAT(IM)-FLOAT(ILF)                                          MAIN5660
      PC2 = (FLOAT(IKS)/P2)*100.                                         MAIN5670
      PC3 = (FLOAT(IIF)/P2)*100.                                         MAIN5680
  740 PC4 = FLOAT(NASM)                                                  MAIN5690
```

```
      PC5 = (FLOAT(IMF)/PC4)*100.
      WRITE (6,1350) IM,ILF,P1,IKS,PC2,IIF,PC3
      WRITE (6,1360) IMF,PC5
      JL = 0
      IL = 0
      IK = NMIF-IFIX(X(75)+AMF)
      IF (NSAM.EQ.0) GO TO 770
C
      DO 760 L=1,NSAM
      IN = IFIX(SAMASM(L,5))
      IF (IN.NE.-1) GO TO 750
      IL = IL+IFIX(SMAT(L,10)+AMF)
      GO TO 760
  750 IF (IN.NE.0) GO TO 760
      JL = JL+IFIX(SMAT(L,10)+AMF)
  760 CONTINUE
C
      X(66) = X(66)+FLOAT(IK)
      X(67) = X(67)+FLOAT(JL)
      X(68) = X(68)+FLOAT(IL)
      PC1 = FLOAT(JL)
      P2 = (PC1/FLOAT(IK))*100.
      PC2 = FLOAT(IL)
      P3 = (PC2/FLOAT(IK))*100.
      WRITE (6,1370) IK,JL,P2,IL,P3
      GO TO 780
  770 WRITE (6,1060)
  780 IT = IFIX(X(13)+AMF)
      WRITE (6,1380) IT
      X(14) = X(14)+X(13)
      X(13) = 0.
C
C         DO THE SUMMARY STATISITICS FOR ITERATIONS COMPLETED
C
      WRITE (6,1390) KOUNT
      IM = IFIX(X(61)+AMF)
      IMF = IFIX(X(65)+AMF)
      IM = IM-IMF
      IF (IM.GT.0) GO TO 790
      PC1 = 0.
      PC2 = 0.
      PC3 = 0.
      IMK = 0
      IML = 0
      GO TO 800
  790 PC1 = (X(62)/FLOAT(IM))*100.
      PW = FLOAT(IM)-X(62)
      IMK = IFIX(X(63)+AMF)
```

```
      PC2 = (X(63)/PW)*100.
      IML = IFIX(X(64)+AMF)
      PC3 = (X(64)/PW)*100.
  800 ILF = IFIX(X(62)+AMF)
      PC4 = (X(65)/X(61))*100.
      WRITE (6,1350) IM,ILF,PC1,IMK,PC2,IML,PC3
      WRITE (6,1360) IMF,PC4
      HITS = X(70)/FLOAT(KOUNT)
      WRITE (6,1400) KOUNT,HITS
      IF (X(66).EQ.0.) GO TO 810
      IP = IFIX(X(66)+AMF)
      IP1 = IFIX(X(67)+AMF)
      IP2 = IFIX(X(68)+AMF)
      PC1 = (X(67)/X(66))*100.
      PC2 = (X(68)/X(66))*100.
      WRITE (6,1370) IP,IP1,PC1,IP2,PC2
  810 IT = IFIX(X(14)+AMF)
      WRITE (6,1380) IT
      WRITE (6,1410) KOUNT
      WRITE (6,1420)
      WRITE (6,1430) (L,(JACK(L,ND),ND=1,7),L=1,NS)
      WRITE (6,1080)
C
      DO 860 L=1,NS
      IF (JACK(L,5).EQ.0) GO TO 820
      USC = STATS(L,1)/FLOAT(JACK(L,5))
      GO TO 830
  820 USC = 0.
  830 IF (JACK(L,7).EQ.0) GO TO 840
      PC1 = STATS(L,2)/FLOAT(JACK(L,7))
      GO TO 850
  840 PC1 = 0.
  850 WRITE (6,1090) L,USC,PC1
  860 CONTINUE
C
      WRITE (6,1070) IFAG,IFBG
      WRITE (6,1100)
C
      DO 870 L=1,NS
      WRITE (6,1110) L,ISAG(L),ISBG(L)
  870 CONTINUE
C
      WRITE (6,1120)
C
      DO 880 L=1,NS
      WRITE (6,1130) L,IHIT(L)
  880 CONTINUE
C
```

```
C
      DO 890 L=1,NS
      IJ = IHIT(L)
      MARY(IJ) = MARY(IJ)+1
  890 CONTINUE
C
      IF (X(85).EQ.0.) GO TO 920
      L = IFIX(X(85)+AMF)
      HISX(I) = FLOAT(IHIT(L))
      TMA = 0.
      TMIN = 10000.
C
      DO 900 LL=1,NASM
      IF (ASMT(LL,1).NE.4.) GO TO 900
      IF (ASMT(LL,4).NE.X(85)) GO TO 900
      IF (ASMT(LL,2).LT.TMIN) TMIN=ASMT(LL,2)
      IF (ASMT(LL,2).GT.TMA) TMA=ASMT(LL,2)
  900 CONTINUE
C
      IF (HISX(I).LE.1.) GO TO 910
      HISA(I) = TMA-TMIN
      HISO(I) = HISX(I)/HISA(I)
      GO TO 920
  910 HISO(I) = 0.
      HISA(I) = 0.
  920 IF (KOUNT.GT.1) GO TO 980
      IJ = IFIX(X(51)+AMF)
      IF (IJ.EQ.0) GO TO 930
      WRITE (6,1440) (L,(SHIP(L,J),J=1,6),L=1,NS)
  930 IJ = IFIX(X(52)+AMF)
      IF (IJ.EQ.0) GO TO 940
      WRITE (6,1450) (L,(RAID(L,J),J=1,13),L=1,NR)
  940 IJ = IFIX(X(53)+AMF)
      IF (IJ.EQ.0) GO TO 950
      WRITE (6,1460)
      WRITE (6,1470) (L,(SMAT(L,J),J=1,10),L=1,NSAM)
  950 IJ = IFIX(X(54)+AMF)
      IF (IJ.EQ.0) GO TO 970
      X(54) = 0.
      WRITE (6,1480)
      IS = IFIX(X(14)+AMF)
C
      DO 960 L=1,IS
      CALL RANDOM (INSECT,Z,1)
      WRITE (6,1490) Z
  960 CONTINUE
C
  970 X(55) = 0.
```

```
      X(9) = 0.
      X(53) = 0.
      X(52) = 0.
      X(56) = 0.
      X(57) = 0.
      X(58) = 0.
      X(59) = 0.
      X(60) = 0.
  980 X(75) = FLOAT(NMIF)
      WRITE (6,1500)
  990 CONTINUE
C
      WRITE (6,1140)
      WRITE (6,1150)
C
      DO 1000 I=1,10
      IN = I+10
      IT = I+20
      IS = I+30
      IY = I+40
      IJ = I+50
      LA = I+60
      IO = I+70
      IG = I+80
      JC = I+90
      WRITE (6,1160) I,MARY(I),IN,MARY(IN),IT,MARY(IT),IS,MARY(IS),IY,MA
     1RY(IY),IJ,MARY(IJ),LA,MARY(LA),IO,MARY(IO),IG,MARY(IG),JC,MARY(JC)
 1000 CONTINUE
C
      IF (X(85).EQ.0.) GO TO 1010
      CALL HISTG (HISX,NI,0)
      WRITE (6,1030) X(85)
      CALL HISTG (HISA,NI,0)
      WRITE (6,1040) X(85)
      CALL HISTG (HISO,NI,0)
      WRITE (6,1050) X(85)
      WRITE (7,1020) HISO
 1010 STOP
C
 1020 FORMAT (F8.3)
 1030 FORMAT (40X,'HIT FREQUENCY ANALYSIS FOR SHIP',F4.0)
 1040 FORMAT (40X,'ASM/SSM ARRIVAL TIME INTERVAL ANALYSIS FOR SHIP',F4.0
     1)
 1050 FORMAT (' ',25X,'HITS PER MINUTE ON SHIP',F3.0)
 1060 FORMAT ('  NO SAMS FIRED')
 1070 FORMAT ('0',I6,' ASMS WERE FIRED ON BY THE FLEET'/(I7,' BOMBERS WE
     1RE FIRED ON BY THE FLEET'))
 1080 FORMAT ('0','SHIP   AVE DET RANGE   AVE DET RANGE'/(12X,'BOMBERS',
```

```
     110X,'ASMS'/))                                                     MAIN7620
1090 FORMAT (' ',I3,7X,F9.2,7X,F9.2)                                    MAIN7630
1100 FORMAT ('0','SHIP   ASMS FIRED AT     BOMBERS FIRED AT'/)          MAIN7640
1110 FORMAT (I4,I13,I19)                                                MAIN7650
1120 FORMAT ('0','                             HITS ON SHIP "I"')       MAIN7660
1130 FORMAT (I6,I5)                                                     MAIN7670
1140 FORMAT ('0',47X,'FINAL STATISTICS'/(' ',10('NO. OF',6X)))          MAIN7680
1150 FORMAT (' ',10('HITS FREQ. '))                                     MAIN7690
1160 FORMAT (' ',10(I4,I6,1X))                                          MAIN7700
1170 FORMAT (I10,5F10.3)                                                MAIN7710
1180 FORMAT ('1',45X,'THE X ARRAY AFTER CONVERSION'//,8(' ADR  CONTENT MAIN7720
    1 '))                                                               MAIN7730
1190 FORMAT ('1',48X,'THE X ARRAY AS READ IN'//,8(' ADR  CONTENT  '))   MAIN7740
1200 FORMAT (8(I5,F10.2))                                               MAIN7750
1210 FORMAT (30I2)                                                      MAIN7760
1220 FORMAT (9(I3,I2,5X))                                               MAIN7770
1230 FORMAT (10F8.4)                                                    MAIN7780
1240 FORMAT ('0',39X,'THE TARGET LISTS'/(6('  SHIP C-PROB  ')))         MAIN7790
1250 FORMAT (6(I5,F7.4,2X))                                             MAIN7800
1260 FORMAT ('0','   NMIF IS',I6)                                       MAIN7810
1270 FORMAT ('1','THE NUMBER OF SAMS LEFT AT THE END OF ITERATION',I3,'MAIN7820
    1 WAS',I5//)                                                        MAIN7830
1280 FORMAT ('0',55X,'ASM TABLE'/(55X,'ITERATION ',I2//(26X,'ASM     STMAIN7840
    1ATUS TIME OF  LAUNCH  TARGET  LAUNCH  WEAPON     X      Y')))      MAIN7850
1290 FORMAT (' ',24X,'NUMBER   CODE    IMPACT   RAID    SHIP   TIME   CMAIN7860
    1LASS   LAUNCH  LAUNCH'/)                                           MAIN7870
1300 FORMAT ('0',25X,I3,F8.0,F8.3,2F8.0,F9.2,F6.0,2F10.2)               MAIN7880
1310 FORMAT ('0',/(' ','THE ORIGINAL ATTACK CONSISTED OF',I3,' RAIDERS.MAIN7890
    1',I3,' WERE LEFT AFTER THE ATTACK.'/))                             MAIN7900
1320 FORMAT ('0',49X,'SAM-ASM-RAID TABLE'/(21X,'TGT NO.    LAUNCHER     MAIN7910
    1 CHANNEL   TIME OF LAUNCH    KILL CODE    TIME CHANNEL FREE'/))    MAIN7920
1330 FORMAT (20X,F7.0,F10.0,F12.0,F16.2,F14.0,F21.2)                    MAIN7930
1340 FORMAT ('1',10X,'STATISTICS FOR ITERATION NO.',I4//(20X,'THE SHIPSMAIN7940
    1 TOOK',I5,' HITS.'))                                               MAIN7950
1350 FORMAT (20X,'OF',I5,' ASMS FIRED,',I5,' (',F6.2,' PERCENT) FAILED MAIN7960
    1AT LAUNCH,'/(20X,'OF LAUNCHED ASM,',I5,' (',F6.2,' PERCENT) WERE KMAIN7970
    2ILLED BY SAMS,'/(38X,I5,' (',F6.2,' PERCENT) FAILED IN FLIGHT.')))MAIN7980
1360 FORMAT (' ',20X,'OF THE TOTAL ASMS AVAIL,',I5,' (',F6.2,' PERCENT)MAIN7990
    1 WERE DESTROYED WHEN'/(62X,'MOTHER KILLED BY SAMS.'))              MAIN8000
1370 FORMAT ('0',16X,I4,' SAMS WERE LAUNCHED OF WHICH ',I4,' (',F6.2,' MAIN8010
    1PERCENT) FAILED AND'/(50X,I4,' (',F6.2,' PERCENT) WERE OVERKILLS.'MAIN8020
    2))                                                                 MAIN8030
1380 FORMAT ('0'////(12X,I8,' RANDOM NUMBERS WERE DRAWN.'//))           MAIN8040
1390 FORMAT ('0','SUMMARY FOR',I3,' ITERATIONS')                        MAIN8050
1400 FORMAT ('0'//(20X,'AVERAGE NUMBER OF HITS ON FLEET FOR',I3,' ITERAMAIN8060
    1TIONS IS',F6.2//))                                                 MAIN8070
1410 FORMAT ('0',19X,'SHIP STATS FOR ITERATION',I3)                     MAIN8080
1420 FORMAT (' '/(' SHIP   CHANCES    NUMBER OF     TARGETS   BOMBERS   MAIN8090
```

```
      1    BOMBERS   ASMS',9X,'ASMS'/('  NO.    TO ENGAGE  ENGAGEMENTS    MAIN8100
     2KILLED    DETECTABLE   DETECTED   DETECTABLE   DETECTED'/)))         MAIN8110
 1430 FORMAT (' ',I4,I10,I11,I13,I10,I13,I9,I11)                          MAIN8120
 1440 FORMAT ('0','SHIP   CLASS       X         Y      HITS TO SINK  CLA-ADRMAIN8130
      1    UP/DOWN'//(I4,F7.0,F10.2,F10.2,F9.0,F12.0,F11.0))              MAIN8140
 1450 FORMAT (' '//,' RAIDER CLASS   TIMER(R)    XR       YR        XEOG   YMAIN8150
     1EOG    XL        YL      ALT      HORZ  WEP-CLASS  WEPS  UP/DOWN'/(' ',MAIN8160
     2I3,F10.0,F9.2,F7.1,3F8.1,2F7.1,F8.1,F7.0,F10.0,F8.0,F7.0))          MAIN8170
 1460 FORMAT (' '//(5X,'SAM   SAM CLASS INDEX   SHIP NUMBER   SAM LEFT   MAIN8180
      1 CH1 FREE   CH2 FR   CH3 FR   CH4 FR   CH5 FR   CH6 FR   SALVO'))  MAIN8190
 1470 FORMAT (' ',4X,I3,F13.0,F17.0,F11.0,2F11.2,4F9.2,F8.0)              MAIN8200
 1480 FORMAT ('  RANDOM NUMBERS USED IN ITERATION ONE.'/)                 MAIN8210
 1490 FORMAT (F8.4)                                                       MAIN8220
 1500 FORMAT ('1')                                                        MAIN8230
      END                                                                 MAIN8240


      SUBROUTINE EFFECT                                                   EFT   10
C                                                                         EFT   20
C         THIS SUBROUTINE CONTROLS EACH ITERATION                         EFT   30
C                                                                         EFT   40
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SMEFT   50
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEEEFT   60
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASOEFT   70
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,QEFT   80
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),EFT   90
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4EFT  100
     60),AMF,HISX(100),JACK(40,7),HISA(100)                               EFT  110
      IJ = X(55)                                                          EFT  120
      IF (IJ.EQ.0) GO TO 20                                               EFT  130
      CALL GETIMX (IET)                                                   EFT  140
      TIMES = IET*.000026                                                 EFT  150
      WRITE (6,130) TIMES                                                 EFT  160
   20 CALL JULIE                                                          EFT  170
   30 CALL GETEV (TIME,NEXTEV,IUNIT)                                      EFT  180
      IF (NEXTEV.EQ.0) GO TO 100                                          EFT  190
      IF (NEXTEV.GT.6) GO TO 120                                          EFT  200
      GO TO (40,50,60,70,80,90), NEXTEV                                   EFT  210
   40 CALL SUBE1 (TIME,IUNIT)                                             EFT  220
      GO TO 30                                                            EFT  230
   50 CALL SUBE2 (TIME,IUNIT)                                             EFT  240
      GO TO 30                                                            EFT  250
   60 CALL SUBE3 (TIME,IUNIT)                                             EFT  260
      GO TO 30                                                            EFT  270
   70 CALL SUBE4 (IUNIT)                                                  EFT  280
      GO TO 30                                                            EFT  290
   80 CALL SUBE5 (IUNIT)                                                  EFT  300
      GO TO 30                                                            EFT  310
```

```
   90 CALL SUBE6 (IUNIT)
      GO TO 30
  100 IF (IJ.EQ.0) GO TO 110
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,140) TIMES
  110 RETURN
  120 WRITE (6,150) NEXTEV
      STOP
C
  130 FORMAT ('0',' EFFECT LOGIN AT ',F11.7)
  140 FORMAT ('0',' EFFECT LOGOUT AT',F11.7)
  150 FORMAT ('0','**  NON-EXISTENT EVENT CALLED **** TYPE ',I3)
      END


      SUBROUTINE JULIE
C
C          THIS SUBROUTINE INITIALIZES  OR RESETS ITEMS REQUIRED
C          TO BE RESET FOR EACH ITERATION.
C
      REAL *8RAT,BNG,RAN
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     20,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      IF (X(55).EQ.0.) GO TO 20
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,530) TIMES
   20 CALL EVENT
      IFIRST = 0
      NASM = 0
      IFLASK = 0
      ITEMS = 0
      NSAM = 0
      IFAG = 0
      IFBG = 0
C
C         SET CHANNEL ARRAY TO ZERO
C
      DO 60 I=1,NS
C
      DO 50 J=1,4
C
```

```
      DO 40 K=1,3
C
      DO 30 L=1,6
      CHANEL(I,J,K,L) = 0.
   30 CONTINUE
C
   40 CONTINUE
C
   50 CONTINUE
C
   60 CONTINUE
C
C           SET JAM ARRAY TO ZERO
C
      DO 70 I=1,4
      JAM(I) = 0
   70 CONTINUE
C
C
      DO 100 I=1,NS
      IHIT(I) = 0
      ISAG(I) = 0
      ISBG(I) = 0
C
      DO 80 J=1,7
      JACK(I,J) = 0
   80 CONTINUE
C
C
      DO 90 J=1,2
      STATS(I,J) = 0.
   90 CONTINUE
C
  100 CONTINUE
C
C
      DO 120 I=1,190
      TGTAV(I) = 0.
      IGAGE(I) = 0
C
      DO 110 J=1,40
      NONO(I,J) = 0
  110 CONTINUE
C
  120 CONTINUE
C
C
      DO 140 I=1,300
```

```
C
      DO 130 J=1,7
      SAMASM(I,J) = 0.
  130 CONTINUE
C
  140 CONTINUE
C
C          RESET RAID TABLE
C
      DO 150 I=1,NR
      IOR = IFIX(RAID(I,1)+AMF)
      MISLFT(I,1) = IFIX(X(IOR+2)+AMF)
      MISLFT(I,2) = IFIX(X(IOR+4)+AMF)
      IF (X(IOR+9).GT.0.) JAM(1)=JAM(1)+1
      IF (X(IOR+10).GT.0.) JAM(2)=JAM(2)+1
      IF (X(IOR+11).GT.0.) JAM(3)=JAM(3)+1
      IF (X(IOR+12).GT.0.) JAM(4)=JAM(4)+1
      RAID(I,13) = X(IOR+1)
      RAID(I,2) = 0.
      RAN = DBLE(X(2202+(I-1)*6))
      BNG = DBLE(X(2203+(I-1)*6))
      RAT = RADIAN(BNG)
      RAID(I,3) = RAN*DCOS(RAT)
      RAID(I,4) = RAN*DSIN(RAT)
      RAID(I,14) = RAID(I,5)
      RAID(I,15) = RAID(I,6)
      RAID(I,17) = 1000.
  150 CONTINUE
C
C          RESET SHIP TABLE
C
      DO 180 I=1,NS
      ID = IFIX(SHIP(I,1)+AMF)
      IID = (ID-1)*40
      SHIP(I,4) = X(121+IID)
      IJ = IFIX(X(101+IID)+AMF)
      IF (IJ.EQ.0) GO TO 180
      SHIP(I,6) = 1.
C
      DO 160 K=1,4
      IJ = IFIX(X(100+K+IID)+AMF)
      IF (IJ.EQ.0) GO TO 170
      IBAT(I,K) = IFIX(X(527+(IJ-1)*40)+AMF)
  160 CONTINUE
C
  170 IF (X(58).EQ.0.) GO TO 180
      WRITE (6,540) (I,(IBAT(I,J),J=1,K))
  180 CONTINUE
```

```
C
C
      DO 200 I=1,300
C
      DO 190 J=1,10
      SMAT(I,J) = 0.
  190 CONTINUE
C
  200 CONTINUE
C
      IF (X(2).EQ.0.) GO TO 450
C
C             IF THE FLAG IS SET, THE AVAILABILITY OF ALL RADARS AND
C             LAUNCHERS IS DETERMINED FOR EACH ITERATION.  IF ALL
C             SEARCH RADARS ARE DOWN, THE ROUTINE CHECKS TO SEE IF
C             THE FIRE CONTROL RADARS ARE HORIZON SEARCH CAPABEL. AT
C             THIS TIME IT WILL ALSO CHECK TO SEE IF THE MISSILES ARE
C             PASSIVE (HOJ OR ARM) OR IF THEY REQUIRE A REMOTE ILLUM-
C             INATOR.
C
C          DOES SHIP I HAVE A MISSILE SYSTEM?  NO GO TO 540
C
      DO 440 I=1,NS
      IF (SHIP(I,6).EQ.0.) GO TO 440
      IS = IFIX(SHIP(I,5)+AMF)
      IFOG = 0
      WRITE (6,550) I
      KFOG = 0
C
C          CHECK FOR AT LEAST ONE UP SEARCH RADAR
C
      DO 220 J=4,6
      K = J-3
      IT = IFIX(X(IS+J)+AMF)
      IF (IT.EQ.0) GO TO 230
      PROB = X(913+(IT-1)*20)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      IF (Z.GT.PROB) GO TO 210
      KFOG = 1
      WRITE (6,560) K
      NADAR(I,K) = 1
      GO TO 220
  210 WRITE (6,570) K
      NADAR(I,K) = 0
  220 CONTINUE
C
  230 IF (KFOG.EQ.1) GO TO 270
```

```
C
      DO 240 J=1,4
      IP = IFIX(X(IS+J-1)+AMF)
      IF (IP.EQ.0) GO TO 260
      IC = IFIX(X(518+(IP-1)*40)+AMF)
      IF (IC.NE.0) GO TO 250
  240 CONTINUE
C
  250 IF (X(7).EQ.1.) GO TO 270
  260 SHIP(I,6) = 0.
      GO TO 440
C
  270 DO 390 J=1,4
      IP = X(IS+J-1)
      IF (IP.EQ.0) GO TO 390
C
      DO 380 L=1,3
      IA = IFIX(X(516+(IP-1)*40+2*L)+AMF)
      IF (IA.GT.0) GO TO 300
      GO TO (280,390,390), L
  280 IFOG = 1
      WRITE (6,580) J
C
      DO 290 LK=1,3
      NADAR(I,(3+(J-1)*3+LK))=2
  290 CONTINUE
C
      GO TO 390
  300 PROB = X(913+(IA-1)*20)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      IF (PROB.GT.Z) GO TO 320
  310 NADAR(I,(3+(J-1)*3+L))=0
      WRITE (6,590) L,J
      GO TO 380
  320 IF (KFOG.EQ.0) GO TO 330
      NADAR(I,(3+(J-1)*3+L))=1
      WRITE (6,600) L,J
      IFOG = 1
      GO TO 380
  330 ID = IFIX(X(523+(IP-1)*40)+AMF)
      GO TO (340,350,360), L
  340 IF (ID.EQ.1.OR.ID.EQ.11.OR.ID.EQ.101.OR.ID.EQ.111) GO TO 370
      GO TO 310
  350 IF (ID.EQ.10.OR.ID.EQ.11.OR.ID.EQ.110.OR.ID.EQ.111) GO TO 370
      GO TO 310
  360 IF (ID.EQ.100.OR.ID.EQ.101.OR.ID.EQ.110.OR.ID.EQ.111) GO TO 370
      GO TO 310
```

```
  370 IFOG = 1
      WRITE (6,610) L,J
      NADAR(I,(3+(J-1)*3+L))=3
  380 CONTINUE
C
  390 CONTINUE
C
      IF (KFOG.EQ.1.OR.IFOG.EQ.1) GO TO 400
      SHIP(I,6) = 0.
      WRITE (6,620)
      GO TO 440
  400 IFOG = 0
C
C           DETERMINE LAUNCHER STATUS
C
      DO 420 J=1,4
      IP = IFIX(X(IS+J-1)+AMF)
      IF (IP.EQ.0) GO TO 430
      PROB = X(526+(IP-1)*40)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      IF (PROB.GT.Z) GO TO 410
      NADAR(I,(15+J)) = 0
      WRITE (6,630) J
      GO TO 420
  410 IFOG = 1
      NADAR(I,(15+J)) = 1
      WRITE (6,640) J
  420 CONTINUE
C
  430 IF (IFOG.EQ.1) GO TO 440
      SHIP(I,6) = 0.
      WRITE (6,620)
  440 CONTINUE
C
C
  450 DO 460 I=1,NR
      CALL STOREV (0.,1,I)
      ID = IFIX(RAID(I,11)+AMF)
      IF (ID.EQ.0) GO TO 460
      RGE = X(2003+(ID-1)*10)
      CALL TMOBOM (I,RGE,T)
      CALL STOREV (T,2,I)
      IF (X(60).EQ.0.) GO TO 460
      WRITE (6,650)
      WRITE (6,660) (LOP,(IVENTS(LOP,J),J=1,2),LOP=1,LOVER)
  460 CONTINUE
C
```

```
      MAMBAT = 0
C
      DO 480 I=1,NS
C
      DO 470 J=16,19
      MAMBAT = MAMBAT+NADAR(I,J)
  470 CONTINUE
C
  480 CONTINUE
C
C
      DO 500 I=1,NS
C
      DO 490 J=1,4
      RELOAD(I,J) = 0.
  490 CONTINUE
C
  500 CONTINUE
C
      IF (X(60).EQ.0.) GO TO 510
      WRITE (6,650)
      WRITE (6,660) (I,(IVENTS(I,J),J=1,2),I=1,LOVER)
  510 IF (X(55).EQ.0.) GO TO 520
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,670) TIMES
  520 RETURN
C
  530 FORMAT ('0','   JULIE CALLED AT',F15.5)
  540 FORMAT ('0'/(17X,'IBAT TABLE'//(' ',5I10)))
  550 FORMAT ('0',' EQUIPMENT STATS FOR SAM SHIP ',I3)
  560 FORMAT (' ',' RADAR ',I2,' UP')
  570 FORMAT (' ',' RADAR ',I2,' DOWN')
  580 FORMAT ('      SYSTEM',I2,' REQUIRES NO DEDICATED FC RADAR')
  590 FORMAT (' ',' FC RADAR',I2,' SYSTEM',I2,' DOWN.')
  600 FORMAT (' ',' FC RADAR',I2,' SYSTEM',I2,' UP.')
  610 FORMAT (' ',' FC RADAR',I2,' SYSTEM',I2,' OPERATING IN HORIZON SEA
     1RCH MODE.')
  620 FORMAT ('         SHIP DESIG NON-COMBAT DUE TO DOWN EQUIPMENT.')
  630 FORMAT ('      LAUNCHER',I2,' DOWN.')
  640 FORMAT ('      LAUNCHER',I2,' UP.')
  650 FORMAT ('          EVENTS LIST'/('   SLOT     TIME           CODE'))
  660 FORMAT (I7,I12,I13)
  670 FORMAT ('0','  JULIE OUT AT',F15.8)
      END

      SUBROUTINE EVENT
```

```
C
C         THIS SUBROUTINE PUTS THE END OF GAME TIME IN THE
C         EVENTS LIST AFTER ZEROING OUT THE OLD LISTING.
C
      REAL *8HOTDOG
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(300)
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 20
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,210) TIMES
   20 LOVER = 1
C
C
C
      DO 40 I=1,1000
C
      DO 30 J=1,2
      IVENTS(I,J) = 0
   30 CONTINUE
C
   40 CONTINUE
C
      T = X(82)
      IT = 1000*(T+.0005)
      IVENTS(1,2) = 0
      IVENTS(1,1) = IT
      IF (IQ.EQ.0) GO TO 50
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,220) TIMES
   50 RETURN
C
      ENTRY GETEV(TIME,NEXTEV,IUNIT)
C
C         THIS ENTRY GETS THE NEXT EVENT FROM THE EVENTS LIST AND
C         CONVERTS IT INTO:
C            (A)  TIME..........THE TIME OF THE EVENT
C            (B)  NEXTEV........THE TYPE OF NEXT EVENT
C            (C)  IUNIT.........THE UNITS INVOLVED IN THE NEXT EVENT
C
      IF (X(55).EQ.0.) GO TO 60
```

```
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,230) TIMES
   60 ITIM = IVENTS(1,1)
      IVET = IVENTS(1,2)
      HOTDOG = DFLOAT(IVET)
      NEXTEV = HOTDOG*1.0D-8
      TIME = .00100*ITIM
      IVEN = NEXTEV*100000000
      IUNIT = IVET-IVEN
      ITEMS = ITEMS+1
      IF (X(60).EQ.0.) GO TO 70
      WRITE (6,240) TIME,NEXTEV,IUNIT,ITEMS
C
   70 DO 80 I=2,LOVER
      IVENTS((I-1),1) = IVENTS(I,1)
      IVENTS((I-1),2) = IVENTS(I,2)
   80 CONTINUE
C
      IVENTS(I,1) = 0
      LOVER = LOVER-1
      IF (X(55).EQ.0.) GO TO 90
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,250) TIMES
   90 RETURN
C
      ENTRY STOREV(T,ID,NUMBER)
C
C          THIS SUBROUTINE ENTRY STORES THE INDICATED EVENT IN
C          ITS PROPER PLACE IN THE EVENTS STORE LIST.  INPUTS ARE
C
C             T..........THE TIME OF THE EVENT
C             ID.........THE TYPE OF EVENT
C             NUMBER.....THE CODE FOR THE UNITS INVOLVED
C
C          THE PROGRAM STORES THE EVENT BY TIME. IN CASE OF TIES
C          IT SORTS BY LOWEST EVENT TYPE. IF THERE IS STILL A DUPLI-
C         CATION IT RESORTS TO THE LARGEST "NUMBER".
C
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 100
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,260) TIMES
  100 IF (X(82).LT.T) GO TO 190
      IF (LOVER.GE.1000) GO TO 160
      ICODE = ID*100000000
```

```
      ICODE = ICODE+NUMBER
      IT = 1000*(T+.0005)
C
      DO 110 I=1,1000
      ITB = IVENTS(I,2)
      IF (ITB.EQ.0) GO TO 130
      ITM = IVENTS(I,1)
      IF (IT.LT.ITM) GO TO 130
      IF (IT.EQ.ITM) GO TO 120
  110 CONTINUE
C
      WRITE (6,200)
      STOP
  120 IF (ITB.LT.ICODE) GO TO 110
  130 LOVER = LOVER+1
      IF (X(60).EQ.0.) GO TO 140
      WRITE (6,270) IT,ICODE,LOVER
C
  140 DO 150 M=I,LOVER
      MORE1 = IVENTS(M,1)
      MORE2 = IVENTS(M,2)
      IVENTS(M,1) = IT
      IVENTS(M,2) = ICODE
      IT = MORE1
      ICODE = MORE2
  150 CONTINUE
C
      GO TO 170
  160 WRITE (6,280) T,ID,NUMBER
  170 IF (IQ.EQ.0) GO TO 180
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,290) TIMES
  180 RETURN
  190 IF (X(60).EQ.1.) WRITE (6,300) T,ID,NUMBER
      GO TO 170
C
  200 FORMAT ('      EVENTS LIST ITEREATED THROUGH   ERROR')
  210 FORMAT ('      EVENT IN AT',F15.8)
  220 FORMAT ('       EVENT OUT AT',F15.8)
  230 FORMAT ('       GETEV IN AT',F15.8)
  240 FORMAT ('0','     THE DATA RETREVED WAS,  TIME',F7.3,'  EVENT TYPE',
     1I2,'  IUNIT',I12,'.'/('              THIS WAS THE',I6,'TH RETREVAL.'))
  250 FORMAT ('        GETEV OUT AT ',F15.8)
  260 FORMAT ('0','    STOREV IN AT',F15.8)
  270 FORMAT ('0',' EVENT STORED    TIME ',I8,' , CODE IS',I10,',  THERE
     1ARE NOW',I5,' EVENTS LISTED')
  280 FORMAT ('0','   EVENT LOST--TIME',F9.3,' TYPE',I2,'  CODE',I9,'. OV
```

```
     1ER 1000 EVENTS LISTED.')
 290 FORMAT ('0','  STOREV OUT AT',F15.8)
 300 FORMAT ('0','     EVENT LOST DUE TO TIME CONSTRAINT. TIME OF LOST
    1EVENT WAS',F15.4,'  TYPE',I2,', CODE',I10)


      SUBROUTINE REMOVE (IUNYT)
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
C
C         THIS ENTRY REMOVES THE TARGET FROM THE EVENTS LIST THAT
C         PRIOR HAS DETERMINED IS MORE THREATENING THAT THE ONE
C         GIVEN IT BY SUBE3.
C
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 20
      CALL GETIMX (IET)
      TIM = IET*2.6E-5
      WRITE (6,70) TIM
C
   20 DO 30 I=1,LOVER
      IF (IVENTS(I,2).EQ.IUNYT) GO TO 40
   30 CONTINUE
C
      WRITE (6,80) IUNYT
      WRITE (6,90) ((IVENTS(I,J),J=1,2),I=1,LOVER)
      STOP
C
   40 DO 50 J=I,LOVER
      IVENTS(J,1) = IVENTS((J+1),1)
      IVENTS(J,2) = IVENTS((J+1),2)
   50 CONTINUE
C
      LOVER = LOVER-1
      IF (IQ.EQ.0) GO TO 60
      CALL GETIMX (IET)
      TIM = IET*2.6E-5
      WRITE (6,100) TIM
   60 RETURN
C
   70 FORMAT (' REMOVE IN AT',F15.8)
   80 FORMAT ('0','ERROR IN REMOVE  IUNYT NOT FOUND'/(' IUNYT WAS',I10/(
```

```
     1' EVENTS LIST IS'/)))
  90 FORMAT (2I14)
 100 FORMAT (' REMOVE OUT AT',F15.8)

      SUBROUTINE TARGET (TYME,IUNIT,MORE)
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
C
C         THIS ENTRY CHECKS THE EVENTS LIST TO SEE IF THERE IS A
C         CANDIDATE FOR EVALUATION BY PRIOR
C
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 20
      CALL GETIMX (IET)
      T = IET*2.6E-5
      WRITE (6,70) T
   20 IT = 1000*(TYME+5.0E-4)
      IF (IFIRST.NE.0) GO TO 30
      IFIRST = 1
      MORE = 1
      NPGS = 1
   30 IF (IVENTS(NPGS,1).NE.IT) GO TO 60
      IF (IVENTS(NPGS,2).LT.300000000.OR.IVENTS(NPGS,2).GE.400000000) GO
     1 TO 60
      IUNIT = IVENTS(NPGS,2)
      NPGS = NPGS+1
   40 IF (IQ.EQ.0) GO TO 50
      CALL GETIMX (IET)
      T = IET*2.6E-5
      WRITE (6,80) T
   50 RETURN
   60 IFIRST = 0
      MORE = 0
      GO TO 40
C
   70 FORMAT (' TARGET IN AT',F15.8)
   80 FORMAT (' TARGET OUT AT',F15.8)
      END

      SUBROUTINE SUBE1 (TIME,IR)
```

```
C
C     THIS SUBROUTINE INPUTS THE RAIDER NUMBER AND TIME OF THE EVENT,
C     AND WILL STORE SHOOT EVENTS IN THE EVENTS LIST AS APPROPRIATE.
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      NASA = 0
      NSURF = 0
      NJ = IFIX(X(17)+AMF)
      IF (X(55).EQ.0.) GO TO 20
      CALL GETIMX (IET)
      TIM = IET*.000026
      WRITE (6,910) TIM
   20 IF (X(16).EQ.0.) GO TO 30
      IF (NASM.GT.0) GO TO 30
      TIME = TIME+2.
      CALL STOREV (TIME,1,IR)
      GO TO 740
C
C     CHECK AIR TARGET OR SURFACE TARGET
C
   30 ID = 2201+(IR-1)*6
      IF (X(ID+3).GT.50.) NASA=1
      NOTE = IFIX(X(ID)+AMF)
      NIP = (NOTE-1)*20
C
C     GET RAIDER PARAMETERS
C
      AREA = X(1816+NIP)
      SIGMA = X(1815+NIP)
      ALT = X(ID+3)
      VEL = X(1814+NIP)
      SALT = X(81)*SQRT(ALT)
      IFLAG = 0
      KFLAG = 0
C
      DO 730 I=1,NS
      IF (SHIP(I,6).GT.0.) GO TO 40
      NONO(IR,I) = 7
      GO TO 730
C
C     CHECK  SHIP FOR TYPE MISSILES
C
```

```
   40 ISI = IFIX(SHIP(I,5)+AMF)
      RGE = 0.
      IF (NASA.EQ.1) GO TO 90
C
      DO 70 N=1,4
      IA = IFIX(X(ISI+N-1)+AMF)
      IF (IA.EQ.0) GO TO 80
      IF (X(536+(IA-1)*40).EQ.0.) GO TO 70
      IF (NADAR(I,(15+N)).EQ.0) GO TO 70
      MP = 3+(N-1)*3
C
      DO 50 K=1,3
      MPI = MP+K
      IF (NADAR(I,MPI).GT.0) GO TO 60
   50 CONTINUE
C
      GO TO 70
   60 ICS = 501+(IA-1)*40
      SRAG = X(ICS)
      IF (SRAG.LE.RGE) GO TO 70
      RGE = SRAG
      L2 = N
      SAMVEL = X(ICS+1)
      NSURF = 1
   70 CONTINUE
C
   80 IF (NSURF.EQ.1) GO TO 150
      NCNO(IR,I) = 9
      IF (X(57).EQ.1.) WRITE (6,920) IR,I
      GO TO 730
C
   90 DO 140 L=1,4
      IC = IFIX(X(ISI+L-1)+AMF)
      IF (X(536+(IC-1)*40).EQ.1.) GO TO 140
      IF (NADAR(I,(15+L)).EQ.0) GO TO 140
      MP = 3+(L-1)*3
C
      DO 100 K=1,3
      MPI = MP+K
      IF (NADAR(I,MPI).GT.0) GO TO 110
  100 CONTINUE
C
      GO TO 140
  110 IF (ALT.GT.X(77)) GO TO 120
      ICS = 508+(IC-1)*40
      GO TO 130
  120 ICS = 501+(IC-1)*40
  130 SRAG = X(ICS)
```

```
      IF (SRAG.LE.RGE) GO TO 140
      RGE = SRAG
      L2 = L
      SAMVEL = X(ICS+1)
  140 CONTINUE
C
  150 IF (TIME.EQ.0.) GO TO 160
      CALL LOCTIM (IR,TIME,5,XCP,YCP)
      RAID(IR,14) = RAID(IR,5)
      RAID(IR,15) = RAID(IR,6)
C
C        CHECK TO SEE IF TARGET WILL COME WITHIN MAX MISSILE RANGE
C
  160 CALL TMDASM (IR,0,L2,I,VEL,RGE,TIME,TD,XI,YI,RIP,TIML,IR)
      IF (TD.LT.9120.) GO TO 170
      NONO(IR,I) = 9
      GO TO 730
  170 IF (TD.LT.X(82)) GO TO 180
      NONO(IR,I) = 8
      GO TO 730
  180 TFIR = TIML-RGE/SAMVEL
      IF (TFIR.GT.TD) GO TO 190
      NONO(IR,I) = 8
      GO TO 730
  190 KFLAG = 0
      TA = 0.
      TRGE = 1000.
      RGTI = 0.
      RGE = 0.
      JACK(I,1) = JACK(I,1)+1
      JACK(I,4) = JACK(I,4)+1
      NFOG = 0
      MFOG = 0
      JFOG = 0
      CALL CPA (RAID(IR,3),RAID(IR,4),RAID(IR,14),RAID(IR,15),SHIP(I,2),
     1SHIP(I,3),XCP,YCP)
      RCPS = (SHIP(I,2)-XCP)**2+(SHIP(I,3)-YCP)**2+(RAID(IR,9)/6000.)**2
C
C        CHECK SEARCH RADARS    SEE IF ONE IS OPERABLE
C
      DO 280 NSR=1,3
      IF (NADAR(I,NSR).EQ.0) GO TO 280
      KFLAG = 1
      ID = IFIX(X(ISI+3+NSR)+AMF)
      IA = IFIX(X(901+(ID-1)*20)+AMF)
C
C        BI-STATIC/MONO-STATIC FLAG
C
```

```
      IF (IA.EQ.0) GO TO 790
C
C        MONOSTATIC DETECTION ALGCRITHM
C
      SRAG = X(902+(ID-1)*20)*AREA
      RRH = X(81)*SQRT(X(ISI+6+NSR))
      HRAG = RRH+SALT
      THR = X(905+(ID-1)*20)
      RG = AMIN1(SRAG,HRAG,THR)
      IF (NJ.EQ.0) GO TO 200
      IJ = IFIX(X(907+(ID-1)*20)+AMF)
      IF (JAM(IJ).EQ.0) GO TO 200
      CALL JAMM (RG,I,IR,IJ,ID,VEL,RRH,AREA,0,ICPA,RGI)
      IF (ICPA.EQ.0) GO TO 280
      RAGJ = RGI
      GO TO 210
  200 IF (RCPS.GE.RG**2) GO TO 280
      RGI = RG
      RAGJ = 999.
  210 IF (X(3).EQ.0.) GO TO 220
      WRITE (6,930) I,IR,NSR
      WRITE (6,940) SRAG,RAGJ,HRAG
  220 CALL TMDASM (IR,0,NSR,I,VEL,RGI,TIME,TI,XI,YI,RIP,TIML,IR)
  230 IF (TA.LT.TIML) TA = TIML
C
C        PUT IN DETECTION REACTION TIME
C
      IF (X(10).EQ.0.) GO TO 250
      IF (VEL.LT.1.0E-5) GO TO 240
      RGE = ABS(RIP-DELR*VEL)
      GO TO 260
  240 DELR = X(87)
      GO TO 260
  250 KEY = IFIX(X(906+(ID-1)*20)+AMF)
      CALL REACT (1,KEY,DELR)
  260 TDET = TI+DELR
C
C        CHECK FOR MAXIMUM RANGE
C
      IF (TRGE.LE.TDET) GO TO 270
      RGE = RIP-DELR*VEL
      TRGE = TDET
  270 IF (NFOG.GT.0) GO TO 830
  280 CONTINUE
C
C        IF KFLAG IS 0 NO SEARCH RADARS ARE UP   GO CHECK
C        DIRECTORS FOR HORIZON SEARCH MODE
C
```

```
      IF (KFLAG.EQ.0) GO TO 610
      IF (TRGE.LT.TA) GO TO 290
      IF (MFOG.GT.0) GO TO 830
      NONO(IR,I) = 2
      GO TO 730
  290 RGFC = 0.
C
C        INCRIMENT THE ENGAGEMENTS POSSIBLE AND RAIDERS ENGAGABLE
C        COUNTERS, THEN CHECK TO SEE IF DETECTION OCCURED.. IF
C        IT DID, GET THE TARGET PROCESSING DELAYS.
C
      JACK(I,5) = JACK(I,5)+1
      STATS(I,1) = STATS(I,1)+RGE
C
C        DETERMINATION OF FC RADAR  ACQUISITION RANGE.
C
      DO 380 NB=1,4
      MTY = IFIX(X(ISI-1+NB)+AMF)
      IF (MTY.EQ.0) GO TO 390
      MTQ = NB+15
      IF (NADAR(I,MTQ).EQ.0) GO TO 380
C
      DO 370 NF=1,3
      MTZ = (MTY-1)*40
      NPQ = 3+(NB-1)*3+NF
      NAD = NADAR(I,NPQ)+1
      GO TO (370,300,340), NAD
  300 IA = IFIX(X(516+NF*2+MTZ*40)+AMF)
      SRFC = X(902+(IA-1)*20)*AREA
      FCRH = X(81)*SQRT(X(ISI+9+NB))
      HRFC = FCRH+SALT
      THR = X(905+(IA-1)*20)
      FRG = AMIN1(SRFC,HRFC,THR)
      IF (NJ.EQ.0) GO TO 310
      IJ = IFIX(X(907+(IA-1)*20)+AMF)
      IF (JAM(IJ).EQ.0) GO TO 310
      CALL JAMM (FRG,I,IR,IJ,IA,VEL,FCRH,AREA,0,JCPA,RGF)
      IF (JCPA.EQ.0) GO TO 370
      RFCJ = RGF
      GO TO 320
  310 IF (RCPS.GE.FRG**2) GO TO 370
      RGF = FRG
      RFCJ = 999.
  320 IF (X(57).EQ.0.) GO TO 330
      WRITE (6,950) I,IR,NB,NF
      WRITE (6,940) SRFC,RFCJ,HRFC
  330 IF (RGFC.GE.RGF) GO TO 370
      RGFC = RGF
```

```
      IFC = NB
      IFA = NF
      GO TO 370
  340 IF (ALT.GT.X(77)) GO TO 350
      RGM = X(508+MTZ)
      SVEL = X(509+MTZ)
      GO TO 360
  350 RGM = X(501+MTZ)
      SVEL = X(502+MTZ)
  360 RGF = RGM*(1+VEL/SVEL)
      IF (RGFC.GE.RGF) GO TO 380
      IFC = NB
      IFA = 0
      RGFC = RGF
      GO TO 380
  370 CONTINUE
C
  380 CONTINUE
C
C        CHECK TO SEE IF TARGET WILL COME WITHIN THE MAX
C        FC RADAR ACQUISITION RANGE.
C
  390 CALL TMDASM (IR,0,IFC,I,VEL,RGFC,TIME,TFR,XI,YI,RIP,TIML,IR)
      IF (X(3).EQ.0.) GO TO 410
      IF (JFOG.EQ.1) GO TO 400
      WRITE (6,960) IR,I,TRGE,TFR
      GO TO 410
  400 WRITE (6,970) IR,I,TRGE,TFR
  410 PROB = X(ISI+15)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      IF (PROB.GT.Z) GO TO 420
      NONO(IR,I) = 3
      GO TO 730
  420 IF (TA.LT.TFR) GO TO 810
      KEY = IFIX(X(ISI+16)+AMF)
      CALL REACT (2,KEY,TWDS)
      KEY = IFIX(X(ISI+17)+AMF)
      CALL REACT (3,KEY,TDES)
      MTY = IFIX(X(ISI-1+IFC)+AMF)
      MTZ = (MTY-1)*40
      IF (IFA.EQ.0) GO TO 430
      IA = IFIX(X(516+IFA*2+MTZ)+AMF)
      KEY = IFIX(X(906+(IA-1)*20)+AMF)
      CALL REACT (4,KEY,TAC)
      TSR = TRGE+TWDS+TDES+TAC
      GO TO 440
  430 TSR = TRGE+TWDS+TDES
```

```
  440 IF (ALT.GT.X(77)) GO TO 450
      SVEL = X(508+MTZ)
      GO TO 460
  450 SVEL = X(502+MTZ)
  460 TLS = TIML-RGFC/SVEL
C
C        COMPARE LAST TIME TO FIRE TO FINAL DETECT PLUS PROCESS TIME.
C
      IF (TSR.LT.TLS) GO TO 470
      IF (MFOG.GT.0) GO TO 830
      NONO(IR,I) = 5
      GO TO 730
C
C        GET LAUNCH TIME AND COMPARE WITH LAST TIME TO FIRE.
C
  470 KEY = IFIX(X(ISI+18)+AMF)
      CALL REACT (5,KEY,TFIRE)
      IF (TRGE.GT.TFR) GO TO 480
      TSHOOT = TFR+TFIRE
      GO TO 490
  480 TSHOOT = TSR+TFIRE
  490 IF (X(3).EQ.0.) GO TO 500
      WRITE (6,980) IR,I,TSHOOT,DELR,TWDS
      WRITE (6,990) TDES,TFIRE
  500 IF (TSHOOT.LT.TLS) GO TO 510
      IF (MFOG.GT.0) GO TO 830
      NONO(IR,I) = 5
      GO TO 730
  510 IF (NASA.EQ.0) GO TO 560
C
C        ENTER A SHOOT EVENT FOR EACH OPERATIONAL DIRECTOR LAUNCHER
C        COMBINATION IN THIS BATTERY.
C
C        IF THE MISSILE FLYS INDEPENDENTLY OF THE DIRECTOR THROUGHOUT
C        ITS FLIGHT, ASSIGN THE NUMBER 7 TO THE CHANNEL SLOT.
C
      DO 540 MS=1,4
      MJL = 15+MS
      IF (NADAR(I,MJL).EQ.0) GO TO 540
      ID = IFIX(X(ISI+MS-1)+AMF)
      IF (X(536+(ID-1)*40).EQ.1.) GO TO 540
      MP = 3+(MS-1)*3
C
      DO 530 MT=1,3
      MPI = MP+MT
      LARRY = NADAR(I,MPI)+1
      GO TO (530,520,550,530), LARRY
  520 KS = I*1000+MS*100+MT*10+IR*100000
```

```
      CALL STOREV (TSHOOT,3,KS)
  530 CONTINUE
C
  540 CONTINUE
C
      GO TO 730
  550 KS = I*1000+MS*100+MT*10+IR*100000+7
      CALL STOREV (TSHOOT,3,KS)
      GO TO 540
C
  560 DO 590 MS=1,4
      MJL = 15+MS
      IF (NADAR(I,MJL).EQ.0) GO TO 590
      ID = IFIX(X(ISI+MS-1)+AMF)
      IF (X(536+(ID-1)*40).EQ.0.) GO TO 590
      MP = 3+(MS-1)*3
C
      DO 580 MT=1,3
      MPI = MP+MT
      LOIS = NADAR(I,MPI)+1
      GO TO (580,570,600,580), LOIS
  570 KS = I*1000+MS*100+MT*10+IR*100000
      CALL STOREV (TSHOOT,3,KS)
  580 CONTINUE
C
  590 CONTINUE
C
      GO TO 730
  600 KS = I*1000+MS*100+MT*10+IR*100000+7
      CALL STOREV (TSHOOT,3,KS)
      GO TO 590
C
C        THE HORIZON SEARCH DETECTION ALGORITHM
C
  610 RGFC = 0.
C
      DO 660 NB=1,4
      IFCR = IFIX(X(ISI-1+NB)+AMF)
      IF (IFCR.EQ.0) GO TO 660
      ICH = (NB-1)*3+3
C
      DO 650 NF=1,3
      NFG = ICH+NF
      IF (NADAR(K,NFG).NE.3) GO TO 650
      HT = X(ISI+10+NF)
      HFCSR = X(81)*SQRT(HT)
      HORZ = HFCSR+SALT
      ICL = IFIX(X(516+(IFCR-1)*40+2*NF)+AMF)
```

```
      RRFC = X(902+(ICL-1)*20)*AREA
      THR = X(905+(ICL-1)*20)
      HSFR = AMIN1(RRFC,HORZ,THR)
      IF (NJ.EQ.0) GO TO 620
      IJ = IFIX(X(907+(ICL-1)*20)+AMF)
      IF (JAM(IJ).EQ.0) GO TO 620
      CALL JAMM (HSFR,I,IR,IJ,ICL,VEL,HFCSR,AREA,0,LCPA,RGF)
      IF (LCPA.EQ.0) GO TO 650
      RFCJ = RGF
      GO TO 630
  620 IF (RCPS.GE.HSFR**2) GO TO 650
      RGF = HSFR
      RFCJ = 999.
  630 IF (X(57).EQ.0.) GO TO 640
      WRITE (6,1000) I,IR
      WRITE (6,940) RRFC,RFCJ,HORZ
  640 IF (RGF.LT.RGFC) GO TO 650
      RGFC = RGF
      IFC = NB
      IFA = NF
  650 CONTINUE
C
  660 CONTINUE
C
C        THE HORIZON SEARCH DELAY TO FIRE REACTION TIMES ASSESSMENT
C
      JACK(I,5) = JACK(I,5)+1
      MTY = IFIX(X(ISI-1+IFC)+AMF)
      MTZ = (MTY-1)*40
      KEY = IFIX(X(516+(IA-1)*20)+AMF)
      CALL REACT (6,KEY,TAC)
      KEY = IFIX(X(ISI+18)+AMF)
      CALL REACT (5,KEY,TFIRE)
      KEY = IFIX(X(ISI+19)+AMF)
      CALL REACT (7,KEY,THSCAN)
      CALL TMDASM (IR,0,IFC,I,VEL,RGFC,TIME,TD,XI,YI,RIP,TIML,IR)
      TSHOOT = TD+TAC+AMAX1(TFIRE,THSCAN)
      IF (TSHOOT.GT.X(32)) GO TO 670
      STATS(I,1) = STATS(I,1)+RIP
      IF (NASA.EQ.0) GO TO 680
      IF (ALT.GT.X(77)) GO TO 680
      SVEL = X(508+MTZ)
      GO TO 690
  670 NONO(IR,I) = 4
      GO TO 730
  680 SVEL = X(502+MTZ)
  690 TLS = TIML-RGFC/SVEL
      IF (TSHOOT.LE.TLS) GO TO 700
```

```
      NCNO(IR,I) = 5
      GO TO 730
  700 IF (NASA.EQ.0) GO TO 760
C
      DO 720 MS=1,4
      IPQ = 15+MS
      IF (NADAR(I,IPQ).EQ.0) GO TO 720
      ID = IFIX(X(ISI-1+MS)+AMF)
      IF (X(536+(ID-1)*40).EQ.1.) GO TO 720
      MP = 3+(MS-1)*3
C
      DO 710 MT=1,3
      MPI = MP+MT
      IF (NADAR(I,MPI).NE.3) GO TO 710
      KS = I*1000+MS*100+MT*10+IR*100000
      CALL STOREV (TSHOOT,3,KS)
  710 CONTINUE
C
  720 CONTINUE
C
  730 CONTINUE
C
  740 IF (X(55).EQ.0.) GO TO 750
      CALL GETIMX (IET)
      TIM = IET*.000026
      WRITE (6,1010) TIM
  750 RETURN
C
C        ONE OF THE SURFACE TO SURFACE  SHOOT ENTRIES.
C
  760 DO 780 MS=1,4
      IPQ = 15+MS
      IF (NADAR(I,IPQ).EQ.0) GO TO 780
      ID = IFIX(X(ISI-1+MS)+AMF)
      IF (X(536+(ID-1)*40).EQ.0.) GO TO 730
      MP = 3+(MS-1)*3
C
      DO 770 MT=1,3
      MPI = MP+MT
      IF (NADAR(I,MPI).NE.3) GO TO 770
      KS = I*1000+MS*100+MT*10+IR*100000
      CALL STOREV (TSHOOT,3,KS)
  770 CONTINUE
C
  780 CONTINUE
C
      GO TO 730
C
```

```
C          THE BI-STATIC DETECTION ALGORITHM
C
  790 CALL BIDET (I,NSR,ALT,IR,VEL,SIGMA,ND,TI,TIML,RGI,TD2,TIML2,R2,0)
      IF (TI.LT.TIME) TI = TIME
      IF (ND.EQ.0) GO TO 800
      IF (X(3).EQ.1.) WRITE (6,1020) IR,I,TI,TIML
      IF (ND.EQ.1) GO TO 800
      IF (X(3).EQ.1.) WRITE (6,1030) TD2.TIML2
  800 ND = ND+1
      GO TO (820,230,870), ND
  810 NONO(IR,I) = 4
      GO TO 730
  820 TI = 9120.
      TIML = 0.
      RGI = 0.
      GO TO 230
  830 RGE = 0.
      TRGE = 1000.
      TA = 0.
      GO TO (840,850,860), MFOG
  840 TI = TTA
      TIML = TIMLA
      RGI = RGIA
      MFOG = MFOG-1
      JFOG = 1
      NFOG = 0
      GO TO 230
  850 TI = TTB
      TIML = TIMLB
      RGI = RGIB
      MFOG = MFOG-1
      GO TO 230
  860 TI = TTC
      RGI = RGIC
      TIML = TIMLC
      MFOG = MFOG-1
      GO TO 230
  870 MFOG = MFOG+1
      GO TO (880,890,900), MFOG
  880 TTA = TD2
      TIMLA = TIML2
      RGIA = R2
      GO TO 230
  890 TTB = TD2
      TIMLB = TIML2
      NFOG = 1
      RGIB = R2
      GO TO 230
```

```
  900 TTC = TD2
      TIMLC = TIML2
      RGIC = R2
      GO TO 230
C
  910 FORMAT ('    SUBE1 IN AT',F15.8)
  920 FORMAT ('0','     RAIDER',I3,' SURFACE OR SUBMARINE UNIT, SHIP',I3,
     1' HAS NO SSM')
  930 FORMAT ('0',' DETECTION RANGES OF SHIP',I3,' ON RAIDER',I3,' WITH
     1RADAR',I2)
  940 FORMAT ('    CLEAR ENVIRONMENT... ',F9.4/('    JAMMING ENVIRONMENT.
     1',F9.4/('    RADAR HORIZON....... ',F9.4)))
  950 FORMAT (' AQUISITION RANGES FOR SHIP',I3,' ON RAIDER',I3,' WITH SY
     1STEM',I2,' FC RADAR TYPE',I3)
  960 FORMAT (' INITIAL DETECTION OF RAIDER',I3,' BY SHIP',I3/('    SEARC
     1H RADAR',F9.4,4X,'FC LOCK ON',F9.4))
  970 FORMAT ('  SECOND DETECTION (BI-STATIC) ON RAIDER',I3,' BY SHIP',I
     13/('      SEARCH RADAR DECTECTION TIME',F9.4,' LAST TIME TO SHOOT',
     2F9.4))
  980 FORMAT (' RAIDER',I3,' FIRING SHIP',I3,' TSHOOT WAS',F9.4/('    DET
     1ECTION DELAY WAS',F9.4/('    WDE TRACK DELAY WAS',F9.4)))
  990 FORMAT ('    DESIGNATION DELAY WAS',F9.4/('    FIRING DELAY WAS',F9.
     14))
 1000 FORMAT ('    HORIZON SEARCH DETECTION FOR SHIP',I3,' VS. RAIDER',I3
     1)
 1010 FORMAT ('    SUBE1 OUT AT',F15.8)
 1020 FORMAT ('0','  FIRST BI-STAT DET ON RAIDER',I3,' BY SHIP',I3,' WAS
     1 AT',F9.4,'   DET LOST AT',F9.4)
 1030 FORMAT ('      SECOND DET AT',F9.4,'  LOST DET AT',F9.4)
      END


      SUBROUTINE SUBE2 (TIME,IR)
C
C         THIS SUBROUTINE COMPUTES THE LAUNCH TIME FOR ALL RED
C         ASM/SSM AND DETERMINES THE POSSIBLE SAM LAUNCH TIMES FOR
C         BLUE MISSILE LAUNCHERS
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      DIMENSION TTARG(40,2)
      KING = 0
      NJ = IFIX(X(17)+AMF)
```

```
      IFLAG = 0
      KATHY = 0
      JFLAG = 0
      IF (X(55).EQ.0.) GO TO 20
      CALL GETIMX (IET)
      T = .000026*IET
      WRITE (6,1100) T
   20 NG = RAID(IR,13)
      IF (NG.EQ.0) GO TO 910
C
C        CHECK TO SEE IF RAIDER IS STILL VIABLE
C
C        IF THIS IS THE FIRST ASM TO BE FIRED, SET THE TIMER TMAX.
C
      IF (IFLASK.GT.0) GO TO 30
      TMAX = TIME+X(84)
      IFLASK = 1
   30 ID = 2201+(IR-1)*6
      IJ = IFIX(X(ID)+AMF)
      ICOP = (IJ-1)*20
      NONE = 1
      VEL = X(1814+ICOP)
      IF (X(ID+3).LE.50.) GO TO 1040
      IF (TIME.GE.TMAX) GO TO 1080
C
C        DETERMINE THE RAIDERS LAUNCH INTERVAL FOR THIS SERIES OF
C        MISSILE FIRINGS.
C
      TINT = TIME+X(83)
      IFLASK = 2
      IF (TINT.GT.TMAX) TINT=TMAX
      W = TINT-TIME
      NTOT = MISLFT(IR,1)+MISLFT(IR,2)
C
C        SURFACE TO SURFACE TARGETING DIVERSION
C
      IF (NTOT.EQ.0) GO TO 910
C
C        DETERMINE THE NUMBER AND TYPE OF MISSILES TO BE FIRED
C        FIRST, AND THOSE, IF ANY, TO BE FIRED SECOND.
C
      CALL RANDOM (ISEED,QRT,NTOT)
      X(13) = X(13)+FLOAT(NTOT)
      CALL TJSORT (QRT,NTOT)
   40 LNUM = IFIX(RAID(IR,11)+AMF)
      J = IFIX(X(1804+ICOP)+AMF)
      IF (LNUM.NE.J) GO TO 50
      NCW = 1
```

```
      NEXT = 2
      NMIS = MISLFT(IR,1)
      IF (NMIS.EQ.0) GO TO 50
      GO TO 60
   50 J = IFIX(X(1806+ICOP)+AMF)
      NMIS = MISLFT(IR,2)
      NEXT = 1
      NOW = 2
      IF (NMIS.NE.0) GO TO 60
      J = X(1804+ICOP)
      NMIS = MISLFT(IR,1)
      NOW = 1
C
C        GET THE ASM'S CHARACTERISTICS
C
   60 IK = 2001+(J-1)*10
      SIGMA = X(IK+5)
      SPD = X(IK)
      AREA = X(IK+9)
      ALT = X(IK+1)
      SALT = X(IK+8)
      RELI = X(IK+3)
C
      DO 870 I=1,NMIS
      KAT = JFLAG+1
      GO TO (70,1070,1050), KAT
   70 IF (I.GT.1) GO TO 80
      IF (KATHY.GT.0) GO TO 80
      TM = TIME
      GO TO 100
C
C        GET THE PROPOSE LAUNCH TIME
C        CHOOSE A TARGET FOR THE CANDIDATE ASM
C
   80 TX = W*QRT(NONE)
      NONE = NONE+1
   90 TM = TIME+TX
  100 CALL LOCTIM (IR,TM,5,XU,Y)
      IPQ = NRTGT(IR)
      IDP = 0
C
C        DETERMINE WHICH OF THE TARGETS ON THE RAIDERS
C        TARGET LIST CAN BE TARGETED AND FILL IN THE TEMPORARY
C        TARGET LIST.
C
      DO 140 NT=1,NS
      IF (NT.GT.1) GO TO 110
      RT = PLIST(NT,IPQ)
```

```
      GO TO 120
  110 RT = PLIST(NT,IPQ)-PLIST((NT-1),IPQ)
  120 IF (RT.LT.0.0005) GO TO 140
      IF (SHIP(NT,6).GT.0.) GO TO 130
      IF (X(5).EQ.1.) GO TO 130
      MARIE = IFIX(SHIP(NT,4)+AMF)
      IF (IHIT(NT).GE.MARIE) GO TO 140
  130 CALL RCHECK (NT,IR,KQED)
      IF (KQED.EQ.1) GO TO 140
      IDP = IDP+1
      TTARG(IDP,1) = FLOAT(NT)
      TTARG(IDP,2) = PLIST(NT,IPQ)
  140 CONTINUE
C
  150 IF (IDP.EQ.0) GO TO 180
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
C
      DO 160 NT=1,IDP
      IF (Z.LE.TTARG(NT,2)) GO TO 170
  160 CONTINUE
C
      NT = IDP
      IOU = IFIX(TTARG(IDP,1)+AMF)
      GO TO 190
  170 IOU = IFIX(TTARG(NT,1)+AMF)
      GO TO 190
  180 CALL STOREV ((TM+3.),2,IR)
      IF (NASM.EQ.0) IFLASK=0
      GO TO 910
C
C         CHECK TO SEE IF THE TARGET CHOSEN IS A DESIGNATED
C         NON-RADIATING PICKET
C
  190 ISI = IFIX(SHIP(IOU,5)+AMF)
      L2 = IFIX(X(ISI+30)+AMF)
      IF (L2.EQ.0) GO TO 250
C
C         IF SO DETERMINE IF IT CAN BE TARGETED.   IF IT CAN,
C         PROCEED, OTHERWISE ELIMINATE IT FROM THE TEMPORARY TARGET
C         LIST AND TRY AGAIN.
C
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      IF (RAID(IR,9).GT.X(20)) GO TO 200
      IF (Z.LE.X(22)) GO TO 240
      GO TO 210
  200 IF (Z.LE.X(21)) GO TO 240
```

```
  210 IF (IDP.NE.IOU) GO TO 220
      IDP = IDP-1
      GO TO 150
  220 IF (NT.EQ.1) GO TO 230
      L2 = NT-1
      TTARG(NT,2) = TTARG(L2,2)
      GO TO 150
  230 TTARG(1,2) = 0.
      GO TO 150
  240 TRGE = X(23)**2
      DELTAX = ABS(SHIP(IOU,2)-RAID(IR,3))
      IF (DELTAX.LT.5.0E-5) DELTAX=5.0E-5
      DELTAY = ABS(SHIP(IOU,3)-RAID(IR,4))
      IF (DELTAY.LT.5.0E-5) DELTAY=5.0E-5
      TA = DELTAY**2+DELTAX**2
      IF (TA.LE.TRGE) GO TO 250
      GO TO 210
  250 NASM = NASM+1
  260 NTOT = NTOT-1
      MISLFT(IR,NOW) = MISLFT(IR,NOW)-1
C
C        FILL OUT THE ASM TABLE AND CHECK FOR SUCCESSFUL LAUNCH.
C
      M = NASM+90
      ASMT(NASM,3) = FLOAT(IR)
      ASMT(NASM,4) = FLOAT(IOU)
      ASMT(NASM,5) = TM
      ASMT(NASM,6) = FLOAT(J)
      ASMT(NASM,7) = XU
      ASMT(NASM,8) = Y
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      IF (Z.LT.RELI) GO TO 280
      IF (X(3).EQ.1.) WRITE (6,1090) IR,NASM
      ASMT(NASM,1) = 0.
C
      DO 270 K=1,NS
      NONO(M,K) = 1
  270 CONTINUE
C
      DELTAX = ABS(SHIP(IOU,2)-RAID(IR,3))
      DELTAY = ABS(SHIP(IOU,3)-RAID(IR,4))
      IF (DELTAX.LT.AMF) DELTAX=AMF
      IF (DELTAY.LT.AMF) DELTAY=AMF
      ASMT(NASM,2) = TM+SQRT(DELTAY**2+DELTAX**2+(ALT/6000.)**2)/SPD
      IF (IFLAG.EQ.1) GO TO 1060
      IF (KING.EQ.1) IFLAG=1
C
```

```
C          FLAG FOR TWO MISSILE SALVOS FOR SSN
C
      GO TO 870
  280 XS = SHIP(IOU,2)
      ASMT(NASM,1) = 1.
C
C          DETERMINE THE ASM/SSM FLIGHT TIME AND DETERMINE THE
C          IMPACT TIME.
C
      RAID(IR,14) = XS
      YS = SHIP(IOU,3)
      RAID(IR,15) = YS
      XR = RAID(IR,3)
      YR = RAID(IR,4)
      DELTAX = ABS(XS-XR)
      IF (DELTAX.LT..00005) DELTAX=.00005
      DELTAY = ABS(YS-YR)
      IF (DELTAY.LT..00005) DELTAY=.00005
      DIST = SQRT(DELTAX**2+DELTAY**2+(ALT/6000.)**2)
      ASMT(NASM,2) = TM+DIST/SPD
C
C           STORE AN ASM IMPACT EVENT
C
      NU = M*100+IOU
      CALL STOREV (ASMT(NASM,2),4,NU)
      ALTN2 = (ALT/6000.)**2
C
C          ASM/SSM DETECTION AND FIRING ALGORITHM
C
      DO 860 K=1,NS
      NM = IFIX(SHIP(K,6)+AMF)
      IF (NM.GT.0) GO TO 290
      NONO(M,K) = 7
      GO TO 860
  290 ISI = IFIX(SHIP(K,5)+AMF)
      RGE = 0.
C
C          GET MAX MISSILE RANGE FOR SAM SHIP
C
      DO 340 LO=1,4
      MP = 3+(LO-1)*3
C
      DO 300 LI=1,3
      MPI = MP+LI
      IF (NADAR(K,MPI).GT.0) GO TO 310
  300 CONTINUE
C
      GO TO 340
```

```
  310 IF (NADAR(K,(15+LO)).EQ.0) GO TO 340
      IC = IFIX(X(ISI+LO-1)+AMF)
      IF (ALT.GT.X(77)) GO TO 320
      ICS = 508+(IC-1)*40
      GO TO 330
  320 ICS = 501+(IC-1)*40
  330 SRGE = X(ICS)
      SVEL = X(ICS+1)
      IF (SRGE.LE.RGE) GO TO 340
      RGE = SRGE
      L2 = LO
      SAMVEL = SVEL
  340 CONTINUE
C
C        WILL THE ASM/SSM COME WITHIN THE BLJE MISSILE ENVELOPE?
C
      CALL TMDASM (IR,IOU,L2,K,SPD,RGE,TM,TD,XI,YI,RIP,TIML,NASM)
      IF (TD.GE.9120.) GO TO 350
      IF (TD.LT.X(82)) GO TO 360
      NONO(M,K) = 8
      GO TO 860
  350 NONO(M,K) = 9
      GO TO 860
  360 IF (TD.LT.ASMT(NASM,2)) GO TO 370
      NONO(M,K) = 6
      GO TO 860
  370 TFIR = TIML-RGE/SAMVEL
C
C        IS TIME TO FIRE AFTER TIME OF MISSILE EXITING ENVALOPE?
C
      IF (TFIR.GT.TD) GO TO 380
      NONO(M,K) = 8
      GO TO 860
  380 KFLAG = 0
      TA = 0.
      TRGE = 1000.
      RGE = 0.
      JACK(K,1) = JACK(K,1)+1
      JACK(K,6) = JACK(K,6)+1
      NFOG = 0
      MFOG = 0
      JFOG = 0
C
C         DETERMINE CPA OF ASM TO RADAR UNIT
C
      CALL CPA (XU,Y,XS,YS,SHIP(K,2),SHIP(K,3),XCPA,YCPA)
      RCPS = (SHIP(K,2)-XCPA)**2+(SHIP(K,3)-YCPA)**2+ALTN2
C
```

```
      DO 460 NSR=1,3
      IF (NADAR(K,NSR).EQ.0) GO TO 460
      KFLAG = 1
      ID = IFIX(X(ISI+3+NSR)+AMF)
      IDI = (ID-1)*20
      IA = IFIX(X(901+IDI)+AMF)
C
C        BI-STATIC RADAR DETECTION MODE SWITCH
C
      IF (IA.EQ.0) GO TO 930
C
C        MONOSTATIC SEARCH RADAR DETECTION RANGE ALGORITHM
C
      SRAG = X(902+IDI)*AREA
      RRH = X(81)*SQRT(X(ISI+6+NSR))
      HRAG = RRH+SALT
      THR = X(905+IDI)
      RG = AMIN1(SRAG,HRAG,THR)
      IF (NJ.EQ.0) GO TO 390
      IJI = IFIX(X(907+IDI)+AMF)
      IF (JAM(IJI).EQ.0) GO TO 390
      CALL JAMM (RG,K,IR,IJI,ID,SPD,RRH,AREA,1,ICPA,RGI)
      IF (ICPA.EQ.0) GO TO 460
      RAGJ = RGI
      GO TO 400
  390 RGS = RG**2
      IF (RCPS.GE.RGS) GO TO 460
      RGI = RG
      RAGJ = 999.
  400 IF (X(3).EQ.0.) GO TO 410
      WRITE (6,1110) K,M,NSR
      WRITE (6,1120) SRAG,RAGJ,HRAG
  410 CALL TMDASM (IR,IOU,NSR,K,SPD,RGI,TM,TI,XI,YI,RIP,TIML,NASM)
  420 IF (TA.LT.TIML) TA = TIML
      IF (X(10).EQ.0.) GO TO 430
      DELR = (RIP*X(10))/SPD
      GO TO 440
C        GET DETECTION DELAYS, CHECK TIMES, GET PROBABILITY OF DE-
C        TECTION AND DETERMINE IF DETECTION OCCURRED.
C
  430 KEY = IFIX(X(906+IDI)+AMF)
      CALL REACT (1,KEY,DELR)
  440 TDET = TI+DELR
      IF (TRGE.LE.TDET) GO TO 450
      RGE = ABS(RIP-DELR*SPD)
      TRGE = TDET
  450 IF (NFOG.GT.0) GO TO 960
  460 CONTINUE
```

```
C
C        GO TO HORIZON SEARCH DETECTION MODE
C
      IF (KFLAG.EQ.0) GO TO 740
C
C        WILL ASM/SSM ENTER SEARCH RADAR RANGE?
C
      IF (TRGE.LT.TA) GO TO 470
      IF (MFOG.GT.0) GO TO 960
      NONO(M,K) = 2
      IF (X(3).EQ.1.) WRITE (6,1210) TA
      GO TO 860
  470 RGFC = 0.
      JACK(K,7) = JACK(K,7)+1
      STATS(K,2) = STATS(K,2)+RGE
C
C        FIRE CONTROL ACQUISITION PHASE
C
      DO 560 NB=1,4
      MTY = IFIX(X(ISI+NB-1)+AMF)
      IF (MTY.EQ.0) GO TO 570
      IF (X(536+(MTY-1)*40).EQ.1.) GO TO 560
      MTQ = NB+15
      IF (NADAR(K,MTQ).EQ.0) GO TO 560
C
      DO 550 NF=1,3
      NPQ = 3+(NB-1)*3+NF
      NAD = NADAR(K,NPQ)+1
      GO TO (550,480,520), NAD
  480 IA = IFIX(X(516+NF*2+(MTY-1)*40)+AMF)
      IAI = (IA-1)*20
      SRFC = X(902+IAI)*AREA
      THR = X(905+IAI)
      RRH = X(81)*SQRT(X(ISI+9+NB))
      HRAG = RRH+SALT
      RG = AMIN1(SRFC,HRAG,THR)
      IF (NJ.EQ.0) GO TO 490
      IJI = IFIX(X(907+IAI)+AMF)
      IF (JAM(IJI).EQ.0) GO TO 490
      CALL JAMM (RG,K,IR,IJI,IA,SPD,RRH,AREA,1,ICPA,RGF)
      IF (ICPA.EQ.0) GO TO 550
      RFCJ = RGI
      GO TO 500
  490 RGS = RG**2
      IF (RCPS.GE.RGS) GO TO 550
      RGF = RG
      RFCJ = 999.
  500 IF (X(3).EQ.0.) GO TO 510
```

```
      WRITE (6,1130) K,M,NB,NF
      WRITE (6,1120) SRFC,RFCJ,HRAG
  510 IF (RGFC.GE.RGF) GO TO 550
      RGFC = RGF
      IFC = NB
      IFA = NF
      GO TO 550
  520 MTZ = (MTY-1)*40
      IF (ALT.GT.X(77)) GO TO 530
      RGM = X(508+MTZ)
      SVEL = X(509+MTZ)
      GO TO 540
  530 RGM = X(501+MTZ)
      SVEL = X(502+MTZ)
  540 RGF = RGM*(1+SPD/SVEL)
      IF (RGFC.GE.RGF) GO TO 560
      IFC = NB
      RGFC = RGF
      IFA = 0
      GO TO 560
  550 CONTINUE
C
  560 CONTINUE
C
  570 CALL TMDASM (IR,IOU,IFC,K,SPD,RGFC,TM,TFR,XI,YI,RIP,TIMF,NASM)
      IF (X(3).EQ.0.) GO TO 590
      IF (JFOG.EQ.1) GO TO 580
      WRITE (6,1140) TRGE,TFR
      GO TO 590
  580 WRITE (6,1150) TRGE,TFR
  590 PROB = X(ISI+15)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      IF (PROB.GT.Z) GO TO 600
      NONO(M,K) = 3
      GO TO 860
C
C        DESIGNATION, EVALUATION, AND ACQUISITION DELAYS.
C
  600 IF (TA.LT.TFR) GO TO 940
      KEY = IFIX(X(ISI+16)+AMF)
      CALL REACT (2,KEY,TWDS)
      KEY = IFIX(X(ISI+17)+AMF)
      CALL REACT (3,KEY,TDES)
      MTY = IFIX(X(ISI+IFC-1)+AMF)
      MTZ = (MTY-1)*40
      IF (IFA.EQ.0) GO TO 610
      IA = IFIX(X(516+IFA*2+MTZ)+AMF)
```

```
      KEY = IFIX(X(906+(IA-1)*20)+AMF)
      CALL REACT (4,KEY,TAC)
      TSR = TRGE+TWDS+TDES+TAC
      GO TO 620
  610 TSR = TRGE+TWDS+TDES
  620 IF (ALT.GT.X(77)) GO TO 630
      SVEL = X(509+MTZ)
      GO TO 640
  630 SVEL = X(502+MTZ)
  640 TLS = TIMF-RGFC/SVEL
      IF (TSR.LT.TLS) GO TO 650
      IF (MFOG.GT.0) GO TO 960
      NONO(M,K) = 5
      GO TO 860
C
C        FIRE EVENT STORAGE FOR A REGULAR SHOT
C
  650 KEY = IFIX(X(ISI+18)+AMF)
      CALL REACT (5,KEY,TFIRE)
      IF (TRGE.GT.TFR) GO TO 660
      TSHOOT = TFR+TFIRE
      GO TO 670
  660 TSHOOT = TSR+TFIRE
  670 IF (X(57).EQ.0.) GO TO 680
      WRITE (6,1220) TSHOOT,DELR,TWDS
      WRITE (6,1230) TDES,TFIRE
  680 IF (TSHOOT.LT.TLS) GO TO 690
      IF (MFOG.GT.0) GO TO 960
      NONO(M,K) = 5
      GO TO 860
C
  690 DO 720 MS=1,4
      MJL = 15+MS
      IF (NADAR(K,MJL).EQ.0) GO TO 720
      ID = IFIX(X(ISI+MS-1)+AMF)
      IF (X(536+(ID-1)*40).EQ.1.) GO TO 720
      MP = 3+(MS-1)*3
C
      DO 710 MT=1,3
      MPI = MP+MT
      LARRY = NADAR(K,MPI)+1
      GO TO (710,700,730,710), LARRY
  700 KS3 = M*100000+K*1000+MS*100+MT*10
      CALL STOREV (TSHOOT,3,KS3)
  710 CONTINUE
C
  720 CONTINUE
C
```

```
      GO TO 860                                                         SU2 4970
C                                                                       SU2 4980
C        FIRE EVENT STORAGE FOR A NON DEDICATED SHOT                    SU2 4990
C                                                                       SU2 5000
  730 KS3 = M*100000+K*1000+MS*100+MT*10+7                              SU2 5010
      CALL STOREV (TSHOOT,3,KS3)                                        SU2 5020
      GO TO 720                                                         SU2 5030
  740 RGFC = 0.                                                         SU2 5040
C                                                                       SU2 5050
C        HORIZON SEARCH DETECTION ALGORITHM                             SU2 5060
C                                                                       SU2 5070
      DO 790 NB=1,4                                                     SU2 5080
      IF (X(ISI+NB-1).EQ.0.) GO TO 790                                  SU2 5090
      ICH = (NB-1)*3+3                                                  SU2 5100
C                                                                       SU2 5110
      DO 780 NF=1,3                                                     SU2 5120
      NFG = ICH+NF                                                      SU2 5130
      IF (NADAR(K,NFG).NE.3) GO TO 780                                  SU2 5140
      IFCR = IFIX(X(ISI+NB-1)+AMF)                                      SU2 5150
      ICL = IFIX(X(516+(IFCR-1)*40+2*NF)+AMF)                           SU2 5160
      ICI = (ICL-1)*20                                                  SU2 5170
      RRFC = X(902+ICI)*AREA                                            SU2 5180
      THR = X(905+ICI)                                                  SU2 5190
      HT = X(ISI+10+NF)                                                 SU2 5200
      RRH = X(81)*SQRT(HT)                                              SU2 5210
      HORZ = RRH+SALT                                                   SU2 5220
      RG = AMIN1(RRFC,HORZ,THR)                                         SU2 5230
      IF (NJ.EQ.0) GO TO 750                                            SU2 5240
      IJI = IFIX(X(907+ICI)+AMF)                                        SU2 5250
      IF (JAM(IJI).EQ.0) GO TO 750                                      SU2 5260
      CALL JAMM (RG,K,IR,IJI,ICL,SPD,RRH,AREA,1,ICPA,RGF)               SU2 5270
      IF (ICPA.EQ.0) GO TO 780                                          SU2 5280
      RFCJ = RGF                                                        SU2 5290
      GO TO 760                                                         SU2 5300
  750 RGS = RG**2                                                       SU2 5310
      IF (RCPS.GE.RGS) GO TO 780                                        SU2 5320
      RGF = RG                                                          SU2 5330
      RFCJ = 999.                                                       SU2 5340
  760 IF (X(57).EQ.0.) GO TO 770                                        SU2 5350
      WRITE (6,1160) K,M                                                SU2 5360
      WRITE (6,1120) RRFC,RFCJ,HORZ                                     SU2 5370
  770 IF (RGF.LE.RGFC) GO TO 780                                        SU2 5380
      IF (RGF.LE.RGFC) GO TO 780                                        SU2 5390
      RGFC = RGF                                                        SU2 5400
      IFC = NB                                                          SU2 5410
      IFA = NF                                                          SU2 5420
  780 CONTINUE                                                          SU2 5430
C                                                                       SU2 5440
```

```
  790 CONTINUE
C
      MTY = IFIX(X(ISI-1+IFC)+AMF)
      MTZ = (MTY-1)*40
      IA = IFIX(X(516+IFA*2+MTZ)+AMF)
C
C        HORIZON SEARCH DECISION AND FIRE DELAYS
C
      KEY = IFIX(X(908+(IA-1)*20)+AMF)
      CALL REACT (6,KEY,TAC)
      KEY = IFIX(X(ISI+18)+AMF)
      CALL REACT (5,KEY,TFIRE)
      KEY = IFIX(X(ISI+19)+AMF)
      CALL REACT (7,KEY,THSCAN)
      CALL TMDASM (IR,IOU,IFC,K,SPD,RGFC,TM,TIFA,XI,YI,RIP,TIML,NASM)
      TSHOOT = TIFA+TAC+AMAX1(TFIRE,THSCAN)
      IF (TSHOOT.GT.X(82)) GO TO 800
      STATS(K,2) = STATS(K,2)+RIP
      JACK(K,7) = JACK(K,7)+1
      IF (ALT.GT.X(77)) GO TO 810
      SVEL = X(508+MTZ)
      GO TO 820
  800 NCNO(M,K) = 4
      GO TO 860
  810 SVEL = X(502+MTZ)
  820 TLS = TIML-RGFC/SVEL
      IF (TSHOOT.LE.TLS) GO TO 830
      NCNO(M,K) = 5
      GO TO 860
C
C        HORIZON SEARCH FIRE
C
  830 DO 850 MS=1,4
      IPQ = 15+MS
      IF (NADAR(K,IPQ).EQ.0) GO TO 850
      ID = IFIX(X(ISI+MS-1)+AMF)
      IF (X(536+(ID-1)*40).EQ.1.) GO TO 850
      MP = 3+(MS-1)*3
C
      DO 840 MT=1,3
      MPI = MP+MT
      IF (NADAR(K,MPI).NE.3) GO TO 840
      KS3 = M*100000+K*1000+MS*100+MT*10
      CALL STOREV (TSHOOT,3,KS3)
  840 CONTINUE
C
  850 CONTINUE
C
```

```
  860 CONTINUE
C
      IF (IFLAG.EQ.1) GO TO 1060
      IF (KING.EQ.1) IFLAG=1
  870 CONTINUE
C
C        CHECK TO SEE IF THE RAIDER HAS ANOTHER TYPE OF MISSILE
C
      IF (NTOT.EQ.0) GO TO 900
      IF (NEXT.EQ.1) GO TO 880
      IF (RAID(IR,16).GT.TM) GO TO 890
      NMIS = MISLFT(IR,2)
      NOW = 2
      KATHY = 1
      GO TO 60
  880 IF (RAID(IR,16).GT.TM) GO TO 890
      NMIS = MISLFT(IR,1)
      NOW = 1
      KATHY = 1
      GO TO 60
  890 IF (RAID(IR,16).GT.X(82)) GO TO 900
C
C        IF THE RAIDER IS OUTSIDE THE LAUNCH RANGE FOR THE TYPE
C        TWO MISSILES, STORE A MISSILE (ASM/SSM) LAUNCH EVENT
C
      CALL STOREV (RAID(IR,16),2,IR)
C
C        IF  FLAG SET PRINT THE NONO TABLE
C
  900 IF (X(58).EQ.0.) GO TO 910
      WRITE (6,1190)
      NMIS = 90+NASM
      WRITE (6,1200) (MFOG,(NONO(MFOG,NFOG),NFOG=1,40),MFOG=1,NR)
      WRITE (6,1200) (MFOG,(NONO(MFOG,NFOG),NFOG=1,40),MFOG=91,NMIS)
  910 IF (X(55).EQ.0.) GO TO 920
      CALL GETIMX (IET)
      T = .000026*IET
      WRITE (6,1170) T
  920 RETURN
C        THE BI-STATIC DETECTION ALGORITHM
C
  930 CALL BIDET (K,NSR,ALT,IR,SPD,SIGMA,ND,TI,TIML,RGI,TD2,TIML2,R2,1)
      ND = ND+1
      GO TO (950,420,1000), ND
  940 NONO(M,K) = 4
      GO TO 860
  950 TI = 9120.
      TIML = 0.
```

```
      TA = 0.
      GO TO (970,980,990), MFOG
  970 TI = TTA
      TIML = TIMLA
      RGI = RGIA
      MFOG = MFOG-1
      JFOG = 1
      NFOG = 0
      GO TO 420
  980 TI = TTB
      TIML = TIMLB
      RGI = RGIB
      MFOG = MFOG-1
      GO TO 420
  990 TI = TTC
      RGI = RGIC
      TIML = TIMLC
      MFOG = MFOG-1
      GO TO 420
 1000 MFOG = MFOG+1
      GO TO (1010,1020,1030), MFOG
 1010 TTA = TD2
      TIMLA = TIML2
      RGIA = R2
      GO TO 420
 1020 TTB = TD2
      TIMLB = TIML2
      NFOG = 1
      RGIB = R2
      GO TO 420
 1030 TTC = TD2
      TIMLC = TIML2
      RGIC = R2
      GO TO 420
 1040 IF (IFLASK.EQ.1) IFLASK=0
      NTOT = MISLFT(IR,1)+MISLFT(IR,2)
      JFLAG = 1
      ID = IFIX(X(2201+(IR-1)*6)+AMF)
      KING = IFIX(X(1808+(ID-1)*20)+AMF)
      IF (KING.EQ.1) IFLAG=1
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      TX = X(86)*Z
      GO TO 40
```

```
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
C
      MFOG = 0
      NC1 = 0
      REASON = 0.
      IF (X(55).EQ.0.) GO TO 20
      CALL GETIMX (K)
      TIM = K*0.000026
      WRITE (6,610) TIM
C
C        DETERMINE THE TARGET NUMBER, FIRING SHIP,BATTERY,DIRECTOR,
C        AND CHANNEL NUMBER FOR THE PROPOSED MISSILE FIRING.
C
   20 IU = IU+2
      M = IU*1.0D-5
      KP = M*100000
      ID = IU-KP
      K = ID*1.0D-3
      KP = K*1000
      JP = ID-KP
      NMS = JP*0.0100
      KP = NMS*100
      ID = JP-KP
      NB = ID*0.10
      KP = NB*10
      NC = ID-KP
      NC = NC-2
      IU = IU-2
C
C        IF THE FIRING SHIP HAS BEEN SUNK,EXIT
C
      IF (SHIP(K,6).NE.0.) GO TO 30
      REASON = 1.
      GO TO 560
C
C        IF THE FIRING SHIP IS OUT OF MISSILES, EXIT
C
   30 IF (IBAT(K,NMS).NE.0) GO TO 40
      REASON = 4.
      GO TO 560
C
C        IF THE TARGET HAS ALREADY BEEN SUNK OR SHOT DOWN, EXIT
```

```
C
C      IF THE RAIDER WAS AN AIRCRAFT CARRYING ASMS AND ALL ITS
C      ASMS HAVE BEEN LAUNCHED, AND THE FLAG IS SET, EXIT
C
   40 IF (M.GT.90) GO TO 60
      IF (RAID(M,13).NE.0.) GO TO 70
   50 REASON = 3.
      GO TO 560
   60 JULIA = M-90
      IF (ASMT(JULIA,1).NE.1.) GO TO 50
C
C      GET THE MAXIMUM MISSILE RANGE FOR THIS BATTERY AND CHECK
C      TO SEE IF THERE ARE ANY TARGETS MORE THREATENING WHICH
C      IT IS CAPABLE OF HANDLING
C
   70 ID = IFIX(SHIP(K,5)+AMF)
      ID = IFIX(X(ID+NMS-1)+AMF)
      KING = (ID-1)*40
      RGE = AMAX1(X(501+KING),X(508+KING))
      IF (M.GT.90) GO TO 80
      CALL LOCTIM (M,TIME,5,XP,YP)
      CALL CPA (XP,YP,RAID(M,5),RAID(M,6),SHIP(K,2),SHIP(K,3),RJL,EBTJR)
      GO TO 90
   80 LML = IFIX(ASMT(JULIA,4)+AMF)
      CALL ASMTIM (M,TIME,XP,YP)
      CALL CPA (XP,YP,SHIP(LML,2),SHIP(LML,3),SHIP(K,2),SHIP(K,3),RJL,EB
     1TJR)
   90 TT = (RJL-SHIP(K,2))**2+(EBTJR-SHIP(K,3))**2
      IF (TT.LE.RGE**2) GO TO 100
      REASON = 13.
      GO TO 560
  100 TT = X(532+KING)
      CALL PRIOR (M,K,NMS,NB,RGE,NC,TIME,M1,NC1,XP,YP)
C
C      IS NEW TARGET THE SAME AS THE OLD, OR HAS NEW BEEN DESTROYED?
C
      IF (M1.EQ.M) GO TO 120
      IF (M1.GT.90) GO TO 110
      IF (RAID(M1,13).EQ.0.) GO TO 50
      GO TO 120
  110 JULIA = M1-90
      IF (ASMT(JULIA,1).NE.1.) GO TO 50
C
C      IF THERE IS COORDINATION,CHECK TO SEE IF THE TARGET IS
C      AVAILABLE
C
  120 IF (X(4).EQ.0.) GO TO 140
      KP = IFIX(SHIP(K,5)+AMF)
      IF (X(KP+14).EQ.0.) GO TO 140
```

```
      IF (TIME.GT.TGTAV(M1)) GO TO 140
      REASON = 2.
      T = TGTAV(M1)+1.0E-3
  130 KP = M1*100000+K*1000+NMS*100+NB*10+NC1
      CALL STOREV (T,3,KP)
      GO TO 560
  140 IF (NONO(M1,K).EQ.0) GO TO 150
      REASON = 8.
      GO TO 560
C
C        FLAG ICOT SET TO 1 IF A)  CHANNEL PREVIOUSLY ASSIGNED AND
C        THAT CHANNEL IS BUSY,    B) NO CHANNEL PREVIOUSLY ASSIGNED
C        ALL CHANNELS ARE BUSY, EARLIEST AVAILABLE ASSIGNED, AND THE
C        MISSILE ONLY REQUIRES THE DIRECTOR FOR THE FINAL PHASE
C        OF THE INTERCEPT.
C        THE FLAG ICOT IS SET TO 0 IF THE MISSILE REQUIRES NO
C        DIRECTOR, OR IF THE CHANNEL ASSIGNED IS FREE.
C
  150 NCH = IFIX(X(515+KING+NB*2)+AMF)
      IF (NC1.EQ.7) GO TO 250
      IF (NC1.EQ.0) GO TO 160
      IF (CHANEL(K,NMS,NB,NC1).EQ.0.) GO TO 250
      GO TO 180
C
  160 DO 170 I=1,NCH
      IF (CHANEL(K,NMS,NB,I).EQ.0.) GO TO 240
  170 CONTINUE
C
  180 TA = 5000.
      KP = 0
      I = NSAM
  190 KA = IFIX(SMAT(I,2)+AMF)
      IF (KA.NE.K) GO TO 220
      KA = IFIX(SAMASM(I,2)*1.0E-1+AMF)
      T = FLOAT(KA)
      KA = IFIX(SAMASM(I,2)-T*10.+AMF)
      IF (KA.NE.NMS) GO TO 220
      KA = IFIX(SAMASM(I,3)*1.0E-1+AMF)
      IF (KA.NE.NB) GO TO 220
C
      DO 200 J=1,NCH
      KA = J+3
      TIM = SMAT(I,KA)
      IF (TA.LE.TIM) GO TO 200
      JAPL = J
      TA = TIM
  200 CONTINUE
C
```

```
      NC1 = JAPL                                                         SU3 1500
      IF (TT.GT.0.) GO TO 260                                            SU3 1510
      T = TA+1.0E-3                                                      SU3 1520
      KP = 3+(NMS-1)*3+NB                                                SU3 1530
      IF (NADAR(K,KP).EQ.3) GO TO 210                                    SU3 1540
      REASON = 5.                                                        SU3 1550
      GO TO 130                                                          SU3 1560
  210 REASON = 12.                                                       SU3 1570
      GO TO 130                                                          SU3 1580
  220 I = I-1                                                            SU3 1590
      IF (I.NE.0) GO TO 190                                              SU3 1600
  230 WRITE (6,620) K,NMS,NB                                             SU3 1610
      STOP                                                               SU3 1620
  240 NC1 = I                                                            SU3 1630
  250 TA = TIME                                                          SU3 1640
      ICOT = 0                                                           SU3 1650
      GO TO 270                                                          SU3 1660
  260 ICOT = 1                                                           SU3 1670
C                                                                        SU3 1680
C        CHECK TO SEE IF OWN SHIP IS ENGAGING THE TARGET WITH ANOTHER    SU3 1690
C        SYSTEM                                                          SU3 1700
C                                                                        SU3 1710
  270 DO 280 I=1,300                                                     SU3 1720
      IF (SAMASM(I,1).EQ.0.) GO TO 300                                   SU3 1730
      LML = IFIX(SAMASM(I,1)+AMF)                                        SU3 1740
      IF (LML.NE.M1) GO TO 280                                           SU3 1750
      LML = IFIX(SAMASM(I,2)*1.0E-1+AMF)                                 SU3 1760
      IF (LML.NE.K) GO TO 280                                            SU3 1770
      IF (SAMASM(I,6).GT.TIME) GO TO 290                                 SU3 1780
  280 CONTINUE                                                           SU3 1790
C                                                                        SU3 1800
      GO TO 300                                                          SU3 1810
  290 REASON = 6.                                                        SU3 1820
      GO TO 560                                                          SU3 1830
  300 IF (M1.LE.90) GO TO 320                                            SU3 1840
C                                                                        SU3 1850
C        CHECK TO SEE IF ASM WILL IMPACT BEFORE LAUNCH OF SAM.           SU3 1860
C                                                                        SU3 1870
      IF (ASMT(JULIA,2).GE.TA) GO TO 320                                 SU3 1880
  310 REASON = 10.                                                       SU3 1890
      GO TO 560                                                          SU3 1900
C                                                                        SU3 1910
C        CHECK TO SEE IF LAUNCHER IS LOADED                              SU3 1920
C                                                                        SU3 1930
  320 IF (RELOAD(K,NMS).LE.TIME) GO TO 340                               SU3 1940
      REASON = 9.                                                        SU3 1950
      IF (M1.LE.90) GO TO 330                                            SU3 1960
      IF (RELOAD(K,NMS).GE.ASMT(JULIA,2)) GO TO 310                      SU3 1970
```

```
  330 T = RELOAD(K,NMS)
      GO TO 130
  340 IF (ICOT.EQ.0) GO TO 380
      I = NSAM
C
C        IF SAM/SSM CHANNELS BUSY STORE A SHOOT EVENT WHEN THE CHANNEL
C        WILL BE FREE
C
  350 LML = IFIX(SAMASM(I,2)*1.0E-1+AMF)
      IF (LML.NE.K) GO TO 360
      T = FLOAT(LML)
      KP = IFIX(SAMASM(I,2)-T*10.+AMF)
      IF (KP.NE.NMS) GO TO 360
      LML = IFIX(SAMASM(I,3)*1.0E-1+AMF)
      IF (LML.NE.NB) GO TO 360
      T = FLOAT(LML)
      KP = IFIX(SAMASM(I,3)-T*10.+AMF)
      IF (KP.EQ.NC1) GO TO 370
  360 I = I-1
      IF (I.NE.0) GO TO 350
      GO TO 230
  370 T = SAMASM(I,6)+1.0E-3
      REASON = 7.
      GO TO 130
C
C        CHECK TO SEE IF, IN FACT THIS MISSILE CAN INTERCEPT
C        A TARGET WITH ITS FLIGHT CHARACTERISTICS
C
  380 CALL TIMMY (M1,K,NMS,XP,YP,PROB,FT,KP,TIME)
C
C        IF IND RETURNS FROM TIMMY WITH A VALUE OF 1, THE INTERCEPT
C        IS POSSIBLE
C
      IF (KP) 390,410,420
  390 REASON = 14.
  400 T = TIME+0.1
      GO TO 130
  410 REASON = 13.
      GO TO 560
  420 IF (KP.EQ.1) GO TO 430
      REASON = 15.
      GO TO 400
  430 IF (TT.EQ.0.) GO TO 500
      IF (NSAM.EQ.0) GO TO 500
C
C        IF THE SYSTEM REQUIRES THE DIRECTOR ONLY PART TIME,
C        CHECK TO SEE IF THE DIRECTOR WILL BE AVAILABLE WHEN NEEDED
C
```

```
      KIM = K*100+NMS*10+NB
      TG = TIME+FT-TT
      TL = TIME+FT
      TFT = 9000.
      I = NSAM
  440 IF (I.EQ.0) GO TO 490
      ID = IFIX(SAMASM(I,2)*10.+AMF)+IFIX(SAMASM(I,3)*1.0E-1+AMF)
      IF (KIM.EQ.ID) GO TO 450
      I = I-1
      GO TO 440
  450 NC2 = NC1+3
      IPQ = NCH+3
C
      DO 470 ID=4,IPQ
      IF (ID.EQ.NC2) GO TO 470
      IF (SMAT(I,ID).LT.TG) GO TO 460
      T = SMAT(I,ID)-TT
      IF (T.GT.TL) GO TO 460
      GO TO 480
  460 TCF(ID-3) = SMAT(I,ID)
  470 CONTINUE
C
      GO TO 490
  480 IF (TFT.LT.SMAT(I,ID)) GO TO 460
      MFOG = 1
      TFT = SMAT(I,ID)
      GO TO 460
  490 IF (MFOG.EQ.0) GO TO 500
      REASON = 11.
      T = TFT-FT+TT+1.0E-3
      GO TO 130
C
C        IF OK, FILL IN THE SAMASM AND SAM TABLES, LOG THE
C        CHANNEL BUSY, STORE THE INTERCEPT EVENT
C
  500 MFOG = IFIX(X(524+KING)+AMF)
      NSAM = NSAM+1
      IF (NSAM.GT.300) GO TO 600
      I = NSAM
      SAMASM(I,1) = FLOAT(M1)
      SAMASM(I,7) = PROB
      SAMASM(I,2) = FLOAT(K*10+NMS)
      SAMASM(I,3) = FLOAT(NB*10+NC1)
      SAMASM(I,4) = TIME
      SAMASM(I,6) = TIME+FT
      CALL STOREV (SAMASM(I,6),5,I)
      IBAT(K,NMS) = IBAT(K,NMS)-MFOG
      SMAT(I,10) = FLOAT(MFOG)
```

```
      X(75) = X(75)-X(524+KING)
      IF (X(4).EQ.0.) GO TO 510
      KP = IFIX(SHIP(K,5)+AMF)
      IF (X(KP+14).EQ.0.) GO TO 510
      TGTAV(M1) = TIME+FT
  510 RLT = X(525+KING)
      RELOAD(K,NMS) = TIME+RLT
      IP = IFIX(SHIP(K,5)+AMF)
      IP = IFIX(X(IP+NMS-1)+AMF)
      SMAT(I,1) = FLOAT(IP)
      SMAT(I,2) = FLOAT(K)
      ID = IBAT(K,NMS)
      SMAT(I,3) = FLOAT(ID)
      IF (TT.EQ.0.) GO TO 540
C
      DO 520 J=1,NCH
      JP = J+3
      IF (J.EQ.NC1) GO TO 530
      SMAT(I,JP) = TCF(J)
  520 CONTINUE
C
      GO TO 550
  530 SMAT(I,JP) = TIME+FT
      GO TO 520
  540 SMAT(I,4) = TIME+FT
  550 REASON = 0.
      IF (NC1.EQ.7) GO TO 560
      CHANEL(K,NMS,NB,NC1)=1.
  560 IF (X(9).EQ.0.) GO TO 570
      WRITE (6,630) K,M1,REASON
  570 IF (X(9).EQ.0.) GO TO 580
      KIM = K*100+NMS*10+NB
      WRITE (6,640) NSAM,KIM,NC1
  580 IF (X(55).EQ.0.) GO TO 590
      CALL GETIMX (K)
      TIM = K*0.000026
      WRITE (6,650) TIM
  590 RETURN
  600 WRITE (6,660)
      STOP
C
  610 FORMAT (' SUBE3 IN AT',F15.8)
  620 FORMAT (' ERROR IN SMAT OR SAMASM. CHANEL INDICATES ALL SEGMENTS.S
     1HIP',I3,' BATTERY',I2,' DIRECTOR',I2,' BUSY.'/(' NO AVAILABLE TIME
     2 RETREVABLE.'))
  630 FORMAT ('0','EV 3  SHIP',I3,' TARGET',I4,'  REASON',F4.0)
  640 FORMAT (' EV 3  SAM',I4,' BATTERY',I6,' CHANNEL',I2)
  650 FORMAT (' SUBE3 OUT AT',F15.8)
```

```
  660 FORMAT (' NUMBER OF SALVOS FIRED EXCEEDS 300, PROGRAM TERMINATING'
     1)
      END

      SUBROUTINE SUBE4 (IU)
C
C        THIS SUBROUTINE EVALUATES RED ASM/SSM FLIGHT PERFORMANCE
C        AND DETERMINES IF THE RED MISSILE HIT THE BLUE TARGET
C
      CCMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 110
      CALL GETIMX (ID)
      PB = ID*2.6E-5
      WRITE (6,170) PB
C
C        DETERMINE THE ASM/SSM NUMBER AND TARGET SHIP NUMBER
C
  110 ID = (IU/10)*10
      IF (ID.EQ.IU) IU = IU-1
      NAS = IFIX(FLOAT(IU)*1.0E-2+AMF)
      KP = NAS*100
      K = IU-KP
      NAM = NAS-90
      IU = IU+1
      IF (ID.EQ.IU) K = K+1
C
C        IF THE ASM/SSM HAS BEEN SHOT DOWN OR THE SHOT INVALIDATED
C        BY A BOMBER KILL, EXIT
C
      IF (ASMT(NAM,1).EQ.2..OR.ASMT(NAM,1).EQ.5.) GO TO 140
      ID = IFIX(ASMT(NAM,6)+AMF)
      PB = X(2005+(ID-1)*10)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
C
C        DETERMINE INFLIGHT PERFORMANCE
C
      IF (Z.LE.PB) GO TO 120
      ASMT(NAM,1) = 3.
      GC TO 140
```

```
C
C        INCREMENT THE HIT COUNTER
C
  120 IHIT(K) = IHIT(K)+1
      IF (X(5).EQ.1.) GO TO 130
      ID = IFIX(SHIP(K,4)+AMF)
      IF (IHIT(K).EQ.ID) SHIP(K,6)=0.
  130 ASMT(NAM,1) = 4.
  140 IF (X(56).EQ.0.) GO TO 150
      WRITE(6,180) NAM,K,ASMT(NAM,1),IHIT(K),ASMT(NAM,2)
  150 IF(IQ.EQ.0) GO TO 160
      CALL GETIMX (ID)
      PB = ID*2.6E-5
      WRITE (6,190) PB
  160 RETURN
C
  170 FORMAT (' SUBE4 IN AT',F15.8)
  180 FORMAT ('0','EV 4   ASM',I4,' SHIP',I3,' ASM STATUS',F3.0,' HITS O
     1N SHIP',I4,' TIME',F8.3)
  190 FORMAT (' SUBE4 OUT AT',F15.8)
      END


      SUBROUTINE SUBE5 (M)
C
C        THIS SUBROUTINE EVALUATES THE BLUE MISSILE INTERCEPT OF
C        THE RED RAIDER OR ASM/SSM
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      IF (X(55).EQ.0.) GO TO 20
      CALL GETIMX (ID)
      PB = ID*2.6E-5
      WRITE (6,230) PB
C
C        DETERMINE THE PARTICIPANTS
C
   20 NT = IFIX(SAMASM(M,1)+AMF)
      NA = NT-90
      NMS = IFIX(SAMASM(M,2)+AMF)
      K = IFIX((SAMASM(M,2)*1.0E-1)+AMF)
      JACK(K,2) = JACK(K,2)+1
      NMS = NMS-K*10
```

```
      NB = IFIX((SAMASM(M,3)*1.0E-1)+AMF)
      ID = IFIX(SHIP(K,5)+AMF)
      NC = IFIX(SAMASM(M,3)+AMF)-NB*10
      IF (NC.EQ.7) GO TO 30
C
C        IF THE MISSILE USES A DIRECTOR, DETERMINE THE APPROPRIATE
C        REASSIGNMENT DELAYS
C
      KEY = IFIX(X(ID+18)+AMF)
      CALL REACT (KEY,5,TF)
      IQ = IFIX(X(ID+NMS-1)+AMF)
      IP = IFIX(X(516+(IQ-1)*40+NB*2)+AMF)
      KEY = IFIX(X(906+(IP-1)*20)+AMF)
      CALL REACT (KEY,4,TA)
C
C        INCREMENT THE FLEET AND SHIP COUNTERS IF APPROPRIATE
C
   30 IF (IGAGE(NT).NE.0) GO TO 50
      IGAGE(NT) = 1
      IF (NT.GT.90) GO TO 40
      ISBG(K) = ISBG(K)+1
      IFBG = IFBG+1
      GO TO 50
   40 ISAG(K) = ISAG(K)+1
      IFAG = IFAG+1
C
C        DETERMINE IF THE TARGET IS STILL THERE.
C
   50 IF (NT.GT.90) GO TO 60
      IF (RAID(NT,13).GT.0.) GO TO 80
      GO TO 70
   60 IF (ASMT(NA,1).EQ.2.) GO TO 70
      IF (ASMT(NA,1).NE.1.) GO TO 100
      GO TO 80
   70 SAMASM(M,5) = -1.00
      IF (NC.EQ.7) GO TO 190
      GO TO 170
   80 IQ = IFIX(X(ID+NMS-1)+AMF)
      PB = X(530+(IQ-1)*40)
      N = IFIX(X(524+(IQ-1)*40)+AMF)
C
C        DETERMINE KILL-NO KILL
C
      PROB = PB*(1.-(1.-SAMASM(M,7))**N)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      IF (Z.LE.PROB) GO TO 110
      IF (NC.EQ.7) GO TO 90
```

```
      CHANEL(K,NMS,NB,NC)=0.
   90 KP = NT*100000+K*1000+NMS*100+NB*10+NC
C
C        STORE A CHANNEL FREE EVENT
C
      CALL STOREV (SAMASM(M,6),3,KP)
      SAMASM(M,5) = 0.
      GO TO 220
  100 SAMASM(M,5) = 0.
      IF (NC.EQ.7) GO TO 190
      GO TO 170
  110 JACK(K,3) = JACK(K,3)+1
      IF (NT.GT.90) GO TO 150
      RAID(NT,13) = RAID(NT,13)-1.0001
      IF (RAID(NT,13).GT.0.) GO TO 160
      RAID(NT,13) = 0.0
C
C        IF A RAIDER HAS BEEN SHOT DOWN OR SJNK, AND IT WAS A JAMMER,
C        ADJUST THE JAM TABLE
C
      NDU = IFIX(RAID(NT,1)+AMF)
      IF (X(NDU+9).GT.0.) JAM(1)=JAM(1)-1
      IF (X(NDU+10).GT.0.) JAM(2)=JAM(2)-1
      IF (X(NDU+11).GT.0.) JAM(3)=JAM(3)-1
      IF (X(NDU+12).GT.0.) JAM(4)=JAM(4)-1
      RAID(NT,17) = SAMASM(M,6)
      NTOT = MISLFT(NT,1)+MISLFT(NT,2)
C
C        IF THE RAIDER HAS MISSILES LEFT ON BOARD- INDICATE "LOST
C        WHEN A/C SHOT DOWN OR SHIP SUNK"
C
      IF (NTOT.EQ.0) GO TO 130
      MISLFT(NT,1) = 0
      MISLFT(NT,2) = 0
C
      DO 120 I=1,NTOT
      NASM = NASM+1
      ASMT(NASM,1) = 5.
      ASMT(NASM,2) = 0.
      ASMT(NASM,4) = 0.
      ASMT(NASM,6) = 0.
      ASMT(NASM,7) = 0.
      ASMT(NASM,8) = 0.
      ASMT(NASM,5) = SAMASM(M,6)
      ASMT(NASM,3) = FLOAT(NT)
  120 CONTINUE
C
  130 ID = NASM-NTOT
```

```
      IF (RAID(NT,12).EQ.0.) GO TO 160
C
C        CHECK ASM LAUNCH TIME OF SHOT DOWN RAIDER OR SSM LAUNCH
C        TIME OF SURFACE RAIDER- IF AFTER TARGET KILL TIME- INDICATE
C        NO LAUNCH.
C
      DO 140 I=1,ID
      IQ = IFIX(ASMT(I,3)+AMF)
      IF (NT.NE.IQ) GO TO 140
      IF (ASMT(I,5).LT.SAMASM(M,6)) GO TO 140
      ASMT(I,1) = 5.
      ASMT(I,2) = 0.
      ASMT(I,5) = SAMASM(M,6)
      ASMT(I,4) = 0.
      ASMT(I,7) = 0.
      ASMT(I,8) = 0.
  140 CONTINUE
C
      GO TO 160
  150 ASMT(NA,1) = 2.
  160 SAMASM(M,5) = FLOAT(N)
      TK = X(533+(IQ-1)*40)
      IF (NC.EQ.7) GO TO 190
      SAMASM(M,6) = SAMASM(M,6)+TK+TA+TF
      JFK = NC+3
      SMAT(M,JFK) = SAMASM(M,6)
      GO TO 180
  170 SAMASM(M,6) = SAMASM(M,6)+TA+TF
      JFK = NC+3
      SMAT(M,JFK) = SAMASM(M,6)
  180 KP = SAMASM(M,2)*100.+SAMASM(M,3)
      CALL STOREV (SAMASM(M,6),6,KP)
  190 IF (X(58).EQ.0.) GO TO 200
      WRITE (6,240) M,SAMASM(M,5)
  200 IF (X(56).EQ.0.) GO TO 210
      WRITE (6,250) NT,K,NMS,M,SAMASM(M,6)
  210 IF (X(55).EQ.0.) GO TO 220
      CALL GETIMX (ID)
      PB = ID*2.6E-5
      WRITE (6,260) PB
  220 RETURN
C
  230 FORMAT (' SUBE5 IN AT',F15.8)
  240 FORMAT ('0','FOR SHOT',I4,' RESULTS',F4.0)
  250 FORMAT ('0','EV 5   TGT NO.',I4,' FS',I3,' LAUNCHER',I2,' SAMS',I4
     1,' TIME',F8.3)
  260 FORMAT (' SUBE5 OUT AT',F15.8)
      END
```

```
      SUBROUTINE SUBE6 (M)
C
C          THIS SUBROUTINE RELEASES A FIRE CONTROL CHANNEL
C          FOR FURTHER USE AFTER AN ENGAGEMENT AND THE APPROPRIATE
C          TIME DELAYS.
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
C
      IF (X(55).EQ.0.) GO TO 110
      CALL GETIMX (I)
      T = I*2.6E-5
      WRITE (6,130) T
  110 M = M+2
      K = M*1.0E-3
      NMS = (M-K*1000)*1.0E-2
      I = M-K*1000-NMS*100
      NB = I*1.0E-1
      NC = I-NB*10
      NC = NC-2
      CHANEL(K,NMS,NB,NC)=0.
      IF (X(55).EQ.0.) GO TO 120
      CALL GETIMX (I)
      T = I*2.6E-5
      WRITE (6,140) T
  120 RETURN
C
  130 FORMAT (' SUBE6 IN AT ',F14.8)
  140 FORMAT (' SUBE6 OUT AT',F15.8)
      END


      SUBROUTINE TIMMY (M,K,NMS,XP,YP,PROB,FT,IND,TIME)
C
C        TIMMY COMPUTES THE PREDICTED INTERCEPT POINT AND
C        DETERMINES WHETHER OR NOT IT LIES WITHIN THE INTERCEPT
C        CONTOUR.
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
```

```
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      REAL *8RAT
      FZLI = 0.
      FZR1 = 0.
      IF (X(55).EQ.0.) GO TO 20
      CALL GETIMX (KP)
      XT = KP*2.6E-5
      WRITE (6,510) XT
   20 IND = 0
C
C        GET THE END OF GAME POSIT OR TARGET SHIP POSIT.
C
      PI = 3.1415926
      IF (M.GT.90) GO TO 30
      XT = RAID(M,5)
      YT = RAID(M,6)
      GO TO 40
   30 JULIA = M-90
      KP = IFIX(ASMT(JULIA,4)+AMF)
      XT = SHIP(KP,2)
      YT = SHIP(KP,3)
   40 KP = IFIX(SHIP(K,5)+AMF)
      KL = IFIX(X(KP+NMS-1)+AMF)
      KL = 501+(KL-1)*40
C
C        IF FIRING SECTORS HAVE BEEN ASSIGNED TO BLUE UNITS,
C        DETERMINE THE SHIP'S FREE FIRE ZONES.
C
      IF (X(8).EQ.0.) GO TO 120
      RAT = X(1204+(K-1)*5)
      RXATA(1) = RADIAN(RAT)
      RAT = X(1205+(K-1)*5)
      RXATA(3) = RADIAN(RAT)
      RAT = X(KP+19+NMS*2)
      RXATA(2) = RADIAN(RAT)
      RAT = X(KP+20+NMS*2)
      RXATA(4) = RADIAN(RAT)
C
      DO 50 I=1,4
      KMART(I) = I
   50 CONTINUE
C
      CALL ZLNKA (RXATA,KMART,4)
      KP = KMART(1)*1000+KMART(2)*100+KMART(3)*10+KMART(4)
      IF (KP.EQ.4123.OR.KP.EQ.3124.OR.KP.EQ.4213.OR.KP.EQ.3214) GO TO 70
```

```
      IF (KP.EQ.3412.OR.KP.EQ.4312.OR.KP.EQ.3421.OR.KP.EQ.4321) GO TO 90
      IF (KP.EQ.2341.OR.KP.EQ.2431.OR.KP.EQ.1342.OR.KP.EQ.1432) GO TO 80
      IF (KP.EQ.1234.OR.KP.EQ.1243.OR.KP.EQ.2134.OR.KP.EQ.2143) GO TO 11
     10
      IF (KP.EQ.1324.OR.KP.EQ.2413) GO TO 60
      IF (KP.EQ.3241.OR.KP.EQ.4132) GO TO 100
C
C        IF THE SHIP HAS NONE, EXIT.
C
      REASON = 2.
      GO TO 480
   60 FZL = RXATA(1)
      FZR = RXATA(4)
      FZL1 = RXATA(3)
      FZR1 = RXATA(2)
      KP = 2
      GO TO 120
   70 FZL = RXATA(2)
      FZR = RXATA(1)
      KP = 1
      GO TO 120
   80 FZL = RXATA(4)
      FZR = RXATA(3)
      KP = 1
      GO TO 120
   90 FZL = RXATA(3)
      FZR = RXATA(2)
      KP = 1
      GO TO 120
  100 FZL = RXATA(4)
      FZR = RXATA(3)
      FZL1 = RXATA(2)
      FZR1 = RXATA(1)
      KP = 2
      GO TO 120
  110 FZL = RXATA(1)
      FZR = RXATA(4)
      KP = 1
C
C       COLLECT TARGET ALTITUDE AND MAXIMUM SAM ALTITUDE. GET
C         MISSILE PARAMETERS UNLESS TARGET TOO HIGH, IN WHICH
C         CASE, EXIT.
C
  120 KPX = 7
      IF (X(KL+35).EQ.1.) KPX=0
      IF (M.GT.90) GO TO 130
      ALT = RAID(M,9)
      GO TO 140
```

```
  130 ID = IFIX(ASMT(JULIA,6)+AMF)
      ALT = X(2002+(ID-1)*10)
  140 IF (ALT.GT.X(77)) KPX=0
      IF (ALT.LE.X(KL+15)) GO TO 150
      REASON = 11.
      GO TO 480
  150 KPX = KL+KPX
      RMAX = X(KPX)
      SSPD = X(KPX+1)
      A = X(KPX+2)
      B = X(KPX+3)
      EX = X(KPX+4)
      SLOPE = X(KPX+5)
      SEEK = X(KPX+6)
      RMIN = X(KL+14)
C
C         DETERMINE IF THE TARGET WILL ENTER THIS SYSTEM'S ENVELOPE
C
      IF (M.LE.90) GO TO 160
      IR = IFIX(ASMT(JULIA,3)+AMF)
      VEL = X(2001+(ID-1)*10)
      PQ = RAID(IR,14)
      PS = RAID(IR,15)
      RAID(IR,14) = XT
      RAID(IR,15) = YT
      PT = RAID(IR,2)
      RAID(IR,2) = TIME
      RAID(IR,3) = XP
      RAID(IR,4) = YP
      GO TO 170
  160 RAID(M,14) = XT
      RAID(M,15) = YT
      CALL LOCTIM (M,TIME,5,XI,YI)
      KR = IFIX(X(2201+(M-1)*6)+AMF)
      VEL = X(1814+(KR-1)*20)
      IR = M
  170 CALL TMDASM (IR,0,NMS,K,VEL,RMAX,TIME,TX,XI,YI,RIP,TXL,M)
      IF (M.LE.90) GO TO 180
      RAID(IR,14) = PQ
      RAID(IR,15) = PS
      CALL LOCTIM (IR,PT,5,PS,PQ)
C
C        IF TARGET BEYOND RANGE AND OUTBOUND,EXIT.
C
  180 IF (TX.NE.9120.) GO TO 190
      REASON = 7.
      GO TO 480
C
```

```
C         DETERMINE TARGET COURSE.
C
  190 IF (M.LE.90) GO TO 220
      KPX = IFIX(ASMT(JULIA,4)+AMF)
      IF (KPX.NE.K) GO TO 210
C
C         IF TARGET SHIP SAME AS FIRING SHIP, DETERMINE FLIGHT TIME
C
      DELTAX = XT-ASMT(JULIA,7)
      DELTAY = YT-ASMT(JULIA,8)
      CS = ATAN2(DELTAY,DELTAX)
      IF (CS.LT.0.) CS = CS+2.*PI
      THETA = CS+PI
      IF (THETA.GT.2.*PI) THETA=THETA-2.*PI
      IF (X(59).EQ.0.) GO TO 200
      WRITE (6,520) M,K,CS
  200 IF (ABS(DELTAY).LT.1.0E-4) DELTAY=0.0
      IF (ABS(DELTAX).LT.1.0E-4) DELTAX=0.0
      DLT = SQRT(DELTAX**2+DELTAY**2)
      FT = DLT/(SSPD+VEL)
      PK = X(KL+28)
      GAMMA = 0.
      GO TO 320
  210 DELTAX = XT-ASMT(JULIA,7)
      DELTAY = YT-ASMT(JULIA,8)
      CS = ATAN2(DELTAY,DELTAX)
      IF (CS.LT.0.) CS = CS+2.*PI
      GO TO 230
  220 RAT = X(2205+(M-1)*6)
      CS = RADIAN(RAT)
C
C         ALPHA:  ANGLE BETWEEN THE TARGET FLIGHT PATH
C                 AND THE LINE OF SIGHT.
C
C         THETA:  THE ANGLE FROM THE FIRING SHIP TO PRESENT
C                 TARGET POSITION.
C
  230 DELTAY = YP-SHIP(K,3)
      DELTAX = XP-SHIP(K,2)
      IF (DELTAX.EQ.0..AND.DELTAY.EQ.0.) GO TO 250
  240 THETA = ATAN2(DELTAY,DELTAX)
      IF (THETA.LT.0.) THETA=THETA+2.*PI
      TH1 = THETA-PI
      IF (TH1.LT.0.) TH1 = TH1+2.*PI
      ALPHA = ABS(CS-TH1)
      IF (ALPHA.GT.PI) ALPHA=ALPHA-PI
      GSINE = SIN(ALPHA)*(VEL/SSPD)
C
```

```
C         IF GSINE GREATER THAN ONE, INTERCEPT IS IMPOSSIBLE
C
      IF (ABS(GSINE).LE.1.) GO TO 280
      REASON = 10.
      GO TO 480
  250 IF (M.LE.90) GO TO 260
      DELTAX = ASMT(JULIA,7)-SHIP(K,2)
      DELTAY = ASMT(JULIA,8)-SHIP(K,3)
      GO TO 240
  260 CALL LOCTIM (M,0.,1,XW,YW)
      DELTAX = XW-SHIP(K,2)
      DELTAY = YW-SHIP(K,3)
      GO TO 240
  270 IF (PHI.LT.0.) PHI = PHI+2.*PI
  280 GAMMA = ARSIN(GSINE)
C
      BETA = PI-GAMMA-ALPHA
      DELTA = PI-BETA
C
      IF (ABS(DELTAX).LT.5.0E-5) DELTAX=0.0
      IF (ABS(DELTAY).LT.5.0E-5) DELTAY=0.0
      DLT = SQRT(DELTAX**2+DELTAY**2)
      IF (X(59).EQ.0.) GO TO 290
      WRITE (6,530)
      WRITE (6,540)
      WRITE (6,550) M,K,ALPHA,BETA,GAMMA,XP,YP,GSINE,DLT
  290 P4 = PI/4.
      IF (DELTA.LE.P4) GO TO 300
      PK = X(KL+36)
      GO TO 310
  300 PK = X(KL+28)
C
C         IF TARGET NOT ASM TARGETED ON FIRING SHIP, COMPUTE FLIGHT
C         TIME
C
  310 FT = DLT*SIN(ALPHA)/(SSPD*SIN(BETA))
      IF (M.LE.90) GO TO 320
      KPX = IFIX(SHIP(K,5)+AMF)
      IF (X(KPX+29).EQ.0.) GO TO 320
      TKIL = TIME+FT
      IF (TKIL.LT.ASMT(JULIA,2)) GO TO 320
      REASON = 8.
      GO TO 480
  320 TKIL = TIME+FT
C
C         GET INTERCEPT POSIT
C
      IF (M.GT.90) GO TO 330
```

```
      CALL LOCTIM (M,TKIL,44,XI,YI)
      GO TO 340
  330 CALL ASMTIM (M,TKIL,XI,YI)
  340 IF (X(8).EQ.1.) GO TO 350
      KPX = IFIX(SHIP(K,5)+AMF)
      RAT = X(KPX+19+NMS*2)
      FZL = RADIAN(RAT)
      RAT = X(KPX+20+NMS*2)
      FZR = RADIAN(RAT)
      KP = 1
  350 DELTAY = YI-SHIP(K,3)
      DELTAX = XI-SHIP(K,2)
      IF (DELTAX.EQ.0..AND.DELTAY.EQ.0.) GO TO 370
      PHI = ATAN2(DELTAY,DELTAX)
C
C        DETERMINE FIRING BEARING AND CHECK FOR GO
C
      IF (PHI.LT.0.) PHI = PHI+2.*PI
      RXATA(1) = FZL
      RXATA(2) = PHI
      RXATA(3) = FZR
      KMART(1) = 1
      KMART(2) = 2
      KMART(3) = 3
      CALL ZLNKA (RXATA,KMART,3)
      ID = KMART(1)*100+KMART(2)*10+KMART(3)
      IF (ID.EQ.213.OR.ID.EQ.132.OR.ID.EQ.321) GO TO 410
      IF (KP.EQ.2) GO TO 400
  360 REASON = 3.
      IND = -1
      GO TO 480
  370 IF (M.GT.90) GO TO 390
      CALL LOCTIM (M,0.,1,XW,YW)
      DX = XW-SHIP(K,2)
      DY = YW-SHIP(K,3)
  380 PHI = ATAN2(DY,DX)
      GO TO 270
  390 DX = ASMT(JULIA,7)-SHIP(K,2)
      DY = ASMT(JULIA,8)-SHIP(K,3)
      GO TO 380
  400 RXATA(1) = FZL1
      RXATA(2) = PHI
      RXATA(3) = FZR1
      KMART(1) = 1
      KMART(2) = 2
      KMART(3) = 3
      CALL ZLNKA (RXATA,KMART,3)
      ID = KMART(1)*100+KMART(2)*10+KMART(3)
```

```
      IF (ID.EQ.123.OR.ID.EQ.231.OR.ID.EQ.312) GO TO 360
C
C        CAN MISSILE ENGAGE
C
  410 IF (ABS(DELTAX).LT.5.0E-5) DELTAX=0.
      IF (ABS(DELTAY).LT.5.0E-5) DELTAY=0.
      DIS = SQRT(DELTAX**2+DELTAY**2)
      IF (DIS.GE.RMIN) GO TO 420
      REASON = 4.
      IND = -1
      GO TO 480
  420 PHI1 = CS-3.*PI/2.
      IF (PHI1.LT.0.) PHI1=PHI1+2.*PI
      XIS = XI-SHIP(K,2)
      YIS = YI-SHIP(K,3)
      XIP = ABS(XIS*COS(PHI1)+YIS*SIN(PHI1))
      YIP = YIS*COS(PHI1)-XIS*SIN(PHI1)
      Z = (XIP/A)**EX+(ABS(YIP)/B)**EX
      IF (Z.LE.1.) GO TO 430
      REASON = 5.
      IND = -1
      GO TO 480
  430 Z = YIP-SLOPE*XIP
      IF (Z.GE.SEEK) GO TO 440
      REASON = 6.
      GO TO 480
  440 IF (X(76).EQ.0.) GO TO 460
      PIZ = X(76)**2
C
      DO 450 I=1,NS
      IF (I.EQ.K) GO TO 450
      DELTAX = ABS(SHIP(I,2)-XI)
      DELTAY = ABS(SHIP(I,3)-YI)
      IF (DELTAX.LT.5.0E-5) DELTAX=0.0
      IF (DELTAY.LT.5.0E-5) DELTAY=0.0
      DIS = DELTAX**2+DELTAY**2
      IF (DIS.GE.PIZ) GO TO 450
      IND = 2
      REASON = 9.
      GO TO 480
  450 CONTINUE
C
C        IF THE TARGET IS AN ASM WHOS TARGET IS THE FIRING SHIP,
C        OR THE TARGET IS A SURFACE/SUB SURFACE UNIT, PROCEED,
C        OTHERWISE ADD IN THE TARGET ALTITUDE TO MISSILE FLIGHT TIME
C
  460 IF (M.LE.90) GO TO 470
      JOAN = IFIX(ASMT(JULIA,4)+AMF)
```

```
      IF (JOAN.EQ.K) GO TO 470
      RUTH = ALT/6000.
      IF (RUTH.LT.AMF) GO TO 470
      FT = SQRT(DIS+RUTH**2)/SSPD
  470 IND = 1
      PM = X(KL+30)
      PROB = PK*PM
      IF (X(9).EQ.0.) GO TO 490
      WRITE (6,560) M,K
      WRITE (6,570) SHIP(K,2),SHIP(K,3),XI,YI,KP,FZL,FZR,FZL1,FZR1,PHI
  480 IF (X(9).EQ.0.) GO TO 490
      WRITE (6,580) M,K,IND,REASON
  490 IF (X(55).EQ.0.) GO TO 500
      CALL GETIMX (KP)
      XT = KP*2.6E-5
      WRITE (6,590) XT
  500 RETURN
C
  510 FORMAT (' TIMMY IN AT',F15.8)
  520 FORMAT ('0','     TARGET',I4,'     SHIP',I3,'    TARGET COURSE',F5.2)
  530 FORMAT ('0','CROSSING TARGET DATA')
  540 FORMAT ('0','TARGET  SAM SHIP  ANGLE BETWEEN   ANGLE BETWEEN    AN
     1GLE BETWEEN',6X,'XP',7X,'YP    GSINE  RANGE TO'/(19X,'TGT PATH & L
     2OS  SAM FLIGHT PATH   SAM FLIGHT PATH',27X,'TARGET'/(19X,'AT LAUNC
     3H TIME  TGT FLIGHT PATH  LOS AT LAUNCH')))
  550 FORMAT ('0',I4,I7,F13.4,F15.4,F17.4,F13.2,F9.2,F8.4,F9.2)
  560 FORMAT ('0'.'INTERCEPT DATA... TGT',I4,'  FIRING SHIP',I3)
  570 FORMAT ('0',' X-FS',7X,'Y-FS',7X,'X-INT',6X,'Y-INT',2X,'# ZONES',3
     1X,'LEFT 1. RIGHT 1',5X,'LEFT 2  RIGHT 2',5X,'SAM TRAJ. BEARING'/(F
     27.2,2(5X,F6.2),4X,F6.2,4X,I4,2(3X,F6.3),5X,F6.3,3X,F6.3,10X,F6.3))
  580 FORMAT ('0','TGT',I4,' SHIP',I3,' IND IS',I3,' REASON IS',F3.0)
  590 FORMAT (' TIMMY OUT AT',F15.8)
      END

      SUBROUTINE JAMM (RG,I,IR,IJ,ID,VEL,RRH,AREA,IBM,ICPA,RGI)
C          THIS SUBROUTINE DETERMINES THE RANGE AT WHICH A TARGET CAN
C          BE DETECTED IN A MULTIPLE STAND-OFF JAMMING ENVIRONMENT
C          RG---------MINIMUM OF CLEAR AND HORIZON RANGES (INPUT)
C          I----------SHIP NUMBER OF RADAR PLATFORM (INPUT)
C          IR---------RAIDER NUMBER OF TARGET (INPUT)
C          IJ---------FREQUENCY BAND OF RADAR (INPUT)
C          ID---------RADAR CLASS NUMBER (INPUT)
C          VEL--------VELOCITY OF RAIDER IN NM/MIN (INPUT)
C          RRH--------RADAR'S HORIZON (INPUT)
C          AREA-------FOURTH ROOT OF RAIDER'S RADAR CROSS SECTION (INPUT)
C          IBM--------FLAG INDICATING IF TARGET IS AN AIRCRAFT OR MISSILE.
C                     EQUAL TO 0 IF TARGET IS A BOMBER.   IF TARGET IS A
```

```
C            MISSILE SET TO 1 (INPUT)
C      ICPA------FLAG INDICATING IF TARGET'S CPA LESS THAN DETECTION
C            RANGE. EQUAL TO 0 IF CPA IS BEYOND RANGE (OUTPUT)
C      RGI-------DETECTION RANGE (OUTPUT)
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      REAL *4JJTBR,LJBW,JRBRG,LTBRG
      IF (X(55).EQ.0.) GO TO 20
      CALL GETIMX (IET)
      T = IET*.0000026
      WRITE (6,290) T
   20 ICPA = 1
C
C         SET SITE VALUES
C
      XR = SHIP(I,2)
      YR = SHIP(I,3)
C
C         SET TRACK VALUES
C
      IF (IBM.EQ.0) GO TO 30
      X1 = ASMT(NASM,7)
      Y1 = ASMT(NASM,8)
      ITS = IFIX(ASMT(NASM,4)+AMF)
      XO = SHIP(ITS,2)
      YO = SHIP(ITS,3)
      MIC = IFIX(ASMT(NASM,6)+AMF)
      Z1 = X(2002+(MIC-1)*10)/6000.
      TIMC = ASMT(NASM,5)
      GO TO 40
   30 X1 = RAID(IR,3)
      Y1 = RAID(IR,4)
      XO = RAID(IR,14)
      YO = RAID(IR,15)
      Z1 = RAID(IR,9)/6000.
      TIMC = RAID(IR,2)
   40 CALL CPA (X1,Y1,XO,YO,XR,YR,XC,YC)
      RCPS = (XR-XC)**2+(YR-YC)**2+Z1**2
      RGS = RG**2
C
C         DETERMINE IF CPA IS WITHIN RANGE
C
```

```
      IF (RCPS.GE.RGS) GO TO 250
      DELX = ABS(XO-X1)-0.1
      DELY = ABS(YO-Y1)-0.1
      IF (DELX.LT.0.0.AND.DELY.LT.0.0) GO TO 90
      IF (DELX) 50,60,60
   50 TIMI = (ABS(YC-Y1))/VEL+TIMC
      GO TO 100
   60 IF (DELY) 70,80,80
   70 TIMI = (ABS(XC-X1))/VEL+TIMC
      GO TO 100
   80 VD = SQRT((XO-X1)**2+(YO-Y1)**2)
      VX = VEL*(XO-X1)/VD
      VY = VEL*(YO-Y1)/VD
      TIMX = (XC-X1)/VX
      TIMY = (YC-Y1)/VY
      TIMI = (TIMX+TIMY)/2.+TIMC
      GO TO 100
   90 TIMI = 3.0
C
C         DETERMINE BEARING FROM RADAR TO TARGET IN RADIANS
C
  100 YRT = YC-YR
      XRT = XC-XR
      IF (YRT.NE.0.0.OR.XRT.NE.0.0) GO TO 110
      YRT = Y1-YR
      XRT = X1-XR
  110 TBRG = ATAN2(YRT,XRT)
      IF (TBRG.LT.0.) TBRG=TBRG+6.2831853
C
C         GET HALF RADAR BEAM WIDTH IN RADIANS
C
      HRBW = X(911+(ID-1)*20)
      SUMJP = 0.0
      RCOF = X(904+(ID-1)*20)
C
C         DETERMINE JAMMING LEVEL AT THE RADAR IN DIRECTION OF TARGET
C
      DO 240 N=1,NR
      IF (TIMI.GT.RAID(N,17)) GO TO 240
      RCOFF = RCOF
      IRC = IFIX(X(2201+(N-1)*6)+AMF)
      IIR = (IRC-1)*20
C
C         FIND JAMMING POWER OF RAIDER ON RADAR BAND
C
      PJ = X(1809+IIR+IJ)
      IF (PJ.LE.0.0) GO TO 240
      CALL LOCTIM (N,TIMI,1,XJ,YJ)
```

```
C
C        DETERMINE IF JAMMER IS ABOVE RADAR HORIZON
C
      RJHS = (RAID(N,10)+RRH)**2
      RRJS = (XR-XJ)**2+(YR-YJ)**2
      IF (RRJS.GT.RJHS) GO TO 240
C
C        DETERMINE COORDINATES OF JAMMER TARGET
C
      JS = IFIX(X(2206+(N-1)*6)+AMF)
      IF (JS.EQ.0) GO TO 240
      IF (JS.GT.40) GO TO 120
      XJT = SHIP(JS,2)
      YJT = SHIP(JS,3)
      GO TO 130
  120 XJT = 0.0
      YJT = 0.0
  130 YJR = YR-YJ
      XJR = XR-XJ
      IF (YJR.NE.0.0.OR.XJR.NE.0.0) GO TO 140
      CALL LOCTIM (N,0.0,1,XJJ,YJJ)
      YJR = YR-YJJ
      XJR = XR-XJJ
C
C        GET BEARING FROM JAMMER TO RADAR AND FROM RADAR TO JAMMER
C
  140 JRBRG = ATAN2(YJR,XJR)
      IF (JRBRG.LT.0.0) JRBRG=JRBRG+6.2831853
      RJBRG = JRBRG-3.1415926
      IF (RJBRG.LT.0.0) RJBRG=RJBRG+6.2831853
C
C        FIND BEARING FROM JAMMER TO JAMMER'S TARGET
C
      YJJT = YJT-YJ
      XJJT = XJT-XJ
      IF (YJJT.NE.0.0.OR.XJJT.NE.0.0) GO TO 150
      CALL LOCTIM (N,0.0,1,XJJ,YJJ)
      YJJT = YJT-YJJ
      XJJT = XJT-XJJ
  150 JJTBR = ATAN2(YJJT,XJJT)
      IF (JJTBR.LT.0.0) JJTBR=JJTBR+6.2831853
C
C        GET JAMMER SIDE-LOBE COEFFICIENT AND HALF-BEAM WIDTH
C
      COFF = X(1818+IIR)
      HJBW = X(1817+IIR)
      LJBW = JJTBR+HJBW
      RJBW = JJTBR-HJBW
```

```
C
C         DETERMINE IF RADAR IS IN MAIN BEAM OF JAMMER
C
      IF (LJBW.GT.6.2831853) GO TO 160
      IF (RJBW.LT.0.0) GO TO 170
      IF (LJBW.GT.JRBRG.AND.RJBW.LT.JRBRG) COFF=1.0
      GO TO 190
  160 LJBW = LJBW-6.2831853
      GO TO 180
  170 RJBW = RJBW+6.2831853
  180 IF (LJBW.GT.JRBRG.OR.RJBW.LT.JRBRG) COFF=1.0
C
C         DETERMINE IF JAMMER IS IN MAIN BEAM OF RADAR
C
  190 LTBRG = TBRG+HRBW
      RTBRG = TBRG-HRBW
      IF (LTBRG.GT.6.2831853) GO TO 200
      IF (RTBRG.LT.0.0) GO TO 210
      IF (LTBRG.GT.RJBRG.AND.RTBRG.LT.RJBRG) RCOFF=1.0
      GO TO 230
  200 LTBRG = LTBRG-6.2831853
      GO TO 220
  210 RTBRG = RTBRG+6.2831853
  220 IF (LTBRG.GT.RJBRG.OR.RTBRG.LT.RJBRG) RCOFF=1.0
  230 XCOF = COFF*RCOFF
      RRJS = (RRJS+(RAID(N,9)/6000.)**2)
      SUMJP = SUMJP+(XCOF*PJ/RRJS)
  240 CONTINUE
C
      IF (SUMJP.LE.0.0) GO TO 260
      RGI = AREA*SQRT(X(903+(ID-1)*20))/SUMJP**.25
      RGI = AMIN1(RG,RGI)
      IF (RGI**2.LE.RCPS) GO TO 250
      GO TO 270
  250 ICPA = 0
      RGI = 0.0
      GO TO 270
  260 RGI = RG
  270 IF (X(55).EQ.0.) GO TO 280
      CALL GETIMX (IET)
      T = IET*.0000026
      WRITE (6,300) T
  280 RETURN
C
  290 FORMAT ('  JAMM IN AT ',F15.8)
  300 FORMAT ('  JAMM OUT AT',F15.8)
      END
```

```
      SUBROUTINE BIDET (N,NSR,ALT,IR,VEL,SIGMA,ND,TD,TL,RR,TD2,TL2,R2,IB
     1M)
C            THIS SUBROUTINE DETERMINES WHEN A TARGET IS DETECTED BY A
C            BISTATIC RADAR SYSTEM IN EITHER A CLEAR OR A JAMMING SITUATION
C            N----------RADAR RECEIVER SHIP NUMBER (INPUT)
C            NSR--------SEARCH RADAR NUMBER (INPUT)
C            ALT--------TARGET ALTITUDE IN FEET (INPUT)
C            IR---------RAIDER NUMBER OF TARGET (INPUT)
C            SIGMA------TARGET CRUSS SECTION IN METERS SQUARED (INPUT)
C            ND---------NUMBER OF DETECTION ZONES. 0--TARGET NOT DETECTED
C                       1--ONE DETECTION ZONE, 2--TWO ZONES (OUTPUT)
C            TD---------TIME OF FIRST DETECTION (OUTPUT)
C            TL---------TIME CONTACT LOST IN FIRST DETECTION ZONE (OUTPUT)
C            RR---------RANGE FROM RECEIVER TO TARGET AT TIME OF DETECTION
C                       IN FIRST ZONE (OUTPUT)
C            TD2--------TIME OF INITIAL DETECTION IN SECOND ZONE. USED ONLY
C                       IF ND=2  (OUTPUT)
C            TL2--------IF ND=2, TIME CONTACT LOST IN SECOND ZONE (OUTPUT)
C            R2---------IF ND=2, RANGE AT WHICH TARGET FIRST DETECTED IN
C                       SECOND DETECTION ZONE (OUTPUT)
C            IBM--------FLAG INDICATING IF TARGET IS AN AIRCRAFT OR MISSILE.
C                       SET TO 0 FOR BOMBER, 1 FOR MISSILE (INPUT).
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      REAL *4JJTBR,LJBW,JRBRG,LTBRG
      DIMENSION XCOF(5), COF(5), TIME(4), TIMEI(4), T(4)
      REAL *8VD,VX,VY,A,B,C,E,F,XCOF,TIME,SUMJP,XCOFF
      REAL *8CHAR,RRAD,COF,TIMEI,A2,CC,EE,XRAD,A1,CHAS
      IF (X(55).EQ.0.) GO TO 20
      CALL GETIMX (IET)
      PIGS = IET*.000026
      WRITE (6,530) PIGS
C
C    SET SITE VALUES
C
   20 XR = SHIP(N,2)
      YR = SHIP(N,3)
      L = IFIX(SHIP(N,5)+AMF)
C
C    BISTATIC RECEIVER HEIGHT IN FEET
C
      ZR = X(L+6+NSR)
```

```
      ZRN = ZR/6000.
      LL = IFIX(X(L+3+NSR)+AMF)
      LLJ = (LL-1)*20
      CHAS = DBLE(X(902+LLJ))
      CHAR = DBLE(SIGMA)*CHAS**4
      IT = IFIX(X(917+LLJ)+AMF)
      XX = SHIP(IT,2)
      YX = SHIP(IT,3)
C
C   BISTATIC TRANSMITTER HEIGHT IN FEET
C
      ZX = X(918+LLJ)
      ZXN = ZX/6000.
C
C   SET TRACK VALUES
C
      X1 = RAID(IR,3)
      Y1 = RAID(IR,4)
      X0 = RAID(IR,14)
      Y0 = RAID(IR,15)
      Z1 = ALT/6000.0
      Z12 = (ALT/6000.0)**2
      V = VEL
      TIMC = RAID(IR,2)
      IFLAG = 0
      JFLAG = 0
C
C         CHECK FOR TARGET CPA AT RECEIVER OR TRANSMITTER LOCATION
C
      DELX = ABS(X0-XR)-0.01
      DELY = ABS(Y0-YR)-0.01
      IF (DELX.LT.0.0.AND.DELY.LT.0.0) GO TO 70
      DELX = ABS(X0-XX)-0.01
      DELY = ABS(Y0-YX)-0.01
      IF (DELX.LT.0.0.AND.DELY.LT.0.0) GO TO 80
C
C    DETERMINE IF NO MOTION IN X AND Y DIRECTIONS
C
      DELY = ABS(Y1-Y0)-0.1
      DELX = ABS(X1-X0)-0.1
      IF (DELX.LT.0.0.AND.DELY.LT.0.0) GO TO 40
C
C   COMPUTE CPA TO RECEIVER AND TRANSMITTER
C
      IF (DELY) 30,50,50
   30 XRC = XR
      YRC = Y1
      XXC = XX
```

```
      YXC = Y1
      GO TO 100
   40 IFLAG = 1
      GO TO 100
   50 IF (DELX) 60,90,90
   60 XRC = X1
      YRC = YR
      XXC = X1
      YXC = YX
      GO TO 100
   70 XRC = XO
      YRC = YO
      CALL CPA (X1,Y1,XO,YO,XX,YX,XXC,YXC)
      GO TO 100
   80 XXC = XO
      YXC = YO
      CALL CPA (X1,Y1,XO,YO,XR,YR,XRC,YRC)
      GO TO 100
   90 Q = (Y1-YO)/(X1-XO)
      XRC = (YR-YO+XR/Q+Q*XO)/(Q+1/Q)
      YRC = YR-(XRC-XR)/Q
      XXC = (YX-YO+XX/Q+Q*XO)/(Q+1/Q)
      YXC = YX-(XXC-XX)/Q
C
C      DETERMINE TIME OF DETECTION
C   COMPUTE RADAR HORIZON FOR TRANS AND RECEIVER
C
  100 HORRS = (X(81)**2)*(SQRT(ALT)+SQRT(ZR))**2
      HORXS = (X(81)**2)*(SQRT(ALT)+SQRT(ZX))**2
      IF (IFLAG.EQ.1) GO TO 200
      CPRS = (XR-XRC)**2+(YR-YRC)**2+(ZRN-Z1)**2
      CPXS = (XX-XXC)**2+(YX-YXC)**2+(ZXN-Z1)**2
C
C    IF CPA TO RCVR OR TRANS BEYOND RESPECTIVE RADAR HORIZON NO DET
C
      IF (CPRS.GE.HORRS.OR.CPXS.GE.HORXS) GO TO 220
      VD = DBLE(SQRT((X1-XO)**2+(Y1-YO)**2))
      VX = V*(XO-X1)/VD
      VY = V*(YO-Y1)/VD
      A = VX**2+VY**2
      A2 = A**2
      B = -2*(VX*(XRC-X1)+VY*(YRC-Y1))
      C = CPRS+(XRC-X1)**2+(YRC-Y1)**2
      D = -2*(VX*(XXC-X1)+VY*(YXC-Y1))
      E = CPXS+(XXC-X1)**2+(YXC-Y1)**2
C
C    DETERMINE TIME TARGET ABOVE BI-STATIC RADAR HORIZON(T10,T20)
C
```

```
      A1 = 2*A
      CC = C-HORRS
      RRAD = DSQRT(B**2-4*A*CC)
      TR1 = (-B-RRAD)/A1
      TR2 = (-B+RRAD)/A1
      EE = E-HORXS
      XRAD = DSQRT(D**2-4*A*EE)
      TX1 = (-D-XRAD)/A1
C
C   T10 IS TIME FIRST ABOVE BI-STATIC RADAR HORIZON, T20 TIME DOWN
C
      T10 = AMAX1(TR1,TX1)
      T20 = AMIN1(TR2,TX2)
      T10 = T10+TIMC
      T20 = T20+TIMC
C
C   DETERMINE IF TRANS/RCVR RADAR HORIZONS OVERLAP ALONG TRACK AND
C    AFTER START OF GAME AND BEFORE END OF GAME
C
      IF (T10.GE.T20.OR.T20.LE.0.0.OR.T10.GE.X(82)) GO TO 220
      XCOF(5) = 0.1D01
      XCOF(4) = (B+D)/A
      XCOF(3) = ((C+E)/A)+(B*D/A2)
      XCOF(2) = ((B*E)+(C*D))/A2
  110 XCOF(1) = (C*E-CHAR)/A2
      CALL DTIMD (XCOF,COF,4,TIME,TIMEI,IER)
      IF (IER.NE.0) GO TO 220
C
C   CHECK FOR REAL ROOTS, IE, TIMEI=0 IF ROOT IS REAL
C
      IF (TIMEI(1).NE.0.0D0.AND.TIMEI(2).NE.0.0D0.AND.TIMEI(3).NE.0.0D0.
     1AND.TIMEI(4).NE.0.0D0) GO TO 220
      KRR = 1
C
      DO 120 NNN=1,4
      IF (TIMEI(NNN).NE.0.0D0) GO TO 120
      T(KRR) = TIME(NNN)+TIMC
      KRR = KRR+1
  120 CONTINUE
C
C   CHECK FOR TWO OVALS OF CASSINI; IF TWO COMPUTE 2ND DET FIRST
C
      IF (KRR.EQ.5) GO TO 160
      IF (T(1).LT.T(2)) GO TO 130
      T1 = T(2)
      T2 = T(1)
      GO TO 140
  130 T1 = T(1)
```

```
      T2 = T(2)
C
C   CHECK FOR LOSS OF TARGET PRIOR TO START OF GAME
C
  140 IF (T2.LE.0.0.OR.T1.EQ.T2) GO TO 220
  150 IF (T1.LT.0.0) T1 = 0.0
C
C   CHECK TO SEE IF TARGET WAS DETECTED PRIOR TO START OF GAME
C
      IF (T2.LE.T10.OR.T20.LE.T1) GO TO 220
C
C   ONE DETECTION ZONE
C
      TD = AMAX1(T1,T10)
      TL = AMIN1(T2,T20)
      ND = 1
C
C   COMPUTE DETECTIONS TWO OVALS
C
      GO TO 230
C
  160 DO 170 II=1,4
  170 T(II) = TIME(II)
C
      IF (T(1).LT.T(2).AND.T(2).LT.T(3).AND.T(3).LT.T(4)) GO TO 180
      CALL TJSORT (T,4)
  180 IF (T(4).LE.0.0) GO TO 220
      IF (T(2).LE.T10.AND.T(3).GE.T20) GO TO 220
      IF (T(4).LE.T10.OR.T(1).GE.T20) GO TO 220
      IF (T(2).LE.0.0.OR.T(2).LE.T10) GO TO 190
      IF (T(3).GE.T20) GO TO 130
      IF (T(1).LE.0.0) T(1)=0.0
      TD = AMAX1(T(1),T10)
      TL = T(2)
      TD2 = T(3)
      TL2 = AMIN1(T(4),T20)
      ND = 2
      GO TO 230
  190 T1 = T(3)
      T2 = T(4)
      GO TO 150
C
C   DETERMINE IF NON-MOVING TARGET IS DETECTED
C
  200 RD2 = (X1-XR)**2+(Y1-YR)**2+Z12
      XD2 = (X1-XX)**2+(Y1-YX)**2+Z12
      IF (RD2.GT.HORRS.OR.XD2.GT.HORXS) GO TO 220
  210 IF (RD2*XD2-CHAR.GT.0.0D0) GO TO 220
```

```
      TD = 0.0                                                          BID 2400
      TL = X(82)                                                        BID 2410
      ND = 1                                                            BID 2420
      GO TO 230                                                         BID 2430
  220 ND = 0                                                            BID 2440
      GO TO 510                                                         BID 2450
  230 IF (JFLAG.EQ.1) GO TO 400                                         BID 2460
      NJ = IFIX(X(17)+AMF)                                              BID 2470
      IF (NJ.EQ.0) GO TO 400                                            BID 2480
      IJ = IFIX(X(907+LLJ)+AMF)                                         BID 2490
      IF (JAM(IJ).EQ.0) GO TO 400                                       BID 2500
      IF (IFLAG.EQ.1) GO TO 240                                         BID 2510
      TIMI = (TD+TL)/2.                                                 BID 2520
      GO TO 250                                                         BID 2530
  240 TIMI = 3.0                                                        BID 2540
C                                                                       BID 2550
C          DETERMINE BEARING FROM RADAR TO TARGET IN RADIANS            BID 2560
C                                                                       BID 2570
  250 YRT = Y1+VY*(TIMI-TIMC)-YR                                        BID 2580
      XRT = X1+VX*(TIMI-TIMC)-XR                                        BID 2590
      IF (YRT.NE.0.0.OR.XRT.NE.0.0) GO TO 260                           BID 2600
      YRT = Y1-YR                                                       BID 2610
      XRT = X1-XR                                                       BID 2620
  260 TBRG = ATAN2(YRT,XRT)                                             BID 2630
      IF (TBRG.LT.0.) TBRG=TBRG+6.2831853                               BID 2640
C                                                                       BID 2650
C          GET HALF RADAR BEAM WIDTH IN RADIANS                         BID 2660
C                                                                       BID 2670
      HRBW = X(911+LLJ)                                                 BID 2680
      RCOF = X(904+LLJ)                                                 BID 2690
      RRH = X(81)*SQRT(ZR)                                              BID 2700
      SUMJP = 0.0D0                                                     BID 2710
C                                                                       BID 2720
C          DETERMINE JAMMING LEVEL AT THE RADAR IN DIRECTION OF TARGET  BID 2730
C                                                                       BID 2740
      DO 390 L=1,NR                                                     BID 2750
      IF (TIMI.GT.RAID(L,17)) GO TO 390                                 BID 2760
      RCOFF = RCOF                                                      BID 2770
      IIR = IFIX(RAID(L,1)+AMF)                                         BID 2780
C                                                                       BID 2790
C          FIND JAMMING POWER OF RAIDER ON RADAR BAND                   BID 2800
C                                                                       BID 2810
      PJ = X(IIR+8+IJ)                                                  BID 2820
      IF (PJ.LE.0.0) GO TO 390                                          BID 2830
      CALL LOCTIM (L,TIMI,1,XJ,YJ)                                      BID 2840
C                                                                       BID 2850
C          DETERMINE IF JAMMER IS ABOVE RADAR HORIZON                   BID 2860
C                                                                       BID 2870
```

```
      RJHS = (RAID(L,10)+RRH)**2
      RRJS = (XR-XJ)**2+(YR-YJ)**2
      IF (RRJS.GT.RJHS) GO TO 390
C
C         DETERMINE COORDINATES OF JAMMER TARGET
C
      JS = IFIX(X(2206+(L-1)*6)+AMF)
      IF (JS.EQ.0) GO TO 390
      IF (JS.GT.40) GO TO 270
      XJT = SHIP(JS,2)
      YJT = SHIP(JS,3)
      GO TO 280
  270 XJT = 0.0
      YJT = 0.0
  280 YJR = YR-YJ
      XJR = XR-XJ
      IF (YJR.NE.0.0.OR.XJR.NE.0.0) GO TO 290
      CALL LOCTIM (L,0.0,1,XJJ,YJJ)
      YJR = YR-YJJ
      XJR = XR-XJJ
C
C         GET BEARING FROM JAMMER TO RADAR AND FROM RADAR TO JAMMER
C
  290 JRBRG = ATAN2(YJR,XJR)
      IF (JRBRG.LT.0.0) JRBRG=JRBRG+6.2831853
      RJBRG = JRBRG-3.1415926
      IF (RJBRG.LT.0.0) RJBRG=RJBRG+6.2831853
C
C         FIND BEARING FROM JAMMER TO JAMMER'S TARGET
C
      YJJT = YJT-YJ
      XJJT = XJT-XJ
      IF (YJJT.NE.0.0.OR.XJJT.NE.0.0) GO TO 300
      CALL LOCTIM (L,0.0,1,XJJ,YJJ)
      YJJT = YJT-YJJ
      XJJT = XJT-XJJ
  300 JJTBR = ATAN2(YJJT,XJJT)
      IF (JJTBR.LT.0.0) JJTBR=JJTBR+6.2831853
C
C         GET JAMMER SIDE-LOBE COEFFICIENT AND HALF-BEAM WIDTH
C
      HJBW = X(IIR+16)
      COFF = X(IIR+17)
      LJBW = JJTBR+HJBW
      RJBW = JJTBR-HJBW
C
C         DETERMINE IF RADAR IS IN MAIN BEAM OF JAMMER
C
```

```
      IF (LJBW.GT.6.2831853) GO TO 310
      IF (RJBW.LT.0.0) GO TO 320
      IF (LJBW.GT.JRBRG.AND.RJBW.LT.JRBRG) COFF=1.0
      GO TO 340
  310 LJBW = LJBW-6.2831853
      GO TO 330
  320 RJBW = RJBW+6.2831853
  330 IF (LJBW.GT.JRBRG.OR.RJBW.LT.JRBRG) COFF=1.0
C
C          DETERMINE IF JAMMER IS IN MAIN BEAM OF RADAR
C
  340 LTBRG = TBRG+HRBW
      RTBRG = TBRG-HRBW
      IF (LTBRG.GT.6.2831853) GO TO 350
      IF (RTBRG.LT.0.0) GO TO 360
      IF (LTBRG.GT.RJBRG.AND.RTBRG.LT.RJBRG) RCOFF=1.0
      GO TO 380
  350 LTBRG = LTBRG-6.2831853
      GO TO 370
  360 RTBRG = RTBRG+6.2831853
  370 IF (LTBRG.GT.RJBRG.OR.RTBRG.LT.RJBRG) RCOFF=1.0
  380 XCOFF = DBLE(COFF*RCOF)
      RRJS = (RRJS+(RAID(L,9)/6000.)**2)
      SUMJP = SUMJP+XCOFF*DBLE(PJ/RRJS)
  390 CONTINUE
C
      IF (SUMJP.LE.0.0D0) GO TO 400
      CHAR = DBLE(X(903+LLJ)**2)*SIGMA/SUMJP
      JFLAG = 1
      IF (IFLAG.EQ.1) GO TO 210
      GO TO 110
  400 IF (ND.EQ.1) GO TO 410
      R2 = DSQRT((X1+VX*(TD2-TIMC)-XR)**2+(Y1+VY*(TD2-TIMC)-YR)**2+(Z1-Z
     1RN)**2)
      IF (R2.GT.X(905+LLJ)) GO TO 420
  410 RR = DSQRT((X1+VX*(TD-TIMC)-XR)**2+(Y1+VY*(TD-TIMC)-YR)**2+(Z1-ZRN
     1)**2)
      IF (RR.GT.X(905+LLJ)) GO TO 460
      GO TO 510
  420 R2 = X(905+LLJ)
      IF (IBM.EQ.0) GO TO 430
      IL = IFIX(ASMT(NASM,4)+AMF)
      ILL = NASM+90
      CALL TMDASM (IR,IL,NSR,N,VEL,R2,TIMC,TD2,XI,YI,RIP,TLL,ILL)
      GO TO 440
  430 CALL TMDASM (IR,0,NSR,N,VEL,R2,TIMC,TD2,XI,YI,RIP,TLL,IR)
  440 IF (TD2.LT.TL2) GO TO 450
      ND = 1
```

```
      GO TO 410
  450 IF (TLL.LT.TL2) TL2=TLL
      GO TO 410
  460 RR = X(905+LLJ)
      IF (IBM.EQ.0) GO TO 470
      IL = IFIX(ASMT(NASM,4)+AMF)
      ILL = NASM+90
      CALL TMDASM (IR,IL,NSR,N,VEL,RR,TIMC,TD,XI,YI,RIP,TLL,ILL)
      GO TO 480
  470 CALL TMDASM (IR,0,NSR,N,VEL,RR,TIMC,TD,XI,YI,RIP,TLL,IR)
  480 IF (TD.LT.TL) GO TO 490
      IF (ND.EQ.2) GO TO 500
      ND = 0
      GO TO 510
  490 IF (TLL.LT.TL) TL = TLL
      GO TO 510
  500 ND = 1
      TD = TD2
      TL = TL2
      RR = R2
  510 IF (X(55).EQ.0.) GO TO 520
      CALL GETIMX (IET)
      COOL = IET*.000026
      WRITE (6,540) COOL
  520 RETURN
C
  530 FORMAT ('  BIDET IN AT ',F15.8)
  540 FORMAT ('  BIDET OUT AT',F15.8)
      END

      SUBROUTINE DTIMD (XCOF,COF,M,ROOTR,ROOTI,IER)
C
C          DETERMINES THE TIMES OF DETECTION FOR THE BISTATIC RADAR
C          BY SOLVING THE FOURTH ORDER POLYNOMIAL
C          REAL ROOTS ARE RETURNED IN THE REAL ARRAY WITH THE
C          CORRESPONDING  IMAGINARY ROOTS IN ROOTI.    THE METHOD
C          EMPLOYED IS THAT OF THE NEWTON-RAPHSON ITERATIVE TECH-
C          NIQUE.  THE FINAL ITERATIONS ON EACH ROOT ARE PERFORMED
C          USING THE ORIGINAL POLYNOMIAL RATHER THAN THE REDUCED
C          POLYNOMIAL TO AVOID ACCUMULATED ERRORS IN THE REDUCED
C          POLYNOMIAL
C
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      DIMENSION XCOF(1), COF(1), ROOTR(1), ROOTI(1)
      IFIT = 0
      N = M
      IER = 0
```

```
      IF (XCOF(N+1)) 20,50,20                                           DTM  180
   20 IF (N) 30,30,70                                                   DTM  190
C                                                                       DTM  200
C        SET ERROR CODE TO 1                                            DTM  210
C                                                                       DTM  220
   30 IER = 1                                                           DTM  230
   40 RETURN                                                            DTM  240
C                                                                       DTM  250
C        SET ERROR CODE TO 4                                            DTM  260
C                                                                       DTM  270
   50 IER = 4                                                           DTM  280
      GO TO 40                                                          DTM  290
C                                                                       DTM  300
C        SET ERROR CODE TO 2                                            DTM  310
C                                                                       DTM  320
   60 IER = 2                                                           DTM  330
      GO TO 40                                                          DTM  340
   70 IF (N-36) 80,80,60                                                DTM  350
   80 NX = N                                                            DTM  360
      NXX = N+1                                                         DTM  370
      N2 = 1                                                            DTM  380
      KJ1 = N+1                                                         DTM  390
C                                                                       DTM  400
      DO 90 L=1,KJ1                                                     DTM  410
      MT = KJ1-L+1                                                      DTM  420
   90 COF(MT) = XCOF(L)                                                 DTM  430
C                                                                       DTM  440
C                                                                       DTM  450
C        SET INITIAL VALUES                                             DTM  460
C                                                                       DTM  470
  100 XO = .00500101                                                    DTM  480
      YO = 0.01000101                                                   DTM  490
C                                                                       DTM  500
C        ZERO INITIAL VALUE COUNTER                                     DTM  510
C                                                                       DTM  520
      IN = 0                                                            DTM  530
  110 X = XO                                                            DTM  540
C                                                                       DTM  550
C        INCREMENT INITIAL VALUES AND COUNTER                           DTM  560
C                                                                       DTM  570
      XO = -10.0*YO                                                     DTM  580
      YO = -10.0*X                                                      DTM  590
C                                                                       DTM  600
C        SET X AND Y TO CURRENT VALUE                                   DTM  610
C                                                                       DTM  620
      X = XO                                                            DTM  630
      Y = YO                                                            DTM  640
      IN = IN+1                                                         DTM  650
```

```
      GO TO 130
  120 IFIT = 1
      XPR = X
      YPR = Y
C
C        EVALUATE POLYNOMIAL AND DERIVATIVES
C
  130 ICT = 0
  140 UX = 0.0
      UY = 0.0
      V = 0.0
      YT = 0.0
      XT = 1.0
      U = COF(N+1)
      IF (U) 150,280,150
C
  150 DO 160 I=1,N
      L = N-I+1
      XT2 = X*XT-Y*YT
      YT2 = X*YT+Y*XT
      U = U+COF(L)*XT2
      V = V+COF(L)*YT2
      FI = I
      UX = UX+FI*XT*COF(L)
      UY = UY-FI*YT*COF(L)
      XT = XT2
  160 YT = YT2
C
      SUMSQ = UX*UX+UY*UY
      IF (SUMSQ) 170,240,170
  170 DX = (V*UY-U*UX)/SUMSQ
      X = X+DX
      DY = -(U*UY+V*UX)/SUMSQ
      Y = Y+DY
      IF (DABS(DY)+DABS(DX)-1.0E-05) 220,180,180
C
C        STEP ITERATION COUNTER
C
  180 ICT = ICT+1
      IF (ICT-500) 140,190,190
  190 IF (IFIT) 220,200,220
  200 IF (IN-5) 110,210,210
C
C        SET ERROR CODE TO 3
C
  210 IER = 3
      GO TO 40
C
```

```
  220 DO 230 L=1,NXX
      MT = KJ1-L+1
      TEMP = XCOF(MT)
      XCOF(MT) = COF(L)
  230 COF(L) = TEMP
C
      ITEMP = N
      N = NX
      NX = ITEMP
      IF (IFIT) 260,120,260
  240 IF (IFIT) 250,110,250
  250 X = XPR
      Y = YPR
  260 IFIT = 0
      IF (DABS(Y/X)-1.0E-04) 290,270,270
  270 ALPHA = X+X
      SUMSQ = X*X+Y*Y
      N = N-2
      GO TO 300
  280 X = 0.0
      NX = NX-1
      NXX = NXX-1
  290 Y = 0.0
      SUMSQ = 0.0
      ALPHA = X
      N = N-1
  300 COF(2) = COF(2)+ALPHA*COF(1)
C
      DO 310 L=2,N
  310 COF(L+1) = COF(L+1)+ALPHA*COF(L)-SUMSQ*COF(L-1)
C
  320 ROOTI(N2) = Y
      ROOTR(N2) = X
      N2 = N2+1
      IF (SUMSQ) 330,340,330
  330 Y = -Y
      SUMSQ = 0.0
      GO TO 320
  340 IF (N) 40,40,100
      END

      SUBROUTINE CPA (XI,YI,XO,YO,XP,YP,XC,YC)
C         XI---BOMBER OR ASM START POSITION
C         YI---   "    "     "      "
C         XO---BOMBER END OF GAME POSIT,
C         YO---ASM TARGET SHIP POSIT
C         XP---REFERENCE POINT, IE FIRING SHIP
```

```
C         YP---        "        "    "       "        "
C         XC---CPA COORDINATES
C         YC--- "               "
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NK,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 110
      CALL GETIMX (IET)
      T = IET*.000026
      WRITE (6,200) T
  110 DELX = ABS(XO-XP)-0.01
      DELY = ABS(YO-YP)-0.01
      IF (DELX.LT.0.0.AND.DELY.LT.0.0) GO TO 190
      DELY = ABS(YI-YO)-0.1
      DELX = ABS(XI-XO)-0.1
      IF (DELX.LT.0.0.AND.DELY.LT.0.0) GO TO 160
      IF (DELY) 120,130,130
  120 XC = XP
      YC = YI
      GO TO 170
  130 IF (DELX) 140,150,150
  140 XC = XI
      YC = YP
      GO TO 170
  150 Q = (YI-YO)/(XI-XO)
      XC = ((YP-YO)+(XP/Q)+(Q*XO))/(Q+1/Q)
      YC = YP-(XC-XP)/Q
      GO TO 170
  160 XC = XI
      YC = YI
  170 IF (IQ.EQ.0) GO TO 180
      CALL GETIMX (IET)
      T = IET*.000026
      WRITE (6,210) T
  180 RETURN
  190 XC = XO
      YC = YO
      GO TO 170
C
  200 FORMAT ('   CPA IN AT',F15.8)
  210 FORMAT ('   CPA OUT AT',F15.8)
      END
```

```
      SUBROUTINE REACT (ICL,ITP,DELAY)
C
C        THIS SUBROUTINE COMPUTES THE REACTION TIME DELAY FOR
C        DETECTION, ACQUISITION, FIRING, ETC., WHERE
C           ICL----------CLASS OF THE REACTION TIME DISTRIBUTION,
C                        IE. WDE, FIRE, DETECT, ETC.
C           ITP----------TYPE OF DISTRIBUTION, UP TO FIVE PER CLASS
C           DELAY--------OUTPUT
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      ID = 1400+(ICL-1)*50
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      INT = Z*10
      IF (INT.EQ.0) GO TO 120
      IF (INT.EQ.10) INT = 9
      ID = ID+(ITP-1)*10+INT
      RANGE = X(ID+1)-X(ID)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      DELAY = X(ID)+Z*RANGE
  110 RETURN
  120 RANGE = X(ID+1)
      CALL RANDOM (ISEED,Z,1)
      X(13) = X(13)+1.
      DELAY = RANGE*Z
      GO TO 110
      END


      SUBROUTINE PRIOR (M,K,NMS,NB,RGE,NC,TIME,MM,MC,XM,YM)
C
C        THIS SUBROUTINE DETERMINES IF THERE IS A MORE THREATENING
C        TARGET TO BE ENGAGED AT THIS TIME BY THIS SHIP,THIS SYSTEM
C        AND THIS DIRECTOR. PRESENT LOGIC CONSIDERS THE CLOSEST
C        UNENGAGED TARGET THE MOST THREATENING.
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
```

```
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),PRI
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4PRI
     60),AMF,HISX(100),JACK(40,7),HISA(100)                              PRI
C                                                                        PRI
      IQ = X(55)                                                         PRI
      IF (IQ.EQ.0) GO TO 110                                             PRI
      CALL GETIMX (MV)                                                   PRI
      DELTAX = MV*2.6E-5                                                 PRI
      WRITE (6,210) DELTAX                                               PRI
  110 IF (M.GT.90) GO TO 120                                             PRI
      CALL LOCTIM (M,TIME,3,XP,YP)                                       PRI
      GO TO 130                                                          PRI
  120 CALL ASMTIM (M,TIME,XP,YP)                                         PRI
  130 DELTAX = ABS(SHIP(K,2)-XP)                                         PRI
      DELTAY = ABS(SHIP(K,3)-YP)                                         PRI
      IF (DELTAX.LT.5.0E-5) DELTAX=5.0E-5                                PRI
      IF (DELTAY.LT.5.0E-5) DELTAY=5.0E-5                                PRI
      RGI = SQRT(DELTAX**2+DELTAY**2)                                    PRI
      CHECK = RGE/RGI                                                    PRI
      MM = M                                                             PRI
      MC = NC                                                            PRI
      XM = XP                                                            PRI
      YM = YP                                                            PRI
C                                                                        PRI
C        GET ANOTHER CANDIDATE IF THERE IS ONE                           PRI
C                                                                        PRI
  140 CALL TARGET (TIME,MT,MORE)                                         PRI
      IF (MORE.EQ.0) GO TO 170                                           PRI
      IU = MT-300000000+2                                                PRI
      MV = IU*1.0D-5                                                     PRI
      KP = MV*100000                                                     PRI
      ID = IU-KP                                                         PRI
      KP = ID*1.0D-3                                                     PRI
      IF (KP.NE.K) GO TO 140                                             PRI
      KT = KP*1000                                                       PRI
      JP = ID-KT                                                         PRI
      MSO = JP*1.0E-2                                                    PRI
      IF (MSO.NE.NMS) GO TO 140                                          PRI
      KP = NMS*100                                                       PRI
      ID = JP-KP                                                         PRI
      NBO = ID*1.0E-1                                                    PRI
      IF (NBO.NE.NB) GO TO 140                                           PRI
      KP = NB*10                                                         PRI
      NCO = ID-KP-2                                                      PRI
      IF (MV.GT.90) GO TO 150                                            PRI
      CALL LOCTIM (MV,TIME,3,XP,YP)                                      PRI
      GO TO 160                                                          PRI
  150 CALL ASMTIM (MV,TIME,XP,YP)                                        PRI
```

```
  160 DELTAX = ABS(SHIP(K,2)-XP)
      DELTAY = ABS(SHIP(K,3)-YP)
      IF (DELTAX.LT.5.0E-5) DELTAX=5.0E-5
      IF (DELTAY.LT.5.0E-5) DELTAY=5.0E-5
      RGI = SQRT(DELTAX**2+DELTAY**2)
C
C        COMPARE THREAT VALUES
C
      CHERRY = RGE/RGI
      IF (CHECK.GE.CHERRY) GO TO 140
      CHECK = CHERRY
      MM = MV
      MC = NCO
      XM = XP
      YM = YP
      IUNYT = MT
      GO TO 140
  170 IF (MM.EQ.M) GO TO 180
      CALL REMOVE (IUNYT)
      IUNYT = M*100000+K*1000+NMS*100+NB*10+NC
      CALL STOREV (TIME,3,IUNYT)
  180 IF (X(9).EQ.0.) GO TO 190
      NSB = K*100+NMS*10+NB
      WRITE (6,220) M,MM,NSB,NC,MC
  190 IF (IQ.EQ.0) GO TO 200
      CALL GETIMX (MV)
      DELTAY = MV*2.6E-5
      WRITE (6,230) DELTAY
  200 RETURN
  210 FORMAT (' PRIOR IN AT',F15.8)
  220 FORMAT ('0',' PRIOR DEBUG DATA     INPUT TGT',I4,' OUTPUT TGT',I4/(
     122X,'BATTERY NUMBER',I5/(22X,'INPUT CHANNEL',I2,' OUTPUT CHANNEL',
     2I2)))
  230 FORMAT (' PRIOR OUT AT',F15.8)
      END


      SUBROUTINE RCHECK (IT,IR,K)
C
C        IT........TARGET NUMBER
C        IR........RAIDER NUMBER
C        K.........FLAG FOR GO/NOGO ON TARGET ASSIGNMENT
C
C        THIS SUBROUTINE EVALUATES THE PROSPECTIVE TARGET TO
C        SEE IF THE RAIDER CAN IN FACT "SEE" THE TARGET.
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
```

```
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 20
      CALL GETIMX (IET)
      TM = .000026*IET
      WRITE (6,70) TM
C
C        GET THE DISTANCE FROM THE RED RAIDER TO THE BLUE SHIP
C
   20 XS = RAID(IR,3)
      YS = RAID(IR,4)
      XT = SHIP(IT,2)
      YT = SHIP(IT,3)
      DELX = ABS(XS-XT)
      IF (DELX.LT..00005) DELX=.00005
      DELY = ABS(YS-YT)
      IF (DELY.LT..00005) DELY=.00005
      DIS = DELX**2+DELY**2
      ID = IFIX(X(1201+(IT-1)*4)+AMF)
      IC = 108+(ID-1)*40
      IB = IC+6
      HT = 15.
C
C        SET THE TARGET SHIP HEIGHT TO A NOMINAL VALUE OF 15
C        FEET AND THEN PROCEED TO DETERMINE THE HEIGHT OF HIS HIGHEST
C        RADAR, IF HE HAS ONE.
C
      DO 30 M=IC,IB
      IF (HT.LT.X(M)) HT = X(M)
   30 CONTINUE
C
C        DETERMINE THE MUTUAL RADAR HORIZON BETWEEN THE ATTACKER
C        AND THE TARGET. GET THE MAX RANGE FOR THE ATTACKER'S RADAR.
C        IF THE COMBINED RADAR HORIZONS ARE LESS THAN THE ACTUAL
C        DISTANCE, OR THE ACTUAL DISTANCE IS GREATER THQN THE RADAR'S
C        CAPABILITY, SET THE FLAG TO NO GO AND EXIT. OTHERWISE SET
C        THE FLAG TO OK AND EXIT.
C
      HTR = X(81)*SQRT(HT)
      RTR = RAID(IR,10)
      RADAR = RTR+HTR
      RADAR2 = RADAR**2
      IF (RADAR2.LT.DIS) GO TO 40
      IC = IFIX(X(2201+(IR-1)*6)+AMF)
```

```
      IB = IFIX(X(1807+(IC-1)*20)+AMF)
      RR = X(905+(IB-1)*20)
      RR2 = RR**2
      IF (DIS.GT.RR2) GO TO 40
      K = 0
      GO TO 50
   40 K = 1
   50 IF (IQ.EQ.0) GO TO 60
      CALL GETIMX (IET)
      TM = .000026*IET
      WRITE (6,80) TM
   60 RETURN
C
   70 FORMAT ('    RCHECK IN AT',F15.8)
   80 FORMAT ('    RCHECK OUT AT',F15.8)
      END

      SUBROUTINE TMDASM (IRAID,ITS,ISAMBT,NFS,VEL,SRG,TML,TIME,XI,YI,RIP
     1,TIML,M)
C
C          THIS SUBROUTINE  COMPUTES THE TIME OF ENTERING AND
C          LEAVING A GIVEN ENGAGEMENT CIRCLE
C
C
C             IRAID------------RAID LAUNCHING THE ASM
C             ITS--------------TARGET SHIP
C             ISAMBAT----------SAM BATTERY NUMBER
C             NFS--------------NUMBER OF THE SAM FIRING SHIP
C             VEL--------------ASM VELOCITY (SPEED IN NM/MIN)
C             SRG--------------MAXIMUM SAM RANGE FOR THIS ALTITUDE AND TYP
C             TML--------------TIME OF ASM LAUNCH
C             TIME-------------(OUTPUT) TIME OF MAX RANGE SAM INTERCEPT
C             XI---------------(OUTPUT) X POSIT    MAX INTERCEPT RANGE
C             YI---------------(OUTPUT) Y POSIT    MAX INTERCEPT RANGE
C             RIP--------------(OUTPUT) DISTANCE TO INTERCEPT OR
C                              DETECTION
C             TIML-------------(OUTPUT) TIME OF OUT OF RANGE
C             M----------------AIRBORNE TARGET NUMBER
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      IQ = X(55)
```

```
      IF (VEL.EQ.0.) VEL = .0001
      IF (IQ.EQ.0) GO TO 20
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,270) TIMES
   20 IFLAG = 0
      KFLAG = 0
C
C        DETERMINE THE PRESENT RANGE
C
      DEX = ABS(SHIP(NFS,2)-RAID(IRAID,3))
      DEY = SHIP(NFS,3)-RAID(IRAID,4)
      IF (ABS(DEY).LT.0.00005) DEY=.00005
      IF (DEX.LT..00005) DEX=.00005
      DIP = DEX**2+DEY**2
      RIP = SQRT(DIP)
      IF (RIP.GT.SRG) GO TO 30
      XI = RAID(IRAID,3)
      YI = RAID(IRAID,4)
      TIME = TML
      IFLAG = 1
      IF (VEL.LT..005) GO TO 260
      GO TO 40
   30 RIP = SRG
      IF (VEL.LT..005) GO TO 200
C
C         SOLVE THE LINEAR EQUATION FOR THE TARGET
C
   40 DELTAX = RAID(IRAID,14)-RAID(IRAID,3)
      DELTAY = RAID(IRAID,15)-RAID(IRAID,4)
      IF (ABS(DELTAX).LT.5.0E-2) GO TO 50
      SLM = DELTAY/DELTAX
      GO TO 60
   50 B = -2*SHIP(NFS,3)
      C = (RAID(IRAID,3)-SHIP(NFS,2))**2+SHIP(NFS,3)**2-SRG**2
      TEST = B**2-4.*C
      IF (TEST.LE.0.) GO TO 200
      TOAST = SQRT(TEST)
      X1 = RAID(IRAID,3)
      X2 = X1
      Y1 = (-B+TOAST)/2.
      Y2 = (-B-TOAST)/2.
      KFLAG = 1
      GO TO 80
   60 BI = RAID(IRAID,15)-SLM*RAID(IRAID,14)
      XS = SHIP(NFS,2)
      YS = SHIP(NFS,3)
C
```

```
C
C         SOLVE THE JOINT QUADRATIC EQUATION
C
      A = SLM**2+1.
      B = 2.*(SLM*(B1-YS)-XS)
      C = XS**2+(B1-YS)**2-SRG**2
      TEST = B**2-4.*A*C
      IF (TEST) 200,200,70
   70 ROOT = SQRT(TEST)
      X1 = (-B+ROOT)/(2.*A)
      X2 = (-B-ROOT)/(2.*A)
      Y1 = SLM*X1+B1
      Y2 = SLM*X2+B1
   80 DELX = ABS(X1-RAID(IRAID,3))
      IF (DELX.LT..00005) DELX=.00005
      DELY = ABS(Y1-RAID(IRAID,4))
      IF (DELY.LT..00005) DELY=.00005
      D1 = SQRT(DELX**2+DELY**2)
      DELX = ABS(X2-RAID(IRAID,3))
      IF (DELX.LT..00005) DELX=.00005
      DELY = ABS(Y2-RAID(IRAID,4))
      IF (DELY.LT..00005) DELY=.00005
      D2 = SQRT(DELX**2+DELY**2)
      IF (IFLAG.EQ.1) GO TO 120
      IF (KFLAG.EQ.1) GO TO 160
      DIF = ABS(DELTAX)
      IF (D2.LT.D1) GO TO 100
      DIC = ABS(RAID(IRAID,14)-X1)
   90 IF (DIF.LT.DIC) GO TO 200
C
C         DETERMINE THE INTERSECTION TIMES
C
      XI = X1
      YI = Y1
      TIME = TML+D1/VEL
      TIML = TML+D2/VEL
      GO TO 220
  100 DIC = ABS(RAID(IRAID,14)-X2)
  110 IF (DIF.LT.DIC) GO TO 200
      XI = X2
      YI = Y2
      TIME = TML+D2/VEL
      TIML = TML+D1/VEL
      GO TO 220
  120 IF (KFLAG.EQ.1) GO TO 180
      IF (DELTAX.LT.0.) GO TO 150
      IF (X1.LT.X2) GO TO 140
  130 TIML = TML+D1/VEL
      GO TO 220
```

```
140 TIML = TML+D2/VEL                                                   TMD 1270
    GO TO 220                                                           TMD 1280
150 IF (X1.LT.X2) GO TO 130                                             TMD 1290
    GO TO 140                                                           TMD 1300
160 DIF = ABS(DELTAY)                                                   TMD 1310
    IF (D2.LT.D1) GO TO 170                                             TMD 1320
    DIC = ABS(RAID(IRAID,15)-Y1)                                        TMD 1330
    GO TO 90                                                            TMD 1340
170 DIC = ABS(RAID(IRAID,15)-Y2)                                        TMD 1350
    GO TO 110                                                           TMD 1360
180 IF (DELTAY.LT.0.) GO TO 190                                         TMD 1370
    IF (Y1.LT.Y2) GO TO 140                                             TMD 1380
    GO TO 130                                                           TMD 1390
190 IF (Y1.LT.Y2) GO TO 130                                             TMD 1400
    GO TO 140                                                           TMD 1410
200 TIME = 9120.                                                        TMD 1420
    XI = 9999.                                                          TMD 1430
    YI = 9999.                                                          TMD 1440
    RIP = 0.                                                            TMD 1450
    IF (X(57).EQ.0.) GO TO 240                                          TMD 1460
    IF (ITS.EQ.0) GO TO 210                                             TMD 1470
    WRITE (6,280) IRAID,ITS,ISAMBT,NFS,M                                TMD 1480
    GO TO 240                                                           TMD 1490
210 WRITE (6,310) M,NFS                                                 TMD 1500
    GO TO 240                                                           TMD 1510
220 IF (X(57).EQ.0.) GO TO 240                                          TMD 1520
    IF (ITS.EQ.0) GO TO 230                                             TMD 1530
    WRITE (6,290) M,NFS,ITS,SRG,TIME,TIML                               TMD 1540
    GO TO 240                                                           TMD 1550
230 WRITE (6,320) M,NFS,ISAMBT,SRG,TIME,TIML                            TMD 1560
240 IF (IQ.EQ.0) GO TO 250                                              TMD 1570
    CALL GETIMX (IET)                                                   TMD 1580
    TIMES = .000026*IET                                                 TMD 1590
    WRITE (6,300) TIMES                                                 TMD 1600
250 RETURN                                                              TMD 1610
260 TIML = X(82)                                                        TMD 1620
    GO TO 220                                                           TMD 1630
C                                                                       TMD 1640
270 FORMAT ('            TMDASM IN AT ',F15.8)                          TMD 1650
280 FORMAT (' NO ENGAGEMENT POSSIBLE, RAID ',I4,' TARGET SHIP',I3, '    TMD 1660
   1 BATTERY',I2,' MISSILE SHIP',I3,' ASM',I3)                          TMD 1670
290 FORMAT ('0',' ENGAGEMENT DATA, ASM',I4,' AND SHIP',I3,/(' TARGET TMD 1680
   1SHIP   DETEC RANGE   GAIN CONTACT   LOOSE CONTACT'/(I6,9X,F10.3,6XTMD 1690
   2,F7.3,9X,F7.3)))                                                    TMD 1700
300 FORMAT ('            TMDASM OUT AT ',F15.8)                         TMD 1710
310 FORMAT ('0',' NO ENGAGEMENT POSSIBLE BETWEEN BOMBER',I3,' AND SHITMD 1720
   1P',I3)                                                              TMD 1730
320 FORMAT ('0',' ENGAGEMENT DATA, BOMBER',I3,' AND SHIP',I3/(4X,'BATTMD 1740
```

```
      SUBROUTINE TMDBOM (IR,RGE,T)
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MANBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGI(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
C
C        COMPUTES THE TIME THE RAIDER WILL REACH A GIVEN RANGE FROM
C        FORCE CENTER.
C        IR......BOMBER NUMBER
C        RGE.....GIVEN RANGE IN MILES
C        T.......(OUTPUT) TIME OF ATTAINING THE RANGE. IF PRESENT
C                GAME TIME.PUTS BOMBER INSIDE GIVEN RANGE, PRESENT
C                GAME TIME IS RETURNED.
C
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 20
      CALL GETIMX (IET)
      TM = .000026*IET
      WRITE (6,110) TM
C
C        COMPUTE PRESENT RAIDER RANGE FROM FORCE CENTER
C
   20 DIP = RAID(IR,3)**2+RAID(IR,4)**2
      RIP = SQRT(DIP)
      IF (RIP.GT.RGE) GO TO 30
      T = RAID(IR,2)
      GO TO 80
   30 DELTAX = RAID(IR,5)-RAID(IR,3)
      IP = IFIX(X(2201+(IR-1)*6)+AMF)
      VEL = X(1814+(IP-1)*20)
      DELTAY = RAID(IR,6)-RAID(IR,4)
C
C        SOLVE THE LINEAR EQUATION FOR RAIDER TRACK AND THE
C        QUADRATIC FOR INPUT RANGE SIMULTANEOUSLY
C
      IF (ABS(DELTAX).LT.5.0E-2) GO TO 40
      SLM = DELTAY/DELTAX
      GO TO 50
   40 IF (ABS(RAID(IR,3)).GE.RGE.OR.DELTAY*RAID(IR,4).GT.0.) GO TO 100
      IF (VEL.EQ.0.) GO TO 100
```

```
      T = RAID(IR,2)+(ABS(RAID(IR,4))-SQRT(RGE**2-RAID(IR,3)**2))/VEL
      IF (T.GT.X(82).OR.T.LT.RAID(IR,2)) GO TO 100
      GO TO 80
   50 BA = RAID(IR,6)-SLM*RAID(IR,5)
      A = SLM**2+1.
      B = 2.*SLM*BA
      C = BA**2-RGE**2
      TEST = B**2-4.*A*C
      IF (TEST) 100,100,60
   60 RCOT = SQRT(TEST)
      X1 = (-B+ROOT)/(2.*A)
      X2 = (-B-ROOT)/(2.*A).
      Y1 = SLM*X1+BA
      Y2 = SLM*X2+BA
      DELX = ABS(X1-RAID(IR,3))
      IF (DELX.LT..00005) DELX=.00005
      DELY = ABS(Y1-RAID(IR,4))
      IF (DELY.LT..00005) DELY=.00005
      D1 = DELX**2+DELY**2
      DELX = ABS(X2-RAID(IR,3))
      IF (DELX.LT..00005) DELX=.00005
      DELY = ABS(Y2-RAID(IR,4))
      IF (DELY.LT..00005) DELY=.00005
      D2 = DELX**2+DELY**2
      IF (VEL.EQ.0.) GO TO 100
      DIF = ABS(RAID(IR,5)-RAID(IR,3))
      IF (D2.LT.D1) GO TO 70
      DIC = ABS(RAID(IR,5)-X1)
      IF (DIF.LT.DIC) GO TO 100
C
C        COMPUTE TIME RAIDER IS WITHIN GIVEN RANGE
C
      DIX = SQRT(D1)
      T = RAID(IR,2)+DIX/VEL
      GO TO 80
   70 DIC = ABS(RAID(IR,5)-X2)
      IF (DIF.LT.DIC) GO TO 100
      DIX = SQRT(D2)
      T = RAID(IR,2)+DIX/VEL
   80 IF (IQ.EQ.0) GO TO 90
      CALL GETIMX (IET)
      TM = IET*.000026
      WRITE (6,120) TM
   90 RETURN
  100 T = 9120.
      GO TO 80
C
  110 FORMAT ('   TMDBOM IN AT',F15.8)
```

```
  120 FORMAT ('  TMDBOM OUT AT',F15.8)
      END

      SUBROUTINE ZLNKA (A,KEY,N)
C
C        THIS SUBROUTINE CAN BE FOUND IN THE NPGS SUBROUTINE
C        LIBRARY UNDER THE NAME SHSORT
C
C        THE SUBROUTINE REORDERS THE INPUT ARRAY "A" FROM SMALLEST
C        TO LARGEST, AND SIMULTANEOUSLY REORDERS THE INTEGER
C        ARRAY "KEY" TO THE SAME ORDER AS "A".
C
      DIMENSION A(N), KEY(N)
      M1 = 1
   20 M1 = M1*2
      IF (M1.LE.N) GO TO 20
      M1 = M1/2-1
      MM = MAX0(M1/2,1)
      GO TO 40
   30 MM = MM/2
      IF (MM.LE.0) GO TO 70
   40 K = N-MM
C
      DO 60 J=1,K
      II = J
   50 IM = II+MM
      IF (A(IM).GE.A(II)) GO TO 60
      TEMP = A(II)
      IT = KEY(II)
      A(II) = A(IM)
      KEY(II) = KEY(IM)
      A(IM) = TEMP
      KEY(IM) = IT
      II = II-MM
      IF (II.GT.0) GO TO 50
   60 CONTINUE
C
      GO TO 30
   70 RETURN
      END

      SUBROUTINE TJSORT (A,N)
C
C      ID:          M1-NPG-SHSORT  (F-IV)
C
C       A  -  ARRAY OF NUMBERS TO BE SORTED. THIS ARRAY IS SORTED
```

```
C           (RE-ORDERED) BY A SHELL SORT.
C      N  -  NUMBER OF MEMBERS IN ARRAY  "A".
C
      DIMENSION A(N)
      M1 = 1
   20 M1 = M1*2
      IF (M1.LE.N) GO TO 20
      M1 = M1/2-1
      MM = MAXO(M1/2,1)
      GO TO 40
   30 MM = MM/2
      IF (MM.LE.0) GO TO 70
   40 K = N-MM
C
      DO 60 J=1,K
      II = J
   50 IM = II+MM
      IF (A(IM).GE.A(II)) GO TO 60
      TEMP = A(II)
      A(II) = A(IM)
      A(IM) = TEMP
      II = II-MM
      IF (II.GT.0) GO TO 50
   60 CONTINUE
C
      GO TO 30
   70 RETURN
      END


      SUBROUTINE LOCTIM (K,T,IND,XP,YP)
C
C         THIS SUBROUTINE COMPUTES THE PRESENT, FUTURE OR PAST BOMBER
C         POSITION. THE INPUTS ARE:
C               K------------BOMBER NUMBER
C               T------------THE TIME OF INTEREST
C               IND----------AN INDICATOR;  DO YOU WANT THIS INFORMATION
C                            STORED IN THE RAID ARRAY?
C               XP-----------(OUTPUT) POISTION AT TIME T.
C               YP-----------(OUTPUT) POISTION AT TIME T.
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
```

```
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 20
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,70) TIMES
   20 ID = IFIX(RAID(K,1)+AMF)
      VEL = X(ID+13)
      PTIME = RAID(K,2)
      DL = (T-PTIME)*VEL
      YZ = RAID(K,4)
      XZ = RAID(K,3)
      DELTAX = XZ-RAID(K,5)
      IF (ABS(DELTAX).LT..00005) DELTAX=).0
      DELTAY = YZ-RAID(K,6)
      IF (ABS(DELTAY).LT..00005) DELTAY=0.0
      D = SQRT(DELTAX**2+DELTAY**2)
      IF (D.EQ.0.) GO TO 60
      DT = DL/D
      XP = XZ-DELTAX*DT
      YP = YZ-DELTAY*DT
   30 IF (IND.NE.5) GO TO 40
      RAID(K,4) = YP
      RAID(K,3) = XP
      RAID(K,2) = T
   40 IF (IQ.EQ.0) GO TO 50
      CALL GETIMX (IET)
      TIMES = IET*.000026
      WRITE (6,80) TIMES
   50 RETURN
   60 XP = XZ
      YP = YZ
      GO TO 30
C
   70 FORMAT ('      LOCTIM IN AT',F15.8)
   80 FORMAT ('    LOCTIM OUT AT',F15.8)
      END

      SUBROUTINE ASMTIM (M,T,XP,YP)
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SMA
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
C
```

```
      IQ = X(55)
      IF (IQ.EQ.0) GO TO 20
      CALL GETIMX (ID)
      VEL = ID*0.000026
      WRITE (6,60) VEL
   20 MA = M-90
      ID = IFIX(ASMT(MA,6)+AMF)
      VEL = X(2001+(ID-1)*10)
      DL = (T-ASMT(MA,5))*VEL
      ID = IFIX(ASMT(MA,4)+AMF)
      DELTAX = ASMT(MA,7)-SHIP(ID,2)
      IF (ABS(DELTAX).LT.5.0E-5) DELTAX=0.0
      DELTAY = ASMT(MA,8)-SHIP(ID,3)
      IF (ABS(DELTAY).LT.5.0E-5) DELTAY=0.0
      D = DELTAX**2+DELTAY**2
      IF (D.EQ.0.) GO TO 50
      D = SQRT(D)
      DT = DL/D
      IF (ABS(DT).LT.1.0E-4) DT=0.0
      XP = ASMT(MA,7)-DELTAX*DT
      YP = ASMT(MA,8)-DELTAY*DT
   30 IF (IQ.EQ.0) GO TO 40
      CALL GETIMX (ID)
      VEL = ID*0.000026
      WRITE (6,70) VEL
   40 RETURN
   50 XP = ASMT(MA,7)
      YP = ASMT(MA,8)
      GO TO 30
C
   60 FORMAT (' ASMTIM IN AT',F15.8)
   70 FORMAT (' ASMTIM OUT AT',F15.8)
      END


C
C     SUBROUTINE  HISTG/HISTF         (J5)
C
C     PURPOSE
C
C         SUBROUTINE HISTG IS INTENDED TO PRODUCE A ONE-PAGE SUMMARY
C     OF A SET OF DATA BY COMPUTING A SET OF BASIC STATISTICS AND
C     PRINTING A HISTOGRAM.  IN ADDITION, THE CAPABILITY OF DISPLAYING
C     A SMOOTHED EMPIRICAL DENSITY FUNCTION PLOT IS PROVIDED.
C
```

```
C     CALLING SEQUENCES                                                  HSTG0120
C                                                                        HSTG0130
C        CALL HISTG(X, N, NBAR)                                          HSTG0140
C                                                                        HSTG0150
C        CALL HISTF(X, N, NBAR)                                          HSTG0160
C                                                                        HSTG0170
C        CALL NOSTAT                                                     HSTG0180
C                                                                        HSTG0190
C        CALL FIX(SCALE)                                                 HSTG0200
C                                                                        HSTG0210
C        CALL NOFIX                                                      HSTG0220
C                                                                        HSTG0230
C     ARGUMENTS                                                          HSTG0240
C                                                                        HSTG0250
C        X      A SINGLE DIMENSIONAL ARRAY OF DATA (REAL*4) TO BE        HSTG0260
C               ANALYZED. SORTED IN ASCENDING ORDER ON OUTPUT FROM       HSTG0270
C               HISTF/HISTG.                                             HSTG0280
C                                                                        HSTG0290
C        N      THE NUMBER OF DATA POINTS TO BE CONSIDERED. MUST BE      HSTG0300
C               GREATER THAN 9.                                          HSTG0310
C                                                                        HSTG0320
C        NBAR   THE NUMBER OF BARS IN THE HISTOGRAM.  MUST BE BETWEEN    HSTG0330
C               15 AND 32.  IF NBAR IS SUPPLIED AS ZERO, THE NUMBER OF   HSTG0340
C               BARS IN THE HISTOGRAM WILL BE AUTOMATICALLY DETERMINED   HSTG0350
C               BASED ON THE NUMBER OF DATA POINTS.                      HSTG0360
C                                                                        HSTG0370
C        SCALE  A VECTOR (REAL*4) OF TWO VALUES USED TO OPTIONALLY FIX   HSTG0380
C               THE SCALE FOR THE HISTOGRAM.  SCALE(1) IS THE LOWER      HSTG0390
C               VALUE AND SCALE(2) THE UPPER.                            HSTG0400
C                                                                        HSTG0410
C     USAGE                                                              HSTG0420
C                                                                        HSTG0430
C         THE CALLING SEQUENCE USING THE ENTRY POINT HISTG WILL PRODUCE  HSTG0440
C     A SINGLE PAGE OF OUTPUT WITH THE HISTOGRAM AND THE COMPUTED        HSTG0450
C     STATISTICS (BUT SEE ERROR CONDITIONS BELOW).  USING THE ALTERNATE  HSTG0460
C     ENTRY POINT HISTF WILL PRODUCE TWO PAGES OF OUTPUT:  THE FIRST     HSTG0470
C     WILL BE IDENTICAL TO THE PAGE FOR HISTG WHILE THE SECOND WILL      HSTG0480
C     SHOW THE EMPIRICAL DENSITY FUNCTION PLOTTED OVER THE BASIC         HSTG0490
C     HISTOGRAM.  THE CALCULATION AND PRINT OUT OF STATISTICS MAY        HSTG0500
C     BE SUPPRESSED BY CALLING NOSTAT.                                   HSTG0510
C                                                                        HSTG0520
C         HISTG/HISTF CALLS ON SUBROUTINE PXSORT TO PERFORM AN IN-PLACE  HSTG0530
C     SORT OF THE INPUT DATA SO THAT IT WILL BE IN INCREASING ORDER      HSTG0540
C     UPON RETURN FROM THE SUBROUTINE.  THE VALUES OF THE DATA POINTS    HSTG0550
C     ARE UNCHANGED.                                                     HSTG0560
C                                                                        HSTG0570
C         THE HISTOGRAM SCALE MAY BE FIXED BY CALLING THE ALTERNATE      HSTG0580
C     ENTRY POINT FIX; THE SCALE REMAINS SET AT THESE LIMITS UNLESS      HSTG0590
```

```
C     IT IS RESET BY ANOTHER CALL TO FIX OR ALLOWED TO FLOAT (THE          HSTG0600
C     DEFAULT) BY CALLING NOFIX.  WHEN THE SCALE IS FIXED, POINTS          HSTG0610
C     FALLING OUTSIDE THE INTERVAL ARE DISPLAYED IN THE END BARS           HSTG0620
C     OF THE HISTOGRAM.                                                    HSTG0630
C                                                                          HSTG0640
C        THE FOLLOWING BASIC STATISTICS ARE ALSO COMPUTED AND PRINTED      HSTG0650
C     OUT, IF NOSTAT IS NOT CALLED:                                        HSTG0660
C        MEAN, MEDIAN, TRIMEAN, MIDMEAN                                    HSTG0670
C        GEOMETRIC AND HARMONIC MEANS (POSITIVE SAMPLES ONLY)              HSTG0680
C        VARIANCE, STANDARD DEVIATION, COEFFICIENT OF VARIATION, RANGE     HSTG0690
C           AND MIDSPREAD                                                  HSTG0700
C        THIRD AND FOURTH CENTRAL MOMENTS, COEFFICIENTS OF SKEWNESS AND    HSTG0710
C           KURTOSIS                                                       HSTG0720
C        MAXIMUM, MINIMUM AND 5 QUANTILES                                  HSTG0730
C     IN ADDITION, THE MEAN IS DISPLAYED ON THE HISTOGRAM BY A VERTICAL    HSTG0740
C     COLUMN OF "M"'S AND THE QUARTILES BY COLUMNS OF DOTS.                HSTG0750
C                                                                          HSTG0760
C     INTERPRETING THE OUTPUT                                              HSTG0770
C                                                                          HSTG0780
C        THE DEFINITIONS OF THE BASIC STATISTICS COMPUTED BY               HSTG0790
C     HISTG/HISTF ARE LISTED BELOW.  PAGE NUMBER REFERENCES ARE            HSTG0800
C     TO THE CRC STANDARD MATH TABLES, 19TH EDITION (1971).                HSTG0810
C                                                                          HSTG0820
C     MEAN       AVERAGE OF THE SAMPLE (P 554).                            HSTG0830
C                                                                          HSTG0840
C     MEDIAN     MID-VALUE OF THE SAMPLE, IF THERE ARE AN ODD NUMBER       HSTG0850
C                OF SAMPLE POINTS, OR THE AVERAGE OF THE TWO MIDDLE        HSTG0860
C                VALUES FOR AN EVEN NUMBER OF POINTS (P 555).              HSTG0870
C                                                                          HSTG0880
C     TRIMEAN    0.25 * (Q  + 2Q  + Q ), WHERE THE Q  ARE THE QUARTILES.   HSTG0890
C                         1     2    3             I                       HSTG0900
C                                                                          HSTG0910
C     MIDMEAN    THE AVERAGE OF ALL SAMPLE VALUES LYING BETWEEN THE        HSTG0920
C                UPPER AND LOWER QUARTILES.                                HSTG0930
C                                                                          HSTG0940
C     MIDRANGE   AVERAGE OF THE MAXIMUM AND MINIMUM.                       HSTG0950
C                                                                          HSTG0960
C     GEOMETRIC (P 554).                                                   HSTG0970
C       MEAN                                                               HSTG0980
C                                                                          HSTG0990
C     HARMONIC  (P 555).                                                   HSTG1000
C       MEAN                                                               HSTG1010
C                                                                          HSTG1020
C     VARIANCE  (P 557). UNBIASED ESTIMATORS FOR VARIANCE AND              HSTG1030
C               STANDARD DEVIATION ARE USED.                               HSTG1040
C                                                                          HSTG1050
C     STANDARD  (P 557).                                                   HSTG1060
C     DEVIATION                                                            HSTG1070
```

```
C                                                                               HSTG1080
C     COEFFICIENT OF VARIATION = STANDARD DEVIATION / |MEAN|                    HSTG1090
C               WHEN THE MEAN IS LESS THAN 10. ** -30, THE COEFFICIENT          HSTG1100
C               OF VARIATION IS SET TO ZERO.                                    HSTG1110
C                                                                               HSTG1120
C     MEAN      (P 556). THE AVERAGE OF THE SUM OF THE ABSOLUTE                 HSTG1130
C     DEVIATION DIFFERENCES BETWEEN THE SAMPLE VALUES AND THE                   HSTG1140
C               MEDIAN                                                          HSTG1150
C                                                                               HSTG1160
C     RANGE     MAXIMUM - MINIMUM (P 557).                                      HSTG1170
C                                                                               HSTG1180
C     MIDSPREAD Q  - Q  , ALSO CALLED INTERQUARTILE DISTANCE.                   HSTG1190
C                3    1                                                         HSTG1200
C                                                                               HSTG1210
C     M3        THIRD CENTRAL MOMENT.  UNBIASED ESTIMATOR IS USED.              HSTG1220
C                                                                               HSTG1230
C     M4        FOURTH CENTRAL MOMENT.  UNBIASED ESTIMATOR IS USED.             HSTG1240
C                                                                               HSTG1250
C     COEFFICIENT OF SKEWNESS  M3 / (STD DEV) ** 3                              HSTG1260
C                                                                               HSTG1270
C     COEFFICIENT OF KURTOSIS  (M4 / (STD DEV) ** 4) - 3                        HSTG1280
C                                                                               HSTG1290
C     BETA1     BIASED ESTIMATE OF THIRD CENTRAL MOMENT. CAN BE USED            HSTG1300
C               IN TESTING FOR NORMALITY.                                       HSTG1310
C                                                                               HSTG1320
C     BETA2     BIASED ESTIMATE OF FOURTH CENTRAL MOMENT.                       HSTG1330
C                                                                               HSTG1340
C     MAXIMUM   LARGEST SAMPLE VALUE.                                           HSTG1350
C                                                                               HSTG1360
C     MINIMUM   SMALLEST SAMPLE VALUE.                                          HSTG1370
C                                                                               HSTG1380
C     QUANTILES THE ALPHA-QUANTILE, X(ALPHA), IS THE SOLUTION TO                HSTG1390
C               THE EQUATION                                                    HSTG1400
C                  PROBABILITY (X .LE. X(ALPHA) ) = ALPHA.                      HSTG1410
C                                                                               HSTG1420
C                                                                               HSTG1430
C         THE EMPIRICAL DENSITY FUNCTION IS PLOTTED WHEN THE ENTRY POINTHSTG1440
C     HISTF IS USED.  THE EMPIRICAL DENSITY FUNCTION GIVES ESSENTIALLY          HSTG1450
C     A "MORE EXACT" PICTURE OF THE DATA THAN DOES THE HISTOGRAM                HSTG1460
C     ALONE.  IT IS DEFINED BY THE RELATION                                     HSTG1470
C                                                                               HSTG1480
C                          1        N                                           HSTG1490
C           F (Z)   =   --------   SUM  W( (X  - Z) / B(N) ),                   HSTG1500
C            N           N*B(N)    I=1        I                                 HSTG1510
C                                                                               HSTG1520
C     WHERE N IS THE NUMBER OF DATA POINTS, B(N) IS A BANDWIDTH                 HSTG1530
C     FUNCTION,                                                                 HSTG1540
C                                                                               HSTG1550
```

```
C           B(N) = RANGE / SQRT(N),                                      HSTG1560
C                                                                        HSTG1570
C     AND W IS A WEIGHT FUNCTION,                                        HSTG1580
C                                                                        HSTG1590
C           W(Z) = 0          IF |Z| > 1                                 HSTG1600
C                = 1 - |Z|   OTHERWISE.                                  HSTG1610
C                                                                        HSTG1620
C     F (Z) IS COMPUTED FOR VALUES OF Z BETWEEN THE MAXIMUM AND THE     HSTG1630
C      N                                                                 HSTG1640
C     MINIMUM OF THE SAMPLE AND PLOTTED ON TOP OF THE HISTOGRAM          HSTG1650
C     USING THE SYMBOL "F".  THE RELATIVE FREQUENCY MARKS ON THE         HSTG1660
C     LEFT OF THE OUTPUT REFER TO THE HISTOGRAM AND NOT TO THE           HSTG1670
C     DENSITY FUNCTION.                                                  HSTG1680
C                                                                        HSTG1690
C     SUBROUTINES REQUIRED                                               HSTG1700
C                                                                        HSTG1710
C         SUBROUTINE PXSORT IS USED TO ORDER THE SAMPLE.  PXSORT HAS     HSTG1720
C         BEEN COMPILED AND ADDED TO MPSLIB UNDER OS.                    HSTG1730
C                                                                        HSTG1740
C     TIMING AND CORE REQUIREMENTS                                       HSTG1750
C                                                                        HSTG1760
C         HISTG/HISTF REQUIRE 15,984 DECIMAL BYTES OF CORE.  WITH ALL    HSTG1770
C     THE IBM BUILT-IN FUNCTIONS (IBCOM, SQRT, ALOG, ETC.) AND           HSTG1780
C     SUBROUTINE PXSORT, THE TOTAL REQUIREMENT IS ABOUT 38,500 BYTES.    HSTG1790
C                                                                        HSTG1800
C         THE FOLLOWING APPROXIMATE TIME REQUIREMENTS FOR AN IBM/360-67 HSTG1810
C     WERE OBSERVED:                                                     HSTG1820
C                                                                        HSTG1830
C         SAMPLE SIZE (N)         HISTG          HISTF                   HSTG1840
C                                                                        HSTG1850
C             1000                2.17 SEC       3.97 SEC                HSTG1860
C              500                1.73 SEC       3.40 SEC                HSTG1870
C              200                1.49 SEC       3.04 SEC                HSTG1880
C              100                1.42 SEC       2.91 SEC                HSTG1890
C                                                                        HSTG1900
C                                                                        HSTG1910
C     ERROR CONDITIONS                                                   HSTG1920
C                                                                        HSTG1930
C     SAMPLE SIZE MUST BE AT LEAST 10; IF NOT, AN ERROR MESSAGE IS       HSTG1940
C          PRINTED AND NO OUTPUT IS PRODUCED.                            HSTG1950
C                                                                        HSTG1960
C     THERE MUST BE SOME DIFFERENCES IN THE DATA; IF ALL VALUES OF       HSTG1970
C          X ARE THE SAME, AN ERROR MESSAGE IS PRINTED.                  HSTG1980
C                                                                        HSTG1990
C     PROGRAMMER : D.W. ROBINSON                                         HSTG2000
C                                                                        HSTG2010
C                 ADAPTED FROM A PREVIOUS VERSION CALLED 'HIST'          HSTG2020
C                 PROVIDED BY IBM AND MODIFIED AT NPS BY                 HSTG2030
```

```
C                   S.D. RANEY 11/72.
C
C     DATE :          DEC 73
C     REVISION 1:     APR 74     ADDED ADDITIONAL STATISTICS AND CALLED
C                                NEW SORT ROUTINE. PUT IN NOSTAT, FIX AND
C                                NOFIX SECTIONS.
C
C
C-----------------------------------------------------------------------
C
C
      SUBROUTINE  HISTG  (X, N, NBAR)
      INTEGER     F, ARRAY, BLK, BMARK, PROB, RVERT
      REAL        MIDMN, MIDSPD, MNDEV, MDRNG
      LOGICAL*1   PLOT, EM, DOT, VERT, CROSS, BLANK, NUM, FMT, FUNC, EFF
      LOGICAL*1   STAT, FX
      REAL*8   SUM, SUM2, SUM3, SUM4, SUMH
      DIMENSION  X(N), XLABEL(33), F(32), PROB(50), NUM(4), FMT(24)
      DIMENSION  ARRAY(1600), PLOT(128,50), FN(127), SCALE(2)
      EQUIVALENCE  (PLOT(1,1), ARRAY(1))
C
      DATA BLK/'    '/, MARK/'*** '/, BMARK/'***+'/, NOMARK/'---+'/
      DATA RVERT/'   |'/, FUNC/.TRUE./, STAT/.TRUE./, FX/.FALSE./
      DATA NOUT/6/
      DATA  EFF/'F'/,EM/'M'/,DOT/'.'/,VERT/'|'/,CROSS/'+'/, BLANK/' '/
C-----THIS STATEMENT WILL PRODUCE A SIZE WARNING:-
      DATA  NUM/4H3210/, FMT/24H(1XF6.2,15(3XF5.2),F6.2)/
C
C     SET FLAG SO EMPIRICAL D.F. IS NOT PLOTTED
      FUNC = .FALSE.
C
C     ENTRY POINT FOR DENSITY FUNCTION PLOT ROUTINE
      ENTRY HISTF (X, N, NBAR)
C
C     TEST N
      AN = N
      IF(N .GE. 10) GO TO 10
      WRITE(NOUT,5) N
    5 FORMAT('1***HISTG  SAMPLE SIZE TOO SMALL:',I11)
      FUNC = .TRUE.
      RETURN
C
C     SORT THE SAMPLE
   10 CALL PXSORT(X, 1, N)
C
C     GET FREQUENCY INTERVALS FOR BARS AND COUNT NO VALUES IN EACH
C-----SEE WHETHER TO USE AUTO NBARS DETERMINATION
      NBARS = NBAR
```

```
      IF ( (NBAR .GE. 15) .AND. (NBAR .LE. 32) ) GO TO 20               HSTG2520
      NBARS = 15                                                        HSTG2530
      IF(N .GE. 80) NBARS = MINO(32, N/5)                               HSTG2540
   20 MAX = NBARS * 4                                                   HSTG2550
      NBP1 = NBARS + 1                                                  HSTG2560
      NBM1 = NBARS - 1                                                  HSTG2570
      LIM E = MINO(NBP1, 32)                                            HSTG2580
      RANGE = X(N) - X(1)                                               HSTG2590
      IF(RANGE .GT. 0.0) GO TO 40                                       HSTG2600
C     RANGE OF SAMPLE IS .LE. 0; PRINT ERROR MESSAGE                    HSTG2610
      WRITE(NOUT,30) X(1)                                               HSTG2620
   30 FORMAT('1***HISTG  SAMPLE IS CONSTANT. SAMPLE VALUES ARE',1PE16.6)HSTG2630
      GO TO 450                                                         HSTG2640
C                                                                       HSTG2650
   40 IF(FX) GO TO 50                                                   HSTG2660
      XMIN = X(1)                                                       HSTG2670
      HSCALE = RANGE                                                    HSTG2680
   50 DELTA = HSCALE / FLOAT(NBARS)                                     HSTG2690
      K = 1                                                             HSTG2700
      XLABEL(1) = XMIN                                                  HSTG2710
      DO 80 J=1,NBARS                                                   HSTG2720
      F(J) = 0                                                          HSTG2730
      TOP = XMIN + J * DELTA                                            HSTG2740
      XLABEL(J+1) = TOP                                                 HSTG2750
      IF(K .GT. N) GO TO 80                                             HSTG2755
      DO 60 I=K,N                                                       HSTG2760
      IF(X(I) .GT. TOP) GO TO 70                                        HSTG2770
      F(J) = F(J) + 1                                                   HSTG2780
   60 CONTINUE                                                          HSTG2790
      I=N + 1                                                           HSTG2800
   70 K = I                                                             HSTG2810
   80 CONTINUE                                                          HSTG2820
      IF ( K .LE. N) F(NBARS) = F(NBARS) + N - K + 1                    HSTG2830
C     SCALE FREQUENCIES                                                 HSTG2840
   90 FMAX = F(1)                                                       HSTG2850
      DO 100 I=2,NBARS                                                  HSTG2860
      FMAX = AMAX1(FMAX, FLOAT(F(I)))                                   HSTG2870
  100 CONTINUE                                                          HSTG2880
C                                                                       HSTG2890
C     BLANK PRINT ARRAY                                                 HSTG2900
      DO 110 I=1,1600                                                   HSTG2910
      ARRAY(I) = BLK                                                    HSTG2920
  110 CONTINUE                                                          HSTG2930
C                                                                       HSTG2940
C     PLOT THE BARS                                                     HSTG2950
      LIM = 1536                                                        HSTG2960
      FSCALE = 49. / FMAX                                               HSTG2970
      DO 140 I=1,NBARS                                                  HSTG2980
```

```
      LIM = LIM + 1
      ARRAY(LIM+32) = BMARK
      LINE = 50.5 - FSCALE * F(I)
      IF (LINE .GE. 50) GO TO 130
      LINE = LINE * 32 + I - 32
      DO 120 J = LINE, LIM, 32
      ARRAY(J) = MARK
  120 CONTINUE
      GO TO 140
  130 IF(F(I) .EQ. 0) ARRAY(LIM+32) = NOMARK
  140 CONTINUE
C
C     FIND PROBABILITY MARKERS
      PROBMX = FMAX / AN
      DO 150 I=1,50
      PROB(I) = 0
  150 CONTINUE
      INCR = 5
      IF(PROBMX .LT. .20) INCR = 2
      IF(PROBMX .LT. .10) INCR = 1
      J = INCR
  160 Q = J / 100.0
      IF ( Q .GE. PROBMX) GO TO 170
      LINE = (1. - Q / PROBMX) * 49. + 1.5
      PROB(LINE) = J
      J = J + INCR
      GO TO 160
C
C     FIND THE MEAN
  170 SUM = 0.D0
      DO 175 I=1,N
      SUM = SUM + X(I)
  175 CONTINUE
      XMEAN = SUM/AN
C
C     FIND THE QUANTILE ESTIMATES
      IQ1 = N / 4
      IQ2 = N / 2
      IQ3 = N - IQ1
      M2M = 1 - MOD(N,2)
      M1 = 1 - MOD(N,4)/2
      M12 = 1 + M1
      XQ1 = (M1 * X(IQ1) + X(IQ1 + 1)) / M12
      XQ2 = (M2M * X(IQ2) + X(IQ2+1)) / (1 + M2M)
      XQ3 = (M1 * X(IQ3+1) + X(IQ3)) / M12
      XQ10 = X(N/10)
      XQ90 = X(9*N/10)
C
```

```
C     PLOT THE MEAN AND THE QUANTILES                                   HSTG3470
      R = (MAX-2.0) / HSCALE                                            HSTG3480
      IQ1 = (XQ1 - XMIN) * R + 1.5                                      HSTG3490
      IQ2 = (XQ2 - XMIN) * R + 1.5                                      HSTG3500
      IQ3 = (XQ3 - XMIN) * R + 1.5                                      HSTG3510
      MEAN = (XMEAN - XMIN) * R + 1.5                                   HSTG3520
C-----INSURE MEAN AND QUARTILES PLOT ON GRAPH                           HSTG3530
      IF(MEAN .LT. 1) MEAN = 1                                          HSTG3540
      IF(MEAN .GT. MAX) MEAN = MAX                                      HSTG3550
      IF(IQ1 .LT. 1) IQ1 = 1                                            HSTG3560
      IF(IQ2 .LT. 1) IQ2 = 1                                            HSTG3570
      IF(IQ3 .LT. 1) IQ3 = 1                                            HSTG3580
      IF(IQ1 .GT. MAX) IQ1 = MAX                                        HSTG3590
      IF(IQ2 .GT. MAX) IQ2 = MAX                                        HSTG3600
      IF(IQ3 .GT. MAX) IQ3 = MAX                                        HSTG3610
  179 DO 180 I=1,50                                                     HSTG3620
      PLOT(IQ1,I) = DOT                                                 HSTG3630
      PLOT(IQ2,I) = DOT                                                 HSTG3640
      PLOT(IQ3,I) = DOT                                                 HSTG3650
      PLOT(MEAN,I) = EM                                                 HSTG3660
      PLOT(MAX,I) = VERT                                                HSTG3670
  180 CONTINUE                                                          HSTG3680
C-----BYPASS STATISTIC COMPUTATION, IF DESIRED                          HSTG3690
      IF(.NOT. STAT) GO TO 198                                          HSTG3700
C                                                                       HSTG3710
C     COMPUTE MEAN AND MOMENT ESTIMATES                                 HSTG3720
      MDRNG = 0.5 * (X(1) + X(N))                                       HSTG3730
      SUM  = 0.0D0                                                      HSTG3740
      SUM2 = 0.0D0                                                      HSTG3750
      SUM3 = 0.0D0                                                      HSTG3760
      SUM4 = 0.0D0                                                      HSTG3770
      DO 190 I=1,N                                                      HSTG3780
      SUM = SUM + ABS(X(I) - XQ2)                                       HSTG3790
      DEV = X(I) - XMEAN                                                HSTG3800
      SUM2 = SUM2 + DEV * DEV                                           HSTG3810
      SUM3 = SUM3 + DEV ** 3                                            HSTG3820
      SUM4 = SUM4 + DEV ** 4                                            HSTG3830
  190 CONTINUE                                                          HSTG3840
      VAR = SUM2 / (AN - 1.0)                                           HSTG3850
      STDV = SQRT(VAR)                                                  HSTG3860
      MNDEV = SUM / AN                                                  HSTG3870
      XSUM3 = SNGL(SUM3) * AN /( (AN-1.) * (AN-2.))                     HSTG3880
      SKEW = XSUM3  / STDV ** 3                                         HSTG3890
      BETA1 = SUM3 / AN                                                 HSTG3900
      XSUM4 = SUM4 * ((AN-2.) * AN + 3.) / ((AN-1.) * (AN-2.) * (AN-3.))HSTG3910
      XSUM4 = XSUM4 - VAR*VAR*3.*(AN-1.)*(AN+AN-3.)/(AN*(AN-2.)*(AN-3)) HSTG3920
      CKURT = XSUM4 / (VAR * VAR) - 3.                                  HSTG3930
      BETA2 = SUM4 / AN                                                 HSTG3940
```

```
      CVAR = 0.0                                                        HSTG3950
      IF (ABS(XMEAN) .GT. 1.E-30) CVAR = STDV / ABS(XMEAN)              HSTG3960
C                                                                       HSTG3970
      MIDSPD = XQ3 - XQ1                                                HSTG3980
      TRIMN = 0.25 * (XQ1 + XQ2 + XQ2 + XQ3)                            HSTG3990
      SUM = XQ1 + XQ3                                                   HSTG4000
      IQ1 = N/4 + 2                                                     HSTG4010
      IQ3 = N - IQ1 + 1                                                 HSTG4020
      DO 92 I=IQ1,IQ3                                                   HSTG4030
      SUM = SUM + X(I)                                                  HSTG4040
   92 CONTINUE                                                          HSTG4050
      MIDMN = SUM/(IQ3 - IQ1 + 3)                                       HSTG4060
C                                                                       HSTG4070
C     GET GEOMETRIC AND HARMONIC MEANS FOR POSITIVE RANDOM VARIABLES    HSTG4080
      IF(X(1) .LE. 0.0) GO TO 198                                       HSTG4090
      SUM = 0.0D0                                                       HSTG4100
      SUMH = 0.0D0                                                      HSTG4110
      DO 195 I=1,N                                                      HSTG4120
      SUM = SUM + ALOG(X(I))                                            HSTG4130
      SUMH = SUMH + 1./X(I)                                             HSTG4140
  195 CONTINUE                                                          HSTG4150
      GEOM = SUM / AN                                                   HSTG4160
      GEOM = EXP(GEOM)                                                  HSTG4170
      HARM = AN / SUMH                                                  HSTG4180
C                                                                       HSTG4190
C     WRITE THE FREQUENCIES                                             HSTG4200
  198 WRITE(NOUT,200) N, (F(I), I=1,NBARS)                              HSTG4210
  200 FORMAT('1',5X,'FREQUENCIES',36X,'SAMPLE SIZE =',I5//4X32I4)       HSTG4220
C                                                                       HSTG4230
C     DRAW THE TOP OF THE GRAPH                                         HSTG4240
      WRITE(NOUT,210) BLK, CROSS, (NOMARK,I=1,NBARS)                    HSTG4250
      WRITE(NOUT,210) BLK, VERT, (BLK,I=1,NBM1), RVERT                  HSTG4260
  210 FORMAT(A4,A1,32A4)                                                HSTG4270
C                                                                       HSTG4280
C     PLOT THE GRAPH PART                                               HSTG4290
      DO 240 I=1,50                                                     HSTG4300
      IF(PROB(I) .EQ. 0) WRITE(NOUT,220) (PLOT(J,I), J=1,MAX)           HSTG4310
  220 FORMAT(4X,'|',128A1)                                              HSTG4320
      IF( (PROB(I) .GT. 0) .AND.                                        HSTG4330
     *    (PROB(I) .LT. 10)) WRITE(NOUT,225) PROB(I),(PLOT(J,I),J=1,MAX)HSTG4340
  225 FORMAT(' .0',I1,'|',128A1)                                        HSTG4350
      IF(PROB(I) .GT. 9) WRITE(NOUT,230) PROB(I), (PLOT(J,I),J=1,MAX)   HSTG4360
  230 FORMAT(1X,'.',I2,'|',128A1)                                       HSTG4370
  240 CONTINUE                                                          HSTG4380
C                                                                       HSTG4390
C     WRITE OUT THE SCALE ALONG THE BOTTOM                              HSTG4400
      WRITE(NOUT,250) (RVERT, I=1,NBP1,2)                               HSTG4410
  250 FORMAT(1X,16(A4,4X),A4)                                           HSTG4420
```

```
      XS = AMAX1 (ABS(X(1)), ABS(X(N)) )
      IF ( (XS .LT. .001) .OR. (XS .GE. 1000.) ) GO TO 260
      J = ALOG10(XS) + 2.0
      IF(J .LE. 0) J = 1
      FMT(7) = NUM(J)
      FMT(17) = NUM(J)
      FMT(23) = NUM(J)
      WRITE(NOUT,FMT) (XLABEL(I), I=1,NBP1, 2)
      GO TO 290
  260 WRITE(NOUT,270) (XLABEL(I), I=1, LIM E, 4)
  270 FORMAT(1X1PE9.2,2X,'|',7(3XE9.2,3X,'|'),7X,'|')
      WRITE(NOUT,280) (XLABEL(I), I=3, LIM E, 4)
  280 FORMAT(8X1PE9.2, 7(7X, E9.2))
C
  290 IF (FX) WRITE(NOUT,295) XMIN, XMAX
  295 FORMAT(//'      SCALE FIXED FROM',1PE14.6,' TO',E14.6)
C
C     BYPASS STATISTICS, IF REQUESTED
      IF(.NOT. STAT) GO TO 330
C     WRITE OUT THE STATISTICS
      WRITE(NOUT,300) XMEAN, VAR, XSUM3, X(1), XQ2, STDV, XSUM4, XQ10,
     *             TRIMN, CVAR, SKEW, XQ1, MIDMN, MNDEV, CKURT, XQ2,
     *             MDRNG, RANGE, BETA1, XQ3
  300 FORMAT(///'OCENTRAL TENDENCY',11X, 'SPREAD',21X, 'HIGHER CENTRAL M
     *OMENTS', 4X, 'DISTRIBUTION'/
     *       'OMEAN',5X1PE14.6,4X,'VARIANCE ',E14.6,4X,'M3',6X,E14.6,4X,
     *          'MINIMUM',14X,E14.6/
     *       ' MEDIAN    ',E14.6,4X,'STD DEV   ',E14.6,4X,'M4',6XE14.6,
     *          4X, '.10 QUANTILE',9X,E14.6/
     *       ' TRIMEAN   ',E14.6,4X,'COEF VAR ',E14.6,4X,'SKEWNESS',E14.6
     *          ,4X, '.25 QUANTILE (HINGE) ',E14.6/
     *       ' MIDMEAN   ',E14.6,4X,'MEAN DEV ',E14.6,4X,'KURTOSIS',
     *          E14.6,4X,'.50 QUANTILE (MEDIAN)',E14.6/
     *       ' MIDRANGE ',E14.6,4X,'RANGE     ',E14.6,4X,'BETA1    ',
     *          E14.6,4X,'.75 QUANTILE (HINGE) ',E14.6)
C
      IF(X(1) .GT. 0.0) WRITE(NOUT,310)GEOM, MIDSPD, BETA2, XQ90,
     *                                  HARM, X(N)
  310 FORMAT(' GEOM MEAN',1PE14.6,4X,'MIDSPREAD',E14.6,4X,'BETA2    ',
     *       E14.6,4X,'.90 QUANTILE',9X,E14.6/
     *       ' HARM MEAN',E14.6,57X,'MAXIMUM',14X,E14.6//)
C
      IF(X(1) .LE. 0.0) WRITE(NOUT,320)MIDSPD, BETA2, XQ90, X(N)
  320 FORMAT(28X,'MIDSPREAD',1PE14.6,4X,'BETA2    ',E14.6,4X,
     *          '.90 QUANTILE',9X,E14.6/
     *       81X,'MAXIMUM',14X,E14.6//)
C
  330 CONTINUE
```

```
      IF(.NOT. FUNC) GO TO 450                                          HSTG4910
      FUNC = .FALSE.                                                    HSTG4920
C                                                                       HSTG4930
C     FIND AND PLOT FN (EMPIRICAL DENSITY FUNCTION)                     HSTG4940
C-----BANDWIDTH                                                         HSTG4950
      BN = SQRT(AN) / RANGE                                             HSTG4960
      XN = BN / AN                                                      HSTG4970
C-----EMPIRICAL DENSITY FUNCTION CALCULATION                            HSTG4980
      LOW = 1                                                           HSTG4990
      FMAX = -10.0                                                      HSTG5000
      XINC = HSCALE /MAX                                                HSTG5010
      MAXM1 = MAX - 1                                                   HSTG5020
      DO 430 I=1,MAXM1                                                  HSTG5030
      U = XMIN + I * XINC                                               HSTG5040
      FN(I) = 0.0                                                       HSTG5050
      DO 410 J=LOW,N                                                    HSTG5060
      TRY = (U - X(J) ) * BN                                            HSTG5070
      IF(TRY .GT.  1.0) GO TO 400                                       HSTG5080
      IF(TRY .LT. -1.0) GO TO 420                                       HSTG5090
C-----WEIGHT FUNCTION                                                   HSTG5100
      FN(I) = FN(I) + 1.0 - ABS(TRY)                                    HSTG5110
      GO TO 410                                                         HSTG5120
  400 LOW = J                                                           HSTG5130
  410 CONTINUE                                                          HSTG5140
  420 FN(I) = FN(I) * XN                                                HSTG5150
      FMAX = AMAX1(FMAX, FN(I))                                         HSTG5160
  430 CONTINUE                                                          HSTG5170
C-----PLOT THE DENSITY FUNCTION                                         HSTG5180
      DO 440 I=1,MAXM1                                                  HSTG5190
      LINE = 50.5 - 49.0 * FN(I) / FMAX                                 HSTG5200
      PLOT(I, LINE) = EFF                                               HSTG5210
  440 CONTINUE                                                          HSTG5220
C-----PRINT THE REVISED HISTOGRAM                                       HSTG5230
      GO TO 198                                                         HSTG5240
C                                                                       HSTG5250
C     RESET FUNCTION FLAG AND QUIT                                      HSTG5260
  450 FUNC = .TRUE.                                                     HSTG5270
      RETURN                                                            HSTG5280
C                                                                       HSTG5290
      ENTRY NOSTAT                                                      HSTG5300
C                                                                       HSTG5310
C     ENTRY POINT TO SET STATISTIC FLAG TO SUPPRESS                     HSTG5320
C     CALCULATION AND PRINTOUT OF STATISTICS                            HSTG5330
C                                                                       HSTG5340
      STAT = .FALSE.                                                    HSTG5350
      RETURN                                                            HSTG5360
C                                                                       HSTG5370
      ENTRY FIX(SCALE)                                                  HSTG5380
```

```
C
      FX = .TRUE.
      XMIN = AMIN1(SCALE(1), SCALE(2))
      XMAX = AMAX1(SCALE(1), SCALE(2))
      HSCALE = XMAX - XMIN
C-----TEST FOR VALIDITY
      IF (HSCALE .LE. 0.0) FX = .FALSE.
      RETURN
C
      ENTRY NOFIX
      FX = .FALSE.
      RETURN
C
      END


      SUBROUTINE DISPL
C
C         SUBROUTINE DISPL PROVIDES A GEORGRAPHICAL DISPLAY OF THE
C         BLUE SURFACE UNITS.  THE MAXIMUM RANGE SCALE IS 200 NM FROM
C         FORCE CENTER.  A 100 NM CIRCLE IS DRAWN ON THE PLOT.
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLIST(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      DIMENSION XSHIP(40),YSHIP(40),CSHIP(40),XXSHIP(40)
      DIMENSION CX(1257),CY(1257)
      DO 1 I = 1,NS
      XSHIP(I) = 4.0 + 0.0200*SHIP(I,2)
      YSHIP(I) = 4.0 + 0.0200*SHIP(I,3)
    1 CONTINUE
      CALL PLOTS
      CALL AXIS(4.0,0.0,' ',1,8.0,90.0,-200.0,50.0)
      CALL AXIS(0.0,4.0,' ',1,8.0,00.0,-200.0,50.0)
      CALL LINE(XSHIP,YSHIP,NS,1,-4)
      DO 2 I = 1,NS
      XXSHIP(I) = XSHIP(I) + 0.1
      CSHIP(I) = I
      CALL NUMBER(XXSHIP(I),YSHIP(I),0.14,CSHIP(I),0.0,-1)
    2 CONTINUE
      DO 33 N = 1,1257
      T = FLOAT(N)/100.
      CX(N) = 4.0 + 2.00*SIN(T)
      CY(N) = 4.0 + 2.00*COS(T)
```

```
   33 CONTINUE
      CALL LINE(CX,CY,1257,1,7)
      CALL SYMBOL(0.0,8.0,0.14,'SHIP POSITIONS',0.0,14)
      CALL SYMBOL(0.0,8.5,0.14,'FLT AIR DEF SIM',0.0,15)
      CALL SYMBOL(3.7,8.25,0.14,'NORTH',0.0,5)
      CALL PLOT(0.0,12.0,-3)
      CALL PLOTE
      RETURN
      END


      SUBROUTINE RAIDIS
C
C        SUBROUTINE RADIS PROVIDES A GEOGRAPHICAL DISPLAY OF RAIDER
C         TRACKS FROM THEIR INITIAL POSITION (IP) TO LAUNCH POSITION.
C         IN THE CASE WHERE A RAIDER DOES NOT HAVE A LAUNCH POSITION,
C         THE 30 MIN GAME TIME POSITION IS DISPLAYED.  IF THE LAUNCH
C         OR 30 MIN POSITION ARE MORE THAN 800 NM FROM THE CENTER,
C         NO TRACK IS DISPLAYED AT THE LOCATION OF THE RAIDER'S IP.
C         800 NM IS THE MAXIMUM DISTANCE FOR THE PLOT
C
      COMMON X(2744),ASMT(100,8),SAMASM(300,7),RAID(90,17),SHIP(40,6),SM
     1AT(300,10),IVENTS(1000,2),STATS(40,2),IBAT(40,4),NONO(190,40),ISEE
     2D,MAMBAT,KOUNT,NADAR(40,19),RXATA(5),LOVER,ITEMS,PLISI(40,6),REASO
     3N,CHANEL(40,4,3,6),NASM,JAM(4),TMAX,NRTGT(90),MISLFT(90,2),NR,NS,Q
     4RT(50),IFLASK,IGAGE(190),ISAG(40),IFAG,ISBG(40),IFBG,RELOAD(40,4),
     5MARY(100),CHECK,IFIRST,NPGS,KMART(4),NSAM,TGTAV(190),TCF(6),IHIT(4
     60),AMF,HISX(100),JACK(40,7),HISA(100)
      DIMENSION XRAID(2),YRAID(2)
      CALL PLOTS
      CALL AXIS(0.0,4.0,' ',1,8.0,00.0,-800.0,200.0)
      CALL AXIS(4.0,0.0,' ',1,8.0,90.0,-800.0,200.0)
      DO 30 I=1,NR
      LFLAG = 0
      CRAID=I
      XRAID(1) = 4.0 + 0.00500*RAID(I,3)
      YRAID(1) = 4.0 + 0.00500*RAID(I,4)
      IF(ABS(RAID(I,7)).GT.800.0.OR.ABS(RAID(I,8)).GT.800.0) GO TO 40
        XRAID(2) = 4.0 + 0.00500*RAID(I,7)
        YRAID(2) = 4.0 + 0.00500*RAID(I,8)
   20     CONTINUE
      CALL LINE(XRAID,YRAID,2,1,3)
      XXRAID = XRAID(1) + 0.05
      CALL NUMBER(XXRAID,YRAID(1),0.07,CRAID,0.0,-1)
      XXRAID = XXRAID   + 0.15
      CALL SYMBOL(XXRAID,YRAID(1),0.07,'IP',0.0,2)
      XTRAID = XRAID(2) + 0.05
      CALL NUMBER (XTRAID,YRAID(2),0.07,CRAID,0.0,-1)
```

```
      XTRAID = XTRAID + 0.15
      IF (LFLAG.EQ.1) GO TO 70
      CALL SYMBOL(XTRAID,YRAID(2),0.07,'LAUNCH',0.0,6)
   25    CONTINUE
      CALL WHERE(XNOW,YNOW)
      CALL PLOT(XNOW,YNOW,3)
      GO TO 30
   40    LFLAG = 1
      CALL LOCTIM (I,30.,1,XRAID(2),YRAID(2))
      IF (ABS(XRAID(2)).GT.800.0.OR.ABS(YRAID(2)).GT.800.0) GO TO 60
      XRAID(2) = 4.0 + 0.00500*XRAID(2)
      YRAID(2) = 4.0 + 0.00500*YRAID(2)
         GO TO 20
   60    CONTINUE
      CALL SYMBOL (XTRAID,YRAID(1),0.07,CRAID,0.0,-1)
      XTRAID = XRAID(1) + 0.05
      CALL SYMBOL (XTRAID,YRAID(1),0.07,'NO TRACK',0.0,8)
         GO TO 25
   70    CONTINUE
      CALL SYMBOL (XTRAID,YRAID(2),0.07,'30 MIN',0.0,6)
         GO TO 25
   30 CONTINUE
      CALL SYMBOL(0.5,8.0,0.14,'RAIDER TRACKS',0.0,13)
      CALL SYMBOL(0.5,8.5,0.14,'FLT AIR DEF SIM',0.0,15)
      CALL SYMBOL(3.7,8.25,0.14,'NORTH',0.0,5)
      CALL PLOT(0.0,12.0,-3)
      CALL PLOTE
      RETURN
      END

SETIMX   START
         SAVE   (14,12)
         LR     12,15
         USING SETIMX,12
         ST     13,TEMP
         LA     13,SAVE
         STIMER TASK,FIXUP,TUINTVL=TIME
         L      13,TEMP
         RETURN (14,12),T
         DROP   12
FIXUP    SAVE   (14,12)
         RETURN (14,12)
         ENTRY GETIMX
GETIMX   SAVE   (14,12)
         LR     12,15
         USING GETIMX,12
         L      2,0(1)
```

```
         ST     13,TEMP
         LA     13,SAVE
         TTIMER
         L      1,TIME
         SR     1,0
         ST     1,0(2)
         L      13,TEMP
         RETURN (14,12),T
SAVE     DS      18F
TEMP     DS      F
TIME     DC      X'7FFFFFFF'
         END

      FUNCTION RADIAN (RD)
C
C          CONVERTS FROM DEGREES TO RADIANS.
C
      IMPLICIT REAL*8(R)
      RADIAN = RD*0.017453293
      RETURN
      END


//LINK.SYSPRINT DD SYSOUT=0
//LINK.SYSIN DD *
//GO.FT06F001 DD SPACE=(CYL,(4,2)),SYSOUT=0
//GO.SYSIN DD *
C                    JOB COMPLETE
```

BIBLIOGRAPHY

1. Caspars, J.W., Bistatic and Multistatic Radar: Principles and Applications, NELC Report 1624, 15 Apr. 1969.

2. Emshoff, James R. and Sisson, Roger L., Design and Use of Computer Simulation Models, Macmillan Co., New York, 1970.

3. Farrell, J.W., MFS-E-238, FLOATS IIB Input Specifications, Applied Physics Laboratory, Johns Hopkins University memo dated 23 April 1973.

4. Farrell, J.W., MFS-E-239, FLOATS IIB Output Formats, Applied Physics Laboratory, Johns Hopkins University memo dated 23 April 1973.

5. Farrell, J.W., MFS-E-240, Description of FLOATS IIB Subroutines, Applied Physics Laboratory, Johns Hopkins University memo dated 15 May 1973.

6. Fox, C.M., MSF-E-244, A functional Description of the FLOATS Model, Applied Physics Laboratory, Johns Hopkins University memo dated 31 May 1973.

7. Learmonth, G.P. and Lewis, P.A.W., Naval Postgraduate School Random Number Generator Package LLRandom, NPS Research Report NPS 55LW73061A, June 1973.

8. Meyer, C.F. and Radcliffe, A., A Simple Practical Technique for Estimating Radar Ranges in Clear and ECM Environments, CLA Division, Applied Physics Laboratory, Johns Hopkins University, CLA Internal Memo 885 of 1 December 1960.

9. Deleted.

10. Sholnik, Merrill I., Introduction to Radar Systems, McGraw-Hill Book Company, New York, 1962.

11. Waddell, M.C., Air Battle Analyzer Handbook, TG-421, Applied Physics Laboratory, Johns Hopkins University, December 1961.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Chief of Naval Personnel ATTN: Pers 11B Department of the Navy Washington, D.C. 20370	1
2.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
3.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
4.	Professor A. F. Andrus, Code 55As Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	5
5.	Professor D. E. Harrison, Jr., Code 61Hx Department of Physics and Chemistry Naval Postgraduate School Monterey, California 93940	5
6.	CDR D. L. Cover, USN, Code AA NPRO P. O. Box 2505 Pomona, California 91766	2
7.	ENS G. G. Brown, USN, Code 55Zr Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
8.	Earl B. Gardner Manager, Operations Research General Dynamics, Pomona Division 1675 West Mission Blvd. Pomona, California 91766	2
9.	Professor William Reese, Code 61Rc Department of Physics and Chemistry Naval Postgraduate School Monterey, California 93940	1

		No. Copies
10.	Library University of Notre Dame Notre Dame, Indiana 46556	1
11.	Johns Hopkins University Applied Physics Laboratory ATTN: William Sweet 8621 Georgia Ave. Silver Spring, Maryland 20910	1
12.	Johns Hopkins University Applied Physics Laboratory ATTN: Richard Hunt 8621 Georgia Ave. Silver Spring, Maryland 20910	1
13.	LCDR Charles R. Schlegelmilch, USN USS Leonard F. Mason (DD-852) FPO San Francisco 96601	1
14.	LCDR Richard J. Leonhardt, USN USS Guadalcanal (LPH-7) FPO New York 09501	2
15.	Department Chairman, Code 55 Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	2
16.	ADM W. E. Meyer, USN PMO 403 Naval Sea Systems Command Hdqtrs Department of the Navy Washington, D.C. 20360	1

Thesis 160078
L542 Leonhardt
c.1 A computer simulation for the evaluation of fleet air defense employing monostatic and bistatic search radars.