30 April 1971

AD

TECHNICAL REPORT T.R.9-71

AD734885

ACN 16782

SURVEILLANCE, TARGET ACQUISITION AND NIGHT OBSERVATION (STANO) PHASE I SYSTEM ASSESSMENT MODEL (SAM)

(Short Title: STANO PHASE I SAM)

FINAL REPORT

VOLUME I MODEL DESCRIPTION

PART I

UNITED STATES ARMY COMBAT DEVELOPMENTS COMMAND

SYSTEMS ANALYSIS GROUP

466

30 April 1971

AD

ACN 16782

SURVEILLANCE, TARGET ACQUISITION AND NIGHT OBSERVATION (STANO) PHASE I SYSTEM ASSESSMENT MODEL (SAM)

(Short Title: STANO PHASE I SAM)

FINAL REPORT

VOLUME I-MODEL DESCRIPTION

PART I

PREPARED FOR:

UNITED STATES ARMY
COMBAT DEVELOPMENTS COMMAND

SYSTEMS ANALYSIS GROUP

CONTRACT NO. DAAB07-69-C-0069
AMENDMENT P00002

CORNELL AERONAUTICAL LABORATORY, INC.
BUFFALO, NEW YORK 14221
CAL REPORT NO. UM-2709-H-6

NOTICES

DISPOSITION

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

CAVEAT

USACDC Institute of Systems Analysis (USACDCISA) was redesignated USACDC Systems Analysis Group (USACDCSAG) by General Order Number 184, effective 15 April 1971. Reference to the Institute of Systems Analysis (ISA) throughout this report should be Systems Analysis Group (SAG).

DISCLAIMER

The findings in this report are not be be construed as an official Department of the Army position, unless so designated by other authorized documents.

ACKNOWLEDGMENTS

The size and complexity of the technical effort conducted under the CAL Project SAM I as recorded in this document required the technical skills of a diverse group of CAL staff members. The successful completion of the Systems Assessment Model reflects the technical excellence, the high workmanship standards and the personal dedication of the Project SAM I team.

Major technical achievements and program guidance were provided by the following:

Adler, Paul A.	Kinzly, Robert E.
Black, Harold	Schneeberger, Richard S.
Fleck, John T.	Snelting, John B.
Fryer, William D.	Taylor, Richard J.
Garrett, William O.	Wojcinski, Thaddeus J.

Significant technical contributions and support efforts were made by the following:

Campbell, James B.	Lehmann, Robert L.
Capps, Joseph R.	Mack, Robert J.
Ellis, Neil L.	Magorian, Thomas R.
Ferris, Robert C.	Marione, Simon A.
Gavin, William J.	Persico, John P.
Goehring, Oran	Ryan, Charles T. Jr.
Hammill, Harry	Ryll, Ewald
Hayman, Robert A.	

CAL gratefully acknowledges the technical contributions of the above named project team members.

Lastly, receipt of excellent technical guidance, cooperative liaison and prompt fulfillment of scenario and data requirements by the following CDC/ISA representatives is gratefully acknowledged.

Mr. Raymond V. Attarian - CDC/ISA
Maj. Claude B. Donovan - CDC/ISA
Mr. Herbert H. Gibson - CDC/ISA
Maj. Richard J. Girouard - CDC/ISS
Lt. William R. Lane - CDC/ISA
Mr. Paul T. Long - CDC/ISA
Mr. Keith Thorp - CDC/ISA
Mr. John M. Tucker - CDC/ISA

SECURITY CHECKLIST

1. TITLE OF STUDY. Surveillance, Target Acquisition and Night Observation (STANO) Phase I System Assessment Model (SAM) (Short Title: STANO Phase I SAM)

2. This study does not contain NOFORN or non-CDC information.

3. No limitations on dissemination have been imposed.

4. The overall classification of this document is UNCLASSIFIED.

FOREWORD

This final report describes the work conducted by Cornell Aeronautical Laboratory (CAL) for the Institute of Systems Analysis of the United States Army Combat Developments Command under Amendment P00004 to US Army Electronics Command Contract DAAB07-69-C-0069. Earlier efforts under this contract, performed in response to technical direction of the Systems Technical Area, Combat Surveillance, Target Acquisition and Night Observation Laboratory ECOM, were concerned with the definition of STANO* system concepts and the analytical evaluation of system performance. The results of these efforts were reported in System Integration and Testing (U), Quarterly Progress Reports Nos. 1, 2, 3, 4, and 5, R&D Technical Report Nos. ECOM-0069-1, -2, -3, -4, -5, April 1969, June 1969, October 1969, November 1969 and May 1970. The technical efforts performed under the amended contracts and reported herein were directed toward the synthesis of an assessment methodology and the design and development of a computerized simulation for use in generating information about the performance of STANO systems. The digital computer simulation model will hereinafter be referred to as the Phase I Systems Assessment Model (SAM I).

The results of this present technical effort are presented in three volumes, as follows:

Vol I - Model Description (this volume)
Vol II - Users' Manual
Vol III - Program Listings and Flow Diagrams

The Model Description (Vol I) includes a detailed discussion of all major model components and is presented in non-machine-oriented language. Program subroutines describing system components and parameters, environmental aspects and the analysis of derived data are presented in terms of the physical processes being simulated and the mathematical and logical processes that have been employed. Detailed discussions are also given delineating the methodologies and techniques employed for computer implementation of the model. Overall model structure, program flow and computer control logic are provided in depth.

The Users' Manual (Vol II) presents the requisite data required for exercising the model. In addition to providing operating and

*STANO is the acronym for Surveillance, Target Acquisition, and Night Observation.

machine processing instructions it contains detailed instructions on the format and preparation of planners' input. Also included are data on designer input tables and block data incorporated in the model design.

Program Listings and Flow diagrams (Vol III) contains listings and flow diagrams for all subprograms of the overall model. The flow diagrams were processed through a proprietary AUTOFLOW program leased by CAL from Applied Data Research, Inc. Unlike Volumes I and II, this volume consists of unbound computer printout, two copies of which have been submitted to USACDCISA.

ACN 16782

SURVEILLANCE, TARGET ACQUISITION
AND NIGHT ACQUISITION (STANO)
PHASE I SYSTEM ASSESSMENT MODEL (SAM)
STANO PHASE I SAM

VOLUME INDEX

CONTENTS

CONTENTS (Con't)

CONTENTS (Con't)

VOLUME I, PART II, MODEL DESCRIPTION

CONTENTS

CONTENTS (Cont)

CONTENTS

CONTENTS (Cont)

ABSTRACT

This report was prepared by Cornell Aeronautical Laboratory, Inc., for US Army Combat Developments Command, Systems Analysis Group, under Contract DAAB07-69-C-0069, Amendment P00002. This model was developed to support the STANO Program.

Abstract: The STANO Phase I SAM is designed to simulate a brigade or smaller STANO System in a low-intensity conflict. The model will permit the establishment and evaluation of numerous effectiveness criteria for individual STANO sensors and subsystems. It will facilitate the formation of improved candidate STANO Systems, through better understanding of shortcomings in organization, materiel and concepts of employment. It has the capability of producing information permitting scientifically supportable evaluations and judgments of interface requirements and trade-off options of STANO subsystems. The model can be used for parametric analysis, trade-off analysis, and system performance sensitivity tests.

The two volumes of the report are as follows:

Volume I - MODEL DESCRIPTION
PART I

MODEL DESCRIPTION
PART II

Volume II - USERS' MANUAL

Section 1
SUMMARY

1.1 GENERAL

The Phase I System's Assessment Model, described in detail in the subsequent sections of this report, is designed to provide a comprehensive and realistic digital simulation of STANO systems in a real world environment. The simulation utilizes operational type inputs concerning men and equipment derived from postulated battlefield situations to exercise the sensor performance routine and determines the outcomes of sensor versus target encounters. These encounters are conducted under realistic conditions with respect to sensor reliability, emplacement, terrain, foliage, wind, temperature, rain, day, night, battle activities (artillery, vehicles, aircraft), local cultural aspects (indigenous personnel, animals) to examine the total system performance. Thus, SAM I simulates STANO systems within the context of their operating environment and the variety of interactions which exist between and among these elements.

1.2 MODEL OBJECTIVES

The general objectives pursued in developing the SAM I were to provide by use of digital simulation a means of generating information concerning STANO systems performance that would aid in:

a. evaluating candidate STANO systems

b. reducing the number of candidate systems to be tested

c. reducing number and type of field tests to be conducted

d. determining where improvement in system performance is warranted

e. establishing system requirements.

Specifically, the objective was to design a computerized model to simulate the performance of a brigade STANO system in a low intensity SEA environment. Significant features required to be incorporated in the model design were as follows:

a. Capability to accept detailed inputs in terms of what comprises the candidate system. Also be able to operate with less detailed data.

b. Permit economical expansion.

c. Model performance of STANO systems in terms of range, resolution, system accuracy, sensor logic, line of sight requirements, false alarms, reliability, life expectancy and failure rates.

d. Model processing of STANO derived data insofar as practicable.

e. Reflect organizational concepts and methods of employment.

f. Model movement of men and equipment with some measure of tactical realism.

g. Model sources of false alarms incidental to operation of combat forces as realistically as possible.

h. Provide model outputs suitable for use in quantitative assessment of candidate STANO systems.

1.3 BACKGROUND

The technical effort leading to the development of the Phase I System Assessment Model described herein evolved from technical analyses conducted by CAL's Project MASS* under the technical guidance and sponsorship of the United States Army Electronics Command. The initial work comprised analysis of a variety of problems associated with utilizing electronic data collection devices in a battlefield environment.** Results of these analyses and particularly studies of remote unattended sensors, demonstrated the need for a digital computer simulation capable of adequately evaluating the performance of the various STANO sensor systems. The diversity of potential system configurations and the number of parameters to be considered under differing operational envrionments, as well as the complexities of the interactions, made it all but impossible to assess system capabilities without a computer supported simulation. Accordingly, during the later phases of the systems studies under ECOM auspices, emphasis shifted to the broader problem of developing a computer model for simulating the performance of electronic sensor systems. The initial model activity was reported in the fourth and fifth quarterly reports,** covering the period July-December 1969.

---

*MASS is an acronym for Mobile Army Surveillance System.

**Refer to System Integration and Testing (U), Quarterly Progress Report Nos. 1, 2, 3, 4, and 5, R&D Technical Report Nos. ECOM-0069-1, -2, -3, -4, and -5, Cornell Aeronautical Laboratory, Inc. April 1969, June 1969, October 1969, November 1969, and May 1970 CONFIDENTIAL.

While the model development was progressing under U. S. Army Electronics Command monitorship, there was evolving at the same time within the U. S. Army STANO project a requirement for a STANO Systems Assessment Model for use in various portions of that project. The U. S. Army Institute for Systems Analysis (ISA) of the Combat Developments Command, after reviewing CAL's Mobile Army Surveillance System (MASS) model concept determined that this concept, when implemented, would most closely satisfy the STANO Systems Assessment Model requirements. Accordingly the Electronics Command contract with CAL was modified (Amendment P0002) to provide a new work statement and CDC/ISA assumed technical direction of the contract.

1.4 OVERALL MODEL STRUCTURE

1.4.1 The Simulation Problem

The multitude of physical processes and operational activities embodied in the installation and operation of STANO systems requires the comprenhensive simulation of numerous factors. This section presents a broad overview of the many significant factors which were considered in the development of the SAM I. Important features of military activities and critical system parameters requisite to the realistic simulation of STANO systems are delineated in order to provide a situational background for subsequent detailed discussions about model components.

In developing the SAM I concepts it was assumed that the battlefield environment consists of a U. S. Army brigade area of operations in a low intensity SEA environment. Friendly and enemy units are assumed to be based in and move about the area of operations in the process of meeting assigned military objectives. A 30 km x 30 km area of operations was selected as sufficient to adequately circumscribe the brigade's activities. Within this area, friendly forces are assumed to carry out various military functions including the installation and operation of a variety of STANO systems to monitor or detect enemy activities. These military activities also include the establishment of base camps and fire bases requiring defensive surveillance activities, and reconnaissance sweeps using all types of sensors. Both manned and unmanned ground based sensors and manned airborne sensors are assumed to be employed by all echelons in accordance with the appropriate employment doctrines and the availability of equipment. Artillery ambushes or fire traps are assumed established to provide target information for artillary fire missions. Friendly and enemy forces, including men and vehicles are assumed to move throughout the area. Non-combatants are also assumed to be present in the area of operations.

Since the basic objectives in the design of SAM I was to provide a simulation model for assessing the performance of STANO systems it was necessary to consider not only the sensor system technical parameters and the potentical targets (friendly and enemy men and vehicles) but also the environmental factors that degrade system performance.

Line-of-sight considerations significantly affect the performance of both data links and many specific sensor types. Accordingly, terrain and foliage features must be simulated. Atmospheric environment also has considerable impact on sensor performance, therefore such factors as wind, rain, cloud cover, fog, daylight, moonlight, and temperature must be accounted for.

The performance of imaging type sensors is sensitive to ambient light conditions ; therefore, the presence of outside sources of illumination must be included in the simulation. In addition, seismic and acoustic sensors are affected by battlefield activities such as artillery firing, mines, vehicles and aircraft;hence these events must be included to provide simulated "real life" conditions. Analogous to the battlefield activities are the sensor responses occasioned by the presence in the area of operations of non-combatants and their associated activities such as personnel and vehicular movements, aircraft, and domestic animals.

Seismic, accoustic and other type of sensors are also subject to false alarms caused by natural phenomena such as lightening, thunder, earthquakes as well as internal electronic circuit noise and means must be provided to account for the effects of these phenomena.

The many sensors available for employment in the brigade STANO system also permit the use of alternative emplacement means, resulting in the introduction of different values of sensor location errors and in variances in the commanders' knowledge of the actual location. Hence, for purposes of assessing sensor performance (i.e. actual target detection opportunities versus potential opportunities) the true sensor location as well as the desired or commanders' location estimates must be determined.

Equipment reliability, maintenance down periods and battery life for battery-power equipment materially affect the performance of STANO systems and each of these factors must be incorporated to achieve a useful simulation model.

Lastly, in the final analysis the worth of any STANO system must be assessed in terms of the intelligence that can be derived from the sensor detection or non-detection reports. Accordingly, simulation of the processing of the sensor responses in terms of the nature of the targets, their locations, direction and rates of movements and the assessment of whether the responses are true targets or false alarms must be made. Finally, the transmission of the processed data to appropriate command levels completes the STANO system simulation and ensures a complete and practical systems assessment model.

1.4.2 Model Features

The SAM I described in this report is a large scale digital simulation model that incorporates a number of features designed to provide wide model utility and to provide a wide range of options to the model user. While primarily structured around the sensor systems the simulation includes submodels of the terrain and foliage; atmospheric, battle field and cultural environments; communication flow processes and the interaction of all these elements. This results in a large number of factors being modelled. Specific features incorporated into the model design that materially enhance the model usefulness are tabulated below:

(1) Selected levels of simulation are available in many portions of the model. Thus the planner may input quite specific weather conditions or use the model's capability to generate a complete representative weather profile for the scenario area and duration of game. Similarly, sensor line-of-sight checks, where appropriate, can be bypassed or included. Most other factors considered can be similarly controlled bv planner control of scenario inputs.

(2) The model has been designed for ease of economical expansion in the future. A modular concept is employed such that the model is operated in a series of sequential but not continuous computer time operations. This permits model expansion capability in terms of required computer storage while limiting computer running time for any particular job step to reasonable quantities. Data sets are constructed to permit easy expansion to larger quantities if computer storage is available. Additional sensor system types can be added with minimum difficulty.

(3) Since the model is designed at present to accept problems at the Brigade level, with a nominal capacity of 100 enemy targets and 400 sensors plus supporting equipments, planner inputs can be sizable. On the other hand, wherever possible, internal data sets called Model Designer Inputs are provided to assist in reducing the planner's problems where practicable.

(4) By the use of auxiliary output programs, the planner may select the level of information output desired. That is, if sensor detection performance is all that is required, the user can stop at that level of model design output. If sensor report analysis is desired, this may be further investigated by using the auxiliary output information processing programs.

(5) Related to (2) above, computer running time was recognized early as becoming sizable if not carefully controlled. The basic approach to minimizing running time was to design

model steps of prerequisite functions so that the functions must be satisfied before the next step is called, and before calling the most complicated and time consuming operations, such as exercising the sensor performance detection subroutines.

(6) A final feature of the model which assists in parametric analysis is the retention of the outcome of the many decisions made during a game. Thus the planner may select one (or a few) factors for parametric analysis and arrange that the decision outcomes for other factors be retained fixed for repeated runs. This allows for the feature under investigation to be incremented over a range of interest without the confusion of varying other factors.

1.4.3 Model Overall Design

The Phase I Systems Assessment Model is a time-stepped simulation. That is, the model first accepts planner inputs and prepares all necessary data holding time at t = 0. The model then proceeds to generate a series of events associated with the sensor systems, potential target detections, false alarms, etc. These events are then played in sequence through the sensor systems with the outcomes becoming the primary output. The simulation is very detailed in many of the effects considered and because of its resultant size and complexity has been divided into a series of submodels. The overall structure of these submodels in relation to the total Phase I SAM is shown in Figure 1.4-1.

As can be seen from Figure 1.4-1, inputs to the model come from digitized topological and terrain data tapes which are then "thinned" and used as inputs to the PRERUN program and as additional inputs to the Radar Contour submodel and the Data Link submodel. Atmospheric conditions are also entered as needed and the game planner inputs all other desired scenario conditions. PRERUN and the Main Simulation Model (MSM) are the major operating programs which provide the basic output for further analytical processing. The main design goal was to provide the user with complete flexibility in selecting and specifying inputs and in exercising post-MSM processing or in choosing any combination of the many model options.

Nine separate submodels make up the total model as follows:

<u>Auxiliary Input Programs</u>

Atmospheric

Make Sparse Terrain Tape

Radar Contour Plots

RF Data Link Analysis

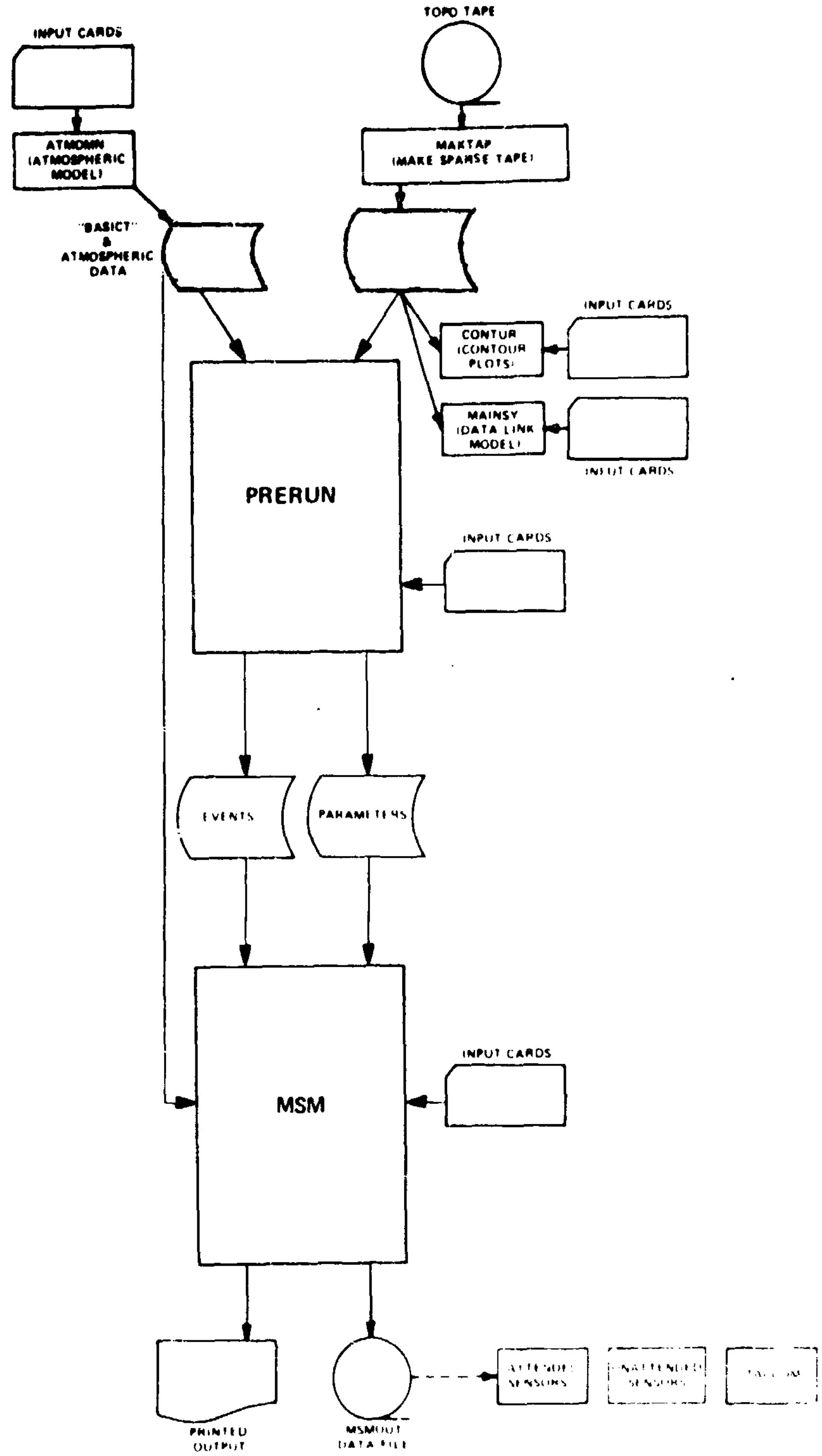


Figure 1.4-1 PHASE I SAM MODEL AND SUBMODEL RELATIONSHIPS

Primary Programs

PRERUN

Main Simulation (MSM)

Information Processing Programs

Unattended Sensor Analysis

Attended Sensor Analysis

Tactical Communications

The following summarized description of each submodel provides an understanding of how the overall model operates to achieve the previously cited task objectives.

As an initial auxiliary input step, the Atmospheric model is designed to generate the atmospheric environment present in the scenario area for the duration of the game time. Its output is used by both the PRERUN and MSM and consists of some 18 subroutines. Using optional levels of planner inputs, this program generates 16 meteorological values necessary for use in later parts of the simulation. These values are produced and stored as tables. (The values change with time and as significant changes occur in the parameters.) As currently supplied, the model is capable of supplying the atmospheric environment in South Vietnam for any game up to six weeks duration based on actual weather statistics supplied by the U.S. Air Force Environmental Technical Applications Center, Washington, D.C. This model is discussed in detail in Section 5.2 of this Volume.

The Make Sparse Terrain Tape model is used as an auxiliary input submodel to produce a digital terrain tape of up to a 30 x 30 km scenario area based on ISA modified digital topographic tapes supplied by the U.S. Army Topographic Command. Two subroutines make up this submodel and its output (data points on tape spaced approximately 100m apart) is used as an input to PRERUN and the other two auxiliary input models, Radar Contour Plot and RF Data Link Analysis. A closely related program of the model provides for the development of input unit terrain descriptor tables and an input terrain index/x, y coordinate table which specifies a terrain type for each 500 square meters of the scenario area. These terrain indices and their associated nineteen terrain feature parameters of interest (foliage, soil type, etc.) are used as required inputs for both primary models (PRERUN and MSM) and in the RF Data Link Analysis and Radar Contour plot submodels. These terrain feature tables are not computer generated but are planner prepared based on a grid overlay of scenario area. Both the terrain tape submodel and terrain parameter development process are discussed in more detail in Section 5.3 of this Volume.

The Radar Contour Plot model is an auxiliary input submodel, which can be used to check coverage of selected radars on the scenario. The 20 subroutines making up this program provide plotted contours of radar coverage, taking into account foliage and terrain masking in the area surrounding the radar site. This submodel is discussed in more detail in Section 6.2 of this Volume. The use of this submodel is optional.

The RF Data Link Analysis is similarly an optional auxiliary input model, operated independently of the main simulation flow. Using planner inputs on sensor transceivers, monitors, and relay locations and equipment characteristics together with computed terrain and foliage path data, this model computes the prospective transmission losses of each link and the likelihood of communication success. This model's 22 subroutines operate in two steps--first extracting the terrain and foliage data for the RF path between terminals and then computing RF losses anticipated over the path. It is designed as an aid in checking RF data link communications efficiency. This submodel is discussed in detail in Section 6.3 of this Volume.

Referring to Fig. 1.4-1 again, the next step in model processing involves the use of atmospheric model outputs, the digital terrain tape, and other planner inputs to carry out the series of job steps, titled PRERUN. The PRERUN portion of the model was designed as a means of reducing overall computer storage requirements and for conserving continuous computer running time. By processing all the interactions that are not required to be run in actual game time sequence, it was possible to derive an ordered set of significant events which could then be processed in the Main Simulation. This processing was further broken down into a series of 15 sequential job steps covering the functions shown in Tab. 1.4-I. Some 96 subroutines and 20 internal data sets are included in PRERUN. This job step breakdown has alleviated the need for large blocks of storage and allows the program to be run without requiring a large single block of computer time. The steps have been so designed that further sub-division is easily done if it is desired to reduce the storage requirements and/or to process a much larger scenario than the original design specified. Thus, although this version was run on an IBM 360/65, it is easily adaptable to other machines.

Table 1.4-I

PRERUN JOB STEPS

Step	Description
0	Sets game variables - Reads and converts planner input.
1	Computes up/down times of monitors, firetraps, relays, data links and unattended ground sensor arrays.
2	Computes up/down times of sensors in unattended ground sensor arrays and stationary scanning arrays.

Table 1.4-I (Cont.)

PRERUN JOB STEPS

Step	Description
3	Computes ground truth positions for sensors in unattended ground sensor arrays and stationary scanning arrays.
4	Computes ground truth positions and up/down times for moving sensor arrays.
5	Prepares system parameter data set for use by MSM.
6	Plays Battle and Culture activity.
7	Computes false alarms and sensor parameter changes.
8	Generates targets from BLUE-RED forces - Plays earliest/latest possible detection times for sensor-target combinations.
9	Line of Sight checks.
10	Creates MSM sensor interrogation events (Type 1) - Adds false target information where required.
11	Merges all MSM events generated by PRERUN.
12	Blocks MSM events (900 or fewer words/block).

The primary PRERUN output is a time-sequenced listing of events which are processed in the Main Simulation (MSM). These events comprise 10 types requiring MSM processing plus an END event. These ten types are shown in Table 1.4-II. Also, output from PRERUN provide various sensor and background parameters necessary for later sensor detection processing.

Table 1.4-II

LIST OF EVENT TYPES

Event Type Code	Event Descriptive Name	From PRERUN Step No.
1	Sensor Interrogate (against Target(s))	10
2	Sensor False Alarm	7
3	Sensor Parameter Change	7
4	Sensor Up/Down Status Control	2, 4
5	Monitor Up/Down Status Control	1
6	Data Link Up/Down Status Control	1
7	Firetrap Up/Down Status Control	1

Table 1.4-II

LIST OF EVENT TYPES

Event Type Code	Event Descriptive Name	From PRERUN Step No.
8	Arrays: Emplace/Cease Operations	1, 4
9	Battlefield Illumination	6
10	Sensor Reposition (coordinate change if reemplacement occurs)	3
99	END (terminate MSM processing; no more event data)	3

PRERUN processing details are presented in Section 2.1 of this Volume; the sensor background subroutines used in Step 7 are discussed in Section 3.2; the interaction subroutines involving emplacement and reliability, ground truth position, sensor/target interaction, false alarms, and line of sight are discussed in Section 4. Cultural and battlefield effects are discussed in Section 5.4.

The next step in the model structure involves the use of PRERUN output event schedules to drive a single job step the Main Simulation Model (MSM). The sequence of these events dictates the dynamic execution within the MSM. Additional inputs are required involving time, terrain, equipment parameters, atmospheric conditions, and other miscellaneous data.

The MSM is a complex of 75 subroutines that simulate the actual sensor equipment performance and also provide for necessary processing control, input and output processing and other auxiliary computations. These 75 MSM subroutines fall into 10 general categories as shown in Table 1.4-III.

Table 1.4-III

MSM SUBROUTINES

Category	No. Programs
Block Data	1
Executive Routines, Level 1	3
Executive Routines, Level 2	8
Executive Routines, Level 3	8
Sensor Subroutines	11
Output & Output-Related	9

Table 1.4-III

MSM SUBROUTINES

Category	No. Programs
Input Auxiliaries	3
System Utility Routines	9
Storage Access Utilities	6
Geometry & Other Auxiliary	17
Total	75

Included in the MSM are nine generic sensor type performance models as follows:

- o Seismic
- o Acoustic
- o Magnetic
- o ARF BUOY
- o Passive Infrared
- o Radar
- o Imaging devices
- o Thermal devices
- o Breakwire

The MSM output, in the form of an "immediate" printed output and a data set stored on tape or disc, consists of the time histories of the various sensor detection results together with amplifying data on target identity, game play and game truth. In addition at the end of a game certain summarized information is printed. The MSM processing is discussed in Section 2.2 of this Volume and the nine Sensor Subroutines themselves in Section 3.3.

Three further information processing submodels are included in the overall model structure. These submodels are in the category of output processing--that is, they are intended for use in deriving further information from the sensor detection decisions output by the MSM. They are not, at this time, designed to be directly linked to the MSM output, which means the inputs to these models require manual preparation using the MSM output data.

The Unattended Sensor Analysis submodel has been designed for the purpose of processing and analyzing activation signals (including false alarms) received at a remote monitor(s) from various types and combinations of unattended sensors/sensor arrays and providing target reports in terms of timeliness, accuracy and content. This submodel consists of 22 subroutines which presently analyze inputs that consist of manually prepared activation events received over data links from the various unattended sensors and arrays of sensors. These subroutines perform a series of evaluations of the sensor activation data in developing a variety of information for inclusion in target reports. These reports can include target presence, type, speed, direction of movement, length, number of elements, estimated times of arrival at future positions, times of occurrence of various events and target locations as a maximum depending on mission objectives and sensor array configurations utilized. This submodel is discussed in detail in Section 6.4 of this report.

The Attended Sensor Analysis submodel is designed to transform target detection signals from manned type sensors (radar, image devices) which are the output of the attended sensor performance subroutines in the MSM into the type of target information which an operator would derive. For a radar, for example, this submodel will yield target range and bearing based on MSM detection decisions and sensor game play locations. This model has been designed with exemplary linkages to the MSM output tape in order to demonstrate how this can be accomplished in future model expansion. This submodel is discussed in detail in Section 6.5.

The Tactical Communications submodel is designed to simulate the processing of STANO system derived target information messages between various command levels of the brigade. It consists of 12 subroutines and is designed around time delay considerations in such message communications. It is designed as an auxiliary model which also requires manually prepared input based on MSM output. Four communications nets, four levels of command, and interaction with non-STANO traffic are included in the programs design. Total delay from operator target recognition to message delivery is the output for each message introduced in the programs. This submodel requires manual preparation of input message parameters and is discussed in Section 6.6.

1.5 APPLICATION OF THE PHASE I SYSTEM ASSESSMENT MODEL

The preceding paragraphs have delineated the situational problem simulated in SAM I. Also, the model features have been discussed and the overall model structure in terms of the major submodels has been described. It seems appropriate now to discuss briefly some of the more significant applications of the Systems Assessment Model. Experience in using the model will undoubtly aid in the discovery of important applications other than those included herein, however, the uses discussed below will serve to indicate the potential worth of applying the SAM I to U.S. Army problems.

The acceptance of detailed input data and the modular features of the model provide the necessary flexibility required for application of SAM I to a broad spectrum of problems ranging from single-sensor simulations to complete large scale brigade STANO system operations. Specific applications are outlined below.

a. SINGLE-SENSOR PARAMETRIC/SENSITIVITY ANALYSES

Purpose

Evaluation of changes in sensor performance, versus controlled changes in:

(1) sensor parameters (including hypothetical changes not necessarily available in current hardware)

(2) target types and target-sensor geometries

(3) environmental factors (meteorological, terrain, and noise-producing background)

Characteristics of Program Operation

Single sensor evaluations

Low storage requirements

Fast computation

Explicit planner control of game (e.g., no red vs. blue strategy; parameter variations controlled, not stochastic)

Critique

Very simple application of SAM I model

Absolute performance values of less importance than relative changes caused by controlled perturbations

In the early post-development period, the sensitivity measures (and, to a limited extent, the absolute performance values) are important in:

(a) providing limited-scope data that can be compared against field data, for validation of program, and/or

(b) indicating what parameters are critical, so that the problem of verifying simulation accuracy, or of obtaining better numerical data (by field test, for example) can be split into "critical" vs. "noncritical" categories.

b. SINGLE-ARRAY PARAMETRIC/SENSITIVITY ANALYSES

Purpose

Evaluation of relative effects on array performance versus controlled changes in:

(1) target types and target locations relative to array

(2) environmental factors (meteorological, terrain, and noise-producing background)

(3) local geometries of sensor emplacements within the array

(4) mixes of different generic types of sensors within the array

(5) reporting and decision logic involved in multiple-sensor-array data interpretation

Characteristics of Program Operation

Single arrays (with variable number of sensors, possibly of mixed types)

Low storage requirements

Fast computation

Explicit planner control of game (e.g., no red vs. blue strategy; parameter and logic variations controlled, not stochastic)

Critique

Simple application of SAM I Phase I model, but moderate amount of pre-planning required.

Application to an entire level of complexity higher than the single-sensor simulations.

Results, however, are more useful in evaluating the doctrine of sensor use (at a local level) than in evaluating individual sensors, so this application does not replace the single-sensor simulations in terms of value.

c. LARGE-SCALE SYSTEMS OVER LARGE AREAS, LONG TIMES

Purpose

Bona fide use of the SAM I to answer basic military questions on sensor system deployments, operations, etc.

Two basic problems connected with bona fide large-scale simulations hinge on:

(1) Problem definition: exactly what useful answers are expected from the simulation runs?

(2) Experimental design: exactly how is the program to be controlled over a sequence of runs, so that statistically valid conclusions can be drawn within reasonable computer running time?

Characteristics of Program Operation

Requires exercise of complete model

Maximum storage requirements

Planner data inputs large

Parameter and logic variations stochastic

Provides data applicable to assessing large-scale system performance

Allows alternative levels of activity to be evaluated.

Critique

The stochastic elements of the game, including the implied independent red-vs-blue strategy in target and sensor layouts, imply that "experimental design" carries most of its formal statistical meaning. Thus the collective "user" needs statistical/analytical support as well as military judgment.

The use of a portion of a "previous" run as a partial basis for a "new" run requires careful attention to the control of the numerous random generators internal to the program.

d. OTHER USES OF THE MODEL ARE:

(1) Support of field tests by employing the SAM I to:

(a) Screen candidate systems to be tested

(b) Assess the effect of small changes in a candidate system shown from field test to be better than other candidate systems. The type of "what if" questions that could be further explored without additional field test might be as follows:

- What if environmental conditions (weather, time of day, etc.) had been different?
- What if failure rate of equipment had been different?
- What if sensor location errors are different?

(c) Screening organizational and operational concepts to be tested

(2) Support for development of new items of STANO equipment by assessing effect of environment on conceptual hardware designs.

(3) Support of Engineering Test by screening items to be tested.

1.6 CONCLUSION

The Phase I Systems Assessment Model as now developed is a large and complex simulation. To give some perspective to its present size, the dimensions of the basic planner input data sets are shown in Table 1.6-I. It totals among its nine submodels approximately 240 subroutines and the total program involves some 40,000 FORTRAN statements. As currently implemented in terms of designer inputs, terrain, environment and other aspects, it is applicable to low-intensity Southeast Asia conflicts involving a U.S. brigade size operation. There are, however, no known limitations to its being expanded to larger size operations, higher conflict situations, or different areas of conflict. Nine major generic sensor types have been modeled and additional ones can be easily added as required.

In addition to its capability of simulating rather large scale (brigade size) STANO systems, it also provides efficient simulation for smaller scale problems - for example, sensitivity analyses - that may have nearly equal near term importance. A large number of system features are treated in detail, yet the program is not self-limiting; flexibility is maintained for additions, deletions, or changes that would naturally be indicated as the base of available data, the development of new system hardware or operational and organizational concepts, and the experiences with the initial version of the model - all change with time.

Table 1.6-I

PLANNER INPUT DATA SET DIMENSIONS

Data Set	Dimension
Unattended Ground Sensor Arrays	200
Position Error Parameter Set	50
Sensors	400
Sensor Description Parameter Set	100
Firetrap Kill Point Systems	50
Monitors	100
Monitor Parameter Set	10
Relays	50
Relay Reliability Parameter Set	10
Data Links	500
Receiver/Transmitter Parameter Set	10
Path Data	275
Force Type Parameter Set	100
Coverage Scan Parameter Set	150
Navigation Systems (4 types)	4 (each type)
Stationary Scanning Arrays	300
Moving Arrays	200
Blue Forces (Potential False Targets)	300
Red Forces (Targets)	100

Notes:

1. Total sensors is 400, which is less than the sum of the (maximum) dimensions for each sensor type array.
2. The Blue Force includes moving arrays; thus, if 200 moving arrays are input, an additional 100 Blue Forces (non-sensors) maximum is allowed.

Section 2
MODEL PROCESSING

2.1 PRERUN

2.1.1 Introduction

PRERUN is the first of the two primary and major Phase I System Assessment Model sections. Its purpose is to simulate all the activities associated with the scenario and STANO system operation up to the point of sensor detection simulation. The basic purpose of PRERUN is thus to produce an ordered sequence of events for MSM processing through the various generic sensor type performance models. As such, PRERUN includes all the reliability subroutines; ground truth positions are computed and the space-time intersections of targets and sensors are computed. Line-of-sight is treated where necessary. The Culture and Battle programs are called to produce background effects, false targets, and illumination-type events. False alarms and sensor parameters which are functions of environmental conditions are computed. An attempt has been made to include as much preprocessing as possible in order to pass only significant events to the Main Simulation Model (MSM).

The primary input to PRERUN is the Planner Scenario Data. Other inputs include the Atmospheric Data generated by the Atmospheric submodel, the Digital Terrain Data (from MAKTAP Program) and terrain description data (subroutine TERAN). These are discussed further in Section 2.1.2.

As noted above, the primary output is a time-ordered sequence of events for processing in the MSM. Ten types of events are provided as shown in Table 2.1-I. These events are generated in PRERUN as the result of simulating activities in the scenario area leading up to sensor detection simulation. Based on planner inputs, sensors in arrays, data links, relays, and monitors are emplaced and up/down times computed. Emplacement errors are computed and ground truth positions established. Targets (from both red and blue forces) are moved through scenario area as dictated by planner and times of possible detection by those sensors which are operational are computed. Battlefield and cultural environment events and background effects are assessed and false alarms and sensor parameter changes computed. Sensor line of sight is checked where appropriate. The PRERUN thus serves to simulate all the movements and activities on the scenario area that may influence sensor performance. All such activities are scheduled by time and type for playing in the MSM.

PRERUN has been subdivided into 13 job steps, containing 96 subprograms and 20 internal data sets. The steps are shown in Table 2.1-II. This has alleviated the need for large blocks of storage and allows the program to be run without requiring a large single block of computer time. The steps have been so designed that further subdivision is easily done if it is desired to reduce the storage requirements and/or process a much larger scenario than the original design specified. Thus, although this version was run on an IBM 360/65, it is adaptable to other machines. The use of USASI FORTRAN throughout frees PRERUN of any machine dependence.

Table 2.1-I

LIST OF EVENT TYPES

EVENT TYPE CODE	EVENT DESCRIPTIVE NAME
1.	Sensor Interrogate (against target(s))
2.	Sensor False Alarm
3.	Sensor Parameter Change
4.	Sensor Up/Down Status Control
5.	Monitor Up/Down Status Control
6.	Data Link Up/Down Status Control
7.	Firetrap Up/Down Status Control
8.	Arrays: Emplace/Cease Operations
9.	Battlefield Illumination
10.	Sensor Reposition (coordinate change if reemplacement occurs)
99.	END (terminate MSM processing; no more event data)

Table 2.1-II

PRERUN STEP DEFINITIONS

STEP	DESCRIPTION	MSM EVENT CREATED
0	Sets Game Variables - Reads and Converts Planner Input	
1	Computes Up/Down Times of Monitors, Firetraps, Relays, Data Links and UGS Arrays	5,6,7 & 8
2	Computes Up/Down Times of Sensors in UGS Arrays and STASCAN Arrays	4
3	Computes Ground Truth Positions for UGS Arrays and STASCAN Arrays	10
4	Computes Ground Truth Positions and Up/Down Times for MOV Arrays	4,8
5	Prepares System Parameter Data Set for use by MSM	
6	Plays Battle and Culture	9
7	Computes False Alarms and Sensor Parameter Changes	2,3
8	Generates Targets from Blue-Red Forces - Plays ELPDT (Sensor-Target Detections)	
9	Line-of-Sight	
10	Creates MSM Event 1 - Adds False Target Information Where Required	1
11	Merges all MSM Events Generated by [illegible]UN	
12	Blocks MSM Events (900 or Fewer Words/Block)	

The relation of PRERUN to remainder of SAM is shown in Figure 2.1-1. The following paragraph will discuss its operation in more detail. Since what it accomplishes is best understood in terms of its inputs and outputs, these are covered first in Sections 2.1.2 and 2.1.3. Following this the internal PRERUN structure is discussed together with several important features (Section 2.1.4). Following this, the descriptive comments of each subroutine are presented (Section 2.1.5) and, finally, the PRERUN common areas are presented (Section 2.1.6) since these govern the size of the model.

2.1.2 Input Data

Three input data sets are required for PRERUN as follows:

(a) Planner Input Scenario (data on cards via SYSIN).

(b) Time parameters (BASICT) and Atmospheric data tables (ATMON) (disc file; MASSDAT; prepared by Atmospheric model prior to PRERUN execution).

(c) Digital Terrain* (tape file; JPOUT; prepared by MAKTAP prior to PRERUN execution).

The planner prepared scenario data is entered into the program by a card deck that includes an initial block of 18 header cards followed by cards, corresponding to 29 major data sets. The card deck is illustrated in Figure 2.1-2 and the data sets are listed in Table 2.1-III. Suggested formats of the header cards and a detailed description of the data sets and their preparation is contained in Volume II (Section 6 and Appendix F).

2.1.3 Output Data

PRERUN generates two outputs, both of which become the primary inputs to the MSM. These two outputs are:

(a) System Parameters (edited for MSM use; disc file; JTFWDF).

(b) Events (disc file; EVENT 1).

The system parameters are the sensor system parameters as received from planner input and prepared or modified in PRERUN. These system parameters are prepared for the MSM in PRERUN Step 5

---

*Not required if dummy line-of-sight routine is used.

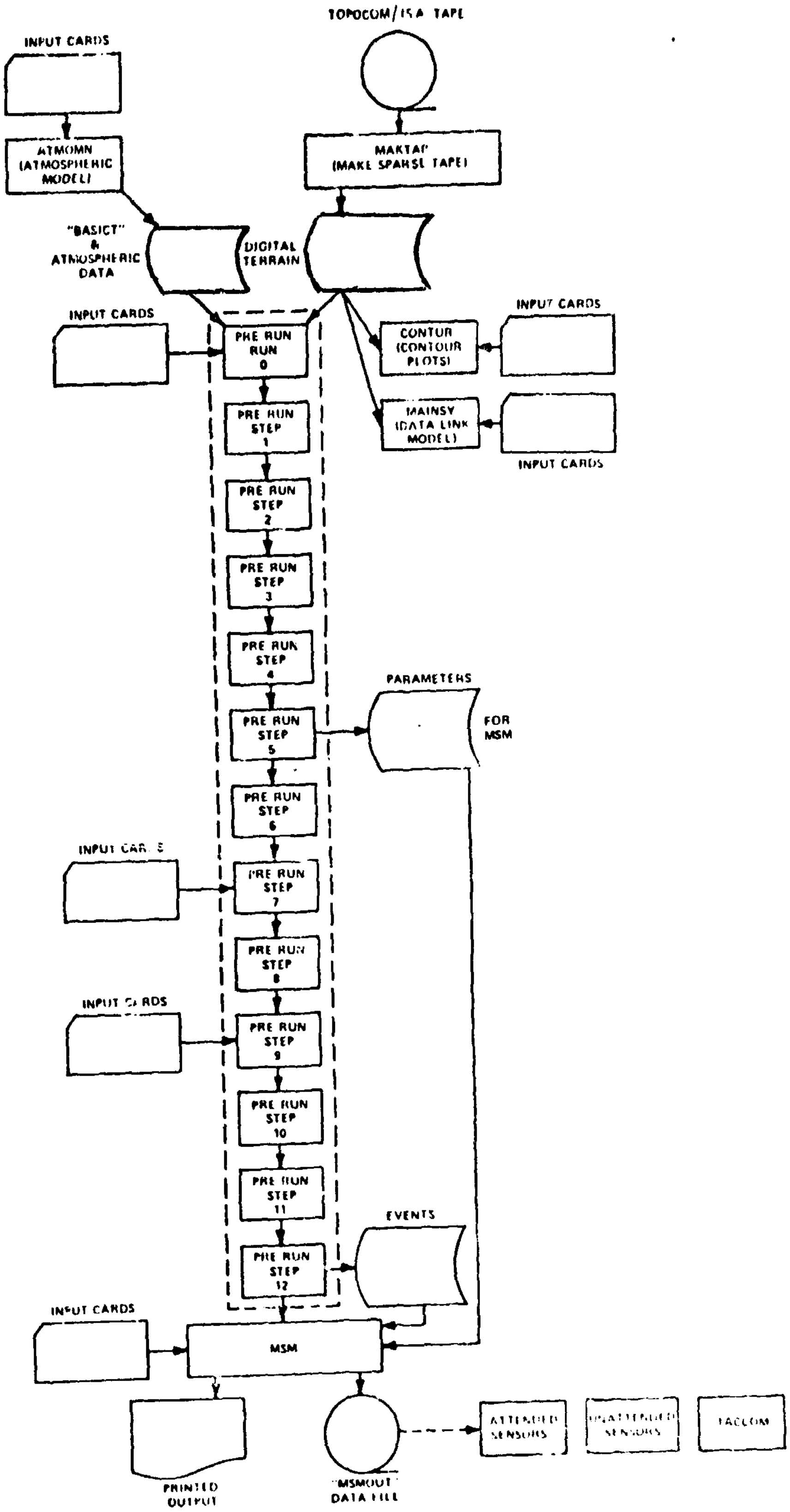


Figure 2.1-1 RELATIONSHIP OF PRERUN TO REMAINDER OF SAM

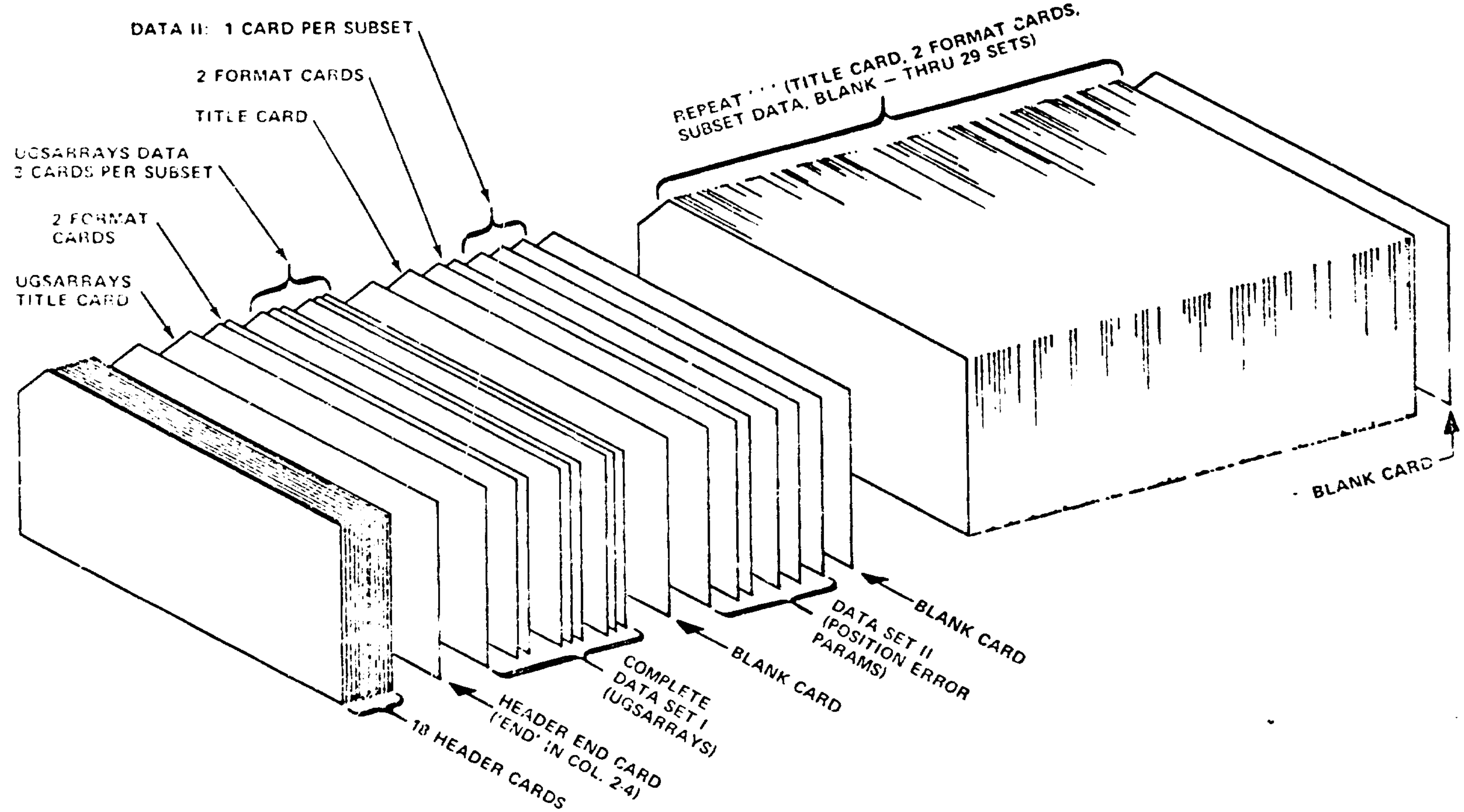


Figure 2.1-2 INPUT CARD DECK FOR PRERUN
(HEADER CARDS AND PLANNER SCENARIO CARDS)

Table 2.1-III

PLANNER INPUT DATA SETS FOR PRERUN
(SCENARIO SPECIFICATIONS)

DATA SET	NAME
0	Header Cards
I	Arrayugs
II	Position Error Parameter Set
III	Sensors
IV	Sensor Descriptor Parameter Set
V	Firetrap Kill Point System
VI	Monitors
VII	Monitor Parameter Set
VIII	Relays
IX	Relay Reliability Parameter Set
X	Data Links
XI	Receiver/Transmitter Parameter Set
XII	Path Data
XIII	Force Type Parameter Set
XIV	Coverage/Scan Parameter
XV	Navigation System (HYPERBOLIC)
XVI	Navigation System (RHO-THETA)
XVII	Navigation System (DOPPLER)
XVIII	Navigation System (Normally Distributed Errors)
XIX	STASCAN Arrays
XX	MOV Arrays
XXI	Blue Forces
XXII	Red Forces
XXIII	Battle PIEVT Table (Planner Events)
XXIV	Battle RSEVT Table (Random Events)
XXV	Battle XCLUA Table (Exclusion Area)
XXVI	Battle FSPTB Table (Fire Support Base)
XXVII	Culture PCEVT Table
XXVIII	Culture RCEVT Table
XXIX	Culture SNFDX-Y

using a converted data set of planner input (DATAIN) and the ground truth position tables of the sensors which in turn are generated in earlier PRERUN steps. The parameters transferred to MSM include only those used in the MSM and thus are subsets of the original planner inputs. Specific numeric values of data may also be different from original planner values due to unit conversion and "game truth" changes. MSM uses only "game truth" values for coordinates and for operational times of sensor system elements. These "game truth" values are derived in PRERUN. The system parameter output data to MSM is contained in a data set labeled JTFWDF and the categories of information are shown in Table 2.1-IV. Appendix C contains additional information.

The other PRERUN output, the EVENTS provide the dynamic linkage between PRERUN and MSM, and the entire concept of being able to have a PRERUN submodel is based on being able to establish such an event schedule. By following such an event schedule and only calling the sensor models into the simulation as required, computer running time and storage needs are minimized. PRERUN computes 10 types of events as previously shown in Table 2.1-I and merges them into a time-ordered sequence. These are placed on a disc file in groups not exceeding 900 words for use by the MSM (again for computer storage requirements considerations). Each event has an associated sublist of words with the structure designed for model growth in two respects:

(a) Although only 10 event types are provided in this initial model, no restriction on number of types exists in basic format.

(b) Lists for each event are not restricted to a fixed length.

Referring again to Table 2.1-I, the type 1 events are the primary items of interest. It is these events which call a sensor into action against a potential target. The creation of the type 1 events occurs in PRERUN Step 10 based on Steps 8 and 9 results. In Step 8, all sensors are played against all targets for "geometrical detection" through ELPDT subroutine. That is, each target which is in the geometrical area of contact by a sensor is computed. In Step 9, the "geometrical detection" of Step 8 is checked for line-of-sight if the sensor is line-of-sight sensitive. The type 1 events thus cover those times when a target is within the potential detection area or range of a sensor. It should be noted that false targets will also generate type 1 events based on targets or events output from the battle and culture environment in Step 6.

Type 2 events (false alarms) are created in PRERUN Step 7 using planner, atmospheric, battle, and cultural data. Type 3 events (sensor parameter changes) are also computed in this PRERUN step due to atmospheric variations and background noise levels computed from battle and cultural subroutines.

Table 2.1-IV

DATA CATEGORIES WITHIN PRERUN OUTPUT, JTFWDF

TITLE	ANALOGOUS PLANNER SET
UGSARRAYS	I
STASCAN ARRAYS	XIX
MOV ARRAYS	XX
BLUE FORCES	XXI
RED FORCES	XXII
SENSORS	III
SENSOR DESCRIPTOR PARAMETERS	IV
FIRETRAPS	V
MONITORS	VI
DATA LINKS	X
PATH DATA	XII
FORCE TYPE PARAMETERS	XIII
COVERAGE/SCAN PARAMETERS	XIV

Types 4, 5, 6 and 7 events are up/down status controls associated with sensors, monitors, data links, and firetraps respectively, while type 8 is the emplace/cease operation flags on sensor arrays. These events come from the PRERUN Steps 1, 2, or 4 where up/down times of each item is computed.

Type event 9 (battlefield illumination) is derived in PRERUN Step 6 based on battlefield illumination event inputs and type event 10 (sensor reposition) occurs based on planner input in PRERUN Step 3.

Complete details regarding the Event 1 output are included in Appendix A.

2.1.4 PRERUN Structure

2.1.4.1 Job Steps

As noted earlier, the PRERUN has been divided in 13 job steps (numbered 0 - 12 as shown in Figure 2.1-3) and is comprised of 96 subroutines. Table 2.1-II defined the steps and related them to the output events for MSM. These job steps must be run in the order shown although in separate units of computer time if desired. Use of this technique has significantly reduced the amount of computer storage required.

The 96 subroutines may be divided into five classes as shown in Table 2.1-V as:

1. There are 14 executive subroutines. These are the main programs which control each job step. Note that Step 9 has two main programs. MAINLS controls the main line-of-sight routine which computes time line-of-sight using subroutine LOS, MICTER, TERAN, BRKLOS, and FOLAGE. PREMNC controls a call to a dummy line-of-sight routine LSGT which is used when it is not desired to play true line of sight.

2. There are 15 sub-executive subroutines. These routines locate the planner input data appropriate for the job step and call the model routines.

3. There are 24 model subroutines. These are the main computing routines of prerun which compute up/down times, ground truth positions, target, sensor locations early/late detection times, line-of-sight, etc.

TABLE 2.1-V
PRERUN SUBROUTINES

STEP	EXECUTIVE	SUB-EXEC	MODEL	SPECIAL PURPOSE	UTILITY
0	INMAIN	READIN	(SCREEN) CONVRT	TIMER	ERASE
1	PREMN1	UPDN1 UPDN5 UPDN6 UPDN10 UPDN8	READUP COMMUP		FINDX FINDY MERGDR DORDER GMERGE MERG1 URN GRN TRAN TRAN2 TRNSFR
2	PREMN2	UPDN3 UPDN19	RUSUP	EVNT 48	
3	PREMN3	PSNP PSNP19	SNPGT	DOPLER HYPERB RERR NORMER RHOTHE JFBLK3	
4	PREMN4		MVS		
5	TRNPAR	TRNPR1		CKSOUT VALID	
6	PREMN6	CULTEX BATEX	BATLBK BATLTG BSCHDL CSCHDL CULTBK	EVNT9 PATHS SENXY NUMBER SELCTR JFBLK6 BEMAP	
7	EVNT23		SEISBK ACOUBK ARFBK ENVIR PIRBK BWIRBK	FAINTV IUTEVL TWBLKD TERAN VALID	
8	PREMN8	TARGEX ELPEX	TARGBR ELPDT	SENSQ CIRC SECT GREC SECLOG VALID	
9	PREMN9 MAINLS		LOS	LSGT MICTER IUTEVL TERAN TERANE BRKLOS FOLAGE	
10	PREMNA		FLSTG SEQ		
11	PREMNB		FMERGE		
12	PREMNC				

SUBROUTINE COUNTS

EXECUTIVE	14
SUB-EXEC	15
MODEL	24
SPECIAL PURPOSE	31
UTILITY	12

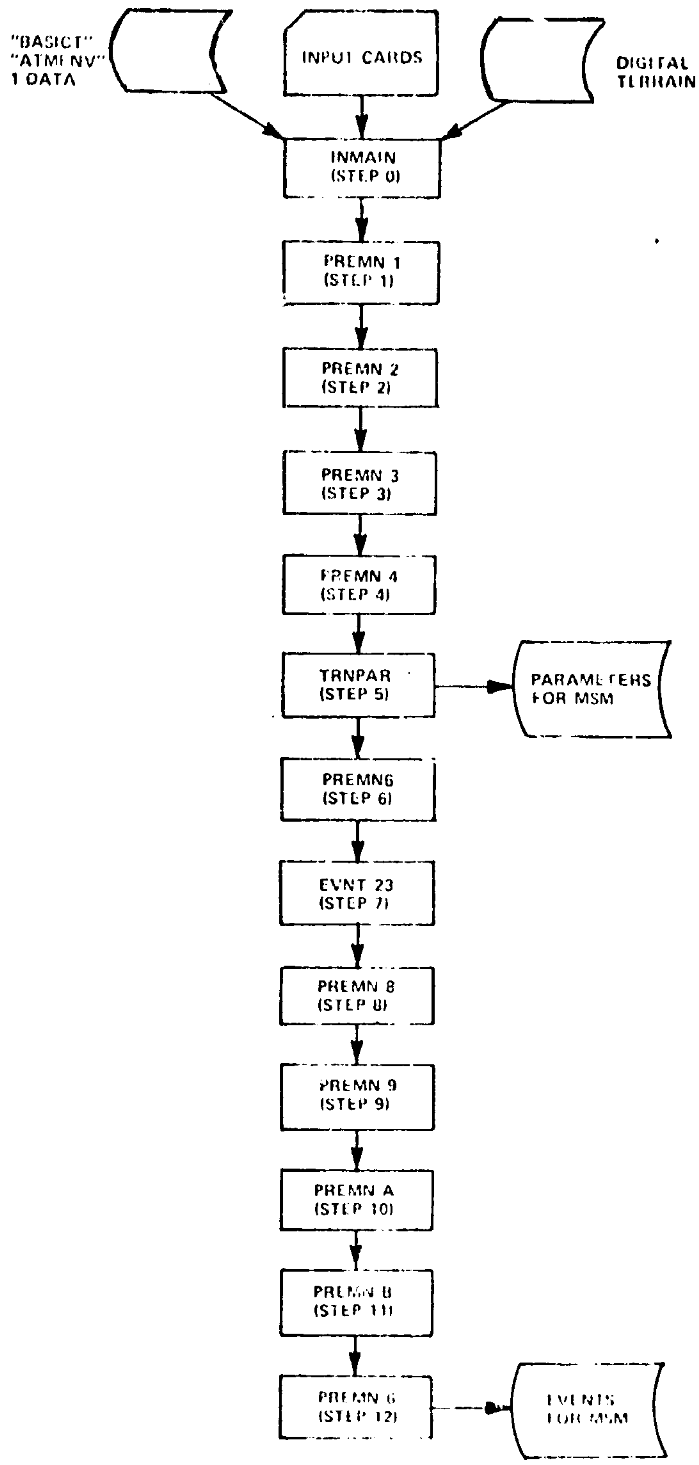


Figure 2.1-3 PRERUN JOB STEPS

4. There are 31 special purpose subroutines. Most of these are short subroutines which perform special computations such as GREC, SECT, and SIRC which are geometry subroutines used by ELPDT to calculate sensor-target interaction or perform special tasks such as VALID which determines valid sensor-target combinations as specified by designer table.

5. There are 12 utility subroutines used by many of the programs to locate parameter, order data sets, merge events transfer data, and generate random numbers.

2.1.4.2 Common Features

PRERUN has several common features which are important to model design and processing. These include the following:

1. Compressed storage is used whenever possible to eliminate the need of assigning unused storage space. The primary compressed data sets are:

   a. The master data stream in the common statement:

COMMON/DATAIN/NDATA, NSETS, LDATA(M), NDATC(M), IDATA (N).

where:

NDATA	=	actual number of data points in IDATA
NSETS	=	number of major data sets
LDATA(J)	=	pointer locating the location of the first point of major data set J in stream
NDATA(J)	=	count of the number of points in each subset of set J
IDATA( )	=	the master data stream
M	=	storage allocated for pointer LDATA - must be one greater than the number of major data sets
N	=	storage allocated for master stream - depends on the planner scenario

For brevity, the subroutines use this in the form:

COMMON/DATAIN/NNN, MMM, LD(M), NENT(M), ID(N)

The number of major sets is easily extended but the order of the defined sets must not be changed. The format is controlled by header cards read in front of each major set. A data set may be empty but the header cards and a set of blank data cards must be in the proper order. The number of sub-sets in each major set is variable and determined automatically by the read-in routine.

Internally, PRERUN locates a data set by use of a subroutine called FINDX. For example:

> CALL FINDX (3, II, IA, IB) returns the II as the location of first point in set 3, IA the location of the last point, and IB the number of points in each sub-set. Set 3 is the sensor data. A do loop.
>
> DO 100 I = II, IA, IB will scan all the sensors.

If the set is void (no sensor data given), FINDX will return a zero for II.

b. The up-down times of the various elements in the common statement as used in Step 1.

COMMON/UPDOWN/NT, UDTM(M1), KREL(M3), KDLK(M4), KARR(M5)

where:

NT	=	number of points in the data stream UDTM
UDTM(M1)	=	the sequence of up/down times
M1	=	storage allocated for UDTM
KMUD(M2)	=	pointer for monitor data
M2	=	must be one greater than the number of monitors
KREL(M3)	-	pointer for relay data
M3	-	must be one greater than the number of relays
KDLK(M4)	-	pointer for data links
M4	-	must be one greater than the number of data links
KARR(M5)	-	pointer for UGS arrays
M5	-	must be one greater than the number of UGS arrays

Since the number of up/down times depends on the statistical results of the reliability routines, the dimension of the storage allocated for UDTM may be difficult to determine in advance.

At the end of Step 1, this is further compressed and only the times for the UGS arrays are saved with the appropriate pointer and written on disc for use of subsequent steps.

The common statement is then used as:

COMMON/UPDOWN/NT, UDTM(M1), KARR(M2), KSDN(M3)
or COMMON/UPDOWN/NT, UDTM( ), KARR( ), KSEN( )

where:

KARR(M2) is the pointer for all arrays

KSDN(M3) is the up-down pointer for all sensors (KSEN)

(M2, M3 must be one greater than the total number of arrays and sensors, respectively).

Updown times are located by use of a subroutine FINDY. For example, the call to FINDY:

> Call FINDY (KARR, M2, I, IA, IB) will return IA equal to the location of the first up time of array I and IB equal to the location of the last down time for this array. For each element, the times have been stored sequentially in pairs. If the element was never up, no times were stored and FINDY will return a zero for IA.

c. Ground truth positions of sensors and corresponding times are stored in the compressed set SXYTT.

COMMON/STASEN/NXY, KSSN(M1), SXYTT(M2)
COMMON/PXYTP/NT, KXYT(M1), SXYT(M2)
(these two sets are equivalent)

where:

NXY	=	number of points in SXYTT
KSSN(M1)	=	time and position pointer for sensors
M1	=	must be one greater than the number of sensors
SXYTT	=	compressed storage

For the stationary sensor, the data is stored in the form:

DX, DY, X, Y, TA, TB

where DX and DY specify the orientation of a path defining the sensor position. If the sensor is not located relative to a path, DX and DY are zero. X, Y are the ground truth positions for the sensor and TA and TB are initial and final times. If the sensor is relocated, the stream is extended by adding X, Y, T, T as many times as necessary.

For moving sensors, the data is stored in the form -

XA, YA, TA, XB, YB, TB

which defines to beginning and end points of the sensor position. The sensor is assumed to move in a straight line at constant velocity between the space time point XY, YA, TA and the point XB, YB, TB.

If a second leg is used, the triad XC, YC, TC is added, etc.

The subroutine FINDY is used to locate the data referring to a particular sensor.

d. The MSM event list is stored in

COMMON/EVENTS/MEV, IEV(M1), IVE(M2), MVE

where:

MEV	=	number of cards in IEV
IEV	=	master storage of MSM events
IVE	-	temporary working storage
MVE	-	number of words in IVE

In use, events as created are placed on the array IVE and then ordered and merged by blocks into the master stream IEV. At different places in PRERUN the master list is stored as a record on a disc and a new list started in order to handle the complete event list for MSM which may require a very large amount of storage.

Events are located by starting at the first word in any record and using the format of the event itself to find the next event.

e. In several places where a variable number of fixed length sets are required, the subscripting convention A(N, M) is used where N is the fixed length and M is the variable. This packs the sets in the core to use only the space required.

1. An element may be removed from play by specifying a planned up time equal to or greater than the planned down time. This creates a great versatility for the planner when he chooses to modify the scenario.

2. Various play and print options are available. Using the play option, entire sets may be played using planner input data as given. The print options control various BCD output; the same is used for reporting the results of various reliability routines and some are used for program checkout purposes.

3. Designer input values are inserted in PRERUN by the use of data statements in the appropriate routines and by the use of block data sub-programs.

4. The following basic game information is input through PRERUN step 0 and is transmitted to all subsequent PRERUN steps via disc:

TSTART	=	start of game in seconds
TMAX	=	end of game in seconds
ZMAP	=	standard deviation of map error in meters
XLOC	=	1 play location errors, = 0 don't play
RELOC	=	1 play relocation errors, = 0 don't play
ANAV	=	1 play navigation errors, = 0 don't play
ARTY	=	1 play out/mortar location errors, = 0 don't play
AIRD	=	1 play vertical fall error
XSW	=	southwest X coordinate of game area, meters
YSW	=	southwest Y coordinate of game area, meters

XNE	=	northeast X coordinate of game area, meters
YNE	=	northeast Y coordinate of game area, meters
IPRINT	=	BCD output unit
ICARD	=	card input unit
MPRINT( )	=	BCD output options
MTAPES( )	=	defines disc units
NPLAY( )	=	option to use planner input

The above variables are set by PRERUN - STEPO - INMAIN executive routine.

5. Where both fixed (integer) and floating point variables are stored in the same array, an equivalence statement is used. This makes it easier to do the Fortran coding. For example:

```
DIMENSION      FR(10), IR(10)
EQUIVALENCE  (FR(1), IR(1))
```

This allows the FORTRAN statements (without conversion)

```
A  =  FR(1) + B
I  =  IR(2) + K
```

where a floating number is stored in the 1st cell of the array and an integer in the second.

2.1.4.3 PRERUN Data Sets

Figure 2.1-4 provides a detailed master diagram of the roles of data sets in linking together the job steps of the overall model in general, and of the job steps in PRERUN in particular. The boxes along the left edge represent job steps or subroutine packages. The symbols across the top represent data sets with currently assigned data set names (except that 'SYSIN', card reader, and PRINTER are not job unique names).

The connection matrix is interpreted in terms of the horizontal line from a subroutine box (to a vertical line for a data set):

(a) if the arrow points to the subroutine box, then that subroutine reads data from the corresponding data set,

(b) if the arrow points away from the subroutine box, then that subroutine writes data onto the data set,

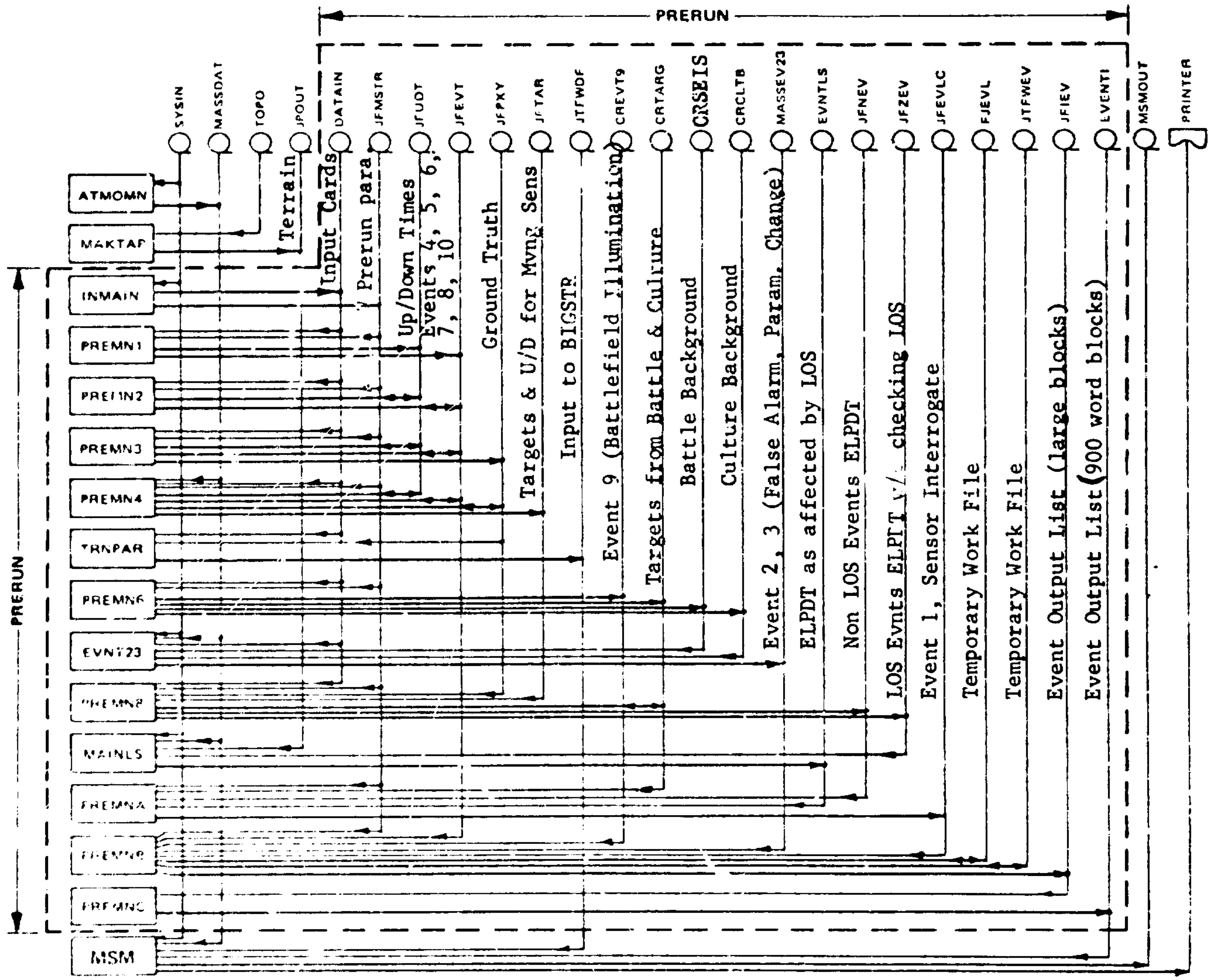


Figure 2.1-4 DATA SET LINKAGE

(c) if a double arrow appears (both directions), then both reading and writing occurs; i.e., the data set is altered (updated).

Except for the printer, data sets are indicated for convenience by a common (tape) symbol. The actual physical devices are, however, chosen by the programmer/user. Recommended or typical device choices are:

SYSIN	card reader
TOPO) JPOUT)	tape
MSMOUT	tape or disc
all others	disc

The Data Sets used in PRERUN are as follows:

	M Tape	Name
Master Data Stream	1	DATAIN
Common Game Information	2	JFMSTR
Up-Down Times	3	JFUDT
Master Event List types 4-5-6-7-8-10	4	JFEVT
Ground Truth Information	5	JFPXY
Atmospheric Data	6	MASSDAT**
Targets from MVS	7	JFTAR
Targets from Battle-Culture	8	CRTARG
Event type 9 from Battle	9	CREVT9
Events 2-3	10	MASSEV23
Type 1, Events	11	JFEVLC
Work unit all events	12	JTFWEV
Work unit for FMERGE	13	FJEVL
Merged Events	14	JFIEV
Early late detection - NLOS	15	JFNEV
Early late detection - LOS	16	EVNTLS
Early late detection info for LOS	17	JFZEV
Final MSM Blocked Events	18	EVENTJ*
Culture Background	19	CRCLTB
Battle Background	20	CRSEIS
Planner Input for MSM		JTFWDF*

*These two data sets are the primary input to MSM.

**This data set comes from the Atmospheric Model

All other data sets are internal to PRERUN. They contain information that may be of value to any output processor.

The executive subroutines reading or writing the data sets are shown pictorially in Figure 2.1-4.

2.1.5 PRERUN Descriptive Summaries of Subroutines

In this section are contained descriptive summaries of each subroutine comprising the PRERUN submodel. The subroutines are grouped by job step and then by type of subroutine within job step. All the utility subroutines are grouped together at end of this section.

2.1.5.1 PRERUN Step 0

Step 0 comprises a main program (INMAIN) and four subroutines. It uses data set, SYSIN, and enters operating parameters for the PRERUN steps by three mechanisms:

(a) DATA statements within subroutines INMAIN and CONVRT.

(b) FORTRAN statements (of form 'parameter name = number') within INMAIN.

(c) Planner prepared data cards (Scenario Specifications).

In addition to providing the direct read of planner input data, this job step also:

(a) converts data from "external units" to consistent "internal units" of measurement (e.g., all angles are converted to radians, all distances to meters).

(b) stores on a disc file (DATAIN) the so-called Master Data Set.

(c) stores on a disc file (JFMSTR) other data (common information) common to many subsequent job steps.

The Step 0 subroutine descriptive summaries are shown in Figures 2.1-5, 2.1-6, 2.1-7, and 2.1-8. The ERAS2 subroutine is not described herein since it is a general utility routine used in several of the models.

2.1.5.2 PRERUN Step 1

Step 1 in PRERUN comprises the main program (PREMN1) with 7 subroutines shown in Table 2.1-V plus 5 utility subroutines.

External data sets required as input are "DATAIN" and "JFMSTR", both generated in Step 0.

```
C************ ****************** INPUT ***************************************
C*                                                                           *
C*                PRERUN EXECUTIVE -     STEP  0                             *
C*                                                                           *
C*     PURPOSE                                                               *
C*         THIS STEP INITIATES THE MAIN PRERUN SEQUENCE                      *
C*                                                                           *
C*     USAGE                                                                 *
C*         MAIN PROGRAM                                                      *
C*                                                                           *
C*     DESCRIPTION OF PARAMETERS                                             *
C*                 IT DEFINES:                                               *
C*                      IPRINT-FORMATED OUTPUT TAPE                          *
C*                      ICARD - CARD READER                                  *
C*                      TSTART-TIME OF GAME START  (DAY-HOUR-MINUTE)         *
C*                      TMAX  -TIME OF GAME END    (DAY-HOUR-MINUTE)         *
C*                      ZMAP  -STANDARD DEVIATION OF MAP ERROR (METERS)      *
C*                      XLOC =1 PLAY LOCATION ERROR ,=0 DON'T PLAY           *
C*                      RELOC=1 PLAY RELOCATION ERROR ,=0 DON'T PLAY         *
C*                      ANAV =1 PLAY NAVIGATION ERROR ,=0 DON'T PLAY         *
C*                      ARTY =| ART/MORTAR ERROR       ,=0 DON'T PLAY        *
C*                      AIRD =1 PLAY VERTICAL FALL ERROR ,=0 DON'T PLAY      *
C*                      XSW  SOUTH WEST X COORDINATE OF PLAY AREA            *
"*                      YSW  SOUTH WEST Y COORDINATE OF PLAY AREA            *
C*                      XNE  NORTH EAST X COORDINATE OF PLAY AREA            *
C*                      YNE  NORTH EAST Y COORDINATE OF PLAY AREA            *
C*                                                                           *
C*                      MPRINT -AN ARRAY USED TO CONTROL BCD PRINTING        *
C*                              0- DON'T PRINT   1- PRINT                    *
C*                      MPRINT(1)   -UPDN1                                   *
C*                              2   -PSNP                                    *
C*                              3   -UPDN3                                   *
C*                              5   -UPDN5                                   *
C*                              6   -UPDN6                                   *
C*                              8   -UPDN8                                   *
C*                             10   -UPDN10                                  *
C*                             13     CULTURE                                *
C*                             14     BATTLE                                 *
C*                             15     CULTURE                                *
C*                             16     BATTLE                                 *
C*                             18     PSNP18                                 *
C*                             19   -UPDN19                                  *
C*                             20   -MVS                                     *
C*                                                                           *
C*                 IT REQUESTS  DISC (OR TAPE )FILES FOR TRANSMITTAL OF      *
C*                      INFORMATION TO SUBSEQUENT STEPS                      *
C*                      MTAPE  -AN ARRAY USED TO DEFINE BINARY               *
C*                              STORAGE UNITS                                *
C*                      MTAPE(1)    MASTER DATA STREAM                       *
C*                      MTAPE(2)    COMMON INFO                              *
```

Figure 2.1-5

```
C*                        MTAPE(3)    UPDOWN TIMES                              *
C*                        MTAPE(4)    MSM-EVENTS                                *
C*                        MTAPE(5)    GROUND TRUTH                              *
C*                        MTAPE(6)    ATMOSPHERIC DATA                          *
C*                        MTAPE(7)    TARGET INFO FROM MVSNE                    *
C*                               8    TARGET INFO FROM BATTLE CULTURE           *
C*                               9    EVENT TYPE 9 FROM BATTLE                  *
C*                              10    EVENTS 2-3 FROM SENS. PARM.               *
C*                              11    EVENTS 1                                  *
C*                              12    WORK TAPE FOR FINAL MERGE                 *
C*                              13    WORK TAPE FOR FINAL MERGE                 *
C*                              14    MERGED TAPE OF EVENTS NOT BLOCKED         *
C*                              15    EARLY LATE DETECTION- NON LOS             *
C*                              16    EARLY LATE DETECTION- LOS                 *
C*                              17    EARLY LATE ZEV'S FOR LOS INPUT            *
C*                              18    FINAL OUTPUT FOR  MSM BLOCKED EVENTS      *
C*                              19    CULTURE BACKGROUND                        *
C*                              20    BATTLE BACKGROUND                         *
C*                        NSETS NUMBER OF DATA SETS                             *
C*        IF NPLAY=0 PLANNED UP-DOWN TIMES USED                                 *
C*     NPLAY( 1)  UPDN1      ARRAY UGS/MONITOR-DATA LINK                        *
C*     NPLAY( 3)  UPDN3      ARRAY UGS-SENSORS                                  *
C*     NPLAY( 6)  UPDN6      MONITORS                                           *
C*     NPLAY( 8)  UPDN8      RELAYS                                             *
C*     NPLAY(10)  UPDN10     DATA LINKS                                         *
C*     NPLAY(19)  UPDN19     ARRAY STASCAN-SENSORS                              *
C*     NPLAY(20)  MVS        MOVE ARRAYS/SENSORS                                *
C*                                                                              *
C*     REMARKS                                                                  *
C*                                                                              *
C*         ALL OF THE ABOVE PARAMETERS MUST BE SET BY DATA STATEMENTS           *
C*         OR BY FORTRAN STATEMENTS IN THE BEGINNING OF THIS PROGRAM            *
C*         NOTHING IS READ OFF OF THE HEADER CARDS BY THE PROGRAM               *
C*                                                                              *
C*     METHOD                                                                   *
C*         THE BASIC GAME INFORMATION IS SET AND RECORDED ON MTAPE(2)           *
C*         DATA SET-JFMSTR. THE HEADER CARDS ARE READ AND PRINTED.              *
C*         SUBROUTINE READIN IS CALLED TO READ IN THE PLANNER INPUT.            *
C*         THE MASTER DATA STREAM IS RECORDED ON MTAPE(1)-DATAIN.               *
C*         PRERUN ASSUMES THAT THE PLANNER INPUT DATA HAS BEEN PROCESSED*
C*         BY SUBROUTINE SCREEN.                                                *
C*                                                                              *
C*     SUBROUTINES REQUIRED                                                     *
C*         ERASE                                                                *
C*         TIMER                                                                *
C*         READIN                                                               *
C*                                                                              *
C*******************************************************************************
```

Figure 2.1-5 (Cont.)

```
C******************************** READIN ************************************
C*                                                                         *
C*                        SUBROUTINE READIN                                *
C*                                                                         *
C*      PURPOSE                                                            *
C*          READ IN PLANNER DATA. GENERATE MAIN DATA STREAM.               *
C*          DEFINE POINTERS                                                *
C*                                                                         *
C*      USAGE                                                              *
C*          CALL READIN(NSET)                                              *
C*                                                                         *
C*      DESCRIPTION OF PARAMETERS                                          *
C*           * INPUT *                                                     *
C*          NSET - NUMBER OF DATA SETS                                     *
C*       DATA SETS                                                         *
C*     1 ARRAY UGS                                                         *
C*     2 POSITION ERROR PARAMETER SET                                      *
C*     3 SENSORS                                                           *
C*     4 SENSOR DESCRIPTION PARAMETER SET                                  *
C*     5 FIRETRAP KILL POINT SYSTEMS                                       *
C*     6 MONITORS                                                          *
C*     7 MONITOR PARAMETER SET                                             *
C*     8 RELAYS                                                            *
C*     9 RELAY RELIABILITY PARAMETER SET                                   *
C*    10 DATA LINKS                                                        *
C*    11 RECEIVER/TRANSMITTER PARAMETER SET                                *
C*    12 PATH DATA                                                         *
C*    13 FORCE TYPE PARAMETER SET                                          *
C*    14 COVERAGE/SCAN PARAMETER SET                                       *
C*    15 NAVIGATION SYSTEM (HYPERBOLIC)                                    *
C*    16 NAVIGATION SYSTEM (RHO THETA)                                     *
C*    17 NAVIGATION SYSTEM (DOPPLER)                                       *
C*    18 NAVIGATION SYSTEM (NORMALLY DISTRIBUTED ERRORS)                   *
C*    19 STASCAN ARRAYS (RADAR AND VISUAL)                                 *
C*    20 MOVE ARRAYS                                                       *
C*    21 BLUE FORCES                                                       *
C*    22 RED FORCES                                                        *
C*    23 BATTLE      PIEVT   TABLE                                         *
C*    24 BATTLE      RSEVT   TABLE                                         *
C*    25 BATTLE      XCLUA   TABLE                                         *
C*    26 BATTLE      FSPTB   TABLE                                         *
C*    27 CULTURE     PCEVT   TABLE                                         *
C*    28 CULTURE     RCEVT   TABLE                                         *
C*    29 CULTURE   SNFOX-Y TABLE                                           *
C*          MAX        MAXIMUM STORAGE ALLOCATED FOR IDATA                 *
```

Figure 2.1-6

```
C*                                                                          *
C*         * OUTPUT *                                                       *
C*        NDATA   NUMBER OF DATA POINTS                                     *
C*        LDATA(J) POINTER LOCATING SET J                                   *
C*        NDATC(J) NUMBER OF POINTS IN J SET                                *
C*        IDATA( ) MASTER DATA STREAM                                       *
C*                                                                          *
C*    METHOD                                                                *
C*           THE PLANNER INPUT DATA IS READ UNDER CONTROL OF A DO LOOP.*
C*        EACH SET IS PRECEDED BY 3 HEADER CARDS WHICH ARE READ AND      *
C*        PRINTED. COLUMNS 1-28 OF THE FIRST CARD SHOULD BE USED TO      *
C*        IDENTIFY THE SETS. THE FIRST FOUR COLUMNS OF THE SECOND CARD *
C*        ARE READ WITH AN I4 FORMAT AND USED AS THE COUNT OF THE NUMBER *
C*        OF WORDS IN EACH SUBSET.  COLUMNS 5-72 OF THE SECOND CARD AND*
C*        COLUMNS 5-72 OF THE THIRD CARD MUST CONTAIN THE FORMAT FOR     *
C*        THE SUBSET.                                                       *
C*           THE NUMBER OF SUBSETS NEED NOT BE SPECIFIED. THE ROUTINE     *
C*        DETERMINES THE END BY LOOKING FOR A ZERO OR A BLANK IN THE     *
C*        FIRST WORD. THUS A SET OF BLANK CARDS IS INSERTED BY THE USER*
C*        AT THE END OF EACH MAIN DATA SET. IF THE SUBSETS HAVE THREE    *
C*       CARDS THEN THREE BLANKS MUST BE USED, ETC.                         *
C*           ALL HEADER CARDS MUST BE IN THE INPUT DECK. IF IT IS         *
C*        DESIRED TO OMIT A SET-THE HEADER CARDS MUST BE FOLLOWED BY     *
C*        THE PROPER NUMBER OF BLANKS.                                      *
C*           SUBROUTINE CONVRT IS CALLED TO PERFORM THE NECESSARY         *
C*        CONVERSIONS OF THE PLANNED INPUT TO INTERNAL FORMAT.            *
C*           THE POINTERS ARE SET.                                          *
C*           IF THE DATA EXCEEDS THE STORAGE AS SET BY MAX READING IS     *
C*        TERMINATED WITH AN ERROR MESSAGE. THE DIMENSION MUST BE         *
C*        INCREASED OR THE PLANNER INPUT REDUCED.                           *
C*                                                                          *
C*        SUBROUTINES REQUIRED                                              *
C*            CONVRT                                                        *
C*                                                                          *
C***************************************************************************
```

Figure 2.1-6 (Cont.)

```
C******************************** CONVRT ********************************
C*                                                                      *
C*                         SUBROUTINE CONVRT                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*        TO PERFORM NECESSARY CONVERSIONS OF PLANNER INPUT DATA        *
C*                                                                      *
C*    USAGE                                                             *
C*        CALL CONVRT(K,IR,FR)                                          *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*          * INPUT *                                                   *
C*        K   DATA SET NUMBER                                           *
C*        IR-FR  DATA SET  INTEGER-FLOATING                             *
C*     DATA SETS                                                        *
C*   1 ARRAY UGS                                                        *
C*   2 POSITION ERROR PARAMETER SET                                     *
C*   3 SENSORS                                                          *
C*   4 SENSOR DESCRIPTION PARAMETER SET                                 *
C*   5 FIRETRAP KILL POINT SYSTEMS                                      *
C*   6 MONITORS                                                         *
C*   7 MONITOR PARAMETER SET                                            *
C*   8 RELAYS                                                           *
C*   9 RELAY RELIABILITY PARAMETER SET                                  *
C*  10 DATA LINKS                                                       *
C*  11 RECEIVER/TRANSMITTER PARAMETER SET                               *
C*  12 PATH DATA                                                        *
C*  13 FORCE TYPE PARAMETER SET                                         *
C*  14 COVERAGE/SCAN PARAMETER SET                                      *
C*  15 NAVIGATION SYSTEM (HYPERBOLIC)                                   *
C*  16 NAVIGATION SYSTEM (RHO THETA)                                    *
C*  17 NAVIGATION SYSTEM (DOPPLER)                                      *
C*  18 NAVIGATION SYSTEM (NORMALLY DISTRIBUTED ERRORS)                  *
C*  19 STASCAN ARRAYS (RADAR AND VISUAL)                                *
C*  20 MOVE ARRAYS                                                      *
C*  21 BLUE FORCES                                                      *
C*  22 RED FORCES                                                       *
C*  23 BATTLE     PIEVT   TABLE                                         *
C*  24 BATTLE     PSEVT   TABLE                                         *
C*  25 BATTLE     XCLUA   TABLE                                         *
C*  26 BATTLE     FSPTB   TABLE                                         *
C*  27 CULTURE    PCEVT   TABLE                                         *
C*  28 CULTURE    RCEVT   TABLE                                         *
C*  29 CULTURE   SNFDX-Y  TABLE                                         *
C*                                                                      *
```

Figure 2.1-7

```
C*     METHOD                                                              *
C*         TRANSFER TO THE APPROPIATE CONVERSIONS IS DONE BY A            *
C*      COMPUTED GO TO.                                                    *
C*         ALL TIMES ARE CONVERTED TO SECONDS                              *
C*         ABSOLUTE TIMES ARE CONVERTED TO SECONDS SINCE START OF         *
C*      GAME BY A CALL TO SUBROUTINE TIMER.                                *
C*         ALL MAP COORDINATES ARE CONVERTED TO RELATIVE GAME COORD. *
C*         ALL ANGLES ARE CONVERTED TO MATHEMATICAL ANGLES IN RADIANS*
C*         ALL DISTANCES ARE CONVERTED TO METERS                           *
C*         ALPHABETIC INFORMATION IS CONVERTED TO 0 OR 1 WHERE            *
C*      REQUIRED.                                                          *
C*         NUMBERS ARE FIXED OR FLOATED WHERE REQUIRED.                    *
C*         VARIOUS DESIGNER INPUT VALUES ARE SET USING THE PLANNER        *
C*      INPUT CODE AND VALUES SPECIFIED BY DATA STATEMENTS.               *
C*                                                                         *
C*     SUBROUTINES REQUIRED                                                *
C*         TIMER                                                           *
C*                                                                         *
C**************************************************************************
```

Figure 2.1-7 (Cont.)

```
C************************************** TIMER ********************************
C*                                                                           *
C*                          SUBROUTINE TIMER                                 *
C*                                                                           *
C*    PURPOSE                                                                *
C*        TO CONVERT A PLANNER INPUT TIME GIVEN IN DAYS-HOURS-MINUTES        *
C*        (DDHHMM) TO SECONDS SINCE START OF GAME                            *
C*                                                                           *
C*    USAGE                                                                  *
C*        T= TIMER(T)                                                        *
C*                                                                           *
C*                                                                           *
C*    SUBROUTINES REQUIRED                                                   *
C*        NONE                                                               *
C*                                                                           *
C*****************************************************************************
```

Figure 2.1-8

This step (a) computes up-down times of monitors, data links, firetraps, and UGSARRAYS, and stores this information on disc (JFUDT), and (b) creates event types 5, 6, 7, and 8 for MSM and stores them on disc (JFEVT).

The 8 subroutines unique to this job step are described in the following figures:

Figure 2.1-9	PREMN1
2.1-10	UPDN1
2.1-11	UPDN5
2.1-12	UPDN6
2.1-13	UPDN8
2.1-14	UPDN10
2.1-15	READUP
2.1-16	COMMUP

Further descriptions of the up/down simulation are contained in Section 4.2.

2.1.5.3 PRERUN Step 2

Step 2 in PRERUN comprises the main program (PREMN2) with 4 subroutines listed in Table 2.1-V plus 8 utility subroutines.

External data sets required as input are "DATAIN", "JFMSTR", "JFUDT", and "JFEVT", generated in Steps 0 and 1.

This step computes up/down times of those sensors from UGSARRAYS and STASCAN ARRAYS, the primary subroutine being RUSUP. The up/down disk file "JFUDT" is updated and events type 4 (sensor up/down) logic is contained in Section 4.2.

The five subroutines unique to this step are described in the following figures:

Figure 2.1-17	PREMN2
2.1-18	UPDN3
2.1-19	UPDN19
2.1-20	RUSUP
2.1-21	EVNT48

```
C******************************** PREMN1 ********************************
C*                                                                     *
C*              PRE-RUN EXECUTIVE - STEP1                              *
C*                                                                     *
C*        PURPOSE:  THIS ROUTINE INITIATES THE PRE-RUN SEQUENCE        *
C*                  IT INITIATES THE UP-DOWN TIME SEQUENCES AND THE MSM *
C*                      EVENT STREAM                                   *
*      USAGE                                                           *
C*         MAIN PROGRAM                                                *
C*                                                                     *
C*     METHOD                                                          *
C*            THE MASTER DATA STREAM AND THE COMMON GAME INFORMATION   *
C*         ARE READ.  CALLS TO THE UPDN ROUTINES ARE MADE. BEFORE EACH *
C*         CALL THE VARIABLE MVE IS SET TO ZERO TO INITIATE A NEW      *
C*         TEMPORARY EVENT LIST.                                       *
C*            AFTER ALL UPDNS HAVE BEEN CALLED THE DATA STREAM FOR THE *
C*         UP-DOWN TIMES IS COMPRESSED - PRESERVING ONLY THE TIMES FOR *
C*         THE ARRAYS AND THE POINTER KARR.                            *
C*            THE UP-DOWN TIMES ARE RECORDED ON MTAPE(3)-JFUDT         *
C*            THE EVENTS ARE RECORDED ON MTAPE(4)-JFEVT                *
C*                                                                     *
C*        SUBROUTINES CALLED:                                          *
C*                       TIMER - CONVERTS DAY-HOUR-MIN  TO SECONDS SINCE *
C*                               START OF GAME                         *
C*                       UPDN8   PLAYS RELAYS THRU COMMUP              *
C*                       UPDN5   CREATES MSM EVENT 7 FOR FIRETRAPS     *
C*                       UPDN6   PLAYS MONITORS THRU READUP  -MSM EVENT 5 *
C*                       UPDN10  UP-DOWN DATALINKS- USES RESULTS OF UPDN8 *
C*                                                          -MSM EVENT 6 *
C*                       UPDN1   UP-DOWN OF ARRAY UGS -MONITOR-DATALINK *
C*                               USES RESULTS OF UPDN6,UPDN10,         *
C*                                                          -MSM EVENT 8 *
C*                                                                     *
C*        SUBROUTINES REQUIRED OTHER THAN THOSE DIRECTLY CALLED        *
C*                       COMMUP                                        *
C*                       READUP                                        *
C*                       ERASE                                         *
C*                       FINDX                                         *
C*                       MERGDR                                        *
C*                       DORDER                                        *
C*                       GMERGE                                        *
C*                       FINDY                                         *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-9

```
C******************************** UPDN1 ********************************
C*                                                                     *
C*                          SUBROUTINE UPDN1                           *
C*                                                                     *
C*    PURPOSE                                                          *
C*        THIS ROUTINE COMPUTES THE UP-DOWN TIMES OF THE UGS ARRAYS    *
C*        DUE TO THE AND-OR COMBINATIONS OF THE UP-DOWN TIMES OF       *
C*        THE MONITORS AND THE DATA LINKS                              *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL UPDN1(NPLAY)                                            *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*            * INPUT *                                                *
C*                                            PLANNER TABLE  ITEM      *
C*        TT(1)  PLANNED UP TIME                      1        9       *
C*        TT(2)  PLANNED DOWN TIME                    1       10       *
C*                                                                     *
C*            * OUTPUT *                                               *
C*        TT( )  ACTUAL UP-DOWN TIMES                                  *
C*        MR     NUMBER OF TIMES                                       *
C*        KARR   POINTER FOR ARRAY TIMES IN STRING                     *
C*        UDTM   UP-DOWN STRING                                        *
C*        NPLAY  PLAY OPTION                                           *
C*                                                                     *
C*    METHOD                                                           *
C*            . IE PLANNER INPUT DATA FOR UGS ARRAYS, MONITORS AND DATA *
C*        LINKS ARE LOCATED BY CALLING FINDX.                          *
C*            THE PLAY OPTION IS CHECKED.                              *
C*            THE UP-DOWN TIMES FOR THE MONITORS AND THE DATA LINKS ARE *
C*        LOCATED IN THE UP-DOWN TIME DATA STREAM UDTM BY CALLS TO     *
C*        FINDY. THE NECESSARY AND/OR COMBINATIONS ARE COMPUTED.       *
C*                                                                     *
C*            MSM EVENT TYPE 8 IS CREATED AND THE UP-DOWN TIMES ARE    *
C*        PUT INTO UDTM AND POINTER KARR IS SET.                       *
C*            AFTER ALL ARRAYS HAVE BEEN PROCESSED MERGOR IS CALLED    *
C*        TO ORDER AND MERGE THE EVENTS INTO THE MASTER LIST.          *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*        FINDX                                                        *
C*        FINDY                                                        *
C*        MERGOR                                                       *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-10

```
C******************************* UPDN5  ******************************
C*                                                                   *
C*                          SUBROUTINE UPDN5                         *
C*                                                                   *
C*    PURPOSE                                                        *
C*        THIS ROUTINE CREATES MSM EVENT TYPE 7, THE UP DOWN TIMES   *
C*        OF THE FIRETRAPS                                           *
C*                                                                   *
C*    CALLIN  SEQUENCE                                               *
C*        CALL UPDN5(NN)                                             *
C*                                                                   *
C*    DESCRIPTION OF PARAMETERS                                      *
C*            * INPUT *                                              *
C*                                          PLANNER TABLE  ITEM      *
C*        FD(N+2)  PLANNED UP TIME                  5        3       *
C*        FD(N+3)  PLANNED DOWN TIME                5        4       *
C*        ID(N)    IDENTITY                         5        1       *
C*        NN       DUMMY                                             *
C*                                                                   *
C*    METHOD                                                         *
C*            THE PLANNER INPUT FOR FIRETRAPS IS LOCATED BY FINDY.   *
C*            THE OPTION TO PLAY IS CHECKED AND MSM EVENT TYPE 7 IS  *
C*        GENERATED. MERGDR IS CALLED TO ORDER AND MERGE THE EVENTS  *
C*                                                                   *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*        FINDX                                                      *
C*        MERGDR                                                     *
C*                                                                   *
C*********************************************************************
```

Figure 2.1-11

```
C******************************** UPDN6  ********************************
C*                                                                      *
C*                          SUBROUTINE UPDN6                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*       THIS ROUTINE COMPUTES THE UP-DOWN TIMES OF THE MONITORS        *
C*       USING SUBROUTINE READUP                                        *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*       CALL UPDN6(MPRINT,NPLAY)                                       *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*          * INPUT *                                                   *
C*                                              PLANNER TABLE  ITEM     *
C*       A     MEAN TIME BETWEEN FAILURES               7        3      *
C*       B     MEAN TIME TO REPAIR                      7        4      *
C*       C     STANDARD DEVIATION OF REPAIR TIME        7        5      *
C*       T(1)    PLANNED UP TIME                        6        6      *
C*       T(2)    PLANNED DOWN TIME                      6        7      *
C*       MPRINT PRINT OPTION                                            *
C*       NPLAY  PLAY OPTION                                             *
C*                                                                      *
C*         * OUTPUT *                                                   *
C*       TT( )  ACTUAL UP-DOWN TIMES                                    *
C*       MM      NUMBER OF TIMES IN TT                                  *
C*                                                                      *
C*    METHOD                                                            *
C*          THE PLANNER INPUT DATA FOR THE MONITORS AND THE MONITOR     *
C*       PARAMETER SETS IS LOCATED BY USE OF SUBROUTINE FINDX.          *
C*          THE OPTIONS TO PLAY PLANNED UP DOWN TIMES ARE CHECKED.      *
C*       SUBROUTINE READUP IS CALLED IF REQUIRED. THE PRINT OPTION      *
C*       MPRINT IS CHECKED AND MSM EVENT TYPE 5 CREATED.                *
C*          AFTER ALL MONITORS HAVE BEEN PROCESSED THE EVENTS ARE       *
C*       ORDERED AND MERGED BY A CALL TO SUBROUTINE MERGOR.             *
C*          THE UPDOWN TIMES ARE PLACED IN THE DATA STREAM UDTM         *
C*      AND THE POINTER KMUD IS SET.                                    *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*       FINDX                                                          *
C*       READUP                                                         *
C*       MERGOR                                                         *
C*                                                                      *
C************************************************************************
```

Figure 2.1-12

```
C******************************** UPDN8 *********************************
C*                                                                      *
C*                       SUBROUTINE UPDN8                               *
C*                                                                      *
C*    PURPOSE                                                           *
C*        THIS ROUTINE COMPUTES THE UP-DOWN TIMES OF THE RELAYS         *
C*        USING  SUBROUTINE COMMUP                                      *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL UPDN8(MPRINT,NPLAY)                                      *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*           * INPUT *                                                  *
C*                                              PLANNER TABLE  ITEM     *
C*        TT(1)    PLANNED UP TIME                      8       7       *
C*        TT(2)    PLANNED DOWN TIME                    8       8       *
C*        M        NUMBER OF TIMES                                      *
C*        TSD      SELF DESTRUCT                       8(9)   16(6)     *
C*        MPRINT   PRINT OPTION (SET IN PRERUN STEP Ø)                  *
C*        NPLAY    PLAY OPTION  (SET IN PRERUN STEP Ø)                  *
C*                                                                      *
C*          * OUTPUT *                                                  *
C*        TT( )    UP DOWN TIMES                                        *
C*        M        NUMBER OF TIMES                                      *
C*        LL( )    CODE IDENTIFYING TIMES (SEE COMMUP)                  *
C*        UDTM( )  UP-DOWN STRING                                       *
C*        KREL( )  POINTER FOR RELAYS IN UDTM                           *
C*                                                                      *
C*    METHOD                                                            *
C*           THE PLANNER INPUT FOR THE RELAYS AND THE PARAMETER SETS    *
C*        ARE FOUND BY CALLING SUBROUTINE FINDX.                        *
C*           THE PLAY AND PRINT OPTIONS ARE CHECKED.                    *
C*           SUBROUTINE COMMUP IS CALLED WHEN NECESSARY TO COMPUTE      *
C*        THE UP DOWN TIMES.                                            *
C*           THE UP-DOWN TIMES ARE PLACED IN THE DATA STREAM UDTM AND   *
C*        THE POINTER KREL IS SET.                                      *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*        FINDX                                                         *
C*        COMMUP                                                        *
C*                                                                      *
C************************************************************************
```

Figure 2.1-13

```
C******************************** UPDN10 ********************************
C*                                                                       *
C*                         SUBROUTINE UPDN10                             *
C*                                                                       *
C*    PURPOSE                                                            *
C*        THIS ROUTINE COMPUTES THE UP-DOWN TIMES OF THE DATA LINKS      *
C*        USING THE RELAY UP-DOWN TIMES IF REQUIRED                      *
C*                                                                       *
C*    CALLING SEQUENCE                                                   *
C*        CALL UPDN10(NPLAY)                                             *
C*                                                                       *
C*    DESCRIPTION OF PARAMETERS                                          *
C*                                               PLANNER TABLE  ITEM     *
C*        NR      NUMBER OF RELAYS                        10     4       *
C*        M1      IDENTITY OF FIRST RELAY                 10     5       *
C*        M2      IDENTITY OF SECOND RELAY                10     6       *
C*        NPLAY   PLAY OPTION (SET IN PRERUN STEP Ø)                     *
C*                                                                       *
C*    METHOD                                                             *
C*            THE LOCATION OF THE PLANNER INPUT DATA FOR THE RELAYS      *
C*        AND THE DATA LINKS IS FOUND BY CALLS TO FINDX. THE UP-DOWN     *
C*        TIMES FOR THE RELAYS ARE LOCATED IN UDTM BY CALLING FINDY.     *
C*        IF MORE THAN ONE RELAY IS SPECIFIED THE ' AND ' OF THE TIMES   *
C*        IS COMPUTED. THE TIMES ARE PLACED IN THE STRING UDTM AND       *
C*        THE POINTER KDLK IS SET. MSM EVENT TYPE 6 IS GENERATED.        *
C*        AFTER ALL OF THE DATA LINKS HAVE BEEN PROCESSED MERGOR IS      *
C*        CALLED TO ORDER AND MERGE THE EVENTS WITH THE MASTER LIST.     *
C*                                                                       *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                       *
C*        FINDX                                                          *
C*        FINDY                                                          *
C*        MERGOR                                                         *
C*                                                                       *
C*************************************************************************
```

Figure 2.1-14

```
C******************************* READUP **********************************
C*                                                                       *
C*                                                                       *
C*                         SUBROUTINE READUP                             *
C*    PURPOSE                                                            *
C*        THIS ROUTINE COMPUTES THE FAILURE AND REPAIR TIMES             *
C*        AND DETERMINES THE TRUE UP DOWN TIMES OF A  DEVICE WITH        *
C*        A PLANNED UP-DOWN SEQUENCE                                     *
C*                                                                       *
C*    CALLING SEQUENCE                                                   *
C*        CALL READUP(S,NK,FMUF,FMUR,SIGR,T,N)                           *
C*                                                                       *
C*    DESCRIPTION OF PARAMETERS *                                        *
C*            * INPUT *                                                  *
C*        S( )    PLANNED UP-DOWN TIMES                                  *
C*        NK      NUMBER OF PLANNED UP DOWN TIMES                        *
C*        FMUF    MEAN TIME BETWEEN FAILURES                             *
C*        FMUR    MEAN TIME TO REPAIR                                    *
C*        SIGR    STANDARD DEVIATION OF REPAIR TIME                      *
C*                                                                       *
C*            * OUTPUT *                                                 *
C*        T( )    ACTUAL UP-DOWN TIMES                                   *
C*        N       NUMBER OF ACTUAL UP-DOWN TIMES                         *
C*                                                                       *
C*    REMARKS                                                            *
C*        RELIABILITY LIFE IS ACCUMULATED DURING ON TIME ONLY            *
C*                                                                       *
C*    METHOD                                                             *
C*            THE MEAN TIME BETWEEN FAILURE IS COMPUTED FROM A POISSON   *
C*        DISTRIBUTION AND THE REPAIR TIME IS FOUND FROM A GAUSSIAN      *
C*        THESE TIMES ARE COMPARED TO THE PLANNED UP-DOWN TIMES TO       *
C*        DETERMINE THE ACTUAL TIME HISTORY.                             *
C*                                                                       *
C*    SUBROUTINES REQUIRED                                               *
C*        NONE                                                           *
C*                                                                       *
C*************************************************************************
```

*See UPDN 6, Pg. 2-33

Figure 2.1-15

```
C******************************** COMMUP *********************************
C*                                                                      *
C*                         SUBROUTINE COMMUP                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*        THIS ROUTINE COMPUTES, FOR EACH REMOTE UNATTENDED SENSOR DATA *
C*        LINK RELAY, THE GAME TIMES DURING WHICH THE RELAY IS EMPLACED AND *
C*        OPERABLE.                                                     *
C*    CALLING SEQUENCE                                                  *
C*        CALL COMMUP(XMUE,XMUF,SIGE,XMUR,SIGR,XMUL,SIGL,XMUM,SIGM,NMC,*
C*   *                  TMAX,PA,ISD,IL,IR,M,K,T,N)                      *
C*                                                                      *
C*  INPUTS                                    SOURCE TABLE  ITEM        *
C*    XMUE- PLANNED UP TIME                   8 -RELAYS       7         *
C*    SIGE- ST. DEV. OF UP TIME               8 -RELAYS       9         *
C*    TMAX- PLANNED DOWN TIME                 8 -RELAYS       8         *
C*    XMUF- MTBF                              9 -REL.RELB.    3         *
C*    XMUL- AVERAGE BATTERY LIFE              9 -REL.RELB.    4         *
C*    SIGL- ST. DEV. OF BAT. LIFE             9 -REL.RELB.    5         *
C*    XMUM- AVERAGE MAINT. INTERVAL           8 -RELAYS      11         *
C*    SIGM- ST. DEV. OF MAINT                 8 -RELAYS      12         *
C*    XMUR- AVERAGE REEMPLACEMENT TIME        8 -RELAYS      13         *
C*    SIGR- ST. DEV. OF REEMPLACEMENT TIME    8 -RELAYS      14         *
C*    NMC - MAX # OF REEMPLACEMENT MISSIONS   8 -RELAYS      15         *
C*    PA  - PROB. OF ABORT                    8 -RELAYS      10         *
C*    ISD - SELF DESTRUCT CAPAB. 1-YES 0-NO   9 -REL.RELB.    6         *
C*        - SHALL SELF DEST. USED1-YES 0-NO   8 -RELAYS      16         *
C*        ISD IS.AND. OF THESE -DONE BY CALLING PROGRAM                 *
C*  OUTPUTS                                                             *
C*    T(I),I=1,M  TIME HISTORY                                          *
C*    K(I),I=1,M  CODE    -1 REEMPLACE UP                               *
C*                        -2  NORMAL    UP                              *
C*                        -3  ABORT                                     *
C*                        -4  RELIABILITY FAILURE                       *
C*                        -5 LIFE FAILURE                               *
C*                        -6  NORMAL DOWN                               *
C*    N   NUMBER OF REEMPLACEMENT MISSIONS TRIED                        *
C*                                                                      *
C*        REMARKS                                                       *
C*    MAINTENANCE ASSUMED ONLY TO REPLACE BATTERIES                     *
C*    ROUTINE COUNTS RELIABILITY FAILURE ONLY IF TURNED ON              *
C*                                                                      *
C*                                                                      *
C*    SUBROUTINES REQUIRED                                              *
C*        NONE                                                          *
C*                                                                      *
C*                                                                     **
```

Figure 2.1-16

```
C*                                                                         *
C*                                                                         *
C*   METHOD                                                                *
C*                                                                         *
C*         THE FIRST STEP IS TO COMPUTE THE INITIAL RELAY EMPLACEMENT TIME.  *
C*      IF THIS INITIAL RELAY EMPLACEMENT TIME IS GREATER THAN THE PLANNED  *
C*      DOWN TIME, ADDITIONAL REEMPLACEMENT MISSIONS ARE UNDERTAKEN PROVIDED *
C*      SUCH MISSIONS REMAIN.  IF THE INITIAL EMPLACEMENT TIME IS LESS THAN *
C*      THE PLANNED DOWN TIME, A CHECK IS MADE FOR MISSION ABORT BY COMPARING *
C*      THE MISSION ABORT PROBABILITY (PA) WITH A UNIFORM RANDOM NUMBER.  IF *
C*      THE MISSION ABORTS AND NO MISSIONS REMAIN, THE RELAY REMAINS DOWN.  *
C*      HOWEVER, IF THE MISSION ABORTS AND MORE MISSIONS REMAIN, A NEW EMPLACE- *
C*      MENT TIME IS DETERMINED.                                           *
C*                                                                         *
C*      IF THE MISSION DOES NOT ABORT, A TIME TO FAILURE IS DETERMINED FOR . *
C*      THE RELAY.  THIS INVOLVES THE DETERMINATION OF TIMES TO RELIABILITY *
C*      FAILURE AND LIFE FAILURE AND A TIME TO MAINTENANCE CHECKS.  IF THE  *
C*      RELAY DOES NOT FAIL BEFORE ITS PLANNED DOWN TIME, THE PROCESSING THROUGH*
C*      COMMUP IS COMPLETE.  HOWEVER, THE RELAY CAN FAIL DUE TO EITHER A LIFE *
C*      OF RELIABILITY FAILURE (MUST OCCUR PRIOR TO PLANNED DOWN TIME), AND *
C*      THAN REEMPLACEMENT WILL TAKE PLACE PROVIDED ADDITIONAL MISSIONS REMAIN. *
C***************************************************************************
```

```
C******************************* PREMN2 **********************************
C*                                                                        *
C*             PRE-RUN EXECUTIVE--  STEP 2                                *
C*                                                                        *
C*    PURPOSE                                                             *
C*        CALLS SUB. TO PLAY ARRAY UGS, AND STASCAN THRU RUSUP            *
C*                                                                        *
C*    USAGE                                                               *
C*        MAIN PROGRAM                                                    *
C*                                                                        *
C*    REMARKS                                                             *
C*            THE EVENTS MAY BE WRITTEN AS A SEPARATE RECORD. TO          *
C*        ACCOMPLISH THIS ONE NEED ONLY REMOVE THE REWIND M4, REMOVE      *
C*        THE DO 50 LOOP AND REMOVE THE CALL TO GMERGE FROM THE           *
C*        PROGRAM.                                                        *
C*                                                                        *
C*       SUBROUTINES CALLED                                               *
C*                    UPDN3     PLAYS UGS THRU RUSUP    MSM-EVENT 4       *
C*                    UPDN19    PLAYS STASCAN THRU RUSUP  MSM-EVENT 4     *
C*                                                                        *
C*       SUBROUTINES REQUIRED                                             *
C*                    ERASE                                               *
C*                    FINDX                                               *
C*                    MERGOR                                              *
C*                    DORDER                                              *
C*                    GMERGE                                              *
C*                    FINDY                                               *
C*                    RUSUP                                               *
C*                    EVNT48                                              *
C*                                                                        *
C*    METHOD                                                              *
C*            THE COMMON GAME INFORMATION IS READ FROM JFMSTR, THE        *
C*        PLANNER INPUT FROM DATAIN AND THE UPDOWN TIMES FROM JFUDT.      *
C*        SUBROUTINES UPDN3 AND UPDN19 ARE CALLED. IF MPRINT(30)          *
C*        IS NOT EQUAL TO ZERO THE UPDOWN TIMES FOR THE SENSORS IS        *
C*        PRINTED. THE UPDOWN TIMES ARE RECORDED ON JFUDT. THE EVENTS     *
C*        ARE READ FROM JFEVT AND MERGED WITH THE EVENTS GENERATED BY     *
C*        UPDN3 AND UPDN19. THE MERGED LIST IS WRITTEN ON JFEVT           *
C*                                                                        *
C**************************************************************************
```

Figure 2.1-17

```
C****************************** UPDN3 ***********************************
C*                                                                      *
C*                          SUBROUTINE UPDN3                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*        THIS ROUTINE PLAYS THE SENSORS ASSOCIATED WITH THE            *
C*        UG ARRAYS THROUGH SUBROUTINE RUSUP                            *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL UPDN3(MPRINT,NPLAY)                                      *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*           * INPUT *                                                  *
C*                                               PLANNER TABLE  ITEM    *
C*        N       NUMBER  OF SENSORS                      1       4     *
C*        XMUF    PLANNED UP TIME                         1       9     *
C*        SIGF    STANDARD DEVIATION OF UP TIME           1       28    *
C*        PA      PROBABILITY OF ABORT                    1       23    *
C*        ICR     CRITERIA LEVEL                          1       26    *
C*        NMC     NUMBER OF MISSIONS                      1       25    *
C*        XMUR    REEMPLACEMENT TIME                      1       29    *
C*        SIGR    STANDARD DEVIATION OF REEMPLACEMENT TIME 1      30    *
C*        XMUM    MAINTENANCE INTERVAL                    1       31    *
C*        SIGM    STANDARD DEVIATION OF MAINTENANCE       1       32    *
C*        NPC     NUMBER OF ATTEMPTS PER MISSION          1       24    *
C*        MPL     MODE OF EMPLACEMENT                     1       11    *
C*        TX      PLANNED DOWN TIME                       1       10    *
C*         ** SENIN TABLE **                                            *
C*                                                                      *
C*        1            MEAN BATTERY LIFE                  4       5     *
C*        2            STANDARD DEVIATION OF BATTERY LIFE 4       6     *
C*        3            MEAN TIME BETWEEN FAILURES         4       4     *
C*        4            SELF DESTRUCT                      1(4)    27(7) *
C*        5            AUX OR PRI                         4       8     *
C*        6            PROBABILITY OF SURVIVAL            4       9     *
C*        MPRINT PRINT OPTION                                           *
C*        NPLAY  PLAY OPTION                                            *
C*                                                                      *
C*         * OUTPUT *                                                   *
C*        UDTM  UP-DOWN TIMES                                           *
C*        KSDN  POINTER FOR SENSORS IN UDTM                             *
C*        IEV   MASTER EVENT LIST                                       *
C*                                                                      *
C*    SUBROUTINES REQUIRED                                              *
C*        FINDX                                                         *
C*        RUSUP                                                         *
C*        MERGDR                                                        *
C*        EVNT43                                                        *
```

Figure 2.1-18

```
C*                                                                      *
C*    METHOD                                                            *
C*          THE PLANNER INPUT DATA FOR THE   UGS   ARRAYS, THE SENSORS*
C*       AND THE SENSOR PARAMETERS ARE LOCATED BY CALLING FINDX. THE  *
C*       PLAY OPTION IS CHECKED. THE SENIN TABLE IS SET AND RUSUP IS   *
C*       CALLED. AFTER THE CALL THE PRINT OPTION IS CHECKED, AND THE  *
C*       RESULTS OF RUSUP ARE PRINTED IF DESIRED. EVNT48 IS CALLED TO *
C*       DECODE THE RESULTS OF RUSUP AND SET THE UPDOWN TIMES AND MSM *
C*       EVENT TYPE 4. AFTER ALL ARRAYS HAVE BEEN PROCESSED MERGOR    *
C*       IS CALLED TO ORDER AND MERGE THE EVENTS WITH THE MASTER LIST *
C*                                                                      *
C***********************************************************************
```

Figure 2.1-18 (Cont.)

```
C****************************** UPDN19 **********************************
C*                                                                      *
C*                      SUBROUTINE UPDN19                               *
C*                                                                      *
C*    PURPOSE                                                           *
C*       THIS ROUTINE PLAYS THE SENSORS ASSOCIATED WITH THE             *
C*       STASCAN ARRAYS THROUGH SUBROUTINE RUSUP                        *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*       CALL UPDN19(MPRINT,NPLAY)                                      *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*          * INPUT *                                                   *
C*                                                PLANNER TABLE  ITEM   *
C*       N      NUMBER OF SENSORS                         19       4    *
C*       XMUE   PLANNED UP TIME                           19       9    *
C*       SIGE   STANDARD DEVIATION OF UP TIME             19      15    *
C*       PA     PROBABILITY OF ABORT          =0                        *
C*       ICR    CRITERIA LEVEL                =1                        *
C*       NMC    NUMBER OF MISSIONS            =1                        *
C*       XMUR   REEMPLACEMENT TIME                        19      16    *
C*       SIGR   STANDARD DEVIATION OF REEMPLACEMENT TIME  19      17    *
C*       XMUM   MAINTENANCE INTERVAL                      19      18    *
C*       SIGM   STANDARD DEVIATION OF MAINTENANCE         19      19    *
C*       NPC    NUMBER OF ATTEMPTS PER MISSION  =1                      *
C*       MPL    MODE OF EMPLACEMENT                       19      11    *
C*       TX     PLANNED DOWN TIME                         19      10    *
C*       1            MEAN BATTERY LIFE                   4        5    *
C*       2            STANDARD DEVIATION OF BATTERY LIFE  4        6    *
C*       3            MEAN TIME BETWEEN FAILURES          4        4    *
C*       4            SELF DESTRUCT              =0                     *
C*       5            AUX OR PRI                          4        8    *
C*       6            PROBABILITY OF SURVIVAL             4        9    *
C*       MPRINT  PRINT OPTION(SET IN PRERUN STEP Ø)                     *
C*       NPLAY   PLAY OPTION(SET IN PRERUN STEP Ø)                      *
C*                                                                      *
C*          * OUTPUT *                                                  *
C*       UDTM   UP-DOWN TIMES                                           *
C*       KSDN   POINTER FOR SENSORS IN UDTM                             *
C*       IEV    MASTER EVENT LIST                                       *
C*                                                                      *
C*    SUBROUTINES REQUIRED                                              *
C*       FINDX                                                          *
C*       RUSUP                                                          *
C*       EVNT48                                                         *
C*       MERGDR                                                         *
```

Figure 2.1-19

```
C*                                                                        *
C*    METHOD                                                              *
C*         THE PLANNER INPUT DATA FOR THE STASCAN ARRAYS, THE SENSORS*
C*       AND THE SENSOR PARAMETERS ARE LOCATED BY CALLING FINDX. THE   *
C*       PLAY OPTION IS CHECKED. THE SENIN TABLE IS SET AND RUSUP IS   *
C*       CALLED. AFTER THE CALL THE PRINT OPTION IS CHECKED, AND THE   *
C*       RESULTS OF RUSUP ARE PRINTED IF DESIRED. EVNT48 IS CALLED TO *
C*       DECODE THE RESULTS OF RUSUP AND SET THE UPDOWN TIMES AND MSM *
C*       EVENT TYPE 4. AFTER ALL ARRAYS HAVE BEEN PROCESSED MERGDR    *
C*       IS CALLED TO ORDER AND MERGE THE EVENTS WITH THE MASTER LIST *
C*                                                                        *
C*************************************************************************
```

Figure 2.1-19 (Cont.)

```
C*    METHOD                                                                  *
C*          INITIALLY, EMPLACEMENT TIME (TE) FOR EACH SENSOR IS SET TO        *
C*       PLANNED UP TIME (XMUE) PLUS THE STANDARD DEVIATION OF THE EMPLACE-   *
C*       MENT TIME (SIGE) MULTIPLIED BY A NORMAL RANDOM NUMBER.  THE EM-      *
C*       PLACEMENT TIME MUST BE GREATER THAN HALF THE PLANNED EMPLACEMENT     *
C*       TIME.                                                                *
C*          A MATRIX (LR) IS INITIALIZED TO ZERO (DOWN).  LR WILL CONTAIN     *
C*       A RECORD OF WHETHER EACH SENSOR IS UP OR DOWN.                       *
C*          THE INITIAL TIME FOR EACH EMPLACEMENT MISSION IS CHECKED AS       *
C*       FOLLOWS:  IF THE EMPLACEMENT TIME IS GREATER THAN THE PLANNED DOWN   *
C*       TIME, THE TIMES IN THE TT TABLE FOR ALL SENSORS IN THAT ARRAY ARE    *
C*       SET TO THE COMPUTED EMPLACEMENT TIME AND THE ICODE IS SET TO -18.    *
C*       A SECOND TIME (DOWN TIME) IS CALCULATED BY ADDING 1. TO THE OLD      *
C*       EMPLACEMENT TIME AND THE ROUTINE IS EXITED.                          *
C*          IF THE EMPLACEMENT TIME IS NOT GREATER THAN THE PLANNED DOWN      *
C*       TIME, A CHECK IS MADE TO SEE HOW MANY MISSIONS ARE REQUIRED TO GET   *
C*       THE REQUIRED NUMBER OF SENSORS EMPLACED.  A UNIFORM RANDOM NUMBER    *
C*       IS COMPARED WITH THE PROBABILITY OF MISSION ABORT FROM PLANNER IN-   *
C*       PUT.  IF THE MISSION IS ABORTED, ICODE IS SET TO -13, AND THE TIME   *
C*       FOR REEMPLACEMENT TO OCCUR IS CALCULATED USING SIGR AND A NORMAL     *
C*       RANDOM NUMBER.  THIS ASSUMES A NEW MISSION.  BASED ON THIS NEW       *
C*       MISSION TIME, A TIME FOR THE SENSOR TO GO UP IS SET.  THE COUNT IS   *
C*       THEN SET FOR THE ARRAY ID AND THE TT AND LL TABLES ARE SET FOR ALL   *
C*       OF THE SENSORS IN THIS ARRAY.  IF THE EQUIPMENT IS HAND EMPLACED,    *
C*       A CHECK IS MADE TO SEE IF THE END OF THE GAME OCCURS PRIOR TO THE    *
C*       EQUIPMENT BEING REPAIRED.  IF SO, THE ROUTINE IS EXITED.  IF NOT,    *
C*       A NEW UP TIME IS CALCULATED.  A MISSION TO REPAIR/REPLACE THE        *
C*       SENSORS MAY OR MAY NOT OCCUR.                                        *
C*          IF THE MISSION IS NOT ABORTED, THEN A MISSION WILL BE STARTED     *
C*       WITH AS MANY ATTEMPTS AS ARE ALLOWED (NPC) TO EMPLACE THE SENSORS.   *
C*       THE LR TABLE IS CHECKED TO SEE HOW MANY SENSORS ARE UP.  AS ADDI-    *
C*       TIONAL SENSORS ARE EMPLACED, PROBABILITY OF SURVIVAL OF EACH SEN-    *
C*       SOR IS COMPARED WITH A NORMAL RANDOM NUMBER AND SENSOR STATUS IS     *
C*       UPDATED.                                                             *
C*          IF THE SENSOR IS A PRIMARY SENSOR, THEN THE ASSOCIATED AUXIL-     *
C*       IARY GOES DOWN WITH THE PRIMARY.  THE DOWN TIME OF EACH SENSOR IS    *
C*       CALCULATED AS THE MINIMUM OF:  TIME TO SENSOR FAILURE, BATTERY       *
C*       FAILURE, SCHEDULED MAINTENANCE, OR END OF GAME.  FOLLOWING EQUIP-    *
C*       MENT FAILURE, THE NUMBER OF OPERATIONAL SENSORS IS CHECKED TO SEE    *
C*       IF THE ARRAY MUST BE REPLACED OR ANOTHER REEMPLACEMENT ATTEMPT       *
C*       MUST BE MADE.                                                        *
C*                                                                            *
C*    REMARKS                                                                 *
C*  RUSUP LIMITED BY DIMENSION STATEMENTS ONLY,                               *
C*    SUBROUTINES REQUIRED                                                    *
C*       DORDER                                                               *
C*                                                                            *
C*****************************************************************************
```

Figure 2.1-20 (Cont.)

```
C******************************* EVNT48 *********************************
C*                                                                      *
C*                        SUBROUTINE EVNT48                             *
C*                                                                      *
C*    PURPOSE                                                           *
C*        THIS ROUTINE IS USED BY UPDN3 AND UPDN19 TO DECODE            *
C*        THE OUTPUT OF THE RUSUP ROUTINE AND GENERATE MSM EVENT        *
C*        4 AND THE UP-DOWN SEQUENCE OF THE SENSORS                     *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL EVNT48(MVE,IVE,ID,N,KNT,LL,TT,UDTM,NT,KSEN)              *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*        MVE     NUMBER OF WORDS IN EVENT LIST                         *
C*        IVE( )  EVENT LIST                                            *
C*        ID      IDENTITY OF FIRST SENSOR                              *
C*        N       NUMBER OF SENSORS                                     *
C*        KNT     NUMBER OF TIMES ASSOCIATED WITH EACH SENSOR           *
C*        LL( )   CODE IDENTIFYING TIMES (SEE RUSUP)                    *
C*        TT( )   ARRAY OF TIMES                                        *
C*        UDTM( )  MASTER TIME STRING                                   *
C*        NT      NUMBER OF WORDS IN UDTM                               *
C*        KSEN( )  POINTER FOR SENSORS IN UDTM                          *
C*                                                                      *
C*    REMARKS                                                           *
C*        NONE                                                          *
C*                                                                      *
C*    SUBROUTINES REQUIRED                                              *
C*        NONE                                                          *
C*                                                                      *
C*    METHOD                                                            *
C*            THE RUSUP CODE IN ARRAY LL IS CHECKED TO IDENTIFY THE     *
C*        TIME. ONLY TRUE UPDOWN TIMES ARE PLACED IN UDTM, THE POINTER *
C*        IS SET AND EVENT TYPE 4 IS GENERATED.                         *
C*                                                                      *
C************************************************************************
```

Figure 2.1-21

2.1.5.4 PRERUN Step 3

Step 3 in PRERUN comprises the main program (PREMN3) with 9 subroutines listed in Table 2.1-V, plus 7 utility subroutines.

External data sets required as input are "DATAIN", "JFMSTR", "JFUDT", and "JFEVT" generated and updated in Steps 0, 1, and 2.

Designer input values for SNPGT routine are set by BLOCK DATA (JFBLK3). See volume II, Appendix I.

This step computes the ground truth positions for sensors within ARRAYUGS and STASCAN arrays using SNPGT subroutine. These ground truth positions are stored on disc (JFPXY). In this step also, data set "JFEVT" is updated by inclusion of type 10 events (Sensor Reposition). Section 4 contains additional information on logic used.

The Step 3 subroutines are described in Figures 2.1-22 - 2.1-31.

2.1.5.5 PRERUN Step 4

Step 4 in PRERUN comprises the main program (PREMN4) with 1 of the subroutines listed in Table 2.1-V, together with use of 5 of the subroutines from Step 8 and 6 utility subroutines.

External data sets required as input are "DATAIN", "JFMSTR", "JFUDT", "JFEVT", "JFPXY", and "MASSDAT" generated and updated in previous steps.

Designer input values for MVS routine are arrays PRNV1, PRNV2, PRNV3, PRNV4 (nominal navigation system errors). These values are the same as those in BLOCK DATA (JFBLK3) and in the previous step. Any changes to these values must be made in both steps.

Figures 2.1-32 and 2.1-33 describe the subroutines unique to this step.

This step computes ground paths for moving arrays (MOVARRAY), up/down times for the associated sensors, and defines the moving platforms as targets. These targets are stored on disk (JFTAR). Events type 4 (Sensor Up/Down) and events type 8 (Array Up/Down) are added to the disc event file "JFEVT". Section 4 of this volume has additional material on logic involved.

2.1.5.6 PRERUN Step 5

Step 5 in PRERUN comprises the main program (TRNPAR) with 3 subroutines listed in Table 2.1-V plus 5 utility subroutines.

```
C****************************** PREMN3 *********************************
C*                                                                     *
C*              PRE-RUN EXECUTIVE - STEP 3                             *
C*                                                                     *
C*      PURPOSE                                                        *
C*              CALLS SUBS TO PLAY UGS AND STASCAN THRU SNPGT          *
C*                                                                     *
C*   USAGE                                                             *
C*      MAIN PROGRAM                                                   *
C*                                                                     *
C*   METHOD                                                            *
C*         THE COMMON GAME INFORMATION, THE MASTER DATA STREAM, THE    *
C*      UPDOWN TIMES AND THE MASTER EVENT LIST ARE READ FROM THE       *
C*      DATA SETS. THE POINTER ARRAY KSSN IS SET TO ZERO.              *
C*      SUBROUTINE PSNP AND PSNP19 ARE CALLED TO DETERMINE GROUND      *
C*      TRUTH POSITIONS.                                               *
C*         THE GROUND TRUTH POSITIONS ARE PRINTED-SUBROUTINE FINDY     *
C*      IS USED TO LOCATE THE DATA FOR THE INDIVIDUAL SENSORS.         *
C*      SXYTT AND THE POINTER KSSN ARE WRITTEN ON MTAPE(5)- JFPXY      *
C*      THE UPDATED EVENT LIST IS WRITTEN ON MTAPE(4)-JFEVT            *
C*                                                                     *
C*     SUBROUTINES CALLED                                              *
C*                 ERASE                                               *
C*                 PSNP - PLAYS UGS THRU SNPGT       MSM EVENT 10      *
C*                 PSNP19-PLAYS STASCAN THRU SNPGT   MSM EVENT 10      *
C*                                                                     *
C*     SUBROUTINES REQUIRED                                            *
C*                 FINDX                                               *
C*                 FINDZ                                               *
C*                 FINDY                                               *
C*                 SNPGT                                               *
C*                 DOPLER                                              *
C*                 HYPERB                                              *
C*                 NORMER                                              *
C*                 RHOTHE                                              *
C*                 RERR                                                *
C*                 MERGDR                                              *
C*                 DORDER                                              *
C*                 GMERGE                                              *
C*                                                                     *
C*   REMARKS                                                           *
C*        REQUIRES BLOCK DATA JFBLK3 (DESIGNER INPUT )                 *
C*     INITIATES  GROUND TRUTH POSITIONS IN SXYTT, POINTER KSSN        *
C*     IT REQUESTS DISC SPACE FOR GROUND TRUTH -MTAPE(5)               *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-22

```
C******************************* PSNP  *********************************
C*                                                                     *
C*                         SUBROUTINE PSNP                             *
C*                                                                     *
C*    PURPOSE                                                          *
C*        COMPUTES GROUND TRUTH POSITIONS OF UGS ARRAYS                *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL PSNP(MPRINT)                                            *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*         * INPUT *                                                   *
C*        MPRINT CONTROLS BCD OUTPUT                                   *
C*                                                                     *
C*                                        PLANNER INPUT TABLE  ITEM    *
C*        PRNV1( ,10) NAVIGATION SYSTEM  HYPERBOLIC          15   ALL  *
C*        PRNV2( , 7) NAVIGATION SYSTEM  RHO THETA           16   ALL  *
C*        PRNV3( , 7) NAVIGATION SYSTEM  DOPPLER             17   ALL  *
C*        PRNV4(    ) NAVIGATION SYSTEM  NORMAL              18   ALL  *
C*                                                                     *
C*    SNER TABLE                          PLANNER INPUT TABLE ITEM     *
C*        SNER( ,1)  SENSOR ID                              3     1    *
C*        SNER( ,2)  MODE OF EMPLACEMENT                    1    11    *
C*        SNER( ,3)  ORDER OF EMPLACEMENT                              *
C*        SNER( ,4)  STANDARD DEVIATION OF MAP ERROR                   *
C*        SNER( ,5)  STANDARD DEVIATION  OF LOCATION ERROR  2     3    *
C*        SNER( ,6)  RELATIVE LOCATION ERROR                2     4    *
C*        SNER( ,7)  NAVIGATION SYSTEM OR WEAPON CODE       2   3-4    *
C*        SNER( ,8)  POSITION ERROR PARAMETER SET ID        2     1    *
C*        SNER( ,9)   X (MPL NE 1)                          2     5    *
C*        SNER( ,10)  Y (MPL NE 1)                          2     6    *
C*        SNER( ,11)  PLANNED X                             3 5-6-7    *
C*        SNER( ,12)  PLANNED Y                             3 5-6-7    *
C*        SNER( ,13)  AIRD(PLAY VERTICAL FAIL ERROR: YES = 1, NO = 0)  *
C*        SNER( ,14)  AIRCRAFT TYPE                         2     4    *
C*        SNER( ,15)  DROP SPEED                            2     7    *
C*        SNER( ,16)  DROP ALTITUDE                         2     8    *
C*        SNER( ,17)  PATH ANGLE                                       *
C*        SNER( ,20)  VERSION-NAVIGATION SYSTEM             2     3    *
C*        SNEPE TABLE                                                  *
C*        SNEPE( ,1)  ORDER                                            *
C*        SNEPE( ,I)  I=2,5, REEMPLACEMENT TIMES                       *
C*                                                                     *
C*        * OUTPUT *                                                   *
C*        SXYTT- DX,DY,XO,YO,TO,T1, X1,Y1,T2,T3,X2,Y2,T4,T5,ETC.       *
C*        DX AND DY  ARE PATH DIRECTION SEGMENTS,FOLLOWED BY           *
C*         X-Y LOCATION AND CORRESPONDING TIME INTERVALS TO,T1         *
C*        BASIC XYTT PATTERN IS REPEATED AS OFTEN AS NECESSARY         *
C*        FOR REEMPLACEMENT-IF ANY                                     *
C*        KSSN( N) POINTER GIVING LOCATION OF START OF SXYTT           *
```

Figure 2.1-23

```
C*          INFORMATION FOR SENSOR N IN SXYTT(  )ARRAY                  *
C*          IVE(MVE+  ) EVENT TYPE 10  FOR MSM AS REQUIRED              *
C*          UDTM  UPDOWN TIMES                                          *
C*          KSDN  POINTER FOR SENSOR UPDOWN TIMES IN UDTM               *
C*                                                                      *
C*     SUBROUTINE REQUIRED                                              *
C*          SNPGT  - TO COMPUTE GROUND TRUTH                            *
C*          MERGDR- TO MERGE MSM EVENTS                                 *
C*                                                                      *
C*     METHOD                                                           *
C*          THE PLANNER INPUT PRNV TABLES ARE LOCATED USING FINDX. IF A *
C*          TABLE IS VOID THE VALUES IN THE BLOCK DATA ARE USED. THE    *
C*          PLANNER INPUT DATA FOR THE ARRAYS, POSITION ERRORS, SENSORS *
C*          AND PATHS ARE LOCATED BY CALLS TO FINDX.                    *
C*          THE SNER AND SNEPE TABLES ARE SET. UPDOWN TIMES ARE LOCATED *
C*          IN UDTM BY CALLING FINDY.  THE PATH DATA IS SET INTO ARRAY  *
C*          TRAIL. THE VARIOUS PLAY OPTIONS ARE CHECKED AND THE         *
C*          APPROPIATE PARAMETERS ARE SET. SUBROUTINE SNPGT IS CALLED.  *
C*          THE GROUND TRUTH POSITIONS ARE PLACED IN SXYTT ALONG WITH   *
C*          THE CORRESPONDING TIMES AND THE POINTER KSSN IS SET. IF     *
C*          THE SENSORS WERE RE-EMPLACED SXYTT IS AUGMENTED AND MSM     *
C*          EVENT TYPE 10 IS GENERATED. THE PRINT OPTION IS CHECKED.    *
C*          AFTER ALL ARRAYS ARE PROCESSED MERGDR IS CALLED TO ORDER AND *
C*          MERGE THE EVENTS WITH THE MASTER LIST.                      *
C*                                                                      *
C***********************************************************************
```

Figure 2.1-23 (Cont.)

```
C************************************ PSNP19 ************************************
C*                                                                              *
C*                            SUBROUTINE PSNP19                                 *
C*                                                                              *
C*     PURPOSE                                                                  *
C*         COMPUTES GROUND TRUTH POSITIONS OF STASCAN ARRAYS                    *
C*                                                                              *
C*     CALLING SEQUENCE                                                         *
C*         CALL PSNP19(MPRINT)                                                  *
C*                                                                              *
C*     DESCRIPTION OF PARAMETERS                                                *
C*         * INPUT *                                                            *
C*         MPRINT CONTROLS BCD OUTPUT                                           *
C*                                                                              *
C*                                             PLANNER INPUT TABLE  ITEM        *
C*         PRNV1( ,10) NAVIGATION SYSTEM  HYPERBOLIC          15    ALL         *
C*         PRNV2( , 7) NAVIGATION SYSTEM  RHO THETA           16    ALL         *
C*         PRNV3( , 7) NAVIGATION SYSTEM  DOPPLER             17    ALL         *
C*         PRNV4(    ) NAVIGATION SYSTEM  NORMAL              18    ALL         *
C*                                                                              *
C*     SNER TABLE                              PLANNER INPUT TABLE ITEM         *
C*         SNER( ,1)  SENSOR ID                                3      1         *
C*         SNER( ,2)  MODE OF EMPLACEMENT                     19     11         *
C*         SNER( ,3)  ORDER OF EMPLACEMENT                                      *
C*         SNER( ,4)  STANDARD DEVIATION OF MAP ERROR                           *
C*         SNER( ,5)  STANDARD DEVIATION  OF LOCATION ERROR    2      3         *
C*         SNER( ,6)  RELATIVE LOCATION ERROR                  2      4         *
C*         SNER( ,7)==0                                                         *
C*         SNER( ,8)=0                                                          *
C*         SNER( ,9) NOT USED                                                   *
C*         SNER( ,10)==0                                                        *
C*         SNER( ,11)  PLANNED X                               3 5-6-7          *
C*         SNER( ,12)  PLANNED Y                               3 5-6-7          *
C*         SNER( ,13)==0                                                        *
C*         SNER( ,14)==0                                                        *
C*         SNER( ,15)=0                                                         *
C*         SNER( ,16)==0                                                        *
C*         SNER( ,17)=0                                                         *
C*         SNER( ,20)==0                                                        *
C*         SNEPE TABLE                                                          *
C*         SNEPE( ,1)  ORDER                                                    *
C*         SNEPE( ,I)  I=2,5, REEMPLACEMENT TIMES                               *
C*                                                                              *
C*         * OUTPUT *                                                           *
C*         SXYTT- DX,DY,X0,Y0,T0,T1, X1,Y1,T2,T3,X2,Y2,T4,T5,ETC.               *
C*         DX AND DY  ARE PATH DIRECTION SEGMENTS,FOLLOWED BY                   *
C*          X-Y LOCATION AND CORRESPONDING TIME INTERVALS T0,T1                 *
C*         BASIC XYTT PATTERN IS REPEATED AS OFTEN AS NECESSARY                 *
C*         FOR REEMPLACEMENT-IF ANY                                             *
C*         KSSN( N) DICTIONARY GIVING LOCATION OF START OF SXYTT                *
```

Figure 2.1-24

```
C*        INFORMATION FOR SENSOR N IN SXYTT(   )ARRAY                    *
C*        IVE(MVE+   ) EVENT TYPE 10  FOR MSM AS REQUIRED                *
C*        UDTM  UPDOWN TIMES                                             *
C*        KSDN  POINTER FOR SENSOR UPDOWN TIMES IN UDTM                  *
C*                                                                       *
C*    SUBROUTINE REQUIRED                                                *
C*        SNPGT   - TO COMPUTE GROUND TRUTH                              *
C*        MERGDR- TO MERGE MSM EVENTS                                    *
C*                                                                       *
C*    METHOD                                                             *
C*        THE PLANNER INPUT PRNV TABLES ARE LOCATED USING FINDX. IF A    *
C*        TABLE IS VOID THE VALUES IN THE BLOCK DATA ARE USED. THE       *
C*        PLANNED INPUT DATA FOR THE ARRAYS, POSITION ERRORS, SENSORS    *
C*        AND PATHS ARE LOCATED BY CALLS TO FINDX.                       *
C*        THE SNER AND SNERE TABLES ARE SET. UPDOWN TIMES ARE LOCATED    *
C*        IN UDTM BY CALLING FINDY.  THE PATH DATA IS SET INTO ARRAY     *
C*        TRAIL. THE VARIOUS PLAY OPTIONS ARE CHECKED AND THE            *
C*        APPROPRIATE PARAMETERS ARE SET. SUBROUTINE SNPGT IS CALLED     *
C*        THE GROUND TRUTH POSITIONS ARE PLACED IN SXYTT ALONG WITH      *
C*        THE CORRESPONDING TIMES AND THE POINTER KSSN IS SET. IF        *
C*        THE SENSORS WERE RE-EMPLACED SXYTT IS AUGMENTED AND MSM        *
C*        EVENT TYPE 10 IS GENERATED. THE PRINT OPTION IS CHECKED        *
C*        AFTER ALL ARRAYS ARE PROCESSED MERGDR IS CALLED TO ORDER AND   *
C*        MERGE THE EVENTS WITH THE MASTER LIST.                         *
C*                                                                       *
C************************************************************************
```

Figure 2.1-24 (Cont.)

```
C****************************** SNPGT ********************************
C*                                                                   *
C*                        SUBROUTINE SNPGT                           *
C*                                                                   *
C*    PURPOSE                                                        *
C*        THIS ROUTINE DETERMINES THE GROUND TRUTH POSITIONS         *
C*        OF THE STATIONARY SENSORS                                  *
C*                                                                   *
C*    CALLING SEQUENCE                                               *
C*        CALL SNPGT                                                 *
C*                                                                   *
C*    DESCRIPTION OF PARAMETERS                                      *
C*        INPUT AND OUTPUT VIA BLANK COMMON AND LABELED COMMON BLOCK *
C*        /MVSNP/.
C*                                                                   *
C*    GLOSSARY                                                       *
C*SNER   SNPGT CP  TEMPORARY PLANNER INPUT TABLE, SENSOR GAME-DATA   *
C*ATMOE  SNPGT CP  ARTILLERY / MORTAR EMPLACEMENT ERROR (I,J,K)      *
C*ADRPI  SNPGT CP  AIR DROP SENSOR FALL ERROR-ROTARY WING A/C (J,L,M) *
C*SUNPE  SNPGT CP  SNPGT TABLE FOR STORAGE OF GUN POSITION ERROR     *
C*SNPUO  SNPGT CP  SNPGT TABLE, STORES SENSOR REEMPLACEMENT POSITIONS *
C*S      SNPGT DP  ONE STANDARD DEVIATION OF A NORMAL DISTRIBUTION   *
C*CX     SNPGT DP  DUMMY VARIABLE X DIRECTION,METERS                 *
C*CY     SNPGT DP  DUMMY VARIABLE Y DIRECTION,METERS                 *
C*VX     SNPGT DP  RANDOM NORMAL DEVIATE X DIRECTION,METERS          *
C*VY     SNPGT DP  RANDOM NORMAL DEVIATE Y DIRECTION,METERS          *
C*PX     SNPGT DP  RANDOM NORMAL DEVIATE X DIRECTION,METERS          *
C*PY     SNPGT DP  RANDOM NORMAL DEVIATE Y DIRECTION,METERS          *
C*PGTX   SNPGT CP  TEMP STORAGE OF X COORD OF SENSOR GROUND TRUTH POSN *
C*PGTY   SNPGT CP  TEMP STORAGE OF Y COORD OF SENSOR GROUND TRUTH POSN *
C*IFLAG  SNPGT DP  INDICATOR,IS SENSOR FIRST EMPLACEMENT OF ARRAY    *
C*VD     SNPGT DP  RANDOM NORMAL DEVIATE OF WEAPON DISPERSION IN     *
C*VD     SNPGT DP  DEFLECTION                                        *
C*IMAX   SNPGT CP  MAX NUMBER OF ROWS OF DATA IN THE SNER TABLE      *
C*IX     SNPGT CP  STARTER NUMBER FOR RANDOM NORMAL NUMBER GENERATER *
C*NAVSW  SNPGT DP  INDICATOR OF TYPE AIR NAVIGATION SYSTEM PLAYED    *
C*R      SNPGT DP  LENGTH OF LINE SEGMENT (RANGE),METERS             *
C*SL     SNPGT DP  SLOPE OF LINE SEGMENT,RATIO                       *
C*VXR    SNPGT DP  X COMPONENT OF RAND.NOR.DEV. OF WPN RANGE DISPER'N *
C*VYR    SNPGT DP  Y COMPONENT OF RAND.NOR.DEV. OF WPN RANGE DISPER'N *
C*VXD    SNPGT DP  X COMP'T OF RAND.NOR.DEV. OF WPN DEFL'N DISP'N    *
C*VYD    SNPGT DP  Y COMP'T OF RAND.NOR.DEV. OF WPN DEFL'N DISP'N    *
C*S1     SNPGT DP  ONE STANDARD DEVIATION OF A NORMAL DISTRIBUTION   *
C*S2     SNPGT DP  ONE STANDARD DEVIATION OF A NORMAL DISTRIBUTION   *
C*VR     SNPGT DP  RANDOM NORMAL DEVIATE OF WEAPON DISPERSION IN RANGE *
C*ERRX   SNPGT DP  X COMPONENT OF NAVIGATION SYSTEM ERROR,METERS     *
C*ERRY   SNPGT DP  Y COMPONENT OF NAVIGATION SYSTEM ERROR,METERS     *
C*ATX    SNPGT DP  X COMPONENT OF ALONG-TRACK ERROR,METERS           *
C*ATY    SNPGT DP  Y COMPONENT OF ALONG-TRACK ERROR,METERS           *
C*CTX    SNPGT DP  X COMPONENT OF CROSS-TRACK ERROR,METERS           *
```

Figure 2.1-25

```
C*CTY    SNPGT DP  Y COMPONENT OF CROSS-TRACK ERROR,METERS                *
C*I      SNPGT CP  SENSOR IDENTIFICATION NUMBER,INTERGER                  *
C*MJ     SNPGT DP  INDEXING INTEGER                                       *
C*SNEPE  RUSUP CP  TABLE OF SENSOR*UP TIMES*DURING THE GAME(M,N),SECONDS*
C*DX     SNPGT DP  RANDOM NORMAL DEVIATE X DIRECTION,METERS               *
C*DY     SNPGT DP  RANDOM NORMAL DEVIATE Y DIRECTION,METERS               *
C*AT     SNPGT DP  ALONG-TRACK COMPONENT OF SENSOR FALL ERROR,METERS      *
C*CT     SNPGT DP  CORSS-TRACK COMPONENT OF SENSOR FALL ERROR,METERS      *
C*JPMAX  SNPGT CP  MAX GAME ID NO OF GUN POSITION OF GUNDE TABLE          *
C*MMAX   SNPGT CP  MAX NUMBER OF ROWS OF DATA IN SNEPE TABLE              *
C*NMAX   SNPGT DP  MAX NUMBER OF COLUMNS OF DATA IN THE SNEPE TABLE       *
C*A      SNPGT DP  DUMMY VARIABLE FOR POSN GROUND TRUTH COORDINATE        *
C*B      SNPGT DP  TEMP VALUE TO DETERMINE J FOR ADRP TABLE               *
C*J      SNPGT DP  INDEXING INTEGER                                       *
C*K      SNPGT DP  INDEXING INTEGER                                       *
C*L      SNPGT DP  INDEXING INTEGER                                       *
C*JP     SNPGT DP  INDEXING INTEGER                                       *
C*ADRP2  SNPGT CP  SENSOR FALL DISPERSION ERROR FOR FIXED WING A/C        *
C*TRAIL  SNPGT CP  INPUT TABLE OF PLANNER ESTIMATED TRAIL SEG. COORDS     *
C*GX     SNPGT DP  RANDOM NORMAL DEVIATE X DIMENSION OF ERROR             *
C*GY     SNPGT DP  RANDOM NORMAL DEVIATE Y DIMENSION OF ERROR             *
C*PGX    SNPGT DP  X COORD OF PLANNERS INTENDED SENSOR EMPLACEMENT SITE   *
C*PGY    SNPGT DP  Y COORD OF PLANNERS INTENDED SENSOR EMPLACEMENT SITE   *
C*X1     SNPGT DP  X COORD OF TERMINAL OF LINE SEGMENT METERS             *
C*Y1     SNPGT DP  Y COORD OF TERMINAL OF LINE SEGMENT METERS             *
C*X2     SNPGT DP  X COORD OF TERMINAL OF LINE SEGMENT METERS             *
C*Y2     SNPGT DP  Y COORD OF TERMINAL OF LINE SEGMENT METERS             *
C*THETS  SNPGT DP  SLOPE OF TRAIL SEG. AT SENSOR POSN (FROM POS X AXIS)   *
C*THETR  SNPGT DP  SLOPE OF LOCATION ERROR VECTOR MEASD FROM POS X AXIS   *
C*PI     SNPGT DP  NUMBER OF RADIANS IN 180 DEGREE SECTOR (RADIANS)       *
C*THET1  SNPGT DP  DUMMY VAR. FOR LINE SLOPE (MEASD FROM POS X AXIS)      *
C*THET2  SNPGT DP  DUMMY VAR. FOR LINE SLOPE (MEASD FROM POS X AXIS)      *
C*ALPHA  SNPGT DP  ANGLE FOR PROJECTING ERROR VECTOR TO TRAIL SEGMENT     *
C*BX     SNPGT DP  X COMPONENT OF ERROR VECTOR (METERS)                   *
C*BY     SNPGT DP  Y COMPONENT OF ERROR VECTOR (METERS)                   *
C*ZAP    SNPGT DP  DUMMY VARIABLE (METERS)                                *
C*M      SNPGT DP  ID OF ROWS OF DATA IN SNEPE TABLE                      *
C*N      SNPGT DP  ID OF COLUMNS OF DATA IN SNEPE TABLE                   *
C*THETA  SNPGT DP  SLOPE OF LINE SEGMENT MEASURED FROM POS X AXIS         *
C*                                                                        *
C*                                                                        *
C*      METHOD:  SEE PAR 4.3.1, PAGE 4-28 VOL I, PART II.                 *
C*                                                                        *
C*                                                                        *
C*      SUBROUTINES REQUIRED                                              *
C*          GRN                                                           *
C*          HYPERB                                                        *
C*          RHOTHE                                                        *
C*          DOPLER                                                        *
C*          NORMER                                                        *
```

Figure 2.1-25 (Cont.)

```
C********************************** DOPLER **********************************
C*                                                                          *
C*                                                                          *
C*                        SUBROUTINE DOPLER                                 *
C*      PURPOSE                                                             *
C*           USED TO GENERATE X AND Y COMPONENTS OF SYSTEM ERROR EXPECTED FROM A *
C*           DOPPLER NAVIGATION SYSTEM.                                     *
C*                                                                          *
C*      CALLING SEQUENCE                                                    *
C*           CALL DOPLER(NVSW2,NSLEG,IUPDT,VS,ALT,PX,PY,X2,Y2,PRNV3,        *
C*                       ERRX,ERRY,UPDX,UPDY)                               *
C*                                                                          *
C*      DESCRIPTION OF PARAMETERS                                           *
C*      GLOSSARY                                                            *
C*PRNV3 DOPLERC   INPUT TABLE OF DOPPLER NAVIGATION SYSTEM ERROR DATA       *
C*FAK   DOPLERD   NAV SYSTEM INFLIGHT UPDATING ERROR FACTOR                 *
C*TI    DOPLERDP  TIME REQUIRED TO TRAVERSE LEG (SECONDS)                   *
C*A     DOPLERDP  PRODUCT OF TI AND B                                       *
C*B     DOPLERDP  NOISE BANDWIDTH OF DOPLER SYSTEM (1/SECONDS)              *
C*ATSE  DOPLERDP  ALONG-TRACK DOPPLER SENSOR ERROR (METERS)                 *
C*ATCE  DOPLERDP  ALONG TRACK DOPPLER COMPUTER ERROR (METERS)               *
C*CTSE  DOPLERDP  CROSS TRACK DOPPLER SENSOR ERROR (METERS)                 *
C*CTCE  DOPLERDP  CROSS TRACK DOPPLER COMPUTER ERROR (METERS)               *
C*RS    DOPLERDP  ONE STD DEV (NORMAL) OF DOPPLER REGISTRATION ERROR        *
C*NSLEG DOPLERCP  ID OF LEG INITIATION POINT  INTEGER                       *
C*X2    DOPLERCP  X COORD OF MOVEMENT LEG INITIATION POINT (METERS)         *
C*Y2    DOPLERCP  Y COORD OF MOVEMENT LEG INITIATION POINT (METERS)         *
C*PX    DOPLERCP  X COORD OF INITIATION POINT OF PREVIOUS MOVEMENT LEG      *
C*PY    DOPLERCP  Y COORD OF INITIATION POINT OF PREVIOUS MOVEMENT LEG      *
C*DIST  DOPLERDP  LEG LENGTH (METERS)                                       *
C*VS    DOPLERCP  AVERAGE VELOCITY OF MOVEMENT ON THIS LEG                  *
C*ALT   DOPLERCP  ELEVATION OF THE SENSOR PLATFORM ABOVE GROUND LEVEL       *
C*UPDX  DOPLERCP  X COMPONENT OF NAV SYSTEM ERROR AFTER UPDATING            *
C*IUPDT DOPLERCP  SIGNAL TO SHOW IF DOPPLER ERROR WILL BE UPDATED           *
C*RT    DOPLERDP  FACTOR FOR COMPUTING ALONG AND CROSS TRACK ERRORS         *
C*ATE   DOPLERDP  ONE STD NORMAL DEV OF ALONG TRACK ERROR (METERS)          *
C*CTE   DOPLERDP  ONE STD NORMAL DEV OF CROSS TRACK ERROR (METERS)          *
C*ATS   DOPLERDP  COMBINED ONE STD NORMAL DEV OF ALONG TRACK ERROR          *
C*CTS   DOPLERDP  COMBINED ONE STD NORMAL DEV OF CROSS TRACK ERROR          *
C*AEX   DOPLERDP  X COMPONENT OF ALONG TRACK ERROR (METERS)                 *
C*AEY   DOPLERDP  Y COMPONENT OF ALONG TRACK ERROR (METERS)                 *
C*CEX   DOPLERDP  X COMPONENT OF CROSS TRACK ERROR (METERS)                 *
C*CEY   DOPLERDP  Y COMPONENT OF CROSS TRACK ERROR (METERS)                 *
C*AT    DOPLERDP  RANDOM NORMAL DEVIATE OF ALONG TRACK ERROR (METERS)       *
C*CT    DOPLERDP  RANDOM NORMAL DEVIATE OF CROSS TRACK ERROR (METERS)       *
C*ERRX  DOPLERCP  X-COMPONENT OF TOTAL DOPPLER ERROR (METERS)               *
C*ERRY  DOPLERCP  Y-COMPONENT OF TOTAL DOPPLER ERROR (METERS)               *
C*SL    DOPLERDP  LINE SLOPE REAL                                           *
C*THETA DOPLERDP  ANGULAR SLOPE OF MOVEMENT LEG FROM POSITIVE X AXIS        *
C*NVSW2 DOPLERCP  ID OF THE PARTICULAR SET (OF4) OF DOPPLER NAV SYSTEM      *
C*UPDY  DOPLERCP  Y COMPONENT OF NAV SYSTEM ERROR AFTER UPDATING            *
C*                                                                          *
C*      METHOD:  SEE 2-54.5.                                                *
C****************************************************************************
```

Figure 2.1-26

```
C*      METHOD                                                                  *
C*               FROM THE DOPLER NAVIGATION PARAMETER TABLE, NAVIGATION SYSTEM *
C*          ERROR PARAMETERS AND A SYSTEM UPDATING FACTOR ARE OBTAINED.  ONE   *
C*          STANDARD DEVIATION OF DOPPLER REGISTRATION ERROR IS ESTABLISHED.   *
C*          THE NEXT STEP IS TO CALCULATE THE DISTANCE AND TIME USED IN TRA-   *
C*          VERSING THE LEG.  THE FACTOR FOR COMPUTING ALONG AND CROSS TRACK   *
C*          ERRORS IS FOUND AND IS USED IN COMPUTING ONE STANDARD NORMAL DEVI- *
C*          ATIONS OF ALONG TRACK ERRORS AND CROSS TRACK ERRORS.  RANDOM NOR-  *
C*          MAL DEVIATES OF ALONG TRACK ERRORS AND CROSS TRACK ERRORS ARE COM- *
C*          PUTED.                                                             *
C*               AT THIS TIME THE X COMPONENT OF ALONG TRACK ERROR AND CROSS   *
C*          TRACK ERROR IS CALCULATED AND COMBINED INTO AN X COMPONENT OF SYS- *
C*          TEM ERROR.  THE SAME PROCEDURE IS UTILIZED TO OBTAIN A Y COMPONENT *
C*          OF SYSTEM ERROR.  IF UPDATING IS CALLED FOR, ONE STANDARD DEVIA-   *
C*          TION (NORMAL) OF DOPPLER REGISTRATION ERROR IS CALCULATED FROM AIR-*
C*          CRAFT ALTITUDE AND THE ERROR FACTOR.  THEN X AND Y COMPONENTS OF   *
C*          UPDATED NAVIGATION SYSTEM ERRORS ARE COMPUTED.                     *
C*                                                                             *
C******************************************************************************
```

Figure 2.1-26 (Cont.)

```
C******************************* HYPERB *********************************
C*                                                                      *
C*                                                                      *
C*                          SUBROUTINE HYPERB                           *
C*    PURPOSE                                                           *
C*        USED TO COMPUTE X AND Y COMPONENTS OF SYSTEM ERROR WHEN A SENSOR PLAT- *
C*    FORM IS SUPPORTED BY A HYPERBOLIC NAVIGATION SYSTEM.              *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL HYPERB(NVSW2,NSLEG,ITRAL,PX,PY,X2,Y2,SX,SY,PRNV1,ERRX,   *
C*                    ERRY)                                             *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*    GLOSSARY                                                          *
C*NVSW2 HYPERBCP  ID OF THE PARTICULAR SET OF HYPERBOLIC NAV ERROR      *
C*SLE   HYPERBDP  NAV GRND. STA. LOC. ERROR (ONE STD DEV,CIR NORM)      *
C*TDM   HYPERBDP  TIME DISTANCE MEAS ERROR ONE STD DEV,NORMAL DIST      *
C*XS1   HYPERBDP  X POSITION COORD, FIRST SLAVE STATION (METERS)        *
C*YS1   HYPERBDP  Y POSITION COORD,FIRST SLAVE STATION (METERS)         *
C*XS2   HYPERBDP  X POSITION COORD SECOND SLAVE STATION (METERS)        *
C*YS2   HYPERBDP  Y POSITION COORD SECOND SLAVE STATION (METERS)        *
C*XS3   HYPERBDP  X POSITION COORD THIRD SLAVE STATION (METERS)         *
C*YS3   HYPERBDP  Y POSITION COORD THIRD SLAVE STATION (METERS)         *
C*XMS   HYPERBDP  X POSITION COORD MASTER STATION (METERS)              *
C*YMS   HYPERBDP  Y POSITION COORD MASTER STATION (METERS)              *
C*PRNV1 HYPERBCP  INPUT TABLE OF NAVIGATION SYSTEM ERROR PARAMETERS     *
C*ITRAL HYPERBCP  FLAG TO SHOW IF MOVEMENT FOLLOWS PROMINENT ROUTE      *
C*BETA1 HYPERBDP  ANGLE FROM  OS X-AXIS TO LINE FROM SLAVE 1 TO SENSOR *
C*BETA2 HYPERBDP  ANGLE FROM POS X-AXIS TO LINE FROM SLAVE 2 TO SENSOR *
C*BETA3 HYPERBDP  ANGLE FROM POS X-AXIS TO LINE FROM SLAVE 3 TO SENSOR *
C*BETA4 HYPERBDP  ANGLE FROM POS X-AXIS TO LINE FROM MASTER TO SENSOR  *
C*ALPH1 HYPERBDP  ANGLE FROM L.O.P.1 TO LINE JOINING SENSOR & SLAVE 1  *
C*ALPH2 HYPERBDP  ANGLE FROM L.O.P.2 TO LINE JOINING SENSOR & SLAVE 2  *
C*ALPH3 HYPERBDP  ANGLE FROM L.O.P.3 TO LINE JOINING SENSOR & SLAVE 3  *
C*THET3 HYPERBDP  ANGLE BETWEEN POS X-AXIS AND L.O.P.3 (RADIANS)        *
C*FX1   HYPERBDP  X COMP OF ERROR FOR POSN FIX FROM LOP 1 AND LOP 2     *
C*FY1   HYPERBDP  Y COMP OF ERROR FOR POSN FIX FROM LOP 1 AND LOP 2     *
C*FX2   HYPERBDP  X COMP OF ERROR FOR POSN FIX FROM LOP 2 AND LOP 3     *
C*FY2   HYPERBDP  Y COMP OF ERROR FOR POSN FIX FROM LOP 2 AND LOP 3     *
C*SG1   HYPERBDP  ONE STD NORM DEV OF SENSOR POSN ERR. PERP TO L.O.P.1 *
C*SG2   HYPERBDP  ONE STD NORM DEV OF SENSOR POSN ERR. PERP TO L.O.P.2 *
C*SG3   HYPERBDP  ONE STD NORM DEV OF SENSOR POSN ERR. PERP TO L.O.P.3 *
C*E1    HYPERBDP  RANDOM NORMAL DEVIATE OF SENSOR POS. ERROR TO SG1     *
C*E2    HYPERBDP  RANDOM NORMAL DEVIATE OF SENSOR POS. ERROR TO SG2     *
C*THET1 HYPERBDP  ANGLE BETWEEN POS X-AXIS AND LOP 1 (RADIANS)          *
C*THET2 HYPERBDP  ANGLE BETWEEN POS X-AXIS AND LOP2 (RADIANS)           *
C*E1X   HYPERBDP  X COMPONENT OF SENSOR POSITION ERROR FOR LOP 1        *
C*E1Y   HYPERBDP  Y COMPONENT OF SENSOR POSITION ERROR FOR LOP 1        *
C*E2X   HYPERBDP  X COMPONENT OF SENSOR POSITION ERROR FOR LOP 2        *
C*E2Y   HYPERBDP  Y COMPONENT OF SENSOR POSITION ERROR FOR LOP 2        *
```

Figure 2.1-27

```
C*X2     HYPERBCP  X COORDINATE OF INITIAL POINT OF LEG (METERS)           *
C*Y2     HYPERBCP  Y COORDINATE OF INITIAL POINT OF LEG (METERS)           *
C*E3X    HYPERBDP  X COMPONENT OF SENSOR POSITION ERROR  FOR LOP 3         *
C*E3Y    HYPERBDP  Y COMPONENT OF SENSOR POSITION ERROR FOR LOP 3          *
C*E3     HYPERBDP  RANDOM NORMAL DEVIATE OF POSITION ERROR FROM SG3        *
C*ERRX   HYPERBCP  X COMPONENT OF AGGREGATED ERROR FOR HYPERB SYSTEM       *
C*ERRY   HYPERBCP  Y COMPONENT OF AGGREGATED ERROR FOR HYPERB SYSTEM       *
C*FX3    HYPERBDP  X COMP OF ERROR FOR FIX FROM LOP 1 AND LOP 3            *
C*FY3    HYPERBDP  Y COMP OF ERROR FOR FIX FROM LOP 1 AND LOP 3            *
C*IGAS   HYPERBDP  FLAG TO SHOW IF JUST 2 OR ALL 3 LOPS ARE USEFUL         *
C*R1     HYPERBDP  MAGNITUDE OF ERROR VECTOR FROM LOP 1 AND LOP 2          *
C*R2     HYPERBDP  MAGNITUDE OF ERROR VECTOR FOR LOP 2 AND LOP 3           *
C*R3     HYPERBDP  MAGNITUDE OF ERROR VECTOR FOR LOP 3 AND LOP 1           *
C*NSLEG  HYPERBCP  ID OF LEG INITIATION POINT                              *
C*PX     HYPERBCP  X COORD OF INIT.PT. OF PREVIOUS MOVEMENT LEG            *
C*PY     HYPERBCP  Y COORD OF INIT.PT. OF PREVIOUS MOVEMENT LEG            *
C*SX     HYPERBCP  X COORD OF INIT.PT. OF SUCCEEDING MOVEMENT LEG          *
C*SY     HYPERBCP  Y COORD OF INIT.PT. OF SUCCEEDING MOVEMENT LEG          *
C*IX     HYPERBCP  STARTING AND SUBSEQUENT RAND. INTG. FOR RAND.NO.GEN.    *
C*SL1    HYPERBDP  SLOPE OF LINE JOINING SLAVE STN NO 1 AND SENSOR         *
C*SL2    HYPERBDP  SLOPE OF LINE JOINING SLAVE STN NO 2 AND SENSOR         *
C*SL3    HYPERBDP  SLOPE OF LINE JOINING SLAVE STN NO 3 AND SENSOR         *
C*SL4    HYPERBDP  SLOPE OF LINE JOINING MASTER STN AND SENSOR             *
C*R      HYPERBDP  MAGNITUDE OF HYPERBOLIC SYSTEM ERROR (METERS)           *
C*RX     HYPERBDP  COMPONENT OF HYPERBOLIC SYSTEM ERROR (METERS)           *
C*RY     HYPERBDP  Y MPONENT OF HYPERBOLIC SYSTEM ERROR (METERS)           *
C*RSL    HYPERBDP  SLOPE OF HYPERBOLIC SYSTEM ERROR VECTOR                 *
C*THETR  HYPERBDP  SLOPE OF HYPERBOLIC SYSTEM ERROR VECTOR (RADIANS)       *
C*PSL    HYPERBDP  SLOPE OF LINE FROM PRESENT TO PAST LEG START POINT      *
C*THETP  HYPERBDP  SLOPE OF LINE FROM PRESENT TO PREVIOUS LEG START PT.    *
C*SSL    HYPERBDP  SLOPE OF LINE FROM PRESENT TO NEXT LEG START POINT      *
C*THETS  HYPERBSP  LOPE OF LINE FROM PRESENT TO NEXT LEG START POINT       *
C*PI     HYPERBDP  SYMBOL FOR THE NUMBER OF RADIANS IN 180 DEGREES         *
C*ALPHA  HYPERBDP  THE ANGLE BETWEEN A MOVEMENT LEG AND THE ERR.VECTOR     *
C*ZAP1   HYPERBDP  TEMPORARY NAME FOR THETA1 (RADIANS)                     *
C*ZAP2   HYPERBDP  TEMPORARY NAME FOR THETA2 (RADIANS)                     *
C*                                                                         *
C*    METHOD: SEE 2-56.5.                                                  *
C**************************************************************************
```

Figure 2.1-27 (Cont.)

```
C*    METHOD                                                              *
C*          THE SYSTEM GROUND STATION LOCATION AND ERROR COMPONENTS ARE   *
C*       EXTRACTED FROM THE NAVIGATION PARAMETER TABLE.  THE SLOPE OF THE *
C*       LINE (BETA), WHICH IS MEASURED BETWEEN ONE OF THE FOUR GROUND STA- *
C*       TIONS (ONE MASTER, THREE SLAVES) AND THE SENSOR PLATFORM IS CALCU- *
C*       LATED AND SUCH BETA VALUES ARE UTILIZED TO COMPUTE THE ANGLE ALPH, *
C*       WHICH IS BETWEEN THE BETA LINE AND THE CORRESPONDING LINE OF POSI- *
C*       TION (LOP) TO THE SENSOR PLATFORM.                               *
C*          CALCULATIONS ARE THEN MADE ON THE VARIABLE SG, WHICH IS ONE   *
C*       STANDARD NORMAL DEVIATE OF SENSOR POSITION ERROR PERPENDICULAR TO *
C*       THE APPLICABLE LOP.  SUCH SG VALUES ARE DEPENDENT ON THE SPH VARI- *
C*       ABLES WHICH EQUAL THE SINE OF THE APPROPRIATE ALPH ANGLE.  THE   *
C*       ABOVE NAMED CALCULATIONS IN CONJUNCTION WITH AN IGAS VARIABLE ARE *
C*       USED TO DETERMINE THE NUMBER OF USEABLE MASTER SLAVE COMBINATIONS. *
C*       ALSO, THE SLOPE OF THE LINE OF POSITION THET IS ALSO CALCULATED.  *
C*          NEXT RANDOM NORMAL DEVIATES OF SENSOR POSITION ERRORS TO THE  *
C*       SG VALUES ARE CALCULATED.  UTILIZING THE THET VALUES, X AND Y COM- *
C*       PONENTS OF SENSOR POSITION ERRORS (E1X, E1Y) FOR THE APPLICABLE  *
C*       LOPS ARE COMPUTED.                                               *
C*          A CALL IS MADE TO SUBROUTINE RERR TO FIND THE POINT OF INTER- *
C*       SECTION OF TWO LINES, ONE PASSING THRU THE POINT (E1X, E1Y) AND  *
C*       PARALLEL TO THE LOP WITH SLOPE THET 1 AND THE OTHER PASSING THRU *
C*       THE POINT (E2X, E2Y) AND PARALLEL TO THE LOP WITH SLOPE THET 2.  *
C*       THE VECTOR FROM THE SENSOR POSITION TO THE ABOVE NAMED INTERSEC- *
C*       TION POINT IS THE NAVIGATION ERROR VECTOR FOR THE STATION PAIR   *
C*       DESCRIBED.  ALL THREE POSSIBLE ERROR VECTORS ARE COMPUTED.       *
C*          THESE THREE ERROR VECTORS (R1, R2, R3) ARE COMPARED AND THE   *
C*       SMALLEST VECTOR IS SELECTED ALONG WITH ITS X AND Y COMPONENTS.   *
C*       FURTHER COMPUTATIONS ARE MADE IF THE MOVEMENT IS NOT CONFINED TO A *
C*       TRAIL, AND THE X AND Y COMPONENTS OF THE ERROR VECTOR THAT IS PRO- *
C*       JECTED ONTO THE TRAIL SEGMENTS ARE CALCULATED.                   *
C*                                                                        *
C*************************************************************************
```

Figure 2.1-27 (Cont.)

```
C******************************** RERR    *********************************
C*                                                                        *
C*                            SUBROUTINE RERR                             *
C*                                                                        *
C*     PURPOSE                                                            *
C*        USED IN THE COMPUTATION OF NAVIGATION ERRORS                    *
C*                                                                        *
C*     CALLING SEQUENCE                                                   *
C*        CALL RERR(ZX1,ZY1,ZETA1,ZX2,ZY2,ZETA2,ZX,ZY,R)                  *
C*                                                                        *
C*     GLOSSARY                                                           *
C*ZX     RERR  CP  X-COMP OF ERROR OF A FIX FROM 2 LOPS (METERS)          *
C*ZY     RERR  CP  Y-COMP OF ERROR OF A FIX FROM 2 LOPS (METERS)          *
C*ZX1    RERR  CP  X-COMP OF ERROR FOR LOP 1 (METERS)                     *
C*ZY1    RERR  CP  Y-COMP OF ERROR FOR LOP 1 (METERS)                     *
C*ZX2    RERR  CP  X-COMP OF ERROR FOR LOP 2 (METERS)                     *
C*ZY2    RERR  CP  Y-COMP OF ERROR FOR LOP 2 (METERS)                     *
C*ZETA1 RERR CP ERROR 1 MAGNITUDE                                         *
C*ZETA2 RERR CP ERROR 2 MAGNITUDE                                         *
C*R      RERR  CP  TOTAL ERROR MAGNITUDE FOR FIX FROM 2 LOPS (METERS)     *
C*                                                                        *
C*     METHOD                                                             *
C*             THE X AND Y COMPONENTS OF ERROR AND THE TOTAL ERROR MAGNITUDE FOR  *
C*        A FIX FROM TWO LINES OF POSITIONS ARE SELECTED.  A VARIABLE E IS CALCU- *
C*        LATED BY MULTIPLYING THE X COMPONENT OF ERROR FOR LINE OF POSITION 1 BY *
C*        THE Y COMPONENT OF ERROR FOR LINE OF POSITION 2 AND THEN SUBTRACTING THE *
C*        PRODUCT OF THE Y COMPONENT OF ERROR FOR LINE OF POSITION 1 AND THE X    *
C*        COMPONENT OF ERROR FOR LINE OF POSITION 2.  IF THE E VALUE IS ZERO, THE *
C*        PROCESSING IS COMPLETED; IF THE E VALUE IS NOT EQUAL TO ZERO, THE X AND *
C*        Y COMPONENTS OF ERROR OF A FIX FROM TWO LINES OF POSITION AND THE TOTAL *
C*        ERROR MAGNITUDE FOR A FIX FROM TWO LINES OF POSITION ARE COMPUTED.      *
C*                                                                        *
C**************************************************************************
```

Figure 2.1-28

```
C******************************* NORMER **********************************
C*                                                                      *
C*                                                                      *
C*                        SUBROUTINE NORMER                             *
C*    PURPOSE                                                           *
C*        USED TO COMPUTE X AND Y COMPONENT NORMALLY DISTRIBUTED NAVIGATION ERRORS *
C*    WHEN POSITION LOCATION IS DETERMINED IN OPEN TERRAIN FROM MAPS AND VISUAL *
C*    SIGHTINGS.                                                        *
C*    CALLING SEQUENCE                                                  *
C*        CALL NORMER(NVSW2,NSLEG,ITRAL,PX,PY,X2,Y2,SX,SY,PRNV4,ERRX,   *
C*    *                   ERRY)                                         *
C*                                                                      *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*    GLOSSARY                                                          *
C*PRNV4 NORMERCP  INPUT TABLE OF NORMER NAVIGATION SYSTEM ERROR DATA    *
C*X2    NORMERCP  X COORD. OF MOVEMENT LEG INITIATION PT. (METERS)      *
C*Y2    NORMERCP  Y COORD. OF MOVEMENT LEG INITIATION PT. (METERS)      *
C*SX    NORMERCP  X-COORD OF INITIATION PT. OF SUCCEEDING LEG (METERS)  *
C*SY    NORMERCP  Y COORD OF INITIATION PT. OF SUCCEEDING LEG (METERS)  *
C*PX    NORMERCP  X COORD OF INITIATION POINT OF PREVIOUS LEG (METERS)  *
C*PY    NORMERCP  Y COORD OF INITIATION POINT OF PREVIOUS LEG (METERS)  *
C*ITRAL NORMERCP  INDICATOR TO SHOW WHETHER OR NOT TRAIL FOLLOWING      *
C*ITRAL NORMERCP  APPLIES                                               *
C*PS    NORMERDP  ONE STD DEV,CIRC-NORM DIST,OF NORMER LOCATION ERROR   *
C*IX    NORMERCP  STARTING AND SUBSEQUENT RAND. INTEGERS FOR            *
C*IX    NORMERCP  RAND. NO. GEN.                                        *
C*AR    NORMERDP  NORMER LOCATION ERROR RANDOM DEV. OF CIR NOR.DIST.    *
C*SL    NORMERDP  LINE SLOPE                                            *
C*THETA NORMERDP  ANGULAR SLOPE OF MOVEMENT LEG MEAS FROM POS X AXIS    *
C*ERRX  NORMERCP  X COMPONENT OF ERROR FROM NORMER SYSTEM (METERS)      *
C*ERRY  NORMERCP  Y COMPONENT OF ERROR FROM NORMER SYSTEM (METERS)      *
C*NVSW2 NORMERCP  ID OF A NAV SYSTEM WITHIN A SET OF SYSTEMS ALL OF     *
C*NVSW2 NORMERCP  ONE TYPE                                              *
C*NSLEG NORMERCP  ID OF LEG INITIAL POINT                               *
C*                                                                      *
C*    REMARKS                                                           *
C*        NONE                                                          *
C*                                                                      *
C*    SUBROUTINES REQUIRED                                              *
C*        GRN                                                           *
C*    METHOD:  See 2-58.5.                                              *
C************************************************************************
```

Figure 2.1-29

```
C*     METHOD                                                                *
C*              UTILIZING THE INPUT TABLE OF NORMER NAVIGATION SYSTEM ERROR  *
C*         DATA, ONE STANDARD DEVIATION OF NAVIGATION LOCATION ERROR IS OB-  *
C*         TAINED.  THEN A CHECK IS MADE USING THE INDICATOR ITRAL TO SEE    *
C*         WHETHER A MOVEMENT FOLLOWS A TRAIL. WHEN MOVEMENT FOLLOWS A TRAIL *
C*         (ITRAL = 1), A RANDOM NORMAL DEVIATE IS CALCULATED THAT REPRESENTS *
C*         THE NAVIGATION ERROR VECTOR.  IF MOVEMENT DOES NOT FOLLOW A TRAIL, *
C*         X AND Y ERROR COMPONENTS FROM NORMER SYSTEM ARE COMPUTED.         *
C*              IF THE RANDOM NORMAL DEVIATE CALCULATED FOR MOVEMENT ALONG A *
C*         TRAIL IS NEGATIVE, THE NAVIGATION ERROR VECTOR IS ASSUMED TO BE DI- *
C*         RECTED TOWARD THE STARTING POINT OF THE PREVIOUS TRAIL SEGMENT.  IF *
C*         THE RANDOM NORMAL DEVIATE IS POSITIVE, THE ERROR VECTOR IS DIRECTED *
C*         TOWARD THE STARTING POINT OF THE NEXT SEGMENT.  THEN THE X AND Y  *
C*         COMPONENTS OF THE NORMER ERROR ARE COMPUTED.                      *
C*                                                                           *
C****************************************************************************
```

Figure 2.1-29 (Cont.)

```
C****************************** RHOTHE *********************************
C*                                                                     *
C*                                                                     *
C*                         SUBROUTINE RHOTHE                           *
C*    PURPOSE                                                          *
C*        USED TO COMPUTE X AND Y COMPONENT ERRORS REPRESENTATIVE OF A RHO THETA  *
C*    RADAR TRACKING NAVIGATION SYSTEM.                                *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL RHOTHE(NVSW2,ALT,X2,Y2,PRNV2,ERRX,ERRY)                 *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*    GLOSSARY                                                         *
C*PRNV2 RHOTHECP  INPUT TABLE OF NAV SYSTEM ERROR PARAMETER DATA,REAL  *
C*NVSW2 RHOTHECP  ID OF PARTICULAR SET OF RHOTHETA NAV SYS. DATA USED  *
C*Z1    RHOTHEDP  ELEVATION OF RADAR SITE ABOVE MEAN SEALEVEL (METERS) *
C*HI    RHOTHEDP  HEIGHT OF SENSOR ABOVE RADAR SITE (METERS)           *
C*SLE   RHOTHEDP  NAV SYS STATION LOC ERROR ONE STD DEV CIRCULAR NORM  *
C*BRE   RHOTHEDP  DIRECTION RESOLUTION ERROR OF RHOTHETA,ONE STD       *
C*BRE   RHOTHEDP  NORM. DEV.                                           *
C*RRE   RHOTHEDP  RANGE RESOLUTION ERROR OF RHOTHETA,ONE STD NORM DEV  *
C*HRE   RHOTHEDP  ALTITUDE RESOLUTION ERROR OF ALTIMETER ONE STD       *
C*HRE   RHOTHEDP  NORM DEV                                             *
C*X1    RHOTHEDP  X COORD OF RHOTHETA GROUND STATION POSITION (METERS) *
C*Y1    RHOTHEDP  Y COORD OF RHOTHETA GROUND STATION POSITION (METERS) *
C*ALT   RHOTHECP  SENSOR ALTITUDE ABOVE MEAN SEALEVEL (METERS)         *
C*X2    RHOTHECP  X COORD OF LEG INITIATION POINT (METERS)             *
C*Y 2   RHOTHECP  Y COORD OF LEG INITIATION POINT (METERS)             *
C*DD1   RHOTHEDP  SQUARE OF GROUND DIST FROM RADAR STATION TO SENSOR   *
C*DD2   RHOTHEDP  SQUARE OF SLANT RANGE FROM RADAR STATION TO SENSOR   *
C*GRRS  RHOTHEDP  ONE STD NORMAL DEV OF SENSOR POSITION ERROR IN RANGE *
C*GRAS  RHOTHEDP  ONE STD NORMAL DEV OF SENSOR POS. ERROR IN AZIMUTH   *
C*SL    RHOTHEDP  LINE SLOPE                                           *
C*THETA RHOTHEDP  DIRECTION FROM POS. X-AXIS OF SENSOR GROUND PLANE    *
C*GRE   RHOTHEDP  RAND. NORM. DEV. OF SENSOR POS. ERROR IN RANGE       *
C*GAE   RHOTHEDP  RAND. NORM. DEV. OF SENSOR POSN ERROR IN AZIMUTH     *
C*GRX   RHOTHEDP  X COMPONENT OF SENSOR POSITION ERROR IN RANGE        *
C*GRY   RHOTHEDP  Y COMPONENT OF SENSOR POSITION ERROR IN RANGE        *
C*GAX   RHOTHEDP  X COMPONENT OF SENSOR POSITION ERROR IN AZIMUTH      *
C*GAY   RHOTHEDP  Y COMPONENT OF SENSOR POSITION ERROR IN AZIMUTH      *
C*ERRX  RHOTHECP  X COMPONENT OF TOTAL ERROR (METERS)                  *
C*ERRY  RHOTHECP  Y COMPONENT OF TOTAL ERROR (METERS)                  *
C*IX    RHOTHECP  STARTING AND SUBSEQUENT RANDOM INTEGERS              *
C*IX    RHOTHECP  FROM RAND. NO. GENERATOR.                            *
C*                                                                     *
C*    REMARKS                                                          *
C*        NONE                                                         *
C*                                                                     *
C*    SUBROUTINES REQUIRED                                             *
C*        GRN                                                          *
C*    METHOD: See 2-59.5.                                              *
```

Figure 2.1-30

```
C*     METHOD
C*              THE NAVIGATION PARAMETER TABLE IS UTILIZED TO OBTAIN SYSTEM      *
C*         COMPONENT ERRORS, SUCH AS NAVIGATION SYSTEM LOCATION ERROR, DIREC-   *
C*         TION RESOLUTION ERROR, RANGE RESOLUTION ERROR, AND ALTITUDE RESOLU-  *
C*         TION ERROR.  THE X AND Y COORDINATES OF THE RHO THETA GROUND STATION*
C*         ARE ESTABLISHED AND THE ELEVATION OF THE RADAR SITE IS DETERMINED.  *
C*         AFTER THE ABOVE NAMED PARAMETERS ARE OBTAINED, CALCULATIONS ARE MADE*
C*         TO OBTAIN THE SQUARE OF THE GROUND RANGE AND THE SQUARE OF THE SLANT*
C*         RANGE TO THE SENSOR PLATFORM.  THEN ONE STANDARD NORMAL DEVIATIONS  *
C*         OF SENSOR POSITION ERRORS IN RANGE AND AZIMUTH ARE COMPUTED.        *
C*              UTILIZING THE SLOPE OF THE LINE JOINING THE GROUND STATION AND *
C*         THE SENSOR PLATFORM, X AND Y COMPONENTS OF RANGE AND AZIMUTH ERROR  *
C*         ARE COMPUTED.  THE X COMPONENTS OF RANGE AND AZIMUTH ARE ADDED TO   *
C*         GIVE AN X COMPONENT OF SENSOR POSITION OF TOTAL ERROR AND A Y COM-  *
C*         PONENT IS CALCULATED IN THE SAME MANNER.                            *
C*                                                                             *
C******************************************************************************
```

Figure 2.1-30 (Cont.)

```
C******************************* JFBLK3 *********************************
C*                                                                      *
C*                                                                      *
C*    PURPOSE                                                           *
C*        BLOCK DATA USED TO SET  DESIGNER INPUT VALUES FOR THE         *
C*        DETERMINATION OF GROUND TRUTH POSITIONS                       *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        NONE                                                          *
C*                                                                      *
C************************************************************************
```

Figure 2.1-31

```
C******************************** PREMN4 ********************************
C*                                                                      *
C*                PRE-RUN EXECUTIVE-STEP 4                              *
C*                                                                      *
C*          PURPOSE:                                                    *
C*                PLAYS MOVING ARRAYS THROUGH MANDUP AND MVSNE          *
C*                PRESENTLY MANDUP AND MVSNE ARE COMBINED IN SUB.  MVS  *
C*          INITIATES TARGETS FROM MOVE ARRAYS                          *
C*                                                                      *
C*                                                                      *
C*     USAGE                                                            *
C*         MAIN PROGRAM                                                 *
C*                                                                      *
C*     REMARKS                                                          *
C*       TRNPAR    ROUTINE  MAY BE CALLED AFTER STEP 4                  *
C*          REQUESTS DISC SPACE  FOR TARGET INFO -MTAPE(7)              *
C*                                                                      *
C*          SUBROUTINES CALLED                                          *
C*                    MVS -COMBINED MANDUP-MVSNE  -                     *
C*          SUBROUTINES REQUIRED                                        *
C*                    HYPERB                                            *
C*                    RHOTHE                                            *
C*                    DOPLER                                            *
C*                    NORMER                                            *
C*                    RERR                                              *
C*                    FINDX                                             *
C*                    FINDZ                                             *
C*                    MERGDR                                            *
C*                    DORDER                                            *
C*                    GMERGE                                            *
C*                                                                      *
C*     METHOD                                                           *
C*            THE COMMON GAME INFORMATION, THE MASTER DATA STREAM, THE  *
C*         ATMOSPHERIC DATA, THE UPDOWN TIMES, THE GROUND TRUTH ARRAY   *
C*         AND THE MASTER EVENT LIST ARE READ FROM THE CORRESPONDING    *
C*         UNITS.                                                       *
C*            AS THE ATMOSPHERIC DATA IS READ ONLY THE TIME, VISIBILITY *
C*         AND CEILING ARE SAVED IN ARRAY A(4,200). THE SOLAR ALTITUDE  *
C*         IS USED TO SET THE DAY/NIGHT CODE (0/1) IN A(4,N). ONE       *
C*         ADDITIONAL SET OF POINTS ARE RECORDED TO ACCOMODATE THE SCAN *
C*         LOGIC OF MVS.                                                *
C*            MVS IS CALLED                                             *
C*            THE TARGET INFORMATION IS PRINTED AND RECORDED ON MTAPE(7)*
C*         THE GROUND TRUTH POSITIONS ARE WRITTEN ON MTAPE(5) AND THE   *
C*         EVENTS ON MTAPE(4).                                          *
C*                                                                      *
C************************************************************************
```

Figure 2.1-32

```
C******************************** MVS     **********************************
C*                                                                         *
C*                         SUBROUTINE MVS                                  *
C*                                                                         *
C*    PURPOSE                                                              *
C*        DETERMINES THE GROUND TRUTH PATH OF THE MOVING ARRAYS            *
C*        AND THE UPDOWN TIMES OF THE ASSOCIATED SENSORS                   *
C*                                                                         *
C*    CALLING SEQUENCE                                                     *
C*        CALL MVS(MPRINT,NPLAY)                                           *
C*                                                                         *
C*    DESCRIPTION OF PARAMETERS                                            *
C*          * INPUT *                                                      *
C*        NPLAY    PLAY OPTION                                             *
C*        MPRINT   PRINT OPTION                                            *
C         ATMEN(1,M)   TIME ATMOSHERIC DATA VALID      M=1,MMAX             *
C                      ATMEN(1,MAX+1)  TMAX                                *
C         ATMEN(2,M)   VISIBILITY (METERS)                                 *
C         ATMEN(3,M)   CEILING     (METERS)                                *
C         ATMEN(4,M)   DAY= 0  , NIGHT= 1                                  *
C*                                                                         *
C*                                         PLANNER INPUT TABLE  ITEM *
C*        PRNV1( ,10) NAVIGATION SYSTEM  HYPERBOLIC          15    ALL  *
C*        PRNV2( , 7) NAVIGATION SYSTEM  RHO THETA           16    ALL  *
C*        PRNV3( , 7) NAVIGATION SYSTEM  DOPPLER             17    ALL  *
C*        PRNV4(     ) NAVIGATION SYSTEM  NORMAL             18    ALL  *
C*                                                                         *
C*                                                                         *
C*        NSEN      NUMBER OF SENSORS                        20     4   *
C*        ISN       ID OF FIRST SENSOR                       20     5   *
C*        IBF       ID OF BLUE FORCE                         20     6   *
C*        INAV      NAVIGATION SYSTEM                        20     7   *
C*        ICNV      VERSION OF NAVIGATION                    20     8   *
C*        PA        PROBABILITY OF MISSION ABORT             20     9   *
C*        DAYN      DAY/NIGHT RESTRAINT CODE                 20    10   *
C*        CEIL      MINIMUM CEILING                          20    11   *
C*        VIS       MINIMUM VISIBILITY                       20    12   *
C*        LFC       BLUE FORCE LOCATION IN DATA SET                     *
C*        TUP       START OPERATIONS TIME                    21     6   *
C*        NEL       NUMBER OF ELEMENTS                       21     5   *
C*        IFP       ID OF FORCE TYPE PARAMETER SET           21     7   *
C*        INA       FIRST MODE                               21    10   *
C*        IPH       ID OF ROUTE                              21     9   *
C*        INB       SECOND NODE                              21    10   *
C*        SP        SPEED                                    21    11   *
C*        ALT       ALTITUDE                                 21    12   *
C*        LFP       LOCATION OF FORCE PARAMETER SET                     *
C*        FM        FERROUS METAL PRESENT                    13     2   *
C*        VSD      VISUAL SECURITY DESCRIPTION               13     3   *
C*        SPC       SPACING BETWEEN ELEMENTS                 13     4   *
```

Figure 2.1-33

```
C*        IFRM    FORMATION DESCRIPTION                             13    5   *
C*        FD(LSP+5)  MEAN TIME BETWEEN FAILURES                      4    4   *
C*          * OUTPUT *                                                         *
C*        SXYT( ) ARRAY OF SENSOR SPACE TIME POSITIONS-XYTXYT-                *
C*        TARG( ) TARGET PARAMETERS -BLUE FORCE AS A TARGET                   *
C*        IVE(  ) MSM EVENTS 4,9                                               *
C*        UDTM( ) UP-DOWN TIMES OF SENSORS                                     *
C*        KSEN( ) POINTER FOR UP-DOWN TIMES                                    *
C*        KXYT( ) POINTER FOR POSITIONS                                        *
C*                                                                             *
C*    REMARKS                                                                  *
C*        REPAIR OF MOVING SENSORS NOT PLAYED                                  *
C*                                                                             *
C*    SUBROUTINES REQUIRED                                                     *
C*        FINDX                                                                *
C*        NORMER                                                               *
C*        GRN                                                                  *
C*        URN                                                                  *
C*        HYPERB                                                               *
C*        MERGDR                                                               *
C*        RHOTHE                                                               *
C*        DOPLER                                                               *
C*                                                                             *
C*                                                                             *
C*    METHOD                                                                   *
C*           THIS ROUTINE PROCESSES THE MOVING ARRAYS. THE PLATFORMS           *
C*        ARE CONSIDERED AS TARGETS. THE SENSOR POSITIONS ARE RECORDED         *
C*        AND THE RELIABILITY IS COMPUTED.                                     *
C*           THE DATA FOR THE PRNV TABLES IS LOCATED BY FINDX. IF A SET*
C*        IS VOID THE DATA IN THE DATA STATEMENTS WILL BE USED.                *
C*           FINDX IS USED TO LOCATE ALL THE DATA FOR THE MOVING ARRAYS*
C*        THE BLUE FORCE AND THE SENSORS. FOR EACH ARRAY THE PARAMETERS*
C*        ARE EXTRACTED FROM THE PLANNER INPUT.                                *
C*           THE STARTING TIME FOR THE MISSION IS DETERMINED BY NOTING *
C*        THE PLANNED STARTING TIME, THE MINIMUM CEILING AND VISIBILITY*
C*        AND THE DAY/NIGHT CONDITIONS. THE ATMOSPHERIC TABLES (ATMEN) *
C*        ARE CHECKED AND THE MINIMUM TIME AT OR AFTER THE PLANNED     *
C*        TIME WHEN ALL CONDITIONS ARE MET IS USED AS THE START TIME   *
C*        IF ALL CONDITIONS ARE NOT MET THE MISSION IS SCRATCHED. THE  *
C*        SCAN OF THE ATMEN  TABLE IS CONTINUED TO FIND THE LONGEST    *
C*        POSSIBLE DURATION OF THE MISSION.                             *
C*           THE GROUND TRUTH PATHS ARE COMPUTED AND THE TARGET PARA-   *
C*        METERS ARE SET ALONG WITH THE SENSOR GROUND TRUTH DATA. AS   *
C*        THE POSITIONS ARE FOUND THE DURATION IS DETERMINED. IF THE   *
C*        ATMOSPHERIC CONDITIONS ARE NOT MET FOR THIS INTERVAL THE     *
C*        MISSION IS SCRATCHED.  A RANDOM NUMBER IS DRAWN TO SEE IF THE*
C*        MISSION ABORTS. IF SO, A UNIFORM RANDOM NUMBER IS SELECTED TO*
C*        DETERMINE THE TIME OF THE ABORT.                              *
C*           IF AN ABORT OCCURS THE MISSION IS RETURNED TO ITS          *
C*        STARTING POINT ALONG A STRAIGHT LINE PATH.                    *
```

Figure 2.1-33 (Cont.)

```
C*          MSM EVENT TYPER IS CREATED.
C*          TARGET INFORMATION IS PUT INTO THE TARG ARRAY, SENSOR          *
C*      INFORMATION IS PUT INTO THE SXYT ARRAY AND POINTER KXYT IS         *
C*      SET.                                                               *
C*          THE MEAN FAILURE TIME OF THE SENSORS IS COMPUTED AND THE       *
C*      SENSOR UP-DOWN TIMES ARE STORED IN UDTM WITH POINTER KSEN.         *
C*      EVENT TYPE 4 IS CREATED.                                           *
C*          AFTER ALL ARRAYS ARE PROCESSED THE EVENTS ARE ORDERED AND      *
C*      MERGED WITH THE MASTER STREAM BY A CALL TO MERGOR.                 *
C*                                                                         *
C**************************************************************************
```

Figure 2.1-33 (Cont.)

External data sets required as input are "DATAIN" and "JFPXY", both generated in previous steps.

This step prepares the system parameter data set for use in MSM and stores this set on disc (JFFWDF).

Figures 2.1-34 through 2.1-37 describe the subroutines used in this step.

2.1.5.7 PRERUN Step 6

Step 6 in PRERUN comprises the main program (PREMN6) with 14 subroutines listed in Table 2.1-V, plus 5 utility subroutines.

External data sets required as input are "DATAIN" and "JFMSTR", both generated in previous steps.

Designer input values for Battle and Cultural are set by BLOCK DATA (JFBLK6). See volume II, Appendix I, for Designer Input Tables. In addition, a DATA statement within BSCHDL sets designer values for battlefield illumination parameters (array ELIGHT).

This step creates battle and cultural background noise levels in dB for PRERUN Step 7 and stores these values on disc (CRSEIS for battle values, and CRCLTP for cultural values). Step 6 also stores generated false targets on disc (CRTARG) and stores generated battle illumination events type 9 on disc (CREVT9).

The battle and culture processing is further described in Section 5 of this volume.

Figures 2.1-38 through 2.1-52 describe the subroutines of this step.

2.1.5.8 PRERUN Step 7

Step 7 in PRERUN comprises the main program (EVNT23) with 11 subroutines listed in Table 2.1-V, plus 4 utility subroutines.

External data sets required as input are "DATAIN", "MASSDAT", "CRSEIS", and "CRCLTB", all generated in previous steps.

This step creates events type 3 - sensor parameter changes due to (a) atmospheric variations and (b) background noise levels computed by battle and cultural routines (Step 6). Step 7 also creates events type 2 - false alarms. Event histories for these two event types (2 and 3) are merged in time sequence and stored on disc (MASSEV23).

Figures 2.1-53 through 2.1-60 describe the major subroutines of this step.

```
C****************************** TRNPAR *********************************
C*                                                                     *
C*                  PRE-RUN EXECUTIVE   STEP 5                         *
C*                                                                     *
C*    PURPOSE                                                          *
C*        EXECUTIVE PROGRAM  TO TRANSFER PLANNER INPUT DATA TO MSM     *
C*                                                                     *
C*    USAGE                                                            *
C*        MAIN PROGRAM                                                 *
C*                                                                     *
C*    METHOD                                                           *
C*            THE PLANNER INPUT DATA ARE READ  ND SUBROUTINE TRNPRI IS *
C*        CALLED. THE 13 DATA SETS ARE PRINTED UNDER CONTROL OF A DO   *
C*        LOOP.                                                        *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-34

```
C****************************** TRNPR1 ******************************
C*                                                                  *
C*                       SUBROUTINE TRNPR1                          *
C*                                                                  *
C*    PURPOSE                                                       *
C*        PROGRAM TO EXTRACT DATA FROM DATAIN AND TRANSFER TO MSM    *
C*                                                                  *
C*    CALLING SEQUENCE                                              *
C*        CALL TRNPR1                                               *
C*                                                                  *
C*    DESCRIPTION OF PARAMETERS                                     *
C*        ALL INPUT AND OUTPUT VIA COMMON BLOCKS/DATAIN,JTFWOF,PXYTP/ *
C*                                                                  *
C*    REMARKS                                                       *
C*        NONE                                                      *
C*                                                                  *
C*    SUBROUTINES REQUIRED                                          *
C*        ERASE                                                     *
C*        VALID                                                     *
C*        FINDX                                                     *
C*        FINDY                                                     *
C*        TRAN                                                      *
C*        TRAN2                                                     *
C*        DSKOUT                                                    *
C*                                                                  *
C*    METHOD                                                        *
C*            THE DATA IN THE PLANNER INPUT IS LOCATED BY FINDX AND *
C*        THE PARAMETERS ARE TRANSFERRED INTO THE ARRAY IFRY AS     *
C*        REQUIRED BY MSM. SUBROUTINE DSKOUT IS USED TO WRITE THE   *
C*        BINARY OUTPUT.                                            *
C*            THE SENSOR POSITIONS ARE FOUND BY READING THE GROUND  *
C*        TRUTH POSITIONS ON JFPXY, ARRAY SXYT WITH POINTER KXYT.   *
C*        SUBROUTINE FINDY IS USED TO FIND THE DATA FOR A PARTICULAR *
C*        SENSOR. SUBROUTINE VALID IS USED TO DETERMINE SENSOR TYPE *
C*                                                                  *
C********************************************************************
```

Figure 2.1-36

```
C****************************** DSKOUT **********************************
C*                                                                      *
C*                       SUBROUTINE DSKOUT                              *
C*                                                                      *
C*    PURPOSE                                                           *
C*       WRITES DATA ON BINARY UNIT                                     *
C*                                                                      *
C*    USAGE                                                             *
C*       CALL DSKOUT(M,N)                                               *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*       * INPUT *                                                      *
C*       M     LOCATION MINUS ONE OF ARRAY TO BE WRITTEN                *
C*       N     NUMBER OF VALUES FROM THE ARRAY TO BE WRITTEN            *
C*       ARRAY OF ITEMS TO BE WRITTEN ARE IN LABELED COMMON/JTFWDF/     *
C*                                                                      *
C*    REMARKS                                                           *
C*       THIS ROUTINE WRITES THE BINARY MSM INFORMATION ON UNIT 2       *
C*       IT IS USED SOLELY BY SUBROUTINE TRNPR1                         *
C*                                                                      *
C************************************************************************
```

Figure 2.1-36

```
C******************************* VALID ********************************
C*                                                                    *
C*                       SUBROUTINE VALID                             *
C*                                                                    *
C*    PURPOSE                                                         *
C*        ROUTINE TO DETERMINE VALID SENSOR  TARGET COMBINATIONS      *
C*        IT DETERMINES WHETHER LOS IS NECESSARY                      *
C*                                                                    *
C*    REMARKS                                                         *
C*        IT ASSIGNS A NUMBER TO EACH TYPE SENSOR AND EACH TYPE TARGET *
C*                                                                    *
C*        INPUTS REQUIRED BY SUBROUTINE                               *
C*                                                                    *
C*    SENSOR - SENSOR NAME RIGHT ADJUSTED                             *
C*    TARGET - TARGET NAME RIGHT ADJUSTED                             *
C*    VS     - SENSOR VELOCITY                                        *
C*    VT     - TARGET VELOCITY                                        *
C*    FM     - FERROUS METAL PRESENT                                  *
C*             INTEGER 1 = YES                                        *
C*             INTEGER 0 = NO                                         *
C*                                                                    *
C*        OUTPUTS RETURNED BY SUBROUTINE                              *
C*                                                                    *
C*    NS     - SENSOR NUMBER                                          *
C*    NT     - TARGET NUMBER                                          *
C*    NV1    - LINE OF SIGHT REQUIRMENT                               *
C*               =0  NO LOS OR BREAKWIR REQUIRED                      *
C*               =1  LINE OF SIGHT REQUIRED                           *
C*               =2  BREAKWIR REQUIRED                                *
C*    NV2    - PLAY TARGET IF NONZERO                                 *
C*               =0  NO PLAY                                          *
C*               =1  PERSONNEL                                        *
C*               =2  VEHICLES AND BOATS                               *
C*               =3  AIRCRAFT AND AMMO                                *
C*                                                                    *
C*    METHOD:  See 2-69.5.                                            *
C*                                                                    *
C**********************************************************************
```

Figure 2.1-37

```
C*     METHOD                                                            *
C*              IN A DO-LOOP, EACH SENSOR NAME IS COMPARED WITH THE NAME OF  *
C*         THE FIRST ITEM IN A TABLE CALLED STABLE.  IF THEY ARE THE SAME IT  *
C*         ASSIGNS A CODE (NS) TO THE SENSOR TYPE CORRESPONDING TO THE POSI-  *
C*         TION OF THE SENSOR IN THE STABLE LIST.  A CHECK IS THEN MADE TO SEE *
C*         IF THE TARGET TYPE IS GREATER THAN 19.  IF IT IS NOT, THEN THE TAR- *
C*         GET NUMBER (NT) IS SET TO THAT VALUE.  IF THE NUMBER IS GREATER    *
C*         THAN 19, THEN IT IS BEING READ AS AN ALPHANUMERIC AND A SIMILAR    *
C*         CHARACTER COMPARISON IS MADE BETWEEN THE TARGET NAME AND A TABLE   *
C*         CALLED TTABLE TO OBTAIN THE NT VALUE.                              *
C*              A LIST OF SENSORS AND TARGETS IS PRINTED OUT.                 *
C*              THE SENSOR TYPE (NS) IS THEN USED IN A LOOK-UP TABLE TO DETER- *
C*         MINE LINE-OF-SIGHT REQUIREMENT.  VALUES OF 0, 1, 2 ARE ASSIGNED TO *
C*         SHOW LOS NOT REQUIRED, LOS REQUIRED, OR THAT DEVICE IS A BREAKWIRE *
C*         TYPE DEVICE, RESPECTIVELY.  THE TARGET TYPE (NT) AND SENSOR TYPE   *
C*         (NS) ARE THEN USED IN A LOOK-UP TABLE (CTABLE) TO DETERMINE TARGET/ *
C*         SENSOR COMPATIBILITY.  IF CTABLE SHOWS COMPATIBILITY, ADDITIONAL   *
C*         CHECKS ARE MADE FOR TARGET MOVEMENT REQUIREMENTS AS SPECIFIED IN   *
C*         TABLE DCODE.  DCODE = 0 INDICATES TARGET MUST BE STATIONARY.       *
C*         DCODE = 1 INDICATES TARGET MUST BE MOVING.  DCODE = 2 INDICATES    *
C*         TARGET MAY BE EITHER MOVING OR STATIONARY.  CHECKS ARE ALSO MADE   *
C*         FOR REQUIREMENT FOR FERROUS METAL TO BE PRESENT IN THE TARGET AND  *
C*         FOR THE TARGET VELOCITY TO BE GREATER THAN .035 M/SEC TO BE PICKED *
C*         UP BY MTI RADARS.                                                  *
C*                                                                            *
C*****************************************************************************
```

Figure 2.1-37 (Cont.)

```
C****************************** PREMN6 **********************************
C*                                                                     *
C*                   PRERUN EXECUTIVE -     STEP  6                    *
C*                                                                     *
C*    PURPOSE                                                          *
C*        INITIALIZES TAPES AND PARAMETERS USED BY BATTLE AND CULTURE  *
C*    ROUTINES AND CONTAINS THE CALL STATEMENTS TO THE EXECUTIVE       *
C*    BATTLE AND CULTURE SUBROUTINES.                                  *
C*                                                                     *
C*    USAGE                                                            *
C*        MAIN PROGRAM                                                 *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        ALL INPUT AND OUTPUT VIA LABELED COMMON BLOCKS/ ROUTE,EXCLUD,*
C*        DATAIN,UTMCOM,TIMES,OUTP,INOUT,OPTION,POSERR,BBNDS,SNDXDY,   *
C*        FSBASE/                                                      *
C*                                                                     *
C*    REMARKS                                                          *
C*        NONE                                                         *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*        FINDX                                                        *
C*        CULTEX                                                       *
C*        BATEX                                                        *
C*                                                                     *
C*    METHOD                                                           *
C*            THE COMMON GAME INFORMATION AND THE PLANNER INPUT TABLES *
C*        ARE READ. AN END OF FILE IS WRITTEN ON THE UNIT FOR FALSE    *
C*        TARGETS. THIS INSURES THAT LATER ROUTINES WILL RUN IF NO TYPE*
C*        TARGETS ARE GENERATED BY BATTLE, OR CULTURE.                 *
C*            THE PLANNER INPUT DATA FOR THE PATHS AND FOR THE EXCLUDED *
C*        AREA IS LOCATED BY FINDX AND PUT INTO THE APPROPRIATE ARRAYS *
C*            THE GAME BOUNDARIES ARE SET AND THE NUMBER OF DAYS FOR   *
C*        THE GAME AND THE NUMBER OF QUARTERS(OF A DAY) ARE SET        *
C*            THE SUB-EXECUTIVE ROUTINES TO PROCESS CULTURE AND BATTLE *
C*        ARE CALLED. THE ROUTINES ARE WRITTEN SO THAT EITHER MAY BE   *
C*        CALLED FIRST.                                                *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-38

```
C******************************  CULTEX  *********************************
C*                                                                      *
C*                           SUBROUTINE CULTEX                          *
C*                                                                      *
C*    PURPOSE                                                           *
C*    THE EXECUTIVE ROUTINE THAT CALLS SUBROUTINES CSCHDL,CULTBK,       *
C*    BATLTG FOR PROCESSING THE CULTURE EVENTS DESCRIBED BY THE         *
C*    PLANNER.                                                          *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL CULTEX(NDAYS,DTIME)                                      *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*        ALL INPUT AND OUTPUT VIA LABELED COMMON BLOCKS, DATAIN,PAR,   *
C*    SUBCUL,SNOXDY,ROUTE,TARG,OUTP,INOUT.                              *
C*                                                                      *
C*         * INPUTS *                                                   *
C*                                                                      *
C*        NDAYS      NUMBER OF DAYS                                     *
C*        DTIME      TIME INCREMENT(SECONDS)                            *
C*                                                                      *
C*         * OUTPUTS *                                                  *
C*                                                                      *
C*        TRGTS(I,J), I=1,13),J=1,NTR TARGETS DETECTED                  *
C*    REMARKS                                                           *
C*        NONE                                                          *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*        BATLTG,CSCHDL,CULTBK,                                         *
C*                                                                      *
C*                                                                      *
C*    METHOD                                                            *
C*            THE TARGET DISK IS ADVANCED. FINDX IS CALLED TO LOCATE    *
C*        THE PLANNER INPUT FOR THE CULTURE ROUTINES                    *
C*        THE DETERMINISTIC EVENTS ARE PROCESSED FIRST, AND THE RANDOM  *
C*        EVENTS LAST. AS EACH EVENT IS PROCESSED THE BATTLE CULTURE    *
C*        TARGET ROUTINE IS CALLED TO GENERATE FALSE TARGETS            *
C*            AFTER ALL CULTURE EVENTS HAVE BEEN PROCESSED THE CULTURE  *
C*        BACKGROUND ROUTINE -CULTBK - IS CALLED TO COMPUTE BACKGROUND  *
C*        NOISE LEVELS AND END OF FILE CONDITIONS ARE RECORDED          *
C*                                                                      *
C************************************************************************
```

Figure 2.1-39

```
C****************************** BATEX *********************************
C*                                                                    *
C*                        SUBROUTINE  BATEX                           *
C*                                                                    *
C*    PURPOSE                                                         *
C*        THE EXECUTIVE ROUTINE THAT CALLS THE VARIOUS ROUTINES FOR   *
C*    PROCESSING THE BATTLE EVENTS AND DOES THE INITIALIZATION        *
C*    REQUIRED.                                                       *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL BATEX(NDAYS,DTIME, CSSAC)                              *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        ALL INPUT AND OUTPUT, EXCEPT AS NOTED BELOW, VIA COMMON     *
C*    BLOCKS DATAIN, BATTEL, BB, ROUTE, EXCLUD, BBNOS, EVENTS, TARG,  *
C*    INOUT, OUTP, FSBASE.                                            *
C*                                                                    *
C*         *  INPUT  *                                                *
C*        NDAYS      NUMBER OF DAYS                                   *
C*        DTIME      TIME INCREMENT(SECONDS)                          *
C*        CSSAC      NOISE RATIO FADE CONSTANT FOR SEISMIC AND        *
C*                   ACOUSTIC SENSORS                                 *
C*                                                                    *
C*    REMARKS                                                         *
C*        NONE                                                        *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        FINDX                                                       *
C*        BSCHDL                                                      *
C*        BATLTG                                                      *
C*        BATLBK                                                      *
C*        EVNT9                                                       *
C*        MERGDR                                                      *
C*                                                                    *
C*    METHOD                                                          *
C*        THE TARGET DISK IS ADVANCED.  FINDX IS CALLED TO LOCATE     *
C*        THE PLANNER INPUT  FOR THE BATTLE ROUTINES.                 *
C*        THE DETERMINISTIC EVENTS ARE PROCESSED FIRST, AND THE       *
C*        RANDOM EVENTS LAST. AS EACH EVENT IS PROCESSED THE BATTLE   *
C*        CULTURE TARGET ROUTINE IS CALLED TO GENERATE FALSE TARGETS. *
C*        THE BATTLE BACKGROUND ROUTINE(BATLBK) IS CALLED TO COMPUTE  *
C*        BACKGROUND NOISE LEVELS AND A CALL TO ROUTINE (EVNT9)       *
C*        DETERMINES ANY EVENT DUE TO ILLUMINATION.                   *
C*                                                                    *
C**********************************************************************
```

Figure 2.1-40

```
C*                            BATLBK                                  *
C*                                                                    *
C*                      SUBROUTINE BATLBK                             *
C*                                                                    *
C*    PURPOSE                                                         *
C*        DETERMINES THE SEISMIC AND ACOUSTIC BACKGROUND NOISE,       *
C*    IF ANY.                                                         *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL BATLBK(SCHDL,ACST,SEIS,CSSAC,DTIME,NBMAX)              *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        ALL INPUT AND OUTPUT VIA LABELED COMMON BLOCKS/BATTEL, BB,  *
C*    EXCLUD, ROUTE, BBNDS/  EXCEPT AS NOTED BELOW.                   *
C*                                                                    *
C*        *  INPUT  *                                                 *
C*        SCHDL      BATTLE SCHEDULE TABLE                            *
C*        CSSAC      NOISE RATIO FADE CONSTANT = 0.99                 *
C*        DTIME      TIME INCREMENT = 3600 SECS.                      *
C*        NBMAX      MAXIMUM NUMBER OF TIME PERIODS TO BE PERMITTED   *
C*                                                                    *
C*        *  OUTPUT  *                                                *
C*        SEIS       BATTLE BACKGROUND NOISE DETECTED BY SEISMIC SENSOR *
C*        ACST       BATTLE BACKGROUND NOISE DETECTED BY ACOUSTIC SENSOR*
C*                                                                    *
C*    REMARKS                                                         *
C*        BACKGROUND NOISE IS IN DECIBELS.                            *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        NONE                                                        *
C*                                                                    *
C*    METHOD                                                          *
C*        FOR ALL ACOUSTIC AND SEISMIC SENSORS A BACKGROUND NOISE     *
C*        LEVEL IS COMPUTED AS A FUNCTION OF NUMBER OF VOLLEYS FOR    *
C*        EXPLOSIVE DEVICES AND AS A FUNCTION OF NUMBER OF VEHICLES,  *
C*        TIME, AND DURATION FOR MILITARY GROUND VEHICLES AND AIRCRAFT. *
C*        THE PLANNER TABLE ZNOMAS  (NOMINAL AFFECT ON SENSOR OF A PARTICULAR *
C*        TYPE EVENT) IS USED TO COMPUTE THE NOISE LEVELS.            *
C*                                                                    *
C*                                                                    *
C*********************************************************************
```

Figure 2.1-41

```
C****************************  BATLTG  ************************************
C*                                                                       *
C*                        SUBROUTINE BATLTG                              *
C*                                                                       *
C*     PURPOSE                                                           *
C*         DEFINES THE BATTLE OR CULTURE TARGET POSITION, TIME,          *
C*     ALTITUDE,VELOCITY, LEG OF PATH, AND VISUAL SECURITY DESCRIPTIVE   *
C*     PARAMETER.                                                        *
C*                                                                       *
C*     CALLING SEQUENCE                                                  *
C*         CALL BATLTG(NCALL,C,ROUTX,ROUTY,NRT)                          *
C*                                                                       *
C*     DESCRIPTION OF PARAMETERS                                         *
C*         ALL INPUT AND OUTPUT,EXCEPT AS NOTED BELOW, VIA LABELED       *
C*     COMMON BLOCKS TARG, OUTP.                                         *
C*                                                                       *
C*         * INPUT *                                                     *
C*         NCALL      NCALL = 0 BATTLE TARGET                            *
C*                    NCALL = 1 CULTURE TARGET                           *
C*         C          TABLE SCHDL IF BATTLE TARGET                       *
C*                    TABLE CSHDL IF CULTURE TARGET                      *
C*         ROUTX      X COORDINATE OF PLANNER TARGET PATH NODE           *
C*         ROUTY      Y COORDINATE OF PLANNER TARGET PATH NODE           *
C*         NRT        NUMBER OF TARGETS                                  *
C*                                                                       *
C*     REMARKS                                                           *
C*         NONE                                                          *
C*                                                                       *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*         NONE                                                          *
C*                                                                       *
C*     METHOD                                                            *
C*             TARGET PARAMETERS FOR FALSE TARGETS FROM BATTLE AND       *
C*         CULTURE ARE SET. THE EVENT CODE IS CHECKED TO DETERMINE       *
C*         THE TYPE AND TIME PARAMETERS OF THE TARGET.                   *
C*                                                                       *
C*************************************************************************
```

Figure 2.1-42

```
C******************************  BSCHDL  ********************************
C*                                                                      *
C*                        SUBROUTINE BSCHDL                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*        COMPLETES THE BATTLE SCHEDULE TABLE (SCHDL) FOR EACH EVENT   *
C*    SCHEDULED BY THE PLANNER. COMBINED ALL RANDOM EVENTS THAT ARE    *
C*    SCHEDULED WITH PLANNER INPUT.                                     *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL BSCHDL(S)                                                *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*        ALL INPUT AND OUTPUT VIA LABELED COMMON BLOCKS BATTEL, BB,   *
C*    ROUTE,EXCLUD, FSBASE, BBNDS, INOUT.                               *
C*                                                                      *
C*        *  INPUT  *                                                   *
C*        S    EVENT SCHEDULE TABLE OF PLANNER SCHDL                    *
C*                                                                      *
C*    REMARKS                                                           *
C*        NONE                                                          *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*        BEMAP                                                         *
C*        NUMBER                                                        *
C*        PATHS                                                         *
C*        SELCTR                                                        *
C*                                                                      *
C*    METHOD                                                            *
C*                                                                      *
C*    METHOD                                                            *
C*        ANY PLANNER INPUT VALUE THAT IS NOT SPECIFIED IS SET         *
C*        RANDOMLY.                                                     *
C*                                                                      *
C************************************************************************
```

Figure 2.1-43

```
C****************************** CSCHDL ********************************
C*                                                                    *
C*                        SUBROUTINE  CSCHDL                          *
C*                                                                    *
C*    PURPOSE                                                         *
C*        TO COMPLETE THE CULTURE SCHEDULE FOR EACH TYPE OF EVENT     *
C*    SCHEDULED BY THE PLANNER.  COMBINES ALL RANDOM EVENTS THAT ARE  *
C*    SCHEDULED WITH PLANNER INPUT.                                   *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL CSCHDL(C)                                              *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        ALL INPUT EXCEPT SCHEDULE TABLE(C) VIA LABELED COMMON       *
C*    AREAS SUBCUL, PAR, SNDXDY, ROUTE, INOUT.                        *
C*                                                                    *
C*        *  OUTPUT  *                                                *
C*        C     CULTURE SCHEDULE TABLE CSHDL                          *
C*                                                                    *
C*    REMARKS                                                         *
C*        NONE                                                        *
C*                                                                    *
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*        NUMBER                                                      *
C*        PATHS                                                       *
C*                                                                    *
C*    METHOD                                                          *
C*                                                                    *
C*    METHOD                                                          *
C*        ANY PLANNER INPUT VALUE THAT IS NOT SPECIFIED IS SET        *
C*        RANDOMLY                                                    *
C*                                                                    *
C**********************************************************************
```

Figure 2.1-44

```
C****************************** CULTBK ******************************
C*                                                                  *
C*                      SUBROUTINE  CULTBK                          *
C*                                                                  *
C*    PURPOSE                                                       *
C*        TO DETERMINE THE SEISMIC AND ACOUSTIC CULTURE BACKGROUND  *
C*        NOISE,IF ANY.                                             *
C*                                                                  *
C*    CALLING SEQUENCE                                              *
C*        CALL CULTBK(CSHDL,CLTBGR,NCEV)                            *
C*                                                                  *
C*    DESCRIPTION OF PARAMETERS                                     *
C*        INPUTS VIA COMMON AREAS DATAIN, SUBCUL, PAR, OUTP, INOUT. *
C*                                                                  *
C*        *  INPUT  *                                               *
C*        CSHDL      PLANNER CULTURE SCHEDULE TABLE                 *
C*        NCEV       NUMBER OF CULTURAL EVENTS                      *
C*                                                                  *
C*        *  OUTPUT *                                               *
C*                                                                  *
C*        CLTBGR     ARRAY OF CULTURE BACKGROUND NOISES(DECIBELS)   *
C*                                                                  *
C*    REMARKS                                                       *
C*        NONE                                                      *
C*                                                                  *
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                 *
C*        SENXY                                                     *
C*        FINDX                                                     *
C*                                                                  *
C*    METHOD                                                        *
C*        FOR ALL ACOUSTIC AND SEISMIC SENSORS A BACKGROUND NOISE   *
C*        LEVEL IS COMPUTED AS A FUNCTION OF THE DISTANCE OR THE    *
C*        SOURCE FROM THE SENSOR. SUBROUTINE SENXY IS USED TO LOCATE *
C*        THE SENSORS. DESIGNER INPUT TABLE CEVDBA IS USED TO SET   *
C*        NOISE LEVELS.                                             *
C*                                                                  *
C********************************************************************
```

Figure 2.1-45

```
C**********************************************************************
C*                                                                    *
C****************************  EVNT9  *********************************
C*     PURPOSE                                                        *
C*         ADDS TO THE LIST OF EVENTS ANY EVENT DUE TO ILLUMINATION   *
C*         OF THE PLAY AREA                                           *
C*                                                                    *
C*     CALLING SEQUENCE                                               *
C*         CALL EVNT9(SCHDL)                                          *
C*                                                                    *
C*     DESCRIPTION OF PARAMETERS                                      *
C*         * INPUT *                                                  *
C*         SCHDL     SCHEDULE ARRAY FOR A PARTICULAR TYPE EVENT       *
C*         * OUTPUT *                                                 *
C*         ALL OUTPUT VIA LABELED COMMON BLOCK/EVENTS/.               *
C*                                                                    *
C*     REMARKS                                                        *
C*         USED EXCLUSIVELY BY SUBROUTINE BATEX                       *
C*                                                                    *
C*     SUBROUTINES REQUIRED                                           *
C*         NONE                                                       *
C*                                                                    *
C**********************************************************************
```

Figure 2.1-46

```
C******************************** PATHS ********************************
C*                                                                     *
C*                         SUBROUTINE PATHS                            *
C*                                                                     *
C*    PURPOSE                                                          *
C*       DETERMINES A PATH FOR THE EVENT FROM PLANNER ROUTX, ROUTY      *
C*    TABLES.                                                          *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*     (A) CALL PATHS(S(4),S(5),S(8),S(9),S(10),IV2,NCALL,IROUTX,      *
C*                                                      ROUTX,ROUTY)   *
C*     (B) CALL PATHS(C(4),C(5),C(6),C(7),C8, IV2,NCALL,IROUTX,        *
C*                                                     ROUTX,ROUTY)    *
C*       FORM (A) ABOVE IS CALLING SEQUENCE USED IN SUBROUTINE BSCHDL  *
C*       FORM (B) ABOVE IS CALLING SEQUENCE USED IN SUBROUTINE CSCHDL  *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*         *  INPUT  *                                                 *
C*       IV2        TYPE OF EVENT FROM PLANNER TABLE                   *
C*       NCALL      NCALL=0 BATTLE EVENT, NCALL=1 CULTURE EVENT        *
C*       IROUTX     NUMBER OF ROUTES GIVEN BY PLANNER                  *
C*       ROUTX      X COORD. OF PLANNER PATH POINT                     *
C*       ROUTY      Y COORD. OF PLANNER PATH POINT                     *
C*       S(4)       X COORD. OF BATTLE EVENT, START OF EVENT PATH      *
C*       C(4)       X COORD. OF CULTURE EVENT, START OF EVENT PATH     *
C*       S(5)       Y COORD. OF BATTLE EVENT , START OF EVENT PATH     *
C*       C(5)       Y COORD. OF CULTURE EVENT, START OF EVENT PATH     *
C*                                                                     *
C*         *  OUTPUT  *                                                *
C*       S(8)       X COORD. OF END OF PATH FOR BATTLE EVENT           *
C*       S(9)       Y COORD. OF END OF PATH FOR BATTLE EVENT           *
C*       S(10)      RANDOM SELECTION OF A PATH START POINT IF          *
C*                  NOT GIVEN IN PLANNER SCHEDULE TABLE                *
C*                                                                     *
C*       C(6)       X COORD. OF END OF PATH FOR CULTURE EVENT          *
C*       C(7)       Y COORD. OF END OF PATH FOR CULTURE EVENT          *
C*       C(8)       TYPE OF PATH ,TRAIL, ROADWAY,WATERWAY ETC.         *
C*       C8         C(8)/100.                                          *
C*    REMARKS                                                          *
C*    NONE                                                             *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*       NONE                                                          *
C*                                                                     *
```

Figure 2.1-47

```
C*    METHOD                                                            *
C*       IF A PATH IS NOT GIVEN BY THE PLANNER EVENT TABLE ONE IS       *
C*       RANDOMLY SELECTED. THIS PATH IS CHECKED TO DETERMINE ITS       *
C*       VALIDITY FOR TYPE EVENT BEING CONSIDERED.  START AND END       *
C*       COORDINATES OF THE PATH ARE CHOSEN FROM THE PLANNER DESIGN     *
C*       TABLES(ROUTX,ROUTY) IF WITHIN A 10 METER RADIUS OF THE START   *
C*       AND END COORDINATES GIVEN BY THE PLANNER EVENT SCHEDULE        *
C*       TABLE. IF NO START AND END COORDINATES ARE GIVEN THE           *
C*       PROGRAM DEFINES SOME BY RANDOM SELECTION.                      *
C*                                                                      *
C***********************************************************************
C***********************************************************************
```

Figure 2.1-47 (Cont.)

```
C***************************** SENXY ********************************
C*                                                                    *
C*                      SUBROUTINE SENXY                              *
C*                                                                    *
C*    PURPOSE                                                         *
C*        DETERMINES SENSOR LOCATION COORDINATES (X,Y)                *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL SENXY(SENS,IN)                                         *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        INPUT VIA LABELED COMMON DATAIN                             *
C*                                                                    *
C*           *  INPUTS  *                                             *
C*                                                                    *
C*        IN    SENSOR IDENTIFYING INDEX                              *
C*                                                                    *
C*           *  OUTPUT  *                                             *
C*                                                                    *
C*        SENS(2)   X COORDINATE  OF SENSOR                           *
C*        SENS(3)   Y COORDINATE  OF SENSOR                           *
C*                                                                    *
C*    REMARKS                                                         *
C*        NONE                                                        *
C*                                                                    *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS                             *
C*        NONE                                                        *
C*                                                                    *
C*                                                                    *
C*    METHOD                                                          *
C*        THE X,Y COORDINATES OF THE SENSORS IS DETERMINED. PLANNED   *
C*        X,Y POSITIONS ARE USED.                                     *
C*                                                                    *
C**********************************************************************
```

Figure 2.1-48

```
C*************************************************************************
C**************************** NUMBER* ************************************
C*     PURPOSE                                                           *
C*         TO SELECT A UNIFORM RANDOM NUMBER FROM 1-10 FOR PURPOSES OF   *
C*         INCREASING A GIVEN NUMBER(K) AS FOLLOWS,                      *
C*         K = K        WHEN RAND.NO. <= 6                               *
C*         K = K + 1    WHEN RAND.NO. > 6                                *
C*         K = K + 2    WHEN RAND.NO. > 9                                *
C*                                                                       *
C*     USAGE                                                             *
C*         K = NUMBER(K)                                                 *
C*                                                                       *
C*     DESCRIPTION OF PARAMETERS                                         *
C*         K     ANY INTEGER FROM 1 - 3                                  *
C*                                                                       *
C*     METHOD                                                            *
C*         A NUMBER IS DETERMINED WITH A 60 PERCENT PROBABILITY          *
C*          OF BEING EQUAL TO K,A 30 PERCENT PROBABILITY OF BEING ONE OVER K AND  *
C*          A 10 PERCENT PROBABILITY OF BEING TWO MORE THAN K.           *
C*                                                                       *
C*                                                                       *
C*************************************************************************
```

Figure 2.1-49

* Used in BATTLE and CULTURE to randomly pick value from tables acccrding to distribution 60-30-10.

```
C*********************************** SELCTR ************************************
C*                                                                            *
C*                          SUBROUTINE SELCTR                                 *
C*                                                                            *
C*    PURPOSE                                                                 *
C*         THIS SUBROUTINE IS USED BY SUBROUTINE BSCHDL TO SELECT VALUES OF   *
C*    MILITARY VEHICLE SPEED, NUMBER, AND SPACING VARIABLES FOR THE BATTLE    *
C*    SCHEDULE TABLE (SCHDL) WHEN IT IS DESIRED TO TREAT THESE VARIABLES AS   *
C*    RANDOM.                                                                 *
C*                                                                            *
C*    CALLING SEQUENCE                                                        *
C*         CALL SELCTR (N,NX,INV,JN,SS,Y)                                     *
C*                                                                            *
C*    DESCRIPTION OF PARAMETERS                                               *
C*         *  INPUT  *                                                        *
C*         N   - NUMBER OF BATTLE EVENT CLASSES IN ARRAY Y FROM WHICH VALUES  *
C*               OF THE RANDOM VARIABLE ARE SELECTED.                         *
C*         NX  - NUMBER OF VALUES OF RANDOM VARIABLE IN EACH OF THE N EVENT   *
C*               CLASSES OF ARRAY Y.                                          *
C*         IVN - BATTLE EVENT, EVID, NUMBER CORRESPONDING TO BATTLE           *
C*               EVENT TYPE 18 AND CLASS 1, 2, 3, 4, OR 5.                    *
C*         JN  - INDEX FOR DEFAULT OPTION OF VARIABLE.                        *
C*         Y   - DESIGNER DATA SET FROM WHICH RANDOM SELECTION IS MADE,       *
C*               EITHER VSPED, CNVOY, OR SPACE.                               *
C*                                                                            *
C*         *  OUTPUT  *                                                       *
C*                                                                            *
C*         SS  - VALUE OF RANDOM VALUE SELECTED FROM ARRAY Y AND STORED IN    *
C*               TABLE SCHDL.                                                 *
C*                                                                            *
C*    METHOD                                                                  *
C*                                                                            *
C*              WHEN IT IS DESIRED TO PLAY A MILITARY VEHICLE BATTLE EVENT    *
C*         WITH RANDOM VEHICLE SPEED, RANDOM NUMBER OF VEHICLES IN A CONVOY,  *
C*         OR RANDOM SPACING BETWEEN VEHICLES, THIS SUBROUTINE IS USED TO SE- *
C*         LECT A VALUE FROM THE APPROPRIATE DESIGNER INPUT DATA SET; E.G.,   *
C*         VEHICLE SPEED SET (VSPED), CONVOY SIZE SET (CNVOY), AND SPACING    *
C*         BETWEEN VEHICLES SET (SPACE).  EACH DATA SET PROVIDES FOR UP TO    *
C*         FIVE CLASSES OR SIZES OF MILITARY VEHICLES. VSPED IS DIMENSIONED   *
C*         FOR THREE SPEEDS; CNVOY IS DIMENSIONED FOR EIGHT CONVOY SIZES, AND *
C*         SPACE IS DIMENSIONED FOR FOUR VALUES OF SPACE BETWEEN CENTERS OF   *
C*         VEHICLES.                                                          *
C*                                                                            *
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                           *
C*                                                                            *
C*         URN                                                                *
C*                                                                            *
C******************************************************************************
```

Figure 2.1-50

```
C**************************************************************************
C******************************JBLK6***************************************
C*                                                                        *
C*    PURPOSE                                                             *
C*        BLOCK DATA USED TO SET DESIGNER INPUT VALUES USED TO            *
C*        COMPLETE BATTLE AND CULTURAL SCHEDULE TABLES                    *
C*                                                                        *
C*    CALLING SEQUENCE                                                    *
C*        NONE                                                            *
C*                                                                        *
C*    DESCRIPTION OF PARAMETERS                                           *
C*        ALL INPUT VIA LABELED COMMON BLOCKS/BATTEL,BB,EXCLUD,           *
C*        SUBCUL,PAR/                                                     *
C*                                                                        *
C**************************************************************************
```

Figure 2.1-51

```
C****************************** REMAP ***********************************
C*                                                                      *
C*     PURPOSE                                                          *
C*         TO DETERMINE IF A GIVEN EVENT OCCURS WITHIN A SPECIFIED      *
C*         REGION OF THE PLAY AREA                                      *
C*                                                                      *
C*     USAGE                                                            *
C*         Y = REMAP(A,B,C)                                             *
C*                                                                      *
C*     DESCRIPTION OF PARAMETERS                                        *
C*         * INPUT *                                                    *
C*         A     EVENT IDENTIFICATION CODE                              *
C*         B     X COORDINATE OF EVENT                                  *
C*         C     Y COORDINATE OF EVENT                                  *
C*         ALL OTHER REQUIRED INPUT VIA LABELED COMMON BLOCKS/BATTEL,   *
C*         BB,ROUTE,BNDS,EXCLUD,INOUT/.                                 *
C*                                                                      *
C*     REMARKS                                                          *
C*         NONE                                                         *
C*                                                                      *
C*     METHOD                                                           *
C*         IF THE COORDINATES OF A SPECIFIC EVID CODE ARE IN AN EXCLUDED*
C*         AREA THE PROGRAM REJECTS THE CHOICE BY RETURNING A ZERO      *
C*         IF THE CODE IS NOT FOUND THE PROGRAM WILL PRINT AN ERROR     *
C*         MESSAGE AND ACCEPT THE CHOICE.                               *
C*                                                                      *
C*                                                                      *
C*                                                                      *
C*                                                                      *
C*                                                                      *
C*                                                                      *
C************************************************************************
```

Figure 2.1-52

```
C*********************************** EVNT23 *****************************
C*                                                                       *
C*                PRE-RUN EXECUTIVE    STEP 7                            *
C*                                                                       *
C*    PURPOSE                                                            *
C*       TO CREATE EVENT TYPE 2 (FALSE ALARMS) AND EVENT TYPE 3         *
C*       (SENSOR PARAMETER CHANGES DUE TO BACKGROUND ENVIRONMENT).      *
C*                                                                       *
C*    CALLING SEQUENCE                                                   *
C*       MAIN PROGRAM                                                    *
C*                                                                       *
C*    REMARKS                                                            *
C*       PROGRAM REQUIRES PLANNER INPUT, BATTLE, CULTURAL, BASICT,       *
C*       AND ATMENV DATA SETS.                                           *
C*                                                                       *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED.                      *
C*       FINDX    LOCATES POINTER FOR PLANNER INPUT DATA SETS.           *
C*       TERAN    GENERATES UNTER AND UTVSXY DATA IN LABELLED COMMON.    *
C*       VALID    VALIDATES SENSOR - TARGET COMBINATIONS                 *
C*       IUTFVL   LOCATES INDEX ON UNIT TERRAIN                          *
C*       SEISBK   COMPUTES BACKGROUND CONDITIONS FOR SEISMIC  SENSORS.*
C*       ACOUBK   COMPUTES BACKGROUND CONDITIONS FOR ACOUSTIC SENSORS.*
C*       ARFBK    COMPUTES BACKGROUND CONDITIONS FOR ARFBUOY  SENSORS.*
C*       PIRBK    COMPUTES BACKGROUND CONDITIONS FOR PIRID    SENSORS.*
C*       BWIRBK   COMPUTES BACKGROUND CONDITIONS FOR BREAKWIR SENSORS.*
C*       ENVIR    COMPUTES BACKGROUND TEMPERATURE.                       *
C*       TRNSFR   TRANSFERS ARRAY BLOCKS FROM ONE SET TO ANOTHER.        *
C*       FAINTV   PROVIDES RANDOM FALSE ALARM TIME INTERVALS.            *
C*       MERGE1   MERGES AND ORDERS BY TIME 2 DATA SETS.                 *
C*                                                                       *
C*************************************************************************
```

Figure 2.1-53

```
C*********************************** SEISBK ***************************
C*
C*                          SUBROUTINE SEISBK
C*
C*      PURPOSE:
C*           THIS ROUTINE IS USED DURING PRERUN, TO ESTABLISH OPER-
C*           ATIONAL PARAMETERS FOR SEISMIC SENSORS  (THESE PARAM-
C*           ETERS TO BE USED DURING MSM STAGE BY SUBROUTINE SEISTG).
C*           OUTPUT PARAMETERS ARE--
C*                   AMPLIFIER GAIN
C*                   THRESHOLD VOLTAGE
C*                   RMS BACKGROUND NOISE VOLTAGE
C*                   AVERAGE TIME BETWEEN THRESHOLD CROSSINGS
C*
C*      USAGE:
C*             CALL SEISBK(IUT,DBCULT,DBBATL,ISEXP,ITREE,IFIXGN,
C*                          THRESH,GEQUIL,VNOISE,AVGTHC)
C*
C*      DESCRIPTION OF PARAMETERS
C*           SEE GLOSSARY BELOW
C*
C*      REMARKS:
C*                NONE
C*
C*      SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED:
C*           ALOG10     SUPPLIED BY FORTRAN
C*           SQRT       SUPPLIED BY FORTRAN
C*
C*      METHOD:
C*             BACKGROUND LEVELS ARE ASCRIBED TO THE COMMON SOURCES OF
C*      BACKGROUND MICROSEISMIC LEVEL. THESE LEVELS ARE MODIFIED BY
C*      ENVIRONMENTAL FACTORS SUCH AS FOLIAGE DENSITY, SOIL WETNESS AND THE
C*      LIKE.  THE SENSOR RESPONSE TO THE NOISE LEVEL IS COMPUTED GIVING
C*      AN RMS NOISE VOLTAGE AGAINST WHICH SIGNALS MUST COMPETE AND
C*      FROM WHICH THE AMPLIFIER AUTOMATIC GAIN SETTING AND FALSE ALARM
C*      RATE ARE DETERMINED.
C*
C*      GLOSSARY:
C                                  INPUT VALUES
C DBBATL SEISBK CP   BATTLE NOISE     (DB)
C DBCULT SEISBK CP   CULTURAL NOISE  (DB)
C IFIXGN SEISBK CP   =0 NO FIXED GAIN,=1,2,3,4,5 PLANNER SET GAIN,=6 SEL. GAIN
C ISEXP  SEISBK CP   INDEX FOR BURIED, 1=0.0(BURIED), 2=6.0(NOT BURIED)
C ITREE  SEISBK CP   INDEX ON TREE DISTANCE 0= TREE, 1= NO TREE.
C IUT    SEISBK CP   INDEX ON UNIT TERRAIN
C
C                              LABELLED COMMON INPUTED VALUES
C BIASSF SENVAR      THRESHOLD SETTING (VOLTS)
C BWSEIS SENVAR      EFFECTIVE BAND WIDTH OF NOISE SIGNAL, IN HERTZ
C CONSTS SENVAR      AVERAGE AMPLIFIER OUTPUT.
```

Figure 2.1-54

```
C IPRINT CONST        OUTPUT DATA DEVICE DESIGNATOR = 6
C ISM    UNTER        INDEX DESCRIBING SOIL MOISTURE CONDITIONS.
C IVCGV  UNTER        INDEX VEG. COVER
C LDUMP  CONST        TRUE= INTERMEDIATE CALCULATIONS PRINTED, FALSE= NO PRINT.
C PRATE  ATMENV       RAIN FALL RATE (MM/HR) .
C PTOT24 ATMENV       TOTAL PRECIPITATION DURING THE LAST 24 HRS.
C WSPEED ATMENV       WIND SPEED  (KM/HR)
C
C                       INTERNALLY STORED DESIGNER INPUT VALUES.
C BETA   SEISBK  P    AVGTHC MODIFIER FOR FIXED GAIN SENSORS.
C FBATL  SEISBK  P    VALUE FOR BATTLE EFFECT,1=1.5(LOW INT.),2=.15(M),3=.1(H)
C FCULT  SEISBK  P    VALUE FOR POP. EFFECT,1=.25(REM.),2=.15(RUR.),3=.1(URB.)
C FRAIN  SEISBK  P    VALUE FOR RAIN EFFECT,1=.5(LOW),2=.3(MOD.),3=.2(HEAVY)
C FWIND  SEISBK  P    VALUE FOR WIND EFFECT,1=.5,2=.4,3=.3,4=.2,5=.2
C SENSEX SEISBK  P    EFFECT OF SENSOR BEING BURIED.
C SET    SEISBK  P    TABLE OF GAIN VALUES.
C SOILM  SEISBK  P    SOIL EFFECTS ON RAIN NOISE
C VEGCVR SEISBK  P    VEGETATION COVER EFFECTS ON RAIN NOISE
C VEGCVW SEISBK  P    VEGETATION COVER EFFECTS ON WIND NOISE.
C
C                             COMPUTED VALUES
C DBRAIN SEISBK OP    RAIN NOISE       (DB)
C DBWIND SEISBK OP    WIND NOISE       (DB)
C DELTA  SEISBK OP    MODIFIER OF FALSE ALARM RATE,FIXED GAIN SYSTEM
C DUM    SEISBK OP    RANDOM NUMBER  0-1
C IFIXGS SEISBK OP    ABS. VALUE OF IFIXGN (INDEX FOR GAIN SELECTOR.)
C ISOILM SEISBK OP    IND.-SOIL MODIF.,1=DRY(0.),2=WET(3),3=V.WET(6),4=SAT.(6)
C IVCOVR SEISBK OP    INDEX VEG. COVER,1=HEAVY,2=MED.FOR.,3=L.FOL.,4=H2O,5=OPE
C KBATL  SEISBK OP    INDEX BATTLE NOISE,1=1.5(LOW INT.),2=.7(MED.),3=.4(HIGH)
C KCULT  SEISBK OP    INDEX CULT. BACK.
C KRAIN  SEISBK OP    INDEX FOR RAIN CONDITIONS.
C KWIND  SEISBK OP    INDEX FOR WIND GUSTINESS.
C L      SEISBK OP    DUMMY INDEX
C ONOISE SEISBK OP    OUTPUT NOISE FOR FIXED GAIN SYSTEM
C VBATL  SEISBK OP    DBBATL CONVERTED TO VOLTAGE.
C VCULT  SEISBK OP    DBCULT CONVERTED TO VOLTAGE.
C VRAIN  SEISBK OP    DBRAIN CONVERTED TO VOLTAGE.
C VWIND  SEISBK OP    DBWIND CONVERTED TO VOLTAGE.
C
C                             OUTPUT VALUES
C AVGTHC SEISBK OP    AVG. TIME BETWEEN THRESHOLD CROSSINGS(IN SECONDS).
C GEQUIL SEISBK OP    AMPLIFIER GAIN
C THRESH SEISBK OP    THRESHOLD (VOLTS)
C VNOISE SEISBK OP    RMS SUM OF BACKGROUND NOISE VOLTAGES
C*
C***********************************************************************
```

Figure 2.1-54 (Cont.)

```
C*************************************** ACOUBK ****************************
C*
C*                           SUBROUTINE ACOUBK
C*
C*    PURPOSE:
C*        THIS ROUTINE IS USED DURING PRERUN, TO ESTABLISH OPER-
C*        ATIONAL PARAMETERS FOR ACOUSTIC SENSORS (THESE PARAM-
C*        ETERS TO BE USED DURING MSM STAGE BY SUBROUTINE ACOUTG).
C*        OUTPUT PARAMETERS ARE--
C*                AMPLIFIER GAIN
C*                THRESHOLD VOLTAGE
C*                RMS BACKGROUND NOISE VOLTAGE
C*                       AVERAGE TIME BETWEEN THRESHOLD CROSSINGS
C*
C*    USAGE:
C*          CALL ACOUBK(IUT,DBCULT,DBBATL,THRESH,GEQUIL,VNOISE,AVGTHC)
C*
C*    DESCRIPTION OF PARAMETERS
C*        SEE GLOSSARY BELOW
C*
C*    REMARKS:
C*             NONE
C*
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED:
C*        ALOG10    SUPPLIED BY FORTRAN
C*        SQRT      SUPPLIED BY FORTRAN
C*
C*    METHOD:
C*       BACKGROUND NOISE LEVELS ARE ASCRIBED TO THE COMMON SOURCES OF
C*    ACOUSTIC BACKGROUND. THESE LEVELS WHICH ARE FUNCTIONS OF THE
C*    ENVIRONMENT, TERRAIN, TIME OF DAY, CULTURAL LEVELS, AND BATTLE
C*    CONDITIONS ARE COMBINED TO DEVELOP AN RMS NOISE VOLTAGE AGAINST
C*    WHICH SIGNALS MUST COMPETE AND FROM WHICH AMPLIFIER GAIN SETTINGS
C*    AND DEVICE FALSE ALARM RATE ARE DETERMINED. THE FALSE ALARM RATE
C*    IS OF EMPIRICAL BASIS AND FURTHER EFFORT IN SIGNAL ANALYSIS WILL
C*    BE REQUIRED TO PROVIDE AN ACCURATE ANALYTICAL TREATMENT.
C*
C*    GLOSSARY:
C                               INPUT VALUES
C DBBATL ACOUBK CP   BATTLE NOISE (DB)
C DBCULT ACOUBK CP   CULTURAL NOISE (DB)
C
C                           LABELLED COMMON INPUTED VALUES
C BIASAC SENVAR      THRESHOLD SETTING  (VOLTS)
C BWACOU SENVAR      BAND WIDTH
C CONSTA SENVAR      AVERAGE AMPLIFIER OUTPUT.
C IPRINT CONST       OUTPUT DATA DEVICE DESIGNATOR = 6
C ITOD   BASICT      TIME OF DAY
C LDUMP  CONST       TRUE= INTERMEDIATE CALCULATIONS PRINTED, FALSE= NO PRINT.
C PRATE  ATMENV      RAIN FALL RATE (MM/HR)
```

Figure 2.1-55

```
C WSPEED ATMENV       WIND SPEED  (KM/HR)
C
C                       INTERNALLY STORED DESIGNER INPUT VALUES.
C FANTBL ACOUBK  P    FAUNA NOISE TABLE, SELECTED BY INDEX KFAUN
C FBATL  ACOUBK  P    BATTLE NOISE   SELECTED BY INDEX  KBATL
C FCULT  ACOUBK  P    CULTURAL NOISE SELECTED BY INDEX  KCULT
C FFAUN  ACOUBK  P    FAUNA NOISE SELECTED BY INDEX KFAUN
C*
C                             COMPUTED VALUES
C DBFAUN ACOUBK DP    FAUNA NOISE. (DB)
C DBRAIN ACOUBK DP    RAIN NOISE (DB)
C DBWIND ACOUBK DP    WIND NOISE (DB)
C FRAIN  ACOUBK DP    RAIN NOISE
C FWIND  ACOUBK DP    WIND NOISE
C KBATL  ACOUBK DP    INDEX FOR BATTLE   NOISE
C KCULT  ACOUBK DP    INDEX CULTURAL NOISE
C KFAUN  ACOUBK DP    INDEX FAUNA NOISE
C VBATL  ACOUBK DP    BATTLE NOISE (VOLTS)
C VCULT  ACOUBK DP    CULTURAL NOISE (VOLTS)
C VFAUN  ACOUBK DP    FAUNA NOISE (VOLTS)
C VRAIN  ACOUBK DP    RAIN NOISE (VOLTS)
C VWIND  ACOUBK DP    WIND NOISE (VOLTS)
C*
C                             OUTPUT VALUES
C AVGTHC ACOUBK OP    AVG. TIME BETWEEN THRESHOLD CROSSINGS (IN SECONDS).
C GEQUIL ACOUBK OP    AMPLIFIER GAIN
C THRESH ACOUBK OP    THRESHOLD (AMPLIFIER)
C VNOISE ACOUBK OP    TOTAL BACKGROUND NOISE.
C*
C******************************************************************************
```

Figure 2.1-55 (Cont.)

```
C*************************************** ARFBK  *****************************
C*
C*                          SUBROUTINE ARFBK
C*
C*    PURPOSE
C*       THIS ROUTINE IS PROVIDED TO DETERMINE THE AREA DENSITY OF BUTTON
C*    BOMBLETS FOR USE IN ARFTG, AND TO DEVELOP ESTIMATES FOR FALSE ALARM
C*    RATE AND AVERAGE FALSE ALARM INTERVAL.
C*    USAGE
C*          CALL ARFBK(NBMBLT,IEMPLC,IGEOM,IMAG,DIMMAX,WIDTH,AVFATM,AREAON)
C*
C*    DESCRIPTION OF PARAMETERS
C*        SEE GLOSSARY BELOW
C*
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED:
C*        ALOG10    SUPPLIED BY FORTRAN
C*
C*    METHOD
C*
C*       THE SENSOR BEING SIMULATED CONSISTS OF A NUMBER OF EMITTERS
C*    WHICH ARE EXCITED BY APPLICATION OF FORCE OR BY MOTION OF MAGNETIC
C*    MATERIALS DEPENDING ON THE TYPE OF DEVICES EMPLOYED. THE BASIC
C*    OBJECTIVES OF THIS ROUTINE ARE TO TAKE PLANNER INPUTS TO DEVELOP
C*    THE AREA DENSITY OF DEVICES AND TOGETHER WITH ATMENV DATA TO
C*    PROVIDE AN AVERAGE FALSE ALARM INTERVAL FOR THE FALSE ALARM ROUTINE.
C*    THREE TYPES OF ARRAYS ARE CONSIDERED, NAMELY, 1 (OPEN CIRCLE),
C*    2 (OPEN LINE - WHICH IS BASICALLY A RECTANGULAR ARRAY DISPERSED IN
C*    AN OPEN AREA), 3 (A TRAIL/ROAD ARRAY)  DEVICES MAY BE OF THE
C*    NOISELESS OR MAGNETIC TYPES. THE AREA DENSITY IS DEVELOPED SIMPLY
C*    BY DIVIDING THE NUMBER OF EMITTERS DEPLOYED BY THE AREA OVER WHICH
C*    THEY ARE DEPLOYED.
C*
C*    GLOSSARY
C*
C*                             INPUT VALUES
C DIMMAX ARFBK  CP   MAX. DIM. OF SEEDED AREA(RECT. LGTH. OR CIRC. DIAM.)(MET.
C IEMPLC ARFBK  CP   METHOD OF EMPLACEMENT, 1= HAND , 2=ARTILLERY, 3=AIR
C IGEOM  ARFBK  CP   =1 (OPEN CIRC.), =2(OPEN LINE), =3(ROAD OR TRAIL)
C IMAG   ARFBK  CP   INDEX ON BOMBLET TYPE (0= MAGNETIC, 1= NOISELESS).
C NBMBLT ARFBK  CP   NO. OF BOMBLETS USED.
C WIDTH  ARFBK  CP   WIDTH OF SEEDED AREA (METERS).
C*
C*                        LABELLED COMMON INPUTED VALUES
C IPRINT CONST       OUTPUT DATA DEVICE DESIGNATOR = 6
C LDUMP  CONST       TRUE= INTERMEDIATE CALCULATIONS PRINTED, FALSE= NO PRINT.
C PRATE  ATMENV      RAIN FALL RATE (MM/HR)
```

Figure 2.1-55

```
C*
C*                        INTERNALLY STORED DESIGNER INPUT VALUES.
C*
C*                                  COMPUTED VALUES
C FARATE ARFBK  DP   FALSE ALARM RATE.
C RAINF  ARFBK  DP   FACTOR GIVING EFFECT OF RAIN ON FALSE ALARM RATE.
C SAREA  ARFBK  DP   AREA OF THE NBB ARRAY (SQ. METERS)
C SNUMB  ARFBK  DP   NUMBER OF NBB'S
C*
C*                                  OUTPUT VALUES
C AREADN ARFBK  OP   AREA DENSITY OF THE NBB'S  (SQ. METERS).
C AVFATM ARFBK  OP   AVERAGE FALSE ALARM VALUE
C*
C*****************************************************************************************
```

Figure 2.1-56 (Cont.)

```
C************************************ ENVIR  ****************************
C*
C*     SUBROUTINE ENVIR
C*
C*     PURPOSE
C*        THE TEMPERATURE OF THE BACKGROUND IS A COMPLEX FUNCTION OF A
C*     NUMBER OF FACTORS INCLUDING TIME OF DAY, SEASON, CLOUD COVER,
C*     VEGATATION COVER, VEGATATION CHARACTARISTICS, AIR TEMPERATURE AND
C*     OTHERS. IN THIS ROUTINE ESTIMATES OF BACKGROUND TEMPERATURE ARE
C*     DEVELOPED ALONG WITH ESTIMATES FOR THE VARIATION IN THE TEMPERATURE.
C*
C*     USAGE
C*     CALL ENVIR(IUT,TEMPEV,SIGMA)
C*
C*     DESCRIPTION OF PARAMETERS
C*         SEE GLOSSARY BELOW
C*
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED:
C*         ALOG10    SUPPLIED BY FORTRAN
C*
C*     METHOD
C*        THE TEMPERATURE OF THE BACKGROUND IS CONSIDERED TO BE RELATED
C*     DIRECTLY TO THE AMBIENT AIR TEMPERATURE BUT THE AMOUNT BY WHICH THE
C*     BACKGROUND DIFFERS FROM AMBIENT IS TAKEN TO BE A JUNCTION OF A
C*     NUMBER OF VARIABLES. THESE INCLUDE: WIND SPEED, PRECIPITATION,
C*     VEGETATION TYPE MAKING UP THE BACKGROUND, VEGETATION COVER ABOVE
C*     THE BACKGROUND TO BE VIEWED, SOLAR ALTITUDE, AND CLOUD COVER.
C*     ESTIMATES OF THE EFFECTS OF EACH BASED ON A SMALL AMOUNT EMPIRICAL
C*     INFORMATION ARE INCLUDED AS EQUATIONS OR TABULAR DATA. CORRELATION
C*     BETWEEN VARIABLES IS NOT CONSIDERED. THIS ROUTINE IS TO BE
C*     CONSIDERED TO BE INTERIM WITH CONSIDERABLE EMPHASIS REQUIRED TO
C*     DEPICT MORE COMPLETELY AND ACCURATLY, THE ACTUAL DESCRIPTION OF
C*     BACKGROUND TEMPERATURE.
C*
C*     GLOSSARY
C*
C*                                  INPUT VALUES
C IUT     ENVIR  CP   INDEX UNIT TERRAIN
C*
C*                            INTERNALLY STORED DESIGNER INPUT VALUES.
C CLOUDF ENVIR    P   CLOUD FACTOR DATA
C CLOUDV ENVIR    P   VARIANCE FACTOR CLOUD DATA
C EIGHT  ENVIR    P   CONSTANT (7.99999)
C RAINF  ENVIR    P   RAIN FACTOR DATA
C RAINV  ENVIR    P   VARIANCE FACTOR RAIN DATA
C TYPEF  ENVIR    P   BACKGROUND TYPE FACTOR DATA
C TYPEV  ENVIR    P   VARIANCE FACTOR BACK. TYPE DATA
C VEGF   ENVIR    P   VEG. FACTOR DATA
C VEGV   ENVIR    P   VARIANCE FACTOR VEG. DATA
C WINDF  ENVIR    P   WIND FACTOR DATA
```

Figure 2.1-57

```
C WINDV  ENVIR   P   VARIANCE FACTOR WIND DATA
C*
C*                       LABELLED COMMON INPUTED VALUES
C ATEMP  ATMENV      AMBIENT AIR TEMPERATURE.
C IBACK  UNTER       INDEX IDENT. MOST LIKELY BACK. REFLECTANCE FUNCTION.
C IPRINT CONST       OUTPUT DATA DEVICE DESIGNATOR = 6
C ITOD   BASICT      TIME OF DAY
C IVCOV  UNTER       INDEX VEG. COVER, 1=HEAVY,2=MED,3=LIGHT,4=OPEN,5=WATER
C LDUMP  CONST       TRUE= INTERMEDIATE CALCULATIONS PRINTED, FALSE= NO PRINT.
C PRATE  ATMENV      RAIN FALL RATE (MM/HR)
C SOLALT ATEMNV      SOLAR ALTITUDE (DEGREES)
C TCLOUD ATMENV      TRANSMISSION OF CLOUD COVER.
C WSPEED ATMENV      WIND SPEED  (KM/HR)
C*
C*                           COMPUTED VALUES
C CFACT  ENVIR  OP   CLOUD FACTOR
C CVAR   ENVIR  OP   VARIANCE FACTOR DUE TO CLOUDS
C HRLCCL ENVIR  OP   LOCAL TIME
C ICLD   ENVIR  OP   INDEX ON CLOUD COVER
C IRAIN  ENVIR  OP   INDEX ON RAINFALL RATE
C ITYPE  ENVIR  OP   INDEX ON BACKGROUND TYPE
C IVEGCV ENVIR  OP   INDEX ON VEGETATION COVER
C IWIND  ENVIR  OP   INDEX ON WIND
C RFACT  ENVIR  OP   RAIN FACTOR
C RVAR   ENVIR  OP   VARIANCE FACTOR DUE TO RAIN.
C TFACT  ENVIR  OP   BACKGROUND TYPE FACTOR
C TVAR   ENVIR  OP   VARIANCE FACTOR DUE TO BACK. TYPE
C VFACT  ENVIR  OP   VEG. FACTOR
C VVAR   ENVIR  OP   VARIANCE FACTOR DUE TO VEG.
C WFACT  ENVIR  OP   WIND FACTOR
C WVAR   ENVIR  OP   VARIANCE FACTOR DUE TO WIND.
C*
C*                           OUTPUT VALUES
C SIGMA  ENVIR  OP   STANDARD DEVIATION OF BACKGROUND TEMP. FLUCTUATION.
C TEMPEV ENVIR  OP   BACKGROUND TEMPERATURE (C DEG.)
C*
C***********************************************************************
```

Figure 2.1-57 (Cont.)

```
C************************************ PIRBK  ****************************
C*
C*                              SUBROUTINE PIRBK
C*
C*     PURPOSE
C*        THIS ROUTINE PROVIDES THE POWER LEVEL INCIDENT  ON THE DETECTOR
C*     DUE TO BACKGROUND BALANCE, AND ESTIMATES OF AVERAGE FALSE ALARM
C*     INTERVAL.
C*         THIS ROUTINE IS USED DURING PRERUN, TO ESTABLISH OPER-
C*         ATIONAL PARAMETERS FOR PIRID      SENSORS(THESE PARAM-
C*         ETERS TO BE USED DURING MSM STAGE BY SUBROUTINE PIRBK).
C*     USAGE
C*            CALL PIRBK(IUT,TEMPEV,FIELD,EXPAN,WATTBK,AVGTHC)
C*
C*     DESCRIPTION OF PARAMETERS
C*         SEE GLOSSARY BELOW
C*
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED:
C*         ERFC(X)    COMPLEMENTARY ERROR FUNCTION OF X
C*         ENVIR      COMPUTES TEMPEV AND SIGMA
C*
C*     METHOD
C*        BACKGROUND TEMPERATURE AND ITS FLUCTUATIONS AS DERIVED IN THE
C*     ENVIRONMENT ROUTINE ARE EMPLOYED TO ESTABLISH THE BACKGROUND POWER
C*     INCIDENT ON THE DETECTOR AND THE VARIANCE OF NOISE DUE TO BACK-
C*     GROUND.  THE VARIANCE IS USED IN COMPUTATION OF FALSE ALARM RATE.
C*     THE SIGNAL DUE TO A TARGET PASSING THROUGH THE FIELD OF VIEW IS
C*     DETERMINED AND IF IT EXCEEDS THE THRESHOLD A DETECTION IS DECLARED.
C*     ALL COMPUTATIONS ARE REFERRED TO THE INPUT OF THE SENSOR.
C*
C*     GLOSSARY
C*                                INPUT VALUES
C IUT     PIRBK  CP    INDEX UNIT TERRAIN
C SIGMA   PIRBK  CP    STANDARD DEVIATION OF BACKGROUND TEMP. FLUCTUATION.
C TEMPEV  PIRBK  CP    TEMP. OF BACKGROUND DETERMINED IN THE ENVIR. SUB. (C DEG)
C
C*                          LABELLED COMMON INPUTED VALUES
C BWPIR   SENVAR       BAND WIDTH (HZ./SEC.)
C DEVXMN  SENVAR       OPTICAL SYSTEM TRANSMISSION FACTOR.
C DIAM    SENVAR       DIAMETER OF SENSOR (MM).
C IPRINT  CONST        OUTPUT DATA DEVICE DESIGNATOR = 6
C LDUMP   CONST        TRUE= INTERMEDIATE CALCULATIONS PRINTED, FALSE= NO PRINT.
C PHIAZ   SENVAR       AZIMUTH ANGLE IN RADIANS.
C PHIEL   SENVAR       ELEVATION ANGLE IN RADIANS.
C STEFK   CONST       STEFAN  BOLTZMANN     CONSTANT /PI   (1.8?5455E-8)
```

Figure 2.1-58

```
C
C*                                   COMPUTED VALUES
C AREA    PIRBK   OP    AREA OF SENSOR IN SQUARE METERS.
C PIO4    PIRBK    P    PI/4 (0.785398)
C PROBTH  PIRBK   OP    PROBABILITY OF CROSSING THRESHOLD.
C RADBAK  PIRBK   OP    BACKGROUND RADIANCE.
C TEMPKL  PIRBK   OP    BACKGROUND TEMP. (DEG. KELVIN)
C*
C*                                   OUTPUT VALUES
C*
C AVGTHC  PIRBK   OP    AVERAGE THRESHOLD CROSSING
C EXPAN   PIRBK   OP    INTERMEDIATE CALC. (AREA * FIELD * DEVXMN)
C FIELD   PIRBK   OP    FIELD OF VIEW.
C WATTBK  PIRBK   OP    BACKGROUND POWER INCIDENT ON SENSOR.
C*                                                                            *
C*****************************************************************************
```

Figure 2.1-58 (Cont.)

```
C********************************** BWIRBK ***************************
C*
C          SUBROUTINE BWIRBK
C*
C*    PURPOSE:
C*         THIS ROUTINE IS USED DURING PRERUN, TO DEVELOP FALSE ALARM
C*         DATA FOR USE IN SCHEDULING FALSE ALARM EVENTS THRU FAINTV.
C*
C*    USAGE:
C*    CALL BWIRBK  (IUT,ALENGT,AVGTHC)
C*
C*    DESCRIPTION OF PARAMETERS
C*    SEE GLOSSARY BELOW
C*
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED:
C*
C*    METHOD:
C*         ESTIMATES OF THE EFFECTS OF VEGETATION,WIND,AND FAUNA ON
C*         THE LENGTH OF TIME THE SENSOR MAY BE MAINTAINED WITHOUT
C*         ACTIVATION ARE SELECTED BASED ON THE ENVIROMENTAL
C*         CONDITIONS,(UNTER AND ATMENV).THESE ESTIMATES ARE NOT
C*         SUPPORTED BY FIELD DATA AND ARE THEREFORE SUBJECT TO
C*         CHANGE.FROM THESE ESTIMATES THE AVERAGE TIME TO ACTVATE
C*         THE SENSOR IS DEVELOPED AND SUPPLIED TO FAINTV FOR
C*         DETERMINATION OF BREAK EVENT TIME.IF TIME TO BREAK IS
C*         GREATER THAN DURATION OF ENVIROMENT CONDITIONS USED
C*         IN ESTIMATE DEVELOPMENT,BREAK EVENT IS SCHEDULED FOR NEW
C*         SET OF CONDITIONS.
C*
C*    GLOSSARY:
C                                    INPUT VALUES
C ALENGT BWIRBK CM    LENGTH OF LINE DEPLOYED.   (YDS)
C IUT    BWIRBK CM    INDEX ON UNIT TERRAIN.
C*
C                              INTERNALLY STORED DESIGNER INPUT VALUES
C DLENGT BWIRBK  M    LENGTH OF LINE AVAILABLE (YDS.) SET TO 2500.
C FNCOMP BWIRBK  M    FAUNA COMPONENT FACTOR
C WCOMP  BWIRBK  M    WIND COMPONENT FACTOR.
C*
C                                LABELLED COMMON INPUTED VALUES
C IPRINT CONST        OUTPUT DATA DEVICE DESIGNATOR = 6
C  TOD   BASICT       TIME OF DAY
C LDUMP  CONST        TRUE= INTERMEDIATE CALCULATIONS PRINTED, FALSE= NO PRINT.
C WSPEED ATMENV       WIND SPEED  (KM/HR)
C
C                                    COMPUTED VALUES
C DUM    BWIRBK DM    DUMMY ARGUMENT.
C FAFACT BWIRBK DM    DUMMY ARGUMENT.
C KFAUN  BWIRBK DM    INDEX IN ANIMAL ACTIVITY (1= 6AM-6PM, 2= 6PM-6AM).
C WFACT  BWIRBK DM    WIND COMPONENT FACTOR FOR PARTICULAR VEGETATION TYPE.
C*
C*                                   OUTPUT VALUES
C AVGTHC BWIRBK DM    AVERAGE THRESHOLD CROSSING
C*
C*********************************************************************
```

Figure 2.1-59

```
C************************************ FAINTV ************************************
C*                                                                              *
C*                              FUNCTION FAINTV                                 *
C*                                                                              *
C*    PURPOSE                                                                   *
C*        SPECIAL PURPOSE RANDOM NUMBER GENERATOR.  PROVIDES                   *
C*        RANDOM FALSE ALARM TIME INTERVALS FOR 5 TYPES OF                      *
C*        SENSORS, WITH STATISTICS APPROPRIATE TO (A) SENSOR                    *
C*        LOGIC AND (B) AN AVERAGE TIME PARAMETER SUPPLIED                      *
C*        AS AN INPUT VARIABLE.                                                 *
C*                                                                              *
C*    USAGE                                                                     *
C*        TIME = FAINTV (AVGT, ITYPSN)                                          *
C*                                                                              *
C*    DESCRIPTION OF PARAMETERS                                                 *
C*        ITYPSN   INTEGER CODE FOR SENSOR GENERIC TYPE.                        *
C*                  VALID VALUES ARE 1 (SEISMIC), 2 (ACOUSTIC),                 *
C*                 4 (ARFBUOY), 5 (PASSIVIR), AND 9 (BREAKWIR).                 *
C*                                                                              *
C*        AVGT     FOR ARFBUOY AND BREAKWIR, = AVERAGE FALSE                    *
C*                 ALARM TIME.  FOR OTHER SENSOR TYPES, = AVERAGE               *
C*                 THRESHOLD CROSSING TIME.                                     *
C*                                                                              *
C*    REMARKS:                                                                  *
C*        1. THIS FAINTV ROUTINE REPLACES ALL PREVIOUS VERSIONS,                *
C*           THAT WERE LESS GENERAL AND/OR REQUIRED EXCESSIVE                   *
C*           COMPUTATIONAL TIME.  NOTE THAT CALLING SEQUENCE                    *
C*           DIFFERS FROM PREVIOUS VERSIONS.                                    *
C*                                                                              *
C*        2. DESIGNER VALUES ARE USED IN PROGRAM FOR THE N AND T                *
C*           IN 'FALSE ALARM WHEN N THRESHOLD CROSSINGS IN T                    *
C*           SECONDS', AND FOR THE DEAD TIME (SENSOR INACTIVATED)               *
C*           FOLLOWING A FALSE ALARM.  SPECIFICALLY...                          *
C*                                                                              *
C*                           N        T          DEAD TIME                      *
C*              SEISMIC      4     6 SECONDS     15 SECONDS                     *
C*              ACOUSTIC     4     6 SECONDS     15 SECONDS                     *
C*              PASSIVIR     2   1.5 SECONDS     1.5 SECONDS                    *
C*              ARFBUOY      1    IRRELEVANT      0 SECONDS                     *
C*              BREAKWIR     1    IRRELEVANT      0 SECONDS                     *
C*                                                                              *
C*    METHOD:                                                                   *
C*            IF SUBROUTINE IS CALLED FOR SENSOR TYPES OTHER THAN 1 (SEISMIC),  *
C*       2 (ACOUSTIC), 4 (ARFBUOY), 5 (PASSIVE IR), OR 9 (BREAKWIRE); A DIAGNOS-*
C*       TIC IS PRINTED.                                                        *
C*            BASED ON THE SENSOR TYPE, THE FALSE ALARM INTERVAL IS CALCULATED  *
C*       USING THE APPROPRIATE FORMULA (SEE PAGE 4-64, VOL I, PART II).  THE    *
C*       ACTUAL FALSE ALARM TIME IS DETERMINED BY MULTIPLYING THE AVERAGE RATE  *
C*       BY THE LOGARITHM OF UNIFORM RANDOM NUMBER.                             *
C*                                                                              *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                              *
C*       URN        (UNIFORM RANDOM NUMBER GENERATOR)                           *
C*                                                                              *
C********************************************************************************
```

Figure 2.1-60

2.1.5.9 PRERUN Step 8

Step 8 in PRERUN comprises the main program (PREMN8) with 10 subroutines listed in Table 2.1-V, plus 2 utility subroutines.

External data sets required as input are "DATAIN", "JFMSTR", "JFPXY", and "JFTAR", all generated in previous steps.

This step sets up RED FORCES, and BLUE FORCES not associated with moving arrays, as targets. It plays all sensors against all targets for "geometrical detection" through ELPDT subroutine. These detections are stored on two separate disc files: the non-LOS events (JFNEV) and the LOS events (JFZEV).

Figures 2.1-61 through 2.1-70 describe the subroutines of this step. Subroutine VALID was described previously in Step 5.

2.1.5.10 PRERUN Step 9

Step 9 handles line-of-sight calculations. Options exist on subroutine structure: the user may play Line-of-Sight with full achievable accuracy based on digital terrain tape, or he may use a dummy LOS routine, by simply inserting the subprograms comprising the deck setup desired as Step 9.

(a) For dummy LOS, the Step 9 setup comprises the main program (PREMN9) with 1 subroutine (LSGT). External data sets required as input are "JFMSTR", and "JFZEV", both generated in the previous steps. The dummy LOS routine simply stores the LOS geometrical detections on disc (DSNAME=EVNTLS). Figures 2.1-71 and 2.1-72 describe the subroutines of the dummy LOS Step 9.

(b) For "accurate" LOS, the Step 9 setup comprises the main program (MAINLS) with 7 subroutines listed in Table 2.1-V, plus 1 utility subroutine. External data sets required as input are "MASSDAT", "JPOUT", and "JFZEV" generated in previous steps. This program plays actual LOS and stores them on disc (EVNTLS). Further information is given in Section 4.

Figures 2.1-73 through 2.1-77 describe the subroutines of the LOS program.

```
C***********************************************************************
C****************************** PREMNB *********************************
C*                                                                     *
C*                   PRERUN EXFCUTIVE -     STEP  8                    *
C*                                                                     *
C*     PURPOSE                                                         *
C*         PEADS IN TARGETS FROM MOVE ARRAYS AND FROM BATTLE-CULTURE   *
C*         CALLS TARGBR TO GENERATE TARGETS FROM BLUE-RED FORCES AND   *
C*         CALLS ELPEX THE EXECUTIVE ROUTINE THAT CONTROLS THE PLAY    *
C*         OF THE SENSOR TARGET DETECTIONS. WRITES OUT THE NON LINE    *
C*         OF SIGHT AND THE LOS ON SEPARATE TAPES.                     *
C*                                                                     *
C*     USAGE                                                           *
C*         MAIN PROGRAM                                                *
C*                                                                     *
C*     DESCRIPTION OF PARAMETERS                                       *
C*         ALL INPUT AND OUTPUT VIA LABELED COMMON BLOCKS/TIMES,OUTP,  *
C*         DATAIN,TRGB,NV,PXYTP,POSERR,INOUT,OPTION/.
C*                                                                     *
C*     SUBROUTINES REQUIRED                                            *
C*         TARGEX                                                      *
C*         ELPEX                                                       *
C*                                                                     *
C*     METHOD                                                          *
C*             THE COMMON GAME INFORMATION, THE PLANNER INPUT, THE     *
C*         MOVING ARRAY TARGETS AND THE FALSE TARGETS ARE READ IN      *
C*         TARGEX IS CALLED TO GENERATE TARGETS FROM THE BLUE-RED      *
C*         FORCES. ELPEX IS CALLED TO DETERMINE SENSOR-TARGET DETECTIONS*
C*                                                                     *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-61

```
C****************************** TARGEX ********************************
C*                                                                     *
C*                       SUBROUTINE TARGEX                             *
C*                                                                     *
C*    PURPOSE                                                          *
C*        FORMS TARGETS OF THE BLUE AND RED FORCES BY CALLING TARGBR   *
C*    SUBROUTINES REQUIRED                                             *
C*        FINDX                                                        *
C*        TARGBR                                                       *
C*                                                                     *
C*    METHOD                                                           *
C*           THE PLANNER INPUT FOR THE BLUE FORCES IS LOCATED BY FINDX *
C*        AND TARGBR IS CALLED. IF THE BLUE FORCE IS ASSOCIATED WITH   *
C*        A MOVING ARRAY TARGBR WILL GIVE AN IMMEDIATE RETURN SINCE    *
C*        THIS CASE HAS ALREADY BEEN PROCESSED.                        *
C*           THE RED FORCES ARE LOCATED BY FINDX AND TARGBR  IS CALLED. *
C*                                                                     *
C*                                                                     *
C*                                                                     *
C*                                                                     *
C*                                                                     *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-62

```
C**************************************ELPEX*****************************************
C*                                                                                  *
C*                                                                                  *
C*                                                                                  *
C*                       SUBROUTINE ELPEX(MPRINT)                                   *
C*     PURPOSE                                                                      *
C*         THIS IS THE BASIC ELPOT ROUTINE WHICH SCANS ALL THE SENSORS              *
C*         AND CALLS ELPOT TO PLAY EACH SENSOR AGAINST EVERY TARGET                 *
C*                                                                                  *
C*     USAGE                                                                        *
C*         MAIN PROGRAM                                                             *
C*                                                                                  *
C*     DESCRIPTION OF PARAMETERS                  PLANNER INPUT                     *
C*         IS-SENSOR ID                                   3      1                  *
C*         IJK- COVER SCAN ID                             3     11                  *
C*         KPARM- SENSOR DESCRIPTOR                       3      4                  *
C*         WAVE                                           4     18                  *
C*         CLEAR                                         14     11                  *
C*         SXYT- ARRAY CONTAINING SENSOR UP-DOWN TIMES AND LOCATION                 *
C*         XXYT- DICTIONARY FOR SXYT                                                *
C*                                                                                  *
C*     REMARKS                                                                      *
C*         THE PRESENT VERSION OF THE GEOMETRY ROUTINES CIRC AND SECT               *
C*         REQUIRE THAT ALL MOVING SENSORS USE RECTANGULAR COVERAGE                 *
C*         AND BE PROCESSED BY SUBROUTINE GREC.                                     *
C*         EACH LEG OF THE PATH OF A MOVING SENSOR IS PROCESSED BY A                *
C*         SEPARATE CALL TO ELPOT.                                                  *
C*         THE DETECTIONS ARE WRITTEN ON DISC FOR LATER USE                         *
C*         LINE OF SIGHT ARE WRITTEN SEPARATELY FROM NON LOS                        *
C*                                                                                  *
C*     SUBROUTINES REQUIRED                                                         *
C*         FINDX -TO FIND BASIC DATA SETS                                           *
C*         SENSQ -TO DETERMINE COVER SCAN PARMETERS                                 *
C*         ELPOT -TO PLAY AGAINST ALL TARGETS                                       *
C*         FINDY -TO DETERMINE SENSOR UP-DOWN TIMES AND POSITIONS                   *
C*                                                                                  *
C*     METHOD                                                                       *
C*            FINDX IS CALLED TO LOCATE DATA IN THE PLANNER INPUT TABLES*
C*         THE COVERAGE PARAMETERS ARE FOUND BY CALL TO SENSQ. THE                  *
C*         POSITIONS AND UP TIMES FOR THE SENSORS ARE FOUND BY USING                *
C*         SUBROUTINE FINDY TO LOCATE THE DATA IN THE SXYT ARRAY. SINCE *
C*         THE FORMAT FOR THE MOVING ARRAYS IS DIFFERENT FROM THE FORMAT*
C*         FOR THE STATIONARY ARRAYS A BRANCH IS MADE TO THE APPROPRIATE*
C*         SEQUENCE OF INSTRUCTIONS. EACH BRANCH SETS THE COVERAGE                  *
C*         PARAMETERS AND CALLS ELPOT TO DETERMINE SENSOR TARGET                    *
C*         DETECTIONS.                                                              *
C*            FOR EACH SENSOR THE RESULTS OF ELPOT WILL BE  PRINTED IF              *
C*         THE PRINT OPTION IS SET TO A NON ZERO VALUE.                             *
C*            DETECTION REQUIRING LOS ARE SEPARATED FROM THOSE NOT                  *
C*         REQUIRING LOS BY ELPOT. END OF FILE INDICATORS ARE WRITTEN               *
```

Figure 2.1-63

```
C*        ON THE DATA SETS NON-LINE OF SIGHT MTAPE(15) AND LINE OF      *
C*        SIGHT MTAPE(1"'                                               *
C*                                                                      *
C***********************************************************************
```

Figure 2.1-63 (Cont.)

```
C****************************** TARGBR *********************************
C*                                                                     *
C*                          SUBROUTINE TARGBR                          *
C*                                                                     *
C*     PURPOSE                                                         *
C*         FORMS TARGETS OF THE BLUE AND RED FORCES                    *
C*         FOR A MOVING FORCE A  SEPARATE TARGET IS GENERATED FOR EACH *
C*         LEG OF THE PATH                                             *
C*     REMARKS                                                         *
C*       . BLUE FORCES ASSOCIATED WITH MOVE ARRAYS ARE SKIPPED,HAVING  *
C*         BEEN TREATED EARLIER IN PRE RUN                             *
C*     SUBROUTINES REQUIRED                                            *
C*         FINDX                                                       *
C*                                                                     *
C*     METHOD                                                          *
C*             THE TARGET PARAMETERS FOR THE RED FORCES AND FOR THE BLUE *
C*         FORCES NOT ASSOCIATED WITH MOVING ARRAYS ARE EXTRACTED FROM *
C*         THE PLANNER INPUT.                                          *
C*             A MOVING FORCE IS DEFINED AS A SEPARATE TARGET FOR EACH *
C*         LEG OF ITS MOTION AND THE LEG NUMBER IS CODED WITH THE      *
C*         TARGET ID (I,E, TARG(1,NTAR) = TARGET ID + 1000*LEG )       *
C*         TARG(1,NTAR) = ID+1000*LEG                                  *
C*         TARG(2,NTAR) = XA  X POSITION AT TIME TA                    *
C*         TARG(3,NTAR) = YA  Y POSITION  AT TIME TA                   *
C*         TARG(4,NTAR) = TA  TIME                                     *
C*         TARG(5,NTAR) = XB  X POSITION AT TIME TB                    *
C*         TARG(6,NTAR) = YB  Y POSITION AT TIME TB                    *
C*         TARG(7,NTAR) = TB  TIME                                     *
C*         TARG(8,NTAR) =  SPEED OF TARGET                             *
C*         TARG(9,NTAR) =  FERROUS MATERIAL                            *
C*         TARG(10,NTAR)=  ALTITUDE                                    *
C*         TARG(11,NTAR)= TARGET TYPE                                  *
C*         TARG(12,NTAR)=  LENGTH                                      *
C*         TARG(13,NTAR)=  VISUAL SECURITY DESCRIPTION                 *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-64

```
C****************************** ELPDT *******************************
C*                                                                  *
C*                      SUBROUTINE ELPDT                            *
C*                                                                  *
C*    PURPOSE                                                       *
C*        TO DETERMINE SENSOR TARGET DETECTIONS USING THE APPROPRIATE *
C*        GEOMETRY ROUTINE                                          *
C*        IF LOS IS REQUIRED ADDITIONAL  PARAMETERS ARE RECORDED    *
C*    DESCRIPTION OF PARAMETERS                                     *
C*        TARG ARRAY CONTAINING TARGET INFO                         *
C*        EPDT  EARLIEST POSSIBLE DETECTION TIME                    *
C*        ELPDT LATEST POSSIBLE DETECTION TIME                      *
C*      • NEV   NON-LOS DETECTIONS   SENSOR ID,TARGET ID, EPDT,ELPDT
C*        ZEVS  LOS-DETECTIONS
C*    SUBROUTINES REQUIRED                                          *
C*        VALID  - SENSOR-TARGET COMBINATIONS  TABLE III            *
C*        SECT  CIRCULAR OR SECTOR COVERAGE                         *
C*        GREC  RECTANGULAR COVERAGE                                *
C*                                                                  *
C*    METHOD                                                        *
C*            ELPDT PROCESSES ONE SENSOR AGAINST ALL TARGETS BY MEANS *
C*        OF A DO LOOP. SUBROUTINE VALID IS CALLED TO DETERMINE     *
C*        POSSIBLE  SENSOR TARGET COMBINATIONS AND TO INDICATE WHETHER *
C*        LINE OF SIGHT IS REQUIRED. SEPARATE CALLS TO VALID ARE MADE *
C*        FOR FALSE AND REAL TARGETS TO PROPERLY  IDENTIFY THE TARGET. *
C*        ( FALSE TARGETS HAVE A NEGATIVE ID).                      *
C*            SUBROUTINES SECT OR GREC ARE CALLED DEPENDING ON THE TYPE *
C*        OF COVERAGE.                                              *
C*            IF LINE OF SIGHT IS REQUIRED THE APPROPRIATE DATA FOR THE *
C*        LINE OF SIGHT ROUTINES IS COMPUTED AND PUT INTO ARRAY LENS *
C*        AND LINES OF SIGHT DETECTIONS ARE PUT INTO NEV.           *
C*                                                                  *
C*        NEV(1,M+BX) =  SENSOR ID                                  *
C*        NEV(2,M+BY) =  TARGET ID                                  *
C*        NEV(3,M+BX) =  EARLIEST POSSIBLE DETECTION ON TIME        *
C*        NEV(4,M+BY) =  LATEST  POSSIBLE DETECTION ON TIME         *
C*                                                                  *
C*                                                                  *
C********************************************************************
```

Figure 2.1-65

```
C******************************  SFNSQ  *********************************
C*                                                                      *
C*                       SUBROUTINE SENSQ(TT,IJK,KPARM,KSN)             *
C*                                                                      *
C*    PURPOSE                                                           *
C*        THIS ROUTINE IS USED TO FIND THE COVERAGE PARAMETERS          *
C*        OF THE SENSORS                                                *
C*    USAGE                                                             *
C*        CALL SENSQ(TT,IJK,KPARM,KSN)                                  *
C*                                                                      *
C*    DESCRIPTION FO PARAMETERS                                         *
C*         * INPUT *                                                    *
C*    KPARM-LOCATION OF SENSOR DESCRIPTOR PARAMETER SET                 *
C*    KSN  -LOCATION OF COVER SCAN SET                                  *
C*    IJK =0 NO COVER SCAN SET                                          *
C*        =1 COVER SCAN SET                                             *
C*         * OUTPUT *                                                   *
C*    TT -ARRAY                                   PLANNER INPUT TABLE   *
C*    TT(9)    MINIMUM RADIUS OR WIDTH                      4   12-13   *
C*    TT(10)   MAXIMUM RADIUS OR LENGTH                     4      11   *
C*    TT(11)   AZIMUTH ANGLE -(ORIENTATION ANGLE)           3       9   *
C*    TT(12)   COVERAGE ANGLE                               4      11   *
C*    TT(13)   ALONG  TRACK DISTANCE                        4      11   *
C*    TT(14)   ACROSS TRACK DISTANCE                        4      10   *
C*    TT(15)   TYPE COVERAGE                               14       2   *
C*    TT(20)-TT(21)-TT(22) VARIABLE MAXIMUM RADIUS OR WIDTH  4 13-14-15*
C*                                                                      *
C*    REMARKS                                                           *
C*        DATA FROM COVER SCAN SET IF GIVEN OVERRIDES DATA FROM         *
C*        SENSOR DESCRIPTION SET                                        *
C*                                                                      *
C*    METHOD                                                            *
C*           THE COVERAGE PARAMETERS ARE EXTRACTED FROM THE PLANNER     *
C*        INPUT DATA.  IF A COVER/SCAN SET IS SPECIFIED THE COVERAGE    *
C*        PARAMETERS ARE BOUNDED BY THE NON ZERO VALUES IN THE COVER    *
C*        SCAN SET                                                      *
C*                                                                      *
C************************************************************************
```

Figure 2.1-66

```
C******************************** CIRC *************************************
C*                                                                        *
C*                          SUBROUTINE CIRC                               *
C*    PURPOSE                                                             *
C*        PRERUN GEOMETRY ROUTINE.  DETERMINES INTERSECTION(S),           *
C*        IF ANY, OF A STRAIGHT LINE WITH AN ANNULAR REGION               *
C*        (CONCENTRIC CIRCLES, RADII RMAX, RMIN. RMIN MAY BE              *
C*         ZERO).                                                         *
C*                                                                        *
C*    CALLING SEQUENCE                                                    *
C*        CALL CIRC                                                       *
C*                                                                        *
C*    DESCRIPTION OF PARAMETERS                                           *
C*        ALL INPUT AND OUTPUT VIA COMMON AREA /PGMPAR/.                  *
C*                                                                        *
C*         * INPUTS *                                                     *
C*       RMIN   RADIUS OF INNER CIRCLE                                    *
C*       RMAX   RADIUS OF OUTER CIRCLE                                    *
C*       DX     X COMPONENT OF TARGET PATH                                *
C*       DY     Y COMPONENT OF TARGET PATH                                *
C*       DX1    INITIAL X DISTANCE OF SENSOR FROM TARGET                  *
C*       DY1    INITIAL Y DISTANCE OF SENSOR FROM TARGET                  *
C*       DT     TOTAL TARGET TIME INTERVAL                                *
C*       TITYM  INITIAL TARGET TIME                                       *
C*       TA     MAX(INITIAL SENSOR TIME,INITIAL TARGET TIME)              *
C*       TB     MIN(FINAL   SENSOR TIME,FINAL   TARGET TIME)              *
C*          * OUTPUT *                                                    *
C*       L            NUMBER OF SEGMENTS OF INTERSECTION,=0 IF NONE       *
C*       EPDT(I)   I=1,L  EARLIEST TIME OF SEGMENT I                      *
C*       FLPDT(I)  I=1,L  LATEST   TIME OF SEGMENT I                      *
C*    REMARKS                                                             *
C*        NONE                                                            *
C*                                                                        *
C*                                                                        *
C*                                                                        *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                        *
C*        NONE                                                            *
C*                                                                        *
C*                                                                        *
C*                                                                        *
C*                                                                        *
C*    METHOD                                                              *
C*            SENSOR AND TARGET POSITIONS ARE EXPRESSED PARAMETRICALLY    *
C*        AS FUNCTIONS OF TIME. THE DISTANCES FROM THE CONCENTRIC         *
C*        CIRCLES IS CHECKED TO DETERMINE POSSIBLE SENSOR TARGET          *
C*        DETECTIONS. THE TIMES ARE CHECKED TO ASSURE A TIME INTER-       *
C*        SECTION. THE PARAMETER L IS SET TO 0,1, OR 2 DEPENDING ON       *
C*        THE NUMBER OF SPACE-TIME INTERSECTIONS FOUND                    *
C*                                                                        *
C**************************************************************************
```

Figure 2.1-67

```
C******************************** SECT *********************************
C*                                                                     *
C*                          SUBROUTINE SECT                            *
C*                                                                     *
C*    PURPOSE                                                          *
C*        PRERUN GEOMETRY ROUTINE.  DETERMINES INTERSECTION(S),        *
C*        IF ANY, OF A STRAIGHT LINE WITH A SECTOR                     *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL SECT                                                    *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        ALL INPUT AND OUTPUT VIA COMMON AREA /PGMPAR/.               *
C*                                                                     *
C*          * INPUTS *                                                 *
C*                                                                     *
C*        CVANGL    COVERAGE ANGLE OF SECTOR                           *
C*        DX1       INITIAL X DIST. OF SENSOR FROM TARGET              *
C*        DY1       INITIAL Y DIST. OF SENSOR FROM TARGET              *
C*        DXA       COS(PI/2 -AZIMUTH ANGLE - COVERAGE ANGLE/2)        *
C*        DYA       SIN(PI/2 -AZIMUTH ANGLE - COVERAGE ANGLE/2)        *
C*        DXB       COS(PI/2 -AZIMUTH ANGLE + COVERAGE ANGLE/2)        *
C*        DYB       SIN(PI/2 -AZIMUTH ANGLE + COVERAGE ANGLE/2)        *
C*                                                                     *
C*        TITYM     INITIAL TARGET TIME                                *
C*        TETYM     FINAL   TARGET TIME                                *
C*                                                                     *
C*                                                                     *
C*          * OUTPUTS *                                                *
C*                                                                     *
C*        L         NUMBER OF SEGMENTS OF INTERSECTION, =0 IF NONE     *
C*        EPDT(I)   I =1,L  EARLIEST TIME OF SEGMENT I                 *
C*        FLPDT(I)  I =1,L  LATEST TIME OF SEGMENT I                   *
C*                                                                     *
C*                                                                     *
C*    REMARKS                                                          *
C*        AZIMUTH ANGLE IS MEASURED CLOCKWISE FROM NORTH               *
C*                                                                     *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*        CIRC                                                         *
C*        SECLOG                                                       *
C*                                                                     *
C*    METHOD                                                           *
C*            SUBROUTINE CIRC IS CALLED TO FIND THE POSSIBLE INTER     *
C*        SECTIONS WITH THE CIRCULAR COVERAGE. IF THE COVERAGE         *
C*        ANGLE IS GREATER OR EQUAL TO 6.28 RADIANS NO FURTHER         *
C*        CALCULATIONS ARE MADE.                                       *
C*           FOR TRUE SECTOR TYPE COVERAGE THE INTERSECTION WITH       *
C*        THE SECTOR ARE FOUND BY USE OF SUBROUTINE SECLOG             *
C*           THE PARAMETER L IS SET TO THE NUMBER OF VALID SPACE       *
C*        TIME INTERSECTIONS THAT ARE FOUND (0,1,2,OR 3). THESE        *
C*        INTERSECTIONS ARE FOUND BY USE OF THE FLECK/BUTLER FORTRAN   *
C*    'AND' ALGORITHM                                                  *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-68

```
C******************************* GREC ********************************
C*                                                                   *
C*                         SUBROUTINE GREC                           *
C*                                                                   *
C*   PURPOSE                                                         *
C*       PRERUN GEOMETRY ROUTINE.  DETERMINES INTERSECTION(S),       *
C*       IF ANY, OF A STRAIGHT LINE WITH A MOVING RECTANGLE          *
C*                                                                   *
C*                                                                   *
C*                                                                   *
C*   CALLING SEQUENCE                                                *
C*       CALL GREC                                                   *
C*                                                                   *
C*   DESCRIPTION OF PARAMETERS                                       *
C*       ALL INPUT AND OUTPUT VIA COMMON AREA /PGMPAR/.              *
C*        * INPUTS *                                                 *
C*                                                                   *
C*       DX      X COMPONENT OF TARGET PATH                          *
C*       DY      Y COMPONENT OF TARGET PATH                          *
C*       DX1     INITIAL X DISTANCE OF SENSOR FROM TARGET            *
C*       DY1     INITIAL Y DISTANCE OF SENSOR FROM TARGET            *
C*       DXB     COSINE OF ANGLE OF SENSOR MOVEMENT                  *
C*       DYB     SINE OF ANGLE OF SENSOR MOVEMENT                    *
C*       TA      MAX(INITIAL SENSOR TIME,INITIAL TARGET TIME)        *
C*       TB      MIN(  FINAL SENSOR TIME   FINAL TARGET TIME)        *
C*       DT      TOTAL TARGET TIME INTERVAL                          *
C*       TITYM   INITIAL TARGET TIME                                 *
C*       DA      CROSS TRACK DIST. MINUS 0.5 WIDTH OF RECTANGLE      *
C*       DB      DA + RECTANGLE WIDTH                                *
C*       DC      ALONG TRACK DIST. MUNUS 0.5 LENGTH OF RECTANGLE     *
C*       RMAX    LENGTH OF RECTANGLE                                 *
C*       VS      VELOCITY OF MOVING SENSOR                           *
C*                                                                   *
C*        * OUTPUTS *                                                *
C*       L             NUMBER OF SEGMENTS OF INTERSECTION, =0 IF NONE *
C*       EPDT(I)   I=1,L  EARLIEST TIME OF SEGMENT I                 *
C*       FLPDT(I)  I=1,L  LATEST TIME OF SEGMENT I                   *
C*                                                                   *
C*   REMARKS                                                         *
C*       NONE                                                        *
C*   SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*       NONE                                                        *
C*                                                                   *
C*   METHOD                                                          *
C*           THE POSITIONS OF THE SENSOR AND TARGET ARE EXPRESSED    *
C*       PARAMETRICALLY AS A FUNCTION OF TIME. POSSIBLE INTER-       *
C*       SECTIONS WITH THE BOUNDARIES PARALLEL TO THE RELATIVE MOTION *
C*       ARE CHECKED FIRST. IF ANY EXIST THE INTERSECTIONS WITH THE  *
C*       PERPENDICULAR SPACE TIME BOUNDARIES ARE DETERMINED. THE     *
C*       PARAMETER L IS SET TO THE NUMBER OF INTERSECTIONS (0 OR 1)  *
C*                                                                   *
C*********************************************************************
```

Figure 2.1-69

```
C****************************** SECLOG *************************************
C*                                                                        *
C*                       SUBROUTINE SECLOG(A,B,C,K)                       *
C*                                                                        *
C*     PURPOSE                                                            *
C*         PRERUN GEOMETRY ROUTINE.  DETERMINES IF A POINT LIES IN        *
C*         A SECTOR                                                       *
C*     CALLING SEQUENCE                                                   *
C*         CALL SECLOG(A,B,C,K)                                           *
C*                                                                        *
C*     DESCRIPTION OF PARAMETERS                                          *
C*           * INPUT *                                                    *
C*         A DISTANCE OF POINT FROM LINE L1                               *
C*         B DISTANCE OF POINT FROM LINE L2                               *
C*         C CROSS PRODUCT OF UNIT VECTORS DESCRIBING SECTOR (L1*L2)      *
C*                                                                        *
C*          * OUTPUT *                                                    *
C*         K =0 POINT NOT IN SECTOR                                       *
C*           =1 POINT IN SECTOR                                           *
C*     REMARKS                                                            *
C*         NONE                                                           *
C*                                                                        *
C*                                                                        *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                       *
C*         NONE                                                           *
C*                                                                        *
C*                                                                        *
C*                                                                        *
C*                                                                        *
C*                                                                        *
C*                                                                        *
C*     METHOD                                                             *
C*             THE POSITION (+ OR -) RELATIVE TO THE SECTOR BOUNDARIES    *
C*         ARE CHECKED TO DETERMINE POSSIBLE INTERSECTIONS                *
C*                                                                        *
C*                                                                        *
C**************************************************************************
```

Figure 2.1-70

```
C******************************* PREMN9 ***********************************
C*                                                                        *
C*                  PRERUN EXECUTIVE -     STEP  9                        *
C*                                                                        *
C*     PURPOSE                                                            *
C*         EXECUTIVE TO CALL DUMMY LOS OR OTHER ALTERNATE LOS ROUTINES    *
C*                                                                        *
C*     USAGE                                                              *
C*         MAIN PROGRAM                                                   *
C*                                                                        *
C*     DESCRIPTION OF PARAMETERS                                          *
C*                                                                        *
C*           * INPUT *                                                    *
C*         ZEV  AN ARRAY CONTAINING 22 PARAMETERS FOR EACH SENSOR-TARGET*
C*         DETECTION (SEE COMMENTS IN ELPUT OR MAINLS)                    *
C*                                                                        *
C*         *  OUTPUT *                                                    *
C*         NEV(1, )   SENSOR ID                                           *
C*         NEV(2, )   TARGET ID                                           *
C*         NEV(3, )   EARLIEST POSSIBLE DETECTION TIME                    *
C*         NEV(4, )   LATEST   POSSIBLE DETECTION TIME.                   *
C*                                                                        *
C*     REMARKS                                                            *
C*             THE SIMPLEST  VERSION OF LSGT WHICH ASSUMES PERFECT LINE   *
C*         OF SIGHT (FLAT EARTH) IS SUPPLIED.  MORE ELABORATE VERSIONS    *
C*         MAY BE WRITTEN AND USED WITH THIS EXECUTIVE ROUTINE IF         *
C*         DESIRED.                                                       *
C*             IF THE DETAILED LINE OF SIGHT IS DESIRED EXECUTIVE         *
C*         ROUTINE MAINLS SHOULD BE USED INSTEAD OF PREMN9                *
C*                                                                        *
C*     SUBROUTINES REQUIRED                                               *
C*         LSGT                                                           *
C*                                                                        *
C*     METHOD                                                             *
C*             THE COMMON GAME INFORMATION AND THE ZEV TABLES ARE READ    *
C*         SUBROUTINE LSGT IS CALLED TO GENERATE THE NEV TABLE AND        *
C*         THE NEV ARE WRITTEN ON MTAPE(16). A ONE AND FOUR ZEROES ARE    *
C*         WRITTEN ON MTAPE(16) TO INDICATE THE END OF FILE.              *
C*                                                                        *
C**************************************************************************
```

Figure 2.1-71

```
C******************************  LSGT  **********************************
C*                                                                      *
C*                         SUBROUTINE LSGT                              *
C*                                                                      *
C*    PURPOSE                                                           *
C*        DUMMY LOS ROUTINE - READS IN THE ZEVS AND WRITES THE          *
C*        NEV ARRAY -  EQUIVALENT TO LINE OF SIGHT ALWAYS THERE         *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL LSGT(M,ZEV,N,NEV)                                        *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*          * INPUT *                                                   *
C*        M    NO. ITEMSIN TABLE ZEV                                    *
C*        ZEV  ARRAY OF EVENTS                                          *
C*                                                                      *
C*          * OUTPUT *                                                  *
C*        N    NO. ITEMS IN TABLE NEV                                   *
C*        NEV  ARRAY OF EVENTS                                          *
C*                                                                      *
C*    REMARKS                                                           *
C*        NONE                                                          *
C*                                                                      *
C*    SUBROUTINES REQUIRED                                              *
C*        NONE                                                          *
C*                                                                      *
C*    METHOD                                                            *
C*        THE FIRST FOUR ENTRIES IN THE ZEV TABLE ARE TRANSFERRED TO    *
C*        THE NEV TABLE. THIS HAS THE EFFECT OF ASSUMING LINE OF SIGHT  *
C*        ALWAYS EXISTS.                                                *
C*                                                                      *
C*                                                                      *
C***********************************************************************
```

Figure 2.1-72

```
C******************************** MAINLS ********************************
C*                                                                      *
C*                            MAIN PROGRAM                              *
C*                                                                      *
C*    PURPOSE                                                           *
C*        COMPUTE LINE-OF-SIGHT FOR ALL SCANNING TYPE SENSORS.         *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        MAIN PROGRAM                                                  *
C*                                                                      *
C*    REMARKS                                                           *
C*        PROGRAM REQUIRES EARLIEST AND LATEST POSSIBLE DETECTION       *
C*        INFORMATION AND THE TERRAIN TAPE.                             *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION ROUTINES REQUIRED.                        *
C*        LOS      COMPUTES LINE OF SIGHT                               *
C*        TERAN    GENERATES UNTER AND UTVSXY DATA IN LABELLED COMMON.  *
C*                                                                      *
C************************************************************************
```

Figure 2.1-73

```
C****************************** LOS     **********************************
C*
C*    PURPOSE
C*      TO DETERMINE WHETHER LINE OF SIGHT EXISTS BETWEEN POINTS 1
C*    AND 2 ON A TERRAIN, GIVEN X, Y COORDINATES OF POINTS AND OFFSETS
C*      (H'S) ABOVE GROUND. POINT 1 IS REGARDED AS SENSOR, POINT 2
C*    AS TARGET.
C*
C*    USAGE
C      CALL LOS (X1, Y1, H1, X2, Y2, H2, SRANGE, LFLPEN, RCLEAR, KSECUR,
C                 D, LZO)
C*
C*      X1, Y1        POSITION COORDINATES, POINT 1 (METERS)
C*      X2, Y2        POSITION COORDINATES, POINT 2 (METERS)
C*      H1, H2        OFFSETS ABOVE GROUND FOR POINTS 1, 2 RESP. (METERS)
C*      SRANGE        SENSOR RANGE (METERS)
C*      LFLPEN        IF FOLIAGE PENETRATING, SET TO TRUE, OTHERWISE FALSE
C*       RCLEAR       FOLIAGE FREE DISTANCE AT SENSOR (METERS)
C*       KSECUR       ZERO, IF TARGET TAKES ADVANTAGE OF COVER- OTHERWISE 1
C*       D            DISTANCE TARGET HAS MOVED FROM LAST CALL-NECESSARY
C*       D            FOR LINKAGE TO SUBROUTINE  MICTER
C*                    OUTPUT PARAMETERS
C*      LZO           LOGICAL VARIABLE. .TRUE. IF LOS EXISTS,
C*                    OTHERWISE .FALSE.
C*    REMARKS
C*      A .FALSE. RETURN CAN OCCUR FOR ONE OF THESE REASONS--
C*              1. TERRAIN + FOLIAGE CONTOUR BLOCKS LINE OF SIGHT PATH
C*              2. LOCAL EFFECTS (MICRO-STRUCTURE OF TERRAIN + FOLIAGE
C*                 + CULTURAL UNITS) BLOCK LOS.
C*              3. DISTANCE FROM SENSOR TO TARGET EXCEEDS SENSOR RANGE
C*     THIS ROUTINE PRIMARILY HANDLES CALCULATION FOR THE FIRST OF
C*     THESE. ALTHOUGH THE CALLING PROGRAM (EXECUTIVE ROUTINE)
C*     SHOULD NORMALLY NOT REQUEST A LOS DETERMINATION FOR SENSOR-
C*     TARGET DISTANCES BEYOND SENSOR RANGE, THE POSSIBILITY IS
C*     CHECKED. THE ADDED COMPLEXITY IS MINOR, AND THIS CHECK
C*     ALLOWS PROGRAM TO HANDLE CURRENTLY UNPLANNED TASKS, AND
C*     ALSO PROVIDES PROTECTION AGAINST INPUT ERRORS.
C*     LOCAL MASKING, IN IMMEDIATE VICINITY OF SENSOR OR TARGET,
C*     IS DETERMINED EXTERNALLY (THAT IS, THIS ROUTINE REQUESTS
C*     LOCAL MASKING EFFECT BY CALLING ANOTHER SUBPROGRAM)
C*
C*    SUBROUTINES AND FUNCTIONS REQUIRED:
C*              TERANE  (FETCHES REQUIRED TERRAIN)
C*              FOLAGE  (ADDS FOLIAGE EFFECTS TO SENSOR/TARGET)
C*              MICTER  (ADDS MICRO-ENVIRONMENTAL EFFECTS TO SENSOR/TARGET)
C*              IUTEVL  (COMPUTES AN INDEX BASED ON X,Y POSITION FOR USE
C*                      IN LABEL COMMON HUNTER*
C*
C*    METHOD FOR TERRAIN RETRIEVAL FOR SHORT RANGE SENSORS.
C*      (FIVE KILOMETERS IN MAXIMUM RANGE)
```

Figure 2.1-74

```
C*
C*     THE ARRAY MAP IS DESIGNED TO HOLD IN CORE TERRAIN EXTENDING 30
C*    KILOMETERS IN THE X DIRECTION AND 10 KILOMETERS IN THE Y DIRECTION.
C*    THESE FIGURES ARE BASED ON A TERRAIN GRID RESOLUTION OF 100 METERS.
C*    WHEN TERRAIN IS REQUIRED FOR A PARTICULAR SHORT RANGE SENSOR, A
C*    "STRIP" OF TERRAIN WHOSE DIAGONAL COORDINATES CORRESPOND TO
C*    MINIMUM AND MAXIMUM PAIRS OF X AND Y IS BROUGHT INTO CORE. THESE
C*    COORDINATES ARE CHOSEN SO THAT THE SENSOR IS CENTERED WITHIN THE
C*    "STRIP". SUCCEEDING SENSORS COORDINATES ARE THEN CHECKED AGAINST
C*    THE "STRIP" COORDINATES TO ESTABLISH NEED FOR A SUBSEQUENT TERRAIN
C*    RETRIEVAL. EXCEPT FOR PLAY FIELD EDGE EFFECTS, THE CENTERING OF
C*    SENSOR TO TERRAIN WILL BE ADHERED TO.
C*
C*    METHOD FOR TERRAIN RETRIEVAL FOR LONG RANGE SENSORS.
C*      (MORE THAN 5 KILOMETERS BUT LESS THAN OR EQUAL TO 12 KILOMETERS
C*      IN RANGE)
C*
C*     THE ARRAY MAP WILL HOLD IN CORE 24 KILOMETERS OF TERRAIN IN THE
C*    X DIRECTION  AND 12 KILOMETERS IN THE Y DIRECTION. THE CENTERING
C*    TECHNIQUE DESCRIBED ABOVE WILL BE EXECUTED FOR A SEMI- CIRCLE OF
C*    RADIUS 12 KILOMETERS AND CENTERED IN THE X DIRECTION ON EITHER
C*    THE UPPER OR LOWER EDGE OF THE"STRIP".
C*
C*    METHOD FOR LINE OF SIGHT DETERMINATION.
C*
C*     TERRAIN HEIGHT FOR EACH INCREMENT IN DISTANCE FROM SENSOR TO
C*    TARGET IS DETERMINED AND SIMPLY COMPARED TO THE RISE IN THE LINE OF
C*    SIGHT RAY IN THAT INCREMENT. MASKING OCCURS WHEN THE LOCAL TERRAIN
C*    OBSCURES THE LINE OF SIGHT RAY.
C*
C*    GLOSSARY:
C*
C ITRNHT LOS    CP  INTEGERIZED TERRAIN IN STORAGE
C SHORT  LOS    CP  RANGE OF SHORT RANGE SENSORS
C DGD    LOS    CP  GRID SIZE
C XREF   LOS    CP  X-ORIGIN OF PLAYING FIELD (NORMALLY=0. FOR COMPAT-
C        LOS    CP  IBILITY WITH UNTER TABLES)
C YREF   LOS    CP  Y-ORIGIN OF PLAYING FIELD (SAME REMARKS AS XREF)
C XRANGE LOS    CP  MAXIMUM X LENGTH OF PLAYING FIELD (METERS)
C YRANGE LOS    CP  MAXIMUM Y LENGTH OF PLAYING FIELD (METERS)
C MXREF  LOS    CP  INTEGERIZED X REFERENCE OF PLAYING FIELD
C MYREF  LOS    CP  INTEGERIZED Y REFERENCE OF PLAYING FIELD
C PLIMIT LOS    DP  MAXIMUM SENSOR RANGE ACCOMODATED BY STORED TERRAIN
C MXTENT LOS    CP  INTEGERIZED X RANGE OF PLAYING FIELD
C MINX   LOS    DP  COMPUTED MINIMUM X INDEX REQUIRED FOR STORED TERRAIN
C MYTENT LOS    CP  INTEGERIZED Y RANGE OF PLAYING FIELD
C DRAY   LOS    CP  INCREMENTAL DISTANCE ALONG LINE OF SIGH IN XY PLANE
C MINY   LOS    DP  COMPUTED MINIMUM Y INDEX REQUIRED FOR STORED TERRAIN
C IX1    LOS    DP  INTERNAL VARIABLE (SENSOR X COORDINATE)
C IY1    LOS    DP  INTERNAL VARIABLE (SENSOR Y COORDINATE)
```

Figure 2.1-74 (Cont.)

```
C IX2     LOS     DP   INTERNAL VARIABLE (TARGET X COORDINATE)
C IY2     LOS     DP   INTERNAL VARIABLE (TARGET Y COORDINATE)
C MAXX    LOS     DP   COMPUTED MAXIMUM X INDEX REQUIRED FOR STORED TERRAIN
C MAXY    LOS     DP   COMPUTED MAXIMUM Y INDEX REQUIRED FOR STORED TERRAIN
C MINYO   LOS     DP   INTERNAL VALUE
C MINXO   LOS     DP   INTERNAL VALUE
C R12     LOS     DP   DISTANCE BETWEEN POINTS 1&2 IN THE XY PLANE
C COSTH   LOS     DP   COSINE AZIMUTH ANGLE TO R12
C SINTH   LOS     DP   SINE AZIMUTH ANGLE TO R12
C FETCH   LOS     DP   TERRAIN READ ON FIRST RETRIEVAL
C DR      LOS     DP   LINE OF SIGHT PROJECTION ALONG R12
C X       LOS     DP   INTERNAL GENERALIZED X VALUE
C Y       LOS     DP   INTERNAL GENERALIZED Y VALUE
C Z0      LOS     DP   INTERNAL GENERALIZED  OFFSET VALUE ABOVE LOCAL TERRAIN
C X0      LOS     DP   INTERNAL VALUE
C F1      LOS     DP   INTERNAL VALUE OF TERRAIN INDEX
C F2      LOS     DP   INTERNAL VALUE OF TERRAIN INDEX
C F3      LOS     DP   INTERNAL VALUE OF TERRAIN INDEX
C F4      LOS     DP   INTERNAL VALUE OF TERRAIN INDEX
C Z11     LOS     DP   INTERNAL VALUE- TERRAIN HEIGHT INTERPOLATION
C Z22     LOS     DP   INTERNAL VALUE- TERRAIN HEIGHT INTERPOLATION
C ZPT     LOS     DP   LOCAL TERRAIN INTERPOLATED HEIGHT
C ZSENS   LOS     DP   LOCAL TERRAIN HEIGHT FOR SENSOR
C ZTRGT   LOS     DP   LOCAL TERRAIN HEIGHT FOR TARGET
C DZDR    LOS     DP   TANGENT OF ELEVATION ANGLE TO LINE OF SIGHT RAY
C ZTST    LOS     DP   TEST VALUE FOR LINE OF SIGHT OR MASK
C LZO     LOS     DP   LINE OF SIGHT = T(RUE) OR F(ALSE) ON MASK
C LFLPEN  LOS     DP   LOGICAL VARIABLE IS .TRUE. FOR FOLIAGE PENETRATION
C RMICTR  LOS     DP   BASE DISTANCE FOR SUBROUTINE MICTER (NORMALLY 250
C         LOS     DP   METERS , BUT MAY BE LESS)
C IWORK   LOS     DP   BUFFER WORK AREA DESIGNED TO HOLD X TERRAIN AT CONSTANT
C*        LOS          Y. DIMENSION OF  513 PERMITS AN Y EXTENT OF 51.3
C*        LOS          KILOMETERS  ALLOWABLE IN THE PLAYING FIELD.
C*        LOS          THIS FIGURE IS BASED ON THE CONSTUCTION AF A "SPARSE"
C*        LOS          TAPE FROM THE ORIGINAL TOPOCOM SOURCE TAPE
C*
C*
C*
C*************************************************************************
```

Figure 2.1-74 (Cont.)

```
C***********************************************************************************
C******************************  MICTER  *******************************
C*                                                                        *
C*                       SUBROUTINE MICTER                                *
C*                                                                        *
C*     PURPOSE                                                            *
C*         AN AUXILIARY TO THE BASIC LOS ROUTINE, MICTER DETERMINES       *
C*         WHETHER A TARGET WOULD BE VISIBLE OR NOT, INSOFAR AS THE       *
C*         MICRO STRUCTURE OF ENVIRONMENT IN THE IMMEDIATE VICINITY       *
C*         OF THE TARGET IS CONCERNED.                                    *
C*                                                                        *
C*     CALLING SEQUENCE                                                   *
C*         CALL MICTER (RMICTR,D,IUT,DZDR,KSECUR, LOSMIC)                 *
C*                                                                        *
C*                                                                        *
C*     SUBROUTINES AND FUNCTIONS REQUIRED                                 *
C*                                                                        *
C*         URN        UNIFORM RANDOM NUMBER GENERATOR                     *
C*                                                                        *
C*     DESCRIPTION OF PARAMETERS                                          *
C*         NOTE- FIRST 5 PARAMETERS IN CALLING SEQUENCE ARE INPUT         *
C*              VARIABLES TO MICTER, LAST PARAMETER (LOSMIC)              *
C*              IS OUTPUT VARIABLE.                                       *
C*                                                                        *
C*         RMICTR     DISTANCE FROM TARGET, OVER WHICH MICTER HAS         *
C*                      RESPONSIBILITY. (METERS)                          *
C*         D          DISTANCE TARGET HAS MOVED SINCE LAST CALL TO        *
C*                      MICTER, FOR MOVING TARGET.  LENGTH OF TARGET,     *
C*                      FOR STATIONARY TARGET.  FOR FIRST CALL ON         *
C*                      MOVING TARGET, D SHOULD BE ZERO.  (METERS)        *
C*         IUT        UNIT TERRAIN INDEX (INTEGER) AT TARGET POSITION     *
C*         DZDR       VERTICAL SLOPE OF LINE CONNECTING SENSOR AND        *
C*                      TARGET.  NEGATIVE VALUE CORRESPONDS TO SENSOR     *
C*                      'LOOKING DOWN' AT TARGET (E.G.,AIRBORNE SENSOR).  *
C*         KSECUR     TARGET PARAMETER (INTEGER) ORIGINATING IN PLANNER   *
C*                      INPUT TABLES.  VALUE IS 0 IF TARGET IS TO BE      *
C*                      ASSUMED TO TAKE MAXIMUM ADVANTAGE OF COVER,       *
C*                      VALUE = 1 OTHERWISE                               *
C*                                                                        *
C*         LOSMIC     LOGICAL VARIABLE, OUTPUT.  HAS VALUE .TRUE. IF      *
C*                      VISIBILITY IS IMPLIED BY MICTER CALCULATIONS,     *
C*                      HAS VALUE .FALSE. IF LOCAL MASKING IS IMPLIED.    *
C*                                                                        *
C*     METHOD                                                             *
C*         1. THERE IS ONE MAJOR BRANCH IN THE PROGRAM, BASED ON          *
C*            THE DZDR VARIABLE.  FOR DZDR LESS THAN -1.0, VISI-          *
C*            BILITY IS BASED ON CANOPY CLOSURE  (THIS CASE WOULD         *
C*            NORMALLY OCCUR FOR AN AIRBORNE SENSOR, BUT COULD OC-        *
C*            CUR FOR OBSERVATION OF VALLEY REGIONS FROM HIGH             *
C*            POSITIONS.)                                                 *
```

Figure 2.1-75

```
C*    2. IF GROUND-TO-GROUND VISIBILITY MUST BE CHECKED, THE FOLLOWING CAL-    *
C*       CULATIONS ARE MADE:                                                   *
C*                                                                             *
C*       A. IF THE "REGION OF RESPONSIBILITY" OF MICTER IS GREATER THAN THE   *
C*          UPPER VISIBILITY LIMIT, IT IS ASSUMED LOS DOES NOT EXIST AND      *
C*          THE SUBROUTINE IS EXITED.                                         *
C*                                                                             *
C*       B. IF THE "REGION OF RESPONSIBILITY" (RMICTR) IS LESS THAN THE       *
C*          LOWER LIMIT OF VISIBILITY, IT IS ASSUMED LOS DOES EXIST AND THE   *
C*          SUBROUTINE IS EXITED.                                             *
C*                                                                             *
C*       C. IF RMICTR LIES BETWEEN THE UPPER AND LOWER LIMITS OF VISIBIL-     *
C*          ITY, LOS IS RANDOMLY AS FOLLOWS:                                  *
C*                                                                             *
C*          (1) PROBABILITY OF LOS IS DETERMINED USING LINEAR INTERPRE-       *
C*              TATION.                                                       *
C*                                                                             *
C*          (2) THE NUMBER OF RAYS OR AZIMUTHS ALONG WHICH LOS IS TO BE       *
C*              CHECKED IS CALCULATED BASED ON DISTANCE TARGET MOVED AND      *
C*              LOWER LIMIT OF VISIBILITY.                                    *
C*                                                                             *
C*          (3) IF THE TARGET IS ATTEMPTING TO AVOID DETECTION, THE NUMBER    *
C*              OF "LOOKS" IS DIVIDED BY TWO.                                 *
C*                                                                             *
C*          (4) THE PROBABILITY OF LINE-OF-SIGHT IS SET AS 1- PROBABILITY     *
C*              OF NOT HAVING LOS RAISED TO THE EFNLKS POWER AND IS COM-      *
C*              PARED WITH A UNIFORM RANDOM NUMBER.                           *
C*                                                                             *
C*    3. IF AIR-TO-GROUND VISIBILITY MUST BE CHECKED, THE CALCULATIONS ARE    *
C*       THE SAME EXCEPT THAT THE PROBABILITY OF NOT HAVING LINE-OF-SIGHT     *
C*       IS SET AT A NUMBER CHOSEN RANDOMLY BETWEEN THE UPPER AND LOWER       *
C*       PERCENT CANOPY CLOSURE VALUES.                                       *
C*                                                                             *
C******************************************************************************
```

Figure 2.1-75 (Cont.)

```
C******************************* TERANE ***********************************
C*
C*            SUBROUTINE TERANE (MINX,MAXX,MINY, MAXY, MAP, IDMX, IDMY, IDUM)
C*
C*    PURPOSE:
C*             TO BRING INTO THE ARRAY"MAP,"A TERRAIN MAP SECTION RESIDENT
C*           ON AN EXTERNAL MEDIUM SEQUENTIALLY ORGANIZED. SPECIFICALLY
C*           EACH RECORD ON THE MEDIUM IS CONSIDERED A "SCAN LINE" AT A
C*           CONSTANT "X" AND CONSISTS OF ENOUGH "Y" ORDINATE POINTS TO
C*           DEFINE A PLAYING FIELD.
C*      "PLAYING FIELD" AS USED HERE INCLUDES  ALL THE TERRAIN RESIDENT ON
C*    TAPE OR DISK.
C*
C*    ARGUMENTS:
C*                     MINX, MAXX, MINY, MAXY, MAP, IDMX, IDMY, IDUM
C*
C*    SUBROUTINES REQUIRED: NONE
C*
C*      REMARKS:
C*
C*      THE "END" PARAMETER IS USED IN THE READ OF TERRAIN FROM TAPE
C*    OR DISK TO SIGNIFY END OF PLAY FIELD DATA(LAST "SCAN" OF X- DATA)
C*                 "END" COULD BE ELIMINATED IF INPUT DEVICE INDICATES
C*      NUMBER OF RECORD TO BE READ ON EXTERNAL MEDIUM.
C*
C*    GLOSSARY:
C MINX    TERANE DP  X ABSCISSA COUNT INITIATE X READ
C MAXX    TERANE DP  X ABSCISSA COUNT TERMINATE X READ
C MINY    TERANE DP  Y ORDINATE COUNT INITIATE Y READ
C MAXY    TERANE DP  Y ORDINATE COUNT TERMINATES Y READ
C MAP     TERANE DP  TERRAIN ARRAY IN CORE
C IDMX    TERANE DP  X DIMENSION OF "MAP" (NORMALLY: 301 FOR SHORT
C*        TERANE                RANGE SENSORS, 241 FOR LONG RANGE SENSORS)
C IDMY    TERANE DP  Y DIMENSION OF "MAP" (NORMALLY: 101 FOR SHORT
C*        TERANE                RANGE SENSORS, 121 FOR LONG RANGE SENSORS)
C LASREC  TERANE DP  LAST RECORD READ FROM MEDIUM
C IDUM    TERANE     BUFFER AREA TO HOLD ONE "SCAN" OF TERRAIN AT CONSTANT Y
C XMIN    TERANE CP  X COORDINATE OF TERRAIN HELD IN CORE
C YMIN    TERANE CP  Y COORDINATE OF TERRAIN HELD IN CORE
C XMAX    TERANE CP  MAXIMUM X COORDINATE OF TERRAIN HELD IN CORE
C YMAX    TERANE CP  MAXIMUM Y COORDINATE OF TERRAIN HELD IN CORE
C MXREF   TERANE CP   INTEGERIZED X ORIGIN OF PLAYING FIELD
C MXTENT  TERANE CP   INTEGERIZED X EXTENT OF PLAYING FIELD
C MYTENT  TERANE CP INTEGERIZED Y RANGE OF PLAYING FIELD
C IREAD   TERANE DP NUMBER OF TIMES THIS ROUTINE IS ENTERED
C MASTAP  TERANE DP TAPE OR DISK UNIT NUMBER
C XLONG   TERANE DP LENGTH OF X AXIS OF TERRAIN
C YLONG   TERANE DP LENGTH OF Y AXIS OF TERRAIN
C IOUT    TERANE DP NUMBER OF OUTPUT DEVICE
C*      METHOD: See 2-118.5.
```

Figure 2.1-76

```
C*     METHOD                                                                  *
C*               THE LENGTH OF THE PLAYING FIELD IS DETERMINED.  A CHECK IS    *
C*          MADE TO SEE IF THE WEST BOUNDARY OF THE AREA OF INTEREST IS EAST   *
C*          OF THE LAST RECORD (BLOCK OF TERRAIN) READ FROM TAPE.. IF IT IS,   *
C*          THE TAPE MUST BE ADVANCED TO FIND THE DATA OF INTEREST.  IF NOT,   *
C*          IT MUST BE REWOUND.  A CHECK IS THEN MADE TO SEE IF THE WEST EDGE  *
C*          OF THE AREA OF INTEREST IS THE FIRST SCAN LINE.  IF IT IS, THEN THE *
C*          TAPE IS ADVANCED TO THE DESIRED SCAN LINE.  A NESTED DO-LOOP IS    *
C*          ENTERED WHICH EXTRACTS DATA FROM XMIN TO XMAX FROM THE TAPE AND    *
C*          PUTS IT ONTO LIST.  EXTENT OF THE DATA IN THE NORTH-SOUTH DIRECTION *
C*          ON LIST IS ENTIRE PLAYING FIELD.  DATA IS TRANSFERRED FROM A TEM-  *
C*          PORARY WORK AREA (IDUM) TO A TWO DIMENSIONAL TERRAIN ARRAY IN CORE *
C*          (MAP).  THIS DATA IS FROM MINY TO MAXY AND WILL VARY IN SIZE DEPEND- *
C*          ING ON SENSOR TYPE.  THE COORDINATES OF THE MAP AREA ARE CALCU-    *
C*          LATED AND THE SUBROUTINE IS EXITED.  A PRINT OPTION EXISTS AND RE- *
C*          QUIRES CONVERSION OF A COMMENT CARD TO A PRINT STATEMENT.          *
C*                                                                             *
C******************************************************************************
```

Figure 2.1-76 (Cont.)

```
C******************************* BLKLOS ********************************
C*                                                                     *
C*    PURPOSE                                                          *
C*       BLOCK DATA USED TO SET PARAMETERS XREF,YREF,XRANGE,YRANGE,    *
C*       LDUMP,IPRINT IN THE LOS PROGRAM.                              *
C*                                                                     *
C*       ABOVE PARAMETERS ARE DEFINED IN LABELED COMMON BLOCKS         *
C*       LOSCOM AND CONST.                                             *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-77

2.1.5.11 PRERUN Step 10

Step 10 in PRERUN comprises the main program (PREMNA) with 2 subroutines listed in Table 2.1-V, plus 4 utility subroutines. External data sets required as input are "JFMSTR", "CRTARG", "JFNEV" and "EVNTLS", all generated in previous steps.

This step creates events type 1 (sensor interrogate). False target information is merged into the event 1 lists where required. Event 1 lists are then stored on disk (JFEVLC). Figures 2.1-78 through 2.1-80 describe the subroutines of this step.

2.1.5.12 PRERUN Step 11

Step 11 in PRERUN comprises the main program (PREMNB) with 1 subroutine FMERGE plus the utility subroutine GMERGE. External data sets required as input are "JFMSTR", "MASSEV23", "JFEVLC", "CREVT9", and "JFEVT", all generated in previous steps. This step collects and merges all events of all types that have been generated and stored in the previous PRERUN steps. The merged sequence of events are stored on disc (JFIEV). Figures 2.1-81 and 2.1-82 describe the subroutines of this step.

2.1.5.13 PRERUN Step 12

Step 12 in PRERUN has one (main) program (PREMNC), and requires only 1 external data set "JFIEV" as input. This step takes all the merged events, blocks them for MSM (900 or fewer words/block), and stores them on disc (DSNAME=EVENT1).

Figure 2.1-83 describes subroutine PREMNC.

2.1.5.14 Utility Subroutines

Most of the PRERUN steps use various utility type subroutines. These are listed in Table 2.1-V and Figures 2.1-84 through 2.1-91 describe these subroutines.

2.1.6 PRERUN Common Areas

Common areas are used to allocate storage (and hence limit the size of the model). They serve to transmit information between the various programs of a job step. Included in this section is a table of PRERUN common areas (Table 2.1-VI) arranged alphabetically. For each area, a brief description of the variables is given, along with a list of all subroutines using the common statement. If any dimension statement is changed, all subroutines using the common area involved must be recompiled.

```
C******************************** PREMNA *********************************
C*                                                                       *
C*                 PRERUN EXECUTIVE -    STEP 10                         *
C*                                                                       *
C*    PURPOSE                                                            *
C*       GENERATES MSM TYPE 1 EVENTS FROM OUTPUT OF ELPDT AND LOS        *
C*       ADDS FALSE TARGET INFORMATION                                   *
C*                                                                       *
C*    USAGE                                                              *
C*       MAIN PROGRAM                                                    *
C*                                                                       *
C*    DESCRIPTION OF PARAMETERS                                          *
C*       ALL INPUT AND OUTPUT VIA COMMON BLOCKS/ WEVT,UTMCOM,TIMES,      *
C*       INOUT,OUTP,OPTION,POSERR/                                       *
C*                                                                       *
C*    REMARKS                                                            *
C*       NONE                                                            *
C*    SUBROUTINES REQUIRED                                               *
C*       DORDER  -TO ORDER TIMES                                         *
C*       SEQ     -TO GENERATE MSM EVENTS                                 *
C*       GMERGE  -TO MERGE EVENTS                                        *
C*       FLSTG   -TO ADD FALSE TARGET INFO                               *
C*                                                                       *
C*    METHOD                                                             *
C*          THE COMMON GAME INFORMATION AND THE FALSE TARGET             *
C*       INFORMATION ARE READ.                                           *
C*          THE NON LINE OF SIGHT DETECTIONS (NEV) ARE READ. EACH        *
C*       SENSOR IS A SEPERATE RECORD.  THE ORDER OF THE EARLIEST AND     *
C*       OF THE LATEST DETECTION TIMES IS DETERMINED BY CALLING          *
C*       DORDER.  SEQ IS THEN CALLED TO CREATE EVENT TYPE 1. GMERGE      *
C*       IS CALLED TO MERGE THEM WITH THE MASTER LIST. AFTER A BLOCK     *
C*       OF MAX2 IN LENGTH HAS BEEN GENERATED FLSTG IS CALLED TO ADD     *
C*       THE FALSE TARGET INFORMATION TO THE EVENT AND THE MASTER        *
C*       LIST IS WRITTEN ON MTAPE(11)-JFEVLC AND A NEW MASTER LIST IS    *
C*       STARTED. THE PRINT OPTION IS CHECKED  AS EACH SET OF NEV IS     *
C*       READ.                                                           *
C*          THE LINE OF SIGHT DETECTIONS ARE PROCESSED BY THE SAME       *
C*       SEQUENCE OF INSTRUCTIONS.                                       *
C*          A ONE-ZERO IS WRITTEN ON MTAPE(11) TO INDICATE THE END       *
C*       OF FILE. THE NUMBER OF RECORDS AND THE MAXIMUM LENGTH IS        *
C*       PRINTED IF DESIRED (MPRINTOUT READ)                             *
C*                                                                       *
C*************************************************************************
```

Figure 2.1-78

```
C******************************* FLSTG *********************************
C*                                                                     *
C*                        SUBROUTINE FLSTG                             *
C*                                                                     *
C*    PURPOSE                                                          *
C*        TO ADD FALSE TARGET INFORMATION TO TYPE 1 EVENTS             *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL FLSTG(NEV,IEV,NTAR,TARG,MAX)                            *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*                                                                     *
C*    REMARKS                                                          *
C*        NONE                                                         *
C*                                                                     *
C*    SUBROUTINES REQUIRED                                             *
C*        TRNSFR                                                       *
C*                                                                     *
C*    METHOD                                                           *
C*            THE EVENTS IN IEV ARE PUSHED TO THE TOP OF THE STORAGE   *
C*        AS SET BY MAX. EACH EVENT IS SCANNED FOR A NEGATIVE TARGET   *
C*        ID WHICH INDICATES A FALSE TARGET. IF NO FALSE TARGET IS     *
C*        PRESENT THE EVENT IS ADDED TO THE IEV TABLE WHICH STARTS AT  *
C*        THE FIRST CELL.  IF A FALSE TARGET IS PRESENT THE EVENT IS   *
C*        REORDERED TO CONFORM TO THE MSM FORMAT AND THE FALSE TARGET  *
C*        INFORMATION IS ADDED. (THE MINUS SIGN IS DELETED FROM THE    *
C*        FALSE TARGET ID AND WORDS 2-12 OF THE TARGET INFORMATION     *
C*        IN TARG ARE ADDED BY A CALL TO TRNSFR)                       *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-79

```
C*****************************  SEQ   *********************************
C*                                                                     *
C*                           SUBROUTINE SEQ                            *
C*                                                                     *
C*    PURPOSE                                                          *
C*        TO GENERATE AN ORDERED LIST OF TYPE 1 EVENTS FROM THE        *
C*        ELPOT LIST OF DETECTIONS FOR A GIVEN SENSOR                  *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL SEQ(IT,ID,LE,E,LF,F,MMAX,LL,LV,IP,KV,ITARG)             *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*                                                                     *
C*         * INPUT *                                                   *
C*        IT   EVENT TYPE                                              *
C*        ID   ID OF SENSOR                                            *
C*        LE   ARRAY DEFINING ORDER OF EARLIEST DETECTION TIMES        *
C*        E  EARLIEST TIMES                                            *
C*        LF   ARRAY DEFINING ORDER OF LATEST TIMES                    *
C*        F    LATEST TIMES                                            *
C*        MMAX NUMBER OF EVENTS                                        *
C*        ITARG TARGET ID'S                                            *
C*                                                                     *
C*         * OUTPUT *                                                  *
C*        LL STORAGE FOR EVENTS AS GENERATED.                          *
C*        LV COUNT OF WORDS IN LL.                                     *
C*        IP POINTER LOCATING EVENTS                                   *
C*        KV COUNT OF WORDS IN POINTER                                 *
C*                                                                     *
C*    REMARKS                                                          *
C*        NONE                                                         *
C*                                                                     *
C*    SUBROUTINES REQUIRED                                             *
C*        NONE                                                         *
C*                                                                     *
C*    METHOD                                                           *
C*        THE ARRAYS DEFINING THE ORDER OF TIMES ARE SCANNED AND THE   *
C*        EVENTS ARE GENERATED IN AN ORDERED FORM IN COMPLIANCE WITH   *
C*        THE MSM FORMAT FOR EVENT TYPE 1. THIS ROUTINE DOES NOT       *
C*        DISTINGUISH BETWEEN REAL AND FALSE TARGETS.                  *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-80

```
C****************************** PREMNB ********************************
C*                                                                    *
C*                  PRERUN EXECUTIVE -     STEP 11                    *
C*                                                                    *
C*                          SUBROUTINE PREMNB                         *
C*                                                                    *
C*     PURPOSE                                                        *
C*         FINAL MERGE OF ALL MSM EVENTS CREATED BY PRERUN            *
C*                                                                    *
C*     USAGE                                                          *
C*         MAIN PROGRAM                                               *
C*                                                                    *
C*     DESCRIPTION OF PARAMETERS                                      *
C*         ALL INPUT AND OUTPUT VIA COMMON BLOCKS/ WEVT,UTMCOM,TIMES, *
C*         INOUT,OUTP,OPTION,POSERR/                                  *
C*                                                                    *
C*     REMARKS                                                        *
C*         NONE                                                       *
C*                                                                    *
C*     SUBROUTINES REQUIRED                                           *
C*         FMERGE                                                     *
C*                                                                    *
C*     METHOD                                                         *
C*             THE MSM EVENTS GENERATED BY PREVIOUS STEPS OF PRERUN   *
C*         ARE READ FROM THE VARIOUS DATA SETS AND WRITTEN AS A       *
C*         SEQUENCE OF RECORDS ON A SINGLE UNIT MTAPE(12)             *
C*            FMERGE IS CALLED TO MERGE THESE EVENTS AND WRITE THEM   *
C*         ON UNIT MTAPE(14). MAXIMUM STORAGE IS DEFINED BY MAX. AND  *
C*         THE LENGTH OF RECORDS BY MAX1. A ONE & ZERO IS WRITTEN     *
C*         TO INDICATE END OF FILE.                                   *
C*            THE PRINT OPTION IS CHECKED AND IF DESIRED THE RECORDS  *
C*         WILL BE PRINTED.                                           *
C*                                                                    *
C**********************************************************************
```

Figure 2.1-81

```
C****************************** FMERGE *********************************
C*                                                                     *
C*    PURPOSE                                                          *
C*        MERGES EVENT ARRAY IEV WITH EVENT ARRAY IVE .                *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL FMERGE(IR1,IR2,IR3,IEV,IVE,MAX,MAX1)                    *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*          * INPUT *                                                  *
C*        IR1        DISK OR TAPE UNIT TO BE READ                      *
C*        IR2        DISK OR TAPE ON WHICH ARRAY IEV IS WRITTEN        *
C*        IR3        DISK OR TAPE ON WHICH ARRAY IVE IS WRITTEN        *
C*        IVE        ARRAY TO BE MERGED                                *
C*        IEV        ARRAY TO BE MERGED                                *
C*        MAX        THE MAXIMUM ALLOWABLE NO.OF ITEMS FOR THE         *
C*                   RESULTING MERGED ARRAY(IEV).                      *
C*        MAX1       MAX. ALLOWABLE NO.OF ITEMS IN ARRAY IVE.          *
C*                                                                     *
C*    REMARKS                                                          *
C*        NONE                                                         *
C*                                                                     *
C*    SUBROUTINES REQUIRED                                             *
C*        GMERGE                                                       *
C*                                                                     *
C*    METHOD                                                           *
C*            THE RECORDS ON IR1 ARE READ AND MERGED BY GMERGE.        *
C*        STORAGE IS CONTINUALLY CHECKED (MAX1) THE OVERFLOW IS        *
C*        WRITTEN ON IR2. AFTER A COMPLETE PASS THROUGH IR1 THE        *
C*        ARRAY IVE WILL CONTAIN THE BLOCK OF EVENTS WITH THE SMALLEST *
C*        TIMES. THIS IS WRITTEN ON IR3. THE ROLES OF IR1 AND IR2      *
C*        ARE INTERCHANGED AND THE PROCESS CONTINUED UNTIL ALL EVENTS  *
C*        HAVE BEEN PROCESSED AND WRITTEN ON IR3.                      *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-82

```
C***************************** PREMNC ***********************************
C*                                                                      *
C*                          MAIN PROGRAM                                *
C*                                                                      *
C*    PURPOSE                                                           *
C*       TO SUBDIVIDE EVENT LIST INTO GROUPS LESS THAN OR EQUAL TO 900*
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*    .  MAIN PROGRAM                                                   *
C*       NONE                                                           *
C*    REMARKS                                                           *
C*       EVENTS CANNOT NECESSARILY BE SPLIT UP INTO SETS OF EXACTLY 900, *
C*       THEREFORE GROUPS FOR MSM CAN BE LESS THAN 900.                 *
C*                                                                      *
C*    SUBROUTINES AND FUNCTION ROUTINES REQUIRED.                       *
C*                                                                      *
C*                                                                      *
C***********************************************************************
```

Figure 2.1-83

```
C******************************* FINDX  *******************************
C*                                                                    *
C*                           SUBROUTINE FINDX                         *
C*                                                                    *
C*    PURPOSE                                                         *
C*        THIS IS A UTILITY ROUTINE USED TO LOCATE A PARTICULAR       *
C*        DATA SET IN THE MASTER DATA STREAM                          *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL FINDX(N,I,J,K)                                         *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*                                                                    *
C*          * INPUT *                                                 *
C*       N       ID OF DATA SET                                       *
C*                                                                    *
C*          * OUTPUT *                                                *
C*       I       LOCATION OF FIRST WORD IN MASTER STREAM              *
C*       J       LOCATION OF LAST WORD IN DATA STREAM                 *
C*       K       NUMBER OF WORDS IN SUB SET                           *
C*                                                                    *
C*    REMARKS                                                         *
C*        DO LOOP       DO NN  L=I,J,K  WILL SCAN ALL SUB SETS OF N   *
C*                                                                    *
C*                                                                    *
C*    SUBROUTINES REQUIRED                                            *
C*        NONE                                                        *
C*                                                                    *
C**********************************************************************
```

Figure 2.1-84

```
C******************************* FINDY  ********************************
C*                                                                     *
C*                        SUBROUTINE FINDY                             *
C*                                                                     *
C*    PURPOSE                                                          *
C*        THIS IS A  UTILITY ROUTINE WHICH IS USED TO FIND THE         *
C*        POSITION OF A DATA SET IN A STRING FROM THE POINTER INFO     *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL FINDY(L,M,I,J,K)                                        *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*            * INPUT *                                                *
C*        L( ) POINTER                                                 *
C*        M     MAXIMUM DIMENSION OF POINTER                           *
C*        I     INDEX OF DEVICE WHOSE TIMES ARE TO BE LOCATED          *
C*                                                                     *
C*           * OUTPUT *                                                *
C*        J     LOCATION OF FIRST TIME (=0 IF NEVER ON)                *
C*        K     LOCATION OF LAST TIME                                  *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-85

```
C******************************* MERGOR ********************************
C*                                                                      *
C*                          SUBROUTINE MERGOR                           *
C*                                                                      *
C*     PURPOSE                                                          *
*         THIS ROUTINE TAKES A SET OF FIXED LENGTH MSM EVENTS,          *
C*          ORDERS THEM AND MERGES THEM WITH THE MASTER STRING IEV      *
C*                                                                      *
C*     CALLING SEQUENCE                                                 *
C*          CALL MERGOR(MEV,IEV,IVE,MVE,LL,MR,II,MAX)                   *
C*                                                                      *
C*     DESCRIPTION OF PARAMETERS                                        *
C*          MEV     NUMBER OF POINTS IN MAIN TABLE                      *
C*          IEV     MASTER TABLE                                        *
C*          IVE     TABLE TO BE ORDERED AND MERGED                      *
C*          MVE     NUMBER OF POINTS IN IVE                             *
C*          LL      WORKING STORAGE                                     *
C*          MR      NUMBER OF EVENTS TO BE ORDERED AND MERGED           *
C*          II      LENGTH OF EVENT                                     *
C*          MAX     MAXIMUM DIMENSION OF STORAGE ALLOCATED FOR EVENTS   *
C*                                                                      *
C*     METHOD                                                           *
C*              THE ORDER OF THE EVENTS IS DETERMINED BY A CALL TO DORDER.*
C*              THE ORDERED LIST IS PLACED 'ON TOP OF' THE MASTER LIST  *
C*          AND MERGED BY A CALL TO GMERGE.                             *
C*                                                                      *
C*     SUBROUTINES REQUIRED                                             *
C*          DORDER                                                      *
C*          GMERGE                                                      *
C*                                                                      *
C************************************************************************
```

Figure 2.1-86

```
C******************************** DORDER *********************************
C*                                                                       *
C*                         SUBROUTINE DORDER                             *
C*                                                                       *
C*    PURPOSE                                                            *
C*          THIS ROUTINE DETERMINES THE ORDER OF A SET OF FIXED          *
C*          LENGTH EVENTS WHERE THE VARIABLE IN THE FIRST POSITION       *
C*          OF EACH SET OF LENGTH I IS USED TO DETERMINE THE ORDER       *
C*                                                                       *
C*    CALLING SEQUENCE                                                   *
C*       CALL DORDER(IE,L,N,I)                                           *
C*                                                                       *
C*    DESCRIPTION OF PARAMETERS                                          *
C*       IE     ARRAY TO BE ORDERED ON FIRST WORD                        *
C*       L      DEFINES ORDER                                            *
C*       N      NUMBER OF EVENTS TO BE ORDERED                           *
C*       I      LENGTH OF EACH EVENT                                     *
C*                                                                       *
C*    REMARKS                                                            *
C*       L(N) WILL GIVE THE LOCATION OF THE N'TH EVENT IN THE            *
C*       ORDERED LIST.                                                   *
C*       FOR EXAMPLE: IF M EVENTS ARE TO BE ORDERED, L(1)=K WOULD        *
C*       INDICATE THAT THE KTH WORD IN THE ORIGINAL LIST IS THE          *
C*       BEGINNING OF THE FIRST EVENT.                                   *
C*                                                                       *
C*    SUBROUTINES REQUIRED                                               *
C*       NONE                                                            *
C*                                                                       *
C*************************************************************************
```

Figure 2.1-87

```
C******************************** GMERGE ********************************
C*                                                                      *
C*                          SUBROUTINE GMERGE                           *
C*                                                                      *
C*    PURPOSE                                                           *
C*       THIS IS  A UTILITY ROUTINE USED TO MERGE TWO ORDERED           *
C*       EVENT LISTS                                                    *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*       CALL GMERGE(MEV,N,NVS,MAX)                                     *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*       MEV     LIST TO BE MERGED                                      *
C*       N       NUMBER IN MASTER LIST                                  *
C*       NVS     TOTAL NUMBER IN  LIST                                  *
C*       MAX     MAXIMUM  DIMENSION ALLOCATED FOR MEV                   *
C*                                                                      *
C*    REMARKS                                                           *
C*       PRIORITY IS GIVEN TO POINT TYPE EVENTS                         *
C*       FOR EVENTS TYPE1 WITH EQUAL UP TIMES PRIORITY IS GIVEN         *
C*       TO THE ONE WITH THE SMALLEST DOWN TIME                         *
C*       OTHERWISE PRIORITY IS GIVEN TO EVENTS ALREADY IN THE           *
C*       MASTER LIST                                                    *
C*                                                                      *
C*    SUBROUTINES REQUIRED                                              *
C*       NONE                                                           *
C*                                                                      *
C*    METHOD                                                            *
C*       THE MASTER LIST IS IN WORDS 1-N OF MEV. THE LIST TO BE         *
C*       MERGED IS IN WORDS  N+1 TO NVS. THE ENTIRE LIST IS SHOVED      *
C*       TO THE END OF THE ARRAY AS SPECIFIED BY MAX AND THEN THE       *
C*       MERGE IS BEGUN.  AT THE END THE MERGED LIST EXTENDS FROM       *
C*       1 TO NVS.                                                      *
C*                                                                      *
C************************************************************************
```

Figure 2.1-88

```
C*********************************** MERGE1 ****************************
C*                                                                    *
C*                       SUBROUTINE MERGE1                            *
C*                                                                    *
C*    PURPOSE                                                         *
C*        SPECIAL MERGE ROUTINE, THAT WILL MERGE TWO EXISTING         *
C*        LISTS OF 'EVENTS', INDIVIDUALLY TIME ORDERED, INTO A        *
C*        NEW COMBINED LIST THAT IS TIME ORDERED.                     *
C*                                                                    *
C*        THE 'EVENTS' CORRESPOND TO VARIABLE LENGTH SUBLISTS,        *
C*        WITH                                                        *
C*              (A) SECOND WORD IN EACH SUBLIST GIVES LENGTH          *
C*                  (IN WORDS) OF THAT SUBLIST.                       *
C*                                                                    *
C*              (B) FOURTH WORD IN EACH SUBLIST GIVES INTEGER         *
C*                  TIME VALUE, ON WHICH ORDERING IS BASED.           *
C*                                                                    *
C*                                                                    *
C*    USAGE                                                           *
C*        CALL MERGE1 (LISTA,LISTB,NEVTSA,NEVTSB, LISTC,NEVTSC,NWROSC) *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*                                                                    *
C*             INPUT                                                  *
C*        LISTA, LISTB        TWO ARRAYS OF EVENTS, TO BE MERGED      *
C*        NEVTSA, NEVTSB      NUMBER OF EVENTS (=NUMBER OF SUBLISTS)  *
C*                                IN LISTA, LISTB RESPECTIVELY        *
C*                                                                    *
C*               OUTPUT                                               *
C*        LISTC               OVERALL (MERGED) LIST FORMED FROM       *
C*                                LISTA AND LISTB                     *
C*        NEVTSC              NUMBER OF EVENTS IN LISTC               *
C*        NWRDSC              NUMBER OF WORDS IN LISTC                *
C*                                                                    *
C*                                                                    *
C*    REMARKS                                                         *
C*        EVENTS CAN CORRESPOND EITHER TO A SINGLE INSTANT OF TIME OR TO A TIME *
C*        INTERVAL.                                                   *
C*                                                                    *
C*        THIS MERGE PROGRAM IS INTENDED FOR 'SINGLE INSTANT OF       *
C*        TIME' EVENTS.  IT WILL ALSO ACCOMMODATE 'INTERVAL'          *
C*        EVENTS IF NO OVERLAPS OF TIME OCCUR.                        *
C*                                                                    *
C*                                                                    *
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*        SUBROUTINE TRNSFR                                           *
C*                                                                    *
C*                                                                    *
C**********************************************************************
```

Figure 2.1-89

```
C******************************  TRAN  ******************************
C*                                                                  *
C*                        SUBROUTINE TRAN                           *
C*                                                                  *
C*    PURPOSE                                                       *
C*        FUNCTION ROUTINE TO TRANSFER A VARIABLE                   *
C*                                                                  *
C*    USAGE                                                         *
C*        Y = TRAN(X)                                               *
C*                                                                  *
C*    DESCRIPTION OF PARAMETERS                                     *
C*        * INPUT *                                                 *
C*        X     INPUT VARIABLE TO BE TRANSFERRED                    *
C*        Y     OUTPUT VARIABLE                                     *
C*                                                                  *
C********************************************************************
```

Figure 2.1-90

```
C****************************** TRAN2  *** *****************************
C*                                                                     *
C*                          SUBROUTINE TRAN2                           *
C*                                                                     *
C*    PURPOSE                                                          *
C*        TRANSFERS A BLOCK OF STORAGE                                 *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL TRAN2(A,B,N)                                            *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        * INPUT *                                                    *
C*        N     NUMBER OF ITEMS TRANSFERRED                            *
C*        B     ARRAY OF ITEMS TO BE TRANSFERRED                       *
C*        * OUTPUT *                                                   *
C*        A     ARRAY OF ITEMS TRANSFERRED FROM ARRAY B                *
C*                                                                     *
C***********************************************************************
```

Figure 2.1-91

Table 2.1-VI

PRERUN COMMON AREAS

COMMON AREA IDENTIFICATION

USE	VARIABLES	USED BY	WHERE USED
1 COMMON/ATTIME/IT (1200), NTABLE			
	IT Array of Time Intervals for False Alarm Events NTABLE Number of These Time Intervals	Used Exclusively by EVNT23	PRERUN 7
2 COMMON/ATMENV/ITEFF, SOLALT, ALTLUN, PHSLUN, IPCODE, PRATE, PTOT24, H2ODEN, WSPEED, CCOVER, ATEMP, PRESUR, HUMDTY, VISIB, CEIL, ASID(3), TCLOUD			
Atmospheric Variables	ITEFF - Effective time SOLALT - Solar altitude ALTLUN - Lunar altitude PHSLUN - Lunar phase IPCODE - Precipitation code PRATE - Precipitation rate PTOT24 - Precipitation last 24 hrs. H2ODEN - Water in air WSPEED - Wind speed CCOVER - Cloud cover ATEMP - Air temperature PRESUR - Air pressure HUMDTY - Relative humidity VISIB - Visibility CEIL - Ceiling ASID - Spectral irradiance TCLOUD - Cloud transmission	ACOUBK ARFBK BWIRBK ENVIR EVNT23 PIRBK SEISBK	PRERUN 7
3 COMMON/ATMSP/N, A (4, 200) COMMON/ATMSP/NMSP, ATMEN (4, 200)			
Storage of atmospheric data	N, NMSP, number of sets in table (Second subscript) A, ATMEN (1, N) TIME, seconds (2, N) VISIBILITY, meters (3, N) CEILING, meters (4, N) DAY/NIGHT, 0/1	PREMN4, MVS	PRERUN 4
4 COMMON/BASICT/ITIME, ITOD, ITODST, ITDURN, IDATE, IDAREA			
Storage of basic game times	ITIME - Current time ITOD - Time of day ITODST - Time of day at game start ITDURN - Duration of game IDATE - Game start date IDAREA - Scenario area identifier	ACOUBK ARFBK BWIRBK ENVIR EVENT23 PIRBK SEISBK TWRLKD	PRERUN 7

Table 2.1-VI (Cont.)
PRERUN COMMON AREAS

USE	VARIABLES	USED BY	WHERE USED
5 COMMON/BATTEL/SCHDL(14), SAFETY(15, 4), ZNOMAS (20, 4), FBWPN (4, 13), RNGBN (12, 1), WPFLT (8, 13)			
Used in Battle routines to store Battle schedule and designer tables Designer input values in JFBLK6	SCHDL - Basic Battle schedule - Planner input-battle output SAFETY - Safety margin set - Designer input ZNOMAS - Nominal affect of background on sensors - Designer input FBWPN - Fire base weapons - Designer input RNGBN - Range bin limits - Designer input WPFLT - Weapon projectile flight times - Designer input	BATEX BATLAK BSCHDL BEMAP JFBLK6	PRERUN 6
6 COMMON/BB/ISAFTY, IPIEVT, IWPFLT, JWRLIM, IWRLIM, IACSPD, JACSPD, IACALT, JACALT, IVSPED, JVSPED, ICNVOY, ISPACE, IFCWPN, IRSEVT, INMSS, NDAYS			
Used to store dimensions of tables in Battle routines	ISAFTY - Index limit of table SAFTY (I, J), I I ISAFTY IPIEVT - Index limit of table PIEVT (I, J), I I IPIEVT IWPFLT - Index limit of table WPFLT (I, J), I I IWPFLT JWRLIM - Index limit of table WRLIM (I, J), I J JWRLIM IWRLIM - Index limit of table WRLIM (I, J), I I IWRLIM IACSPD - Index limit of table ACSPD (I, J), I I IACSPD JACSPD - Index limit of table ACSPD (I, J), I J JACSPD IACALT - Index limit of table ACALT (I, J), I I IACALT JACALT - Index limit of table ACALT (I, J), I J JACALT IVSPED - Index limit of table VSPED (I, J), I I IVSPED JVSPED - Index limit of table VSPED (I, J), I J JVSPED ICNVOY - Index limit of table CNVØY (I, J), I I ICNVØY ISPACE - Index limit of table SPACE (I, J), I I ISPACE IFCWPN - Index limit of table FCWPN (I, J, K), I I IFCWPN IRSEVT - Index limit of table RSEVT (I, J, K), I I IRSEVT INMSS - Index limit of table ZNOMAS(I, J), I I INMSS NDAYS - Number of days game is played	BATEX BATLBK BEMAP BSCHDL JFBLK6	PRERUN 6
7 COMMON/BBNDS/XMIN, YMIN, XMAX, YMAX			
Used to store games area coordinates as needed in Battle	XMIN = 0 YMIN = 0 YMAX = X dimension of play area YMAX = Y dimension of play area	BATEX BATLBK BEMAP BSCHDL PREMN6	PRERUN Step 6
8 COMMON/CONST/LDUMP, SQ2, DEG, RAD, PI, STEFK, ICARD, IPRINT			
Used to store constants in PRERUN Step 7	LDUMP = Dump or print option SQ2 = Square root of 2 DEG = 3600 in Radians RAD = 1^{o} in Radians PI = 180^{o} in Radians STEFK = Stefan Boltzmann Constant/PI ICARD, IPRINT = See #16	EVNT23 BWIRBK SEISBK ACOUBK ENVIR TNBLKD ACKBK PIRBK	PRERUN Step 7

Table 2.1-VI (Cont.)

PRERUN COMMON AREAS

	USE	VARIABLES	USED BY	WHERE USED
9	COMMON/CONST/			
	Used to store constants in PRERUN Step 9	LDUMP = Dump or print option SQ2 = Square root of 2 DEG = 3600 in Radians RAD = 1^{o} in Radians PI = 180^{o} in Radians STBPK = Stefan Boltzmann Constant/PI ICARD, IPRINT = See #16	BLKLOS MAINLS	PRERUN Step 9
10	COMMON/DATAIN/NDATA, NSETS, LDATA(40), NDATC(40), IDAT^ (12000) -or- COMMON/DATAIN/NNN, MMM, LD(40), NENT(40), ID (12000)			
	Used to store planner input data	NDATA - Number of words in IDATA NSETS - Number of data sets LDATA - Pointer for data set NDATC - Length of subset IDATA - Master data stream NNN - Number of words in IDATA MMM - Number of data sets LD(40) - Pointer for data set NENT(40) Length of Subset ID(12000)Master data stream	PREMN1 UPDN1 PREMN2 UPDN3 PREMN3 UPDN5 PREMN4 UPDN6 PREMN6 UPDN8 PREMN? UPDN10 IXMAIN* UPDN19 TRNPAR* FINDX EVNT23* READIN* PSNP ELPEX PSNP19 SENSQ MVS SENXY TRNPRI* TARGBR CULTBK CULTEX BATEX ELPDT	PRERUN 1-8, Step 9
11	COMMON/EVENTS/MEV, IEV (10000) IVE (5000), MVE			
	Used to store events as generated and master event list	MEV - Number of words in IEV IEV - Master event list IVE - Temporary event list MVE - Count of words in IVE This list is initiated by Step 1 and transmitted to other steps by Internal data set MTAPE (4), JFEVT. Step 2 adds to the list and writes a second record of two words 1, 0. Step 3 adds to the initial list and writes a second record of two words 1, 0. Step 4 adds to the initial list and writes a second record of two words 1, 0. The storage requirements for IEV may be reduced and the capacity of the program greatly increased by modifying the executive routines (PREMN) to read in the records on JFEVT until the last record 1, 0 is read, backspacing and writing the event generated by each sub-executive routine as a separate record. After writing the record, the master count MEV should be set to zero. No other program changes are required.	PREMN1 UPDN8 PREMN2 UPDN10 PREMN3 UPDN19 PREMN4 PSNP UPDN1 PSNP19 UPDN3 BATEX** UPDN5 EVNTS** UPDN6 MVS	PRERUN 1-4, 6

*These routines use the first form; all the rest use the second. Since this contains all of the planner input data, it is used by most of the major executive and sub-executive routines in PRERUN. Data sets are located in this stream by calls to subroutine FINDX.

**These two routines in the Battle program (PREMN6) write the event list generated on the internal data set MTAPE(9). The sub-executive routines insure that the master list is ordered and step 11 - PREMN8 - is designed to make a final merge of any number of records that are on the data sets.

Table 2.1-VI (Cont.)
PRERUN COMMON AREAS

	USE	VARIABLES	USED BY	WHERE USED
12	COMMON/EXCLUD/IXCLUA, JXCLUA, XCLUA (20, 15)			
	Used by Battle Routines	IXCLUA - Maximum value of first subscript JXCLUA - Maximum value of record subscript XCLUA - Exclusion area set X, Y boundaries	PREMN6 BSCHDL BATEX BEMAP BATLBK JFBLK6	PRERUN Step 6
13	COMMON/FSBASE/JFSPTB, FSPTB (9, 6)			
	Used by Battle routines	JFSPTB - Maximum value of second subscript FSPTB - X, Y coordinates of Fire Support Base and EVIDS - planner input 26	BATEX BSCHLD PREMN6	PRERUN Step [illegible]
14	COMMON/GAME/A(510)			
	Used exclusively by step 0 INMAIN to store header cards	A(510) - Planner input scenario header cards	INMAIN	PRERUN Step Ø
15	COMMON/JTPWDF/LOCATE(14), IPRY (12000)			
	Used exclusively by PRERUN Step 5	LOCATE - Data set counter IPRY - Planner input parameters	TRNPAR TRNPR1	PRERUN Step 5
16	COMMON/INOUT/IPRINT, ICARD			
	Used to define printer and card reader in programs using BCD output or card input	IPRINT - Printer ICARD - Card reader	INMAIN PSNP PREMN1 PSNP19 PREMN2 MVS PREMN3 BATEX PREMN4 BEMAP PREMN6 BSCHDL PREMN8 CSCHDL PREMN9 CULTBK PREMNA CULTEX PREMNB ELPEX UPDN3 TARGMS UPDN6 UPDN8 UPDN19	PRERUN Step Ø - 4, 6, 8 - 11
17	COMMON/LOSCOM/ITRNHT(30601), XREF, YREF, XRANGE, YRANGE, XMIN, YMIN, DGD, MXREF, MYREF, MXTENT, MYTENT			
	Used by line of sight routines	ITRNHT - Array containing terrain elevations (Integerized) XREF - X origin of playing field. (Normally zero for compatibility with Unter tables) YREF - Same as above (Y coordinate) XRANGE - Maximum X length of playing field YRANGE - Same as above (Y length) XMIN - Minimum X length of playing field YMIN - Same as above (Y length) DGD - Grid size of map (100.) MXREF - Integerized X reference of playing field MYREF - Same as above (Y reference) MXTENT - Integerized X range of playing field MYTENT - Same as above (Y range)	MAINLS LOS BLKLOS TERANE	PRERUN Step 9

Table 2.1-V1 (Cont.)

PRERUN COMMON AREAS

USE	VARIABLES	USED BY	WHERE USED
18 COMMON/NVSNP/ADRP1 (10, 2,2), ADRP2 (10, 3, 2), PRNV1 (4, 10), PRNV2 (4, 7) PRNV3 (4, 7), PRNV4 (4), ATMOE (34, 5, 2)			
To store designer input and planner input data Designer input values in JFBLK3	ADRP1 - Designer input ADRP2 - Designer input PRNV1 - Planner input 15 PRNV2 - Planner input 16 PRNV3 - Planner input 17 PRNV4 - Planner input 18 ATMOE - Designer input	JFBLK3 PSNP PSNP19 SNPGT	PRERUN Step 3
19 COMMON/NV/NVES, ZEVS (22,700), NEV (4,200), NVE			
To store early-late detection times for each sensor	NVES - Second index of ZEVS ZEVS - Information required by line of sight routine NEV - Detections not requiring line of sight NVE - Second index of NEV For definition of ZEVS, see subroutine ELPDT NEV(1,) Sensor ID NEV(2,) Target ID NEV(3,) Earliest possible detection time NEV(4,) Latest possible detection time As each sensor is processed, the results are written on internal data sets. NEV Internal data set MTAPE (15) - JFNEV JEVS Internal data set MTAPE (16) - JFZEV After all sensors have been processed, ELPEX writes a 1 followed by 4 zeros on JFNEV, and a 1 followed by 22 zeros on JFZEV.	PREMN8 ELPEX ELPDT	PRERUN Step 8
20 COMMON/OPTION/NPLAY (2)			
To provide an option to play planned up-down times	NPLAY - 0 play planned up-down times - 1 compute up-down times This is recorded on data set MTAPE (2) - JFMSTR	PREMN1 - PREMN8 PREMN2 PREMN9 PREMN3 PREMNA PREMN4 PREMNB PREMN6	PRERUN 1-4, 6, 8-11
21 COMMON/OUTP/MPRINT(30), MTAPE (30)			
Used to store print options and data set numbers	MPRINT - Print option MTAPE - Data set number See comments for INMAIN This is recorded on data set MTAPE (2) - JFMSTR	PREMN1 BATEX PREMN2 BATLTG PREMN3 CULTEX PREMN4 CULTBK PREMN6 ELPEX PREMN8 PREMN9 PREMNA PREMNB	PRERUN 1-4, 6, 8-11

Table 2.1-VI (Cont.)

PRERUN COMMON AREAS

USE	VARIABLES	USED BY	WHERE USED
COMMON/PAR/IRCEVT, IPCEVT, ISCHAR, IPSPED, ICACAL, ICEVD, IANSPD, NLMX			
To store indices	IRCEVT - Index limit of table RCEVT (I, J, K), 1 I IRCEVT IPCEVT - Index limit of table PCEVT (I, J), 1 I IPCEVT ISCHAR - Index limit of table SCHAR (I, J), 1 I ISCHAR IPSPED - Index limit of table PSPED (I, J), 1 I IPSPED ICACAL - Index limit of table CACAL (I, J), 1 I ICACAL ICEVD - Index limit of table CEVDBA (I, J), 1 I ICEVD IANSPD - Index limit of table ANSPD (I, J), 1 I IANSPD NLMX = IRØUTX - INDEX (IMIT TABLE ROUTX (I, J)) 1 I IRØUTX	CULTEX CULTBK CSCHDL JFBLK6	PRERUN Step 6
COMMON/PGMPAR/SITYM, SETYM, TITYM, TETYM, RMIN, RMAX, DX DY, DT, DX1, DY1, DXA, DYA, DXB, DYB, ACB, DA, DB, DC, L, EPDT(3), FLPDT(3), CVANGL, TA, TB, VS			
To communicate to the geometry routines	SITYM - Sensor time 1 SETYM - Sensor time 2 TITYM - Target time 1 TETYM - Target time 2 RMIN - Minimum radius RMAX - Maximum radius DX - X component of target path DY - Y component of target path DT - Total target time interval DX1 - Initial X dist. of sensor from target DY1 - Initial Y dist. of sensor from target DXA - COS (PI/2 - azimuth angle - coverage angle/2) DYA - SIN (PI/2 - azimuth angle - coverage angle/2) DXB - COS (PI/2 - azimuth angle + coverage angle/2) DYB - SIN (PI/2 - azimuth angle + coverage angle/2) ACB - DXA*DYB - DXB*DYA DA - Cross track distance minus 0.5 width of rectangle DB - DA + rectangle width DC - Along track distance minus 0.5 length of rectangle L - Number of segments of intersection, = 0 IF none EPDT - Earliest possible detection times FLPDT - Latest possible detection times CVANGL - Coverage angle TA - Maximum (SITYM, TITYM) TB - Minimum (SETYM, TETYM) VS - Velocity of sensor	CIRC GREC SECT ELPDT ELPEX	PRERUN Step 8
COMMON/POSERR/ZMAP, XLOC, RELOC, ANAV, ARTY, AIRD			
Used to record ground truth position error parameters	ZMAP - Standard deviation of map error (meters) XLOC - 1 - Play location errors - Ø - Don't play location errors RELOC - 1 - Play relocation errors - Ø - Don't play relocation errors ANAV - 1 - Play navigation errors - Ø - Don't play navigation errors ARTY - 1 - Play artillery/mortar errors - Ø - Don't play artillery/mortar errors AIRD - 1 - Play vertical fall errors - Ø - Don't play vertical fall errors See comments for INMAIN This is recorded on MTAPE(2), JFMSTR	INMAIN PREMN9 PREMN1 PREMNA PREMN2 PREMNB PREMN3 PSNP PREMN4 PSNP19 PREMN6 MVS PREMN8	PRERUN Ø - 4, 6, 8 - 11

Table 2.1-VI (Cont.)
PRERUN COMMON AREAS

USE	VARIABLES	USED BY	WHERE USED
25 COMMON/PXYTP/NT, KXYT(401), SXYT(5000)			
Used to record ground truth positions for the sensors	See Item 26	PREMN4, PREMN8, ELPEX, TRNPRI	PRERUN 4, 5, 8
26 COMMON/STASEN/NXY, KSSN(401), SXYTT(5000) (IDENTICAL TO PREVIOUS SET)			
Used to record ground truth	NT, NXY Number of words in SXTY KXYT, KSSN Pointer for SXTY SXYT, SXYTT Space time ground truth for sensors The name STASEN is used by the stationary sensor routine and the data is stored in the format DX, DY, X_1, Y_1, T_1, T_2, X_2, Y_2, T_3, T_4, etc. DX Differential X of path (=0 if location is not relative to a path) DY Differential Y of path (=0 if location is not relative to a path) X_1 X coordinate Y_1 Y coordinate T_1, T_2 Beginning and end of time interval for position X_1, Y_1 If the sensor is relocated, the position and times are stored sequentially using the format X Y T T as often as necessary. The name PXYTP is used for the moving sensors where the space time position of the nodes of the path defined by straightline segments are stored as X Y T X Y T X Y T, etc. The coordinates for a particular sensor may be found by using the FINDY subroutine This is recorded on internal data set MTAPE(5) - JFPXY.	PREMN3, PSNP, PSNP 19	PRERUN 3
27 COMMON/ROUTE/IROUTX, IROUTY, ROUTX (29, 12) ROUTY (20, 12)			
To store path information for Battle and Culture	IROUTX = IROUTY Number of paths ROUTX X coordinate of Nodes and path type ROUTY Y coordinate of Nodes	PREMN6 BATEX CULTEX BATLBK CULTBK BSCHDL CSCHDL REMAP	PRERUN Step 6

Table 2.1-VI (Cont.)

PRERUN COMMON AREAS

USE	VARIABLES	USED BY	WHERE USED
28 COMMON/SNDXDY/JSNFDX, SNFDX (14, 4), JSNFDY, SNFDY (14, 4)			
CULTUR2	JSNFDX - Maximum second index of SNFDX JSNFDY - Maximum second index of SNFDY SNFDX - Planner input 29 SNFDY - Planner input 29	PRIMM6 CULTEX CSCHDL	PRERUN Step 6
29 COMMON/SENVAR/CONSTA, TDEL2A, BIASAC, BWACOU, CONSTS, TDEL2S, BIASSE, BWSEIS, PHIAZ, PHIEL, DIAM, BWPIR, DEVXMN, XMNDEV, BANDTH, ANEP, OPTXMN, THRESH, DELAZ, TIMMAX			
Used exclusively by PRERUN Step 7	CONSTA - Average amplifier output (acoustic sensor) TDEL2A - Time delay times 2 for acoustic sensor BIASAC - Threshold setting in volts (seismic/acoustic) BWACOU - Band width for acoustic sensor CONSTS - Average amplifier output (seismic sensor) TDEL2S - Same as TDEL2A (seismic sensor) BIASSE - Threshold setting seismic sensor (Volts) BWSEIS - Effective band width of noise signal for seismic sensor in hertz. PHIAZ - Azimuth angle of sensor in radians PHIEL - Elevation angle of sensor in radians DIAM - Diameter of sensor (mm) BWPIR - Band width for passive infrared sensor (Hz./sec) DEVXMN - Optical system transmission factor XMNDEV - Optical system transmission factor (thermal) BANDTH - Sensor sensor bandwidth (thermal) ANEP - Noise equivalent power of detector (thermal) OPTXMN - Optical system transmission factor (.8) (image device) THRESH - Threshold value for sensor DELAZ - Angle between center line of two beams of passive infrared sensor TIMMAX - Maximum time allowed for detection (PIRID)	ACOUBK PIRBF SEISBK TWBLKD	PRERUN Step 7
30 COMMON/SUBCUL/PSPED (15, 4), CACAL (5, 4), SCHAR (20, 12), CEVDBA (20, 4) ANSPD (4, 4)			
To store designer inputs for Culture	PSPED - Path speed CACAL - Cultural Aircraft altitude SCHAR - Source character CEVDBA - CEVID Signal Strength ANSPD - Animal speed	CULTEX CSCHDL JFBLK6	PRERUN Step 6
31 COMMON/TIMES/TSTART, TMAX			
Used to store beginning (TSTART) and maximum length (TMAX) of game times	Recorded on data set MTAPE(2) - JFMSTR TSTART - Time of game start (seconds) TMAX - Time of game end (seconds)	INMAIN MVS PREMN1 UPDN1 PREMN2 UPDN3 PREMN3 UPDN10 PREMN4 UPDN19 PREMN6 PSNP PREMN8 PSNP19 PREMN9 ELPEX PREMNA ELPDT PREMNB TARGBR TIMER	PRERUN 0-4, 6, 8-11

Table 2.1-VI (Cont.)

PRERUN COMMON AREAS

USE	VARIABLES	USED BY	WHERE USED
32 COMMON/TARG/NTR, TRGTS (13, 200/			
Used as temporary store of target parameters in Battle and Culture	NTR - Number of targets TRGTS - Target information TRGTS (1)- Target type + 100 - target sequential number (negative) (2)- X coordinate at time 1 (3)- Y coordinate at time 1 (4)- Time 1 (5)- X coordinate at time 2 (6)- Y coordinate at time 2 (7)- Time 2 (8)- Velocity (9)- Ferrous metal (present=1, not present=0, integer) (10)- Altitude (11)- EVID code + 10000 x leg number + 100000 Battle 200000 Culture (12)- Target length (13)- Visual security descripter Recorded on data set MTAPE (8) - CRTARG A one and 13 zeros are written as last record	CULTEX, BATEX, BATLTG	PRERUN 6
33 COMMON/TRGB/NTAR, TARG (13, 200)			
Used to store target parameters for moving arrays	NTAR - Number of targets TARG - Target information TARG - Same as TRGTS in COMMON/TARG/ Except for TARG (1,) target ID + 1000 * leg number TARG (11,) target type location Data is recorded on data set MTAPE (7) - JFTAR	PREMN4, MVS	PRERUN 4
34 COMMON/TRGB/NTAR, TARG (13, 500)			
Used to store target parameters	NTAR and TARG are same as above Step 8 reads in the targets on data sets MTAPE (7) and MTAPE (8) Stores then in TARG and generates the rest of the targets using subroutines TARGEX and TARGBR	PREMN8, TARGEX, TARGBR, ELPEX ELPDT	PRERUN Step 8
35 COMMON/ENTER/			
Used exclusively by PRERUN Step 7 - EVNT23, ENVIR	Refer to Appendix C, Volume 2, User's Manual	EVENT23 ENVIR	PRERUN 7

Table 2.1-VI (Cont.)

PRERUN COMMON AREAS

	USE	VARIABLES	USED BY	WHERE USED
36	COMMON/UPDN/NT, UDTM (5000), KMUD (101), KREL (41), KDLK (301), KARR (301)			
	Used for updown times by PREMN1, UPDN6, UPDN8, UPDN10	NT - Total Number of up/down times UDTM - Up/Down string KMUD - Pointer for Up/Down times in UDTM KREL - Pointer for relays for UDTM KDLK - Pointer for data links for UDTM KARR - Pointer for arrays for UDTM	PREMN1 UPDN10 UPDN6 UPDN8	PRERUN 1
37	COMMON/UPDN/NT, UDTM (5000), KARR (301), KSEN (401)			
	Used for updown times by PREMN2, UPDN3, UPDN19, PREMN3, PSNP, PSNP19	NT - Total Number of up/down times UDTM - Up/down string KARR - Pointer for arrays for UDTM KSEN - Pointer for sensors for UDTM	PREMN2 PSNP UPDN3 PSNP19 UPDN19 PREMN3	PRERUN 2 - 3
38	COMMON/UPDOWN/NS, UDTM (5000), KARR (301), KSEN (401)			
	Used for updown times by PREMN4, MVS	NS - Total Number of up/down times UDTM - Up/down string KARR - Pointer for arrays for UDTM KSEN - Pointer for sensors for UDTM	PREMN4 MVS	PRERUN 4
39	COMMON/UTMCON/XSW, YSW, XNE, YNE			
	Used to store game play area definition	XSW, YSW - X-Y coordinates of South-West Corner XNE, YNE - X-Y coordinates of North-East Corner This is recorded on MTAPE(2) - JFMSTR	INMAIN CONVRT PREMN9 PREMNA PREMNB	PRERUN Ø, 9 - 11
40	COMMON/WEVT/MEV (10000), IVE (10000)			
	Used to store MSM events by PRERUN steps PREMNA and PREMNB	MEV - MSM Events generated by previous steps IVE - Temporary Event list	PREMNA PREMNB	PRERUN 1Ø - 11
41	COMMON/WORK/LL (1000), TT (1000)			
	Used for working storage by UPDN 1, 3, 5, 6, 8, 10, 19	LL - Code identifying times LL = K in sub COMMLP P. 2-37 TT - Actual up/down times TT(+)-planned up time TT(-)-planned down time	UPDN1, 3, 5, 6, 8, 1Ø, 19	PRERUN 1, 2
42	COMMON/WORK/LWORK (1000)			
	Used for working storage by PSNP PSNP19, MVS	LWORK - Working storage	PSNP MVS PSNP19	PRERUN 3, 4

Table 2.1-VI (Cont.)

PRERUN COMMON AREAS

USE	VARIABLES	USED BY	WHERE USED
43 Unlabelled COMMON COMMON IX, SNER, SNEPE, GUNPE, SNPUD, JPMAX, MMAX, NMAX, IMAX, TRAIL			
Used exclusively in PRERUN Step 3 by PSNP, PSNP19, SNPGT for communication between the routines	IX - Starting and subsequent random integers for random numbers SNER - Page 2-48, Volume I, part I SNEPE - Page 2-48, Volume I, part I GUNPE - Page 2-52, Volume I, part I SNPUD - Page 2-52, Volume I, part I JPMAX - Page 2-53, Volume I, part I MMAX - Page 2-53, Volume I, part I NMAX - Page 2-53, Volume I, part I IMAX - Page 2-52, Volume I, part I TRAIL - Page 2-53, Volume I, part I	PSNP PSNP19 SNPGT	PRERUN 3

The number of elements that may be in a game are primarily limited by the amount of storage allocated for the master data stream and the pointers. The present subroutines have been compiled with a dimension of 12,000 for the master data stream, IDATA (or equivalent ID). A change in this dimension would require a recompilation of all the subroutines using DATAIN.

To accommodate a very large data set, it would be desirable to change PRERUN Step 0 so that each planner input set is written as a separate record. A subroutine could then be written which could be used by each PRERUN executive subroutine to read in only the data sets required for that particular step. Such a subroutine could redefine the pointers in LDATA, so that no further changes would be required other than a recompilation of the eight executive subroutines involved.

Present bounds on planner inputs as restricted by dimension statements are as follows:

Sets	Planner Input Set No.	Pointers	Common Area
Arrays	I, XIX, XX	KARR(301)	UPDOWN
Sensors	III	KSDN(401), KSEN(401),	UPDOWN
		KSSN(401)	STASEN
		KXYT(401)	PXYTP
Monitors	VI	KMUD(101)	UPDOWN
Relays	VIII	KREL(41)	UPDOWN
Data Links	X	KDLK(301)	UPDOWN

The pointer must be set to a dimension at least one greater than the desired number of elements. The reference to the last value of the pointer must be changed in PRERUN Steps 1, 2, 3 and 4 (i.e., UPDN6 sets, KMUD(101) = NT+1, etc.).

Secondary limitations have been imposed by the dimension statements for the storage of up/down times in UDTM (common area UPDOWN), SXYTT (common area STASEN), SXYT (common area PXYTP), IEV and IVE (common area EVENTS), and TARG (common area TRGB). The dimensions for these arrays are not readily determined in advance since they depend on probabilistic game results. The PRERUN subroutines are so designed that it is not difficult to modify them such that they will dump these arrays on discs and reload the arrays as often as necessary if storage is a problem. This has already been done for the targets generated by the Battle and Culture subroutines (see BATLTG for an example).

2.2 MAIN SIMULATION MODEL -- MSM

2.2.1 Introduction

The Main Simulation Model (MSM) is a complex of 75 subprograms that controls the simulation of actual sensor system performance, associated input and output, and auxiliary computations necessary for its operations.

The combination of MSM and PRERUN forms the basic simulation structure for the System Assessment Model (SAM). MSM thus uses the results of initial processing by PRERUN. The basic block diagram for this combination, Figure 2.2.1, shows the data linkage from PRERUN to MSM as well as other data inputs to MSM. Functionally, the PRERUN-generated sequence of events is the important input to MSM, as it dictates the dynamic executions to be simulated. The other input files provide time, terrain, atmospheric and miscellaneous data, and the initial set of system parameters.

MSM output consists of "immediate" printer output and a data file, on binary tape or disk, called MSMOUT.* The latter may be used as input for one or more OUTPUT PROCESSOR (post-MSM) analysis programs. These outputs provide time histories of sensor reports, both "game play" and "game truth," and auxiliary but related information.

In the following discussion, the rather complex structure of MSM is summarized according to various viewpoints. From the external point of view, the important aspects of MSM concern only what it accomplishes, which is tantamount to specification of what the inputs are and what the final output results are. Thus, the summary begins with a discussion of input data, including the operational events that control MSM (Section 2.2.2), and of output information flow (Section 2.2.3).

The internal structure of MSM is discussed in several major subsections. Section 2.2.4 provides an overview of the basic program structure and content. Section 2.2.5 then centers discussion on the individual subprograms. Finally, Section 2.2.6 discusses the content and storage logic for data storage arrays required by nearly all of the subprograms. Within this Volume, additional information of a detailed nature is provided in Appendices, referenced at the appropriate points. For individual subprograms, Volume III (program listings, AUTOFLOW diagrams) may be referenced. For the mechanism of program operations, the Users' Manual (Volume II) may be referenced. For the special and important case of MSM sensor routines, extensive documentation may be found in Section 3 of this Volume; in this section, only the placement of sensor routines within the overall MSM program structure is explicitly discussed.

2.2.2 Input Data

Input information for MSM (see Figure 2.2.1) falls into five categories. Header cards, terrain data and system parameters are essentially

*Note: This precludes running PRERUN and MSM on two different machines.

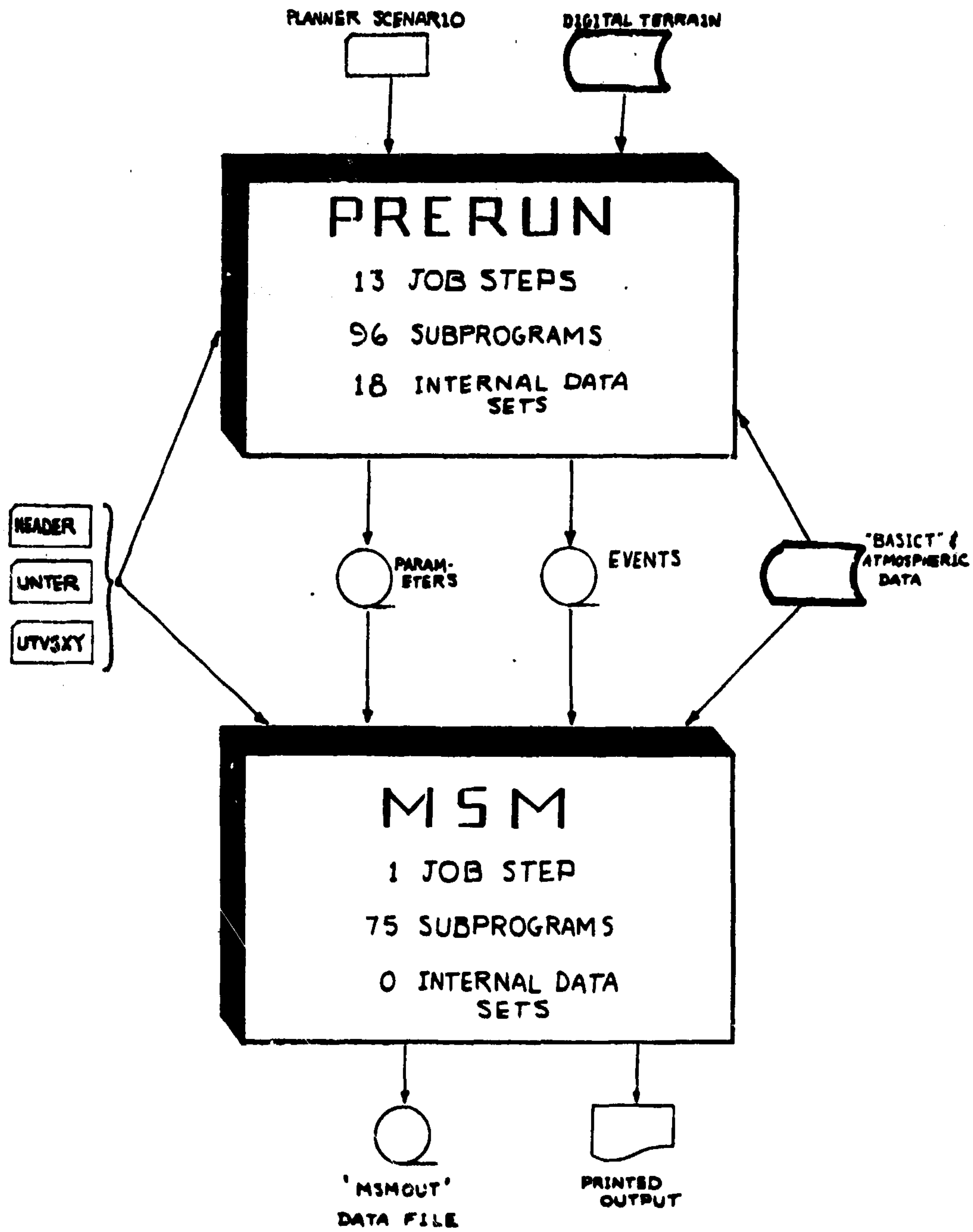


Figure 2.2.1 MSM – PRERUN RELATIONSHIP AND EXTERNAL DATA SETS

static -- although alterations to system parameters are allowed via special events. Atmospheric data and events are dynamic, the former typically changing (as game time advances) much less frequently than the latter. The data sets are briefly discussed, by category, in the following text.

2.2.2.1 "Header Cards"

The so-called "header cards" (18 punched cards), also used as input to PRERUN, are not critical to MSM operations.* With the exception of reference state integers for random number generators, card content is used only in alphanumeric form. Specifically:

(a) The complete content of the cards is printed on the first page of MSM output, in alphanumeric form, for user reference.

(b) Significant alphanumeric content (essentially run identification), on the first two cards is preserved in storage, and used consistently as page heading for each of the MSM printed page outputs.

(c) Provided non-zero (or non-blank) entries are provided on the last two cards, the integer values in the appropriate fields are used to set or initialize the states of the uniform and gaussian random number generators used by MSM.

Exact formats for these 18 header cards, irrelevant to the immediate description of MSM, may be found in Volume II. All operations with header cards are handled by EXEC1.

2.2.2.2 Terrain Data

MSM does not use detailed digital terrain data, the processing of which is a PRERUN task. MSM does, however, require two sets of terrain parameters:

"UNTER" Table	Terrain and foliage descriptor parameters, one set for each (of the typically 8 or so) different unit terrain types.
"UTVSXY" Table	Packed-format data, from which a specific unit terrain type code (IUT value) can be derived from specified x, y coordinates.**

*Indeed, MSM would operate with 18 blank cards for this data set.

**Subroutine IUTEVL (IUT EVaLuate) is supplied to make this evaluation upon call from other programs.

In the current program setup, these data are entered into MSM on punched cards and read into storage by subroutine TERAN (via subroutine TANDT and EXEC 1). Exact cards formats and content are described in Volume II.

2.2.2.3 System Parameters

Physical parameters that define the elements of a scenario are transferred to MSM from PRERUN via data set JTFWDF. These data originate in the 29 sets of planner scenario specifications* that are entered in PRERUN step 0. Because of the difference in data requirements between PRERUN and MSM, however, content of data set JTFWDF differs from content of the original planner data sets in these aspects:

(a) MSM requires only 13 data categories of the original 29. Table 2.2-I lists these categories, and their correspondences with original data set category numbers.

(b) Within the 13 accepted categories, some data are not needed and are not passed to MSM.

(c) As passed to MSM, all data are in consistent internal units (e.g., radians for angles, meters for distances).

(d) For certain parameters, particularly emplacement positions and times of operation, there is a distinction between "game play" values as given by planner data, and "game truth" values reflecting random variations that would occur in field operations. MSM is given only game truth values, as derived in PRERUN by simulation of random variations.

Despite the abridgment of content, the number of parameter values passed to MSM is typically very large**. In addition, these values are constantly referenced throughout computational processing by MSM subprograms. As a result, details of where parameter data are stored, of content of the various parameter lists, of associated pointer tables, of

---

* Explicit content of these data sets is described in Volume II (see PRERUN, Step 0).

** Space for 20000 words was reserved in initial MSM coding.

TABLE 2.2-I

CATEGORIES OF SYSTEM PARAMETER DATA

(INPUT TO MSM)

Category Number*	Title	Analogous Planner Set**
1	UGSARRAYS	I
2	STASCAN ARRAYS	XIX
3	MOVARRAYS	XX
4	BLUE FORCES	XXI
5	RED FORCES	XXII
6	SENSORS	III
7	SENSOR DESCRIPTORS	IV
8	FIRETRAPS	V
9	MONITORS	VI
10	DATA LINKS	X
11	PATHS	XII
12	FORCE TYPES	XIII
13	COVERAGE/SCAN	XIV

* These correspond to the first 13 partitions of storage array MSMPAR. See Section 2.2.6.

** See PRERUN documentation, (Volume I and Volume II) for description of the original 29 planner data sets (scenario specification).

implied subprogram design... were critical to MSM design and are critical to full understanding of MSM program operations. Thus, directly or indirectly, considerable amount of later documentation in this Volume is related to system parameters*.

Finally, it should be noted that not all of the system parameter data stored within MSM are static. Changeable parameters include up/down flags, coordinates of sensors (upon reemplacement), and certain operational-physical parameters for some sensor types. Certain dynamic input events (see Section 2.2.2.5) are designed to effect parameter changes at the appropriate game times.

2.2.2.4 Atmospheric and Time Data

Time parameters and atmospheric data tables are related, in that they are stored within the same physical data file (the former as the first record).

The time parameters (collectively referred to as "BASICT" information, after the name of the labeled common area used for storage) are read into storage by subroutine TANDT, which is called by EXEC1. The only variable in this set explicitly used within MSM is ITODST -- the time of day (integer, seconds) at which simulation is specified to begin.

An atmospheric data table is initially read into storage, then updated by new read operations as game time advances. Control of atmospheric data reading is by subroutine EXEC1B. Content of these tables (called ATMENV tables) is documented in Section 5 of this volume.

2.2.2.5 Events

Primary dynamic input to MSM is a time ordered sequence of simulation events, prepared by PRERUN and placed on a disk data set for access by MSM. Although the total number of computer words required for the event sublists is typically very large, the data set is blocked into groups not exceeding 900 words, and only 900 words of storage are reserved within MSM for these data. Level 1 executive routine EXEC1B has control of reading event data (one 900-word block at a time, as required during the simulation), and of causing these events to be executed in sequence.

---

*Section 2.2.6 and Appendix D discuss details of storage allocation within labeled common area /BIGSTR/. Section 2.2.5 discusses associated subprograms, of which EXECIA, DUMPMS, PARPTR, PARVLU, ARRPTR, ARRVLU, TGTPTR and TGTVLU are especially relevant. Dynamic changes in parameters are handled by level 2 executive routines EX2SPC (sensor parameters), EX2UPD (up/down flags) and EX2SPC (sensor position coordinates).

Each event has an associated sublist (or simply list) of computer words. The structure of these lists is designed to be adaptable to program growth in two respects:

(a) The event type code is an element of a list (first word). Although 11 event types are accepted by the initial MSM program, no restriction on the number of types exists in the basic input format.

(b) Lists are not restricted to a fixed length; the word count for the list is entered as a list data element (second word). List length is not even necessarily the same for a particular event type; two of the 11 event types provided in the initial MSM program have in fact variable length lists.

It may be noted that inclusion of list lengths as data allows contiguous packing of event data, for any mixture of list lengths.

The descriptive titles for the 11 event types programmed for MSM are given in Table 2.2-II following page. Detailed discussion of events -- exact meanings, list contents, formats, units, etc. -- is given in Appendix B.

2.2.3 Output Data

MSM provides formatted output on the system printer for immediate use and information, and unformatted (binary) output on tape or disk for subsequent input to one or more OUTPUT PROCESSOR programs.

The relationship of output channels to the subprograms that write output data is shown in Figure 2.2-2. Details are provided in Section II and its associated Appendices, in support of summary information given below.

2.2.3.1 Printer Output

Immediate printed output covers a variety of information. The current program provides eight categories of information, listed below. The third item (c) is the primary dynamic output; the last item (h) is the basic system summary:

(a) Listing of the alphanumeric content of the 18 header cards provided by the user.

(b) A listing ("dump") of the common area /BIGSTR/, in which system parameters are stored.

(c) Time histories of sensor reports. Printer formatting essentially places "game play" information on the left hand side of the page, and "game truth" information on the right hand side.

Table 2.2-II

LIST OF EVENT TYPES

Event Type Code	Event Descriptive Name
1.	Sensor Interrogate (against target(s))
2.	Sensor False Alarm
3.	Sensor Parameter Change
4.	Sensor Up/Down Status Control
5.	Monitor Up/Down Status Control
6.	Data Link Up/Down Status Control
7.	Firetrap Up/Down Status Control
8.	Arrays: Emplace/Cease Operations
9.	Battlefield Illumination
10.	Sensor Reposition (coordinate change if reemplacement occurs)
99.	END (terminate MSM processing; no more event data)

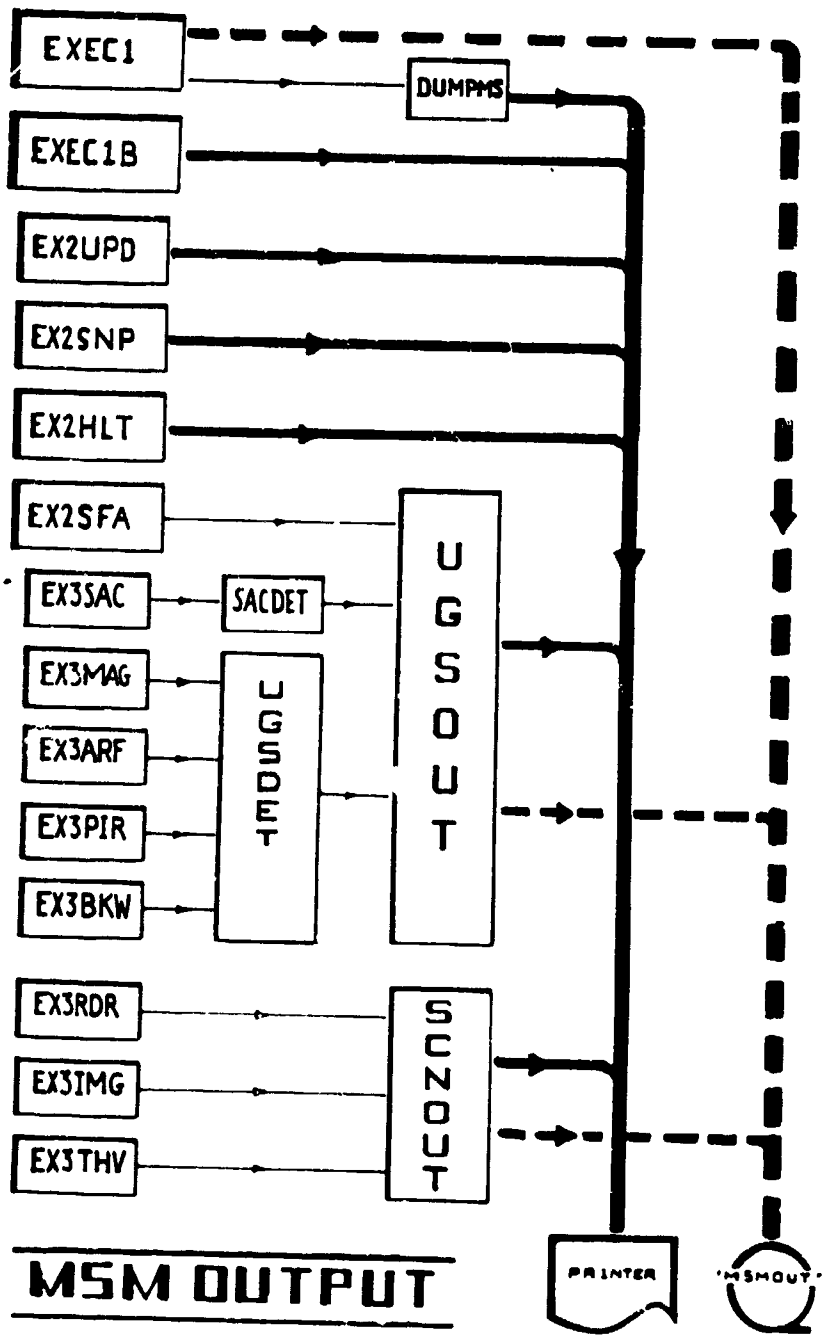


Figure 2.2-2 MSM OUTPUT CHANNELS AND SUBPROGRAMS

(d) Periodic summaries ("system snapshots") of sensors and arrays currently active. As originally coded, but easily altered, these snapshots are provided at the end of every two hours of game time.

(e) Notations on effective times that sensor arrays are emplaced or cease operations.

(f) Notations on begin and cease times of firetrap operations.

(g) Notations on beginning of precipitation and its rate, and on cessation of precipitation.

(h) System summary, provided at termination of simulation, based on some 128 system counters assigned to various events and results.

Subprograms generating printed output (see Figure 2.2-2) are associated with these categories as follows:

Printed Output Category	Responsible Subprogram
a.	EXEC1
b.	DUMPMS/EXEC1
c.	UGSOUT and SCNOUT
d.	EX2SNP
e.	EX2UPD
f.	EX2UPD
g.	EXEC1B
h.	EX2HLT

Additional details, as well as illustrative printed information for each category (as generated by actual exercise of the simulation model) are given in Appendix C.

2.2.3.2 Binary Output -- 'MSMOUT'

The MSM binary output file, called 'MSMOUT,' contains data on tape or disk that would be useful as input for one or more applications of post-MSM processing programs. Structured in many respects like the input event file, this output allows variable length lists and can be readily expanded to any number of different event type codes. In the initial version of MSM coding, three categories of output information are provided corresponding to 10 distinct event type codes. An 11th type code is added to indicate end of data.

The three categories of output information, and the MSM subprogram directly responsible for the write operation, are listed below:

System parameters	EXEC1
Sensor-to-monitor reports by unattended (UGS) sensors	UGSOUT
Sensor reports from scanning (attended) sensors	SCNOUT

The end-of-data indicator is written by subroutine EX2HLT.

The explicit content of the sublists for each of the output event type categories is given in Appendix B.

2.2.4 The MSM Subprogram Structure

MSM comprises a total of 75 subprograms,* in 10 general categories, linked together into a single job step. Table 2.2-III gives the names of the categories into which programs are classified, and the total number of subprograms per category. Table 2.2-IV an expansion of 2.2-III, lists explicit subprogram names by category.

In the immediate sequel, the basic structure of MSM is described in terms of simplified block diagrams, in which subprograms of an auxiliary nature are suppressed. The roles of the various subprograms are then discussed, in a category-oriented sequence of subsections. In general, details supplementing the general text are provided by "comments blocks," extracted from program listings and inserted into this Volume as Figures within the appropriate subsections. Actual program listings and AUTOFLOW diagrams (Volume III) may be referenced for additional details.

2.2.4.1 Basic MSM Flow Diagrams

The basic flow of MSM operations hinges on (a) control of program sequencing by executive routines, (b) sensor routines, and (c) flow of information to the two output channels. Subprograms not within these executive-sensor-output categories have auxiliary roles; they are essential for complete computation, but do not fundamentally influence overall logical structure.

A block diagram showing the relationship of subprograms to the two output channels has previously been shown (Figure 2.2-2; Section 2.2.3).

---

*Not counting standard FORTRAN subprograms (e.g., SQRT). The main program and a single BLOCK DATA subprogram are included in the count.

Table 2.2-III

MSM SUBPROGRAM COUNTS BY CATEGORY

CATEGORY	NO. OF PROGRAMS
0. BLOCK DATA	1
1. EXECUTIVE ROUTINES, LEVEL 1	3
2. EXECUTIVE ROUTINES, LEVEL 2	8
3. EXECUTIVE ROUTINES, LEVEL 3	8
4. SENSOR SUBROUTINES*	11
5. OUTPUT & OUTPUT-RELATED	9
6. INPUT AUXILIARIES	3
7. SYSTEM UTILITY ROUTINES	[illegible]
8. STORAGE ACCESS UTILITIES	6
9. GEOMETRY & OTHER AUXILIARY	17
Total:	75

* Nine basic routines, one for each sensor generic type, plus two sensor-unique auxiliaries for the IMAGE routine.

Table 2.2-IV

MSM SUBPROGRAM STRUCTURE
SUBPROGRAM NAMES BY CATEGORY
(Sheet 1 of 4)

	BLOCK DATA
0. MSMBLK	BLOCK DATA SUBPPOGRAM FOR MSM

	EXEC ROUTINES, LEVEL 1
1. EXEC1	MAIN PROGRAM .
2. EXEC1A	PARAMETER CONTROL. INPUT, EDITING, STORAGE ALLOCATION, POINTER TABLE SETUP.
3. EXEC1B	SUPERVISES DYNAMIC EVENTS. DIRECTS CONTROL TO LEVEL 2 EXEC ROUTINES, ACCORDING TO EVENT TYPE.

	EXEC ROUTINES, LEVEL 2
4. EX2SNR	CONTROLS EVENTS TYPE 1, SENSOR INTERROGATE. DIRECTS CONTROL TO LEVEL 3 SERIES (EX3...) ACCORDING TO SENSOR TYPE.
5. EX2SFA	CONTROLS EVENTS TYPE 2, SENSOR FALSE ALARMS
6. EX2SPC	CONTROLS EVENTS TYPE 3, SENSOR PARAMETER CHANGE
7. EX2UPD	CONTROLS EVENTS TYPE 4 THRU 8 (UP/DOWN)
8. EX2BFL	CONTROLS EVENTS TYPE 9, BATTLEFIELD ILLUMINATION, IN TERMS OF STORAGE CONTROL.
9. EX2SRP	CONTROLS EVENTS TYPE 10, SENSOR REPOSITION (CHANGE OF COORDINATES).
10. EX2SNP	SYSTEM 'SNAPSHOT'.
11. EX2HLT	CONTROLS SYSTEM RESULTS SUMMARY AFTER COMPLETION OF DYNAMIC SIMULATION. TERMINATES SIMULATION.

	EXEC ROUTINES, LEVEL 3
12. EX3SAC	SUPERVISORY PROGRAM FOR SEISMIC, ACOUSTIC ROUTINES
13. EX3MAG	SUPERVISORY PROGRAM FOR MAGNETIC SENSOR ROUTINE
14. EX3ARF	SUPERVISORY PROGRAM FOR ARFBUOY SENSOR ROUTINE
15. EX3PIR	SUPERVISORY PROGRAM FOR PASSIVIR SENSOR ROUTINE
16. EX3RDR	SUPERVISORY PROGRAM FOR RADAR SENSOR ROUTINE
17. EX3IMG	SUPERVISORY PROGRAM FOR IMAGE SENSOR ROUTINE
18. EX3THV	SUPERVISORY PROGRAM FOR THERMVEW SENSOR ROUTINE
19. EX3BKW	SUPERVISORY PROGRAM FOR BREAKWIP SENSOR ROUTINE

Table 2.2-IV

MSM SUBPROGRAM STRUCTURE
SUBPROGRAM NAMES BY CATEGORY
(Sheet 2 of 4)

	SENSOR ROUTINES
20. SEISTG	SENSOR ROUTINE, SEISMIC
21. ACOUTG	SENSOR ROUTINE, ACOUSTIC
22. MAGTG	SENSOR ROUTINE, MAGNETIC
23. ARFTG	SENSOR ROUTINE, ARFBUOY
24. PIRTG	SENSOR ROUTINE, PASSIVE IR
25. RADAR	SENSOR ROUTINE, RADAR
26. IMAGE	SENSOR ROUTINE, IMAGE DEVICES
27. THERML	SENSOR ROUTINE, THERMAL VIEWER
28. BRKWIR	SENSOR ROUTINE, BREAKWIRE
29. ANG	AUXILIARIES TO
30. QUAD	IMAGE ROUTINE

	OUTPUT AND OUTPUT-RELATED ROUTINES
31. UGSOUT	OUTPUT ROUTINE FOR UGS
32. SCNOUT	OUTPUT ROUTINE FOR SCANNING SENSORS
33. SACDET	INTERFACE, EX3SAC TO UGSOUT
34. UGSDET	INTERFACE, EX3... TO UGSOUT FOR OTHER UGS
35. PGSKIP	BASIC PAGE SKIP, HEADER PRINT
36. PGSKP2	SPECIAL PAGE SKIP, SENSOR REPORT HEADING PRINT
37. TIMOUT	INTERNAL-TO-EXTERNAL TIME CONVERSION
38. ALFCVT	SPECIAL INTEGER TO ALPHANUMERIC CONVERSION
39. DUMPMS	STORAGE LISTING AFTER EXEC1A PROCESSING
NOTE:	EX2HLT ALSO GENERATES PRINTED OUTPUT; EXEC1 GENERATES BINARY OUTPUT AND PRINTED OUTPUT

	INPUT AUXILIARIES (EXEC LEVEL 1)
40. TANDT	CALLED BY EXEC1, TO READ 'TIME AND TERRAIN' DATA INTO STORAGE (BASICT, UTVSXY AND UNTER COMMON AREAS)
41. PARMIN	SIMPLE READ ROUTINE FOR PARAMETERS
42. TERAN	CALLED BY TANDT TO READ TERRAIN DATA

Table 2.2-IV

MSM SUBPROGRAM STRUCTURE
SUBPROGRAM NAMES BY CATEGORY
(Sheet 3 of 4)

		SYSTEM TYPE UTILITY ROUTINES
43.	ERASE	CLEARS (SETS TO 0) BLOCK OF STORAGE
44.	TRNSFR	BLOCK TRANSFER OF DATA
45.	URN	UNIFORM RANDOM NUMBER GENERATOR
46.	URNORG	SETS INITIAL STATE OF URN GENERATOR
47.	URNASK	INTERROGATES STATE OF URN GENERATOR
48.	GRN	GAUSSIAN RANDOM NUMBER GENERATOR
49.	GRNORG	SETS INITIAL STATE OF GRN GENERATOR
50.	GRNASK	INTERROGATES STATE OF GRN GENERATOR
51.	ERFC	COMPLEMENTARY ERROR FUNCTION

		STORAGE ACCESS UTILITY ROUTINES
52.	PARPTR	DETERMINES POINTER TO SPECIFIED WORD, ID, AND GENERAL PARAMETER STORAGE AREA
53.	PARVLU	SIMILAR TO PARPTR, BUT RETURNS PARAMETER VALUE RATHER THAN POINTER VALUE
54.	ARRPTR	SPECIAL FORMS OF PARPTR, PARVLU (WITH SIMPLER
55.	ARRVLU	CALLING SEQUENCES), FOR PARAMETERS OF SYSTEM
56.	TGTPTR	ARRAYS (SENSOR ARRAYS) AND OF 'TARGETS'
57.	TGTVLU	(RED, BLUE FORCES), RESPECTIVELY.

		GEOMETRY AND OTHER AUXILIARY ROUTINES
58.	TGTLG	BASIC COMPUTATION OF STATIC MOTION PARAMETERS FOR SPECIFIED TARGET AND LEG
59.	TGTLXY	GETS DYNAMIC VALUES (I.E., FOR EXPLICIT TIME) OF TARGET COORDINATES
60.	STRECT	'STATIONARY RECTANGLE' GEOMETRY CALCULATION
61.	STCIRC	'STATIONARY CIRCLE' GEOMETRY CALCULATION
62.	EVNEFD	EVALUATES NO. OF TARGET ELEMENTS IN SENSOR FIELD
63.	CLOSEL	DETERMINES CLOSEST TARGET ELEMENT (TO SENSOR)

Table 2.2-IV

MSM SUBPROGRAM STRUCTURE
SUBPROGRAM NAMES BY CATEGORY
(Sheet 4 of 4)

64.	KSTVLU	DETERMINES 'KSTRNG' (STRENGTH OF SIGNAL INDEX)
65.	FTPAR1	DETERMINES STATIC AND DYNAMIC PARAMETERS,
66.	FTPAR2	RESPECTIVELY, FOR FALSE TARGETS
67.	BFIASK	USED FOR INITIAL SEARCH (OUTSIDE OF LOOP) AND
68.	BFILUM	ACTUAL ACCESS (INSIDE LOOP), BATTLEFIELD IL- I UMINATION DATA FOR SUBROUTINE IMAGE
69.	SCAN1	SCAN ROUTINES, FOR TWO SCANNING LOGICS, USED FOR
70.	SCAN2	SENSORS IN SECTOR SCAN MODE
71.	SETSC1	PRELIMINARY ROUTINES (OUTSIDE OF LOOP) TO SET WORK-
72.	SETSC2	ING PARAMETERS FOR SCAN1, SCAN 2, RESPECTIVELY
73.	ITODEV	EVALUATES 'TIME OF DAY' FROM ITIME (TIME OF DAY REQUIRED BY SOME SENSOR ROUTINES)
74.	IUTEVL	EVALUATES IUT INDEX (UNIT TERRAIN TYPE) FROM X, Y

A block diagram showing all other essential features of MSM structure is given in Figure 2.2-3. This figure is clearly keyed to the various levels of executive routines, down to the EX3... series that act as interfaces with the primary sensor routines.

The basic sequence of operations, by levels as shown in Figure 2.2-3, proceeds as follows:

LEVEL 1:

EXEC1, EXEC1A	Initial one-time-only processing to load storage with required parameter data.
EXEC1B	General control of dynamic simulation events; passes control to appropriate level 2 exec for specific simulation tasks.
LEVEL 2:	Responsibility for specific tasks, as assigned by EXEC1B.
LEVEL 3:	Sensor routines and level 3 execs, called for the event "sensor interrogate" only by level 2 routine EX2SNR.

In the following Section 2.2.5, the roles of all MSM subprograms are discussed. In particular, the discussions of level 1, level 2 and level 3 executive routines provide additional information about the logical structure of Figure 2.2-3.

2.2.5 MSM Subprogram Descriptive Summaries

2.2.5.1 MSM Executive Routines -- Level 1

Although not significant contributors to the purely computational aspects of MSM simulation, the executive routines have a unique importance in control of computation, and would be the immediate routines of interest in event of program extensions.

As previously indicated (e.g., Figure 2.2-3), executive routines are considered to form a three-level hierarchy. The level 1 exec routines are the main program EXEC1 and two subroutines (EXEC1A and EXEC1B), controlling such major operations as to be considered level 1.

EXEC1, the MSM main program, is relatively short and does little but call other routines and handle some simple input/output operations. EXEC1 specifically:

BASIC PROGRAM STRUCTURE: MSM

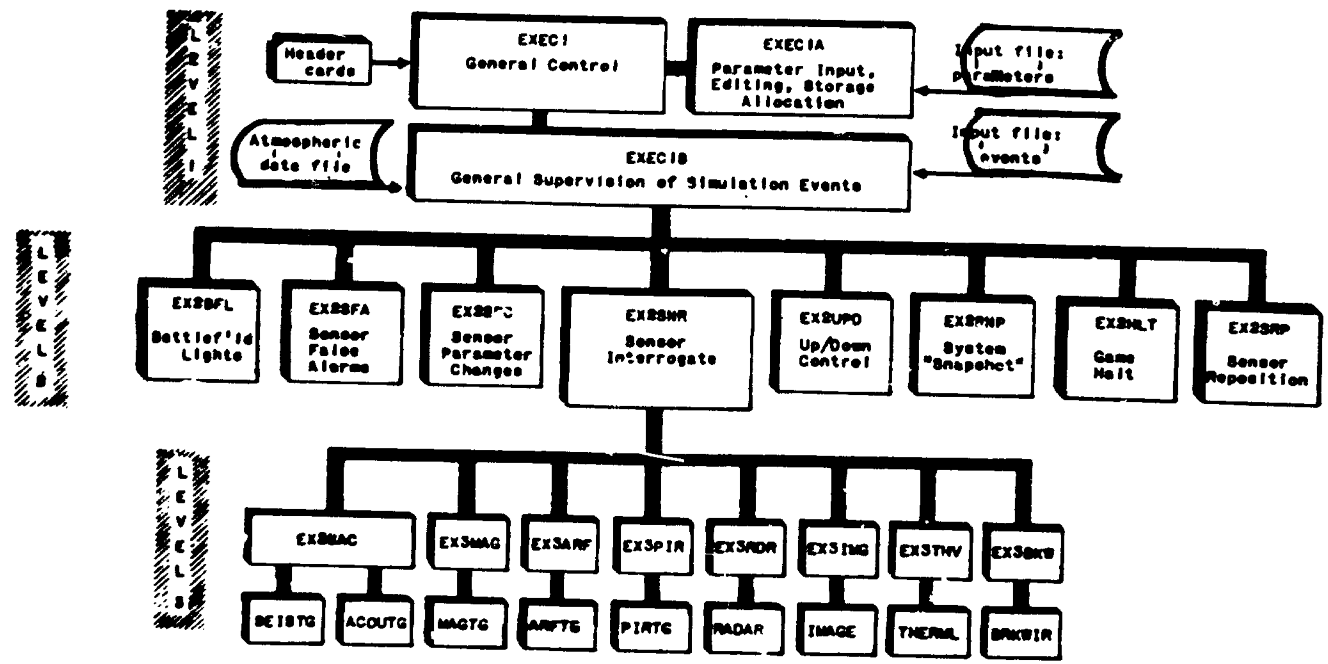


Figure 2.2-3 BASIC PROGRAM STRUCTURE: MSM

(a) Reads and prints the 18 header cards, stores run identification alphanumeric information in storage, and sets the random number generators according to the content of cards 17 and 18.

(b) Indirectly reads "Time and Terrain" data by calling subroutine TANDT (tables read are BASICT, UNTER and UTVSXY).

(c) Calls EXEC1A (see discussion below of EXEC1A).

(d) Calls an output routine, DUMPMS, to list ("dump") system parameter storage that has just been filled by EXEC1A.

(e) Writes the same system parameter data as output event type 9999, on the binary output file.

(f) Transfers control to EXEC1B.

After EXEC1B is called, control never reverts to EXEC1.

EXEC1A is called once, by EXEC1. It supervises the reading of system parameter data (from the PRERUN generated data set), allocates storage locations within storage array MSMPAR, edits the original data, and fills in the tables of pointers to the subsets of MSMPAR.

EXEC1B is called once, by EXEC1, and thereafter becomes the effective main program for MSM simulation. EXEC1B has general control over all dynamic simulation events, most of which are specified by the EVENTS data set from PRERUN, some of which correspond to updating of atmospheric tables or to internally generated events. Specific execution tasks are then directed to the appropriate level 2 subroutines, according to the type of event being processed.

Specific operations under general control of EXEC1B are:

(a) Reading initial atmospheric data table, and updating this table as required as game time advances.

(b) Reading events data from the external data file, in blocks of (at most) 900 words; and regeneration of the stored block after the "current" one has been executed.

(c) Unpacking specific event sublists from the 900 word block, interpreting, and passing control to the appropriate level 2 subroutine according to the event type specified in the first word of each sublist.

(d) Periodically during game time (initial MSM coding: every two hours), request a so-called system snapshot from level 2 subroutine EX2SNP.

(e) Rather minor printout operations (notations about beginning or end of precipitation).

It is perhaps appropriate to mention the role of EXEC1B in the context of potential program growth. The critical operation for dynamic events is the decoding of event types and the transfer of control to a level 2 exec routine tailored to handle the particular event type noted. Coding for this step hinges on a COMPUTED GO TO statement, that branches according to event type to a CALL statement to a level 2 exec. Incorporation into the program of a "new" event type requires, at the EXEC1B level, only the extension of the number of branches for the GO TO statement, and a new CALL for a presumably new level 2 exec.

Additional detail information on the three individual level 1 exec routines is given in Figures 2.2-4 through 2.2-6. These figures are the formal "Comments Sections" extracted from program listings.

```
C*****************************  EXEC1  (MAIN)  ***************************
C*                                                                        *
C*                              EXEC1                                     *
C*                       MAIN PROGRAM, MSM                                *
C*                                                                        *
C*    PURPOSE                                                             *
C*        MAIN EXECUTIVE PROGRAM FOR THE MSM PORTION OF THE               *
C*        PHASE I SYSTEM ASSESSMENT MODEL (SAM), CORNELL AERO             *
C*        LAB PROJECT SAM I.  EXEC1, WITH ITS AUXILIARIES                 *
C*        EXEC1A AND EXEC1B, FORM THE LEVEL 1 SET OF EXECUTIVE            *
C*        ROUTINES FOR MSM.                                               *
C*                                                                        *
C*    CALLING SEQUENCE                                                    *
C*        MAIN PROGRAM                                                    *
C*                                                                        *
C*    REMARKS                                                             *
C*        PROGRAM EXPECTS 18 'HEADER CARDS' IN CARD READER.  ALL 18       *
C*        ARE IMMEDIATELY PRINTED ON FIRST PAGE OF OUTPUT FOR USER'S      *
C*        GUIDANCE (IDENTIFICATION OF RUN ETC.).                          *
C*                                                                        *
C*        SIGNIFICANT CONTENT OF FIRST 2 CARDS IS SAVED TO BE USED        *
C*        AS TITLE FOR SUBSEQUENT PAGES OF PRINTED OUTPUT.  THE NU-       *
C*        MERIC VALUES ON THE LAST TWO CARDS ARE USED AS INITIAL          *
C*        REFERENCE VALUES FOR UNIFORM AND GAUSSIAN RANDOM NUMBER         *
C*        GENERATORS.                                                     *
C*                                                                        *
C*        CONTENTS OF COMMON AREA /BIGSTR/ ARE WRITTEN ONTO MSM           *
C*        BINARY OUTPUT UNIT (73) AS OUTPUT EVENT TYPE 100, AFTER         *
C*        EXEC1A HAS FILLED IT WITH DATA.                                 *
C*                                                                        *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                        *
C*        PGSKIP    CONTROLS PAGE SKIPS (PRINTER), INCLUDING              *
C*                     PRINTING OF STANDARD HEADING READ INTO             *
C*                     STORAGE BY EXEC1.                                  *
C*        URNORG    SETS INITIAL STATE OF UNIFORM RANDOM NUMBER           *
C*                     GENERATOR.                                         *
C*        GRNORG    SETS INITIAL STATE OF GAUSSIAN RANDOM NUMBER          *
C*                   GENERATOR.                                           *
C*        TANDT     CONSIDERED AUXILIARY ROUTINE, TO READ INTO            *
C*                     STORAGE 'TIME AND TERRAIN'...  SPECIFICALLY,       *
C*                     THE CONTENTS OF COMMON AREAS /BASICT/ AND          *
C*                     /UTVSXY/.                                          *
C*        EXEC1A    LEVEL 1 EXEC, TO HANDLE PARAMETER INPUT, EDIT-        *
C*                     ING, STORAGE ALLOCATION AND POINTER TABLES.        *
C*        DUMPMS    CALLED AFTER EXEC1A, TO LIST ('DUMP') THE CON-        *
C*                     TENTS OF THE PARAMETER STORAGE AREA /BIGSTR/,      *
C*                     AFTER ALL EDITING.                                 *
C*        EXEC1B    LEVEL 1 EXEC, TO EXERCISE GENERAL CONTROL OF          *
C*                     SIMULATION EVENTS, ATMOSPHERIC DATA CHANGES,       *
C*                     DYNAMIC SYSTEM 'SNAPSHOTS'.                        *
```

Figure 2.2-4

```
C***************************** EXEC1A ******************************
C*                                                                  *
C*                     SUBROUTINE EXEC1A                            *
C*                                                                  *
C*    PURPOSE                                                       *
C*       AN MSM LEVEL 1 EXECUTIVE ROUTINE.  SUPERVISES THE INPUT,   *
C*       EDITING, STORAGE ALLOCATION AND POINTER TABLES FOR THE     *
C*       13 CATEGORIES OF SYSTEM PARAMETERS PLACED INTO STORAGE     *
C*       AREA MSMPAR.                                               *
C*                                                                  *
C*    CALLING SEQUENCE                                              *
C*       CALL EXEC1A                                                *
C*                                                                  *
C*    DESCRIPTION OF PARAMETERS                                     *
C*       INPUT TO PROGRAM IS A PRERUN-GENERATED FILE OF SYSTEM      *
C*       PARAMETERS, AND THE LJUMP ARRAY IN LABELED COMMON.         *
C*                                                                  *
C*       OUTPUT OF PROGRAM CONTAINED IN LABELED COMMON AREA         *
C*       /BIGSTR/.  DEFINITIONS OF THESE OUTPUT VARIABLES AS        *
C*       FOLLOWS--                                                  *
C*                                                                  *
C*          MSMPAR    LARGE STORAGE AREA (TENTATIVE, BUT AD-        *
C*                      JUSTABLE, DIMENSION OF 20000 WORDS)         *
C*                      CONTAINING SYSTEM PARAMETERS. MSMPAR        *
C*                      IS REGARDED AS COMPRISING 13 BLOCKS,        *
C*                      DENOTED MSM1 THRU MSM13--                   *
C*                         MSM1     UGS ARRAYS                      *
C*                         MSM2     STASCAN ARRAYS                  *
C*                         MSM3     MOVARRAYS                       *
C*                         MSM4     BLUE FORCES                     *
C*                         MSM5     RED FORCES                      *
C*                         MSM6     SENSORS                         *
C*                         MSM7     SENSOR DESCRIPTORS              *
C*                         MSM8     FIRETRAPS                       *
C*                         MSM9     MONITORS                        *
C*                         MSM10    DATA LINKS                      *
C*                         MSM11    PATHS                           *
C*                         MSM12    FORCE TYPE PARAMETERS           *
C*                         MSM13    COVERAGE/SCAN PARAMETERS        *
C*                                                                  *
C*          NOSMSM    TABLE OF NUMBER OF DATA SETS IN EACH BLOCK    *
C*                      (E.G., NOSMSM(1) = NO. OF UGS ARRAYS).      *
C*                                                                  *
C*          IPBIGS    POINTERS TO THE MSMPAR BLOCKS  (E.G. IF       *
C*                       IPBIGS(6) = 4212, THEN THE FIRST WORD      *
C*                       OF MSM6 IS AT MSMPAR(4212) ).              *
C*                                                                  *
C*          IPBLU     TABLE OF POINTERS TO BLUE FORCES DATA SETS.   *
C*                      (E.G., IPBLU(17) WOULD BE THE POINTER TO    *
C*                      THE BEGINNING OF PARAMETERS FOR BLUE        *
```


Figure 2.2-5

```
C*                          FORCE NO. 17).                                 *
C*                                                                         *
C*            IPPATH    TABLE OF POINTERS TO PATHS DATA SETS.              *
C*                                                                         *
C*                                                                         *
C*        THE INPUT TABLE LJUMP HAS VALUES SET BY THE PRIMARY MSM          *
C*        BLOCK DATA SUBPROGRAM, MSMBLK--                                  *
C*                                                                         *
C*            LJUMP     TABLE OF THE LENGTHS OF DATA SETS (E.G.,           *
C*                        LJUMP(6) = 24 -- EACH SENSOR DATA SET            *
C*                        WITHIN MSM6 REQUIRES 24 WORDS).                  *
C*                                                                         *
C*    REMARKS                                                              *
C*        1. DATA SETS WITHIN ANY ONE BLOCK OF MSMPAR ARE OF FIXED         *
C*           LENGTH, WITH TWO EXCEPTIONS --  MSM4  (BLUE FORCES)           *
C*           AND MSM11 (PATHS).  LJUMP(6) AND LJUMP(11) ARE NOT            *
C*           USED FOR DETERMINING STORAGE ALLOCATION OR SUBSEQUENT         *
C*           ADDRESS CALCULATION.  INSTEAD, THE SPECIAL POINTERS           *
C*           IPBLU AND IPPATH ARE USED.                                    *
C*                                                                         *
C*        2. DATA ARE PACKED INTO MSMPAR WITHOUT THE "GAPS" THAT           *
C*           WOULD RESULT FROM FIXED DIMENSION STATEMENTS.  WORDS          *
C*           ARE REAL, INTEGER, LOGICAL OR ALPHANUMERIC AS RE-             *
C*           QUIRED (EVEN THOUGH 'MSMPAR' IS AN INTEGER NAME).             *
C*                                                                         *
C*        4. INPUT PARAMETERS CORRESPOND TO ORIGINAL PLANNER IN-           *
C*           PUTS, BUT (A) SOME OF THE ORIGINAL VALUES, NOT RE-            *
C*           QUIRED BY MSM, ARE OMITTED, (B) SOME OF THE ORIGINAL          *
C*           VALUES REACH MSM AFTER PRERUN MODIFICATION.                   *
C*                                                                         *
C*        5. ALTHOUGH SOME OF THE INPUT PARAMETERS ARE TRANS-              *
C*           FERRED DIRECTLY TO MSMPAR STORAGE, ADDITIONAL EDIT-           *
C*           ING WITHIN THIS ROUTINE MODIFIES OR ADDS TO THE               *
C*           CONTENT.  EDITING OF 'SENSORS' (MSM6) IS THE MOST             *
C*           SIGNIFICANT IN THIS REGARD.                                   *
C*                                                                         *
C*        6. THIS ROUTINE CALLED ONLY BY EXEC1.                            *
C*                                                                         *
C*        7. FOLLOWING MSM13 IS AN AREA OF 160 WORDS RESERVED FOR          *
C*           SUBSEQUENT STORAGE OF BATTLEFIELD ILLUMINATION EVENT          *
C*           PARAMETERS.  THIS AREA MAY BE REGARDED AS MSM14.              *
C*           EXEC1A HAS NO PROGRAM STEPS ASSOCIATED WITH MSM14,            *
C*           EXCEPT TO SUPPLY THE PRIMARY POINTER VALUE IPBIGS(14).        *
C*                                                                         *
C*        8. THE UNCOMMITTED AREA FOLLOWING MSM14 HAS, AT THE              *
C*           CURRENT DATE, NO PLANNED USE.  IPBIGS(15) POINTS TO           *
C*           THE FIRST LOCATION IN THIS 'FREE' OR 'MSM15' AREA.            *
C*                                                                         *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                         *
C*        PARMIN    INPUT SUBROUTINE, RESPONSIBLE FOR ENTERING             *
```

Figure 2.2-5 (Cont.)

```
C*                    ONE DATA SET INTO IDATA ARRAY AT EACH CALL.      *
C*                    (NOTE. FIRST WORD IN IDATA = 9999 SIGNIF-         *
C*                    IES END OF AN MSM* BLOCK).                        *
C*       PGSKIP    CONTROL PRINTER PAGE SKIP, PAGE NUMBERING.           *
C*       ARRVLU    EXTRACTS PARAMETER VALUE FOR A SYSTEM ARRAY.         *
C*       IUTEVL    EVALUEATES THE 'IUT' VALUE, THAT IDENTIFIES          *
C*                    UNIT TERRAIN TYPE AT A GIVEN LOCATION.            *
C*       TRNSFR    BLOCK TRANSFER OF WORDS                              *
C*       PARPTR    PROVIDES POINTER TO A PARAMETER IN STORAGE           *
C*       ERASE     CLEARS (SETS TO 0) BLOCK OF STORAGE                  *
C*       TGTVLU    EXTRACTS PARAMETER VALUE FOR SPECIFIED TARGET        *
C*                    (FORCE), RED OR BLUE                              *
C*                                                                      *
C************************************************************************
```

Figure 2.2-5 (Cont.)

```
C******************************  EXEC1B  ********************************
C*                                                                      *
C*                        SUBROUTINE EXEC1B                             *
C*                                                                      *
C*    PURPOSE                                                           *
C*       MSM LEVEL 1 EXECUTIVE ROUTINE, CALLED BY EXEC1.                *
C*       SUPERVISES THE EXECUTION OF SIMULATION EVENTS--                *
C*                                                                      *
C*            EVENT TYPE 1       SENSOR INTERROGATE                     *
C*            EVENT TYPE 2       SENSOR FALSE ALARM                     *
C*            EVENT TYPE 3       SENSOR PARAMETER CHANGE                *
C*            EVENT TYPE 4       SENSOR UP/DOWN                         *
C*            EVENT TYPE 5       MONITOR UP/DOWN                        *
C*            EVENT TYPE 6       DATA LINK UP/DOWN                      *
C*            EVENT TYPE 7       FIRETRAP UP/DOWN                       *
C*            EVENT TYPE 8       ARRAY UP/DOWN                          *
C*            EVENT TYPE 9       BATTLEFIELD ILLUMINATION               *
C*            EVENT TYPE 10      SENSOR REPOSITION                      *
C*                                                                      *
C*            EVENT TYPE 99      STOP PROGRAM                           *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*       CALL EXEC1B                                                    *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*                                                                      *
C*       EVENTS-TYPE PARAMETERS TAKEN ONE SET AT A TIME FROM            *
C*         STORAGE ARRAY LISTEV, WHICH IS'FILLED' AS REQUIRED           *
C*         FROM A PRERUN -GENERATED FILE.  EXACT FORMAT DEPENDS         *
C*         UPON EVENT TYPE, BUT FIRST FOUR WORDS ARE CONSISTENT--       *
C*                                                                      *
C*            WORD 1    EVENT TYPE CODE                                 *
C*            WORD 2    WORD COUNT OF EVENT SUBLIST                     *
C*            WORD 3    ID NUMBER                                       *
C*            WORD 4    INTEGER TIME VALUE (BEGINNING TIME FOR          *
C*                        INTERVALS, ONLY TIME VARIABLE FOR             *
C*                        POINT-IN-TIME EVENTS                          *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*                                                                      *
C*       THESE ROUTINES (LEVEL 2 EXEC) HANDLE SPECIFIC EVENT            *
C*       TYPES--                                                        *
C*                EX2SNR     EVENT TYPE 1 (SENSOR INTERROGATE)          *
C*                EX2SFA     EVENT TYPE 2 (SENSOR FALSE ALARMS)         *
C*                EX2SPC     EVENT TYPE 3 (SENSOR PARAM. CHANGE)        *
C*                EX2UPD     EVENT TYPES 4 THRU 8 (ALL UP/DOWNS)        *
C*                EX2BFL     EVENT TYPE 9 (BATTLEF'D ILLUM'N)           *
C*                EX2SRP     EVENT TYPE 10 (REPOSITION SENSOR)          *
C*                EX2HLT     EVENT TYPE 99 (PROGRAM STOP, EX2HLT        *
C*                             CLEARS TEMPORARY DATA STORES BE-         *
```

Figure 2.2-6

```
C*                                 FORE EXECUTING ABSOLUTE STOP)           *
C*              EX2SNP           HANDLES AN INTERNALLY-GENERATED EVENT    *
C*                                 CALLED "SNAPSHOT"... THAT PRODUCES      *
C*                                 A PRINTED SUMMARY OF ACTIVE SYSTEM      *
C*                                 ELEMENTS.                               *
C*                                                                         *
C*       ALSO CALLED                                                       *
C*                                                                         *
C*              PGSKP2           SPECIAL PAGE SKIP                         *
C*              TRNSFR           BLOCK TRANSFER OF DATA                    *
C*              TIMOUT           CONVERTS INTERNAL TIME TO EXTERNAL       *
C*                                 FORM, SUITABLE FOR PRINTOUT             *
C*                                                                         *
C**************************************************************************
```

Figure 2.2-6 (Cont.)

2.2.5.2 MSM Executive Routines -- Level 2

There are eight level 2 executive routines, all called by EXECIB. Each is responsible for a particular event type or (in the case of EX2UPD) a class of event types. Table 2.2-V summarizes the functions of these eight routines, and their correspondences to event-type code numbers.

Of the eight level 2 routines, only EX2SNR (which has general control of sensor interrogation events) places calls to level 3 executive routines, discussed in Section 2.2.5.3. All of these routines, of course, do depend upon various auxiliary routines.

Figs. 2.2-7 through 2.2-14 provide additional detail on the level 2 routines, including their direct use of auxiliary routines. Reference is also made to Section 2.2.3 (this Volume) and Appendix C, which discuss and illustrate EX2HLT results in the context of output onto the printer.

```
C*************************************  EX2SNR  ******************************
C*                                                                            *
C*                           SUBROUTINE EX2SNR                                *
C*                                                                            *
C*    PURPOSE                                                                 *
C*        LEVEL 2 EXEC FOR MSM, TO HANDLE TYPE 1 EVENTS (SENSOR               *
C*        INTERROGATE).  ACTUAL EXECUTION OF THESE EVENTS DEPENDS             *
C*        UPON SENSOR TYPE.  EX2SNR DETERMINES THIS TYPE AND                  *
C*        TRANSFERS CONTROL TO THE PROPER EX3--- ROUTINE.                     *
C*                                                                            *
C*    CALLING SEQUENCE                                                        *
C*        CALL EX2SNR (LIST)                                                  *
C*            (CALLED ONLY BY EXEC18)                                         *
C*                                                                            *
C*    DESCRIPTION OF PARAMETERS                                               *
C*        LIST     WILL BE AN EVENT TYPE 1 SUBLIST.  IN PARTICU-              *
C*                 LAR, LIST (3) WILL BE THE SENSOR ID.                       *
C*                                                                            *
C*                 IMPLICIT PARAMETERS FROM LABELLED COMMON.                  *
C*                                                                            *
C*    REMARKS                                                                 *
C*        NO CHECKS FOR CALLING ERRORS                                        *
C*                                                                            *
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                           *
C*        PARPTR     GETS POINTER FOR SENSOR PARAM SUBLIST                    *
C*        EX3SAC     LEVEL 3 ROUTINE FOR SEISMIC (2ND ARG = 1) AND            *
C*                      ACOUSTIC (2ND ARG = 2) SENSORS                        *
C*        EX3MAG     LEVEL 3 ROUTINE FOR MAGNETIC SENSORS                     *
C*        EX3ARF        ''   3  ''      ''  AREBUOY     ''                    *
C*        EX3PIR        ''   3  ''      ''  PASSIVIR    ''                    *
C*        EX3RDR        ''   3  ''      ''  RADAR       ''                    *
C*        EX3IMG        ''   3  ''      ''  IMAGE       ''                    *
C*        EX3THV        ''   3  ''      ''  THERMVEW    ''                    *
C*        EX3BKW        ''   3  ''      ''  BREAKWIR    ''                    *
C*                                                                            *
C******************************************************************************
```

Figure 2.2-7

```
C******************************  EX2SFA  ******************************
C*                                                                    *
C*                         SUBROUTINE EX2SFA                          *
C*                                                                    *
C*    PURPOSE                                                         *
C*        MSM LEVEL 2 EXECUTIVE ROUTINE.  HANDLES INPUT 'EVENTS'      *
C*        OF TYPE 2,  SENSOR FALSE ALARMS. CALLED ONLY BY EXEC1B.     *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL EX2SFA (LIST)                                          *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        LIST      AN EVENT-TYPE-2 SUBLIST.  THE 3'RD WORD GIVES     *
C*                  THE ID OF THE SENSOR,  THE 4'TH WORD GIVES        *
C*                  THE (INTEGER, GAME-) TIME THAT FALSE ALARM IS     *
C*                  TO BE ENTERED INTO SIMULATION.                    *
C*                                                                    *
C*    REMARKS                                                         *
C*        THERE ARE TWO BASIC TASKS WITHIN EX2SFA.  FIRST, A          *
C*        SENSOR OUTPUT REPORT MUST BE SENT TO THE OUTPUT ROUTINE     *
C*        UGSOUT (WITH PARAMETERS INDICATING FALSE ALARM TYPE OF      *
C*        REPORT).  SECOND, ANY SENSOR PARAMETERS INFLUENCED BY       *
C*        A FALSE ALARM MUST BE CHANGED.                              *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        PARPTR    MSM UTILITY ROUTINE.  POINTER DETERMINATION.      *
C*        TRNSFR    BLOCK TRANSFER OF DATA                            *
C*        UGSOUT    OUTPUT ROUTINE FOR UGS SENSORS REPORTS            *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-8

```
C****************************** EX2SPC ********************************
C*                                                                    *
C*                      SUBROUTINE EX2SPC                             *
C*                                                                    *
C*    PURPOSE                                                         *
C*        LEVEL 2 EXECUTIVE ROUTINE FOR MSM.  HANDLES EVENTS OF       *
C*        TYPE 3 (SENSOR PARAMETER CHANGE).  CALLED ONLY BY EXEC1B.   *
C*                                                                    *
C*    USAGE                                                           *
C*        CALL EX2SPC (LIST)                                          *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        LIST      AN EVENT TYPE 3 SUBLIST...                        *
C*                     WORD 3        ID OF SENSOR FOR WHICH NEW PAR-  *
C*                                   AMETER VALUES ARE BEING ENTERED  *
C*                     WORD 5        NUMBER OF PARAMETERS TO BE CHANGED *
C*                     WORDS 6,...   NEW VALUES OF PARAMETERS         *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        PARPTR    UTILITY ROUTINE TO DETERMINE POINTER TO PARAM-    *
C*                  ETER LOCATIONS IN STORAGE AREA MSMPAR. IN THIS    *
C*                  PROGRAM, ARGUMENTS ARE                            *
C*                     1ST = 6 (SENSOR PARAMETERS IN MSM6)            *
C*                     2ND = ID OF SENSOR                             *
C*                     3RD = 16 (PARAMETERS THAT CAN BE CHANGED BY    *
C*                               A TYPE 3 EVENT ALWAYS BEGIN IN       *
C*                               WORD 16 OF A SENSOR DATA SET)        *
C*        TRNSFR    BLOCK TRANSFER. IN THIS PROGRAM, TRNSFR MOVES     *
C*                  NPARCH WORDS INTO MSMPAR STORAGE FROM INPUT LIST. *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-9

```
C******************************  EX2UPD  ******************************
C*                                                                     *
C*                        SUBROUTINE EX2UPD                            *
C*                                                                     *
C*    PURPOSE                                                          *
C*        LEVEL 2 EXEC FOR MSM.  CONTROLS UP/DOWN INDICATORS           *
C*        AND/OR PRINTED OUTPUT FOR SENSORS, MONITORS, DATA            *
C*        LINKS, FIRETRAPS AND ARRAYS.  CALLED BY EXEC18.              *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL EX2UPD (LIST)                                           *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        LIST    WILL BE AN EVENT SUBLIST CORRESPONDING TO            *
C*                   EVENT TYPE 4, 5, 6, 7 OR 8  (ALL UP-DOWN          *
C*                   LOGIC-- EVENT TYPE NO. INDICATES WHAT TYPE        *
C*                   OF SYSTEM ELEMENT EVENT APPLIES TO).              *
C*                                                                     *
C*    REMARKS                                                          *
C*        LOGICAL OPERATION VS. EVENT TYPE INDICATED BELOW             *
C*                                                                     *
C*             EVENT TYPE   ELEMENT   UP/DOWN FLAG     PRINTED         *
C*               NUMBER      TYPE     SET IN STORAGE   MESSAGE         *
C*        FOR THE EVENTS THAT GET PRINTED, A BINARY OUTPUT ON TAPE     *
C*        (OR DISK) IS ALSO GENERATED (OUTPUT EVENTS 10, 11 RESP.)     *
C*                                                                     *
C*                 4         SENSOR          YES            NO         *
C*                 5         MONITOR         YES            NO         *
C*                 6         DATALINK        YES            NO         *
C*                 7         FIRETRAP        YES           YES         *
C*                 8         ARRAY            NO           YES         *
C*                                                                     *
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        PARPTR     PROVIDES POINTER LOCATING SPECIFIED WORD IN       *
C*                      THE MSMPAR ARRAY                               *
C*        TIMOUT     CONVERTS INTERNAL TIME VARIABLE (INTEGER NO.      *
C*                      OF SECONDS INTO GAME) INTO DAY, CLOCK AND      *
C*                      SEC VARIABLES FOR 'EXTERNAL' TIME, IN FORM     *
C*                      SUITABLE FOR PRINT WITH I-FORMAT.              *
C*        PGSKP2     SPECIAL PAGE SKIP CONTROL                         *
C*                                                                     *
C*    METHOD  (COMMENT)                                                *
C*        THE UP/DOWN FLAG IN STORAGE IS ALWAYS INITIALIZED TO THE     *
C*        VALUE 0 (DOWN).  FOR AN UP/DOWN EVENT, HOWEVER, THIS FLAG    *
C*        WILL BE SET TO  +TIME (INTEGER FORM) FOR AN UP CONTROL,      *
C*        TO  -TIME FOR DOWN CONTROL... WITH APPROPRIATE GUARD FOR     *
C*        THE TIME = 0 SPECIAL CASE.  THUS,THIS FLAG CARRIES           *
C*        TIME-UP OR TIME-DOWN INFORMATION AS WELL AS UP/DOWN          *
C*        STATUS.                                                      *
C*                                                                     *
```

Figure 2.2-10

```
C******************************  EX2BFL  ******************************
C*                                                                   *
C*                         SUBROUTINE EX2BFL                         *
C*                        (BATTLEFIELD LIGHT)                        *
C*                                                                   *
C*   PURPOSE                                                         *
C*       MSM LEVEL 2 EXECUTIVE ROUTINE, CALLED ONLY BY EXEC1B.       *
C*       EX2BFL SUPERVISES STORAGE OF DATA FOR EVENT TYPE 9          *
C*       (BATTLE FIELD ILLUMINATION BY FLARES, ETC.), THAT WILL      *
C*       LATER BE USED BY NIGHTTIME-IMAGE-DEVICE ROUTINES.           *
C*                                                                   *
C*   CALLING SEQUENCE                                                *
C*       CALL EX2BFL (LIST)                                          *
C*                                                                   *
C*   DESCRIPTION OF PARAMETERS                                       *
C*       LIST      A TYPE 9 EVENT SUBLIST OF FOLLOWING CONTENT-      *
C*                                                                   *
C*                       WORD                                        *
C*                         1    ITYPEV  EVENT TYPE CODE (=9)         *
C*                         2    LNGTHS  LIST LENGTH (=10)            *
C*                         3    IDILDV  ALPHANUMERIC 1-WORD DEVICE ID *
C*                         4    ITIM1   INTEGER GAME TIME, ILLUM BEGIN *
C*                         5    ITIM2   INTEGER GAME TIME, ILLUM END  *
C*                         6    X       X- GAME COORDINATE           *
C*                         7    Y       Y- GAME COORDENATE           *
C*                         8    H       HEIGHT ABOVE GROUND AT ITIM1 *
C*                         9    AINTNS  INTENSITY OF LIGHT SOURCE    *
C*                        10    MODE    MODE-OF-ILLUMINATION         *
C*                                                                   *
C*   REMARKS                                                         *
C*       1. ONLY THE LAST 8 WORDS ARE STORED.                        *
C*                                                                   *
C*       2. WORD 3 (IDILDV) CAN BE AN ARBITRARY 1-WORD (4-CHARACTER) *
C*          IDENTIFICATION. NOT USED FOR COMPUTATION.                *
C*                                                                   *
C*       3. STORAGE IS RESERVED FOR UP TO 20 SIMULTANEOUSLY ACTIVE   *
C*          ILLUMINATION EVENTS.  IF MORE THAN 20 ARE PRESENTED      *
C*          (ALL ACTIVE), THE NEWEST REPLACES (ERASES) THAT EVENT    *
C*          NEAREST TO EXPIRATION, AND A DIAGNOSTIC MESSAGE IS       *
C*          PRINTED.                                                 *
C*                                                                   *
C*   SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*       TRNSFR     BLOCK TRANSFER OF DATA                           *
C*       PGSKIP     PAGE SKIP CONTROL                                *
C*                                                                   *
C*********************************************************************
```

Figure 2.2-11

```
C******************************  EX2SRP  *******************************
C*                                                                     *
C*                        SUBROUTINE EX2SRP                            *
C*                      (SENSOR REPOSITION)                            *
C*                                                                     *
C*   PURPOSE                                                           *
C*       MSM LEVEL 2 EXECUTIVE ROUTINE, TO HANDLE EVENTS TYPE 10       *
C*       (SENSOR REPOSITION).  SEE 'REMARKS' BELOW.                    *
C*                                                                     *
C*   CALLING SEQUENCE                                                  *
C*       CALL EX2SRP (LIST)                                            *
C*                                                                     *
C*   DESCRIPTION OF PARAMETERS                                         *
C*       LIST       WILL BE A 6-WORD SUBLIST CORRESPONDING TO          *
C*                  EVENT TYPE 10:                                     *
C*                                                                     *
C*                      WORD                                           *
C*                        1   ITYPEV   EVENT TYPE CODE (=10)           *
C*                        2   LNGTHS   LIST LENGTH (=6)                *
C*                        3   IDSNR   ID OF SENSOR                     *
C*                        4   ITIM     INTEGER GAME TIME (SECONDS)     *
C*                                       THIS EVENT TO OCCUR           *
C*                        5   X        NEW COORDINATES (REAL,METERS,   *
C*                        6   Y          GAME VALUES) FOR SENSOR       *
C*                                                                     *
C*   REMARKS                                                           *
C*       1. THIS ROUTINE CALLED ONLY BY EXEC1B.                        *
C*                                                                     *
C*       2. THIS ROUTINE CALLED ONLY WHEN A STATIONARY SENSOR          *
C*          IS REEMPLACED, IN WHICH CASE POSITIONING ERRORS            *
C*          ARE LIKELY TO CAUSE DEVIATIONS FROM THE ORIGINAL           *
C*          X,Y COORDINATES.                                           *
C*                                                                     *
C*       3. NOTE IN PARTICULAR THAT  (A) THIS ROUTINE HAS NO           *
C*          BEARING ON MOVING SENSORS, AND (B) IT IS NOT USED          *
C*          FOR THE INITIAL COORDINATES VALUES OF A SENSOR             *
C*                                                                     *
C*          THUS, EX2SRP CANNOT BE CALLED IN A SIMULATION UNLESS       *
C*          MULTIPLE EMPLACEMENTS OF STATIONARY SENSORS OCCUR,         *
C*                                                                     *
C*   SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*       PARPTR     UTILITY ROUTINE TO LOCATE SYSTEM PARAMETERS        *
C*                    IN STORAGE                                       *
C*       TRNSFR     BLOCK TRANSFER OF DATA                             *
C*                                                                     *
C***********************************************************************
```

Figure 2.2-12

```
C****************************** EX2SNP ***********************************
C*                                                                      *
C*                         SUBROUTINE EX2SNP                            *
C*                       (SYSTEM 'SNAPSHOT')                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*       PROVIDES A PRINTED 'SNAPSHOT' OF SYSTEM ELEMENTS (SENSORS,     *
C*       ARRAYS) ACTIVE AT A FIXED INSTANT OF TIME.                     *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*       CALL EX2SNP (ITSNAP)                                           *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*       ITSNAP     INTEGER GAME TIME AT WHICH SNAPSHOT APPLIES         *
C*                                                                      *
C*          OUTPUTS (PRINTED) AS FOLLOWS, FOR ACTIVE SENSORS ONLY:      *
C*          A. FOR STATIONARY SENSORS/ARRAYS                            *
C*                SENSOR ID                                             *
C*                SENSOR GENERIC TYPE                                   *
C*                ARRAY ID                                              *
C*                GAME X,Y COORDINATES                                  *
C*          B. FOR MOVING SENSORS/ARRAYS                                *
C*                SENSOR ID                                             *
C*                SENSOR GENERIC TYPE                                   *
C*                ARRAY ID                                              *
C*                ID OF ROUTE                                           *
C*                ID OF ASSOCIATED BLUE FORCE                           *
C*                GROUND VS. AIRBORNE NOTATION                          *
C*                                                                      *
C*    REMARKS                                                           *
C*       1. THIS ROUTINE REGARDED AS A LEVEL 2 MSM EXECUTIVE            *
C*          ROUTINE.  CALLED ONLY BY EXEC1B.                            *
C*                                                                      *
C*       2. PRINTED RESULTS BASED ON UP/DOWN FLAGS, SENSOR BY           *
C*          SENSOR.  THUS, IT IS POSSIBLE FOR SOME BUT NOT ALL          *
C*          SENSORS IN SAME ARRAY TO APPEAR IN OUTPUT LIST.             *
C*                                                                      *
C*       3. THE ROUTINE EX2SNP DOES NOT INTERACT WITH BASIC             *
C*          SIMULATION FLOW.  THUS, THIS VERSION CAN BE REVISED         *
C*          FREELY, ACCORDING TO USERS' EVALUATION OF CONTENT,          *
C*          WITHOUT EFFECT ON OTHER PROGRAMS.  FOR PRODUCTION           *
C*          RUNNING, A DUMMY VERSION (NO OPERATIONS AT ALL)             *
C*          MAY BE DESIRABLE.                                           *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*       TIMOUT     CONVERTS GAME TIME INTO DAY, CLOCK VALUES           *
C*       PGSKIP     PAGE SKIP CONTROL                                   *
C*       TGTVLU     USED HERE TO GET IDPATH FOR BLUE FORCE              *
C*                                                                      *
C************************************************************************
```

Figure 2.2-13

```
C******************************** EX2HLT *********************************
C*                                                                      *
C*                       SUBROUTINE EX2HLT                              *
C*                                                                      *
C*    PURPOSE                                                           *
C*        MSM LEVEL 2 EXEC ROUTINE, CALLED BY EXEC1B TO SUMMARIZE       *
C*        SIMULATION RESULTS, THEN TERMINATE MSM PROCESSING.            *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL EX2HLT                                                   *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*        EX2HLT USES SYSTEM COUNTER VALUES STORED IN                   *
C*        COMMON /SYSCNT/.                                              *
C*                                                                      *
C*    REMARKS                                                           *
C*        IN ADDITION TO SUMMARY TABLES DERIVED FROM SYSTEM COUNTERS,   *
C*        THE FINAL REFERENCE VALUES FOR SYSTEM RANDOM NUMBER GEN-      *
C*        ERATORS (UNIFORM AND GAUSSIAN) ARE PRINTED.  THESE VALUES     *
C*        MAY BE USED AS INPUT (ON 'HEADER CARDS') FOR SUBSEQUENT       *
C*        SIMULATION RUN IF DESIRED (ASSURES NEW SEQUENCE OF RANDOM     *
C*        NUMBERS).                                                     *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*        ERASE     CLEARS BLOCK OF STORAGE                             *
C*        PGSKIP    PAGE SKIP CONTROL                                   *
C*        URNASK    OBTAINS INTEGER REFFRENCE STATE FOR                 *
C*                    UNIFORM RANDOM NUMBER GENERATOR                   *
C*        GRNASK    SAME, BUT FOR GAUSSIAN RANDOM NUMBER                *
C*                    GENERATOR                                         *
C*                                                                      *
C************************************************************************
```

Figure 2.2-14

Table 2.2-V

SUMMARY OF LEVEL 2 EXECUTIVE ROUTINES

NAME	EVENT*	PROGRAM PURPOSE AND COMMENTS
EX2SNR	1	Sensor interrogate; potential sensor response to one or more targets.
EX2SFA	2	Sensor false alarm (UGS sensors only). Game play: sensor reports to monitor. Game truth: output program notified that response is false alarm.
EX2SPC	3	Sensor parameter change. Certain physical parameters (gain, for example) depend upon background noise due to atmospheric, cultural and battle effects. These parameters, computed in PRERUN, are passed to MSM as event type 3 in order that MSM can alter these parameters in storage.
EX2UPD	4-8	Up/down (or start/stop) status control for sensors, for monitors, for data links, for arrays, and for firetrap operations. Flags in storage are set and/or printed messages are prepared.
EX2BFL	9	Battlefield lighting. Position, parameters and start and stop times for battlefield illumination source are placed into storage.
EX2SRP	10	Sensor reposition: change of coordinate values upon reemplacement of sensor. (Sensor coordinates for first emplacement are not entered by an event type 10).
EX2SNP	**	System "snapshot." A printed summary of all active sensors is prepared for the instant of time specified.
EX2HLT	99	"HALT" routine. End of event data from PRERUN has been reached. EX2HLT prepares a system results summary based on about 128 system counters of basic events and results; prints other terminal data; and enters an end-of-data indicator on the MSM binary output device... then causes a program STOP to be executed.

* Type codes for PRERUN-generated input events, corresponding to the tasks assigned to the level 2 exec routines.

** Event not from PRERUN; no number assigned. Event generated internally within EXEC1B.

2.2.5.3 MSM Executive Routines -- Level 3

The heart of MSM simulation is the collection of EX3... executive routines and associated sensor routines. Here is the center of computation for the primary simulation event,-- sensor responses to one or more targets. Here, also, is the center of MSM "busyness" and activity, in the sense that most of the demands for auxiliary subroutine support originate at this level.

The eight level 3 executive routines are called only by EX2SNR. There is an EX3... subroutine for each of the different generic sensor types, except that EX3SAC controls both SEISTG and ACOUTG (seismic and acoustic sensor routines) that have nearly identical control requirements and internal logic.

Although represented as executive routines, the EX3... programs enter into MSM at a level where the distinction between "control" and "computation" becomes indistinct. In fact, an EX3... routine exhibits some of the characteristics of a sensor routing per se. An EX3... routine also acts as an extensive interface routine, that links together the: (a) sensor routines, (b) executive commands from EX2SNR, (c) data from variable length storage arrays, (d) auxiliary computational routines and (e) output routines.

Comments on the eight individual EX3... routines are given in Figs. 2.2-15 through 2.2-22 respectively. Because of the direct relationships between these routines and the sensor routines, the following Section 2.2.5.4 is also relevant.

```
C******************************** EX3SAC ********************************
C*                                                                      *
C*                         SUBROUTINE EX3SAC                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*        MSM LEVEL 3 EXEC.  HANDLES DETAILS OF TYPE 1 EVENTS           *
C*        (SENSOR INTERROGATE) FOR SEISMIC AND ACOUSTIC SENSORS.        *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL EX3SAC (LIST,ISORA)                                      *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*        LIST       A TYPE-1-EVENT SUBLIST,  CONTAINING SENSOR ID,     *
C*                     TIME INFORMATION, TARGET ID'S.                   *
C*        ISORA      INDICATOR, SEISMIC OR ACOUSTIC... VALUES 1 OR      *
C*                     2, RESPECTIVELY.                                 *
C*                                                                      *
C*        IMPLICIT PARAMETERS (FOR SENSOR AND TARGETS) FROM COMMON.     *
C*                                                                      *
C*    REMARKS                                                           *
C*        1. THIS SUBROUTINE CALLED ONLY BY EX2SNR.                     *
C*        2. THIS SUBROUTINE IS ONE OF 8 IN THE EX3... SERIES. IT       *
C*             IS THE ONLY ONE TO HAVE TWO EXPLICIT ARGUMENTS, AND      *
C*             RESPONSIBILITY FOR TWO SENSOR GENERIC TYPES --           *
C*             A CONSEQUENCE OF THE NEARLY IDENTICAL LOGICAL            *
C*             CHARACTERISTICS OF SEISMIC AND ACOUSTIC SENSOR           *
C*             ROUTINES.                                                *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*        SEISTG     SUBROUTINE FOR SEISMIC SENSOR RESPONSES            *
C*                     TO TARGETS.                                      *
C*        ACOUTG     SUBROUTINE FOR ACOUSTIC SENSOR RESPONSES           *
C*                     TO TARGETS.                                      *
C*        ITODEV     EVALUATES TIME OF DAY FROM ITIME                   *
C*        ERASE      BLOCK ERASE (SET TO 0)                             *
C*        PARPTR     UTILITY ROUTINE TO LOCATE PARAMS IN COMMON         *
C*        TGTPTR     UTILITY ROUTINE TO LOCATE TARGET PARAMS            *
C*        PARVLU     UTILITY ROUTINE TO GET PARAM VALUES                *
C*        TRNSFR     UTILITY ROUTINE FOR BLOCK TRANSFER OF DATA         *
C*        TGTLXY     UTILITY ROUTINE TO LOCATE X,Y POSITION OF TARGET,  *
C*                     NO. OF ELEMENTS ON A LEG, SPEED OF TARGET.       *
C*        FTPAR1     ROUTINES TO DERIVE STATIC AND DYNAMIC PAR-         *
C*        FTPAR2        AMETERS, RESP., FOR FALSE TARGETS               *
C*        SACDET     INTERFACE TO OUTPUT (FOR UGS SENSOR DETECTIONS)    *
C*                                                                      *
C************************************************************************
```

Figure 2.2-15

```
C******************************  FX3MAG  ******************************
C*                                                                    *
C*                         SUBROUTINE FX3MAG                          *
C*                                                                    *
C*    PURPOSE                                                         *
C*       MSM LEVEL 3 EXEC. HANDLES DETAILS OF TYPE 1 EVENTS           *
C*       (SENSOR INTERROGATE) FOR MAGNETIC SENSOR                     *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*       CALL FX3MAG (LIST)                                           *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*       LIST      A TYPE-1-EVENT SUBLIST,  CONTAINING SENSOR ID,     *
C*                   TIME INFORMATION, TARGET ID'S.                   *
C*       IMPLICIT PARAMETERS (FOR SENSOR AND TARGETS) FROM COMMON.    *
C*                                                                    *
C*    REMARKS                                                         *
C*       1. THIS SUBROUTINE CALLED ONLY BY EX2SNR.                    *
C*                                                                    *
C*    SUBROUTINE  AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*       ERASE     BLOCK ERASE (SET TO 0)                             *
C*       CLOSEL    GET ELEMENT OF A TGT CLOSEST TO SENSOR             *
C*       FTPAR1    ROUTINES TO DERIVE STATIC AND DYNAMIC PAR-         *
C*       FTPAR2       AMETERS, RESP., FOR FALSE TARGETS               *
C*       MAGTG     SUBROUTINE FOR MAGNETIC SENSOR RESPONSES           *
C*                   TO TARGETS                                       *
C*       PARPTR    UTILITY ROUTINE TO LOCATE PARAMS IN COMMON         *
C*       TRNSFR    UTILITY ROUTINE FOR BLOCK TRANSFER OF DATA         *
C*       TGTPTR    UTILITY ROUTINE TO LOCATE TARGET PARAMS            *
C*       TGTLXY    UTILITY ROUTINE TO LOCATE X,Y POSITION OF TARGET,  *
C*                   NO. OF ELEMENTS ON A LEG, SPEED OF TARGET.       *
C*       UGSDET    INTERFACE TO OUTPUT (FOR UGS SENSOR DETECTIONS)    *
C*       URNASK    SAVES THE REF. NO. OF THE URN GENERATOR.           *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-16

```
C******************************  EX3ARF  ******************************
C*                                                                    *
C*                       SUBROUTINE EX3ARF                            *
C*                                                                    *
C*    PURPOSE                                                         *
C*        MSM LEVEL 3 EXEC. HANDLES DETAILS OF TYPE 1 EVENTS          *
C*        (SENSOR INTERROGATE) FOR AREBUOY SENSOR FIELD               *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL EX3ARF (LIST)                                          *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        LIST       A TYPE-1-EVENT SUBLIST,  CONTAINING SENSOR ID,   *
C*                      TIME INFORMATION, TARGET ID'S.                *
C*        IMPLICIT PARAMETERS (FOR SENSOR AND TARGETS) FROM COMMON.   *
C*                                                                    *
C*    REMARKS                                                         *
C*        1. THIS SUBROUTINE CALLED ONLY BY EX2SNP.                   *
C*                                                                    *
C*    SUBROUTINE  AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*        ERASE      BLOCK ERASE (SET TO 0)                           *
C*        FTPAR1     ROUTINES TO DERIVE STATIC AND DYNAMIC PAR-       *
C*        FTPAR2        AMETERS, RESP., FOR FALSE TARGETS             *
C*        ARFTG      SUBROUTINE FOR AREBUOY SENSOR FIELD RESPONSES    *
C*                      TO TARGETS                                    *
C*        NVNEED     NO. OF ELEMENTS IN AN AREBUOY FIELD              *
C*        PARPTR     UTILITY ROUTINE TO LOCATE PARAMS IN COMMON       *
C*        TRNSFR     UTILITY ROUTINE FOR BLOCK TRANSFER OF DATA       *
C*        TGTPTR     UTILITY ROUTINE TO LOCATE TARGET PARAMS          *
C*        UGSDET     INTERFACE TO OUTPUT (FOR UGS SENSOR DETECTIONS)  *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-17

```
C*****************************  EX3PIR  ******************************
C*                                                                   *
C*                      SUBROUTINE EX3PIR                            *
C*                                                                   *
C*    PURPOSE                                                        *
C*        MSM LEVEL 3 EXEC. HANDLES DETAILS OF TYPE 1 EVENTS         *
C*        (SENSOR INTERROGATE) FOR PASSIVE IR SENSORS                *
C*                                                                   *
C*    CALLING SEQUENCE                                               *
C*        CALL EX3PIR (LIST)                                         *
C*                                                                   *
C*    DESCRIPTION OF PARAMETERS                                      *
C*        LIST      A TYPE-1-EVENT SUBLIST,  CONTAINING SENSOR ID,   *
C*                    TIME INFORMATION, TARGET ID'S.                 *
C*        IMPLICIT PARAMETERS (FOR SENSOR AND TARGETS) FROM COMMON.  *
C*                                                                   *
C*    REMARKS                                                        *
C*        1. THIS SUBROUTINE CALLED ONLY BY EX2SNR.                  *
C*                                                                   *
C*    SUBROUTINE  AND FUNCTION SUBPROGRAMS REQUIRED                  *
C*        ERASE     BLOCK ERASE (SET TO 0)                           *
C*        URN       UNIFORM RANDOM NUMBER GENERATOR                  *
C*        GRN       GAUSSIAN RANDOM NUMBER GENERATOR                 *
C*        FTPAR1    ROUTINES TO DERIVE STATIC AND DYNAMIC PAR-       *
C*        FTPAR2       AMETERS, RESP., FOR FALSE TARGETS             *
C*        PIRTG     SUBROUTINE FOR PASSIVE IR DEVICE RESPONSES       *
C*                    TO TARGETS                                     *
C*        PARPTR    UTILITY ROUTINE TO LOCATE PARAMS IN COMMON       *
C*        TRNSFR    UTILITY ROUTINE FOR BLOCK TRANSFER OF DATA       *
C*        TGTPTR    UTILITY ROUTINE TO LOCATE TARGET PARAMS          *
C*        PARVLU    UTILITY ROUTINE TO GET PARAM VALUES              *
C*        UGSDET    INTERFACE TO OUTPUT (FOR UGS SENSOR DETECTIONS)  *
C*        TGTLG     UTILITY ROUTINE FOR TARGET/LEG CALC'NS           *
C*                                                                   *
C*********************************************************************
```

Figure 2.2-18

```
C******************************  EX3RDR  ******************************
C*                                                                    *
C*                      SUBROUTINE EX3RDR                             *
C*                                                                    *
C*    PURPOSE                                                         *
C*        MSM LEVEL 3 EXECUTIVE ROUTINE, SUPERVISING CALLS TO         *
C*        SENSOR SUBROUTINE RADAR, AND PROVIDING CONTROL AND          *
C*        INTERFACE WITH INPUT EVENTS, OUTPUT ROUTINES, STOR-         *
C*        AGE, AND SUCH RELATED ROUTINES AS SCAN1 AND SCAN2           *
C*        IN THE RADAR CONTEXT.                                       *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL EX3RDR (LIST)                                          *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*                                                                    *
C*        LIST      IS AN EXPLICIT INPUT ARRAY, CORRESPONDING         *
C*                  TO AN EVENT TYPE 1 SUBLIST...                     *
C*                        WORD(S)                                     *
C*                          1        ITYPEV  (= 1)                    *
C*                          2        LNGTHS  (LENGTH OF LIST)         *
C*                          3        IDSNSR                           *
C*                          4        ITIM1    (BEGIN AND END TIME, INT-  *
C*                          5        ITIM2     EGER, OF EVENT INTERVAL)  *
C*                          6        NRBTGS  (NUMBER OF RED/BLUE       *
C*                                             TARGETS)                *
C*                        (7 THRU             (TARGET ID/LEG CODES FOR  *
C*                          6+NRBTGS)           RED/BLUE TARGETS. CODE, *
C*                                             DENOTED BY IDTL, IS OF  *
C*                                             FORM ID + 1000*LEGNO.)  *
C*                         (NEXT)             IF FALSE TARGETS EXIST,  *
C*                                              THIS WORD SPECIFIES    *
C*                                              NFTGTS (NO. OF FALSE   *
C*                                              TARGETS).              *
C*                        (REMAINING)         PARAMETERS FOR FALSE TGTS *
C*                                             ARE PASSED IN THE EVENT *
C*                                             LIST PER SE, WITH NUMBER *
C*                                             OF WORDS PER FALSE TGT  *
C*                                             = 12. FIRST WORD FOR A  *
C*                                             GIVEN FALSE TARGET IS A *
C*                                             TYPE CODE (EVID) FORMED *
C*                                             IN BATTLE OR CULTURE    *
C*                                             ROUTINES.               *
C*                                                                    *
C*    REMARKS                                                         *
C*        1. IF DETECTION OCCURS, THE OUTPUT ROUTINE SCNOUT IS        *
C*           CALLED.  INFORMATION IS PASSED BY EXPLICIT ARGUMENTS     *
C*           AND VIA COMMON AREAS /SENSOR/, /TARGET/ AND /TGLGCM/.    *
C*                                                                    *
C*        2.  PARAMETERS FOR RED/BLUE TARGETS ARE OBTAINED FROM       *
```

Figure 2.2-19

```
C*              STORAGE ARRAY MSMPAR, VIA UTILITY ROUTINES IF NEC-    *
C*              ESSARY.  PARAMETERS FOR FALSE TARGETS ARE ACCESSED    *
C*              OR DERIVED FROM THE INPUT EVENT LIST.                 *
C*                                                                    *
C*          3.  STORAGE SPACE AVAILABLE IN COMMON /TARGET/ LIMITS     *
C*              TOTAL NUMBER OF TARGETS (TARGET/LEGS) THAT CAN BE     *
C*              SIMULTANEOUSLY HANDLED TO 16.  IF MORE THAN 16 ARE    *
C*              INDICATED IN THE INPUT LIST, THEN (A) THE TOTAL       *
C*              NUMBER IS TRUNCATED TO 16, WITH PRIORITY TO RED/      *
C*              BLUE TARGETS, AND (B) A DIAGNOSTIC MESSAGE IS         *
C*              PRINTED TO INDICATE LOSS OF TARGETS                   *
C*                                                                    *
C*          4.  PARTIAL PROTECTION AGAINST INCONSISTENT DATA IS       *
C*              PROVIDED, WITH DIAGNOSTIC MESSAGES PRINTED.           *
C*                                                                    *
C*          5.  STORAGE CONTROL IS RELATIVELY COMPLICATED. IN AD-     *
C*              DITION TO THE OBVIOUS COMMON AREAS /SENSOR/ AND       *
C*              /TARGET/, 50 AND 210 WORDS RESPECTIVELY, ADDIT-       -
C*              IONAL STORAGE FOR (TARGET) PARAMETERS EFFECTIVELY     *
C*              IS PROVIDED BY COMMON /TGLGCM/ AS WELL AS THE EVENT   *
C*              LIST.  USE OF /TGLGCM/ INVOLVES KNOWLEDGE OF THE      *
C*              INTERNAL OPERATIONS OF SUBROUTINES TGTLG AND TGTLXY.  *
C*                                                                    *
C*          6.  THIS ROUTINE HANDLES BOTH MOVING RECTANGLE AND SECTOR *
C*              SCAN COVERAGE MODES.  IT DOES NOT HANDLE STATIONARY   *
C*              RECTANGLE OR SECTOR MODES.  TWO TYPES OF SECTOR SCAN  *
C*              LOGICS ARE PROVIDED, CORRESPONDING TO SUBROUTINES     *
C*              SCAN1 AND SCAN2.                                      *
C*                                                                    *
C*          7.  IF RADAR IS MOVING, IT IS ASSUMED TO BE AIRBORNE.     *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION TYPE SUBPROGRAMS REQUIRED               *
C*        (IN ORDER OF OCCURRENCE IN PROGRAM)                         *
C*                                                                    *
C*          ERASE     BLOCK ERASE (SET TO 0)                          *
C*          TRNSFR    BLOCK TRANSFER OF DATA                          *
C*          PARPTR    MSM UTILITY, TO LOCATE PARAMS IN STORAGE        *
C*          TGTPTR    MSM UTILITY, FOR POINTER DETERMINATION          *
C*          TGTVLU    MSM UTILITY. GETS VALUE OF A RED OR BLUE        *
C*                       TARGET (FORCES) PARAMETER                    *
C*          KSTVLU    EVALUATES 'KSTRNG' VARIABLES FOR TARGET         *
C*          FTPAR1    DERIVES AND STORES STATIC FALSE TARGET PAR-     *
C*                       AMETERS                                      *
C*          TGTLG     BASIC TARGET/LEG SUBROUTINE                     *
C*          SETSC1    SETS WORKING PARAMETERS FOR SCAN1               *
C*          SETSC2    SETS WORKING PARAMETERS FOR SCAN2               *
C*          TGTLXY    GETS TARGET/LEG PARAMETERS THAT CHANGE WITH TIME *
C*          GRN       GAUSSIAN RANDOM NUMBER GENERATOR                *
C*          FTPAR2    DERIVES AND STORES DYNAMIC FALSE TARGET PARAMS. *
C*          SCAN1     THE TWO BASIC SECTOR SCAN ROUTINES (NOT CALLED  *
```

Figure 2.2-19 (Cont.)

```
C*        SCAN2      FOR MOVING RECTANGLE COVERAGE)                        *
C*        RADAR     SENSOR SUBROUTINE                                      *
C*        SCNOUT    SENSOR REPORT OUTPUT (SCANNING SENSORS)                *
C*                                                                         *
C***************************************************************************
```

Figure 2.2-19 (Cont.)

```
C******************************* EX3IMG ********************************
C*                                                                     *
C*                      SUBROUTINE EX3IMG                              *
C*                                                                     *
C*    PURPOSE                                                          *
C*        MSM LEVEL 3 EXECUTIVE ROUTINE, SUPERVISING CALLS TO          *
C*        SENSOR SUBROUTINE IMAGE, AND PROVIDING CONTROL AND           *
C*        INTERFACE WITH INPUT EVENTS, OUTPUT ROUTINES, STOR-          *
C*        AGE, AND SUCH RELATED ROUTINES AS SCAN1 AND SCAN2            *
C*        IN THE IMAGE CONTEXT.                                        *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL EX3IMG (LIST)                                           *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*                                                                     *
C*        LIST      IS AN EXPLICIT INPUT ARRAY, CORRESPONDING          *
C*                  TO AN EVENT TYPE 1 SUBLIST...                      *
C*                       WORD(S)                                       *
C*                          1      ITYPEV  (= 1)                       *
C*                          2      LNGTHS  (LENGTH OF LIST)            *
C*                          3      IDSNSR                              *
C*                          4      ITIM1    (BEGIN AND END TIME, INT-  *
C*                          5      ITIM2     EGER, OF EVENT INTERVAL)  *
C*                          6      NRBTGS  (NUMBER OF RED/BLUE         *
C*                                           TARGETS)                  *
C*                       (7 THRU            (TARGET ID/LEG CODES FOR   *
C*                         6+NRBTGS)          RED/BLUE TARGETS. CODE,  *
C*                                          DENOTED BY IDTL, IS OF     *
C*                                          FORM ID + 1000*LEGNO.)     *
C*                        (NEXT)           IF FALSE TARGETS EXIST,     *
C*                                           THIS WORD SPECIFIES       *
C*                                           NFTGTS (NO. OF FALSE      *
C*                                           TARGETS).                 *
C*                       (REMAINING)       PARAMETERS FOR FALSE TGTS   *
C*                                          ARE PASSED IN THE EVENT    *
C*                                          LIST PER SE, WITH NUMBER   *
C*                                          OF WORDS PER FALSE TGT     *
C*                                          = 12. FIRST WORD FOR A     *
C*                                          GIVEN FALSE TARGET IS A    *
C*                                          TYPE CODE (EVID) FORMED    *
C*                                          IN BATTLE OR CULTURE       *
C*                                          ROUTINES.                  *
C*                                                                     *
C*    REMARKS                                                          *
C*        1. IF DETECTION OCCURS, THE OUTPUT ROUTINE SCNOUT IS         *
C*           CALLED.  INFORMATION IS PASSED BY EXPLICIT ARGUMENTS      *
C*           AND VIA COMMON AREAS /SENSOR/, /TARGET/ AND /TGLGCM/.     *
C*                                                                     *
C*        2.  PARAMETERS FOR RED/BLUE TARGETS ARE OBTAINED FROM        *
```

Figure 2.2-20

```
C*           STORAGE ARRAY MSMPAR, VIA UTILITY ROUTINES IF NEC-          *
C*           ESSARY.  PARAMETERS FOR FALSE TARGETS ARE ACCESSED          *
C*           OR DERIVED FROM THE INPUT EVENT LIST.                       *
C*                                                                       *
C*       3.  STORAGE SPACE AVAILABLE IN COMMON /TARGET/ LIMITS           *
C*           TOTAL NUMBER OF TARGETS (TARGET/LEGS) THAT CAN BE           *
C*           SIMULTANEOUSLY HANDLED TO 16.  IF MORE THAN 16 ARE          *
C*           INDICATED IN THE INPUT LIST, THEN (A) THE TOTAL             *
C*           NUMBER IS TRUNCATED TO 16, WITH PRIORITY TO RED/            *
C*           BLUE TARGETS, AND (B) A DIAGNOSTIC MESSAGE IS               *
C*           PRINTED TO INDICATE LOSS OF TARGETS                         *
C*                                                                       *
C*       4.  PARTIAL PROTECTION AGAINST INCONSISTENT DATA IS             *
C*           PROVIDED, WITH DIAGNOSTIC MESSAGES PRINTED.                 *
C*                                                                       *
C*       5.  STORAGE CONTROL IS RELATIVELY COMPLICATED. IN AD-           *
C*           DITION TO THE OBVIOUS COMMON AREAS /SENSOR/ AND             *
C*           /TARGET/, 50 AND 210 WORDS RESPECTIVELY, ADDIT-             -
C*           IONAL STORAGE FOR (TARGET) PARAMETERS EFFECTIVELY           *
C*           IS PROVIDED BY COMMON /TGLGCM/ AS WELL AS THE EVENT         *
C*           LIST.  USE OF /TGLGCM/ INVOLVES KNOWLEDGE OF THE            *
C*           INTERNAL OPERATIONS OF SUBROUTINES TGTLG AND TGTLXY.        *
C*                                                                       *
C*       6.  THIS ROUTINE HANDLES BOTH MOVING RECTANGLE AND SECTOR       *
C*           SCAN COVERAGE MODES.  IT DOES NOT HANDLE STATIONARY         *
C*           RECTANGLE OR SECTOR MODES.  TWO TYPES OF SECTOR SCAN        *
C*           LOGICS ARE PROVIDED, CORRESPONDING TO SUBROUTINES           *
C*           SCAN1 AND SCAN2.                                            *
C*                                                                       *
C*       7.  ROUTINE ACCOMMODATES STATIONARY GROUND, MOVING GROUND       *
C*           AND AIRBORNE IMAGE SENSORS.                                 *
C*                                                                       *
C*       8.  BATTLEFIELD ILLUMINATION (E.G., FLARES), IF ANY,            *
C*           IS ACCESSED FROM STORAGE (MSM14) AND PASSED TO              *
C*           THE SENSOR ROUTINE.  IN THE RARE EVENT THAT MULTIPLE        *
C*           ILLUMINATIONS EXIST SIMULTANEOUSLY (AND NOTING THAT         *
C*           SENSOR ROUTINE IMAGE CAN ACCOMODATE ONLY ONE SUCH           *
C*           EVENT), THAT ILLUMINATION SOURCE CLOSEST TO THE             *
C*           SENSOR WILL BE USED.                                        *
C*                                                                       *
C*   SUBROUTINE AND FUNCTION TYPE SUBPROGRAMS REQUIRED                   *
C*       (IN ORDER OF OCCURRENCE IN PROGRAM)                             *
C*                                                                       *
C*       ERASE      BLOCK ERASE (SET TO 0)                               *
C*       PARPTR     MSM UTILITY, TO LOCATE PARAMS IN STORAGE             *
C*       TRNSFR     BLOCK TRANSFER OF DATA                               *
C*       TGTPTR     MSM UTILITY, FOR POINTER DETERMINATION               *
C*       TGTVLU     MSM UTILITY. GETS VALUE OF A RED OR BLUE             *
C*                     TARGET (FORCES) PARAMETER                         *
C*       KSTVLU     EVALUATES 'KSTRNG' VARIABLES FOR TARGET              *
```

Figure 2.2-20 (Cont.)

```
C*        FTPAR1    DERIVES AND STORES STATIC FALSE TARGET PAR-          *
C*                    AMETERS                                            *
C*        BFIASK    PRELIMINARY BATTLEFIELD ILLUMINATION INFO.           *
C*        TGTLG     BASIC TARGET/LEG SUBROUTINE                          *
C*        SETSC1    SETS WORKING PARAMETERS FOR SCAN1                    *
C*        SETSC2    SETS WORKING PARAMETERS FOR SCAN2                    *
C*        TGLGXY    GETS TARGET/LEG PARAMETERS THAT CHANGE WITH TIME     *
C*        GRN       GAUSSIAN RANDOM NUMBER GENERATOR                     *
C*        FTPAR2    DERIVES AND STORES DYNAMIC FALSE TARGET PARAMS.      *
C*        BFILUM    BATTLEFIELD ILLUMINATION                             *
C*        SCAN1     THE TWO BASIC SECTOR SCAN ROUTINES (NOT CALLED       *
C*        SCAN2       FOR MOVING RECTANGLE COVERAGE)                     *
C*        IMAGE     SENSOR SUBROUTINE                                    *
C*        SCNOUT    SENSOR REPORT OUTPUT (SCANNING SENSORS)              *
C*                                                                       *
C*************************************************************************
```

Figure 2.2-20 (Cont.)

```
C****************************  EX3THV  ******************************
C*                                                                   *
C*                        SUBROUTINE EX3THV                          *
C*                                                                   *
C*     PURPOSE                                                       *
C*         MSM LEVEL 3 EXECUTIVE ROUTINE, SUPERVISING CALLS TO       *
C*         SENSOR SUBROUTINE THERML,AND PROVIDING CONTROL AND        *
C*         INTERFACE WITH INPUT EVENTS, OUTPUT ROUTINES, STOR-       *
C*         AGE, AND SUCH RELATED ROUTINES AS SCAN1 AND SCAN2         *
C*         IN THE THERML CONTEXT.                                    *
C*                                                                   *
C*     CALLING SEQUENCE                                              *
C*         CALL EX3THV (LIST)                                        *
C*                                                                   *
C*     DESCRIPTION OF PARAMETERS                                     *
C*                                                                   *
C*         LIST       IS AN EXPLICIT INPUT ARRAY, CORRESPONDING      *
C*                    TO AN EVENT TYPE 1 SUBLIST...                  *
C*                         WORD(S)                                   *
C*                            1       ITYPEV  (= 1)                  *
C*                            2       LNGTHS  (LENGTH OF LIST)       *
C*                            3       IDSNSR                         *
C*                            4       ITIM1    (BEGIN AND END TIME, INT- *
C*                            5       ITIM2     EGER, OF EVENT INTERVAL) *
C*                            6       NRBTGS  (NUMBER OF RED/BLUE    *
C*                                              TARGETS)             *
C*                         (7 THRU              (TARGET ID/LEG CODES FOR *
C*                           6+NRBTGS)            RED/BLUE TARGETS. CODE, *
C*                                             DENOTED BY IDTL, IS OF *
C*                                             FORM ID + 1000*LEGNO.) *
C*                          (NEXT)            IF FALSE TARGETS EXIST, *
C*                                              THIS WORD SPECIFIES  *
C*                                              NFTGTS (NO. OF FALSE *
C*                                              TARGETS).            *
C*                         (REMAINING)        PARAMETERS FOR FALSE TGTS *
C*                                             ARE PASSED IN THE EVENT *
C*                                             LIST PER SE, WITH NUMBER *
C*                                             OF WORDS PER FALSE TGT *
C*                                             = 12. FIRST WORD FOR A *
C*                                             GIVEN FALSE TARGET IS A *
C*                                             TYPE CODE (FVID) FORMED *
C*                                             IN BATTLE OR CULTURE  *
C*                                             ROUTINES.             *
C*                                                                   *
C*     REMARKS                                                       *
C*         1. IF DETECTION OCCURS, THE OUTPUT ROUTINE SCNOUT IS      *
C*            CALLED.  INFORMATION IS PASSED BY EXPLICIT ARGUMENTS   *
C*            AND VIA COMMON AREAS /SENSOR/, /TARGET/ AND /TGLGCM/.  *
C*                                                                   *
C*         2.  PARAMETERS FOR RED/BLUE TARGETS ARE OBTAINED FROM     *
```

Figure 2.2-21

```
C*             STORAGE ARRAY MSMPAR, VIA UTILITY ROUTINES IF NEC-          *
C*             ESSARY.  PARAMETERS FOR FALSE TARGETS ARE ACCESSED          *
C*             OR DERIVED FROM THE INPUT EVENT LIST.                       *
C*                                                                         *
C*          3. STORAGE SPACE AVAILABLE IN COMMON /TARGET/ LIMITS           *
C*             TOTAL NUMBER OF TARGETS (TARGET/LEGS) THAT CAN BE           *
C*             SIMULTANEOUSLY HANDLED TO 16.  IF MORE THAN 16 ARE          *
C*             INDICATED IN THE INPUT LIST, THEN (A) THE TOTAL             *
C*             NUMBER IS TRUNCATED TO 16, WITH PRIORITY TO RED/            *
C*             BLUE TARGETS, AND (B) A DIAGNOSTIC MESSAGE IS               *
C*             PRINTED TO INDICATE LOSS OF TARGETS                         *
C*                                                                         *
C*          4. PARTIAL PROTECTION AGAINST INCONSISTENT DATA IS             *
C*             PROVIDED, WITH DIAGNOSTIC MESSAGES PRINTED.                 *
C*                                                                         *
C*          5. STORAGE CONTROL IS RELATIVELY COMPLICATED. IN AD-           *
C*             DITION TO THE OBVIOUS COMMON AREAS /SENSOR/ AND             *
C*             /TARGET/, 50 AND 210 WORDS RESPECTIVELY, ADDIT-             -
C*             IONAL STORAGE FOR (TARGET) PARAMETERS EFFECTIVELY           *
C*             IS PROVIDED BY COMMON /TGLGCM/ AS WELL AS THE EVENT         *
C*             LIST.  USE OF /TGLGCM/ INVOLVES KNOWLEDGE OF THE            *
C*             INTERNAL OPERATIONS OF SUBROUTINES TGTLG AND TGTLXY.        *
C*                                                                         *
C*          6. THIS ROUTINE HANDLES BOTH MOVING RECTANGLE AND SECTOR       *
C*             SCAN COVERAGE MODES.  IT DOES NOT HANDLE STATIONARY         *
C*             RECTANGLE OR SECTOR MODES.  TWO TYPES OF SECTOR SCAN        *
C*             LOGICS ARE PROVIDED, CORRESPONDING TO SUBROUTINES           *
C*             SCAN1 AND SCAN2.                                            *
C*                                                                         *
C*          7. ROUTINE ACCOMMODATES STATIONARY GROUND, MOVING GROUND       *
C*             AND AIRBORNE THERMAL SENSORS.                               *
C*                                                                         *
C*      SUBROUTINE AND FUNCTION TYPE SUBPROGRAMS REQUIRED                  *
C*          (IN ORDER OF OCCURRENCE IN PROGRAM)                            *
C*                                                                         *
C*          ERASE     BLOCK ERASE (SET TO 0)                               *
C*          PARPTR    MSM UTILITY, TO LOCATE PARAMS IN STORAGE             *
C*          TRNSFR    BLOCK TRANSFER OF DATA                               *
C*          TGTPTR    MSM UTILITY, FOR POINTER DETERMINATION               *
C*          TGTVLU    MSM UTILITY. GETS VALUE OF A RED OR BLUE             *
C*                       TARGET (FORCES) PARAMETER                         *
C*          KSTVLU    EVALUATES 'KSTRNG' VARIABLES FOR TARGET              *
C*          FTPAR1    DERIVES AND STORES STATIC FALSE TARGET PAR-          *
C*                       AMETERS                                           *
C*          TGTLG     BASIC TARGET/LEG SUBROUTINE                          *
C*          SETSC1    SETS WORKING PARAMETERS FOR SCAN1                    *
C*          SETSC2    SETS WORKING PARAMETERS FOR SCAN2                    *
C*          TGTLXY    GETS TARGET/LEG PARAMETERS THAT CHANGE WITH TIME     *
C*          GRN       GAUSSIAN RANDOM NUMBER GENERATOR                     *
C*          FTPAR2    DERIVES AND STORES DYNAMIC FALSE TARGET PARAMS.      *
```

Figure 2.2-21 (Cont.)

```
C*        SCAN1     THE TWO BASIC SECTOR SCAN ROUTINES (NOT CALLED          *
C*        SCAN2       FOR MOVING RECTANGLE COVERAGE)                       *
C*        THERML    SENSOR SUBROUTINE                                      *
C*        SCNOUT    SENSOR REPORT OUTPUT (SCANNING SENSORS)                *
C*                                                                         *
C**************************************************************************
```

Figure 2.2-21 (Cont.)

```
C******************************  EX3BKW  ********************************
C*                                                                      *
C*                         SUBROUTINE EX3BKW                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*        MSM LEVEL 3 EXEC. HANDLES DETAILS OF TYPE 1 EVENTS            *
C*        (SENSOR INTERROGATE) FOR BREAKWIRE                            *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL EX3BKW (LIST)                                            *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*        LIST       A TYPE-1-EVENT SUBLIST,  CONTAINING SENSOR ID,     *
C*                      TIME INFORMATION, TARGET ID'S.                  *
C*        IMPLICIT PARAMETERS (FOR SENSOR AND TARGETS) FROM COMMON.     *
C*                                                                      *
C*    REMARKS                                                           *
C*        1. THIS SUBROUTINE CALLED ONLY BY EX2SNR.                     *
C*                                                                      *
C*    SUBROUTINE  AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*        ERASE      BLOCK ERASE (SET TO 0)                             *
C*        FTPAR1     DERIVES STATIC PARAMETERS FOR FALSE TARGET         *
C*        BRKWIR     SUBROUTINE FOR BREAKWIRE RESPONSES                 *
C*                      TO TARGETS                                      *
C*        PARPTR     UTILITY ROUTINE TO LOCATE PARAMS IN COMMON         *
C*        TRNSFR     UTILITY ROUTINE FOR BLOCK TRANSFER OF DATA         *
C*        TGTPTR     UTILITY ROUTINE TO LOCATE TARGET PARAMS            *
C*        ITODEV     GIVES THE TIME OF DAY                              *
C*        UGSDET     INTERFACE TO OUTPUT (FOR UGS SENSOR DETECTIONS)    *
C*                                                                      *
C************************************************************************
```

Figure 2.2-22

2.2.5.4 Sensor Routines

In MSM are those sensor subroutines that are concerned with sensor responses to discrete targets., For some sensor types, a corresponding background routine -- part of PRERUN, not MSM -- does calculations of operating parameters as they are affected by atmospheric variables or by background noise sources. Parameters so calculated are transmitted to MSM (for use by the target-interaction sensor routines) by input event type 3 (Sensor Parameter Change).

There are 11 subroutines in MSM that are regarded as sensor routines. The 9 primary ones correspond respectively to the 9 generic types of sensors considered. The names are:

1.	SEISTG	Seismic
2.	ACOUTG	Acoustic
3.	MAGTG	Magnetic
4.	ARFTG	Arfbuoy
5.	PIRTG	Passive infrared
6.	RADAR	Radar
7.	IMAGE	All image type sensors, including unaided human vision, binoculars, starlight scopes, night vision devices
8.	THERML	Thermal viewer
9.	BRKWIR	Breakwire

The numeric order has some significance to MSM coding, inasmuch as sensor generic types are internally referred to by the numeric code 1 through 9. Supporting the IMAGE routine are two auxiliaries, ANG and QUAD, that do not interact with any other MSM subprograms.

Explicit detailed descriptions of sensor routines per se are given in Section 3. The placement of these routines within the overall MSM structure is shown in Figure 2.2-3. Although Fig. 2.2-3 does not show the numerous auxiliary routines, it may be noted that the only major interactions of sensor routines with MSM occur via the EX3... series of sensor executive routines.

2.2.5.5 Output Routines

There are in MSM nine subprograms that write printer or binary tape output under normal* circumstances. Of these, seven are shown on the MSM OUTPUT diagram of Fig. 2.2-2; the other two are minor auxiliary routines.

---

*A number of programs may print diagnostic messages in event of certain [illegible] errors. Programs for which this is the only possible output are not counted.

The primary output routines are considered to be:

UGSOUT	Reports from UGS sensors to monitors
SCNOUT	Reports from scanning (attended) sensors
EX2HLT	System summary preparation and print

The former two write both printer and binary output. The last writes only printer output.

The following two subroutines are strictly output routines, but produce output less "physically" important than that produced by the three routines listed above:

EX2SNP	System snapshot: printed summary of system sensors active at a given instant of time
DUMPMS	Produced a printed dump of the system parameter data from common area/BIGSTR/

These routines are considered to be optional. Because their operations do not influence basic MSM computation, they can be replaced without recompilation of any other subprogram. If the printouts are not desired, for example, dummy versions (with'RETURN' as the only executable statement) could be substituted. Or substitute versions giving lesser or greater detail could be substituted. It is felt, for example, that full DUMPMS output is desirable during exploratory exercise of the simulation model, but that a dummy version may be desirable for later production running.

Printed alphanumeric information from header cards is produced by EXEC1, which also writes a binary output listing of system parameters (counterpart of the printed listing produced by DUMPMS).

The following two subroutines produce minor printout (no binary output):

EXEC1B	Notations on beginning or end of precipitation
EX2UPD	Notations on (a) beginning or end of firetrap operations, and (b) effective times of array emplacements and cease operations.

Two utility routines, not shown in Fig. 2.2-2, provide control of printer page skip, page and line count sequencing, and page heading printouts. PGSKIP is the basic 'page skip' routine. If called, it

(a) advanced the printer page

(b) increments the page number, and resets the line count

(c) prints a top-of-page heading comprising page number and run identification (alphanumeric) that originally came from the first two header cards.

Subroutine PGSKP2 performs a bookkeeping and heading print function for subroutines UGSOUT and SCNOUT. It determines (from line count) if the printer page should be advanced. If so, then PGSKP2:

(a) calls PGSKIP

(b) prints those column headings appropriate for the data output of subroutines UGSOUT, SCNOUT

(c) exits with line count adjusted for the space taken by the column heading print.

It may be noted that the column headings are meaningful only to identify UGSOUT and SCNOUT data, so only PGSKIP (not PGSKP2) is called for other categories of printout.

The following two subroutines do not in themselves create output, but they do act in direct support of UGSOUT:

UGSDET	collects informaton on reports from four types* of UGS sensors, and reduces this information to a common format acceptable to UGSOUT
SACDET	same purpose as UGSDET, but only for seismic and acoustic sensors (that have storage formats considerably different from the other four types of UGS sensors).

Thus, UGSDET and SACDET act as interfaces between the EX3... sensor executive routines and the UGSOUT routine. Both subroutines are shown in Fig. 2.2-2

The following two programs are not shown in Fig. 2.2-2, but do support output programs:

*Magnetic, passive infrared, ARFBUOY and breakwire.

ALFCVT	Special integer to alphanumeric conversion, for sensor report printouts (for positive integer, forms alphanumeric equivalent; for zero integer, forms alphanumeric blanks; for negative integer, forms string of asterisks).
TIMOUT	In MSM printer output of sensor reports, time is printed in "external units," i.e., day, military (24 hour) clock, and seconds. TIMOUT converts simulation internal time (integer seconds into game) to data words corresponding to external time.

ALFCVT is one of the few subprograms supplied that is (slightly) machine dependent. Comments heading the program listing discuss adaptations to machines with, for example, different word or byte structures.

Comments on the nine individual subprograms related to output are given in Figs. 2.2-23 through 2.2-31, respectively.

```
C*****************************  UGSOUT  ******************************
C*                                                                   *
C*                      SUBROUTINE UGSOUT                            *
C*                                                                   *
C*    PURPOSE                                                        *
C*        MSM OUTPUT PROGRAM FOR ALL UGS SENSOR REPORTS. PRINTS      *
C*        'GAME PLAY' AND 'GROUND TRUTH' INFORMATION.  ALSO PUTS     *
C*        ALL BASIC INFORMATION ON BINARY TAPE (OR DISK) FOR USE     *
C*        IN POST-PROCESSING.                                        *
C*                                                                   *
C*    CALLING SEQUENCE                                               *
C*        CALL UGSOUT                                                *
C*                                                                   *
C*    DESCRIPTION OF PARAMETERS                                      *
C*            * INPUT (BASIC) *                                      *
C*        BASIC SENSOR REPORT DATA ARE SUPPLIED AS INPUT BY ONE OF   *
C*        THE 3 CALLING PROGRAMS IN THE FIRST 13 WORDS OF COMMON     *
C*        AREA /OUTCOM/.                                             *
C*                                                                   *
C*            * OUTPUT *                                             *
C*       BASIC OUTPUTS ARE:                                          *
C*             PRINTED -- 'GAME PLAY' + 'GROUND TRUTH' INFORMATION   *
C*                        ON SENSOR REPORTS                          *
C*             BINARY TAPE (OR DISK)  ALL BASIC SENSOR REPORT INFOR- *
C*                        MATION, FOR USE IN POST-PROCESSING.        *
C*             PROGRAM ALSO INCREMENTS SYSTEM COUNTERS.              *
C*                                                                   *
C*    REMARKS                                                        *
C*        1. DATA IN /OUTCOM/ ARE SET BY, AND UGSOUT CALLED BY,      *
C*           THESE THREE MSM PROGRAMS:                               *
C*              SACDET    REPORTS FROM SEISMIC AND ACOUSTIC SENSORS  *
C*              UGSDET    REPORTS FROM  THE 4 TYPES OF UGS SENSORS   *
C*                           OTHER THAN SEISMIC AND ACOUSTIC         *
C*              FX2SFA    FALSE ALARM REPORTS                        *
C*                                                                   *
C*        2. IN THIS ROUTINE IS CHECKED THE INFORMATION FLOW FROM    *
C*           SENSOR TO MONITOR VIA DATA LINK.  OUTPUT CONTAINS       *
C*           INFORMATION ON WHICH MONITOR(S) RECEIVE SIGNAL.         *
C*                                                                   *
C*        3. FOR SENSORS WITHOUT AN AND-LOGIC SPECIFICATION, THERE   *
C*           IS IMMEDIATE PROCESSING AND OUTPUT.  FOR SENSORS WITH   *
C*           AND-LOGIC SPECIFIED, REPORTS ARE QUEUED IF NECESSARY    *
C*           AWAITING REPORT FROM CONFIRMING SENSOR.  NO OUTPUT IS   *
C*           GENERATED IN THIS CASE UNLESS THE TWO ASSOCIATED SEN-   *
C*           SORS BOTH REPORT WITHIN A SPECIFIED TIME INTERVAL.      *
C*                                                                   *
C*        4. THE COUNTERPART OF THIS ROUTINE (FOR UGS REPORTS) IS    *
C*           SCNOUT (FOR SCANNING SENSOR REPORTS).  FORMATS OF       *
C*           PRINTOUT, AND ORDERING OF DATA ON BINARY OUTPUT, ARE    *
C*           EQUIVALENT FOR DATA HAVING COMPARABLE MEANINGS.         *
```

Figure 2.2-23

```
C*                                                                           *
C*        5. THE OUTPUT UNIT DESIGNATION FOR BINARY OUTPUT IS DE-            *
C*           NOTED HERE BY THE NAME IOUT70, AND GIVEN THE NUMERI-            *
C*           CAL VALUE 70 BY DATA STATEMENT.  ROUTINES SCNOUT AND            *
C*           PGSKP2 REFER TO THIS UNIT WITH SAME NAME AND VALUE.             *
C*           ANY CHANGE IN NUMERICAL VALUE MUST BE MADE IN ALL               *
C*           THREE ROUTINES.                                                 *
C*                                                                           *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                           *
C*           (IN ORDER OF OCCURRENCE IN PROGRAM LISTING)                     *
C*        PARPTR    MSM UTILITY, TO SET POINTERS TO PARAMETERS               *
C*        ARRPTR    MSM UTILITY, TO SET POINTERS TO SYSTEM ARRAYS            *
C*        PARVLU    MSM UTILITY. GETS VALUE OF A PARAM                       *
C*        ALFCVT    CONVERTS INTEGERS TO SPECIAL-FORMAT ALPHANU-             *
C*                     MERIC FORM, FOR PRINTOUT                              *
C*        TRNSFR    BLOCK TRANSFER OF STORED DATA                            *
C*        PGSKP2    PAGE SKIP CONTROL PLUS CONTROL OF PAGE HEAD-             *
C*                     INGS UNIQUE TO SENSOR-REPORT PRINTOUT.                *
C*        TIMOUT    CONVERTS INTERNAL GAME TIME TO EXTERNAL  (DAY,           *
C*                     CLOCK,SECOND) FORM                                    *
C*        ARRVLU    MSM UTILITY. GETS VALUE OF AN ARRAY PARAM                *
C*                                                                           *
C****************************************************************************
```

Figure 2.2-23 (Cont.)

```
C******************************** SCNOUT *******************************
C*                                                                    *
C*                        SUBROUTINE SCNOUT                           *
C*                                                                    *
C*    PURPOSE                                                         *
C*        OUTPUT (PRINTED AND BINARY TAPE) ROUTINE FOR THE THREE      *
C*        TYPES OF SCANNING SENSORS.-- RADAR, IMAGE AND THERMVEW.     *
C*        CALLED WHEN A DETECTION OCCURS WITH SCANNING SENSOR.        *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL SCNOUT (TIME,INDEX,SGNOIS,PDET,ISCAN,ARGLST)           *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*            * INPUT, EXPLICIT *                                     *
C*        TIME      GAME TIME, REAL, SECONDS                          *
C*        INDEX     SUBSCRIPT FOR STORAGE ARRAYS IN COMMON /TARGET/,  *
C*                    CORRESPONDING TO THE PARTICULAR TARGET (OF A    *
C*                    POTENTIAL 16) THAT HAS BEEN DETECTED.           *
C*        SGNOIS    FOR RADAR, THE SIGNAL/NOISE (SIGNAL/CLUTTER)      *
C*                    RATIO IN DB. FOR IMAGE AND THERMVEW, THE        *
C*                    ANALOGOUS MEASURE.                              *
C*        PDET      DETECTION PROBABILITY REPORTED BY SENSOR ROUTINE  *
C*        ISCAN     SCAN CODE, = 1 OR 2 FOR SECTOR SCAN OF TYPE       *
C*                    SCAN1, SCAN2 RESP., = 3 FOR MOVING RECTANGLE.   *
C*        ARGLST    'LAST ARGUMENT' = FDOPLR (DOPPLER FREQUENCY)      *
C*          FOR RADAR,  = ILXTRA (INDICATOR OF BATTLEFIELD ILLUMI-    *
C*          NATION, GT 0 IF SUCH EXISTS) FOR IMAGE, NOT SIGNIFICANT   *
C*          FOR THERMVEW.                                             *
C*                                                                    *
C*            * INPUT, IMPLICIT *                                     *
C*          VIA COMMON AREAS /SENSOR/, /TARGET/ AND /TGLGCM/          *
C*                                                                    *
C*            * OUTPUT -- EXTERNAL AND VIA COMMON *                   *
C*          FORMATTED OUTPUT ON PRINTER.  UNFORMATTED (BINARY)        *
C*          OUTPUT ON TAPE OR DISK 'UNIT 70', 33 WORDS.  SYSTEM       *
C*          COUNTERS IN /SYSCNT/ ARE INCREMENTED.                     *
C*                                                                    *
C*    REMARKS                                                         *
C*        THIS OUTPUT ROUTINE FOR SCANNING SENSORS IS THE COUNTER-    *
C*        PART OF UGSOUT, THE OUTPUT ROUTINE FOR UGS SENSORS. BOTH    *
C*        PRINTED AND TAPE OUTPUTS ARE 'MATCHED' FOR THOSE DATA       *
C*        THAT HAVE SAME MEANINGS.                                    *
C*                                                                    *
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*        TRNSFR    BLOCK TRANSFER OF DATA                            *
C*        PGSKP2    SPECIAL PAGE SKIP CONTROL AND HEADER PRINT        *
C*        ALFCVT    SPECIAL INTEGER-TO -ALPHANUMERIC ROUTINE          *
C*        TIMOUT    CONVERTS INTERNAL TIME TO EXTERNAL FORM           *
C*        ARRPTR    MSM UTILITY, FOR GETTING POINTER VALUES           *
C*        PARVLU    MSM UTILITY, TO ACCESS VALUE OF PARAMETER         *
```

Figure 2.2-24

```
C*******************************  SACDET  ********************************
C*                                                                      *
C*                        SUBROUTINE SACDET                             *
C*                                                                      *
C*    PURPOSE                                                           *
C*        MSM INTERFACE ROUTINE BETWEEN EX3SAC (WHICH SUPERVISES        *
C*        SEISMIC AND ACOUSTIC SENSOR ROUTINES) AND THE GENERAL         *
C*        UGS OUTPUT ROUTINE UGSOUT.  CALLED BY EX3SAC WHEN A           *
C*        SEISMIC OR ACOUSTIC SENSOR ROUTINE REPORTS A DETECTION.       *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*        CALL SACDET (IDSNR,ISORA,XSNR,YSNR,NRBTGS,NFTGS)              *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*                                                                      *
C*            * INPUT, EXPLICIT *                                       *
C*        IDSNR      ID OF (SEISMIC, ACOUSTIC) SENSOR                   *
C*        ISORA      = 1 FOR SEISMIC, OR 2 FOR ACOUSTIC                 *
C*        XSNR       COORDINATES                                        *
C*        YSNR         OF SENSOR                                        *
C*        NRBTGS     NUMBER OF RED+BLUE TARGETS                         *
C*        NFTGS      NUMBER OF FALSE TARGETS                            *
C*                                                                      *
C*            * INPUT, IMPLICIT *                                       *
C*                                                                      *
C*        ITIME      DETECTION TIME (INTEGER SECONDS INTO GAME),        *
C*                     ACCESSED VIA COMMON /BASICT/                     *
C*        ALL TAR-   ACCESSED VIA COMMON /TARGET/                       *
C*         GET PAR-    SEE EX3SAC DOCUMENTATION FOR                     *
C*          AMETERS      EXPLICIT DESCRIPTION.                          *
C*                                                                      *
C*    REMARKS                                                           *
C*        SACDET, LIKE OTHERS IN THE ...DET SERIES FOR UGS SENSORS,     *
C*        PERFORMS THE DATA EDITING AND MANAGEMENT NECESSARY THAT       *
C*        THE COMMON OUTPUT ROUTINE, UGSOUT, BE CALLED WITH A CON-      *
C*        SISTENT FORMAT.                                               *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*        UGSOUT     OUTPUT ROUTINE FOR ALL UGS SENSOR DETECTIONS       *
C*                                                                      *
C************************************************************************
```

Figure 2.2-25

```
C******************************  UGSDET  ******************************
C*                                                                    *
C*                       SUBROUTINE UGSDET                            *
C*                                                                    *
C*    PURPOSE                                                         *
C*        MSM INTERFACE ROUTINE BETWEEN EX3MAG, EX3ARF, EX3PIR,       *
C*        EX3BKW AND THE GENERAL OUTPUT ROUTINE, UGSOUT, FOR UGS      *
C*        SENSOR DETECTIONS.  CALLED BY THE EX3... ROUTINE WHEN       *
C*        CORRESPONDING SENSOR ROUTINE REPORTS A DETECTION.           *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL UGSDET (IDSR,ITPSR,XSR,YSR,IDTG,XTG,YTG,NRBTG)         *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*            * INPUT, EXPLICIT *                                     *
C*        IDSR       ID OF SENSOR                                     *
C*        ITPSR      INTEGER TYPE CODE FOR SENSOR (3=MAGNETIC, 4=ARF- *
C*                     BUOY, 5=PASSIVIR, 9=BREAKWIR)                  *
C*        XSR        COORDINATES                                      *
C*        YSR          OF SENSOR                                      *
C*        IDTG       ID OF TARGET                                     *
C*        XTG        COORDINATES                                      *
C*        YTG          OF TARGET                                      *
C*        NRBTG      NO. OF RED/BLUE TARGETS...  =1 OR 0, LATTER      *
C*                     CASE IMPLYING FALSE-TARGET DETECTION           *
C*                                                                    *
C*            * INPUT, IMPLICIT *                                     *
C*        TIME        INTEGER GAME TIME, VIA COMMON /BASICT/          *
C*            * OUTPUT *                                              *
C*        FIRST 13 WORDS OF COMMON /OUTCOM/, TO BE USED BY SUB-       *
C*        ROUTINE UGSOUT.                                             *
C*                                                                    *
C*    REMARKS                                                         *
C*        1.  THERE ARE 6 TYPES OF UGS SENSORS,  DETECTIONS FROM      *
C*            WHICH ARE HANDLED PRIMARILY BY UGSOUT. THE INTER-       *
C*            FACE BETWEEN THE SENSOR EXEC ROUTINE (EX3...) AND       *
C*            UGSOUT IS HANDLED BY                                    *
C*                 UGSDET    FOR TYPES 3 (MAG), 4 (ARF), 5 (PIR),     *
C*                             AND 9 (BKW)                            *
C*                 SACDET    FOR TYPES 1 (SEI) AND 2 (ACO)            *
C*                                                                    *
C*        2.  THE SENSOR TYPES HANDLED BY THIS ROUTINE DIFFER FROM    *
C*            SEISMIC AND ACOUSTIC, IN THAT ONLY ONE TARGET CAN       *
C*            AFFECT A DETECTION AT ANY ONE INSTANT OF TIME.          *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION TYPE SUBPROGRAMS REQUIRED               *
C*        UGSOUT     OUTPUT ROUTINE FOR UGS DETECTIONS                *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-26

```
C*******************************  PGSKIP  *******************************
C*                                                                      *
C*                         SUBROUTINE PGSKIP                            *
C*                                                                      *
C*    PURPOSE                                                           *
C*       CONTROLS PRINTER PAGE SKIP, PRINTING OF HEADING AND PAGE       *
C*       NUMBER, AND RESETTING PAGE AND LINE COUNTS.                    *
C*                                                                      *
C*    CALLING SEQUENCE                                                  *
C*       CALL PGSKIP                                                    *
C*                                                                      *
C*    DESCRIPTION OF PARAMETERS                                         *
C*       PARAMETERS TAKEN FROM COMMON AREA /PGCONT/ --                  *
C*                                                                      *
C*         ILINES     LINE COUNT (NO. OF LINES USED ON PAGE)            *
C*         IPAGE      PAGE COUNT                                        *
C*         HEADNG     ALPHANUMERIC TEXT, PRINTED AS HEADING ON          *
C*                    FIRST LINE OF EACH PAGE OF MSM PRINTOUT.          *
C*                    ARRAY CONTAINS 68 CHARACTERS IN 17 WORDS.         *
C*                                                                      *
C*    REMARKS                                                           *
C*       1. LINE AND PAGE COUNTS ARE INITIALIZED TO ZERO IN MSM         *
C*             BLOCK DATA.  HEADING TEXT IS USER INPUT...               *
C*             SEE COMMENTS IN EXEC1 PROGRAM.                           *
C*       2. OPERATIONS OF PGSKIP--                                      *
C*             A. PRINTER PAGE IS ADVANCED                              *
C*             B. PAGE COUNT IS INCREMENTED BY 1                        *
C*             C. HEADING AND NEW PAGE NO. ARE PRINTED                  *
C*             D.  TWO LINES ARE SKIPPED                                *
C*             E. LINE COUNT IS SET TO 3                                *
C*                                                                      *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*       NONE                                                           *
C*                                                                      *
C************************************************************************
```

Figure 2.2-27

```
C*****************************  PGSKP2  ******************************
C*                                                                   *
C*                        SUBROUTINE PGSKP2                          *
C*                                                                   *
C*    PURPOSE                                                        *
C*       MSM UTILITY ROUTINE FOR OUTPUT CONTROL, SUPPLEMENTING       *
C*       SUBROUTINE PGSKIP.  IN ADDITION TO CONTROLLING PAGE SKIP,   *
C*       THIS ROUTINE PRINTS THE COLUMN HEADINGS REQUIRED FOR        *
C*       SENSOR REPORTS.                                             *
C*                                                                   *
C*    CALLING SEQUENCE                                               *
C*       CALL PGSKP2                                                 *
C*                                                                   *
C*    DESCRIPTION OF PARAMETERS                                      *
C*       NO EXPLICIT ARGUMENTS                                       *
C*       THE VARIABLE ILINES IN COMMON /PGCOUNT/ IS AN INPUT VARI-   *
C*         ABLE FOR BRANCHING, AND IS ALTERED (INCREMENTED) IF       *
C*         HEADINGS ARE PRINTED.                                     *
C*                                                                   *
C*    REMARKS                                                        *
C*       1. THIS SINGLE ROUTINE SUPPLANTS SEPARATELY CODED BUT       *
C*          IDENTICAL CODING IN THE 4 DISTINCT OUTPUT ROUTINES.      *
C*                                                                   *
C*       2. THE 'UNIT NUMBER' FOR THE OUTPUT TAPE (DISK) SPECIFIED   *
C*          HERE MUST AGREE WITH THAT SPECIFIED IN THE OUTPUT ROU-   *
C*          TINES.  THE UNIT NUMBER HERE IS CALLED IOUT70. ITS VALUE *
C*          IS SET BY A DATA STATEMENT (TO VALUE 70).                *
C*                                                                   *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*       PGSKIP          PRIMARY PAGE SKIP/CONTROL ROUTINE           *
C*                                                                   *
C*********************************************************************
```

Figure 2.2-28

```
C******************************** TIMOUT ***********************************
C*                                                                         *
C*                       SUBROUTINE TIMOUT                                 *
C*                                                                         *
C*    PURPOSE                                                              *
C*        CONVERTS ITIME (= INTEGER NUMBER OF SECONDS INTO GAME)           *
C*        INTO 3 INTEGER WORDS REPRESENTING DAY OF GAME (0,1,...),         *
C*        MILITARY CLOCK TIME (HHMM) AND SECONDS, SUITABLE FOR             *
C*        PRINTOUT IN I2, I4 AND I2 FORMATS, RESPECTIVELY.                 *
C*                                                                         *
C*    USAGE                                                                *
C*        CALL TIMOUT (IDAY,ICLK24,ISEC)
C*                                                                         *
C*    DESCRIPTION OF PARAMETERS                                            *
C*                                                                         *
C*             INPUTS (IMPLICIT-- FROM COMMON /BASICT/)                    *
C*                ITIME    INTEGER TIME (SECONDS) FROM GAME START          *
C*                ITODST   TIME OF DAY AT GAME START (INTEGER, SEC-        *
C*                           ONDS SINCE CLOCK 0000 OF START DAY)           *
C*                                                                         *
C*             OUTPUTS (EXPLICIT IN ARGUMENT LIST)                         *
C*                IDAY     DAY COUNT (0 FOR INITIAL 'D-DAY',               *
C*                           1 FOR D-DAY+1, ETC.)                          *
C*                ICLK24   MILITARY (24-HR) CLOCK TIME HHMM, WHEN          *
C*                           PRINTED AS A 4-DIGIT DECIMAL INTEGER          *
C*                ISEC     TWO-DIGIT INTEGER (00 THRU 59) = SECONDS        *
C*                                                                         *
C*    REMARKS                                                              *
C*        THE QUANTITY ITODST IN /BASICT/ IS A GAME CONSTANT.              *
C*        ITIME, HOWEVER, VARIES... SO CALLING PROGRAM MUST ASSURE         *
C*        THAT ITIME IS CORRECTLY ENTERED INTO /BASICT/ BEFORE             *
C*        SUBROUTINE TIMOUT IS CALLED.                                     *
C*                                                                         *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                         *
C*        NONE                                                             *
C*                                                                         *
C***************************************************************************
```

Figure 2.2-29

```
C*******************************  ALFCVT  ********************************
C*                                                                       *
C*                         FUNCTION ALFCVT                               *
C*                                                                       *
C*    PURPOSE                                                            *
C*        CONVERTS INTEGER INPUT TO ALPHANUMERIC FORM REQUIRED BY        *
C*        SUBROUTINE UGSOUT FOR PRINTOUT.                                *
C*                                                                       *
C*    CALLING SEQUENCE  (FUNCTION TYPE)                                  *
C*        ADIGIT  =  ALFCVT(KDIG)                                        *
C*                                                                       *
C*    DESCRIPTION OF PARAMETERS                                          *
C*        KDIG       INTEGER (OF 3 OR FEWER DECIMAL DIGITS IF POSITIVE)  *
C*                                                                       *
C*    REMARKS                                                            *
C*                                                                       *
C*        1. PROGRAM SOMEWHAT MACHINE DEPENDENT.  PRESENT FORM           *
C*           COMPATIBLE WITH IBM 360 SERIES AND OTHER COMPUTERS          *
C*           HAVING THIS WORD STRUCTURE FOR ALPHANUMERICS:               *
C*                 A. 8 BITS PER BYTE                                    *
C*                 B. 4 BYTES PER (NORMAL LENGTH) WORD                   *
C*                 C. ALPHANUMERIC BLANK HAS 0 LEADING BIT (WORD         *
C*                      BEGINNING WITH BLANK WOULD BE CONSIDERED         *
C*                      POSITIVE IF USED IN ARITHMETIC EXPRESSION)       *
C*           MINOR CHANGES WILL ADAPT PROGRAM TO OTHER WORD STRUCT-      *
C*           URES.  FOR EXAMPLE, IF ALL BUT (A) ARE SATISFIED, THEN      *
C*           THE DATA STATEMENT FOR NBYT NEED ONLY BE CHANGED FROM       *
C*           2**8 = 256 TO 2**N (N = NO. BITS PER BYTE).                 *
C*
C*        2. THE CONVERSION RULES ARE                                    *
C*              A. IF KDIG = 0,  OUTPUT = '    ' (ALL BLANKS)            *
C*              B. IF KDIG IS NEGATIVE, OUTPUT = ' ***' (BLANK,          *
C*                    FOLLOWED BY 3 ASTERISKS)                           *
C*              C. IF KDIG IS POSITIVE, THE 3 LOW ORDER DIGITS           *
C*                    ARE CONVERTED TO ONE OF THESE FORMS                *
C*                         'BDBB'  IF KDIG OF FORM    D (1 DIGIT)        *
C*                         'BDDB'  IF KDIG OF FORM   DD (2 DIGITS)       *
C*                         'BDDD'  IF KDIG OF FORM  DDD (3 DIGITS)       *
C*                                                                       *
C*        3. IF THE INPUT VARIABLE KDIG HAS MORE THAN 3 SIGNIF-          *
C*           ICANT DIGITS, PROGRAM OPERATES ON LAST 3.                   *
C*                                                                       *
C*                                                                       *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                       *
C*        NONE                                                           *
C*                                                                       *
C************************************************************************
```

Figure 2.2-30

```
C****************************  DUMPMS  *********************************
C*                                                                     *
C*                         SUBROUTINE DUMPMS                           *
C*                                                                     *
C*    PURPOSE                                                          *
C*       LISTS ('DUMPS') PARAMETER STORAGE AREA /BIGSTR/, AFTER        *
C*       EDITING, FOR REFERENCE AND/OR PROGRAM OR DATA DEBUGGING/      *
C*       VERIFICATION.                                                 *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*       CALL DUMPMS                                                   *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*       NO EXPLICIT PARAMETERS                                        *
C*       INFORMATION REQUIRED BY DUMPMS ALL IN COMMON AREAS            *
C*                                                                     *
C*    REMARKS                                                          *
C*       REAL, INTEGER FORMATS MATCHED TO MSMPAR CONTENT, EXCEPT       *
C*       FOR A FEW VARIABLES FOR WHICH THE TYPE IS VARIABLE.  THE      *
C*       CURRENT (20 OCT 1970) PROGRAM USES Z-FORMAT (HEXADECIMAL)     *
C*       FOR THESE... A MACHINE-DEPENDENT AND NON-USASI-FORTRAN        *
C*       FORMAT.                                                       *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*       PGSKIP     PAGE SKIP (PRINTER), RESETS PACE AND LINE          *
C*                  COUNTS, PRINTS STANDARD HEADING                    *
C*                                                                     *
C***********************************************************************
```

Figure 2.2-31

2.2.5.6 Input Routines

Three short input subroutines are invoked during initial loading of storage. TANDT (Time AND Terrain), called by EXEC 1, directly reads time parameters (BASICT table), and calls TERAN to read the UTVSXY and UNTER terrain parameter tables. PARMIN, called by EXECIA, is a simple read program for the data file of system parameters. Each call to PARMIN causes a small block of storage in EXECIA to be filled with (a) one parameter subset (e.g., the parameters for one sensor, or one UGSARRAY), or (b) the special indicator flag that delimits a major data category.

Figures 2.2-32 through 2.2-34 provide comments for these three subroutines -- TANDT, TERAN and PARMIN -- on an individual basis.*

---

*TERAN is more fully documented in Section 5.3 of this Volume, and in Volume II. It is used in PRERUN and in auxiliary program blocks, as well as in MSM.

```
C******************************** TANDT ********************************
C*                                                                     *
C*                       SUBROUTINE TANDT                              *
C*                                                                     *
C*    PURPOSE                                                          *
C*       MSM ROUTINE, AUXILIARY TO EXEC1. READS 'TIME AND TERRAIN'     *
C*       INTO STORAGE.                                                 *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*       CALL TANDT                                                    *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*       AN INPUT SUBROUTINE, TANDT READS INTO STORAGE THE TIME        *
C*       AND TERRAIN PARAMETERS FOR THESE FOUR COMMON AREAS            *
C*                                                                     *
C*           /BASICT/     BASIC GAME TIME PARAMETERS                   *
C*           /ATTIME/     TABLE OF TIME VALUES AT WHICH THE DIF-       *
C*                          FERENT ATMOSPHERIC TABLES BECOME           *
C*                          EFFECTIVE                                  *
C*           /UNTER/      TABLES OF PARAMETERS FOR UP TO 8 UNIT        *
C*                          TERRAIN TYPES                              *
C*           /UTVSXY/     TABLES FROM WHICH THE UNIT TERRAIN TYPE      *
C*                          CAN BE KEYED TO GEOGRAPHIC POSITION        *
C*                          WITHIN PRE-SET SCENARIO AREA.              *
C*                                                                     *
C*    REMARKS                                                          *
C*       1. ALTHOUGH TANDT COMPRISES IN EFFECT ONLY A FEW READ         *
C*          STATEMENTS, IT IS CODED AS A DISTINCT SUBROUTINE TO        *
C*          FACILITATE CHANGES DUE TO VARYING INPUT SOURCES FOR        *
C*          THE DATA.                                                  *
C*                                                                     *
C*       2. IN PARTICULAR, THIS PARTICULAR VERSION (27 OCTOBER 70)     *
C*          IS TENTATIVE, FOR USE IN MSM PROGRAM DEVELOPMENT.          *
C*          PROBABLY DESIRABLE, LATER, WOULD BE A TAPE OR DISK         *
C*          FILE PREPARED IN PRERUN FOR WHICH 2 SIMPLE READ            *
C*          STATEMENTS COULD BE CODED TO REPLACE STATEMENT 10.         *
C*          SUCH A CHANGE WOULD ELIMINATE THE NEED IN MSM FOR SUB-     *
C*          ROUTINE  TERAN  AND ITS IMPLIED NEED FOR CARD DATA.        *
C*                                                                     *
C*                                                                     *
C*       3. INPUT UNIT (IUNIT3) FOR /BASICT/ AND /ATTIME/ REFERS       *
C*          TO THE TAPE (OR DISK) ATMOSPHERIC DATA FILE.  JCL          *
C*          (JOB CONTROL LANGUAGE) CAN ASSIGN THE CORRESPONDING        *
C*          PHYSICAL INPUT DEVICE.  CONSISTENCY MUST BE MAIN-          *
C*          TAINED WITH SUBROUTINE EXEC13  (CORRESPONDING UNIT         *
C*          ALSO DENOTED IUNIT3, WITH SAME NUMERIC VALUE, 3)...        *
C*          THE ONLY OTHER MSM PROGRAM ACCESSING THIS FILE.            *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*       TERAN          (SEE REMARK 2 ABOVE)                           *
```

Figure 2.2-32

```
C******************************** TERAN *************************************
C                                                                          *
C*    SUBROUTINE TERAN                                                     *
C*                                                                         *
C*    PURPOSE                                                              *
C*                                                                         *
C*        THIS ROUTINE PLACES THE PLANNER INPUT DATA DESCRIBING THE        *
C*    AREAL EXTENT OF THE UNIT TERRAIN TYPES OVER THE SCENARIO AREA        *
C*    INTO THE 'UTVSXY' COMMON AREA AND PLACES THE DESIGNER INPUT FOR      *
C*    THE UNTER TABLES INTO THE 'UNTER' COMMON AREA.  ALL DATA INPUT       *
C*    IS ASSUMED TO BE ON CARDS.                                           *
C*                                                                         *
C***************************************************************************
```

Figure 2.2-33

```
C******************************  PARMIN  ******************************
C                                                                     *
C*                       SUBROUTINE PARMIN                            *
C*                                                                    *
C*    PURPOSE                                                         *
C*        INPUT ROUTINE FOR THE BODY OF SYSTEM PARAMETERS PASSED      *
C*        TO MSM FROM PREPUN.  CALLED ONLY BY EXEC1A.                 *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL PARMIN (IDATA)                                         *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        IDATA      AN ARRAY INTO WHICH PARMIN MUST PLACE (ON ANY    *
C*                   ONE CALL)--                                      *
C*                                                                    *
C*                      A.  PARAMETER SUBLIST FOR ONE SYSTEM ELEMENT  *
C*                          OR ONE DESCRIPTOR DATA SET,  OR           *
C*                                                                    *
C*                      B.  THE FLAG VALUE 9999 IN IDATA(1), SIGNI-   *
C*                          FYING THE END OF A MAJOR DATA CATEGORY.   *
C*                                                                    *
C*        IUNITP     INPUT UNIT DESIGNATION FOR THE DATA SOURCE.      *
C*                                                                    *
C*                                                                    *
C*                   VALUE ASSIGNED HERE IS 11, BUT JCL (JOB CONTROL  *
C*                   LANGUAGE) WILL CONTROL ACTUAL ASSIGNMENT.        *
C*                                                                    *
C*    REMARKS                                                         *
C*        ALTHOUGH NEAR-TRIVIAL IN CONTENT, PARMIN IS CODED AS A      *
C*        DISTINCT SUBROUTINE TO PROVIDE CONVENIENT ACCOMMODATION     *
C*        TO POSSIBLE CHANGES IN INPUT DATA SOURCE OR FORMAT.         *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        NONE                                                        *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-34

2.2.5.7 System Type Utility Routines

Nine of the MSM subprograms have no implications of "physical significance," and are denoted as system utility routines. 'System' in this context refers to a general computational system, not the physical sensor system being simulated.

ERASE and TRNSFR provided convenient control of block storage operations. ERASE is used for erasure (setting to zero) of an arbitrary block of storage. TRNSFR is used for block transfer of data from one storage array to another.

ERFC is a function-type subprogram that evaluates the complementary error function:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

The FORTRAN-coded version supplied is fully general enough and accurate enough for the simulation model; it is not intended as a general routine for computer installations. Required by the sensor routines, this ERFC program is supplied as part of the MSM package for two reasons: (a) some computer installations do not provide an ERFC routine in their software support, and (b) some "standard" ERFC routines supplied by computer manufacturers generate false underflow diagnostics when the argument becomes large (and the mathematical value of ERFC becomes smaller than the smallest floating point number that can be represented internally).

Two random number generating decks, each with three entry points, were supplied as IBM 360 assembly language programs. This is the sole deviation from FORTRAN. Although it is possible to write random number generators in FORTRAN language, nearly all major computer installations use machine language coded generators because of the very significant speed advantage for a program that tends to be called a very large number of times in typical usage. The deck supplied allows CAL-generated results to be duplicated exactly on any IBM 360 large enough to handle the model... a very desirable feature in transfer of programs.

The uniform random number package has three entry points, logically equivalent to three distinct subprograms.

URN acts as a function type subprogram in which the argument is a dummy (not used). Each call provides a "new" random number, from a statistical distribution uniform over 0.0 to 1.0.

URNORG is used to set the initial integer "state" of the generator. Provision is made on one of the header cards for the user to insert an initial state reference value. If this data field is not zero or blank, EXEC 1 calls URNORG with the planner's value as argument.

URNASK provides for interrogation of the random number generator as to its "current" reference state. URNASK is called by EX2HLT, which then prints out the reference value that existed at the termination of MSM processing.

The entry points for the gaussian random number package--GRN, GRNORG and GRNASK--are analogous to URN, URNORG and URNASK, respectively, and the same comments generally apply. The difference, of course, is that each call to GRN provides a standardized (i.e., zero mean, unit variance) random variable with gaussian distribution.

Figures 2.2-35 through 2.2-39 provide comments on individual subprogram (or entry points, in the case of the random number generators) for the routines in the system utility category.

```
C******************************  ERASE  ********************************
C*                                                                     *
C*                         SUBROUTINE ERASE                            *
C*                                                                     *
C*    PURPOSE                                                          *
C*        ERASES (SETS TO ZERO) A SPECIFIED BLOCK OF STORAGE           *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL ERASE(A,N)                                              *
C*               OR                                                    *
C*        CALL ERASE (IA,N)                                            *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        A,IA   VARIABLE NAME, SPECIFYING FIRST WORD OF AREA          *
C*                 TO BE ERASED.  REAL OR INTEGER NAME MAY BE          *
C*                 SPECIFIED IN CALLING SEQUENCE.                      *
C*        N      NUMBER OF WORDS (IN SEQUENCE) TO BE ERASED            *
C*                                                                     *
C*    REMARKS                                                          *
C*        N = 0 IS A VALID IF TRIVIAL ARGUMENT VALUE.  A NEGA-         *
C*        TIVE VALUE OF N IS TREATED AS IF N = 0.                      *
C*                                                                     *
C***********************************************************************
```

Figure 2.2-35

```
C****************************** TRNSFR ********************************
C*                                                                    *
C*                        SUBROUTINE TRNSFR                           *
C*                                                                    *
C*    PURPOSE                                                         *
C*        BLOCK TRANSFER OF INFORMATION IN ONE STORAGE AREA           *
C*        INTO ANOTHER.                                               *
C*                                                                    *
C*    USAGE                                                           *
C*        CALL TRNSFR (ARRAY1,ARRAY2,N)                               *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        FIRST N WORDS OF ARRAY2 ARE LOADED WITH VALUES OF           *
C*        FIRST N WORDS OF ARRAY1.  ARRAY1 IS LEFT UNCHANGED          *
C*        IF THE TWO AREAS DO NOT OVERLAP.                            *
C*                                                                    *
C*        A 'WORD' IN THIS CASE IS 'NORMAL' OR DEFAULT LENGTH         *
C*        (E.G., 4 BYTES ON IBM 360).  WORD TYPES (REAL,              *
C*        INTEGER, LOGICAL, ...) IMMATERIAL.                          *
C*                                                                    *
C*    REMARKS                                                         *
C*        1.  ZERO VALUE FOR N IS ACCEPTABLE.  NEGATIVE VALUES        *
C*            OF N (ERROR) TREATED AS IF N=0-- THAT IS, NO            *
C*            TRANSFER OCCURS.  NO DIAGNOSTIC MESSAGE.                *
C*                                                                    *
C*        2.  THIS FORTRAN PROGRAM, PROVIDED AS A PROJECT SAM I       *
C*            AUXILIARY PROGRAM, SHOULD IDEALLY BE REPLACED WITH      *
C*            AN ASSEMBLY (MACHINE) LANGUAGE PROGRAM FOR WHATEVER     *
C*            COMPUTER IS USED FOR 'PRODUCTION' RUNNING.              *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-36

```
C******************************  ERFC  **********************************
C*                                                                      *
C*     PURPOSE                                                          *
C*        MATHEMATICAL FUNCTION SUBPROGRAM FOR PROJECT SAMI SIMULA-     *
C*        TION MODEL.  EVALUATES COMPLEMENTARY ERROR FUNCTION           *
C*        ERFC(X) FOR POSITIVE (OR ZERO) X,  TO APPROXIMATELY           *
C*        5 DECIMAL PLACE ACCURACY.                                     *
C*                                                                      *
C*     CALLING SEQUENCE (ILLUSTRATIVE)                                  *
C*        Y = ERFC(X)                                                   *
C*                                                                      *
C*     DESCRIPTION OF PARAMETERS                                        *
C*        X     FOR THIS PROGRAM, X MUST BE POSITIVE OR ZERO.           *
C*                                                                      *
C*     METHOD AND REFERENCE                                             *
C*        A. FOR X GREATER THAN 5.0, A ZERO VALUE IS RETURNED           *
C*              (COMPATIBLE WITH ACCURACY CRITERION OF 5 DECI-          *
C*              MAL PLACES)                                             *
C*        B. FOR X IN THE 0 TO 5 INTERVAL, A 'HASTINGS' TYPE AP-        *
C*              PROXIMATION FORMULA WITH ERROR NOT EXCEEDING            *
C*              2.5E-5 IN MAGNITUDE IS USED...  FORMULA 7.1.25          *
C*              FROM FOLLOWING REFERENCE:                               *
C*                                                                      *
C*                 HANDBOOK OF MATHEMATICAL FUNCTIONS                   *
C*                 EDITED BY: ABRAMOWITZ, STEGUN                        *
C*                 NATIONAL BUREAU OF STANDARDS                         *
C*                 APPLIED MATHEMATICS SERIES  55                       *
C*                 JUNE 1964       PAGE 299                             *
C*                                                                      *
C*                 (ORIGINAL SOURCE:  C. HASTINGS, JR.,                 *
C*                 'APPROXIMATIONS FOR DIGITAL COMPUTERS')              *
C*                                                                      *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*        NONE                                                          *
C*                                                                      *
C************************************************************************
```

Figure 2.2-37

```
***************************** URN PACKAGE **************************** 000000
*                                                                    * 000000
*                 UNIFORM RANDOM NUMBER GENERATOR                    * 000000
*                                                                    * 000000
*      PURPOSE                                                       * 000000
*         MULTIPLE-ENTRY PROGRAM FOR GENERATION OF UNIFORM RAND-     * 000000
*         OM NUMBERS, AND SETTING OR INTERROGATING THE REFERENCE     * 000000
*         STATE OF THE GENERATOR.                                    * 000000
*                                                                    * 000001
*      USAGE                                                         * 000001
*         ILLUSTRATIVE FORTRAN CALLS OF THE VARIOUS ENTRIES ARE--    * 000001
*                                                                    * 000001
*         1. BASIC URN ENTRY - FUNCTION TYPE                         * 000001
*                  U = URN (DUMMY)                                   * 000001
*            WHERE THE ARGUMENT (DUMMY) IS IGNORED GIVES INDE-       * 000001
*            PENDENT, FLOATING-POINT, SINGLE PRECISION (4-BYTE)      * 000001
*            NUMBERS UNIFORM ON (0,1.0).                             * 000001
*                                                                    * 000001
*            NOTE. PROGRAM CANNOT RETURN AN EXACT 0.0 OR 1.0, BUT    * 000002
*            CAN RETURN VALUE WITHIN ONE BIT OF 1.0, AND A VALUE     * 000002
*            AS SMALL AS 2**(-24) (ABOUT 6X10**(-8)).                * 000002
*                                                                    * 000002
*         2. URNTP ENTRY (FILL AN ARRAY WITH URN'S)                  * 000002
*                  CALL URNTP (ARRAY,N)                              * 000002
*            PUTS URN'S IN ARRAY(1),ARRAY(2),...ARRAY(N)             * 000002
*                                                                    * 000002
*         3. URNRST ENTRY  (RESET GENERATOR TO LOAD-TIME STATE)      * 000002
*                  CALL URNRST                                       * 000002
*            RESTORES GENERATOR TO ITS LOAD-TIME STATE               * 000003
*            (       = X'7FFFFFFE'  =  2**31-2  =  F'2147483646' )   * 000003
*                                                                    * 000003
*         4. URNORG ENTRY  (SET ORIGIN, OR STATE)                    * 000003
*                  CALL URNORG (IREF)                                * 000003
*            SETS STATE (RN IN CODING) TO THE VALUE OF IREF, EX-     * 000003
*            CEPT THAT                                               * 000003
*                  (A) IF IREF IS NEGATIVE, THE ABSOLUTE VALUE       * 000003
*                      OF IREF IS USED.                              * 000003
*                  (B) IF IREF IS ZERO (MOD P), THE LOAD-TIME        * 000003
*                      VALUE X'7FFFFFFF' IS USED INSTEAD.            * 000004
*            NOTE. THE ALGORITHM PRECLUDES NEGATIVE OR EFFECTIVE-    * 000004
*            LY ZERO VALUES.  PROGRAM GUARDS AGAINST SUCH VALUES     * 000004
*            BEING INADVERTENTLY SUPPLIED BY USERS.                  * 000004
*                                                                    * 000004
*         5. URNASK ENTRY  (INTERROGATE STATE)                       * 000004
*                  CALL URNASK (IREF)                                * 000004
*            SETS VALUE OF IREF TO VALUE OF THE REFERENCE STATE.     * 000004
*                                                                    * 000004
*         6. URNPRT ENTRY  (PRINT STATE)                             * 000004
*                  CALL URNPRT                                       * 000005
*            CAUSES STATE REFERENCE (RN) TO BE PRINTED IN I FOR-     * 000005
```

Figure 2.2-38

```
*         MAT, WITH APPROPRIATE MESSAGE.                                 *
*                                                                        *
*                                                                        *
*     REMARKS                                                            *
*        THE AUXILIARY ENTRIES FACILITATE CONTROL OF 'WHERE THE          *
*        GENERATOR IS'.  FOR EXAMPLE, IF URNPRT (EQUIVALENTLY,           *
*        URNASK PLUS USER'S OWN PRINTOUT) IS USED AT THE TERMI-          *
*        NATION OF A COMPUTER RUN, A SUBSEQUENT RUN CAN BE INITI-        *
*        ATED FROM THE SAME URN STATE BY--                               *
*           (A) READING THE REFERENCE NUMBER FROM A DATA CARD            *
*           (B) USING URNORG TO INITIALIZE THE GENERATOR.                *
*                                                                        *
*                                                                        *
*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                       *
*        ERRMSG                                                          *
*                                                                        *
*                                                                        *
*     METHOD                                                             *
*        THE ALGORITHM USED FOR THE BASIC (INTEGER) RANDOM               *
*        NUMBERS R IS                                                    *
*             R(N+1) = A*R(N)  MOD P                                     *
*                                                                        *
*        WHERE P IS THE LARGEST PRIME CONSISTENT WITH REGISTER           *
*        CAPACITY, AND A IS A PRIMITIVE ROOT OF P.                       *
*                                                                        *
*        IN THIS CASE, P = 2**31-1 (BY COINCIDENCE, A FULL REG-          *
*        ISTER FULL OF BITS (NOTE SIGN BIT NOT USED, SO ONLY A           *
*        31-BIT REGISTER IS ASSUMED), AND A = 3125.                      *
*                                                                        *
*                                                                        *
*        THE VALUES RETURNED TO CALLING PROGRAM FOR THE URN OR           *
*        URNTP ENTRIES ARE FLOATED AND SCALED (TO 0.0-1.0) VER-          *
*        SIONS OF THE BASIC INTEGERS.  THE CURRENT VALUE OF THE          *
*        BASIC INTEGER (RN IN CODING) IS CALLED THE STATE OF THE         *
*        GENERATOR, AND CAN BE INTERROGATED OR RESET WITH THE            *
*        ENTRIES URNRST, URNASK, URNORG OR URNPRT.                       *
*                                                                        *
*                                                                        *
*     REFERENCE                                                          *
*        DAVID W. HUTCHISON                                              *
*        'A NEW UNIFORM PSEUDORANDOM NUMBER GENERATOR'                   *
*        COMM. ACM  VOL.9  NO.6  JUNE,1966  PAGE 432                     *
*                                                                        *
*                                                                        *
C*************************************************************************
```

Figure 2.2-38 (Cont.)

```
***************************  GRN PACKAGE  *************************** 000000
*                                                                   * 000000
*                            CU 0024                                * 000000
*                 GAUSSIAN RANDOM NUMBER GENERATOR                  * 000000
*                   CONTROL SECTION NAME: GAUS#                     * 000000
*                   ENTRY POINTS:                                   * 000000
*                     GRN      GRNTP    GRNPRT                      * 000000
*                     GRNRST   GRNORG   GRNASK                      * 000000
*                                                                   * 000001
*     PURPOSE                                                       * 000001
*        GENERATION OF PSEUDO-RANDOM GAUSSIAN DEVIATES, WITH AUX-   * 000001
*        ILIARY ENTRIES FOR SETTING OR INTERROGATING THE GENERATOR. * 000001
*                                                                   * 000001
*     USAGE                                                         * 000001
*                                                                   * 000001
*        1. GRN ENTRY-  FUNCTION-TYPE ENTRY                         * 000001
*                      CALLED IN FORTRAN WITH DUMMY ARGUMENT, E.G.  * 000001
*                           G = GRN(DUMMY)                          * 000001
*                      THE RESULTS ARE FLOATING POINT, SINGLE-PRE-  * 000002
*                      CISION, ZERO MEAN, UNIT VARIANCE.            * 000002
*                                                                   * 000002
*        2. GRNTP ENTRY-  SUBROUTINE-TYPE ENTRY FOR FILLING AN ARRAY * 000002
*                      WITH GRN'S.                                  * 000002
*                           CALL GRNTP(ARRAY,N)                     * 000002
*                      WILL CAUSE GRN'S TO BE GENERATED AND PLACED  * 000002
*                      INTO THE N FULL WORDS ARRAY(1),ARRAY(2)...   * 000002
*                      .. ARRAY(N)                                  * 000002
*                                                                   * 000002
*        3. GRNRST ENTRY- THE STATEMENT                             * 000003
*                               CALL GRNRST                         * 000003
*                      WILL CAUSE THE REFERENCE STATE (#URNG)       * 000003
*                      OF THE GENERATOR TO BE RESET, THAT IS,       * 000003
*                      SET TO THE LOAD-TIME VALUE X'7FFFFFFE'.      * 000003
*                                                                   * 000003
*        4. GRNORG ENTRY- THE STATEMENT                             * 000003
*                               CALL GRNORG(ISTATE)                 * 000003
*                      WILL SET THE REFERENCE STATE OF THE GEN-     * 000003
*                      ERATOR TO THE VALUE OF ISTATE, EXCEPT        * 000003
*                      THAT:                                        * 000004
*                        A. THE RIGHTMOST 31 BITS OF ISTATE ARE     * 000004
*                           USED (LEADING BIT IS MASKED OUT).       * 000004
*                        B. IF THE RESULT OF A IS ONE OF THE TWO    * 000004
*                           'ILLEGAL' VALUES, 0 OR X'7FFFFFFF',     * 000004
*                           THE CALL IS INTERPRETED AS CALL GRNRST. * 000004
*                           THAT IS, THE VALUE X'7FFFFFFE' IS       * 000004
*                           USED INSTEAD OF ISTATE.                 * 000004
*                                                                   * 000004
*        5. GRNASK ENTRY- FOR INTERROGATING. THE STATEMENT          * 000004
*                           CALL GRNASK(ISTATE)                     * 000005
*                      CAUSES THE REFERENCE STATE OF THE GEN-       * 000005
```

Figure 2.2-39

```
*                   ERATOR TO BE PLACED INTO LOCATION ISTATE          * 000005
*                                                                     * 000005
*    6. GRNPRT ENTRY- ALTERNATIVE METHOD OF INTERROGATION. THE        * 000005
*                   STATEMENT                                         * 000005
*                      CALL GRNPRT                                    * 000005
*                   CAUSES A SYSOUT PRINTOUT OF THE REFERENCE         * 000005
*                   STATE WITH MESSAGE, E.G.                          * 000005
*                                                                     * 000005
*                 CURRENT GRN REFERENCE STATE = 2747483646 (DEC      * 000006
*                 IMAL)  = 7FFFFFFF (BASE 16)                         * 000006
*                   THIS MESSAGE APPEARS ON ONE LINE, IS PRE-         * 000006
*                   CEDED BY A BLANK LINE.                            * 000006
*                                                                     * 000006
*      REMARKS                                                        * 000006
*         1. THE DECK REQUIRES 4E0(BASE 16) = 1248(BASE 10) BYTES.    * 000006
*         2. EXECUTION TIMES VARY RANDOMLY, BUT AVERAGE TIME TO       * 000006
*            GET ONE GRN IS ABOUT 85 MICROSECONDS (ON 360/65).        * 000006
*                                                                     * 000006
*      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  * 000007
*           ERRMSG                                                    * 000007
*           SQRT                                                      * 000007
*           ALOG                                                      * 000007
*           EXP                                                       * 000007
*        (ALL LOADED AUTOMATICALLY FROM CAL SYSTEM LIBRARY)           * 000007
*                                                                     * 000007
*      METHOD, REFERENCES                                             * 000007
*         1. INTERNALLY GENERATED UNIFORM RANDOM NUMBERS ARE MAPPED   * 000007
*            INTO GAUSSIAN DEVIATES ACCORDING TO THE METHOD DESCRIBED * 000007
*            IN                                                       * 000008
*               G.MARSAGLIA AND T.BRAY                                * 000008
*              'A CONVENIENT METHOD FOR GENERATING                    * 000008
*                  NORMAL VARIABLES'                                  * 000008
*               SIAM REVIEW, VOL.6, NO.3, JULY, 1964 PP260-264        * 000008
*                                                                     * 000008
*            BRIEFLY, THE LOGIC IS:                                   * 000009
*               86.39% OF THE TIME: AVERAGE 3 UNIFORM DEVIATES        * 000009
*               11.07% OF THE TIME: AVERAGE 2 UNIFORM DEVIATES        * 000009
*                2.28% OF THE TIME: COMPLICATED REJECTION METHOD      * 000009
*                0.27% OF THE TIME: (VALUES BEYOND  3 SIGMA)          * 000009
*                                   USE POLAR METHOD                  * 000009
*            THE METHOD ACHIEVES HIGH AVERAGE SPEED, BY USING SIMPLE  * 000009
*            AND FAST ALGORITHMS FOR MORE THAN 97% OF THE GRN EVALU-  * 000009
*            ATIONS. OVER-ALL MAPPING IS MATHEMATICALLY EXACT.        * 000009
*                                                                     * 000009
*         2. FOR GENRATING THE UNDERLYING UNIFORM DEVIATES, THIS      * 000009
*            MULTIPLICATIVE-CONGRUENTIAL METHOD IS USED:              * 000009
*               U(N+1) =  A*U(N)    MOD P                             * 000009
*                                                                     * 000009
*            WHERE QUANTITIES ARE INTEGERS. U(K) REPRESENTS THE       * 000010
*            K'TH UNIFORM INTEGER.  P IS THE LARGEST PRIME LESS       * 000010
```

Figure 2.2-39 (Cont.)

```
*          THAN 2**31, WHICH HAPPENS TO BE                                     * 000010
*                    P = 2**31-1 = 2147483647 (BASE 10)                        * 000010
*                                = 7FFFFFFF   (BASE 16)                        * 000010
*                                                                              * 000010
*          A IS A PRIMITIVE ROOT OF P. THE PRIMITIVE ROOT USED IN              * 000010
*          THIS PROGRAM IS                                                     * 000010
*                    A = 87921497 (BASE 10) = 11**19 (MOD P)                   * 000010
*                                                                              * 000010
*          REFERENCE:                                                          * 000011
*                    D. W. HUTCHINSON                                          * 000011
*                   'A NEW UNIFORM PSEUDO-RANDOM                               * 000011
*                         NUMBER GENERATOR'                                    * 000011
*                   COMM. ACM  VOL. 9  NO. 6  JUNE,1966                          000011
*                                                                              * 000011
C******************************************************************************* 000011
```

Figure 2.2-39 (Cont.)

2.2.5.8 Storage Access Utility Routines

Six utility routines are provided, that simplify coding tasks for access to system parameters in storage area MSMPAR, and reduce the amount of storage that would be required if access instructions had to be duplicated many times over in the numerous users (programs) of parameters. These subroutines occur in pairs, according as the returned value is to be a storage pointer (program names ending in... PTR) or an actual parameter value (program names ending in... VLU). The names are

PARPTR
PARVLU
ARRPTR
ARRVLU
TGTPTR
TGTVLU.

The basic pair are PARPTR and PARVLU, that can be used for access to any system parameter. Input arguments specify category (1 through 13, corresponding to MSM1 through MSM13), subset count within category, and word count within subset list. The value returned is the pointer to the specified word if PARPTR is used, or a parameter value (any type: real, integer, logical, alphanumeric) if PARVLU is used.

Because of (a) frequent occurrences of need for parameters of system arrays and of forces (red or blue), and (b) special bookkeeping instructions required for these, four other access routines are supplied that may be regarded as special cases of PARPTR or PARVLU. ARRPTR and ARRVLU apply to system arrays; the subroutines themselves determine whether the specified array ID corresponds to UGSARRAYS (MSM1), SCANARRAYS (MSM2) or MOVARRAYS (MSM3). Analogously, TGTPTR and TGTVLU apply to "target" (red or blue force) parameters; the subroutines themselves determine whether the specified target ID corresponds to a blue force (MSM4) or red force (MSM5).

Exact calling sequences, definitions of arguments, and other comments are given in Fig. 2.2-40 through 2.2-45.

2.2.5.8.1 General Methodology

General Methodology for Determining Internal Storage Locations and their Values for Items of Data in MSM Storage follows.

The general form of the calling sequence to determine the internal storage locations of items of data in MSM storage is:

Call X (I, J, K, INDEX) where:

X - specifies the subroutine.

I - specifies major block; valid values 1 thru 13 for MSM1 thru MSM13.

J - specifies data set or sublist within major block.

K - word number within sublist.

INDEX - the internal storage location for the word specified by I, J, K.

In order to come up with a methodology, the storage arrays IPBIGS( ), IPBLU( ), IPPATH( ), LJUMP( ) and MSMPAR( ) must be used. These are defined respectively as the pointer or storage location of the first word of each major block of data in storage, each blue force, each path, the number of words in each data set of these major blocks, and master stream of data values in MSM storage.

Two equations of the following general form are required:

(1) $INDEX = A + B + C \qquad I = 1, 13, I \neq 4, I \neq 11.$

(2) $INDEX = B + C \qquad I = 4 \text{ or } I = 11.$

In equation (1), let A equal the storage location of the first word of the first data set of major block (I). Thus A = IPBIGS(I) and

(3) $INDEX = IPBIGS(I) + B + C$

$I + 1, 13, I \neq 4, I \neq 11.$

Continuing, A + 1 * LJUMP(I) gives the storage location of the first word of the 2nd data set of major block (I) and A + 2 * LJUMP(I) gives the 1st word of the 3rd data set of major block (I), etc. Thus, A + (J - 1) * LJUMP(I) gives the storage location of the 1st word of the Jth data set of major block (I) and B = (J - 1) * LJUMP(I) giving:

(4) $INDEX = IPBIGS(I) + (J - 1) * LJUMP(I) + C$

$I = 1, 13, I \neq 4, I \neq 11.$

To find C, A + B + 1 gives the storage location of the second word of the Jth data set of major block (I). A + B + 2 gives the storage location of the 3rd word of the Jth data set of major block (I), etc. Thus, A + B + (K - 1) gives the storage location of the Kth word of the Jth data set of major block (I) and C = K - 1.

giving:

(5) INDEX = IPBIGS(I) + (J - 1) LJUMP(I) + K - 1

I = 1, 13, I ≠ 4, I ≠ 11.

which is the required equation of the general form INDEX = A + B + C. Equation (5) will then be equal to the storage location of the Kth word of the Jth data set of the Ith major block.

In equation (2):

(6) INDEX = B + C   I = 4 or I = 11.

In this equation IPBIGS(I) is not needed because the storage arrays IPBLU(J) for I = 4 and IPPATH(J) for I = 11 hold the appropriate starting locations for these major blocks. Thus following the same logic as before:

(7) INDEX = B + C

(8) INDEX = IPBLU(J) + K - 1   I = 4.

This gives the storage location of the Kth word for the Jth blue force in MSM storage and:

(9) INDEX = IPPATH(J) + K - 1   I = 11.

This gives the storage location of the Kth word for the Jth path in MSM storage.

To get the values associated with each data item in MSM storage, the master data stream array, MSMPAR( ) is used. The general form of the calling sequence to determine any value is CALL X (I, J, K, IVALUE) where I, J, K are defined as before and:

(10) IVALUE = MSMPAR(INDEX) where INDEX is calculated as before.

Equation (10) will then give the value, based on INDEX, of the Kth word of the Jth data set of the Ith major block in MSM storage.

```
C*****************************  PARPTR  ******************************
C*                                                                    *
C*                       SUBROUTINE PARPTR                            *
C*                                                                    *
C*    PURPOSE                                                         *
C*        MSM UTILITY ROUTINE.  DETERMINES POINTER TO PARAMETER IN    *
C*        AREA MSMPAR,  FOR SPECIFIED MAJOR BLOCK, SUBLIST WITHIN     *
C*        BLOCK, AND WORD COUNT WITHIN SUBLIST.                       *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL PARPTR (I,J,K, INDEX)                                  *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        I,J,K  --    INPUTS                                         *
C*        INDEX  --    OUTPUT                                         *
C*                                                                    *
C*        I    SPECIFIES MAJOR BLOCK, VALID VALUES 1 (FOR MSM1)       *
C*                THRU 13 (FOR MSM13).                                *
C*        J    SPECIFIES DATA SET OR SUBLIST WITHIN BLOCK             *
C*        K    SPECIFIES WORD NUMBER WITHIN SUBLIST                   *
C*                                                                    *
C*        INDEX THEN IS THE POINTER TO MSMPAR FOR THE WORD SPECI-     *
C*                FIED BY I,J,K                                       *
C*                                                                    *
C*    REMARKS                                                         *
C*        RELATED SUBROUTINES ARE PARVLU, ARRPTR, ARRVLU, TGTPTR,     *
C*        AND TGTVLU.                                                 *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        NONE                                                        *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-40

```
C******************************  PARVLU  ******************************
C*                                                                    *
C*                      SUBROUTINE PARVLU                             *
C*                                                                    *
C*    PURPOSE                                                         *
C*        MSM UTILITY ROUTINE.  EXTRACTS PARAMETER VALUE FROM STOR-   *
C*        AGE AREA MSMPAR, FOR SPECIFIED MAJOR BLOCK, SUBLIST WITHIN  *
C*        BLOCK, AND WORD COUNT WITHIN SUBLIST.                       *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL PARVLU (I,J,K, VALUE)                                  *
C*                  OR                                                *
C*        CALL PARVLU (I,J,K, IVALUE)                                 *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        I,J,K,              -- INPUT                                *
C*        VALUE  OR  IVALUE -- OUTPUT                                 *
C*                                                                    *
C*        THE OUTPUT VARIABLE NAME MAY BE CHOSEN REAL (E.G, VALUE)    *
C*        OR INTEGER (E.G., IVALUE), CORRESPONDING TO WHAT THE        *
C*        STORED FORM IS.                                             *
C*                                                                    *
C*              I   SPECIFIES WHICH MAJOR BLOCK OF MSMPAR IS DESIRED. *
C*                  VALID VALUES 1 (CORRESPONDING TO MSM1, OR UGS     *
C*                  ARRAYS)  THRU 13 (CORRESPONDING TO MSM13, OR      *
C*                  COVERAGE/SCAN PARAMETER SETS)                     *
C*              J   SPECIFIES A PARTICULAR SUBSET WITHIN THE          *
C*                  MAJOR BLOCK.                                      *
C*              K   SPECIFIES WORD NUMBER WITHIN THE SUBLIST SET      *
C*                  BY I AND J.                                       *
C*                                                                    *
C*    REMARKS                                                         *
C*        RELATED SUBROUTINE --  PARPTR                               *
C*                                                                    *
C*        EXAMPLES OF USE OF PARVLU                                   *
C*           1.   CALL PARVLU (1,12,8, IDMON1)                        *
C*                    IDMON1 WILL BE THE VALUE OF THE 8'TH            *
C*                    WORD OF THE 12'TH SUBSET OF MSM1 -- THAT        *
C*                    IS, THE 8'TH WORD FOR ARRAY 12 WITHIN           *
C*                    THE UGS ARRAY PARAMETER BLOCK.  IN PAR-         *
C*                    TICULAR, IDMON1 WILL BE THE ID OF THE           *
C*                    PRIMARY MONITOR FOR UGS ARRAY 12.               *
C*                                                                    *
C*           2.   CALL PARVLU (5,12,10, ORIENT)                       *
C*                    GIVES AS ORIENT THE VALUE OF THE ORIENTA-       *
C*                    TION ANGLE GIVEN IN WORD 10, SENSOR 12.         *
C*                                                                    *
C*           3.   CALL PARVLU (2,12,8, IDETPP)                        *
C*                    GIVES VALUE OF 8'TH WORD, 12'TH SUBSET OF       *
C*                    MSM2 (STASCAN ARRAYS).  NOTE THAT THIS SE-      *
```

Figure 2.2-41

```
C*                       QUENCE KEYS TO THE 12'TH STASCAN ARRAY, NOT     *
C*                       THE SYSTEM ARRAY WITH ID = 12.  THE VALUE OB-   *
C*                       TAINED, INCIDENTALLY, IS ID OF FIRETRAP ASSOC-  *
C*                       IATED WITH THE 12'TH STASCAN ARRAY.             *
C*                                                                       *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                      *
C*         NONE                                                          *
C*                                                                       *
C************************************************************************
```

Figure 2.2-41 (Cont.)

```
C*****************************  ARRPTR  *****************************
C*                                                                   *
C*                        SUBROUTINE ARRPTR                          *
C*                                                                   *
C*    PURPOSE                                                        *
C*        MSM UTILITY ROUTINE.  DETERMINES POINTER TO A PARAMETER    *
C*        OF A SYSTEM SENSOR ARRAY.                                  *
C*                                                                   *
C*    CALLING SEQUENCE                                               *
C*        CALL ARRPTR (I,J, INDEX)                                   *
C*                                                                   *
C*    DESCRIPTION OF PARAMETERS                                      *
C*        INDEX (OUTPUT) WILL CORRESPOND TO THE J'TH WORD OF THE     *
C*        I'TH SYSTEM ARRAY.  THAT IS, THE J'TH WORD OF THE I'TH     *
C*        SYSTEM ARRAY CAN BE LOCATED AT MSMPAR(INDEX)               *
C*                                                                   *
C*    REMARKS                                                        *
C*        'I' REFERS TO THE PLANNER-SPECIFIED ARRAY ID.  PROGRAM     *
C*        WILL DETERMINE WHETHER THIS VALUE CORRESPONDS TO MSM1      *
C*        (UGS ARRAYS), MSM2 (STASCAN ARRAYS) OR MSM3 (MOVARRAYS),   *
C*        AND DETERMINE POINTER VALUE ACCORDINGLY.                   *
C*                                                                   *
C*        RELATED SUBROUTINE --  ARRVLU                              *
C*                                                                   *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*        NONE                                                       *
C*                                                                   *
C*********************************************************************
```

Figure 2.2-42

```
C******************************* ARRVLU ********************************
C*                                                                     *
C*                       SUBROUTINE ARRVLU                             *
C*                                                                     *
C*    PURPOSE                                                          *
C*        MSM UTILITY ROUTINE.  EXTRACTS PARAMETER VALUE FOR A         *
C*        SYSTEM SENSOR ARRAY, FROM PARAMETER STORAGE AREA MSMPAR.     *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL ARRVLU (IDARAY,IWORD, IVALUE)                           *
C*                         OR                                          *
C*        CALL ARRVLU (IDARAY,IWORD, VALUE)                            *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        IDARAY       INPUT. SPECIFIES ID OF ARRAY (WHETHER           *
C*                     UGS, STASCAN OR MOVING)                         *
C*        IWORD        INPUT. SPECIFIES WHICH WORD REQUIRED FROM       *
C*                     PARAMETER SUBLIST FOR INDICATED ARRAY.          *
C*                                                                     *
C*        IVALUE       OUTPUT.  VALUE OF WORD IDENTIFIED BY IDARAY     *
C*         OR VALUE    AND IWORD.  REAL OR INTEGER NAME CAN BE         *
C*                     USED, DEPENDING UPON FORM IN MSMPAR STORAGE.    *
C*                                                                     *
C*    REMARKS                                                          *
C*        RELATED SUBROUTINES ARE ARRPTR, PARVLU, PARPTR, TGTVLU       *
C*        AND TGTPTR.  THIS ROUTINE MAY BE CONSIDERED A MODIFICA-      *
C*        TION OF PARVLU, MORE CONVENIENT TO USE FOR ARRAYS.           *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*        ARRPTR    GETS POINTER (INDEX) OF LOC'N OF DESIRED PARAM.    *
C*                                                                     *
C***********************************************************************
```

Figure 2.2-43

```
C******************************  TGTPTR  ******************************
C*                                                                    *
C*                        SUBROUTINE TGTPTR                           *
C*                                                                    *
C*     PURPOSE                                                        *
C*         MSM UTILITY ROUTINE.  DETERMINES POINTER TO A PARAMETER    *
C*         OF A 'TARGET' (RED OR BLUE FORCE) IN STORAGE AREA MSMPAR.  *
C*                                                                    *
C*     CALLING SEQUENCE                                               *
C*         CALL TGTPTR (I,J, INDEX)                                   *
C*                                                                    *
C*     DESCRIPTION OF PARAMETERS                                      *
C*         INDEX (OUTPUT) WILL CORRESPOND TO THE J'TH WORD OF THE     *
C*         I'TH FORCE.  THAT IS, THE J'TH WORD OF THE PARAMETER       *
C*         LIST FOR THE FORCE WITH ID = I WILL BE STORED IN           *
C*         MSMPAR (INDEX).                                            *
C*                                                                    *
C*     REMARKS                                                        *
C*         NOTE 1ST ARGUMENT IS A FORCE ID.  PROGRAM WILL DETER-      *
C*         MINE WHETHER THE CORRESPONDING FORCE, OR TARGET, IS        *
C*         RED OR BLUE AND COMPENSATE FOR THE DIFFERENT ORIGINS       *
C*         AND SUBLIST LENGTHS OF RED VS. BLUE.                       *
C*                                                                    *
C*         RELATED SUBROUTINES ARE TGTVLU, PARPTR, PARVLU, ARRPTR     *
C*         AND ARRVLU.                                                *
C*                                                                    *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED.                  *
C*         NONE                                                       *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-44

2.2.5.9 Geometry and Other Auxiliary Routines

Seventeen MSM subprograms, listed below, make geometry calculations or perform other auxiliary functions:

TGTLG	Basic computation of static motion parameters for specified target and leg
TGTLXY	Gets dynamic values (i.e., for explicit time) or target coordinates
STRECT	'Stationary Rectangle' geometry calculation
STCIRC	'Stationary Circle' geometry calculation
EVNEFT	Evaluates No. of target elements in sensor field
CLOSEL	Determines closest target element (to sensor)
KSTVLU	Determines 'KSTRNG' (strength of signal index)
FTPAR1	Determines static and dynamic parameters,
FTPAR2	respectively, for false targets
BFLASK	Used for initial search (outside of loop) and
BFILUM	actual access (inside loop), battlefield illumination data for subroutine image
SCAN1	SCAN routines, for two scanning logics, used for
SCAN 2	Sensors in Sector scan mode
SETSC1	Preliminary routines (outside of loop) to set working parameters
SETSC2	for SCAN1, SCAN 2, respectively

ITODEV	Evaluates 'Time of Day' from ITIME (time of day required by some sensor routines)
IUTEVL	Evaluates IUT index (unit terrain type) from X, Y

These are briefly discussed below. Additional information is given in Fig. 2.2-45 through 2.2-48 in the internal comments within the program listings (Volume III), and by AUTOFLOWS (also Volume III).

TGTLG and TGTLXY are closely related subroutines, the former being used by the latter. Both refer to positions of a "target" (in this context, either a blue or red force; false targets are handled by different routines) on a specified path leg. TGTLG gives static parameters, and TGTLXY uses these static parameters to determine dynamic positions at specified time values.

The basic input argument to TGTLG is an integer word combining target (force) ID and leg number, in the form

IDTL = ID + 1000*(LEG NUMBER)

It should be noted that PRERUN passes to MSM target identifications in this form, and that this form is much more useful than would be just the ID itself. TGTLG then supplies the following information in one of 20 available "slots" in common area /TGLGCM/: *

XO, YO	coordinates of initial point of leg
TO	time that target (leading element) reaches (XO, YO)
XI, YI	coordinates at terminal end of leg
TI	time target reaches (XI, YI)
VX, VY	x and y components of target velocity on the indicated leg
SPEEDL	target speed on indicated leg
TLO	time length of target on the leg

* By providing a "circulating storage" for 20 sets of data, redundant calculation is greatly reduced. TGTLG does not need to calculate again any data already in one of the slots; it simply determines the correct KTL value and returns it.

```
C****************************  TGTVLU  *********************************
C*                                                                     *
C*                       SUBROUTINE TGTVLU                             *
C*                                                                     *
C*     PURPOSE                                                         *
C*         MSM UTILITY ROUTINE.  PROVIDES THE VALUE OF A PARAMETER     *
C*         FROM THE MSMPAR STORAGE AREA FOR A SPECIFIED WORD OF A      *
C*         SPECIFIED TARGET (FORCES) ID.                               *
C*                                                                     *
C*     CALLING SEQUENCE                                                *
C*         CALL TGTVLU (ID,IWORD, IVALUE)                              *
C*                       OR                                            *
C*         CALL TGTVLU (ID,IWORD, VALUE)                               *
C*                                                                     *
C*     DESCRIPTION OF PARAMETERS                                       *
C*         ID            INPUT... ID OF (BLUE OR RED) FORCE            *
C*         IWORD         INPUT... WORD NUMBER IN THE STORED LIST OF    *
C*                                PARAMETERS FOR INDICATED FORCE       *
C*         IVALUE        OUTPUT...VALUE OF PARAMETER INDICATED BY      *
C*          OR VALUE              INPUT VARIABLES.  REAL OR INTEGER    *
C*                                NAME CAN BE USED, DEPENDING UPON     *
C*                                FORM IN STORAGE.                     *
C*                                                                     *
C*     REMARKS                                                         *
C*         RELATED SUBROUTINES ARE TGTPTR, ARRVLU, ARRPTR, PARVLU,     *
C*         AND PARPTR.                                                 *
C*                                                                     *
C*         THE TARGET/FORCES ID MAY CORRESPOND EITHER TO BLUE          *
C*         FORCES (PARAMETERS IN MSM4) OR RED FORCES (PARAMETERS       *
C*         IN MSM5).  THE PROGRAM WILL DETERMINE WHICH, AND EXTRACT    *
C*         THE PARAMETER VALUE FROM THE PROPER GENERAL AREA.           *
C*                                                                     *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*         TGTPTR     GETS SPECIFIC LOCATION OF DESIRED PARAMETER      *
C*                                                                     *
C**********************************************************************
```

Figure 2.2-45

```
C****************************  TGTLG  ********************************
C*                                                                     *
C*                         SUBROUTINE TGTLG                            *
C*                          (TARGET/LEG)                               *
C*                                                                     *
C*    PURPOSE                                                          *
C*        MSM UTILITY ROUTINE FOR TARGET/LEG MOTION PARAMETER          *
C*        CALCULATIONS (BLUE AND RED FORCES AS TARGETS... NOT          *
C*        RELEVANT FOR FALSE TARGETS).  THIS ROUTINE CALCULATES        *
C*        AND STORES BASIC TARGET/LEG PARAMETERS THAT DO NOT           *
C*        HINGE ON A PARTICULAR INSTANT OF TIME.                       *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL TGTLG (IDTL)                                            *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        INPUT                                                        *
C*            IDTL       PACKED INTEGER COMBINING A TARGET             *
C*                       ID (IDF) AND LEG NUMBER (LGNO) IN             *
C*                       THE FORM                                      *
C*                               IDF + 1000*LGNO                       *
C*                                                                     *
C*        OUTPUTS (ALL IN COMMON AREA /TGLGCM/                         *
C*                                                                     *
C*                       KTL       SUBSCRIPT VALUE TO COMMON ARRAYS    *
C*                                 DEFINED BELOW, CORRESPONDING TO     *
C*                                 THE IDTL VALUE GIVEN AS INPUT.      *
C*                                 KTL WILL HAVE VALUE 1 THRU 20.      *
C*                                                                     *
C*                       THE FOLLOWING 11 VARIABLES IN /TGLGCM/ARE     *
C*                       ARRAYS WITH DIMENSION 20.  WITH THE EXCEP-    *
C*                       TION OF IDTGTL, THESE VARIABLES ARE ALL       *
C*                       REAL AND IN INTERNAL GAME UNITS               *
C*                                                                     *
C*                       IDTGLG    STORED TARGET ID/ LEG NO.           *
C*                       X0        X-COORD OF ENTRY-NODE OF LEG        *
C*                       Y0        Y-COORD OF ENTRY-NODE OF LEG        *
C*                       T0        TIME LEADING ELEMENT OF TARGET      *
C*                                    REACHES (X0,Y0)                  *
C*                       X1        X-COORD OF LEAVE-NODE OF LEG        *
C*                       Y1        Y-COORD OF LEAVE-NODE OF LEG        *
C*                       T1        TIME LEADING ELEMENT OF TARGET      *
C*                                    LEAVES (X1,Y1)                   *
C*                       TLO       TIME-LENGTH OF TARGET, SECONDS.     *
C*                                   (VARIES WITH LEG NO. AS WELL      *
C*                                    AS NOMINAL TARGET SPEED.)        
C*                       SPEEDL    SPEED OF TARGET ON SPECIFIED LEG.   *
C*                       VX        X-COMPONENT OF VELOCITY ON LEG      *
C*                       VY        Y-COMPONENT OF VELOCITY ON LEG      *
C*                                                                     *
```

Figure 2.2-46

```
C*     REMARKS                                                            *
C*          1. THIS ROUTINE USES A REVOLVING STORAGE CONCEPT.  PROVIDED   *
C*             IDTGTL(K) IS NON-ZERO, THE SET OF ARRAY VARIABLES WITH     *
C*             THE SAME SUBSCRIPT (K) IS VALID AND CONSISTENT.  HENCE,    *
C*             UP TO 20 VALID TARGET/LEG SETS CAN BE HELD SIMULTAN-       *
C*             EOUSLY IN THE COMMON AREA.                                 *
C*                                                                        *
C*          2. WHEN A PROGRAM CALLS TGTLG, THE SCALAR VARIABLE KTL THAT   *
C*             IS RETURNED INFORMS IT OF THE PROPER 'K', OR SUBSCRIPT,    *
C*             TO USE FOR THE INPUT VARIABLE IDTL.  IT IS IMPLIED THAT,   *
C*             AFTER THE CALL, IDTGTL(KTL) WILL EQUAL IDTL.               *
C*                                                                        *
C*          3. WHEN THE ROUTINE IS ENTERED, THE STORED IDTGTL VALUES      *
C*             ARE SEARCHED FOR AGREEMENT WITH IDTL. IF AGREEMENT IS      *
C*             FOUND, NO CALCULATION OR STORES ARE NECESSARY, AND         *
C*             KTL IS RETURNED WITH THE APPROPRIATE VALUE.                *
C*                                                                        *
C*          4. IF NO AGREEMENT IS FOUND, CALCULATION OCCURS.  SUB-        *
C*             SCRIPT USED IS A 'NEW' ONE IF POSSIBLE. OTHERWISE          *
C*             THE 'OLDEST' SET OF PARAMETERS IS ERASED AND REPLACED      *
C*             WITH NEW VALUES.                                           *
C*                                                                        *
C*          5. THE STORAGE CONCEPT IS A COMPROMISE BETWEEN THE FAST-      *
C*             BUT-MAXIMUM-STORAGE POSSIBILITY (STORING 11 PARAMETERS     *
C*             FOR EVERY TARGET/LEG COMBINATION) AND THE MINIMUM-         *
C*             STORAGE-BUT-VERY-SLOW POSSIBILITY OF COMPUTING PARAM-      *
C*             ETERS EVERY TIME (MULTIPLE COMPUTATION OF SAME DATA).      *
C*                                                                        *
C*          6. NOTE T1 IS THE TIME THAT LEADING ELEMENT OF TARGET         *
C*             OR FORCE LEAVES LEG.  TARGET LEAVES LEG COMPLETELY         *
C*             AT TIME T1+TLO (TRAILING ELEMENT LEAVES).                  *
C*                                                                        *
C*                                                                        *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                       *
C*         TGTPTR     PROVIDES POINTER TO A TARGET PARAMETER              *
C*         PARPTR     PROVIDES POINTER TO ARBITRARY PARAMETER             *
C*                                                                        *
C**************************************************************************
```

Figure 2.2-46 (Cont.)

```
C******************************  TGTLXY  ********************************
C*                                                                     *
C*                          SUBROUTINE TGTLXY                          *
C*                                                                     *
C*    PURPOSE                                                          *
C*        MSM AUXILIARY ROUTINE, TO DETERMINE X,Y COORDINATES OF       *
C*        THE LEADING ELEMENT OF A TARGET WITH RESPECT TO A GIVEN      *
C*        PATH LEG, THE NUMBER OF ELEMENTS AND THE SPEED OF THE        *
C*        TARGET ON THAT LEG.                                          *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL TGTLXY (IDTL,NELTOT, XTGTLE,YTGTLE,NELLEG,VELLEG)       *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        IDTL       INTEGER WORD OF FORM 1000*LEGNO+ID  (COMBI-       *
C*                     NATION OF TARGET ID AND LEG NUMBER). (INPUT)    *
C*        NELTOT     TOTAL NUMBER OF ELEMENTS IN TARGET.    (INPUT)    *
C*                                                                     *
C*        XTGTLE     COORDINATES OF LEADING ELEMENT OF TARGET SEG-     *
C*        YTGTLE       MENT ON SPECIFIED LEG.               (OUTPUT)   *
C*        NELLEG     NUMBER OF TARGET ELEMENTS ON LEG       (OUTPUT)   *
C*        VELLEG     SPEED OF TARGET ON LEG                 (OUTPUT)   *
C*        ITIME      IMPLICIT INPUT VIA COMMON.  INTEGER GAME TIME     *
C*                     VALUE FOR WHICH CALCUATIONS APPLY.              *
C*                                                                     *
C*    REMARKS                                                          *
C*        MOTION PARAMETERS REQUIRED ARE PROVIDED BY SUBROUTINE        *
C*        TGTLG, WHICH LEAVES RESULTS IN ONE OF THE SLOTS IN           *
C*        COMMON AREA /TGLGCM/.  THE INTEGER  KTL  USED THROUGH-       *
C*        OUT THE CODING BELOW IS THE SUBSCRIPT POINTER FOR            *
C*        ACCESSING /TGLGCM/.                                          *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*        TGTLG                                                        *
C*                                                                     *
C***********************************************************************
```

Figure 2.2-47

```
C******************************  STRECT  ********************************
C*                                                                     *
C*                      SUBROUTINE STRECT                              *
C*                   (STATIONARY RECTANGLE)                            *
C*                                                                     *
C*    PURPOSE                                                          *
C*        MSM UTILITY (GEOMETRY) ROUTINE.  CALCULATES BASIC            *
C*        VARIABLES RELATED TO THE INTERSECTION OF A TARGET            *
C*        (TRAVELING ON SPECIFIED PATH LEG) WITH A STATIONARY          *
C*        RECTANGLE SENSOR FIELD                                       *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL STRECT (IDTL,XC,YC,RL,RW,THETA, TLEENT,TLEEXT,TIMLT)    *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*                                                                     *
C*              * INPUT VARIABLES *                                    *
C*        IDTL       ID OF TARGET/LEG COMBINATION (IN FORM             *
C*                       IDTGT + 1000*LEGNO)                           *
C*        XC         GAME COORDINATES OF CENTER                        *
C*        YC             OF RECTANGLE                                  *
C*        RL         RECTANGLE LENGTH                                  *
C*        RW         RECTANGLE WIDTH                                   *
C*        THETA      ORIENTATION ANGLE OF RECTANGLE                    *
C*                                                                     *
C*              * OUTPUT VARIABLES *                                   *
C*        TLEENT     TIME LEADING ELEMENT (OF TARGET) ENTERS THE       *
C*                       RECTANGULAR FIELD                             *
C*        TLEEXT     TIME LEADING ELEMENT EXITS FROM RECTANGLE         *
C*        TIMLT      TIME LENGTH OF TARGET FORCE ON SPECIFIED LEG      *
C*                                                                     *
C*    REMARKS                                                          *
C*        1.  ROUTINE INTENDED ONLY FOR MOVING TARGETS.                *
C*        2.  RELATED ROUTINE IS STCIRC (FOR CIRCULAR FIELDS)          *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*        ERASE      ERASE (SET TO ZERO) BLOCK OF WORDS                *
C*        TGTLG      TARGET/LEG MOTION PARAMETER COMPUTATION           *
C*                                                                     *
C*    METHOD                                                           *
C*             INPUT VARIABLES ARE UTILIZED TO COMPUTE THE COORDINATES OF THE FOUR  *
C*        CORNERS OF THE RECTANGULAR SENSOR FIELD.  OTHER VALUES ARE USED TO COM-   *
C*        PUTE LENGTH AND VELOCITY VARIABLES RELATING TO THE FOUR SIDES OF THE REC- *
C*        TANGLE.  TWO VARIABLES, ALPHA AND TAU ARE CALCULATED WHICH ARE USED TO    *
C*        DETERMINE WHETHER A TARGET WOULD HAVE INTERSECTED THE SENSOR RECTANGULAR  *
C*        FIELD.                                                                    *
C*             WHEN THE PROPER CONDITIONS FOR THE VARIABLES KOUNT, ALPHA, AND TAU   *
C*        EXIST, TIMES THAT THE LEADING ELEMENT OF THE TARGET ENTER AND EXIT THE    *
C*        RECTANGULAR FIELD ARE COMPUTED AS IS THE TIME LENGTH OF A TARGET FORCE ON *
C*        A SPECIFIED LEG.  WHEN AN INTERSECTION IS NOT MADE, THE ENTRY AND EXIT    *
C*        TIMES ARE SET EQUAL TO ZERO.                                              *
C*                                                                                  *
C***********************************************************************************
```

Figure 2.2-48

```
C*****************************  STCIRC  ******************************
C*                                                                   *
C*                       SUBROUTINE STCIRC                           *
C*                                                                   *
C*                                                                   *
C*    PURPOSE                                                        *
C*        MSM UTILITY ROUTINE (GEOMETRY), FOR DETERMINING ENTRY      *
C*        AND EXIT [illegible] FOR A TARGET THRU A SENSOR FIELD THAT *
C*        IS A STATIONARY CIRCLE (E.G., ARFBUOY WITH GEOMETRY        *
C*        INDEX = 1)                                                 *
C*                                                                   *
C*    CALLING SEQUENCE                                               *
C*        CALL STCIRC (IDTL,XC,YC,RC, TLEENT,TLEEXT,TLTIM)           *
C*                                                                   *
C*    DESCRIPTION OF PARAMETERS                                      *
C*            * INPUT *                                              *
C*        IDTL       ID OF TARGET/LEG                                *
C*        XC         COORDINATES OF CENTER                           *
C*        YC            OF STATIONARY CIRCLE                         *
C*        RC         RADIUS OF CIRCLE                                *
C*            * OUTPUT *                                             *
C*        TLEENT     TIME THAT LEADING ELEMENT                       *
C*                      OF TARGET ENTERS CIRCLE                      *
C*        TLEEXT     TIME THAT LEADING ELEMENT                       *
C*                      EXITS FROM CIRCLE                            *
C*        TLTIM      TARGET TIME LENGTH IN SECONDS                   *
C*                                                                   *
C*    REMARKS                                                        *
C*        1. THE 'TARGET' MAY BE ANY MOVING BLUE OR RED FORCE.       *
C*           THIS ROUTINE NOT APPLICABLE FOR FALSE TARGETS OR        *
C*           FOR STATIONARY TARGETS.                                 *
C*                                                                   *
C*        2. IF THE TARGET DOES NOT IN FACT INTERSECT THE CIRCLE     *
C*           WHEN ON THE INDICATED PATH LEG, THE OUTPUT VARIABLES    *
C*           TLEENT AND TLEEXT WILL BOTH BE SET TO 0.0.              *
C*                                                                   *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*        TGTLG      CALCULATES PARAMETERS FOR TARGET/LEG.  RESULTS  *
C*                      ACCESSED VIA /TGLGCM/ COMMON AREA.           *
C*                                                                   *
C*    METHOD                                                         *
C*            THE TGLGCM COMMON AREA IS ACCESSED TO OBTAIN THE SPECIFIC PARAME-  *
C*        TERS REQUIRED.  THE INTEGER KTL IS THE SUBSCRIPT POINTER FOR ACCESSING  *
C*        TGLGCM.  CALCULATIONS ARE MADE INVOLVING DISTANCES, VELOCITIES AND TIMES.  *
C*        THE VARIABLE DISCR (DISCRIMINANT) IS DETERMINED AND IF THIS VALUE IS  *
C*        LESS THAN OR EQUAL TO ZERO, THE SUBPROGRAM IS EXITED AND THE LEG HAS  *
C*        NO VALID INTERSECTION WITH THE CIRCLE.  THUS, THE TIMES THAT THE LEADING  *
C*        TARGET ELEMENT ENTERS AND EXITS THE CIRCLE ARE SET EQUAL TO ZERO.  *
C*            IF THE DISCRIMINANT VALUE IS GREATER THAN ZERO THE EXTENDED PATH  *
C*        LEG AT LEAST INTERSECTS THE CIRCLE.  HOWEVER, IT MUST STILL BE DETER-  *
C*        MINED WHETHER THE ACTUAL PATH LEG INTERSECTS THE CIRCLE.  ENTRY AND EXIT  *
C*        TIMES ARE COMPUTED AND IF THE RELATIONSHIP IS SUCH THAT THE TIME THAT THE  *
C*        LEADING ELEMENT OF TARGET ENTERS THE CIRCLE IS LESS THAN THE EXIT TIME  *
C*        A VALID INTERSECTION HAS OCCURRED.  SUCH ENTRY AND EXIT TIMES ARE THEN  *
C*        RETURNED TO THE MSM.                                       *
C*********************************************************************
```

Figure 2.2-49

```
C******************************** EVNEFD *********************************
C*                                                                        *
C*                    SUBROUTINE EVNEFD                                   *
C*             (EVALUATE NO. ELEMENTS IN FIELD)                           *
C*                                                                        *
C*    PURPOSE                                                             *
C*       MSM AUXILIARY ROUTINE, TO EVALUATE THE NUMBER OF TARGET          *
C*       ELEMENTS WITHIN AN ARFBUOY SENSOR FIELD (MOVING TARGETS,         *
C*       RED OR BLUE, ONLY).                                              *
C*                                                                        *
C*       WITH PROPER PREPARATION OF VARIABLES IN COMMON, EVNEFD           *
C*       COULD ALSO BE USED FOR SENSORS OTHER THAN ARFBUOY.               *
C*                                                                        *
C*    CALLING SEQUENCE                                                    *
C*       CALL EVNEFD (IDTL,NOELEM, NELFLD,VELTAR)                         *
C*                                                                        *
C*    DESCRIPTION OF PARAMETERS                                           *
C*            * OUTPUT VARIABLES *                                        *
C*       NELFLD    NO. OF TARGET ELEMENTS IN SENSOR COVERAGE FIELD        *
C*       VELTAR    SPEED OF TARGET ON GIVEN LEG                           *
C*                                                                        *
C*            * INPUT VARIABLES, EXPLICIT *                               *
C*       IDTL      TARGET LEG ID  (IDTGT + 1000*LEGNO)                    *
C*       NOELEM    TOTAL NO. OF ELEMENTS IN TARGET FORCE                  *
C*                                                                        *
C*            * INPUT VARIABLES, IMPLICIT VIA COMMON *                    *
C*       ITIME     INTEGER GAME TIME (VIA /BASICT/)                       *
C*       XC        X,Y COORDINATES OF CENTER OF SENSOR                    *
C*       YC           FIELD (VIA /SENSOR/)                                *
C*       THETA     ORIENTATION ANGLE (VIA /SENSOR/). USED                 *
C*                    ONLY FOR OPEN RECTANGLE OPTION, IGEOM = 2.          *
C*       IGEOM     GEOMETRY INDEX AS DEFINED FOR ARFBUOY FIELDS           *
C*                    (VIA /SENSOR/)...                                   *
C*                         IGEOM = 1   OPEN CIRCLE                        *
C*                               = 2   OPEN RECTANGLE                     *
C*                               = 3   PATH LOCATION                      *
C*       DIMMAX    LENGTH OF RECTANGLE (IGEOM = 2,3), OR DIAMETER         *
C*                    OF CIRCLE (IGEOM = 1)                               *
C*       WIDTH     WIDTH OF RECTANGLE (MEANINGFUL IN MSM ONLY             *
C*                    FOR IGEOM = 2)                                      *
C*                                                                        *
C*    REMARKS                                                             *
C*       1. THIS ROUTINE VALID FOR MOVING RED OR BLUE FORCES RE-          *
C*            GARDED AS TARGETS.                                          *
C*       2. ERROR DIAGNOSTIC PRINTED IF STATIONARY TARGET IMPLIED         *
C*            BY THE IDTL INPUT.  PROGRAM THEN CONTINUED WITH             *
C*            NELFLD SET TO ZERO.                                         *
C*                                                                        *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                        *
C*       TGTLG     TARGET/LEG BASIC COMPUTATIONS                          *
```

Figure 2.2-50

```
C*          STCIRC     CALLED IF IGEOM = 1 (CIRCULAR FIELD)                    *
C*          STRECT     CALLED IF IGEOM = 2 (OPEN RECTANGLE)                    *
C*          PGSKIP     PAGE SKIP CONTROL (POSSIBLY CALLED IF                   *
C*                       DIAGNOSTIC IS PRINTED)                                *
C*                                                                             *
C*          METHOD                                                             *
C*               THE SPEED OF A TARGET ON A SPECIFIED LEG IS OBTAINED THROUGH THE  *
C*          USE OF THE KTL SUBSCRIPT VALUE IN ACCESSING THE COMMON AREA TGLGCM.  IF  *
C*          THE SPEED VALUE EQUALS ZERO AN ERROR DIAGNOSTIC IS PRINTED AND THE  *
C*          NUMBER OF ELEMENTS IN THE FIELD IS SET EQUAL TO ZERO.  IF THE TARGETS  *
C*          ARE OF A MOVING TYPE, AN IGEOM VARIABLE IS CHECKED TO SEE WHETHER THE  *
C*          SENSOR FIELD IS AN OPEN CIRCLE, OPEN RECTANGLE, OR PATH LOCATION.  *
C*               WHEN IGEOM = 1, THE SENSOR FIELD IS AN OPEN CIRCLE AND SUBROUTINE  *
C*          STCIRC IS CALLED.  THE THREE VARIABLES, TLEENT, TLEEXT, AND TIMLT ARE  *
C*          RETURNED TO THE PROGRAM EVNEFD AND FURTHER CALCULATIONS ARE MADE TO DE-  *
C*          TERMINE THE NUMBER OF ELEMENTS IN THE FIELD.  TWO TIME VARIABLES (TIMEP  *
C*          AND TIMEM) ARE COMPARED AND IF TIMEP IS LESS THAN TIMEM THE NUMBER OF  *
C*          ELEMENTS IN THE FIELD IS SET EQUAL TO ZERO.  IF TIMEP IS GREATER THAN  *
C*          TIMEM AND TIMLT IS GREATER THAN ZERO, A CALCULATION IS MADE TO DETERMINE  *
C*          THE NUMBER OF ELEMENTS IN THE FIELD.                               *
C*               WHEN IGEOM = 2, THE SENSOR FIELD IS AN OPEN RECTANGLE AND SUB-  *
C*          ROUTINE STRECT IS CALLED.  THE THREE VARIABLES, TLEENT, TLEEXT, AND  *
C*          TIMLT ARE RETURNED TO THE PROGRAM EVNEFD AND THE SAME CALCULATIONS  *
C*          DISCUSSED ABOVE ARE MADE.                                          *
C*               WHEN IGEOM = 3, A PATH LOCATION IS DENOTED.  CALCULATIONS ARE MADE  *
C*          INVOLVING DISTANCES, SPEEDS, AND TIMES.  THIS LEADS TO THE COMPUTATION  *
C*          OF THE THREE VARIABLES, TLEENT, TLEEXT, AND TIMLT.  AT THIS POINT, THE  *
C*          PROGRAM CONTINUES IN THE SAME MANNER AS PREVIOUSLY DISCUSSED FOR IGEOM  *
C*          VALUES OF 1 AND 2.                                                 *
C*                                                                             *
C*******************************************************************************
```

Figure 2.2-50 (Cont)

```
C*****************************  CLOSEL  ******************************
C*                                                                   *
C*                          SUBROUTINE CLOSEL                        *
C*                        ('CLOSEST ELEMENT')                        *
C*                                                                   *
C*    PURPOSE                                                        *
C*        MSM ROUTINE,PRIMARILY ESTABLISHED TO SUPPORT THE           *
C*        GEOMETRY LINKAGE FOR THE MAGNETIC SENSOR ROUTINE (MAGTG)   *
C*        AND ITS CONTROL ROUTINE EX3MAG.  PROVIDES THE VALUE OF     *
C*        NCLEL (WHICH ELEMENT OF A TARGET IS CLOSEST TO THE         *
C*        SENSOR)                                                    *
C*                                                                   *
C*    CALLING SEQUENCE                                               *
C*        CALL CLOSEL (IDTL,NOELEM,XSNSR,YSNSR, NCLEL)               *
C*                                                                   *
C*    DESCRIPTION OF PARAMETERS                                      *
C*        IDTL       COMBINED TARGET (RED OR BLUE FORCE) ID AND      *
C*                     LEG NUMBER... IN FORM ID+1000*LEGNO.          *
C*        NOELEM     TOTAL NUMBER OF ELEMENTS IN TARGET FORCE        *
C*        XSNSR      X,Y COORDINATES OF SENSOR (OR ARBITRARY         *
C*        YSNSR        REFERENCE POINT)                              *
C*                                                                   *
C*        ITIME      INTEGER GAME TIME (IMPLICIT INPUT VIA COMMON)   *
C*                                                                   *
C*        NCLEL      OUTPUT. INTEGER VALUE FROM 1 TO NOELEM, CORR-   *
C*                     ESPONDING TO WHICH TARGET ELEMENT IS CLOSEST  *
C*                     TO (XSNSR,YSNSR) AT TIME = ITIME.             *
C*                                                                   *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*        TGTLG      PROVIDES BASIC TARGET/LEG MOTION PARAMETERS,    *
C*                   ACCESSED AFTER CALL FROM COMMON /TGLGCM/.       *
C*                                                                   *
C*    METHOD                                                         *
C*        IF A TARGET, REGARDED AS A MOVING LINE SEGMENT, INCLUDES   *
C*        THE POINT OF CLOSEST APPROACH, A PARAMETER ALPHA (BETWEEN  *
C*        0.0 AND 1.0) IS CALCULATED THAT SPECIFIES THE POINT ON     *
C*        THE TARGET AT THE CLOSEST APPROACH.  ALPHA = 0. WOULD COR- *
C*        RESPOND TO LEADING ELEMENT OF TARGET (HENCE NCLEL = 1),    *
C*        ALPHA = 1.0 WOULD CORRESPOND TO TRAILING ELEMENT OF THE    *
C*        TARGET (HENCE NCLEL = NOELEM).                             *
C*          IF THE SEGMENT DOES NOT INCLUDE THE POINT OF CLOSEST     *
C*        APPROACH, THEN EITHER ENDPOINT MUST BE CLOSEST TO          *
C*        (XSNSR,YSNSR). THIS POSSIBILITY IS AUTOMATICALLY ACCOUNTED *
C*        FOR BY SETTING ALPHA TO 1.0 IF THE NOMINAL VALUE EXCEEDS   *
C*        1.0, OR TO 0.0 IF THE NOMINAL VALUE IS LESS THAN 0.0.      *
C*                                                                   *
C*********************************************************************
```

Figure 2.2-31

```
C******************************  KSTVLU  ******************************
C*                                                                    *
C*                         FUNCTION KSTVLU                            *
C*                                                                    *
C*     PURPOSE                                                        *
C*         MSM AUXILIARY ROUTINE (FUNCTION TYPE) TO EVALUATE THE      *
C*         'KSTRNG' PARAMETER (MEASURE OF TARGET'S 'STRENGTH' IN      *
C*         CREATING SIGNAL TO SENSOR).                                *
C*                                                                    *
C*     CALLING SEQUENCE                                               *
C*         KSTRNG = KSTVLU (IDTGT)                                    *
C*                                                                    *
C*     DESCRIPTION OF PARAMETERS                                      *
C*         IDTGT       ID OF A RED OR BLUE FORCE, REGARDED AS A       *
C*                     SENSOR TARGET                                  *
C*                                                                    *
C*     REMARKS                                                        *
C*         THIS ROUTINE ACCESSES 'TARGET' INFORMATION FROM COMMON     *
C*         /BIGSTR/, FOR RED/BLUE TARGETS ONLY.  NOT APPLICABLE       *
C*         FOR FALSE TARGETS.                                         *
C*                                                                    *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                   *
C*         TGTPTR      MSM UTILITY, TO DETERMINE POINTERS TO RED      *
C*                     OR BLUE FORCES DATA                            *
C*         PARPTR      MSM UTILITY, TO DETERMINE POINTERS TO GENERAL  *
C*                     PARAMETER SET                                  *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-52

```
C*****************************  FTPAR1  *******************************
C*                                                                    *
C*                      SUBROUTINE FTPAR1                             *
C*                                                                    *
C*   PURPOSE                                                          *
C*       PROVIDES STATIC PARAMETERS (AS DEFINED BELOW) FOR A          *
C*       FALSE TARGET.                                                *
C*                                                                    *
C*   CALLING SEQUENCE                                                 *
C*       CALL FTPAR1 (LISTF,ITYPSN,  IDCODE,IDTGT,ITGTTP,KSTRNG,NEL)  *
C*                   (KSTRNG,NEL,SPACE)                               *
C*                                                                    *
C*   DESCRIPTION OF PARAMETERS                                        *
C*       * INPUT VARIABLES *                                          *
C*       LISTF     DEFINES A 12-WORD ARRAY OF FALSE TARGET PARAM-     *
C*                    ETERS (IN PRACTICE, A SUBLIST OF A TYPE 1       *
C*                    EVENT LIST)                                     *
C*       ITYPSN    GENERIC TYPE CODE FOR SENSOR THAT WILL USE THE     *
C*                    FALSE TARGET DATA (AFFECTS COMPUTATION OF       *
C*                    THE KSTRNG PARAMETER)                           *
C*                                                                    *
C*       * OUTPUT VARIABLES *                                         *
C*       IDCODE    ANALOGOUS TO 'IDTL' FOR RED/BLUE TARGETS           *
C*       IDTGT     AN ID CODE OF TYPE DEFINED FOR FALSE TARGETS       *
C*       ITGTTP    TARGET GENERIC TYPE CODE AS USED BY SENSOR         *
C*                    ROUTINES (E.G, 1 FOR PERSONNEL OR ANIMALS,      *
C*                    2 FOR VEHICLES, ... )                           *
C*       KSTRNG    'STRENGTH' CODE, AS USED BY SENSOR ROUTINES        *
C*       NEL       NUMBER OF ELEMENTS IN TARGET                       *
C*       SPACE     SPACING BETWEEN TARGET ELEMENTS                    *
C*                                                                    *
C*   REMARKS                                                          *
C*       1.  FTPAR1 IS AN MSM AUXILIARY ROUTINE, PRIMARILY USED       *
C*           BY THE EX3... SERIES OF PROGRAMS.                        *
C*                                                                    *
C*       2.  RELATED TO THIS ROUTINE IS SUBROUTINE FTPAR2, WHICH      *
C*           DETERMINES DYNAMIC PARAMETERS OF FALSE TARGETS.          *
C*                                                                    *
C*       3.  NEL, SPACE ARE NOT EXPLICITLY PROVIDED DATA. ESTI-       *
C*           MATES ARE INFERRED FROM DATA THAT ARE AVAILABLE.         *
C*                                                                    *
C*   SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*       NONE                                                         *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-53

```
C******************************  FTPAR2  ******************************
C*                                                                    *
C*                       SUBROUTINE FTPAR2                            *
C*                                                                    *
C*    PURPOSE                                                         *
C*       PROVIDES DYNAMIC PARAMETERS (AS DEFINED BELOW) FOR A         *
C*       FALSE TARGET.                                                *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*       CALL FTPAR2 (LISTF,TIME, X,Y,VX,VY,SPEED)                    *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*       * INPUT VARIABLES *                                          *
C*       LISTF     DEFINES A 12-WORD ARRAY OF FALSE TARGET PARAM-     *
C*                   ETERS (IN PRACTICE, A SUBLIST OF A TYPE 1        *
C*                   EVENT LIST)                                      *
C*       TIME      TIME (GAME TIME IN SECONDS, REAL)                  *
C*                                                                    *
C*       * OUTPUT VARIABLES (ALL EVALUATED AT GIVEN TIME) *           *
C*       X         COORDINATES OF                                     *
C*       Y           FALSE TARGET (REAL, METERS)                      *
C*       VX        X AND Y COMPONENTS                                 *
C*       VY          OF TARGET VELOCITY (REAL, METERS/SEC)            *
C*       SPEED     SPEED (REAL, METERS/SEC)                           *
C*                                                                    *
C*    REMARKS                                                         *
C*       1.  FTPAR2 IS AN MSM AUXILARY ROUTINE, PRIMARILY USED        *
C*           BY THE EX3... SERIES OF PROGRAMS.                        *
C*                                                                    *
C*       2.  RELATED TO THIS ROUTINE IS SUBROUTINE FTPAR1, WHICH      *
C*           DETERMINES STATIC PARAMETERS OF FALSE TARGETS.           *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*       NONE                                                         *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-54

```
C****************************** BFIASK ****************************
C*                                                                 *
C*                     SUBROUTINE BFIASK                           *
C*                                                                 *
C*   PURPOSE                                                       *
C*       SERVES TO MAKE PRELIMINARY SEARCH OF BATTLEFIELD ILLUM-   *
C*       INATION DATA IN MSM14, TO DETERMINE (A) IF ANY ILLUMI-    *
C*       NATION EVENTS EXIST IN GIVEN TIME INTERVAL, (B) IF SO,    *
C*       TO DETERMINE EARLIEST AND LATEST TIMES WITHIN THE         *
C*       INTERVAL THAT ANY ILLUMINATION EVENT COULD OCCUR, AND     *
C*       NUMBER OF SUCH EVENTS.                                    *
C*                                                                 *
C*   CALLING SEQUENCE                                              *
C*       CALL BFIASK (ITIM1,ITIM2, ILLUM,NLMAX,TILMIN,TILMAX)      *
C*                                                                 *
C*   DESCRIPTION OF PARAMETERS                                     *
C*       * INPUT *                                                 *
C*       ITIM1      INTEGER TIME VALUES, BEGIN-                    *
C*       ITIM2         NING AND END OF TIME INTERVAL               *
C*                                                                 *
C*       * OUTPUT*                                                 *
C*       ILLUM      LOGICAL VARIABLE. TRUE IF ANY ILLUMINATION     *
C*                     EXISTS DURING ITIM1 TO ITIM2.  IF ILLUM IS  *
C*                     FALSE, VALUES FOR FOLLOWING THREE VARIABLES *
C*                     ARE SET TO ZERO.  OTHERWISE...              *
C*       NLMAX      COUNT OF NUMBER OF DISTINCT ILLUM'N EVENTS     *
C*                     ACTIVE DURING ITIM1 TO ITIM2                *
C*       TILMIN     EARLIEST POSSIBLE TIME (REAL) WITHIN TIME INTER- *
C*                     VAL THAT ILLUMINATION CAN OCCUR             *
C*       TILMAX     LATEST POSSIBLE TIME (REAL) WITHIN INTERVAL    *
C*                     THAT ILLUMINATION CAN OCCUR                 *
C*                                                                 *
C*   REMARKS                                                       *
C*       1.  THIS ROUTINE CALLED BY SUBROUTINE EX3IMG              *
C*       2.  DATA ON WHICH THIS ROUTINE OPERATES ARE IN 20         *
C*          BLOCKS, 8 WORDS EACH, IN MSM14.  FOR FULL FORMAT       *
C*          DESCRIPTION SEE COMMENTS FOR SUBROUTINES BFILUM        *
C*          OR EX2BFI.                                             *
C*                                                                 *
C*   SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                  *
C*       NONE                                                      *
C*                                                                 *
C*******************************************************************
```

Figure 2.2-55

```
C*****************************  BFILUM  *******************************
C*                                                                     *
C*                        SUBROUTINE BFILUM                            *
C*                   (BATTLEFIELD ILLUMINATION)                        *
C*                                                                     *
C*    PURPOSE                                                          *
C*        MS ROUTINE, CALLED BY EX3IMG, TO PROVIDE INFORMATION         *
C*        ABOUT BATTLEFIELD ILLUMINATION (FLARES, INDIRECT SEARCH-     *
C*        LIGHT) FOR USE BY IMAGE SUBROUTINE.                          *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*        CALL BFILUM (LITLST)                                         *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*                                                                     *
C*        LITLST       AN ARRAY OF DIMENSION 11 IN CALLING PROGRAM,    *
C*                    INTO WHICH BFILUM PLACES ITS OUTPUT INFOR-       *
C*                    MATION.                                          *
C*                      LITLST(1)  WILL CONTAIN COUNT OF NUMBER OF     *
C*                                   SIMULTANEOUSLY ACTIVE LIGHT       *
C*                                   SOURCES. IF 0, REMAINDER OF       *
C*                                   LITLST ARRAY IS IRRELEVENT.       *
C*                      LITLST(2),LITLST(3),....                       *
C*                                 WILL CONTAIN POINTER VALUES TO      *
C*                                 MSMPAR FOR THOSE ACTIVE LIGHT       *
C*                                 SOURCES (HOWEVER MANY ARE INDI-     *
C*                                 CATED BY LITLST(1)).                *
C*                                                                     *
C*        ITIME       IMPLICIT INPUT PARAMETER, PASSED TO BFILUM VIA   *
C*                    LABELED COMMON /BASICT/.                         *
C*                                                                     *
C*    REMARKS                                                          *
C*                                                                     *
C*        1. EXAMPLE. SUPPOSE LITLST(1) = 2, LITLST(2) = 16300,        *
C*             AND LITLST(3) = 16324.  THEN THERE ARE 2 ACTIVE         *
C*             LIGHT SOURCES AT ITIME.  PARAMETER LIST (8 WORDS        *
C*             LONG) FOR THE FIRST BEGINS AT MSMPAR(16300). PAR-       *
C*             AMETER LIST FOR THE SECOND BEGINS AT MSMPAR(16324).     *
C*                                                                     *
C*        2. PARAMETER LISTS FOR LIGHT SOURCES ARE 8 WORDS LONG,       *
C*             AS FOLLOWS--                                            *
C*                                                                     *
C*                   WORD                                              *
C*                     1    ALPHANUMERIC ONE-WORD IDENTIFICATION       *
C*                     2    ITIM1  TIME ILLUM BEGINS (INTEGER)         *
C*                     3    ITIM2  TIME ILLUM EXPIRES (  "  )          *
C*                     4    X      GAME COORDINATES OF ILLUMINA-       *
C*                     5    Y        TION SOURCE  (REAL, METERS)       *
C*                     6    H      HEIGHT ABOVE GROUND OF ILLUMINATION *
C*                                   SOURCE AT TIME = ITIM1 (REAL,METERS)*
```

Figure 2.2-56

```
C*                    7    AINTNS  INTENSITY OF SOURCE  (REAL)                 *
C*                    8    MODE    MODE-OF-ILLUMINATION                        *
C*                                                                             *
C*        3. NOTE. THE HEIGHT VARIABLE MAY BE OVERRIDDEN BY CEILING            *
C*           HEIGHT FOR CERTAIN MODE VALUES.                                   *
C*                                                                             *
C*        4. THE DIMENSION (11) FOR LITLST LIMITS PROGRAM TO AT MOST           *
C*             10 SIMULTANEOUSLY ACTIVE ILLUMINATION SOURCES. IF               *
C*             MORE THAN 10 ARE IN OPERATION, THE FIRST 10 ENCOUNT-            *
C*             ERED IN SEARCH WILL BE PASSED (LITLST(1) WILL BE 10),           *
C*             AND A DIAGNOSTIC MESSAGE WILL BE PRINTED.                       *
C*                                                                             *
C*        5. EX2BFL IS A RELATED SUBROUTINE AT EXEC LEVEL 2 ,                  *
C*             HANDLING STORAGE OF ILLUMINATION EVENT DATA INTO                *
C*             MSMPAR.                                                         *
C*                                                                             *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                             *
C*        PGSKIP    PAGE SKIP CONTROL                                          *
C*        ERASE     INITIALIZES ARRAYS TO ZERO                                 *
C*                                                                             *
C*******************************************************************************
```

Figure 2.2-56 (Cont.)

```
C****************************** SCAN1 ********************************
C*                                                                    *
C*                          SUBROUTINE SCAN1                          *
C*                                                                    *
C*    PURPOSE                                                         *
C*        DETERMINES WHETHER A TARGET POSITION IS WITHIN AN           *
C*        'ILLUMINATION CELL' FOR A SENSOR WITH A SCANNING            *
C*        PATTERN OF 'TYPE1'.                                         *
C*                                                                    *
C*    DEFINITION                                                      *
C*        A 'TYPE 1' SCAN IS DEFINED BY:                              *
C*            1. FOR A FIXED RANGE WINDOW, ONE OR MORE (NREPRI)       *
C*               AZIMUTH SWEEPS ARE MADE.  AN AZIMUTH SWEEP IS        *
C*               DEFINED IN THIS CONTEXT AS A ONE-WAY TRAVEL          *
C*               FROM ONE AZIMUTH LIMIT TO THE OTHER, REGARDLESS      *
C*               OF DIRECTION.                                        *
C*            2. AFTER THE APPROPRIATE NUMBER OF AZIMUTH SWEEPS,      *
C*               THE RANGE WINDOW IS ADVANCED BY THE SPECIFIED        *
C*               RANGE GATE OR INCREMENT, EXCEPT THAT THE NEXT        *
C*               VALUE AFTER 'RMAX' IS 'RMIN' --  RANGE SWEEPS        *
C*               ARE ALWAYS FROM MIN TO MAX.                          *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL SCAN1 (TIME,RTGT,AZTGT, IHIT)                          *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        * INPUT, EXPLICIT *                                         *
C*        TIME       GAME TIME, REAL, SECONDS                         *
C*        RTGT       RANGE TO TARGET FROM SENSOR, METERS              *
C*        AZTGT      AZIMUTH ANGLE OF TARGET RELATIVE TO SENSOR (REAL, *
C*                     RADIANS, MATH. CONVENTION FOR 0 (EAST) AND     *
C*                     POSITIVE DIRECTION (CC))                       *
C*                                                                    *
C*        * OUTPUT *                                                  *
C*        IHIT       = 1 IF TARGET IS ILLUMINATED                     *
C*                   = 0 OTHERWISE                                    *
C*                                                                    *
C*        IMPLICIT INPUT PARAMETERS FROM COMMON /COMSC1/              *
C*                                                                    *
C*    REMARKS                                                         *
C*        1.  MSM ROUTINE, CALLED BY EX3ROR, EX3IMG AND EX3THV WHEN   *
C*            SENSOR IS DEFINED TO HAVE TYPE 1 SECTOR SCAN.           *
C*                                                                    *
C*        2.  THE STATIC WORKING PARAMETERS FOR SCAN1, THAT ARE       *
C*            ACCESSED FROM COMMON /COMSC1/, MUST BE PLACED THERE     *
C*            BY A PRIOR CALL TO SETSC1.                              *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        NONE                                                        *
C*                                                                    *
```

Figure 2.2-57

```
C*                                                                    *
C*    METHOD                                                          *
C*              TIME VARIABLES ARE COMPUTED THAT ARE USED IN LATER CALCU-   *
C*         LATIONS.  BOTH LOWER AND MAXIMUM LIMITS OF THE RANGE BUCKET ARE  *
C*         COMPUTED AND A CHECK IS MADE ON THE RANGE TO THE TARGET.  IF THE *
C*         RANGE TO THE TARGET IS LESS THAN THE MINIMUM OR GREATER THAN THE *
C*         MAXIMUM RANGE BUCKET LIMITS, THE VARIABLE IHIT IS SET EQUAL TO   *
C*         ZERO (TARGET IS NOT ILLUMINATED) AND THE PROGRAM IS EXITED.      *
C*              IF THE RANGE TO TARGET IS WITHIN THE ABOVE LIMITS, AN AZI-  *
C*         MUTH CHECK TO WITHIN HALF A BEAMWIDTH OF THE BEAM CENTER IS CON- *
C*         DUCTED.  CALCULATIONS ARE MADE INVOLVING COVERAGE ANGLE, AZIMUTH *
C*         SWEEPS, AZIMUTH SWEEP RATE, ETC. THAT LEAD TO A STEP COMPARING   *
C*         THE MODULUS OF ABSOLUTE (ABS) (AZTGT-AZ, TWOPI) WITH BEAMWIDTH/2.0*
C*         IF THE MODULUS VALUE IS GREATER, IHIT = 0; IF THE MODULUS VALUE  *
C*         IS LESS THAN BEAMWIDTH/2.0, IHIT = 1 AND TARGET ILLUMINATION HAS *
C*         OCCURRED.                                                        *
C*                                                                    *
C*********************************************************************
```

Figure 2.2-57 (Cont.)

```
C*****************************  SCAN2  ******************************
C*                                                                  *
C*                        SUBROUTINE SCAN2                          *
C*                                                                  *
C*   PURPOSE                                                        *
C*       DETERMINES WHETHER A TARGET POSITION IS WITHIN AN          *
C*       'ILLUMINATION CELL' FOR A SENSOR WITH A SCANNING           *
C*       PATTERN OF "TYPE 2".                                       *
C*                                                                  *
C*   DEFINITION                                                     *
C*       A "TYPE 2" SCAN IS DEFINED BY:                             *
C*           1.  FOR FIXED AZIMUTH WINDOW, A RANGE GATE IS SWEPT AT *
C*               SPECIFIED SWEEP RATE FROM SPECIFIED MINIMUM TO     *
C*               MAXIMUM RANGE VALUES                               *
C*           2.  AFTER EACH RANGE SWEEP, THE AZIMUTH WINDOW IS MOVED *
C*               BY ONE INCREMENT.                                  *
C*           3.  WITH REGARD TO AZIMUTH, THE STEPPED VALUES GO IN   *
C*               SEQUENCE FROM MINIMUM-TO-MAXIMUM, MAXIMUM-TO-MIN-  *
C*               IMUM, THEN REPEAT.                                 *
C*                                                                  *
C*   CALLING SEQUENCE                                               *
C*       CALL SCAN2 (TIME,RTGT,AZTGT,  IHIT)                        *
C*                                                                  *
C*   DESCRIPTION OF PARAMETERS                                      *
C*       * INPUT (EXPLICIT) *                                       *
C*       TIME      GAME TIME (REAL, SECONDS)                        *
C*       RTGT      RANGE TO TARGET FROM SENSOR (REAL, METERS)       *
C*       AZTGT     AZIMUTH OF TARGET REL. TO SENSOR (REAL, RADIANS) *
C*                    MATHEMATICAL CONVENTION FOR ZERO (=EAST) AND  *
C*                    POSITIVE DIRECTION (COUNTER-CLOCKWISE)        *
C*                                                                  *
C*       * OUTPUT *                                                 *
C*       IHIT        = 1 IF TARGET IS ILLUMINATED                   *
C*                   = 0 OTHERWISE                                  *
C*                                                                  *
C*       IMPLICIT INPUT PARAMETERS FROM COMMON /COMSC2/             *
C*                                                                  *
C*   REMARKS                                                        *
C*       1.  MSM ROUTINE, USED BY EX3MOR, EX3IMG AND EX3THV WHEN    *
C*           SENSOR IS DEFINED TO HAVE A TYPE 2 SECTOR SCAN.        *
C*                                                                  *
C*       2.  THE STATIC WORKING PARAMETERS FOR SCAN2, THAT ARE      *
C*           ACCESSED FROM COMMON /COMSC2/, ARE INSERTED THERE      *
C*           BY SUBROUTINE SETSC2.                                  *
C*                                                                  *
C*           ALTHOUGH THE FUNCTIONS OF SETSC2 AND SCAN2 COULD BE    *
C*           ACCOMMODATED INTO A SINGLE SUBPROGRAM, PRACTICAL       *
C*           USAGE IMPLIES EXTREMELY LARGE NUMBER OF CALLS TO       *
C*           SCAN2 FOR CERTAIN WIDE-COVERAGE SENSORS. BY ASSIGN-    *
C*           ING STATIC-PARAMETER EVALUATIONS TO SETSC2, WHICH      *
```

Figure 2.2-58

```
C*                MAY BE CALLED 'OUTSIDE THE LOOP',  SCAN2 CAN OPERATE      *
C*                WITH REDUCED COMPUTING TIME.                              *
C*                                                                          *
C*           3.  SCAN2 IS ONE OF THE TWO POSSIBLE SECTOR SCAN ROUTINES.     *
C*                                                                          *
C*      SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                        *
C*           NONE                                                           *
C*                                                                          *
C*      METHOD                                                              *
C*                UTILIZING THE RANGE LIMITS, RANGE SWEEP RATE, AND MODULUS FUNC-  *
C*           TION, THE VARIABLE R IS COMPUTED.  IF THE ABSOLUTE VALUE OF THE RANGE *
C*           TO THE TARGET MINUS R IS GREATER THAN THE RANGE GATE/2 THE PROGRAM IS *
C*           EXITED WITH NO TARGET ILLUMINATION.                            *
C*                HOWEVER, IF THE ANSWER TO THE ABOVE QUESTION IF FALSE, FURTHER  *
C*           CALCULATIONS ARE MADE USING RANGE SWEEP RATE, RANGE LIMITS, COVERAGE *
C*           ANGLE, INCREMENT IN AZIMUTH ANGLE, ORIENTATION ANGLE, MODULUS FUNC-  *
C*           TION AND THE INTEGER FUNCTION.  AT THIS TIME, THE MODULUS OF THE     *
C*           ABSOLUTE VALUE OF AZ-AZTGT, TWOPI IS CHECKED AGAINST AZIBY2 (INCRE-  *
C*           MENT IN AZIMUTH ANGLE/2.)IF THE MODULUS VALUE IS LESS THAN AZIBY2,   *
C*            IHIT = 1 AND TARGET ILLUMINATION OCCURS.  IF THE MODULUS VALUE IS   *
C*           GREATER, IHIT = 0, AND THE PROGRAM IS EXITED.                  *
C*                                                                          *
C***************************************************************************
```

Figure 2.2-58 (Con.t)

```
C*****************************  SETSC1  ********************************
C*                                                                     *
C*                        SUBROUTINE SETSC1                            *
C*                                                                     *
C*    PURPOSE                                                          *
C*       PRESETS WORKING PARAMETERS FOR SCANNING ROUTINE SCAN1.        *
C*                                                                     *
C*    CALLING SEQUENCE                                                 *
C*       CALL SETSC1 (NREPRI,RINCR,OMEGA,BEAMW,ISITYM,RMIN,RMAX,       *
C*                    AZCTR,CVANGL)                                    *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*       NREPRI    NUMBER OF AZIMUTH SWEEPS FOR WHICH RANGE            *
C*                    GATE IS HELD CONSTANT                            *
C*       RINCR     RANGE INCREMENT (RANGE GATE)                        *
C*       OMEGA     AZIMUTH ANGLE SWEEP RATE, RADIANS/SEC               *
C*       BEAMW     BEAMWIDTH, RADIANS                                  *
C*       ISITYM    INTEGER GAME TIME RADAR OR OTHER SCANNING           *
C*                    SENSOR IS SITED (TIME OPERATIONS BEGIN)          *
C*       RMIN      MINIMUM AND MAXIMUM RANGES                          *
C*       RMAX         OF SELECTED RADAR SCAN PATTERN                   *
C*       AZCTR     ORIENTATION ANGLE... AZIMUTH VALUE AT               *
C*                    CENTER OF SCAN, IN RADIANS                       *
C*       CVANGL    COVERAGE ANGLE (SECTOR WIDTH), IN RADIANS           *
C*                                                                     *
C*    REMARKS                                                          *
C*       SETSC1 MUST BE CALLED BEFORE SCAN1 IS CALLED,                 *
C*       IN ORDER TO SET THE WORKING PARAMETERS FOR SCAN1.             *
C*       SCAN1 WILL GENERALLY BE CALLED OFTEN (WITHIN LOOP).           *
C*       THE USE OF SETSC1 (OUTSIDE OF LOOP) ELIMINATES THE            *
C*       NEED FOR REPETITIVE REDUNDANT CALCULATIONS IN SCAN1.          *
C*       COMMUNICATIONS WITH SCAN1 (PASSAGE OF WORKING PAR-            *
C*       AMETERS) IS VIA LABELED COMMON /COMSC1/.                      *
C*                                                                     *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*       NONE                                                          *
C*                                                                     *
C***********************************************************************
```

Figure 2.2-59

```
C******************************  SETSC2  ******************************
C*                                                                    *
C*                         SUBROUTINE SETSC2                          *
C*                                                                    *
C*    PURPOSE                                                         *
C*        PRESETS WORKING PARAMETERS FOR SCANNING ROUTINE SCAN2       *
C*                                                                    *
C*    CALLING SEQUENCE                                                *
C*        CALL SETSC2 (AZCTR,CVANGL,RMIN,RMAX,AZINC,DRDT,RGATE,ISITYM) *
C*                                                                    *
C*    DESCRIPTION OF PARAMETERS                                       *
C*        AZCTR      ORIENTATION ANGLE (SCAN CENTER), RADIANS         *
C*        CVANG      TOTAL COVERAGE OF SCAN, RADIANS                  *
C*        RMIN       RANGE LIMITS FOR                                 *
C*        RMAX         SCAN2 ROUTINE                                  *
C*        AZINC      INCREMENT IN AZIMUTH ANGLE (AFTER RANGE SWEEP),RAD *
C*        DRDT       SWEEP RATE IN RANGE, METERS/SEC                  *
C*        RGATE      RANGE GATE, METERS                               *
C*        ISITYM     INTEGER GAME TIME RADAR OR OTHER SCANNING SENSOR *
C*                     IS SITED (TIME OPERATIONS BEGIN)               *
C*    REMARKS                                                         *
C*        SETSC2 MUST BE CALLED BEFORE SCAN2 IS CALLED, IN ORDER TO   *
C*        SET THE WORKING PARAMETERS FOR SCAN2. COMMUNICATIONS OF     *
C*        PARAMETERS IS VIA COMMON /COMSC2/.                          *
C*                                                                    *
C*    SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*        NONE                                                        *
C*                                                                    *
C**********************************************************************
```

Figure 2.2-60

```
C******************************  ITODEV  ********************************
C*                                                                      *
C*                          SUBROUTINE ITODEV                           *
C*                                                                      *
C*     PURPOSE                                                          *
C*         EVALUATES THE TIME-OF-DAY VARIABLE, ITOD, FROM THE           *
C*         INTEGER GAME TIME VARIABLE, ITIME.                           *
C*                                                                      *
C*     CALLING SEQUENCE                                                 *
C*         CALL ITODEV                                                  *
C*                                                                      *
C*     DESCRIPTION OF PARAMETERS                                        *
C*         ALL REQUIRED PARAMETERS ARE IN COMMON /BASICT/.              *
C*         ITIME AND ITODST (THE LATTER A CONSTANT THROUGHOUT           *
C*         GAME PLAY) ACT AS 'INPUTS', ITOD IS 'OUTPUT'.                *
C*                                                                      *
C*     SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED                     *
C*         NONE                                                         *
C*                                                                      *
C************************************************************************
```

Figure 2.2-61

```
C******************************* IUTEVL ********************************
C*                                                                     *
C*                        FUNCTION IUTEVL                              *
C*                                                                     *
C*    PURPOSE                                                          *
C*        FUNCTION TYPE SUBPROGRAM.  GIVEN COORDINATES X,Y AS          *
C*        INPUT ARGUMENTS, IT PROVIDES AN  IUT  VALUE, THAT            *
C*        IDENTIFIES WHICH UNIT TERRAIN TYPE CORRESPONDS TO X,Y.       *
C*                                                                     *
C*    CALLING SEQUENCE (ILLUSTRATIVE)                                  *
C*        IUT = IUTEVL(X,Y)                                            *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*        EXPLICIT ARGUMENTS                                           *
C*            X    GAME COORDINATES OF                                 *
C*            Y        POINT, REAL, METERS                             *
C*                                                                     *
C*        REQUIRED COMMON AREA DATA                                    *
C*            COMMON /UTVSXY/ IUTXY(15,60)                             *
C*                                                                     *
C*            CONTAINS THE IUT VALUES BY 'BLOCK' (A BLOCK BEING        *
C*            A 500METERX500METER SQUARE), IN A SPECIAL PACKED         *
C*            FORMAT.                                                  *
C*                                                                     *
C*    REFERENCE-  SAM I MEMO 1016 (CAL), R. KINZLY, 5 AUG 1970         *
C*                                                                     *
C*    COMMENTS -                                                       *
C*            THIS SUBPROGRAM HAS NO BUILT-IN ERROR CHECKS.            *
C*            DATA ARE ASSUMED PRESENT IN COMMON AREA /UTVSXY/,        *
C*            AND INPUT PARAMETERS X AND Y ARE ASSUMED COMPATIBLE      *
C*            WITH THESE DATA.                                         *
C*                                                                     *
C*                                                                     *
C***********************************************************************
```

Figure 2.2-62

IDTGLG	the input IDTL value is placed in the same slot as the other variables for subsequent identification

Also returned in common is an index, KTL, that identifies to the calling program which of the 20 slots was used for storing requested variables.*

TGTLXY supplies the following information, keyed to a specific game time value:

X, Y	coordinates of target leading element on the specified leg (at the given time)
NELLEG	number of target elements on leg (possibly less than total number of elements in target)
VELLEG	target speed on the leg

TGTLXY may be regarded as a "time interpolator" for the end point data given by TGTLG.

STRECT and STCIRC relate to the geometry of a moving target passing through a stationary rectangular or circular field, respectively. The results are based on a line target, but are reasonably good estimates for non-line targets*.

FOR STRECT, the input variables define the rectangle (coordinates of center, length, width and orientation angle), and the "IDTL" identifier for target and leg. The subroutine returns three variables explicitly:

TLEENT	time that target leading element enters the rectangular field
TLEEXT	time that target leading element exits from the field
TIMLT	time length of target on specified leg.

*Exact calculations for intersections of circle with rectangle, rectangle with rectangle, or circle with circle, are extremely difficult, and not justified by accuracy with which target and sensor field geometries are known.

Because STRECT calls TGTLG, the program that calls STRECT may also access the /TGLGCM/ common area for additional parameters.

STCIRC is exactly analogous to STRECT, and returns the same output variables. The calling sequence is slightly simpler, because no length-vs-width distinction exists for a circle and orientation angle has no significance.

EVNEFD, as used in the initial MSM coding, solves a geometry problem for ARFBUOY sensor calculations only. It provides (a) the number of target elements within the ARFBUOY field for any of the three types of field geometry that may be specified* by planner data, and (b) target speed within the field. The number of input variables, including those implicit via common areas, is relatively large; the reader is referred to Fig. 2.2-50 (or to program listings) for full specification. EVNEFD calls subroutines TGTLG, and either STCIRC or STRECT according to field geometry.

CLOSEL ("closest element"), a geometry routine, supplies an integer that specifies which target element is closest to a fixed position. This rather special information is used only to support the magnetic sensor routine, MAGTG, and its executive routine, EX3MAG. It calls TGTLG, STCIRC, STRECT.

KSTVLU is basically a storage lookup routine, that determines a strength-of-signal index (KSTRNG) variable required by sensor routines.

FTPAR1 and FTPAR2 are the only two MSM routines uniquely addressed to data manipulation and access for false targets. False target data, unlike data for red and blue forces regarded as targets, do not reside in system parameter storage, but are passed within the sublist for any type 1 event that refers to (one or more) false targets.** FTPAR1 and FTPAR2 provide the means for conveniently accessing the appropriate words from these lists and passing to the calling program the input data or values derived from the input data.

FTPAR1 provides static information, and FTPAR2 provides dynamic information. In usage, FTPAR1 is called outside of the "time loops", while FTPAR2 must be called within these loops. FTPAR1 specifically provides these data for false targets:

---

*"Open circle," "open rectangle," or "path (road) emplacement".

**See Appendix B: format for event type 1 sublists.

IDCODE	as provided by PRERUN
ITGTTP	a target generic type code as required by sensor routines (1 = personnel, 2 = vehicles, etc.)
KSTRNG	strength-of-signal index required by sensor routines
NEL	estimate of number of elements in target
SPACE / ALT	estimate of spacing between target elements, or altitude, according as the target type code specifies ground or aircraft target.

FTPAR2 specifically provides these data for false targets, corresponding to a specified time (input variable):

X Y	coordinates of target
VX VY	x and y components of target velocity
SPEED	target speed

Two subroutines, BFIASK and BFILUM, support the IMAGE sensor routine and its executive (EX3IMG) in providing access to battlefield illumination data. Although these routines could have been combined into one, separation allows a potentially significant reduction in computation time by placing a static screening routine (BFIASK) outside of the time-consuming computational loop in EX3IMG.

Battlefield illumination data are stored in partition MSM14 of storage array MSMPAR, which has potential storage room for up to 20 simultaneous sets of data (8 words per set). BFIASK searches this area prior to the main computational loop within EX3IMG. If no illumination events occur during the time interval covered by the loop, a logical variable is set .FALSE., and the loop logic avoids repetitive search and access operations. If one or more illumination events do occur within the time interval, then BFIASK determines the earliest and latest times that illumination needs to be explicitly accessed in the loop.

BFILUM is called within the major time loop (in EX3IMG). If so directed by BFIASK-generated keys, BFILUM accesses battlefield illumination data from storage and passes the appropriate words to subroutine IMAGE via reserved words in common area / SENSOR/. If only one illumination event is active at a given time, this operation is trivial.

If more than one, the program assumption is made that these multiple lightings are not in the same local area or, if so, that the closest one would dominate. At any rate, a search is made over the active light events to determine which is closest to the sensor, and the one set of parameters for this closest one is placed into common /SENSOR/ for use by subroutine IMAGE.

Four subroutines are addressed to scanning logic and calculation, for those sensors that may have sector scanning (radar, image and thermal viewer). Two sector scan logics are defined. For "type 1" scan, the appropriate subroutines are SCAN 1 (used dynamically within computing loop) and SETSC1 (which establishes working parameters for SCAN 1, outside the computing loop). For "type 2", the analogous routines are SCAN2 and SETSC2. Definitions of the two types of scan logic, calling sequences, etc. are given in program listings (Volume III), and in Figs. 2.2-57 and 2.2-58. A qualitative description of the role of SCAN 1 (or SCAN 2) in program operation is, however, given below.

For definiteness, consider a radar sensor interrogation, and a type 1 sector scan logic. In EX3RDR, a double computing loop is used for the sensor-target search: a loop over time values, and for a fixed time a loop over possible targets (up to 16). Consider, then, a fixed time and a fixed target. Program logic then proceeds according to:

(a) a call to SCAN 1 is made, to determine if the target is within the illumination window of the scan; if not, the following step is bypassed.

(b) if the target is illuminated, according to SCAN 1, the RADAR sensor routine is interrogated.

The implicit assumption in this logic is that the radar maintains a consistent search mode based on the planner-specified scan parameters and does not, for example, switch from a search to a track mode. Programming changes to effect such a switch would not per se be difficult, but the doctrine for switching in a multi-target environment would require considerable care in definition.

ITODEV is a short utility routine, that evaluates the time-of-day value from the current value of time. Game time internally is specified by number of seconds from specified game start.

IUTEVL (IUT EVaLuation) is a short utility routine that is used both in MSM and PRERUN. Its input variables are x and y game coordinates. The value returned is the "IUT" index that specifies the unit terrain type. This IUT value may then be used to access the proper variables from the so-called UNTER tables of terrain parameters. IUTEVL, in effect, unpacks the information in common area /UTVSXY/ in order to associate the proper IUT value in the 500 meter by 500 meter square containing (x, y).

2.2.6 MSM Labeled Common Areas and Storage Control

Labeled common areas, used extensively for linkages among MSM subprograms, have considerable influence on the details of MSM structure and design. A very special case exists for labeled common /BIGSTR/, which is used for storage of system parameters and associated pointer tables. Here, the storage logic centers on efficient use of storage for externally supplied data, having an indefinite mix of lists lengths and types of variables... to be implemented with USASI FORTRAN coding. Thus, storage allocation, editing, and storage access control not only indicated requirements for special explicitly associated subprograms, but affects and details of operation of the others.

In the following subsections, a general overview of MSM labeled common areas is followed by specific discussion of areas /BIGSTR/, /SENSOR/ and /TARGET/. These three have unique features, important to MSM program structure understanding. In particular they have variable content depending upon external data (/BIGSTR/) or upon local points of use within MSM (/SENSOR/ and /TARGET/).

2.2.6.1 Overview of MSM Labeled Common Areas

Table 2.2-VI summarizes the significant labeled common areas within MSM. Not shown are those that have very restricted useage -- e.g., ones that only provide linkage between two related routines.

Necessary initialization of values within labeled common blocks is performed in a single BLOCK DATA subprogram called MSMBLK. A reproduction of the FORTRAN listing of MSMBLK, given in Fig. 2.2-63, identifies the common areas so initialized and the numerical values assigned.

The bulk of initialization centers on clearing to zero the starting values of parameters. The non-zero value assignments are:

(a)	/SENVAR/	Designer values for MSM sensor routines. These values are not normally considered subject to change, but the option to change requires only MSMBLK recompilation.
(b)	/CONST/	Mathematical constants here are of course not intended to be changed. LDUMP controls dumps of internal variables of the sensor routines upon each call. The value .TRUE. (which would energize these dumps) would be useful only for debugging purposes.

Table 2.2-VI

SUMMARY OF SIGNIFICANT LABELED COMMON AREAS

NAME	NO. OF WORDS	NO. OF VARIABLES	COMMENTS
/BIGSTR/	20,341*	6	Major storage area: system parameters and pointer tables
/SENSOR/	50	**	Linkage: sensor parameters to sensor routines
/TARGET/	210	**	Linkage: target parameters to sensor routines
/TGLOCM/	221	12	Linkage & storage: "target-leg" dynamic variables
/SYSCNT/	128	128	System counters
/ANDQUE/	250	25	Queue for AND-logic sensor reports awaiting confirmation
/OUTCOM/	25	25	Linkage: UGS sensor reports to output routine UGSOUT
/SENVAR/	20	20	Designer values for sensor parameters
/CONST/	8	8	Miscellaneous constants
/PGCONT/	19	3	Page control: page count, line count, alphanumeric heading
/ATMENV/	19	17	Atmospheric data
/UTVSXY/	900	1	Packed data: to establish unit terrain type from coordinates
/UNTER/	193	20	Terrain, foliage parameters for up to 10 unit terrain types
/BASICT/	6	6	Game time, plus other time parameters

* As provided. A dimension of 20000 for array MSMPAR within /BIGSTR/ can be raised or lowered, depending upon capacity of computer and scenario size.

** Definition in terms of variable names varies according to sensor routine.

ICARD, IPRINT correspond to unit device numbers for the system card reader and printer, respectively. The values 5, 6 assigned correspond to most IBM computer installations.*

(c) /BIGSTR/ One array, LJUMP, has non-zero values, assigned. These correspond to the lengths of data subsets within MSMPAR. For example, the first assigned value (15) gives the number of words allocated to each UGSARRAY... the first category stored within MSMPAR.

```
C******************************  MSMBLK  *******************************
C*                                                                    *
C*                     BLOCK DATA MSMBLK                              *
C*                                                                    *
C*    PURPOSE                                                         *
C*      BASIC BLOCK DATA PROGRAM FOR ALL MSM COMMON AREAS             *
C*                                                                    *
C**********************************************************************
C*                                                                    *
C**********************************************************************
C
C
      BLOCK DATA
C
C
C                    *** /TGLGCM/ ***
      COMMON /TGLGCM/ IDTGLG(20),X0(20),Y0(20),T0(20),X1(20),Y1(20),
     X T1(20),TL0(20),SPEEDL(20),VX(20),VY(20),KTL
      DIMENSION TLDUM(221)
      EQUIVALENCE (TLDUM(1),IDTGLG(1))
      DATA TLDUM/221*0.0/
C
C                    *** /BIGSTR/ ***
      COMMON /BIGSTR/ LJUMP(13),MOSMSM(13),IPBIGS(15),IPBLU(200),
     X  IPPATH(100),MSMPAR(20000)
      DATA  LJUMP/15,8,8,0,14,24,38,10,11,7,0,7,10/, MOSMSM/13*0/,
     X  IPBIGS/15*0/, IPBLU/200*0/, IPPATH/100*0/, MSMPAR/20000*0/
C
C
C                    *** /PGCONT/ ***
      COMMON /PGCONT/ ILINES,IPAGE,HEADNG(17)
       DATA ILINES/0/,IPAGE/0/,HEADNG/17*4H    /
C
C
C
C             *** /SYSCNT/  (SYSTEM COUNTERS) ***
      COMMON / SYSCNT/ KSEI(3),KACO(3),KMAG(3),KARF(3),KPIR(3),
     1 KRDR(3,3),KIMG(3,3),KTHV(3,3),KBKW(3),KDSEI(3),KDACO(3),
     2 KDMAG(3),KDARF(3),KDPIR(3),KDRDR(3,3),KDIMG(3,3),KDTHV(3,3),
     3 KDBKW(3), K3SEI,K3ACO,K3MAG,K3ARF,K3PIR,K3RDR,K3IMG,K3THV,
     4 K3BKW,KSCN1R,KSCN2R,KSCN1I,KSCN2I,KSCN1T,KSCN2T,KFSEI,KFACO,
     5 KFMAG,KFARF,KFPIR,KFBKW,KEV1,KEV2,KEV3,KEV4,KEV5,KEV6,KEV7,
     6 KEV8,KEV9,KEV10,KNTOEV,KRTPRI,KBTPRI,KFTPRI,KRT,KBT,KFT
      DIMENSION KNTSYS(128)
      EQUIVALENCE (KNTSYS(1),KSEI(1))
      DATA KNTSYS /128*0/
C
C                    ***  /ANDQUE/  ***
      COMMON /ANDQUE/ KDCD(10),ITPSN(10),IDSN(10),ITOT(10),NRT(10),
     1  NBT(10),NFT(10),TOPRT(10),XPRT(10),YPRT(10),RPRT(10),XSN(10),
```

Figure 2.2-63 BLOCK DATA SUBPROGRAM, MSMBLK

```
     2  YSN(10),IDCEM(10),NMN(10),IDM1(10),IDM2(10),IDM3(10),IDAR(10),
     3  IDETR(10),ADM1(10),ADM2(10),ADM3(10),ADETR(10),ADCEM(10)
      DIMENSION  IANDO(250)
      EQUIVALENCE (IANDO(1),KOCD(1))
      DATA  IANDO /250*0/
C
C                    ***  /OUTCOM/  ***
      COMMON /OUTCOM/ KOCODE,ITYPSN,IDSNSR,ITIMDT,NRTGTS,NBTGTS,
     1     NETGTS,IDPTGT,XPTGT,YPTGT,RPTGT,XSNSR,YSNSR,
     2     IDCONF,NMONS,IDMON1,IDMON2,IDMON3,IDARAY,IDETRP,
     3     ADMON1,ADMON2,ADMON3,ADETRP,ADCONF
      DIMENSION IOUTCM(25)
      EQUIVALENCE  (IOUTCM(1),KOCODE)
      DATA IOUTCM/25*0/
C
C                    *** /SENVAR/ ***
      COMMON /SENVAR/ CONSTA,TDEL2A,BIASAC,BWACOU,
     1                CONSTS,TDEL2S,BIASSE,BWSEIS
     2, PHIAZ,PHIEL,DIAM,BWPIR,DEVXMN
     3 ,XMNDEV,BANDTH,ANEP,OPTXMN
     4 ,THRESP,DELAZ,TIMMAX
      DATA BIASAC,BWACOU,CONSTA,TDEL2A/0.2,500.0,1.0,40.0/
      DATA BIASSE,BWSEIS,CONSTS,TDEL2S/0.2,100.0,1.0,40.0/
      DATA PHIAZ,PHIEL,DIAM,BWPIR,DEVXMN / 0.0174533,0.0523599,10.0,100.
     1,0.9/
      DATA XMNDEV,BANDTH,ANEP/0.9,2600.0,1.E-10/
      DATA OPTXMN/0.8/
      DATA THRESP,DELAZ,TIMMAX/0.5E-07,0.0349,3.0/
C
C
C                    *** /CONST/ ***
      COMMON /CONST/ LDUMP,SQ2,DEG,RAD,PI,STEFK,ICARD,IPRINT
      LOGICAL LDUMP
      DATA LDUMP,SQ2,DEG,RAD,PI,STEFK/.FALSE.,1.414214,57.29578,0.017453
     13,3.141593,1.805455E-8/
      DATA ICARD,IPRINT/5,6/
C
      END
```

Figure 2.2-63 BLOCK DATA SUBPROGRAM, MSMBLK (Cont.)

2.2.6.2 Labeled Common /BIGSTR/

MSM computational steps constantly require from storage values of various physical system parameters (e.g., of sensors, arrays, forces, monitors, etc.). The number of such parameters will be large in all practical simulation scenarios... large enough that unnecessarily severe restrictions on scenario complexity would be imposed, for a given "reasonable" computer size, if simple FORTRAN coding and routine assignments of DIMENSION statements were to be used for storage control.

In this context, special attention was paid to the storage allocation, control, access and editing of system parameters. A singly-subscripted storage array called MSMPAR, with initial dimension of 20000 words, was established to hold system parameters, whatever their mix in terms of content, variable types (real, integer, logical, alphanumeric), sublist lengths, and numbers of subsets vs. major data categories might be in any particular job application. Storage allocation and access control were then developed (without deviation from USASI FORTRAN coding) to allow contiguous packing of these mixed data. With such logic, the MSM restriction on "scenario size" then applies only to the total number of parameter data, not upon their numbers by individual category; and full utilization of available storage is closely approached.

Conceptually, MSMPAR is conceived as having 14 partitions, labeled MSM1 through MSM14.* The first 13 of these correspond to 13 categories of system parameters. The 14th (MSM14) is used for storage, during dynamic simulation, of battlefield illumination event data. Names of these 14 categories, and some associated numerical values, are given in Table 2.2.VII.

The first 13 partitions are of indefinite length. In any one job application, lengths are determined by externally established scenario specifications. Access to these contiguous partitions is based on pointer tables, that can be used directly by working subroutines, or used indirectly by calls to storage access utility routines.** Labeled common area /BIGSTR/ holds these pointer tables and the storage array MSMPAR. Complete specification of /BIGSTR/ content, and initialization of values by BLOCK DATA, are given in the MSMBLK listing, Fig.2.2-63. Meanings of the tables within /BIGSTR/ are given in Table 2.2.VIII.

---

* The residual, or unused, portion of MSMPAR could be thought of as a 15th partition (MSM15), available for future program growth.

**PARVLU, PARPTR, ARRVLU, ARRPTR, TGTVLU and TGTPTR, discussed in Section 2.2.5.8.

Table 2.2-VII

PARTITIONING OF STORAGE AREA MSMPAR

PARTITION	WORDS PER SUBSET*	PARAMETER STORAGE FOR:
MSM1	15	UGSARRAYS
MSM2	8	STASCAN ARRAYS
MSM3	8	MOVING ARRAYS
MSM4	**	BLUE FORCES
MSM5	14	RED FORCES
MSM6	24	SENSORS
MSM7	38	SENSOR DESCRIPTORS
MSM8	10	FIRETRAPS
MSM9	11	MONITORS
MSM10	7	DATA LINKS
MSM11	**	PATH SPECIFICATIONS
MSM12	17	FORCE TYPE DATA
MSM13	10	COVERAGE/SCAN PARAMETERS
MSM14	8	BATTLEFIELD ILLUMINATION DATA
(MSM15)		(Uncommitted portion of MSMPAR)

NOTES: * For example, data stored for one UGSARRAY requires 15 words of storage in MSM. Total storage for MSM1 would therefore be 15 times the number of UGSARRAYs.

** Variable length. Separate pointer tables generated.

*** EXEC1A provides a pointer value to MSM14, but does not load MSM14 with data. Reserved space is 160 words total, to accommodate up to 20 sets of battlefield illumination data simultaneously in storage.

Table 2.2-VIII

ARRAY VARIABLE DEFINITIONS FOR LABELED COMMON/BIGSTR/

ARRAY NAME	DIMENSION	DEFINITION
LJUMP	13	LJUMP(K) specifies amount of storage for one data subset within partition or category K. For K = 4 (BLUE FORCES) and K = 11 (PATHS), the LJUMP value is not used.
NDSMSM	13	NDSMSM(K) specifies the number of data subsets stored within partition K.
IPBIGS	15	Master pointer table. IPBIGS(K) is the pointer value (subscript of MSMPAR) for the beginning of data in partition K.
IPBLU	200	Auxiliary pointer table, for BLUE FORCES data subsets within MSM4. IPBLU(K) gives the pointer value for the Kth blue force.
IPPATH	100	Auxiliary pointer table, for PATHS data subsets within MSM11. IPPATH(K) gives the pointer value for the Kth path.
MSMPAR	20000	Primary storage array for system parameters.

Notes: 1. All arrays except LJUMP are initialized to zero by BLOCK DATA subprogram MSMBLK. Nonzero values of LJUMP are set in MSMBLK. See Figure 2.2-63.

2. Values within all arrays except LJUMP are entered by subroutine EXEC1A, during the processing of parameter input data.

3. The DIMENSION values for MSMPAR, IPBLU and IPPATH are partially arbitrary. Provided consistent recompilation is carried throughout all MSM programs in which /BIGSTR/ is defined, these dimensions can be changed to accommodate computer storage capacity. The dimension of MSMPAR is, of course, the most critical.

The loading of MSMPAR with system parameter data is a major MSM task, that is handled under control of a level 1 executive routine EXECIA (see Section 2.2.5). The overall execution of this task involves several elements of control:

(a) Reading parameter data, one subset at a time, from the PRERUN-generated parameter data file ("JTFWDF") into a buffer area.

(b) Editing content of each subset (additions and/or changes, depending upon the data category represented).

(c) Transferring the edited subset to a pre-allocated section of MSMPAR storage.

(d) Updating pointer for subsets (in preparation for the next step (c)); and if a change in major data category occurs, or if variable length sublists for <u>blue forces</u> or for <u>paths</u> are being controlled, inserting appropriate values into the pointer tables.

(e) After all parameters are initially stored, a final editing pass is made, that implements changes not possible at the time step (b) occurred in the storage sequencing.

Details of these steps, and their effect on the final storage map, are extensive and important only to user programmers having an interest in internal program details (as, for example, in extending the initial scope of the SAM). These details are discussed in Appendix D.

2.2.6.3 Labeled Common Areas /SENSOR/ and /TARGET/

In a general sense, /SENSOR/ is used to pass sensor parameter values to the sensor routines and /TARGET/ is used to pass target parameter values to the sensor routines.* The feature of these common areas that required special discussion is that their content, in the sense of variables (variable names) and storage map, varies depending upon the point of use. For example, the tenth word within /SENSOR/ has a completely different meaning in the EX3SAC/SEISTG (seismic sensor) linkage than it would in, say, the EX3RDR/RADAR linkage.

---

*Recall that a sensor routine represents only a generic sensor type, e.g., seismic. Distinction between two different sensors of the same generic type is based on sensor-specific parameter values dynamically made available (in common area/SENSOR/) as required.

This variable definition logic is, of course, perfectly valid and does not imply any deviations from USASI FORTRAN. It does imply some extra attention to details for those interested in a full description of MSM program structure.

Details for /TARGET/ are relatively simple, and needed information can be deduced from individual program listings (for the sensor routines and associated EX3... executives). Details for /SENSOR/, on the other hand, interact with details of the MSMPAR (see Section 2.2.6.2) storage map, and require specific discussion on a sensor-by-sensor basis. Details for common areas /TARGET/ and /SENSOR/, per se, are covered in appropriate program listings. The information placed into /SENSOR/ hinges strongly upon the MSM6 partition of MSMPAR (words 16 f.f., that are sensor type dependent); reference to Appendix D, which discusses list structures within MSMPAR, may provide additional useful data.

TABLE 2.2-IX

MSM COMMON AREAS

USE	VARIABLES		USED BY
COMMON/BIGSTR/LJUMP (13), NDSMSM (13), IPBIGS (15), IPBLU (200), IPPATH (100), MSMPAR (20000)			
Major Storage Area System Parameters & Pointer Tables	LJUMP (13)	- p 2-266 designer input: specifies amt. of storage for one data subset w/in partition or category K	EXEC 1 EXEC 1A EX2 BFL EX2 HLT EX2 SFA EX2 SNP EX2 SNR EX2 SPC EX2 SRP EX2 UPD EX3 ARF EX3 BKW EX3 IMG EX3 MAG EX3 PIR EX3 RDK EX3 SAC EX3 THV DUMPMS SACDET SCNOUT UGSDET UGSOUT DUMPMS ARRPTR ARRVLU PARPTR PARVLU TGTPTR TGTVLU TGTLG KSTVLU BFIASK BFILUM MSMBLK
	NDSMSM (13)	- designer input: 13*0 specifies # of data subsets stored w/in partition K	
	IPBIGS (15)	- designer input: 15* 0 master pointer table	
	IPBLU (200)	- designer input: 200*0 auxiliary pointer table for BLUE FORCES	
	IPPATH (100)	- designer input: 100*0 auxiliary pointer table for PATHS	
	MSMPAR (20000)	- designer input: 20000*0 primary storage array for system parameters (p 2-266)	

USE	VARIABLES	USED BY

COMMON/TGLCCM/IDTGLG (2Ø), XØ (2Ø), YØ (2Ø), TØ (2Ø), X1 (2Ø), Y1 (2Ø), T1 (2Ø), TLØ (2Ø), SPEEDL (2Ø), VX (2Ø), VY (2Ø), KTL

USE	VARIABLES		USED BY
LINKAGE & STORAGE:	IDTGLG (2Ø)	- stored target ID/leg no.	EX3 IMG
"Target-Leg" dynamic variables	XØ (2Ø)	- X-coord of entry-node of leg	EX3 MAG EX3 PIR
DIMENSION TLDUM (200) TLDUM (2ØØ) = 2ØØ * Ø	YØ (2Ø)	- Y-coord of entry-node of leg	EX3 RDR EX3 THV
EQUIVALENCE TLDUM(1), IDTGLG(1)	TØ (2Ø)	- time leading element of target reaches	SCNOUT TGTLG
p 2-234	X1 (2Ø)	- X-coord of leave-node of leg	TGTLXY STRECT
	Y1 (2Ø)	- Y-coord of leave-node of leg	STCIRC EVNEFD
	T1 (2Ø)	- time leading element of target leaves	CLOSEL MSMBLK
	TLØ (2Ø)	- time length of target, seconds	
	SPEEDL (2Ø)	- speed of target on specified leg	
	VX (2Ø)	- X-component of velocity on leg	
	VY (2Ø)	- Y-component of velocity on leg	
	KTL	- subscript value to above arrays	

COMMON/SYSCNT/KSEI (3), KACO (3), KMAG (3), KARF (3), KPIR (3), KRDR (3, 3), KDARF (3), KDPIR (3), KDRDR (3,3), KDIMG (3, 3), KDTHV (3, 3), KDBKW (3), K3SEI, K3ACO, K3MAG, K3ARF, K3PIR, K3RDR, K3IMG, K3THV, K3BKW, KSCN1R, KSCN2R, KSCN1I, KSCN2I, KSCN1T, KSCN2T, KFSEI, KFACO, KFMAG, KFARF, KFPIR, KFBKW, KEV1, KEV2, KEV3, KEV4, KEV5, KEV6, KEV7, KEV8, KEV9, KEV1Ø, KMTOEV, KRTPRI, KBTPRI, KFTPRI, KRT, KBT, KFT

USE	VARIABLES			USED BY
SYSTEM COUNTERS	KSEI	counts calls to	SEISTG	EXEC 1B
EQUIVALENCE	KACO		ACOUTG	EX2 HLT
KNTSYS(1),	KMAG		MAGTG	EX2 UPD
KSEI(1)	KARF		ARFTG	EX3 ARF
where:	KPIR		PIRTG	EX3 BKW
KNTSYS(128) =	KRDR		RADAR	EX3 IMG
128 * Ø	KIMG		Imag	EX3 MAG
	KTHV		Thermal	EX3 PIR
	KBKW		Breakwire	EX3 RDR

USE	VARIABLES			USED BY
COMMON/SYSCNT/ Con't				
			Counts number of real target detections made by:	EX3 SAC EX3 THV SCNOUT UGSOUT MSMBLK
	KDSEI		Seismic	
	KDACO		Acoustic	
	KDMAG		Magnetic	
	KDARF		Arfbuoy	
	KDPIR		PIR	
	KDRDR		Radar	
	KDIMG		Image	
	KDTHV		Thermal	
	KDBKW		Breakwire	
	K3SEI	- seismic	counters of calls to sensor exec routines	
	K3ACO	- acoustic		
	K3MAG	- magnetic		
	K3ARF	- arfbuoy		
	K3PIR	- passive ir		
	K3RDR	- radar		
	K3IMG	- image		
	K3THV	- thermal viewers		
	K3BKW	- breakwire		
	KSCN1R		Counts # of calls to scan routines 1 and 2 for radar, image devices and thermal viewers.	
	KSCN2R			
	KSCN1I			
	KSCN2I			
	KSCN1T			
	KSCN2T			
			Counts false alarms for:	
	KFSEI		seismic	
	KFACO		acoustic	
	KFMAG		magnetic	
	KFARF		arfbuoy	
	KFPIR		passive infra-red	
	KFBKW		breakwire	

USE	VARIABLES	USED BY
COMMON/SYSCNT/ Con't		
	KEV1 - sensor interrogation counter KEV2 - sensor false alarm counter KEV3 - sensor parameter change counter KEV4 - up/down status changes counter KEV5 - monitor up/down counter KEV6 - data link up/down counter KEV7 - firetrap begin/end counter KEV8 - array emplace/cease counter KEV9 - battlefield illumination counter KEV1Ø - # sensor repositions (counter) KNTØEV - page control counter KRTPRI - red forces } KBTPRI - blue forces } total sensor reports by category of primary "target" KFTPRI - false } KRT - red forces } KBT - blue forces } total multiple targets affecting sensor detections counted according to multiplicity KFT - false targets }	

COMMON/ANDQUE/KDCD(1Ø), ITPSN(1Ø), IDSN(1Ø), ITDT(1Ø), NRT(1Ø), NBT(1Ø), NFT(1Ø), IDPRT(1Ø), XPRT(1Ø), YPRT(1Ø), RPRT(1Ø), XSN(1Ø), YSN(1Ø), IDCFM(1Ø), NMN(1Ø), IDM1(1Ø), IDM2(1Ø), IDM3(1Ø), IDAR(1Ø), IDFTR(1Ø), ADM1(1Ø), ADM2(1Ø), ADM3(1Ø), ADFTR(1Ø), ADCFM(1Ø)

USE	VARIABLES	USED BY
QUEUE FOR AND-Logic sensor reports awaiting confirmation EQUIVALENCE IANDQ(1),KDCD(1) where: IANDQ(25Ø) = 25Ø * Ø	KDCD - target code 1 = red, 2 = blue, 3 = false ITPSN - sensor type IDSN - sensor ID ITDT - target ID NRT - Number of red targets in sensor range NBT - Number of blue targets in sensor range NFT - Number of false targets in sensor range IDPRT - ID of primary target XPRT - X coordinate of primary target YPRT - Y coordinate of primary target RPRT - range of primary target XSN - X coordinate of sensor YSN - Y coordinate of sensor IDCFM - ID of confirming sensor NMN - Number of monitors	UGSOUT MSMBLK

USE	VARIABLES	USED BY
COMMON/ANDQUE/ Con't		
	IDM1 - ID of 1st Monitor IDM2 - ID of 2nd Monitor IDM3 - ID of 3rd Monitor IDAR - ID of array IDFTR - ID of fire trap ADM1 - Alphanumeric conversion of IDM1 ADM2 - " " " IDM2 ADM3 - " " " IDM3 ADFTR - " " " IDFTR ADCFM - " " " IDCFM	

COMMON/OUTCOM/KDCODE, ITYPSN, IDSNSR, ITIMDT, NRTGTS, NBTGTS, NFTGTS, IDPTGT, XPTGT, YPTGT, RPTGT, XSNSR, YSNSR, IDCONF, NMONS, IDMON1, IDMON2, IDMON3, IDARAY, IDFTRP, ADMON1, ADMON2, ADMON3, ADFTRP, ADCONF

USE	VARIABLES	USED BY
LINKAGE: UGS sensor reports to output routine UGSOUT EQUIVALENCE IOUTCM(1) KDCODE where IOUTCM(25) = 25 * Ø	KDCODE - 1 = red target 2 = blue target 3 = false target 4 = initial setting ITYPSN - 1 = seismic 2 = acoustic 3 = magnetic 4 = arfbuoy 5 = passive ir 9 = breakwire IDSNSR - ID sensor (seismic, acoustic) ITIMDT - detection time (integer seconds into game) NRTGTS - # red targets NBTGTS - # blue targets NFTGTS - # false targets IDPTGT - ID present target XPTGT - X coordinate of present target YPTGT - Y coordinate of present target RPTGT - range $\sqrt{(X_{PTGT} - X_{SNSR})^2 + (Y_{PTGT} - Y_{SNSR})^2}$ XSNSR - X coord of sensor YSNSR - Y coord of sensor IDCONF - ID confirming sensor NMONS - # monitors IDMON1 - monitor 1 ID IDMON2 - monitor 2 ID IDMON3 - monitor 3 ID IDARAY - planner specified array ID	EX2 ØFA SACDET USSOUT UGSDET MSMBLK

USE	VARIABLES	USED BY
COMMON/OUTCOM/ Con't		
	IDFTRP - fire trap ID ADMON1 - alphanumeric conversion of IDMON1 ADMON2 - " " " IDMON2 ADMON3 - " " " IDMON3 ADFTRP - " " " IDFTRP ADCONF - " " " IDCONF	
COMMON/SENVAR/CONSTA, TDEL2A, BIASAC, BWACOU, CONSTS, TDEL2S, BIASSE, BWSEIS, PHIAZ, PHIEL, DIAM, BWPIR, DEVXMN, XMNDEV, BANDTH, ANEP, OPTXMN, THRESP, DELAZ, TIMAX		
Designer Values for Sensor Parameters	See Page 2-142, Vol I, Part I, Item # 29.	SEISBK SEISTG ACOUBK ACOUTG PIRBK PIRTG THERML IMAGE MSMBLK
COMMON/CONST/LDUMP, SQZ, DEG, RAD, PI, STEFK, ICARD, IPRINT		
Miscellaneous Constants	LDUMP - Designer input value · FALSE ·; controls dumps of internal variable of sensor routines SQ2 - Designer input value 1.414214 DEG - " " " 57.29578 degrees/radian RAD - Designer input value Ø.Ø174533 radians/degree PI - Designer input value 3.141593 π STEFK - " " " 1.8Ø5455E-8 Stefan-Boltzmann / π ICARD - Designer input value 5 card reader IPRINT - " " " 6 output data drive designator	SEISBK SEISTG ACOUBK ACOUTG ARFBK ARFTG ENVIR MAGTG PIRBK PIRTG THERML RADAR IMAGE BWIRBK BRKWIR MSMBLK

USE	VARIABLES	USED BY
COMMON/PGCONT/ILINES, IPAGE, HEADING (17)		
Page control; page count, line count, alpha-numeric heading	ILINES - No of lines used on page IPAGE - Page count Heading - Alphanumeric text, printed as heading on first line of each page of MSM printout. (68 characters in 17 words).	MSMBLK EXEC1 EXEC1B EX2BFL EX2SNP EX2UPD EX3IMG EX3RDR EX3SAC EX3THV DUMPMS SCNOUT PGSKP2 PGSKIP UGSOUT DUMPMS EVNEFD BFILUM
COMMON/ATMENV/ITEFF, SOLALT, ALTLUN, PHSLUN, IPCODE, PRATE, PTOT24, H2ODEN, WSPEED, CCOVER, ATEMP, PRESUR, HUMDTY, VISIB, CEIL, ASID(3),TCLOUD		
Atmospheric Data	(See Table 2.1-VI Prerun Common Area, page 2-135.)	EXEC1B SEISBK SEISTG ACOUBK ACOUTG ARFBK PIRTG BWIRBK THERML ENVIR RADAR PIRBK IMAGE
COMMON/UTVSXY/IUTXY (15, 60)		
Packed Data: to establish unit terrain type from coordinates	IUTXY - Contains the IUT values by "Block" (Block = 500 meters x 500 meter sq.) in special packed format	TERAN TANDT IUTEVL

USE	VARIABLES	USED BY
COMMON/UNTER/XFIELD, YFIELD, ZFIELD, SGLL(1Ø), SGUL(1Ø), CHLL(1Ø), CHUL(1Ø), TDLL(1Ø), TDUL(1Ø), SPLL(1Ø), SPUL(1Ø), ITRDEN(1Ø), CCLL(1Ø), CCUL(1Ø), IVCOV(1Ø), IBACK(1Ø), TVEG(1Ø), ISM(3,1Ø), VISBLL(1Ø), VISBUL(1Ø)		
Terrain, foliage parameters for up to 1Ø unit terrain types TERAN places designer input for UNTER tables into UNTER common area; all data assumed to be on cards (page C-2, Vol2)	XFIELD - X component of earth's magnetic field XFIELD - Y component of earth's magnetic field ZFIELD - Z component of earth's magnetic field SGLL - lower limit of slope gradient, % SGUL - upper limit of slope gradient, % CHLL - lower limit,on foliage transmission,% CHUL - upper limit, canopy or vegetation TDLL - lower limit of tree diameters, (DBH), meters TDUL - upper limit of tree diameters,(DBH), meters SPLL - lower limit stem or clump spacing, meters SPUL - upper limit stem or clump spacing, meters ITRDEN - tree density; 1 = sparce, 2 = light forest, 3 = dense forest CCLL - lower limit, % canopy closure CCUL - upper limit, % canopy closure IVCOV - index vegetation cover; 1 = heavy, 2 = medium, 3 = light, 4 = open, 5 = water IBACK - index on background type(page 5-127) TVEG - transmittance of vegetation cover of canopy for light ISM - index describing soil moisture conditions (page 5-127) VISBLL - lower limit, ground to ground visibility (meters) VISBUL - upper limit, ground to ground visibility (meters)	EXEC1B SEISBK ACOUTG MAGTG RADAR IMAGE TERAN TANOT FOLAGE SEISTG ENVIR

USE	VARIABLES	USED BY
COMMON/BASICT/ITIME, ITOD, ITODST, ITDURN, IDATE, IDAREA		
Game time, plus other time parameters	(See Table 2.1-VI Prerun Common Areas, page 2-135.)	EXEC1B EX2SFA EX2SNP EX2UPD EX3ARF EX3BKW EX3IMG EX3MAG EX3PIR EX3RDR EX3SAC EX3THV SEISTG ACOUTG SEISBK ACOUBK ARFTG ARFBK MAGTG PIRTG PIRBK THERML IMAGE BWIRBK BBKWIR SACDET SCNOUT TIMOUT UGSDET UGSOUT TERAN TANDT TGTLG TGTLXY EVNEFD CLOSEL BFILUM ITODEV ENVIR RADAR
COMMON/COMSC1/TSTART, TMAJOR, TSWEFP, TAZCYC, TFLXDR, CVA, CVAX2, BMW, BMWBY2, OMEG, AZMIN, RMINSC, NGAT		
SETSC1 sets working parameters for SCAN1	TSTART = ISITYM (integer game time radar or other scanning sensor is sited; time op. begin)	SCAN1 SETSC1

USE	VARIABLES	USED BY
COMMON/COMSC1/ Con't		
	TMAJOR = TFIXDR * FLT(NRBKTS) TSWEEP = CVANGL/OMEGA=(coverage angle in radians/azimuth angle sweep rate) TAZCYC = 2.Ø * TSWEEP TFIXDR = TSWEEP * FLT(NREPRI = # of azimuth sweeps for which range gate held constant) CVA = CVANGL (coverage angle in radians) CVAX2 = 2.Ø * CVA BMW = BEAMW (beam width radians) BMWBY2 = BMW/2.Ø OMEG = OMEGA (azimuth angle sweep rate, radians/sec) AZMIN = AZCTR (orientation angle) - CVANGL/2.Ø RMINSC = RMIN (minimum and maximum ranges) RGAT = RINCR (range increment, range rate)	
COMMON/COMSC2/TSTART, TMAJOR, TRSWEP, HFTMAJ, RMINSC, RGTBY2, RSWPRT, AZINCR, AZMIN, AZIBY2		
SETSC2 sets working parameters for SCAN2	TSTART = ISITYM (integer game time radar or other scanning sensor is sited; time op begins) TRSWEP = (RMAX-RMIN = range limits for SCAN2)/DRDT = sweep rate in range TMAJOR = TRSWEP * FLT (NAZ = CVANGL/AZINC + 1.Ø) HFTMAJ = TMAJOR/2.Ø RMINSC = RMIN RGTBY2 = RGATE/2.Ø range gate, meters RSWPRT = DRDT sweep rate in range, meters/sec AZINCR = AZINC increment in azimuth angle after range sweep AZMIN = AZCTR(orientation angle) - CVANGL (total coverage of scan)/2.Ø AZIBY2 = AZINC/2.Ø	SCAN2 SETSC2
COMMON/DUMMY/NTAR (1Ø)		
	NTAR = dummy array	SEISTG

USE	WORD	VARIABLES	USED BY
COMMON/SENSOR			
(Definition and names of variables in argument list vary based on 3d level MSM routine using the parameters. Linkage: Sensor Parameters to Sensor routines.	1	IDSNSR - Vol II, page F-16, Item 1	EX3ARF
	2	IDARAY - Vol II, page F-16, Item 2	EX3BKW
	3	ITYPSN - " " " 3	EX3IMG
	4	IMOVE - Vol I, page C-19	EX3MAG
	5	IDSDPS - Vol II, page F-16, Item 4	EX3PIR
	6	IDPATH - " " , Item 5	EX3RDR
	7	XSNSR - " " , Item 6	EX3SAC
	8	YSNSR - " " , Item 7	EX3THV
	9	OFFSET - " " , Item 8	SCNOUT
	10	ORIENT - " " , Item 9	
	11	IAUX - " " , Item 10	
	12	IDCSPS - " " , Item 11	
	13	IAND - " " , Item 12	
	14	IUPFLG - Up/down flag 0 = down 1 = up	
	15	IUT - Unit Terrain Index, Vol I, pages 3-36, 3-123	

* *

USE	WORD	VARIABLES	USED BY
	1	IDSNSR - Vol II, page F-16, Item 1	EVNEFD
	2	DUM 1(5) - Dummy filler	
	3	XC - Vol I, page 2-239	
	4	YC - Vol I, page 2-239	
	5	OFFSET - Vol II, page F-16, Item 8	
	6	THETA - Vol I, page 2-239	
	7	DUM 2(7) - Dummy filler	
	8	IGEOM - Vol I, page 2-239	
	9	DIMMAX - Vol I, page 2-239	
	10	WIDTH - Vol I, page 2-239	
	11	DUM 3(30) - Dummy filler	

* *

USE	WORD	VARIABLES	USED BY
	1	IDSNSR - Sensor ID	ACOUTG
	2	NOTUS 1(5) - Dummy filler	ARFTG
	3	XSENS - X coordinate of sensor	MAGTG
	4	YSENS - Y coordinate of sensor	PIRTG
	5	NOTUS 2(6) - Dummy filler	SEISTG
	6	IUT - Unit Terrain Index	THERML

* *

USE	WORD	VARIABLES		USED BY
COMMON/SENSOR Con't				
	1	IDSNSR	- Vol I, page 3-122	BRKWIR
	2	NOTUS 1(13)	- Vol I, page 3-122	
	3	IUT	- " "	
	4	IUT	- " "	
	5	NOTUS 2(35)	- Vol I, page 3-122	

* *

	1	IDSNSR	- ID Sensor	IMAGE
	2	NOTUS 1(2)	- Dummy filler	
	3	IAIR	- Airborne Sensor	
	4	NOTUS (2)	- Dummy filler	
	5	XSENS	- X coordinate of sensor	
	6	YSENS	- Y coordinate of sensor	
	7	NOTUS 3(6)	- Dummy filler	
	8	IUT	- Unit Terrain Index	
	9	ITIMEP	- Time of previous sensor use	
	10	ICLEAR	- Clear or Cloud Shadow	
	11	IVISUL	- Sensor Class. Direct = 0, Electronic Aided = 1	
	12	ITYPE	- 0 = Daylight Type 1 = Night Vision	
	13	TIMCON	- Sensor Time Constant	
	14	FOCALL	- Focal length of optical system	
	15	XMTF	- Area under Modulation Transfer Function	
	16	FNUMBR	- Focal length to Diameter Ratio	
	17	ISERCH	- Index, 0 = Natural Light, 1 = Searchlight	
	18	XSRCH	- X coordinate of sensor using searchlight	
	19	YSRCH	- Y coordinate of sensor using searchlight	
	20	BWIDTH	- Beam width of sensor using searchlight	
	21	CPOWER	- Peak candlepower of sensor	
	22	HTGAC	- Height of aircraft	
	23	ILXTRA	- Is there external Illumination, 0 = No	
	24	XLITE	- X coord of flare or indirect light	
	25	YLITE	- Y coord of flare or indirect light	
	26	FLARHT	- Height of Flare	

USE	WORD	VARIABLES		USED BY
COMMON/SENSOR	Con't			
	27	AINTNS	- Intensity of Flare or Searchlight	
	28	MODE Unused(16)	- See page 3-100, Vol I	

* *

	1	IDSNSR	- Sensor ID	Radar (see page 3-77)
	2	NOTUS (5)	- Not used (Dummy)	
	3	XSENS	- X coordinate of Sensor	
	4	YSENS	- Y coordinate of Sensor	
	5	NOFF		
	6	RAZANG	- Radar Azimuth with respect to Path	
	7	NOTUS 2(4)	- Not used	
	8	IUT	- Unit Terrain Index	
	9	PRIMFR	- Precipitation Improvement Factor	
	10	CLIMFR	- Clutter Improvement Factor	
	11	RAMBDA	- Radar Wave Length	
	12	FNKTB	- Filter Thermal Noise	
	13	RADCAR	- Radar Characteristic	
	14	SCANRT	- Scan Rate	
	15	ICON	- Code, 0 means coherent, 1 means noncoherent	
	16	BEAMAZ	- Antenna Azimuth Beam Width	
	17	BEAMEL	- Antenna Elevation "	
	18	RGATE	- Range Gate	
	19	SIGSTB	- Sigma of Clutter Spectrum	
	20	FCUTLO	- Lower Corner of Filter	
	21	FCUTHI	- Upper Corner of Filter	
	22	HGTANT	- Height of Antenna	
	23	PFA	- Probability of False Alarm	
	24	FDOPLER	- Doppler Frequency	
	25	FTGMIN	- Minimum Usable Doppler Freq.	
	26	FTGMAZ	- Maximum " " "	
	27	NOTUS (17)	- Not used	

* *

	7	GEQUIL	- Vol I, page 3-49	ACOUTC SEISTC
	8	THRESH	" "	
	9	VNOISE	- " , output noise	
	10	ITIMLR	- Vol I, page 3-49, time of latest report	

* *

USE	WORD	VARIABLES		USED BY
COMMON/SENSOR Con't				
	11	ITIMLE	- Vol I, page 3-49, time last entry to subroutine	
	12	GAIN	- Vol I, page 3-49, amplifier gain	
	13	GINLST	- Vol I, page 3-49, gain at last entry	
**************************************				
	16	AREADN	- Area density of NBB	EX3 ARF
	17	IMAG	- Bomblet type: Ø = Magnetic 1 = NBB	(continued) See page 3-31
	18	IGEOM	- 1 = open circle, 2 = open line, 3 = road or trail	
	19	DIMMAX	- Vol II, page F-22, Item 19	
	20	WIDTH	- Vol II, page F-23, Item 20	
**************************************				
	16	ITIMEP	- See Item 9 when used by IMAGE	EX3 IMG
	17	ICLEAR	- " " 10	(continued)
	18	IVISUL	- " " 11	
	19	ITYPE	- " " 12	
	20	TIMCON	- " " 13	
	21	FOCALL	- " " 14	
	22	XMTF	- " " 15	
	23	FNUMBR	- " " 16	
	24	ISERCH	- " " 17	
	25	XSRCH	- " " 18	
	26	YSRCH	- " " 19	
	27	BWIDTH	- " " 20	
	28	CPOWER	- " " 21	
	29	HGTAC	- " " 22	
	30	ILXTRA	- " " 23	
	31	XLITE	- " " 24	
	32	YLITE	- " " 25	
	33	FLARHT	- " " 26	
	34	AINTNS	- " " 27	
	35	MODE	- " " 28	
	36	Unused (4)	- Dummy filler	
	37	RMIN	- Vol II, Page F-60, Item 3	
	38	RMAX	- " " " " " 4	
	39	CVANGL	- " " " " " 5	
	40	CTCRCT	- " " " " " 6	
	41	NREPRI	- " " " " " 7	
	42	RINCR	- " " " " " 8	
	43	CMEGA	- " " " " " 9	

USE	WORD	VARIABLES		USED BY
COMMON/SENSOR Con't				
	44	BEAMW	- Vol II, Page F-60, Item 10	
	45	ASPEED	- Average speed of sensor	
	46	IDTLS	- Vol I, Page 2-236, Same as IDTL	
	47	KTLS	- Vol I, Page 2-234, Same as KTL	

* *

USE	WORD	VARIABLES		USED BY
	16	PRIMFR	- See item 9 when used by RADAR	EX3 RDR (continued)
	17	CLIMFR	- " " 10 " " " "	
	18	RAMBDA	- " " 11 " " " "	
	19	FNKTB	- " " 12 " " " "	
	20	RADCAR	- " " 13 " " " "	
	21	SCANRT	- " " 14 " " " "	
	22	ICOH	- " " 15 " " " "	
	23	BEAMAZ	- " " 16 " " " "	
	24	BEAMEL	- " " 17 " " " "	
	25	RGATE	- " " 18 " " " "	
	26	SIGSTB	- " " 19 " " " "	
	27	FCUTLO	- " " 20 " " " "	
	28	FCUTHI	- " " 21 " " " "	
	29	HGTANT	- " " 22 " " " "	
	30	PFA	- " " 23 " " " "	
	31	FDOPLR	- " " 24 " " " "	
	32	FTGMIN	- " " 25 " " " "	
	33	FTGMAX	- " " 26 " " " "	
	34	TIME	- Game time, real	
	35	Unused (5)	- Dummy filler	
	36	RMIN	- Vol II, Page F-60, Item 3	
	37	RMAX	- " " " " " 4	
	38	CVANGL	- " " " " " 5	
	39	CTCRCT	- " " " " " 6	
	40	NREPRI	- " " " " " 7	
	41	RINCR	- " " " " " 8	
	42	OMEGA	- " " " " " 9	
	43	BEAMW	- " " " " " 10	
	44	ASPEED	- See EX3 IMG, item 45	
	45	IDTLS	- Vol I, Page 2-236, Same as IDTL	
	46	KTLS	- Vol I, Page 2-234, Same as KTL	

* *

USE	WORD	VARIABLES		USED BY
COMMON/SENSOR Con't				
	16	TEMPEV	- Background Temperature	EX3 THV
	17	FOCALL	- Vol II, Page F-22, item 20	
	18	RESOL	- Vol II, Page F-23, item 21	
	19	FNUMBER	- Vol II, Page F-23, item 22	
	20	HTGAC	- Aircraft Height	
	21	TIME	- Game time, seconds	
	22-28	Unused (18)	- Dummy filler	
* * * * * * *	* *	* * * * * * *	* * * * * * * * * * * * * * * *	* * * * * * *
	16	TLTC	- Time of last threshold crossing	EX3 MAG
	17	THRESH	- Vol II, Page F-23, item 21e	
* * * * * * *	* *	* * * * * * *	* * * * * * * * * * * * * * * *	* * * * * * *
	16	EXPAN	- AREA X FIELD X DEVXMN	EX3 PIR
	17	FIELD	- Field of View	
	18	TEMPEV	- Temperature of background	
	19	WATTBK	- Background power incident on sensor	
* * * * * * *	* *	* * * * * * *	* * * * * * * * * * * * * * * *	* * * * * * *
	16-28	Unused (24)	- Dummy filler	SCNOUT
* * * * * * *	* *	* * * * * * *	* * * * * * * * * * * * * * * *	* * * * * * *
	29	RMIN	- Vol II, Page F-60, Item 3	EX3 THV
	30	RMAX	- " " " " " 4	SCNOUT
	31	CVANGL	- " " " " " 5	
	32	CTCRCT	- " " " " " 6	
	33	NREPRI	- " " " " " 7	
	34	RINCR	- " " " " " 8	
	35	OMEGA	- " " " " " 9	
	36	BEAMW	- " " " " " 10	
	37	ASPEED	- Average sensor speed	
	38	IDTLS	- Vol I, Page 2-236, Same as IDTL	
	39	KTLS	- Vol I, Page 2-234, Same as KTL	
* * * * * * *	* *	* * * * * * *	* * * * * * * * * * * * * * * *	* * * * * * *

USE	WORD	VARIABLES	USED BY
COMMON/SENSOR Con't			
	7	AREADN - Vol I, Page 3-65	ARFTG
	8	IMAG - Vol I, Page 3-65	
	9	IGEOM - Vol I, Page 3-31	
	10	DIMMAX - Vol I, Page 3-31	
	11	WIDTH - Vol I, Page 3-31	
* * * * * * * *	* *	* *	* * * * * * * *
	7	TLTC - Vol I, Page 3-66	MAGTG
	8	THRESH - Vol I, Page 3-66	
	9	Unused (33) - Vol I, Page 3-66	
* * * * * * * *	* *	* *	* * * * * * * *
	7	EXPAN - Vol I, Page 3-57	PIRTG
	8	FIELD - Vol I Page 3-57	
	9	TEMPEV - Vol I Page 3-57	
	10	WATTBK - Vol I Page 3-57	
	11	Unused (31) - Vol I Page 3-57	
* * * * * * * *	* *	* *	* * * * * * * *
	14	IFIXGN - Vol I, Page 3-37	SEISTG
	15	Unused (27) - Vol I, Page 3-37	
* * * * * * * *	* *	* *	* * * * * * * *
	7	TEMPEV - Background temperature	THERML
	8	FOCALL - Focal length of viewer	
	9	RESOL - Resolution of viewer	
	10	FNUMBR - F-number of viewer	
	11	HTGAC - Height of aircraft	
	12	Unused (30) - Unused dummy	
* * * * * * * *	* *	* *	* * * * * * * *

USE	WORD	VARIABLES	USED BY
COMMON/TARGET/			
(Definition and names of variables in arguments list vary based on 3d level MSM routines using the parameters.	1	ITTAB (9, 10) - Target parameter list ID	EX3 ARF
	2	Unused (120) - Unused dummy	EX3 BKW
			EX3 MAG
			EX3 PIR
			SEISTG
			ACOUTG
			SACDET
Linkage: *			
Target Paremeters to Sensor routines (First 6 variables remain constant during execution, next 7 vary with time.)	1	ITTAB (9, 10) - Target parameter list ID	EX3 SAC
	2	LEG NO (10) - Leg of path target presently on	
	3	NOELEM (10) - Number of elements in target	
	4	IDCODE (10) - ID code of target (one of 19)	
	5	Unused (90) - Unused dummy	
* *			
	1	IDCODE (16) - Target code	EX3 IMG
	2	IDTGT (16) - Target ID	EX3 RDR
	3	LEGNO (16) - Leg number on path	EX3 THV
	4	ITGTTP (16) - Target type	
	5	KSTRNG (16) - Strength constant, Vol II, Page F-56	
	6	NOELEM (16) - Number of elements in target	
	7	XTGT (16) - X coordinate of target	
	8	YTGT (16) - Y coordinate of target	
	9	RTGT (16) - Range from target to sensor	
	10	AZTGT (16) - Azimuth to target	
	11	VRADL (16) - Radial velocity of target	
	12	NELFLD (16) - Number of elements in field	
	13	KTLT (16) - Detection counter	
	14	NRBTGS - Number of RED/BLUE targets	
	15	NFTGTS - Number of false targets	

Section 3

SENSOR PERFORMANCE MODELS

3.1 INTRODUCTION

3.1.1 General

The sensors of the Systems Assessment Model are the devices (expressed in the format of computer subroutines) through which possibilities for detection of enemy, friendly, or other target types are developed. The sensor subroutines are limited in scope to the determination of probabilities of detection and certain other expressions of sensor performance given that a target is within the field of view and within some maximum range of the sensor.* To determine the sensor-target interaction outcomes, a considerable portion of the Systems Assessment Model is employed to specify sensor-target encounters, conditions prevailing at the times of encounters such as atmospheric and cultural environment, terrain data, foliage data, light levels and the like. In addition, sensor parameters, as defined by the planner or taken from designer tables, are required input as also is the description of the target, in terms of type, rate of movement, structure and the like. In this section, discussion is limited to the sensor subroutines except where for clarity or completeness reference to other subroutines is required.

3.1.2 Subroutines and Classification

The subroutines to be described are the following:

Background Routines	Sensor Performance Subroutines
SEISBK	SEISTG
ACOUBK	ACOUTG
ENVIR	PIRTG
PIRBK	ARFTG
ARFBK	MAGTG
BWIRBK	RADAR
	THERML
	IMAGE
	BRKWIR

The sensors considered have capabilities for detection which are functions of target characteristics, sensor parameters, and the environment. Target characteristics as seen by the sensors are functions of time and can be treated properly only as a part of the main simulation. Thus, the sensor performance subroutines are included as part of the Main Simulation Model (MSM). However, some factors which influence sensor performance, notably environment, are factors which remain relatively constant over a period of time long compared to some target-sensor engagements. Where

* Estimation of additional target information is developed in subroutines ANALMN (6.4) and attended sensor analysis (6.5) making use of sensor subroutine outputs and other data such as prior detection history, sensor location, sensor position in array, geometry of engagement, and others.

practical, the long term effects are developed as part of the PRERUN processing. Thus the gain of an adaptive sensor, adapting to background noise, can be determined for each period of constant background and can then be supplied to the MSM as a parameter. False alarm statistics in particular are a major computation of PRERUN, being independent of target activity. The six background routines of the above table are thus included as part of the PRERUN model (PRERUN).

It is to be noted that all sensors do not have background elements as part of the PRERUN. While all sensors have performance characteristics modified by the environment, data processing convenience in some cases dictates that background or environmental effects be treated as part of the MSM. For example, the clutter return from a clutter patch in radar problems could be computed in PRERUN but the size of the patch and its range from the sensor will not be defined until the range of the target is defined. A similar argument applies for atmospheric attenuation, background irradiance and others. Thus some background computation will be found in sensor performance subroutines of the MSM, in particular those relating to radar, thermal viewer, and imaging type sensors.

3.1.3 Sensor Types

It can be seen that the generic types of sensors treated in the simulation are nine in number. However, each subroutine is designed to accommodate a number of actual equipment types to the limits imposed by assumptions on which the models are based. For example, the seismic sensor model (SEISBK and SEISTG together) will provide a simulation for the Minisid, the Handsid, the ATMOD, and a number of others including fixed gain and adaptive gain control equipments. A limitation is imposed by the simulation of the sensor logic where detection determinations are based on signal to threshold ratios. Some seismic or acoustic sensors may have processors which enhance detection of vehicles by filtering techniques. Because of the great variability in the seismic path loss particularly as a function of frequency, a frequency dependency could not be developed within the limits of the present study. Hence, such specific detail is not treated at this time.

The radar subroutine is basically limited to simulation of MTI radars at this time and more specifically to MTI radars operating against ground targets. Radars simulated at this time include the PPS-4, PPS-5, PPS-9, TPS-25, APD/9, and CS II. This listing may be easily extended to additional ground and airborne radars as data is acquired. Some radars such as the MPQ-4 are MTI equipments that are not simulated in their primary role at this time (for the MPQ-4 in the counterbattery role), because of the target/ground clutter relationships. Changes to accommodate such expansion are not extensive, but have not been included because of time and data limitations.

The IMAGE subroutine embraces a wide range of sensors including natural sight, binocular-aided vision, passive night vision devices, low light level and daylight TV. Requirements imposed are only that the proper set of equipment descriptors be provided.

It is not to be implied that the models have been tested against the full range of equipments. For the IMAGE routine, simulation exercising was carried out for natural sight, binocular-aided sight, and a number of night vision devices, namely the TVS-4, PVS-2 and TVS-2.

3.2 BACKGROUND SUBROUTINES

3.2.1 General

Some sensor types, and particularly the general class of remote unattended sensors, have background noise and false alarm statistics which are essentially independent of range.

Since the parameters on which background noise is based are established as part of the PRERUN processing, it was decided to include those elements of the sensor routines in PRERUN as well, thus eliminating a substantial part of processing required for target detection from MSM in which time and speed are important considerations. In the following sections, each of the sensor background routines are presented.

At the initiation of sensor subroutine development, attempts were made to define in as much detail as possible the relations between driving functions and system responses for noise (and target) signals at the sensor. Examples of the general nature of the approach are contained in the functional flow diagrams, Figures 3.2-1 through 3.2-4.

Studies were conducted chiefly through examination of the pertinent literature in order to provide the mathematical relations linking cause and effect. The references listed at the end of this section are a significant but far from complete set. It was found in many cases that very little data was available and that in some instances the data was lacking in specific detail to link cause and effect. Because of this void, it was necessary to develop arguments relating cause and effect on the basis of interpretations of the limited data, on specific findings reported in the literature, on examination of the broad background of geophysics, and in some cases on intuitive reasoning. Thus some of the arguments are open to question and it is anticipated that reviews of existing engineering data and new field data will provide the basis for correction, refinement, or verification of the arguments proposed. The models are structured in such a way that updating may easily be incorporated and even restructuring of the cause and effect arguments including addition of some not foreseen can be carried out within the limits of the information provided to the subroutines through the CALL, COMMON, and DATA statements. These models are considered to provide relatively crude estimates of noise levels and false alarm rates but nonetheless are considered adequate for present simulation purposes in that the output levels and rates agree to a significant extent with the limited field data.

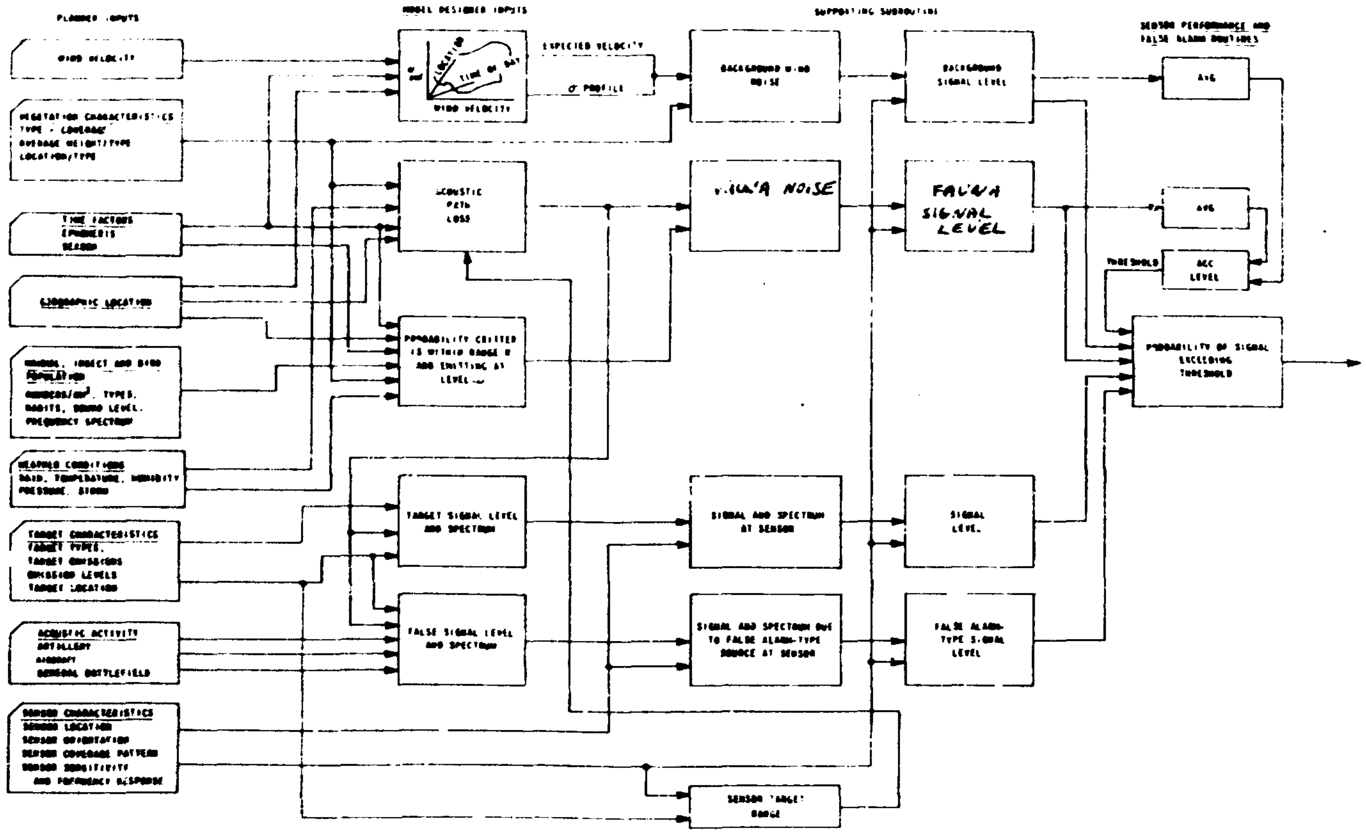


Figure 3.2-1 THEORETICAL FUNCTIONAL FLOW DIAGRAM - ACOUSTIC SENSOR BACKGROUND

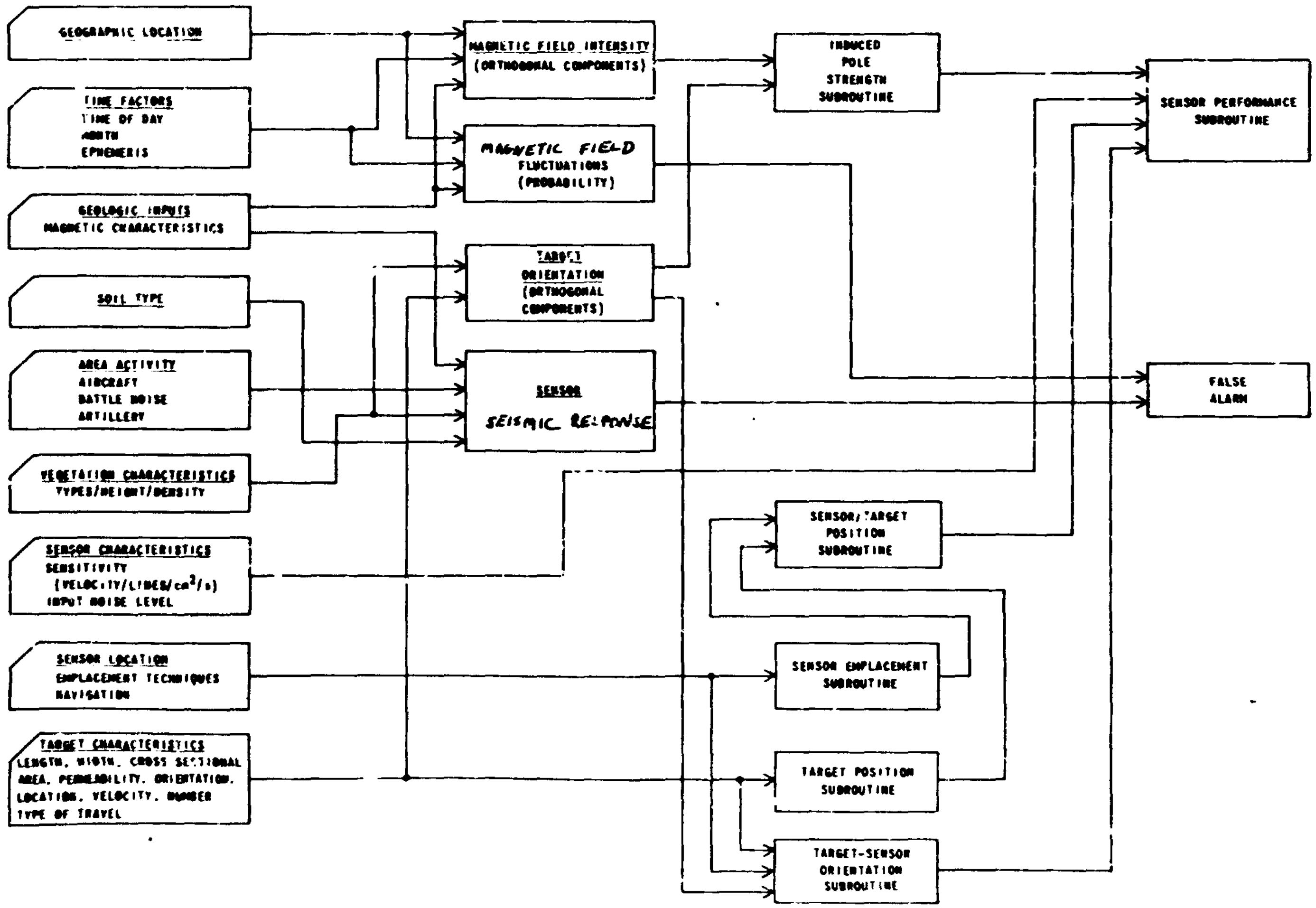


Figure 3.2-2 THEORETICAL FUNCTIONAL FLOW DIAGRAM - MAGNETIC SENSOR BACKGROUND

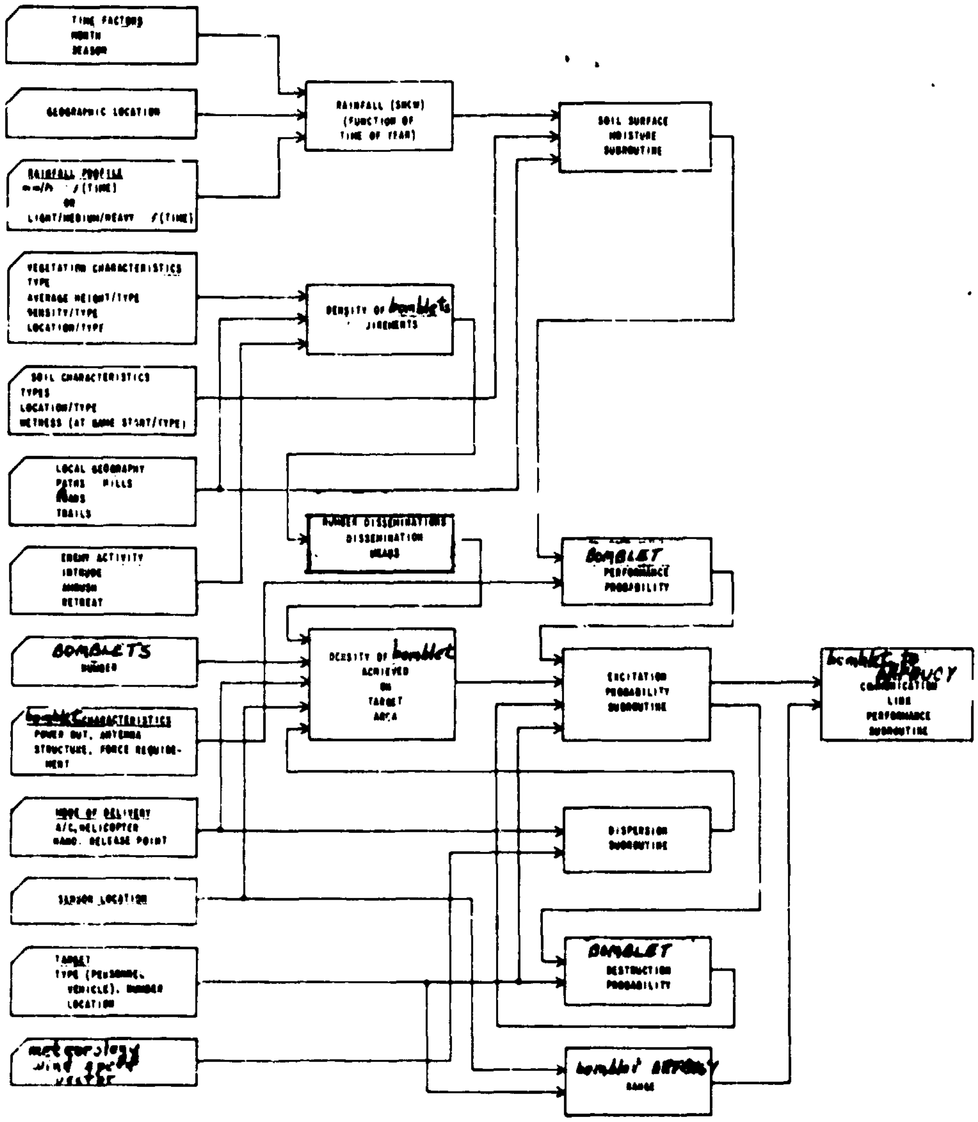


Figure 3.2-3 THEORETICAL FUNCTIONAL FLOW DIAGRAM - ARFBUOY SENSOR BACKGROUND

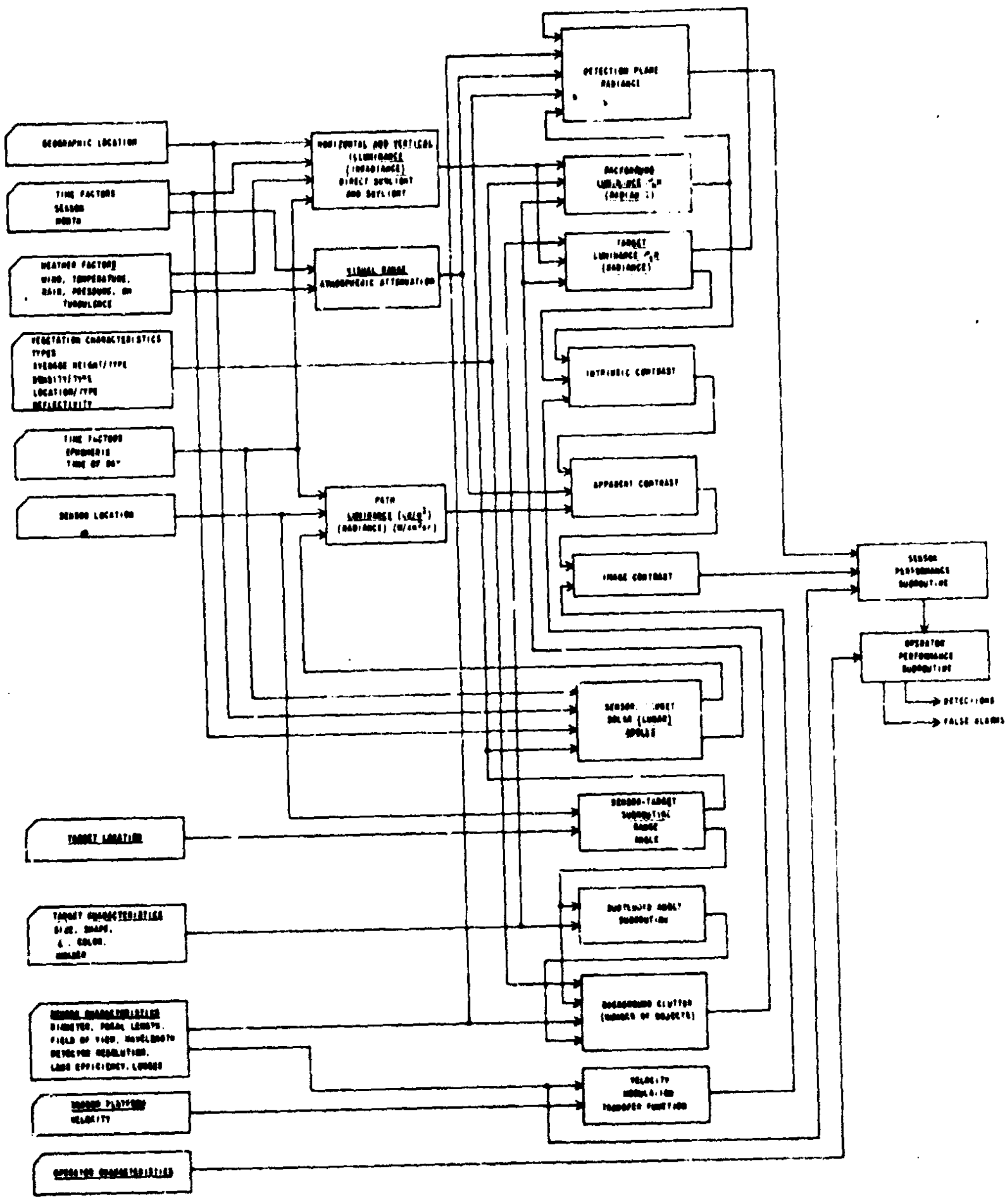


Figure 3.2-4 FUNCTIONAL FLOW DIAGRAM - LCW LIGHT LEVEL TELEVISION, PASSIVE NIGHT VISION DEVICE, INFRARED AND VISUAL BACKGROUND

It is emphasized that the structure is available within these subroutines to allow expansion as additional pertinent data and relations become available. The major needs so far as data input is concerned are included in the common statements giving atmospheric (ATMENV) and terrain (UNTER) data for each time period of interest. Thus, for example, effects on seismic background due to soil type, temperature, wind-rain correlation, time of day, solar altitude, and others can easily be introduced without major program revision, provided the interrelations can be defined.

The descriptions that follow have been prepared to serve as a guide by means of which the reader may follow the listing of the programs being described. It is not essential that the program listings be consulted for reading the following. It is emphasized, however, that the comment statements in the program listing also provide valuable insights into the program organization. All of the designer input values required for functioning of the program are contained in those listings.

3.2.2 Subroutine SEISBK

3.2.2.1 Purpose

Subroutine SEISBK is employed in PRERUN to establish the seismic noise environment in which each seismic sensor will be operated. From the environmental data, the noise level at the sensor and the root mean square (RMS) output noise level (for fixed gain sensors) or amplifier gain (for AGC type sensors) is determined for use in the MSM.

3.2.2.2 Glossary of Inputs, Computed Values, and Outputs

Input Values

DBBATL	Battle Noise (dB)
DBCULT	Cultural Noise (dB)
IFIXGN	=0 No Fixed Gain, =1, 2, 3, 4, 5 Planner Set Gain, =6, Sel. Gain, = Negative Number Fixed for Game
ISEXP	Index for Buried: 1=0.0 (Buried), 2=6.0 (Not Buried)
ITREE	Index on Tree Distance: 0=Tree, 1=No Tree
IUT	Index on Unit Terrain

Labelled Common Inputed Values

BIASSE	Threshold Setting (Volts)
BWSEIS	Effective Band Width of Noise Signal, in Hertz
CONSTS	Average Amplifier Output
IPRINT	Output Data Device Designator = 6
ISM	Index Describing Soil Moisture Conditions
IVCOV	Index Vegetation Cover
LDUMP	True = Intermediate Calculations Printed, False = No Print
PRATE	Rain Fall Rate (mm/hr)
PTOT24	Total Precipitation During the Last 24 Hours
WSPEED	Wind Speed (km/hr)

Internally Stored Designer Input Values

BETA	AVGTHC Modifier for Fixed Gain Sensors
FBATL	Value for Battle Effect, 1=1.5 (low int.), 2=.15(M), 3=.1(H)
FCULT	Value for Population Effect, 1=.25(Rem.), 2=.15(Rur.), 3=.1(Urb.)
FRAIN	Value for Rain Effect, 1=.5(Low), 2=.3(Mod.), 3=.2(Heavy)
FWIND	Value for Wind Effect, 1=.5, 2=.4, 3=.3, 4=.2, 5=.2
SENSEX	Effect of Sensor Being Buried
SET	Table of Gain Values
SOILM	Soil Effects on Rain Noise
VEGCVR	Vegetation Cover Effects on Rain Noise
VEGCVW	Vegetation Cover Effects on Wind Noise

Computed Values

DBRAIN	Rain Noise (dB)
DBWIND	Wind Noise (dB)
DELTA	Modifier of False Alarm Rate, Fixed Gain System
DUM	Random Number 0-1
IFIXGS	Absolute Value of IFIXGN (Index for Gain Selector)
ISOILM	Index - Soil Modifier, 1=Dry (0), 2=Wet (3), 3 = Very Wet (6), 4=Saturated (6)
IVCOVR	Index Vegetation Cover, 1=Heavy, 2=Med. Forest, 3=L. Fol., 4=H20, 5=Open
KBATL	Index Battle Noise, 1=1.5 (Low Int.), 2=.7 (Med.), 3=.4(High)
KCULT	Index Cultural Background
KRAIN	Index for Rain Conditions
KWIND	Index for Wind Gustiness
L	Dummy Index
ONOISE	Output Noise for Fixed Gain System
VBATL	DBBATL Converted to Voltage
VCULT	DBCULT Converted to Voltage
VRAIN	DBRAIN Converted to Voltage
VWIND	DBWIND Converted to Voltage

Output Values

AVGTHC	Average Time Between Threshold Crossings (Seconds)
GEQUIL	Amplifier Gain
THRESH	Threshold (Volts)
VNOISE	RMS Sum of Background Noise Voltages

3.2.2.3 Description of Subroutine

a. Outline of Problem. Seismic waves are generated by a wide range of sources and may propagate in many modes. The study of earthquakes, tremors, high energy explosives, and the use of seismics in geological surveying are well known. Because of the fact that vehicles and troops also give rise to seismic waves which are of sufficient strength to be observed at some distance from the actual source, seismic sensors have found use as intrusion detection devices. While in general waves of a number of types can be propagated, those of interest in the intrusion detection problem are surface waves. Thus the intruder, interacting with the earth's surface through his movement, causes seismic surface waves to be generated and propagated. The generation and propagation processes are very complex depending on the surface characteristics, particle size, dryness or hardness, vegetation, rock formations, surface undulations and the like. Because of these factors source strength and path loss are found to be highly variable from location to location in a given area and even more so when widely different areas are compared.

In seismic processing and similar operations where data storage and extensive post processing are allowed, waves of a number of types and measurements at a number of locations can be treated. While array processing of real time signals may some day be introduced in the intrusion detection scheme, current state of the art and tactical limitations and economic considerations require that in vast cases relatively simple processing must be employed in an unattended sensor, the output of which will in general be a "yes-no" detection decision.

In this simulation, an output report is generated whenever the signal due to a target exceeds a threshold level computed or set for the conditions prevailing. Thus the target detection aspect of the problem consists of computing the target strength at the sensor amplifier output and comparing that level with the threshold level. That threshold level may, however, also be exceeded by the noise in the sensor output. Further, the threshold may be made adaptive to the noise level to limit the false alarm rate for high noise conditions (reducing of course sensitivity to target signals at the same time). It is the function of the SEISBK to establish the noise level at the sensor input and at the sensor output. For a fixed gain sensor the output noise level is variable while for an adaptive sensor, the output noise level is fixed but the gain is varied.

In a number of seismic programs, observation of seismic background levels were undertaken, for example in References 1 to 3 (page 3-126). While the seismic background is seen to be a function of time and location, under very quiet conditions, levels of the order of -120 dB relative to a particle velocity of one centimeter per second have been observed. This level is undoubtedly a function of the general location of the sensor and probably varies over a substantial range over the earth's surface. A seismic sensor (as for example, the mass, spring, damper system that constitutes a geophone), if properly constructed, will have a sensitivity adequate to respond to the -120 dB signal level. In general, however, noise background will be significantly greater than -120 dB even in relatively quiet areas. Wind, rain and other sources will have an appreciable effect on the noise level. One of the main purposes of the SEISBK subroutine is the development of the noise background level to which each seismic sensor is exposed for each period of constant environment as developed in the PRERUN processing.

b. Outline of Subroutine Flow. The simulation of seismic sensors follows the general outline given below and is shown in Figure 3.2-5.

1. Components of noise due to each of the general forcing functions(wind, rain, battle and cultural activity)are developed. Only the first two are developed in the seismic subroutine. Battle and cultural noise are provided from other subroutines.

2. These noise levels are converted to voltages (taken to be RMS values) and are summed by the root mean square procedure.

3. For adaptive sensors, the gain required to achieve an output RMS voltage level of one volt is computed and assumed to apply. For fixed gain sensors the gain employed is selected by one of three means based on input data.

4. The average threshold crossing rate for noise is computed.

5. The values of gain, threshold, output noise level and average threshold crossing rate (GEQUIL, THRESH, VNOISE, and AVGTHC) are supplied to other subroutines and to the MSM for further computations.

c. Assumptions Embodied in the Subroutine.

1. It is assumed that the microseismic limit of -120 dB is effective for all seismic sensors.

2. The noise level to which the sensor is exposed will be increased by wind, rain, battlefield, and cultural effects. Factors as high as 10^3 (60 dB) can be expected for battlefield and high level cultural activity.

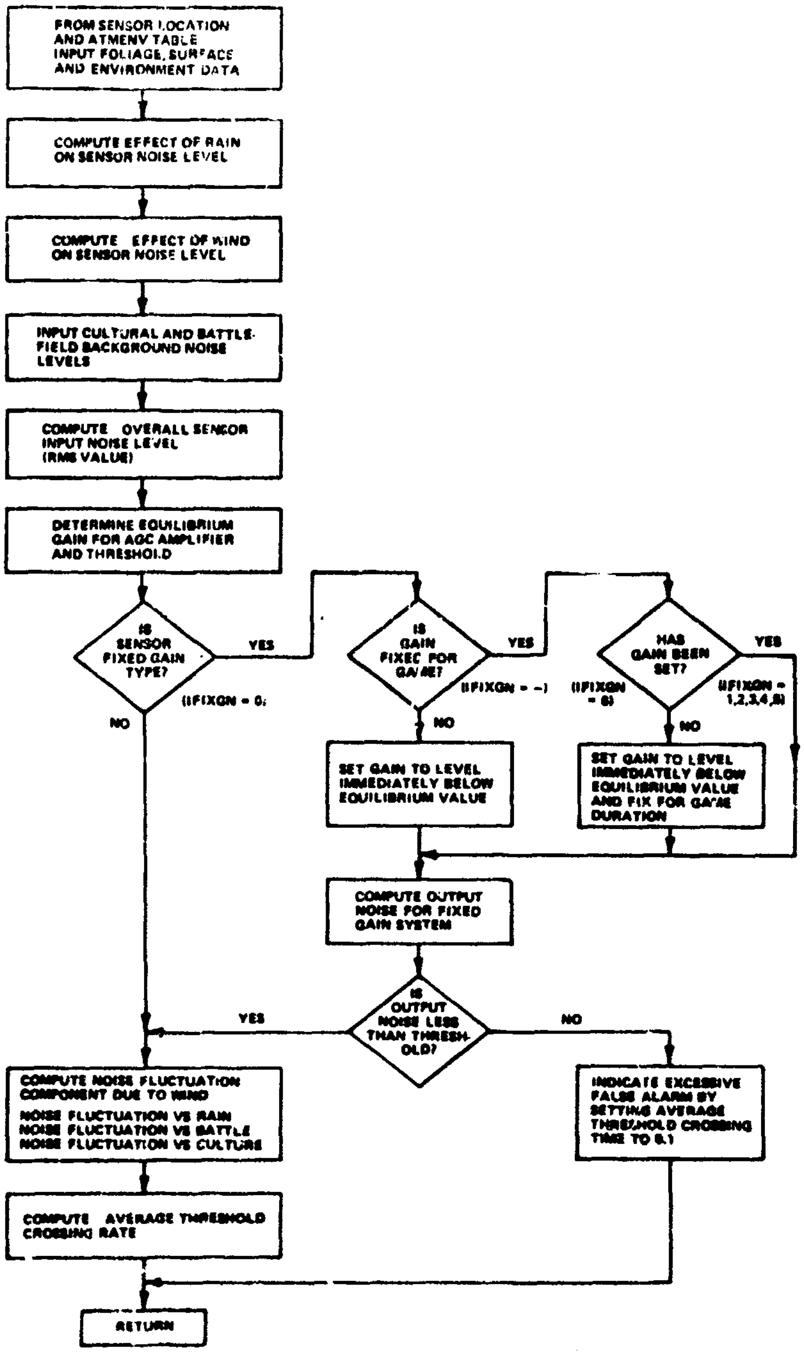


Figure 3.2-5. SEISMIC BACKGROUND MACROFLOW

3. The noise is considered to possess a Gaussian distribution, the -120 dB level cited above being the RMS value. The variance of the Gaussian distribution will be related to the magnitude of the several noise components. The variance is the determining factor in false alarm (threshold crossings by noise) determinations.

4. All sensors employed are of design adequate to sense the background noise level. Sensors are not limited by internal electronic noise.

5. All sensors are designed to have gains adequate to raise the input signal level (as low as -120 dB) to an output level of 0.0 dB (0.0 dB equals one volt thus giving an assumed geophone sensitivity of one volt/cm/sec.). Fixed gain sensors have specific gains assigned, but for geophones of the same sensitivity defined above.

d. Description of Subroutine Logic and Processing. This subroutine derives data from a number of areas. As can readily be observed in the listing (see Volume III), the common areas CONST, BASICT, ATMENV, SENVAR and UNTER are referenced. In addition, the calling sequence provides additional input information specific to each sensor namely the IUT value, DBCULT and DBBATL, ISEXP, ITREE, and IFIXGN, which refer to unit terrain type in which the sensor is placed, the seismic background levels due to cultural and battle noise, the index on sensor exposure (buried or ground emplaced), the nearness of the sensor to trees, and the type of gain control employed in this sensor. Data stored within the routine include effects of soil moisture (SOILM), vegetation cover for rain effects and wind effects (VEGCVR, VEGCVW) and the variance factors associated with the several levels of wind, rain, battle, and cultural noise (FWIND, FRAIN, FCULT, FBATL). Also included in the stored data set are the five values of fixed gain that may be assigned to a fixed gain sensor.

The index defining vegetation coverage is taken directly from the input description of foliage (IUT) at the sensor location. The index of soil moisture (ISOILM) is taken directly from the unit terrain (UNTER) table, but requires use of a dummy index (L) for complete specification. L is an integer function of the integrated rainfall over the preceeding 24-hour period (PTOT24) and can take on the values 1, 2, or 3.

The noise level in dB due to rain is then derived. The equation provided shows the empirical relation between noise, rainfall rate, soil moisture, vegetation coverage, and exposure as:

Noise = 20 Log_{10} (Rainfall Rate + 1.0) + Soil Moisture Effect + Vegetation Cover Factor + Sensor Exposure Factor.

The effect of wind speed is also considered to be logarithmic with its influence modified by vegetation coverage and nearness to trees. A sensor emplaced in heavy vegetation is considered to be less strongly influenced by wind than a sensor emplaced in an exposed area. A sensor emplaced near a tree in an exposed area will also be subjected to a higher noise level than one far removed from trees. The effects of wind speed, an input variable from atmospheric environment common area, are contained in the following statement including the getation and emplacement modifiers.

$$\text{Wind noise (dB)} = 20 \log_{10} (\text{wind speed} + 1.0) + \text{Vegetation Cover Factor} + \text{Tree Amplifier Factor}$$

The final statement thus assigns an increase of six dB to the background wind component for a sensor emplaced near a tree. The data describing effects of vegetation are stored in the subroutine under the label VEGCVW and are selected on the basis of the index IVCOVR developed from unit terrain (UNTER) table data.

Input noise levels from the cultural and battlefield background subroutines are contained in the arguments DBCULT and DBBATL in the subroutine calling sequence, and are available as inputs to the subroutine from cultural and battle subroutines.

As shown in the flow diagram and listing, each of the noise levels in dB is converted to a noise voltage by statements such as:

$$\text{Wind noise voltage} = 10^{\frac{(\text{Windnoise}-120)}{20}}$$

This statement gives the conversion from dB to voltage with the note that the reference level of -120 dB is introduced. All the noise levels are computed with respect to this reference and to convert to voltage it is necessary to introduce the reference. The reference is to be taken as the minimum seismic noise level that will be experienced. Converted to particle velocity, the minimum level is seen to be 10^{-6} centimeters per second. This level may vary somewhat over the earth's surface, but adequate data for further expansion of the model is not available at this time. Note that the thermal noise* at the amplifier input is taken to be less than 10^{-6} volts and that the conversion from particle velocity to voltage is taken to be unity, 10^{-6} cm/sec is equivalent to 1 microvolt. This assumption implies a specific transducer sensitivity but if the sensitivity is adequate to observe the microseismic level before thermal noise becomes predominant, no loss in generality is introduced. Sensors of adequate design will generally meet this requirement.

*See References 8,9,12, & 16

Having computed the various component noise voltages, a composite of the background noise level can be determined by computing the RMS value derived from the components.

A substantial fraction of the seismic sensors presently employed are designed to be adaptive to the noise environment by inclusion of an automatic gain control system in which the amplifier output noise level is held constant for varying input levels through a feedback gain control system. The average output level of the amplifier, CONSTS, located in the SENVAR common input area is assigned to be one volt. The actual equipment value may differ from this value, but since the gain is arbitrary, it will differ from the equipment amplifier gain in the same ratio as the thresholds. The gain level, GEQUIL, is found by the statement:

GEQUIL = CONSTS/VNOISE

where VNOISE is the RMS noise level. GEQUIL is an output parameter that is supplied to SEISTG subroutine in the MSM.

Fixed gain sensors must also be considered. It is assumed that these sensors will have five gain settings available, the settings differing in sequence by 6 dB with position 1 having the highest gain, 6.31×10^4 volts/volt as contained in the data set of this subroutine. An index, IFIXGN, is used to denote the planner or sensor parameter designer dictated selection of gain type. The following index descriptions apply:

IFIXGN = 0	AGC System
1, 2, 3, 4, 5	Fixed Gain Selected by Planner and Fixed for Game Duration
6	Fixed Gain Selected by Routine on First Entry and Maintained by Game Duration
Negative Number	Fixed Gain Selected by Routine on Each Entry to Simulate Commanded Sensors

For IFIXGN = 6 or negative number, the fixed gain value that is closest to the AGC gain value on the low gain side is selected by the subroutine. IFIXGN settings of 1 through 5 are not likely to be employed because the planner does not have *a priori* information on the background noise level and, therefore, has no basis for selection. It may, however, be a useful method of parametrically examining effects of gain changes.

For the fixed gain equipments the amplifier output RMS noise level is computed. Note that as the game progresses, the background noise level will change. Increases in background can be such that the output RMS noise level is greater than the threshold in which case a high false alarm rate will be experienced. If this should be the case, the average threshold crossing rate (AVGTHC) is set to 0.1 seconds. For all other conditions, however, the AVGTHC is developed from stored data and indices developed from each of the component noise levels. The basic AVGTHC term is developed in the statement:

AVGTHC = (Wind Effect + Rain Effect + Cultural Effect + Battle Effect) x 6.0

This statement is very much of empirical origin designed most specifically to produce, through the False Alarm (FAINTV) subroutine, false alarm events at rates that are consistent with field experience. There is only a small data base on which to draw and it is to be expected that by correlation with specific field data, modifications to this argument may be required.

For fixed gain sensors the AVGTHC value is modified to account for the fact that the fixed gain system RMS output (ONOISE) may differ significantly from CONSTS on which basis the AVGTHC empiricism is based.

The outputs from this subroutine are passed through the subroutine argument list and include amplifier gain (GEQUIL), RMS sum of background noise voltages (VNOISE), average time between threshold crossings (AVGTHC) and threshold (THRESH) where

THRESH = CONSTS + BIASSE

and CONSTS is typically 1.0 and BIASSE is 0.2.

3.2.3 Subroutine ACOUBK

3.2.3.1 Purpose

Subroutine ACOUBK is employed in PRERUN to determine the acoustic ambient noise environment in which each acoustic sensor will be operated. From environmental data in unit terrain (UNTER) and atmospheric (ATMENV) tables, battlefield and cultural backgrounds and time of day, the noise level at the sensor and the gain level of the AGC amplifier are determined for the MSM. In addition the average threshold crossing time (AVGTHC) is computed and supplied to the false alarm (FAINTV) subroutine for scheduling of false alarm events. Outputs from this routine are THRESH (amplifier threshold, a parameter defined by equipment parameters), GEQUIL (AGC determined amplifier gain level for noise only), VNOISE (input RMS noise level to the sensor), and AVGTHC already defined.

3.2.3.2 Glossary of Inputs, Computed Values, and Outputs

<u>Input Values</u>

DBBATL	Battle Noise (dB)
DBCULT	Cultural Noise (dB)

<u>Labelled Common Inputed Values</u>

BIASAC	Threshold Setting (Volts)
BWACOU	Bandwidth
CONSTA	Average Amplifier Output
IPRINT	Output Data Device Designator = 6
ITOD	Time of Day
LDUMP	True = Intermediate Calculations Printed, False = No Print
PRATE	Rain Fall Rate (mm/hr)
WSPEED	Wind Speed (km/hr)

<u>Internally Stored Designer Input Values</u>

FANTBL	Fauna Noise Table Selected by Index KFAUN
FBATL	Battle Noise Selected by Index KBATL
FCULT	Cultural Noise Selected by Index KCULT
FFAUN	Fauna Noise Selected by Index KFAUN

<u>Computed Values</u>

DBFAUN	Fauna Noise (dB)
DBRAIN	Rain Noise (dB)
DBWIND	Wind Noise (dB)
FRAIN	Rain Noise
FWIND	Wind Noise
KBATL	Index for Battle Noise
KCULT	Index Cultural Noise
KFAUN	Index Fauna Noise
VBATL	Battle Noise (Volts)
VCULT	Cultural Noise (Volts)
VFAUN	Fauna Noise (Volts)
VRAIN	Rain Noise (Volts)
VWIND	Wind Noise (Volts)

<u>Output Values</u>

AVGTHC	Average Time Between Threshold Crossings (Seconds)
GEQUIL	Amplifier Gain
THRESH	Threshold (Amplifier)
VNOISE	Total Background Noise

3.2.3.3 Description of Subroutine Logic and Processing

The arguments leading to the development of the noise level to which the acoustic sensor is exposed are similar to those for the seismic sensor. In order to determine the detection performance of a sensor when a target of given strength is located at a specific distance from the sensor it is necessary to compare the target signal at the sensor with the background noise whether that noise be derived principally from sensor internal sources or from the environment. Hence it is necessary to establish the external noise level to which the sensor is exposed (an assumption in the model being that internal noise will always be less than that due to external sources).

The noise environment for acoustic sensors may be due to one or more of a number of sources including wind, precipitation, cultural sources such as urban or rural activity, road traffic, heavy machinery, battlefield sources, especially weapons effects, animal, bird and insect noise and others. These sources are modified to some extent by foliage attenuation and the similar effects.

The flow of this routine is shown in Figure 3.2-6, where it can be seen that five basic noise sources are included. First the effect of rain is considered in the expression:

$$\text{Rain Noise (dB)} = 40 \log_{10} (\text{Precipitation rate} + 1.0) + 12.0$$

where the logarithmic relation between rain rate and noise is evident. At nine millimeters an hour the rain noise would be 52 dB while at 99 millimeters an hour 92 dB would be computed, all with respect to a reference level of 0 dB, equivalent to 10^{-16} watts/square centimeter or 2×10^4 dynes/square centimeter. The constant value, 12, included in this equation and in the equation for wind combine to define the assumed minimum background noise level to which the sensor would be exposed under field conditions.

Wind effects are seen to follow a linear relation, namely:

$$\text{Wind Noise (dB)} = 3.0 \left(\frac{\text{Wind Speed}}{4.0}\right) + 12.0$$

Two effects are included in this relation, namely the effect of wind on the microphone itself, important in the case in which the microphone is exposed to the wind, and the effect of wind on vegetation for the case where the sensor is located in or near foliage.

Battle and cultural inputs are provided from other routines and except for conversion to voltage are not processed further.

Another major source of noise is that due to the fauna in the locale of the sensor. Based on very limited data, the argument employed shows that maximum noise would be experienced in the 0300 to 0900 hour period with minimum from 2100 to 0300 hours, and with time being the only parameter of importance.

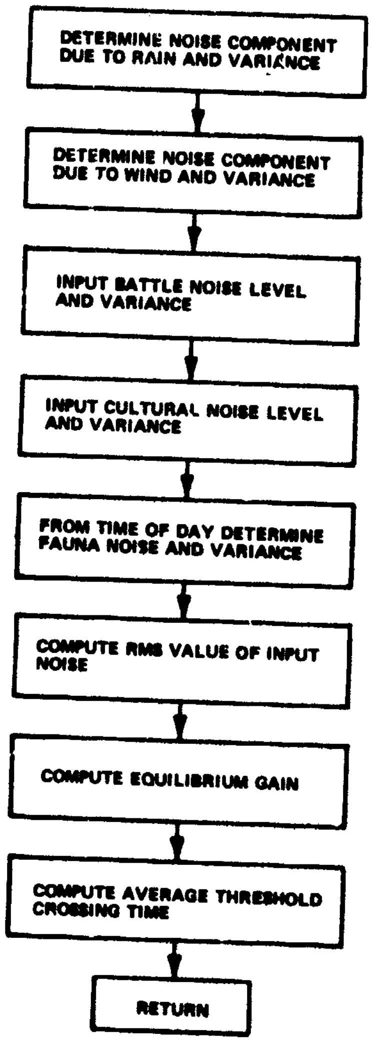


Figure 3.2-6 ACOUBK MACROFLOW

It must be evident that the relations given are very crude indicators for the real-world situation. In the case of fauna for example, the user must be aware of activity of many species, their acoustic levels, the population density for various types of terrain, vegetation coverage, climatic conditions, distance of major sources from the sensor and others. For both rain and wind, the placement of the sensor relative to vegetation, the vegetation coverage, type of terrain, soil type and moisture and other factors need to be considered. However, a review of the available literature shows that the environment is only poorly described in terms of acoustics and that for a more complete description, significant effort must be applied to the development of a field data based exposition.

Because of the wide variation in noise level, AGC is normally employed in which the RMS output level for a no-target condition is maintained at an average constant level (CONSTA). As the noise changes from period to period as defined by changes in atmospheric environment parameters, the gain is increased or decreased as appropriate to maintain the fixed output level. Since, as will be described in the ACOUTG subroutine, target signal can also modify the RMS long term output of the sensor, both the noise background in volts and the gain setting are supplied as outputs of ACOUBK to ACOUTG. A threshold (THRESH) is defined for the detection process by adding to the constant level (CONSTA), an offset threshold setting level BIASAC. For detection* the signal must exceed the threshold (THRESH = CONSTA + BIASAC) some prescribed number of times in a given time interval, but that level can be exceeded by signal or noise and hence statistical fluctuations in the noise level must be considered.

In this subroutine estimates of variance in RMS acoustic noise are developed in an empirical expression defining the average threshold crossing time. This parameter is described by the equation:

AVGTHC = (Wind Noise + Rain Noise + Battle Noise + Fauna Noise + Cultural Noise) x 6.0

Cultural and battle noise are considered to be most variable at lower levels of activity, that is, at low levels of the noise effects. Similar accounting is included for wind, rain, and fauna effects. ~~Future design effort should place~~

*The detection processes that may be employed in acoustic sensors are varied ranging from the threshold crossing type defined here to types that include an actual operator listening mode. While one mode may provide a decided detection advantage over another or a significant false alarm advantage over another for the same detection capability, significant effort would be required to define those differences. Hence at this time, only the single detection criterion is provided as representative of the capabilities of all systems. Should field results show a bias due to this assumption, modifiers for equipment types can be easily introduced.

emphasis on developing average threshold crossing time as a function of threshold and bias so that problems of a more general nature, i.e., study of effects of threshold, can be carried out.

3.2.4 Subroutine ENVIR

3.2.4.1 Purpose

A number of sensors contained in the STANO equipment listing are limited in performance by or depend for reference on the radiance of the background. The purpose of this subroutine is thus to compute background temperature and standard deviations of background temperature fluctuations for use in the sensor performance subroutines in the MSM. Depending on the wave length region in which the sensor operates that radiance may be provided by reflection of sun or moonlight, or by self emission of the background acting as a black or gray body at the temperature of the background. The latter case applies to sensors operating in the 7-15 micron region because radiance due to reflection of sunlight in this region is less than the radiance of the black or gray body background.

3.2.4.2 Glossary of Inputs, Computed Values, and Outputs

Input Values

IUT	Index Unit Terrain

Internally Stored Designer Input Values

CLOUDF	Cloud Factor Data
CLOUDV	Variance Factor Cloud Data
EIGHT	Constant (7.99999)
RAINF	Rain Factor Data
RAINV	Variance Factor Rain Data
TYPEF	Background Type Factor Data
TYPEV	Variance Factor Background Type Data
VEGF	Vegetation Factor Data
VEGV	Variance Factor Vegetation Data
WINDF	Wind Factor Data
WINDV	Variance Factor Wind Data

Labelled Common Inputed Values

ATEMP	Ambient Air Temperature
IBACK	Index Identification Most Likely Background Reflectance Function

Labelled Common Inputed Values (continued)

IPRINT	Output Data Device Designator = 6
ITOD	Time of Day
IVCOV	Index Vegetation Cover, 1=Heavy, 2=Medium, 3=Light, 4=Open, 5=Water
LDUMP	True = Intermediate Calculations Printed, False = No Print
PRATE	Rain Fall Rate (mm/hr)
SOLALT	Solar Altitude (Degrees)
TCLOUD	Transmission of Cloud Cover
WSPEED	Wind Speed (km/hr)

Computed Values

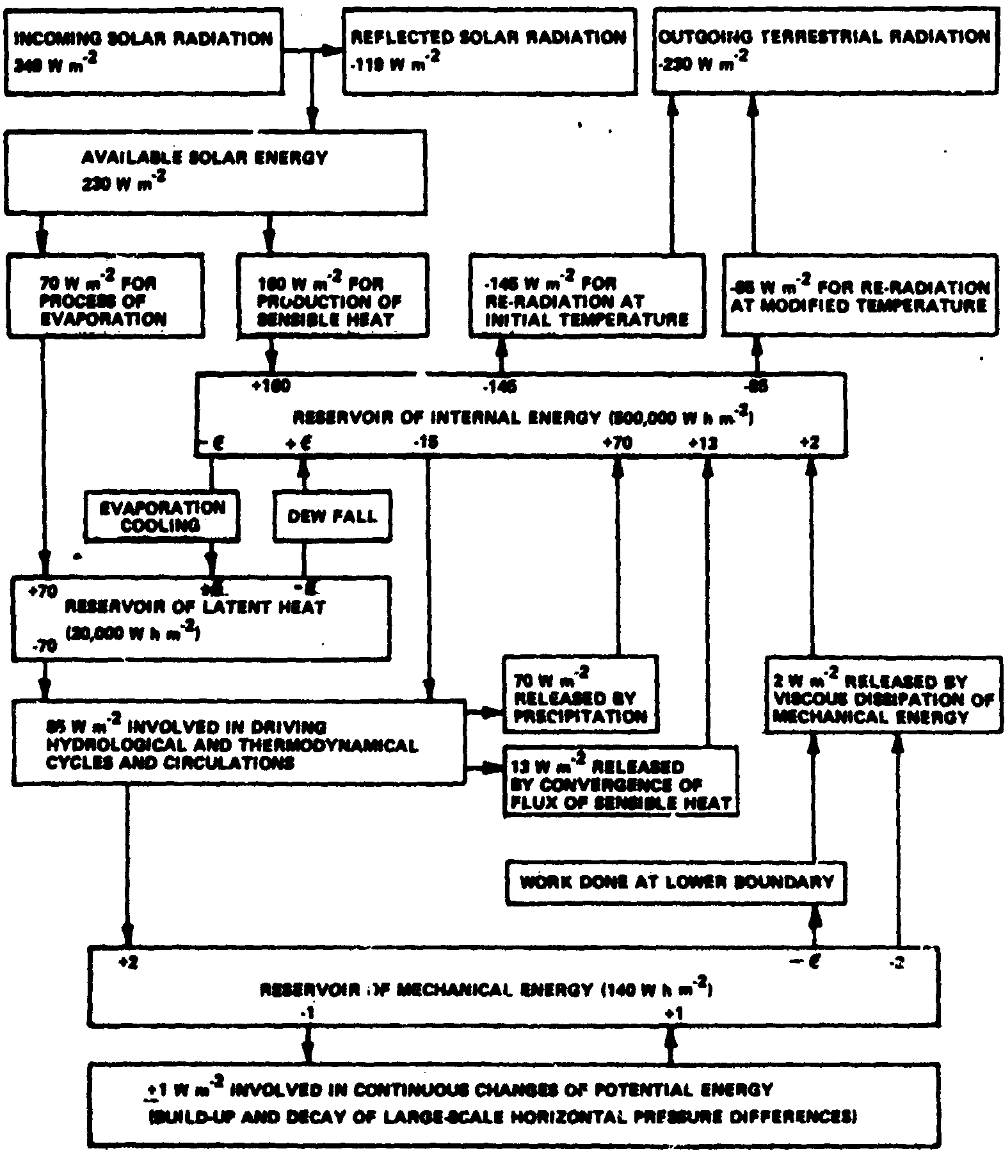
CFACT	Cloud Factor
CVAR	Variance Factor Due to Clouds
HRLOCL	Local Time
ICLD	Index on Cloud Cover
IRAIN	Index on Rainfall Rate
ITYPE	Index on Background Type
IVEGCV	Index on Vegetation Cover
IWIND	Index on Wind
RFACT	Rain Factor
RVAR	Variance Factor Due to Rain
TFACT	Background Type Factor
TVAR	Variance Factor Due to Background Type
VFACT	Vegetation Factor
VVAR	Variance Factor Due to Vegetation
WFACT	Wind Factor
WVAR	Variance Factor Due to Wind

Output Values

SIGMA	Standard Deviation of Background Temperature Fluctuation
TEMPEV	Background Temperature (C Degrees)

3.2.4.3 Description of Subroutine Logic and Processing

The temperature of the background is a complex function of the background type, its component parts, the environment, surface features, soil conditions, exposure to the sun, viewing angle with respect to sun-background direction and others. An examination of Figure 3.2-7 shows the total cycle of the major processes that are involved, the pertinence of each in a given location being a function of the above factors. Little work has been reported on the in situ measurement of background temperature in the detail required to identify background-environment interdependencies. Airborne measurement of surface temperature is now widely employed for several problems, but these show the temperature differences for small changes in distance without ascribing any description of the backgrounds so that a cause and effect might be observed. Because of the limited data, an argument describing the relationship between air temperature and background temperature was developed as a set of empirical dependencies derived from interpretation of the meager



*NOTE: FIGURE TAKEN FROM REFERENCE 4.

A SOLAR CONSTANT OF 1395 W m^{-2} AND A GLOBAL ALBEDO VALUE OF 0.34 ARE ASSUMED. THE AVERAGE TOTAL INCOMING RADIATION TO THE GLOBE IS 1/4 OF THE SOLAR CONSTANT. ϵ DENOTES AN AVERAGE RATE OF LESS THAN 0.5 W m^{-2}. THE ESTIMATED RELIABILITY OF THE SOLAR CONSTANT IS 3%; OF THE DERIVED ENERGY RATES, THIS TOTALS APPROXIMATELY 10% (LETTAU, 1954 a).

Figure 3.2-7 THE GLOBAL MEAN ENERGY CYCLES OF THE ATMOSPHERE literature. These relationships are the basic content of ENVIR subroutine using the atmospheric environment (ATMENV), unit terrain (UNTER) and time as inputs and provides estimates for background temperature (TEMPEV) and temperature fluctuations (SIGMA) as outputs.

The argument developed is that the background temperature will differ from the ambient air temperature by a factor:

$$\left(\frac{\text{Solar Altitude}}{5.0}\right)(\text{Vegetation Factor})(\text{Cloud Factor})$$
$$(\text{Wind Factor})(\text{Rain Factor})(\text{Temperature Factor})$$

where the altitude of the sun will be positive during the day and negative at night. Since all the modifiers are positive, the prediction is made that background temperatures will be warmer than ambient air during the day and cooler at night. Solar altitude can reach a maximum value of 90° for tropical regions so that the maximum difference between background and air temperatures will be 18°C because the multiplying factors range between zero and one.

The effects of the five modifying factors in the equation above are contained in the subroutine in a set of data statements. The indices required to select one set of values assigned to each factor are developed from input data describing wind speed, background type, vegetation cover, cloud cover, and precipitation. For example, the wind speed values between 0 and 50 kilometers per hour are divided into ten intervals, each of which is denoted by an index (IWIND) from the expression:

$$\text{Wind Index} = 1.0 + \frac{\text{Wind Speed}}{5.0}$$

Thus, for a wind speed of 0 to 4.9 kilometers per hour maximum deviation between background and air temperature is permitted, while for speeds of 45 kilometers per hour the difference will be minimum.

Similar arguments are developed for each factor. Since cloud cover transmission (TCLOUD) varies from zero to one and is maximum for clear skies, the index on cloud cover (ICLD) is developed as:

$$\text{Cloud Index} = 8(\text{TCLOUD}) + 1.0$$

yielding integers varying from one to nine as TCLOUD varies from zero to one. For clear skies, TCLOUD = 1 and maximum difference between background and air temperature is allowed, CLOUDF = 1; while for heavy overcast, TCLOUD = 0 and the background takes on the same value as the air temperature, CLOUDF = 0.

For vegetation coverage, concern centers on transmission by the foliage of solar energy so that conditions of heavy forest cover, medium cover, or light or open areas are defined. Since IVCOV, the index on vegetation cover supplied from unit terrain (UNTER) ranges from one to five but with values three, four, and five describing relatively open areas, the index is reduced to range from one to three as the index for vegetation cover (IVEGCV) in this subroutine. The background and rain indices are developed in similar fashion.

It is known from limited field data that false alarms are experienced with PIRID type sensors in field use and in field test. These alarms are attributable to the fact that the power incident on a single detector is not constant but is continually fluctuating and hence for a prescribed threshold, a noise signal may be generated which will be identified as a target event. Unfortunately, the literature contains practically no information on this particular problem. As was the case with background temperature, an empirical expression for the standard deviation of background temperature fluctuations (SIGMA) was developed to provide the inputs to the false alarm process as follows:

```
SIGMA = 0.72 - (0.001465 X MTIME X WVAR X CVAR X VVAR
        X TVAR X RVAR)
Where   MTIME  =  Absolute value of (local time - 9)
        WVAR   =  Wind variance factor
        CVAR   =  Cloud variance factor
        VVAR   =  Vegetation variance factor
        TVAR   =  Temperature variance factor
        RVAR   =  Rain variance factor
```

The rationale for the expression developed is that, through it, a value of SIGMA is produced which when entered into the PIRBK subroutine and subsequently to FAINTV subroutine, produced false alarms events consistent with the very limited field data. Based on field data the mid-morning time centered on 9:00 AM is ascribed the maximum SIGMA, i.e., highest false alarm rate, but with SIGMA and false alarm rate modified by the five factors shown. The same factor indices described earlier are used to select appropriate variance factors from the stored data set.

The outputs of this subroutine, background temperature (TEMPEV) and background temperature fluctuation standard deviation (SIGMA), are supplied through the call statement to PIRBK subroutine for development of background power information. The descriptive flow of Figure 3.2-8 shows the simple sequential process of this subroutine.

3.2.5 Subroutine PIRBK

3.2.5.1 Purpose

The PIRID sensor is a hand-emplaced sensor operating as a passive intrusion detection sensor in the 7-15 micron region. Two fields of view and appropriate timing logic are included to provide for target detection. Noise, however, may also satisfy the logic requirements and hence false alarms events need also to be considered. It is the purpose of the PIRBK subroutine to develop the power incident on each detector due to the background and the average threshold crossing rate for noise (fluctuations in background radiance) making use of background temperature (TEMPEV) and

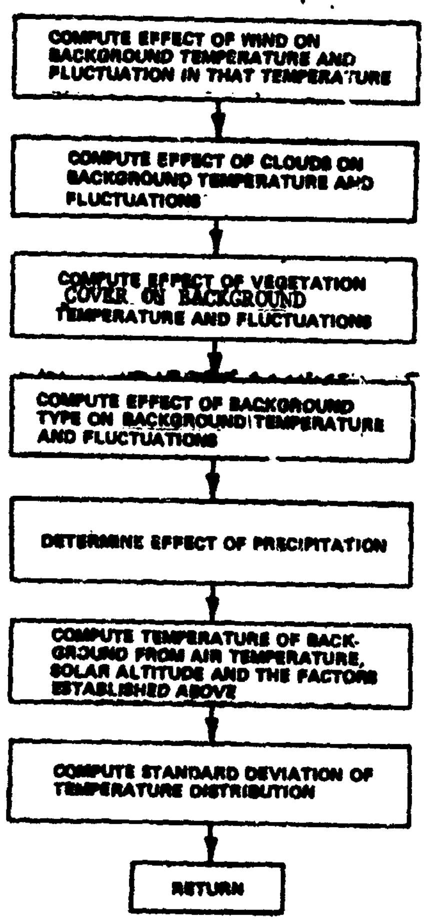


Figure 3.2-8 ENVIR SUBROUTINE MACROFLOW

fluctuation estimate (SIGMA) supplied by the ENVIR subroutine for the location, background type and environment supplied from planner input, unit terrain, and the atmospheric environment tables. The outputs, background power incident on sensor (WATTBK) and average threshold crossing (AVGTHC) are supplied to the PIRTG and FAINTV subroutines respectively.

3.2.5.2 Glossary of Inputs, Computed Values, and Outputs

Input Values

IUT	Index Unit Terrain
SIGMA	Standard Deviation of Background Temperature Fluctuation
TEMPEV	Temperature of Background Determined in the Environment Subroutine (C. Degrees)

Labelled Common Inputed Values

BWPIR	Band Width (Hz/Sec)
DEVXMN	Optical System Transmission Factor
DIAM	Diameter of Sensor (mm)
IPRINT	Output Data Device Designator = 6
LDUMP	True = Intermediate Calculations Printed, False = No Print
PHIAZ	Azimuth Angle in Radians
PHIEL	Elevation Angle in Radians
STEFK	Stefan-Boltzmann Constant/PI (1.805455E-8)

Computed Values

AREA	Area of Sensor Input Aperture in Square Meters
PIO4	PI/4 (0.785398)
PROBTH	Probability of Crossing Threshold
RADBAK	Background Radiance
TEMPKL	Background Temperature (Degrees Kelvin)

Output Values

AVGTHC	Average Threshold Crossing
EXPAN	Intermediate Calculations (Area * Field * DEVXMN)
FIELD	Field of View
WATTBK	Background Power Incident on Sensor

3.2.5.3 Description of Subroutine Logic and Processing

The PIRID sensor consists of two detectors and an optical system through which two fields of view are defined. These fields are separated from one another by an unobserved region to provide for two independent sequential indications of target passage. At all times except during target passage, power incident on the detectors will be due to radiance of the background as contained within each field and by radiance of the atmosphere which will be important under conditions of poor visibility. The power incident on both detectors for the non-target case need not be the same, since a nulling system is included in the sensor design. However, short-term fluctuations in received power (those that fall within the pass band of the target signal) cannot be eliminated and hence must be considered as potential sources of false alarms.

This device operates as a passive system in the 7-15 micron region. Each field of view will be filled by a composite radiating background consist'ng of trees, grass, brush, water and sky, depending on the emplacement location and the manner in which the device is emplaced. Background radiance will be due predominantly to the radiant emittance of the background constituents, radiating as black or gray bodies. Reflection of sunlight can be expected to be considerably less than the black body radiance as may be seen in Figure 3.2-9 but in some cases may be adequate to cause changes sufficient to produce false alarm problems.

The first problem treated in the PIRBK subroutine (see Figure 3.2-10) is computing the power incident on each detector. Here it is assumed that the powers will be the same for both fields, and since differences are removed by nulling, this assumption is considered to impose no serious limitation at this point. First, a call is made to the ENVIR subroutine to provide estimates of the background temperature, TEMPEV, and fluctuation in background temperature, SIGMA, for the appropriate location, location description (IUT) and environmental conditions (ATMENV). The background temperature is then employed to compute the radiance of the background by the expression:

$$\text{Background Radiance} = 0.27\ \sigma\ (\text{Background Temperature})^4$$

where σ is the Stefan-Boltzmann constant over PI (STEFK) and the background temperature (TEMPKL) is in degrees K. The factor 0.27 is included to account for the fact that for the temperatures of interest, approximately 27 percent of the radiance lies in the 7-15 micron region. This is obtained by integrating the Planck radiation law:

$$w = \int_{\lambda_1}^{\lambda_2} \frac{c_1 \lambda^{-5}}{e^{c_2/\lambda} - 1}\, d\lambda$$

over the wavelength region from 7-14 microns for $T = 300°$ Kelvin (see Reference 4).

While the actual percentage will deviate a small amount from 27 percent (25 to 29 percent) over the range of background temperatures to be included, the target temperature will in general be related to the environmental temperature and will differ from it by only of the order of 10° C so that actual integration of the Planck equation is not required.

It is further assumed that both fields of view are filled by black bodies radiating at the background temperature. Thus the power incidence on each detector can be computed from the data defining the sensor aperture and the angular dimensions of each field of view (assumed to be the same). The area of background subtended at a range, R, by the field of view would be:

$$\text{Background area} = (\text{Field of View})(R^2)$$

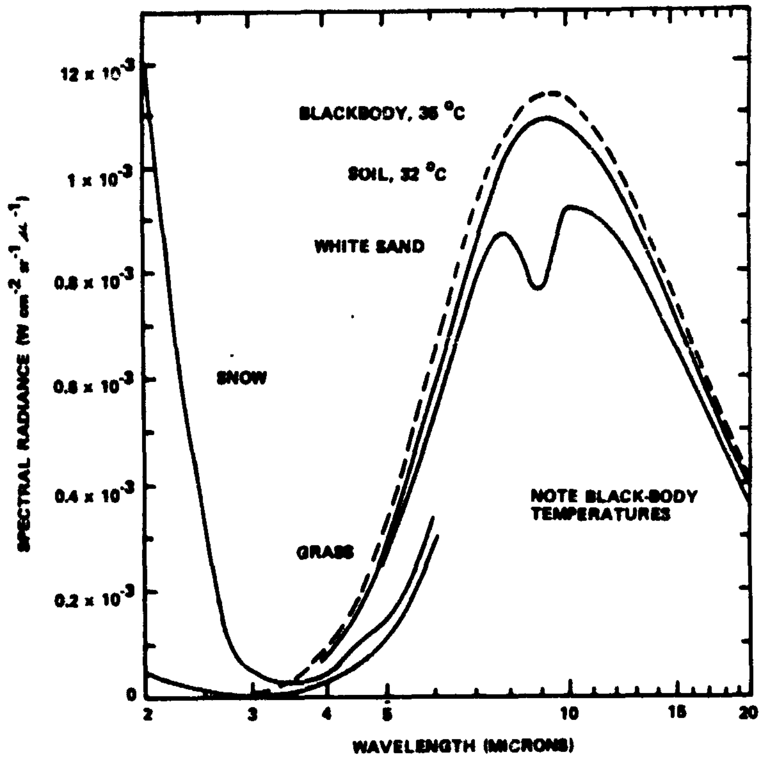


Figure 3.2-9 SPECTRAL RADIANCE OF TYPICAL TERRAIN MATERIALS AS OBSERVED DURING THE DAYTIME. (FROM REFERENCE 7)

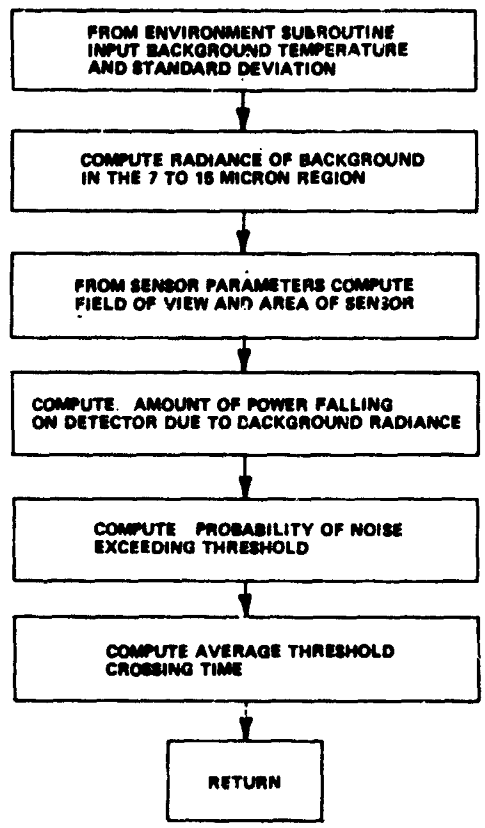


Figure 3.2-10 PIRBK SUBROUTINE MACROFLOW

that is, the area contributing to the power incident on the detector. The solid angle subtended by the sensor for background at range, R, is the ratio of the area of the sensor to range squared; this is:

$$\text{Angle subtended} = \frac{\text{(Sensor Area)}}{R^2}$$

Combining both expressions and including the transmission (7-15 microns) of the optical system, the geometrical factor reduces to:

(Field of view)(Area)(Transmission factor)

which is seen to be independent of range (hence range is not a required input for this set of computations). The effects of atmospheric attenuation on background power have been included in an indirect way, through the definition of background temperature (TEMPEV) as a function of background and environmental parameters in the ENVIR subroutine.

Because of the lack of data describing the fluctuations in background radiance and its spectral distribution, the detailed analysis of false alarm statistics originally planned had to be modified such that the average threshold crossing is an approximation designed to provide false alarm data consistent with the limited field data. The probability of crossing the threshold (PROBTH) is computed as follows:

$$\text{PROBTH} = \text{Complementary Error Function of } \frac{0.18}{\text{SIGMA}}$$

with the constant (0.18) included in place of Threshold/2, the value used in the standard form.

The Average Threshold Crossing (AVGTHC) is computed from the probability of crossing the threshold on each chance, the number of independent chances per second provided by the bandwidth and the factor 2 to account for the fact that the crossing may take place in either channel as follows:

$$\text{AVGTHC} = \frac{1}{2\text{(PROBTH)(BANDWIDTH)}}$$

The AVGTHC is then supplied to FAINTV for scheduling of false alarm events. The other outputs, FIELD, EXPAN and WATTBK are stored so that they may be supplied for the appropriate time interval for use in the PIRTG subroutine.

3.2.6 Subroutine ARFBK

3.2.6.1 Purpose

The ARFBUOY sensor consists of a number of button bomblets distributed over an area to be monitored and a transceiver located within 100 meters of the farthest limits of the array of bomblets. When an intrusion takes place, and if a bomblet is disturbed, a signal is generated by the bomblet which is received by the transceiver, coded and transmitted to a monitor. The ARFBK subroutine is included to determine, from planner inputs, the area density of bomblets deployed from which probabilities of excitation will be developed in the ARFTG subroutine of the MSM. In addition, an estimate of the average false alarm interval is developed from the number of bomblets deployed, the method of deployment, and the environmental conditions considered to be pertinent.

3.2.6.2 Glossary of Inputs, Computed Values, and Outputs

Input Values

DIMMAX	Maximum Dimension of Seeded Area (Rectangle Length or Circle Diameter)(Meters)
IEMPLC	Method of Emplacement, 1=Hand, 2=Artillery, 3=Air
IGEOM	=1 (Open Circle), =2(Open Line), =3(Road or Trail)
IMAG	Index on Bomblet Type (0=Magnetic, 1=Noiseless)
NBMBLT	Number of Bomblets Used
WIDTH	Width of Seeded Area (Meters)

Labelled Common Inputed Values

IPRINT	Output Data Device Designator = 6
LDUMP	True = Intermediate Calculations Printed, False = No Print
PRATE	Rain Fall Rate (mm/hr)

Computed Values

FARATE	False Alarm Rate
RAINF	Factor Giving Effect of Rain on False Alarm Rate
SAREA	Area of the NBB Array (Square Meters)
SNUMB	Number Within Area Containing NBB's and Target

Output Values

AREADN	Area Density of the NBB's (Square Meters)
AVFATM	Average False Alarm Interval

3.2.6.3 Description of Subroutine Logic and Processing

The deployment of an ARFBUOY may be extremely varied depending on the purposes for the emplacement, the type of terrain in which emplacement is made, vegetation density, and other considerations. Three basic types of geometry are allowed the planner in this simulation: a uniform distribution emplanted in open areas with either circular or rectangular limits, or a distribution along a road, also described by rectangular limits. In all cases, the planner must specify the type of configuration (IGEOM) and the number of emitters to be employed (NBMBLT).

This simulation is based on several additional assumptions. First, the emitters are uniformly distributed, retain the uniform distribution over the game, and are not lost to the game by exitation or other causes except for the transceiver, the reliability of which is treated in the subroutine designed specifically for reliability analysis. It is also assumed that the principal cause of false alarm is basically rain, with other sources being of such small import that they are neglected. It is also assumed that button bomblets of the magnetic type will eventually be available and that the false alarm rate for this class will be different from the shock excited noiseless units.

In the subroutine (see Figure 3.2-11) the area over which the bomblets are distributed is computed. From that value and the planner input of number of bomblets, the area density (AREADN) is determined for use in the ARFTG subroutine. Then, based on the rainfall data and the type of bomblet deployed, the false alarm rate (FARATE) and average false alarm time interval (AVFATM) are computed.

False alarm rate is considered to be related logarithmically to rain rate by the expression:

$$\text{False Alarm Rate Rain Factor (RAINF)} = \log_{10} (\text{rainfall rate} + 1.0)$$

This expression results from an intuitive argument and is not supported by data at this time. A constant is also included in the false alarm rate calculations to account for other disturbance sources such as animals, birds and wind since lack of data prevented any different treatment for this equipment.

The noiseless bomblets are considered to have a false alarm rate (FARATE) given by:

$$\text{FARATE} = \frac{(\text{RAINF} + 0.5)(\text{No. Bomblets in Area})}{0.36 \times 10^6}$$

where FARATE is given in events per second.

The magnetic bomblets are considered to have a lower basic false alarm rate given by:

$$\text{FARATE} = \frac{(\text{RAINF} + 0.1)(\text{No. Bomblets in Area})}{0.36 \times 10^6}$$

except that for air-dropped units, the rain effect factor is increased by a factor of 2. The average false alarm interval (AVFATM = 1/FARATE) is supplied to the FAINTV subroutine for false alarm event scheduling and area density of bomblets (AREADN) is supplied to ARFTG subroutine in MSM.

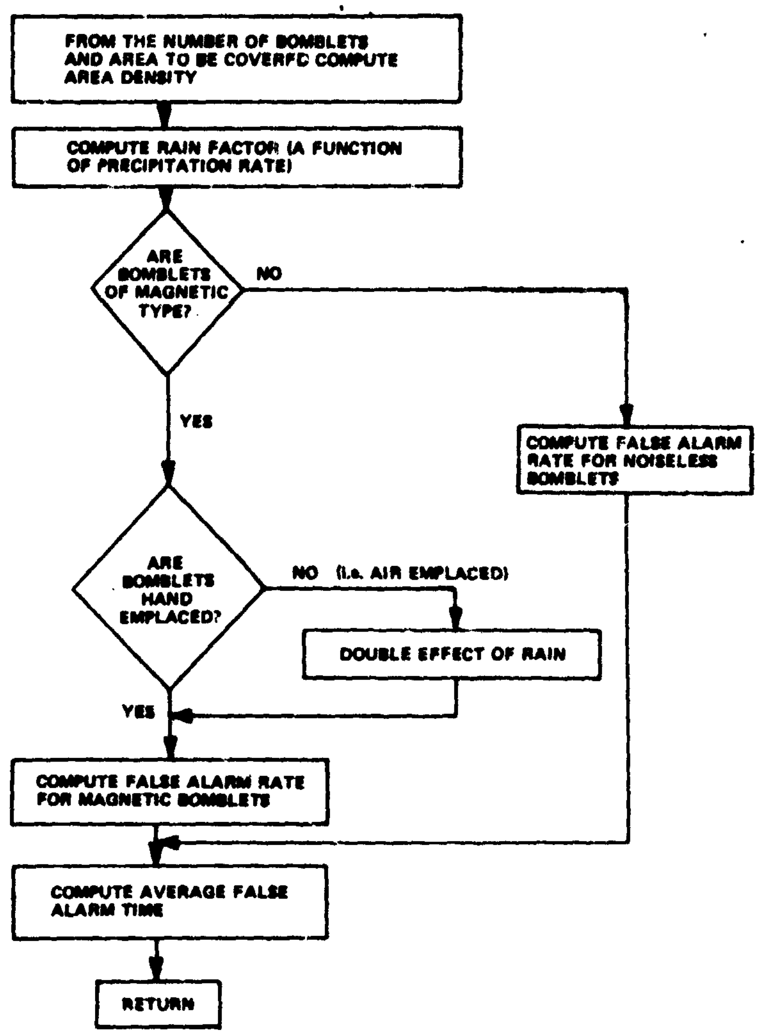


Figure 3.2-11 ARFBK SUBROUTINE MACROFLOW

3.2.7 Subroutine BWIRBK

3.2.7.1 Purpose

The breakwire device of which the AN/GSS-9 equipment is an example provides an alarm when a thin wire is broken. Since the wire may be broken by natural causes such as by wind driven brush, animals, birds, falling branches and others in addition to target intrusion events, there is a need to compute the probability of time of occurrence of the non-intrusion or false alarm event. This is the purpose for the BWIRBK routine.

3.2.7.2 Glossary of Inputs, Computed Values, and Outputs

Input Values

ALENGT	Length of Line Deployed (Yards)
IUT	Index on Unit Terrain

Internally Stored Designer Input Values

DLENGT	Length of Line Available (Yards) Set to 2500
FNCOMP	Fauna Component Factor
WCOMP	Wind Component Factor

Labelled Common Inputed Values

IPRINT	Output Data Device Designator = 6
ITOD	Time of Day
LDUMP	True = Intermediate Calculations Printed, False = No Print
WSPEED	Wind Speed (km/hr)

Computed Values

DUM	Dummy Argument
FAFACT	Dummy Argument
KFAUN	Index in Animal Activity (1=6 AM-6 PM, 2=6 PM-6 AM)
WFACT	Wind Component Factor for Particular Vegetation Type

Output Values

AVGTHC	Average Threshold Crossing

3.2.7.3 Description of Subroutine Logic and Processing

The two major causes for accidental activation of the breakwire device are considered to be wind and fauna (see Figure 3.2-12). The wind will be effective in those cases in which brush and ground cover are available to be driven by the wind against the device as well as by wind caused movement of those natural objects to which the breakwire is anchored, such as trees. The factor, WFACT, associated with the wind is given for a particular vegetation type by the relation:

WFACT = (Wind Component Factor)(Wind Speed)/100

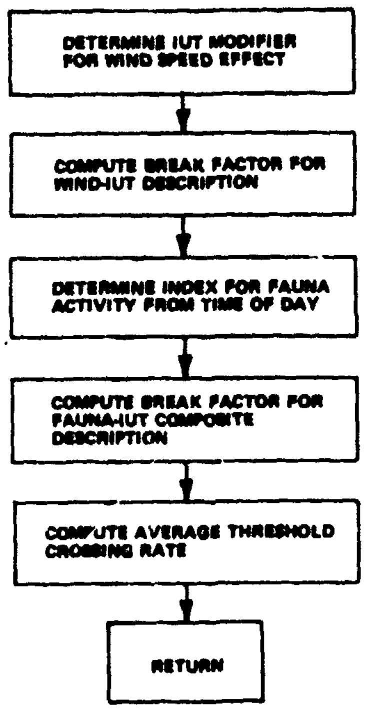


Figure 3.2-12 BWIRBK SUBROUTINE MACROFLOW

that is, a linear function of wind speed. The wind component factor (WCOMP) is derived from a stored data set in this subroutine which is keyed to the index of unit terrain (IUT) descriptions. Based on the descriptions of vegetation types used in the unit terrain (UNTER) tables, the following modifiers were assigned to WCOMP and to a similar factor for the fauna component (FNCOMP).

Terrain Index (IUT)	Terrain Descriptions	WCOMP	FNCOMP
1	Rice Paddy	0.2	1.0
2	Single Canopy - Light Undergrowth	0.7	1.0
3	Brushwoods - Coffee and Tea Plantations, Rolling Hills	1.0	1.0
4	Brushwoods - Coffee and Tea Plantations, Flat Valleys	1.0	1.0
5	Multicanopied Dense Undergrowth Forest, Upper Slopes	0.5	0.6
6	Multicanopied Dense Undergrowth Forest, Lower Slopes	0.5	0.6
7	Single Canopy Light Undergrowth Forest with Bamboo	0.1	0.3
8	Dune Grass and Casuarina on Sand	0.2	1.0
9	To Be Defined	0.0	0.0
10	To Be Defined	0.0	0.0

These values assigned are considered to be preliminary, requiring further validation or modification.

As noted above, the wire may also be disturbed by fauna. Animal and bird activity are assumed to be most extensive during the hours 6 AM to 6 PM. The fauna index (KFAUN) is computed for the time of day and may take on the values 0 (0000 to 0600), 1 (0600-1800), and 2 (1800-2400). The range of KFAUN, however, is limited to 1 and 2 by setting KFAUN = 2 whenever the computation result is 0. The fauna factor (FAFACT) is computed as inversely proportional to KFAUN, thus making the effect of time apparent. In addition to time, an estimate for the concentration of fauna and hence the total activity level is used. These estimates are included by use of the fauna component numbers keyed to terrain description for a sensor's location as shown in the table above. FAFACT is calculated as directly proportional to these estimates.

The average threshold crossing time (AVGTHC) or average time for accidental breaks of the wire is derived as inversely proportional to the sum of the wind and fauna factors and also is modified by the proportion of the wire actually deployed (ALENGT) to the total wire available for the AN/GSS-9 (2500 yards), upon which the break factors were based. AVGTHC is supplied to the false alarm subroutine for scheduling of breakwire false alarm events.

$$\text{AVGTHC} = \frac{\text{Numerator}}{((\text{Wind factor} + \text{fauna factor}) \times \text{ALENGT})}$$

3.3 TARGET DETECTION SUBROUTINE

3.3.1 Subroutine SEISTG

3.3.1.1 Purpose

This subroutine is employed to determine target detection events for seismic sensors. It is called by appropriate executive subroutines in the MSM at event times when a target-seismic sensor interaction exists and some probability of detection is possible.

3.3.1.2 Glossary of Inputs, Computed Values and Outputs

Input Values

ALPHA	Propagation Factor
ALPHAB	Propagation Factor
CUPCOF	Acoustic to Seismic Coupling Coefficient
GEQUIL	Gain for an Equilibrium Noise - Only Situation
HTAC	Height of Aircraft (Meters)
HTFOL	Height of Foliage (Meters)
HTMUN	Height of Detonation of Munitions (Meters)
IACTYP	Aircraft Index Selector, 1= Helicopter, 2 =Propeller, 3=Jet
IBOAT	Boat Index Selector, 1=Raft or Sampan, 2= Outboard, 3= Patrol Boat
IDSNSR	Sensor ID
IDTGT	Target Number
IFIXGN	=0 No Fixed Gain, =1,2,3,4,5 Planner Set Gain, =6 Selected Gain
IMAN	Man Index Selector, 1=Small Man, 2=Large, 3=Large Animal
IMNTYP	Munition Index Selector, 1=Small, 2=Medium, 3=Large
ITGTTP	Target Type
IUT	Index on Unit Terrain
IVEGDN	Index for Foliage Density
IVEH	Vehicle Index Selector, 1=Jeep, 2=Truck, 3=Tank, 4=Train
MFORM	Index Type of Formation (Troops), 1=S.F., 2=D.F., 3=Open
NMSURF	Index for Men Noise Modifier
NOELEM	Number of Elements in a Target
NOTGTS	Total Number of Targets
NTAR	Dummy Array
NVSURF	Index for Vehicle Noise Modifier
RA	Range to Target From Sensor (Aircraft)
RM	Range to Target From Sensor (Munitions)

Input Values (continued)

RT	Range to Target from Sensor (Troops)
RV	Range to Target from Sensor (Vehicles)
RW	Range to Target from Sensor (Boats)
TARNOT	Number of Targets in Group for Troops
TARNOV	Number of Targets in Group for Vehicles
THRESH	Threshold (Volts)
TSPACE	Space Between Targets in Group for Troops
VNOISE	Background Noise Voltages
VSPACE	Space Between Targets in Group for Vehicles
XSENS	X Sensor Position
XTGT	X Target Position
YSENS	Y Sensor Position
YTGT	Y Target Position

Labelled Common Inputed Values

ATEMP	Ambient Air Temperature
CHUL	Upper Limit, Canopy or Vegetation
CONSTS	Average Amplifier Output
H2ODEN	Grams/CC of Water in the Air
IPRINT	Output Data Device Designator = 6
ITIME	Game Running Time
ITTAB	Array for Target Parameter List
IVCOV	Index for Vegetation Cover
LDUMP	True = Intermediate Calculations Printed, False = No Print
TDEL2S	Time Delay Times 2

Internally Stored Designer Input Values ***

ACTAR	Aircraft Noise by Type (105.0, 115.0, 125.0)**
ARTTAR	Value for Munition by Type (dB) (-18.0, -12.0, -6.0)
DBBOAT	Noise Value for Boats (dB) (-54.0, -51.0, -48.0)
DBMAN	Input Value for Man Target (dB) (-66.0, -60.0, -57.0)*
DBSURF	Target Noise Modifier due to Surface for Vehicle (dB)
DBVEH	Input Value for Vehicle Target (-48.0, -45.0, -42.0, -39.0)*
FOLATN	Foliage Acoustic Attenuation Factor (0.15, 0.2, 0.05)*
TARORG	Seismic Source Strength modifier (dB) based on target formation value - Formation, 1=0 (Single File), 2=6(Double), 3=-6(OP)
TMM	Target Noise Modifier due to Surface for Man (dB)

Computed Values

ATMAIN	Atmospheric Attenuation
DALPHA	Random Number
DBATAR	Aircraft Signal at Sensor
DBMTAR	Munition Signal at Sensor
DBSENS	Target Signal Level at Detector
DBVTAR	Vehicle Signal at Sensor
DELRV	Distance Traveled by Lead Vehicle Between First and Second Entry

* Ref 1,2,)
** Ref 2,16,17) Pg 3-126

*** Designer input values are given in the Program Listing.

Computed Values (continued)

DELTAT	Difference Between Beginning of Game and Time of Last Entry
FN	Total Number of Increments of Time After Passing Point of Closest Approach
GAIN	Amplifier Gain
GINLST	Gain Value at Last Entry
GINPUT	Nominal Gain Detected from Noise + Target or Noise Only Environment
ISENPR	Previous Sensor ID
IT	Dummy Index
ITARPR	Previous Target ID
ITIMLE	Time of Last Entry to Subroutine
ITIMLR	Time of Latest Report
KI	Dummy Index
LFIRST	Logical Indicator
LSEC	Logical Indicator
OUTMAX	Value of Signal for Threshold Computation
RAIR	Range in Air (Meters)
RANGE	Range to Target from Sensor
RFOL	Range Through Foliage (Meters)
RHO	Intermediate Calculation
RHYPOT	Indicator for Arrival of Last Vehicle at Range of Closest Approach
RVREF	Range to Lead Vehicle on Previous Look for Approach Target
SIGVAR	Random Number
TARLEN	Target Length Squared
TEMAMK	Ambient Air Temperature (Kelvin)
YFL	Dummy Argument
VSENSQ	Sum of Voltages Squared

Output Values

LDET	Detection Decision

3.3.1.3 Description of Subroutine Logic and Processing

Seismic sensors are a class of the unattended ground sensors employed in the surveillance or target acquisition mission. Being emplaced near or on the earth's surface they are responsive principally to Rayleigh waves travelling through the earth along the surface.* These waves are generated by a variety of sources some of which were considered previously in the SEISBK subroutine, namely, those sources which are quasi-continuous leading to a background noise level. In addition, signals attributable to a set of objects referred to as targets and including men and animals, vehicles, aircraft, munitions and boats which are of a transient nature are also observed. If the signals due to these latter sources are of adequate strength when referenced to the noise level or threshold for a particular logic processing, they can be detected as targets. Thus the objective of this subroutine is the determination of detection events through analysis of target signal strengths.

* Ref 1,2, Pg 3-126

The simulation of the sensor is carried out in the following way (see Fig. 3.3-1). The transducer is assumed to respond to ground particle velocities at the transducer over the frequency range of 5 to 100 Hz, but with bandwidth as a design parameter that may be specified. The sensitivity of the transducer is not considered specifically, the assumption being made that the transducers will in all cases be responsive to the microseismic background level which is taken to be -120 dB relative to 0 dB for one centimeter per second. Secondly, the amplifier employed with the transducer will have a gain capability sufficient to cause the amplified RMS noise level to achieve an average amplifier output (CONSTS) for AGC systems, that is, a constant output reference level. Thus, differing sensitivites would require differing gains in the ratio of Gain = CONSTS/(microseismic noise)(sensitivity) so that reduced sensitivity would only require increased gain. A third assumption is made that the amplifier thermal noise output is less than that due to the microseismic background level so that gain is always a function of input noise level.

A threshold is defined for target detection as a voltage level above CONSTS, the reference average amplifier output level, where the offset is given as BIASSE so that threshold is determined as:

Threshold = BIASSE + CONSTS

where BIASSE is a design parameter which may be varied as is the parameter CONSTS. In this simulation, CONSTS has been arbitrarily set to the value of 1.0 volt.

It is also noted that in the sensor logic requirements are imposed on the number of threshold crossings per timing interval such as four events in six seconds, or on the integrated output signal level, where integration is over some period of time, for example, six seconds. These criteria are applied for noise. For targets, the fact that a signal exceeds threshold on a specific entry to the subroutine is adequate. This compromise is required in order to limit the amount of computer time required.

Detection logic may also impose some dead time in sensor reporting, for example, a period of ten seconds following a report in which no additional reports may be issued. Such logic is included to limit the battery drain in field equipments and is included as part of the seismic subroutine.

In addition to the above, computations for the signal levels at the sensor are required for completion of the detection assessment. The SEISTG subroutine is called at time intervals of five seconds of game time typically whenever a target or targets are within a maximum range specified for each target type. The actual determination of target position with respect to the sensor and maximum range checks are performed in PRERUN subroutines with range and target description being input to this subroutine. The target source strengths with reference to seismic wave generation are specific to the seismic sensor only and are, therefore, contained in a set of data statements in the SEISTG subroutine. Similarly, surface effects, effects of target organisation, attenuation factors and the like are also specific and are, therefore, contained within the subroutine as may be seen by examination of the program listing (Volume III).

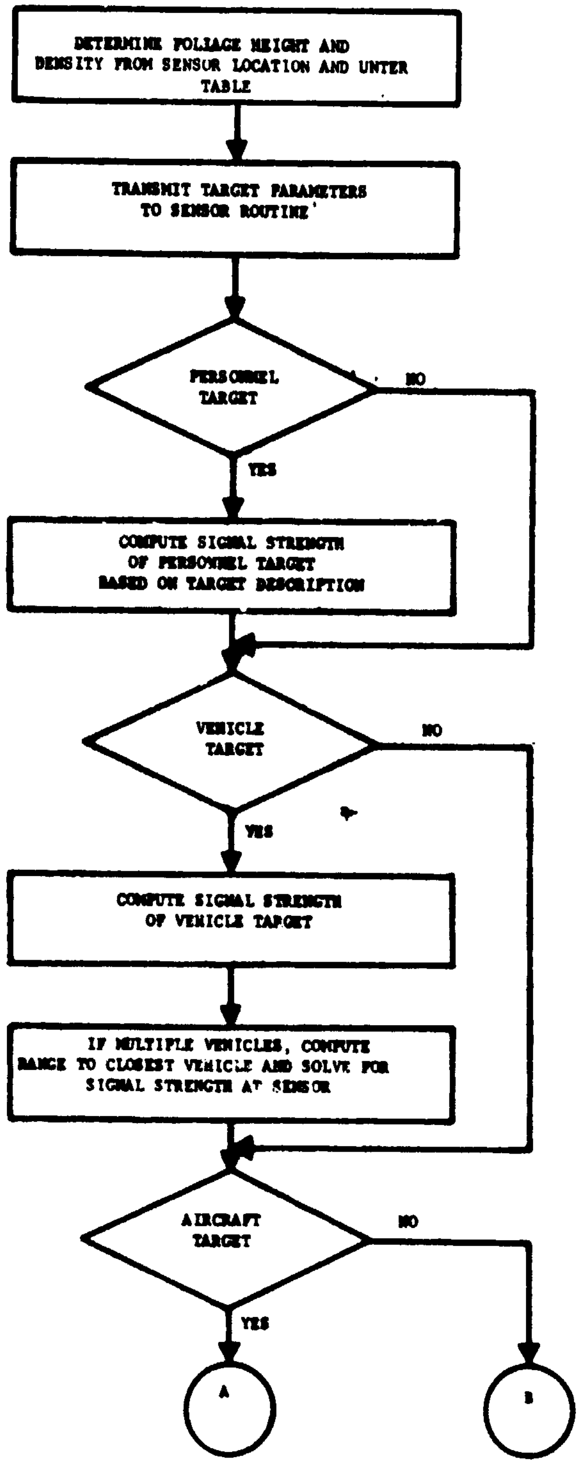


FIGURE 3.3-1 SEISTC MACROFLOW
(Sheet 1 of 3)

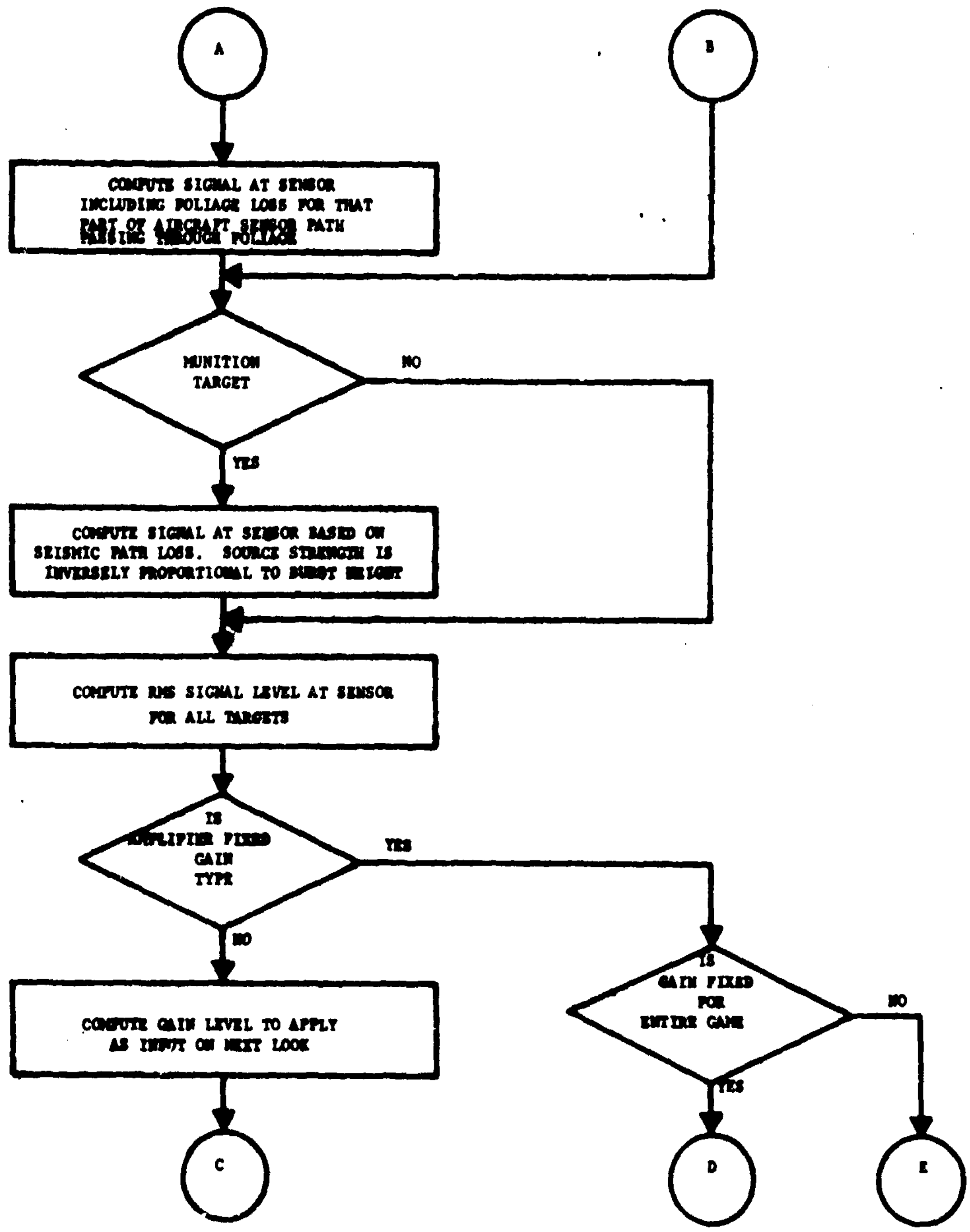


FIGURE 3.3-1 SEISTG MACROFLOW
(Sheet 2 of 3)

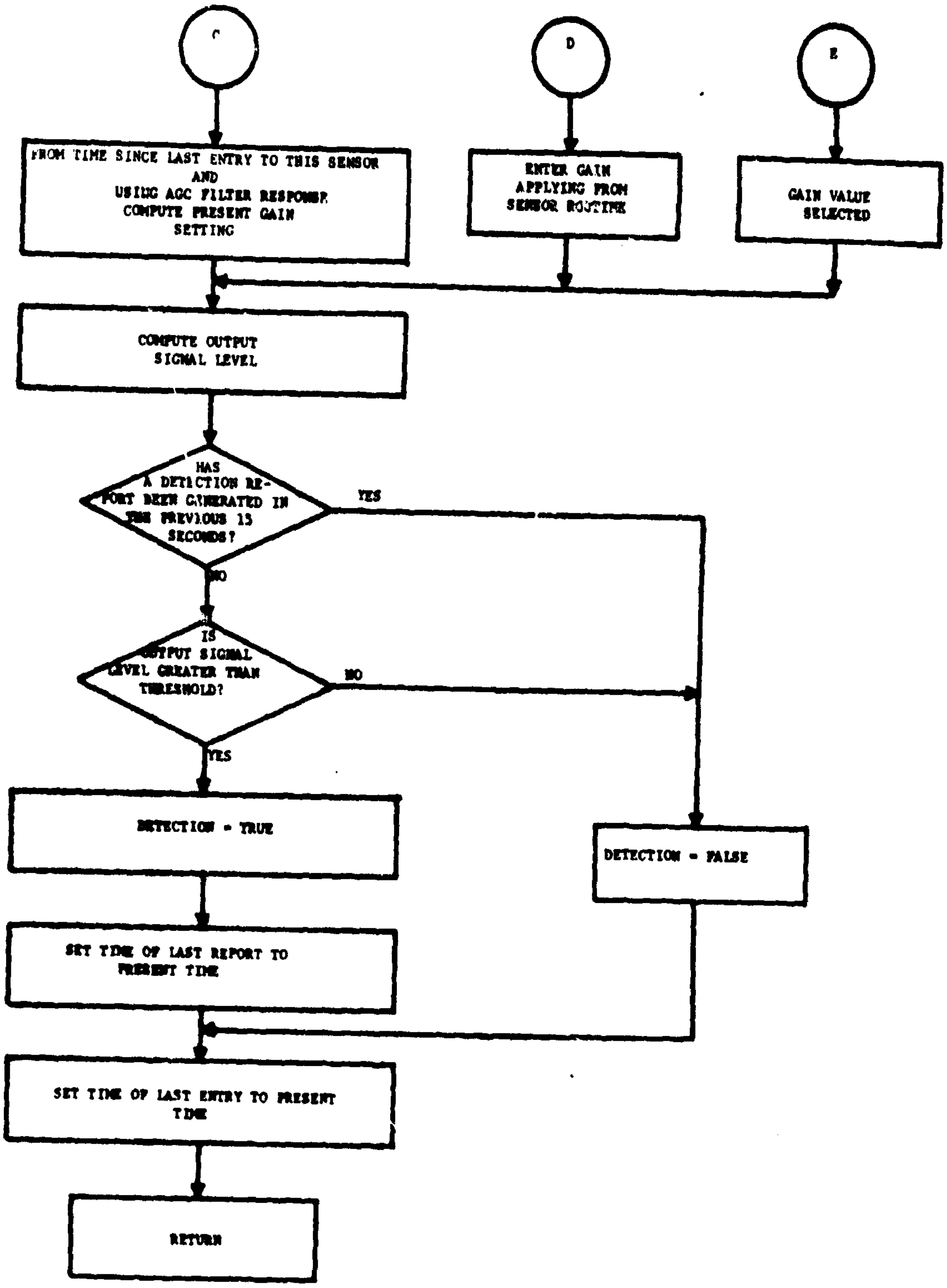


FIGURE 3.3-1 SEISTC MACROFLOW
(Sheet 3 of 3)

On entering the subroutine, VSENSQ, the square of the output signal level is initialized to zero. Vegetation factors, temperature, and other required values are inputed from the SENSOR, ATMENV, and UNTER common areas. Entry is then made into the target signal analysis loop wherein the loop is entered once for each target within range of the sensor. The index, IT, is developed and used in the CALL TRNSFR statement to introduce individual target descriptions. After computing the range to a specific target, a branch is made to the appropriate signal analysis segment of the subroutine based on target type (troops, vehicles, aircraft, munitions, boats).

If the target is of foot troop type (ITGTTP=1), an argument describing the source strength at the target is employed in which the target is considered as a single element even though a number of elements may be included. The single element source strength, derived from stored data, is keyed to a particular target description and is modified by the number of elements in the target, by element spacing, by formation type of the target and by surface conditions. A multielement target location is taken as the coordinates of the leading element and a single speed is used. Consideration should be given during later simulation activity to the treatment of such troop targets as a column such as is done for vehicles (described later in this section). The multielement troop target source strength (DBMTAR) is computed as follows:

$$\text{DBMTAR} = \text{DBMTAR} \quad + 20\left[\text{Log}_{10}(\text{No. Targets in Group}) - \log_{10}\left(\frac{\text{Space Between Targets}}{3}\right)\right] + \text{Formation Type Modifier} + \text{Surface Noise Modifier}$$

A path loss between target and sensor proportional to R^n is employed. Literature searches have shown that in general attenuation is proportional to

$$R^{-.5}\, e^{-\alpha R}$$

where α is a function of frequency and of the propagating medium. Because of the very limited descriptions of target, target-medium interaction leading to seismic wave generation, and of the dispersive characteristics of the soils and geological structure to be encountered, this expression cannot be employed in this simulation. However, as a result of extensive field tests in Southeast Asia, it has been established that a simplified expression of the form of R^{-n} where n varies from 1.6 to 2.6 applies (see Figure 3.3-2.) In this simulation,n is given a value of 2.2 in the product of(20)(propagation factor) in the development of the target signal at the sensor (DBSENS). In addition, the path loss is found to vary from position to position about the sensor for positions at the same range. To introduce this variability, a random component (DALPHA) which may vary from -0.1 to + 0.1 is added to the propagation factor (ALPHA) before the multiplication. Future effort should consider relating the field observed values of n to soil types as defined in the terrain table.

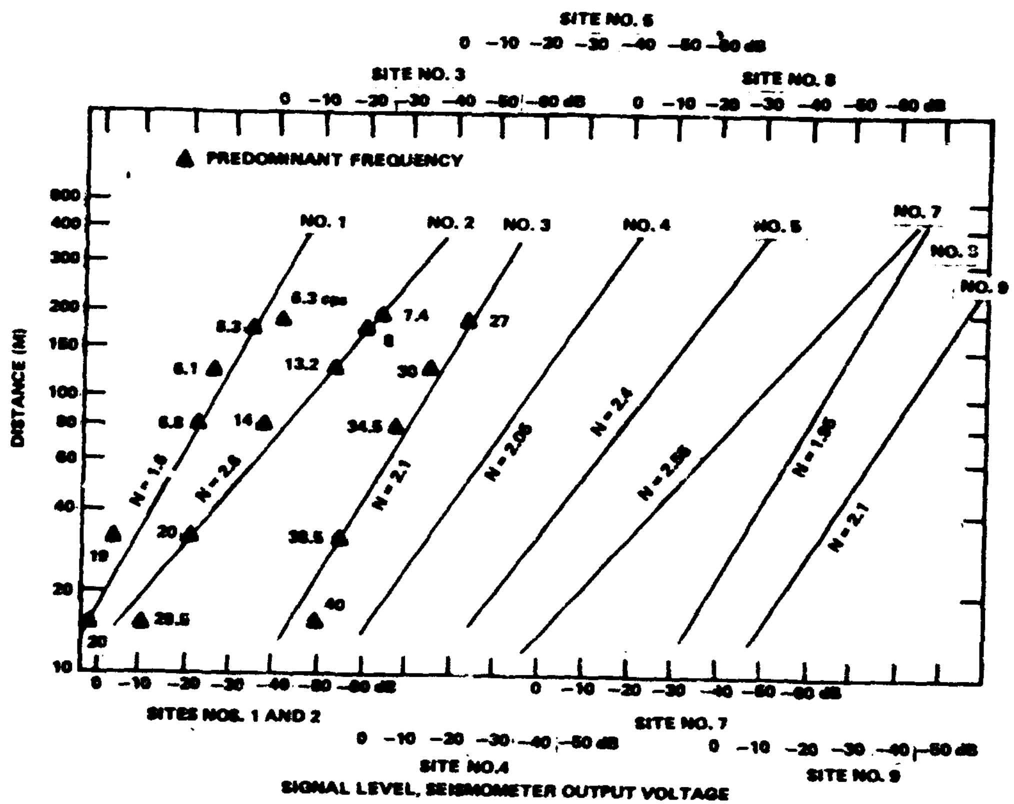


Figure 3.3-2 AVERAGE SURFACE WAVE ATTENUATION FOR EIGHT SITES (FROM REFERENCE 1)

Having determined the source strength at target (DBMTAR) and the attenuation coefficient (ALPHA + DALPHA) the signal level at the sensor is determined from the relation

$$DBSENS = DBMTAR - (ALPHA + DALPHA) \left[20 \log_{10} (\text{Target Range} + 1.0)\right]$$

This signal is converted to voltage, squared, and then added with any other Signals to form the sum of the squares of the target signals from which the RMS value will be obtained. Return is then made to the beginning of the target loop where the target index is modified to provide the proper index for the next set of target descriptors to be introduced.

If the target is of vehicle type (ITGTTP=2) computations for path loss quite similar to that described above are carried out. However, multielement targets (NOELEM > 1) are treated in a different manner than that described above for troops. Multivehicle targets may cover an extensive length of path or road so that treating the lead or centroid of the column would not be satisfactory. Rather the vehicle closest to the sensor is treated as the major signal source at a specific time. Thus for a group of vehicle targets the lead element of which is approaching the sensor, signal level is determined by the source strength of the first vehicle and its range from the sensor. After the lead element passes the point of closest approach but before the trailing vehicle reaches that point, the range employed is that to the point of closest approach. After the trailing vehicle has passed the point of closest approach, range to the trailing vehicle is employed. Since the range given as input is that to the lead vehicle, computations internal to the sensor subroutine must be employed to determine where the lead and trailing elements are with respect to the sensor. The signal strength at the sensor (using same formulation as for troop targets) is again converted to voltage, squared and added to the sum of target signal squares before return is made to the beginning of the target loop.

Aircraft targets are treated next. The values of aircraft source strength were taken from Reference 2 and are given as acoustic source strengths in the data statements. The acoustic wave generated by the aircraft is assumed to be propagated through the air to the sensor location at which location the acoustic waves give rise to seismic waves through acoustic to seismic coupling. (A discussion of this coupling process for which the above model is a rough approximation is contained in Reference 2.) The conversion factor is given as a coupling coefficient which relates dB in the acoustic system (relative to 5×10^{-6} cm/sec) to the reference system for seismic signals 0 dB = 1 cm/sec).

The sensor may be emplaced in a region of heavy foliage determined from canopy or vegetation upper limit from unit terrain description of sensor location so that foliage attenuation of the acoustic signal must be included. This factor is introduced into the model as a simple geometrical factor through similar triangle relations in the expressions:

$$\text{Range in air} = (\text{Range to aircraft from sensor}) \cdot \left(\frac{Hgt_{a/c} - Hgt_{fol}}{Hgt_{a/c}} \right)$$

and Range through foliage = Range to aircraft from sensor - Range in air

It is suspected that refraction problems are also present in the air-foliage acoustic path so the estimates developed in this simulation are quite preliminary and further examination of the acoustic propagation process should be made.

Atmospheric attenuation (ATMATN) as a function of water content and temperature was developed from a set of data given in Reference 1. Additional consideration might be given to aircraft altitude in the expression used because of the relation between air density and altitude. However, because of the expected low altitude of target aircraft, inclusion of altitude was not considered necessary.

Munition targets present a complex problem for seismic signal generation and very little information regarding such actual weapon and munition effects is available.* In this model, the seismic signal, generated by acoustic to seismic coupling at the source, (the weapon munition functioning location) is considered to be the major component of seismic energy at the sensor for a munition source. The source signal is attenuated in the same manner as for the walking target except the randomness (SIGVAR) is imposed on the source strength rather than on the path loss. For a munition bursting at some height above ground (HTMUN) the approximation is made that signal strength is reduced by the amount $6.0 \log_{10}$ (HTMUN + 1); that is, that signal strength is related logarithmically to burst height.

Boats are also considered to be targets for a seismic sensor and treated in the same manner as walking targets. In practice the pressure waves generated and propagated in the water couple with the earth to produce seismic waves. The overall attenuation law is taken to be R^{-n} where n is nominally 2.2 but with some random variation included.

*Air and surface changes have been employed in geophysical research, e.g., Reference 3, P.N.S. O'Brien, "The Efficient Use of Large Changes."

Seismic sensors of several classes may be employed and a distinction must be made between fixed gain and automatic gain control (AGC) types. For the AGC types, the long time average output is maintained constant by modifying amplifier gain in an appropriate manner. Since time constants of 20 seconds are usually employed and since targets will often be within range of the sensor for that period of time, target effects on AGC must also be included. To determine the level to which the AGC system has been driven since the last time the gains for a particular sensor were examined requires the use of several estimates of gain, namely:

GAIN-The value of amplifier gain to be used in computing outputs signal and to be stored for reference on the next entry to this subroutine.

GEQUIL-The gain value which would apply if equilibrium background noise only were effective, i.e., no target.

GINPUT-The value of gain which would be effective if present RMS value of signal and noise remained effective for a long period of time.

GINLST-The value of GINPUT computed on the previous entry to this subroutine for the sensor in question.

These are combined in the expression:

$$GAIN = \frac{1.0}{1.0+\rho}\left[(GINLST + GINPUT)\,\rho + (1-\rho)\,GAIN\right]$$

where

$$\rho = \frac{\text{Game Time - Time of Last Entry}}{\text{2(Sensor Time Delay)}}$$

The response predicted by this expression is illustrated in Figure 3.3-3

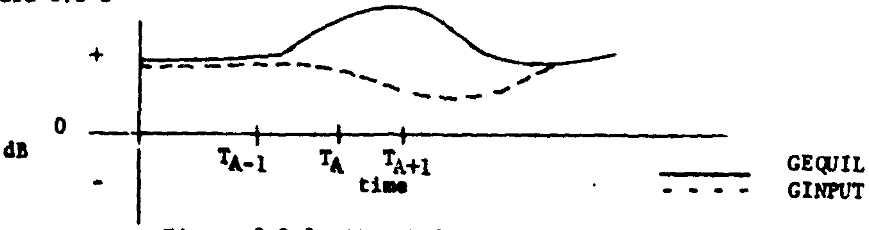


Figure 3.3-3 GAIN CONTROL TIME RESPONSE

Initially only noise is present. At time T_A, however, GEQUIL will be greater than GINPUT because the signal level input at the sensor (signal + noise) has increased. At time T_{A-1} noise only was present so GAIN and GINLST as read are equal to GEQUIL. If we assume 2 (sensor time delay) is 40 seconds and $T_A - T_{A-1}$ is four seconds then $\rho = 0.1$ and the computation specifies GAIN to be:

$$GAIN = \frac{1}{1.1} [(GEQUIL_{t_{a-1}} + GINPUT)(0.1) + (1 - 0.1)(GEQUIL)]_{t_{a-1}}$$

$$GAIN = \frac{1}{1.1} [(GEQUIL)_{t_{a-1}} + (GINPUT)(0.1)]$$

since GINPUT is less than GEQUIL, GAIN will be also less than GEQUIL. (NOTE: If GINPUT had been equal to GEQUIL the result, GAIN = GEQUIL. would have been obtained.) If, for example, a munition signal had been processed and GINPUT was found to be 0.1 (GEQUIL) then GAIN would have been computed as:

$$GAIN = \frac{1}{1.1} [(GEQUIL)_{t_{a-1}} + 0.1(GEQUIL)_{t_{a-1}}]0.1 + (1 - 0.1)GEQUIL)_{t_{a-1}}$$

$$= \frac{1}{1.1} (GEQUIL)_{t_{a-1}} + 0.01(GEQUIL)_{t_{a-1}}$$

$$= \frac{1.01}{1.1} GEQUIL_{t_{a-1}}$$

For later times of entry the same argument follows but the computation is somewhat more complicated because the value of GAIN employed is a function of prior values of GINLST and GAIN.

If the time since last entry is greater than twice the filter time constant, GAIN will be based only on GEQUIL and the current value of GINPUT where the latter is weighted by ρ to account for the fact that the target must have produced some small effect on GAIN if GINPUT is less than GEQUIL as seen in the expression:

$$GAIN = \frac{1}{1 + \rho} [\rho(GINPUT) + GEQUIL]$$

In the event that fixed gain is employed, a number of options are open as have already been discussed in the SEISBK discussion (Section 3.2.2). In this subroutine, the value of GAIN associated with the sensor with index IFIXGN in input through the common area labelled SENSOR. Then the value of GAIN is used to compute output signal level (OUTMAX) in the expression:

$$OUTMAX = \sqrt{(VSENSQ)(GAIN)}$$

Sensor characteristics such as a ten-second inhibit period following each message have already been discussed. It is assumed that this particular characteristic applies to all sensors. If the time since last report is less than 15 seconds (ten second inhibit + five second detection period), a report cannot be generated so the detection decision is declared to be false in all cases. If the time interval is 15 seconds or more and the signal exceeds the threshold, a logical TRUE is generated and ITIMLR, the time of last report, is updated to ITIME, the present time, for future use. Then finally time of last entry, ITIMLE, is also set to ITIME for use by the GAIN computation on the next entry. This latter statement applies for all entries, whether detection is TRUE or FALSE. Control then is passed back to the calling subroutine.

3.3.2 Subroutine ACOUTG

3.3.2.1 Purpose

This subroutine is provided to determine detection events for acoustic type sensors for troop, vehicular, aircraft, artillery, and river craft type targets.

3.3.2.2 Glossary of Inputs, Computed Values, and Output

<u>Input Values</u>

BOATNO	Number of Boats
BSPACE	Spacing Between Boats
GAIN	Amplifier Gain (Units)
GEQUIL	Gain for an Equilibrium Noise - Only Situation
GINLST	Gain Value at Last Entry
HGTAC	Height of Aircraft (Meters)
HGTFOL	Thickness of Foliage (Meters)
IAA	Index for Aircraft Noise 1=105(Helicopter), 2=115(Propeller), 3=125(Jet)
IDSNSR	Sensor ID
IDTGT	Target ID
ITGTTP	Target Type
ITIMLE	Time of Last Entry to Subroutine
ITIMLR	Time of Latest Report
IUT	Index on Unit Terrain
KMAN	Index for Man Noise, 1=20(Silent), 2=50(Talk), 3=65 (Animal)
KMUN	Index on Munitions Noise, 1=Small(100), 2=Medium(130), 3=Large(150)
KWATR	Index for Boat Noise, 1=Motor Sampan 82, 2=Patrol 88, 3=94
KVEH	Index for Vehicle Type, 1=Jeep(85), 2=Medium Truck(107), 3=Tank(113)
MFORM	Index for Type of Formation for Troops, 1=Open, 2=Single File, 3=Double File.
MFORM	Index for Type of Formation for Boats (Single File only)
NMSURF	Index for Noise Modifier Man, 1= -3(Open), 2=0(Hard), 3=3(Grav.)
NOELEM	Number of Elements in the Target

Input Values (continued)

NOTGTS	Total Number of Targets
NTAR	Dummy Array
NVSURF	Index for Noise Modifier, Vehicles, 1=Field (-3), 2=Hard (0), 3=Grav (6)
RA	Range from Aircraft to Sensor (Meters)
RM	Range from Man to Sensor (Meters)
RV	Range from Vehicle to Sensor (Meters)
RU	Range from Munitions to Sensor (Meters)
RW	Range from Water-borne Target to Sensor (Meters)
TARNOT	Number of Men
TARNOV	Number of Targets in Group for Vehicle
THRESH	Threshold (Amplifier)(Volts)
TSPACE	Spacing Between Men (Meters)
XSENS	X Sensor Position
XTGT	X Target Position
YSENS	Y Sensor Position
YTGT	Y Target Position
VNOISE	Total Background Noise (Volts)
VSPACE	Spacing Between Vehicles

Labelled Common Inputed Values

ATEMP	Ambient Air Temperature
CHUL	Upper Limit, Canopy or Vegetation
CONSTA	Average Amplifier Output
H2ODEN	Grams/cc of Water in the Air
IPRINT	Output Data Device Designator = 6
ITIME	Game Running Time
ITTAB	Array for Target Parameter List
IVCOV	Index Vegetation Cover, 1=Heavy, 2=Medium, 3=Light, 4=Open, 5=Water
LDUMP	True = Intermediate Calculations Printed, False = No Print
TDEL2A	Time Delay Times 2

Internally Stored Designer Input Values

ALPAIR	Target Noise Due to Aircraft (dB) (105.0, 115.0, 125.0)*
BOTORG	Formation Function for Boats (1.0, 2.0, .2)
DBSURF	Target Noise Modifier Due to Surface (dB) (-3.0, 0.0,6.0)
FOLTBL	Attenuation Due to Foliage Density (0.05,0.1, 0.15)
TBLMAN	Man Signal (dB) (20., 50., 65.0)*
TBLMUN	Target Signal Due to Munitions (dB) (100.0,130.0,150.0)*
TMM	Target Signal Modifier Due to Surface (dB) (-3.0,0.0,3.0)
TROORG	Terrain Function for Men (-3.0,0.0,3.0)
VEHTBL	Vehicle Signal (dB) (85.0,107.0,113.0,125.0)*
WTRTBL	Target Signal Due to Boats (dB) (82.0,88.0,94.0)

Computed Values

ALA	Attenuation Coefficient for Free Air (dB/Meter)
ALPFOL	Attenuation Due to Foliage Density

*Estimates Developed from Reference 1,17, Pg 3-126.

Computed Values (continued)

DBAC	Target Signal Due to Aircraft (dB)
DBMAN	Target Signal Due to Man (dB)
DBMUN	Target Signal Due to Munitions (dB)
DBTARG	Noise Level at Sensor Due to Target (Use Maximum Value)
DBVEH	Target Signal Due to Vehicles (dB)
DBWATR	Target Signal Due to Boats (dB)
DELRV	Distance Travelled by Lead Vehicle Between First and Second Entry
DELTAT	Difference Between Beginning of Game and Time of Last Entry
FN	Total Number of Increments of Time After Passing Point of Closest Approach
GINPUT	Nominal Gain Detection from Noise + Target or Noise Only Environment
ISENPR	Previous Sensor ID
IT	Dummy Index
ITARPR	Previous Target ID
KFOL	Index for Foliage Attenuation, 1=0.05, 2=0.1, 3=0.15
KI	Dummy Index
LFIRST	Logical Indicator
LSEC	Logical Indicator
OUTMAX	Previous Values of OUTMAX
PLOSAC	Total Attenuation Due to Range (Aircraft)(dB)
PLOSSB	Total Attenuation Due to Range (Water)(dB)
PLOSSE	Total Attenuation Due to Range (Munitions)(dB)
PLOSSM	Total Attenuation Due to Range (Man)(dB)
PLOSSV	Total Attenuation Due to Range (Vehicles)(dB)
PLOS6S	Attenuation Due to Divergence (Aircraft)(dB)
RANGE	Range to Target from Sensor
RHO	Intermediate Calculation
RHYPOT	Indicator for Arrival of Last Vehicle at Range of Closest Approach
RVREF	Range to Lead Vehicle on Previous Look for Approach, Target
TARLEN	Target Length Squared
TEMAMK	Air Temperature (Kelvin)
VSIGSQ	Sum of the Voltages Squared
X	Dummy Argument for Random Number Generator

Output Values

LDET	Detection Decision

3.3.2.3 Description of Subroutine Logic and Planning

This subroutine is very similar in structure to the SEISTG subroutine already described so that only a general outline will be presented. Emphasis will be placed on those areas which are not treated in the SEISTG subroutine.

When this subroutine is called, considerable information must be passed to it in order that the proper computations can be carried out and the required outputs generated. These inputs are contained in the arguments of the CALL statement (NOTGTS) and in the BASICT, ATMENV, UNTER, TARGET, CONST, SENSOR, and SENVAR labelled common areas. Some of the input is derived from PRERUN as output from the ACOUBK subroutine and others from previous entries to this subroutine itself, e.g., time of latest report (ITIMLR), time of last entry for a specific sensor (ITIMLE), the amplifier gain (GAIN), and the gain value at last entry (GINLST), all contained in the SENSOR common area. Target information is input through the target common area with descriptors for target type (ITGTTP) and classifications within that type. The target parameters associated with these descriptors are contained within the subroutine as data statements.

As with SEISTG, on entry into the subroutine a target loop is encountered through which passage must be made once for each target determined to be within range by external subroutines (Figure 3.3.-4); thus cycles for total number of targets are made through the target signal development loop. The first type of target considered is the vehicular which is treated in identical manner as that described in SEISTG including the multielement target. Source strengths and surface effects are, of course, those pertinent to the vehicles and contained in the data statements for acoustic source strength. Four vehicle types are included, namely: small (1/4 ton) trucks at 86 dB; medium (107 dB); large (tank, 112 dB); and trains (125 dB). These source strengths are estimates developed from the literature but should be considered to be tentative only at this time. Path loss is proportional to $1/R^2$ but with an additional loss due to foliage included.* The index of foliage loss is developed from data supplied from the terrain table for the location of the sensor in question.

Troop-type targets are treated in composite fashion as in SEISTG with source strength (DBMAN) being a function of troop activity (quiet (20 dB) or noisy (50 dB)) and modified by number of elements in the target, element spacing, troop organization, and surface condition as shown in the expression:

$$\begin{aligned}\text{DBMAN} = {} & (\text{Source Strength})_{\text{single man}} \\ & + 20\,[\log_{10}(2)(\text{No. Men}) - \log_{10}(\tfrac{\text{Spacing}}{3})] \\ & + f(\text{Terrain Function})(\text{Formation Index}) \\ & + f(\text{Surface Modifier})(\text{Surface Noise Index})\end{aligned}$$

This expression is seen to be an elementary estimate for source strength which may be subject to modification when additional data are developed through field experiment or test. Path loss is of the same type as that for vehicles described above.

*See References 1 and 5.

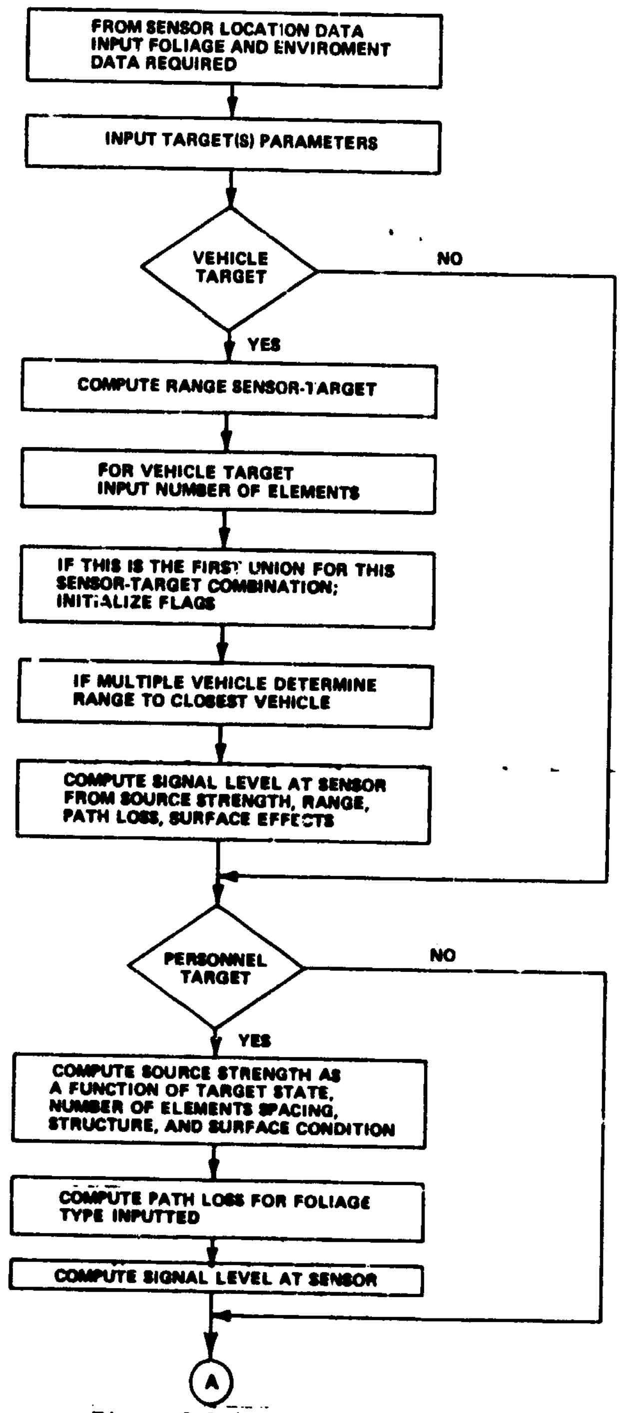


Figure 3.3-4 ACOUTG MACROFLOW
(Sheet 1 of 3)

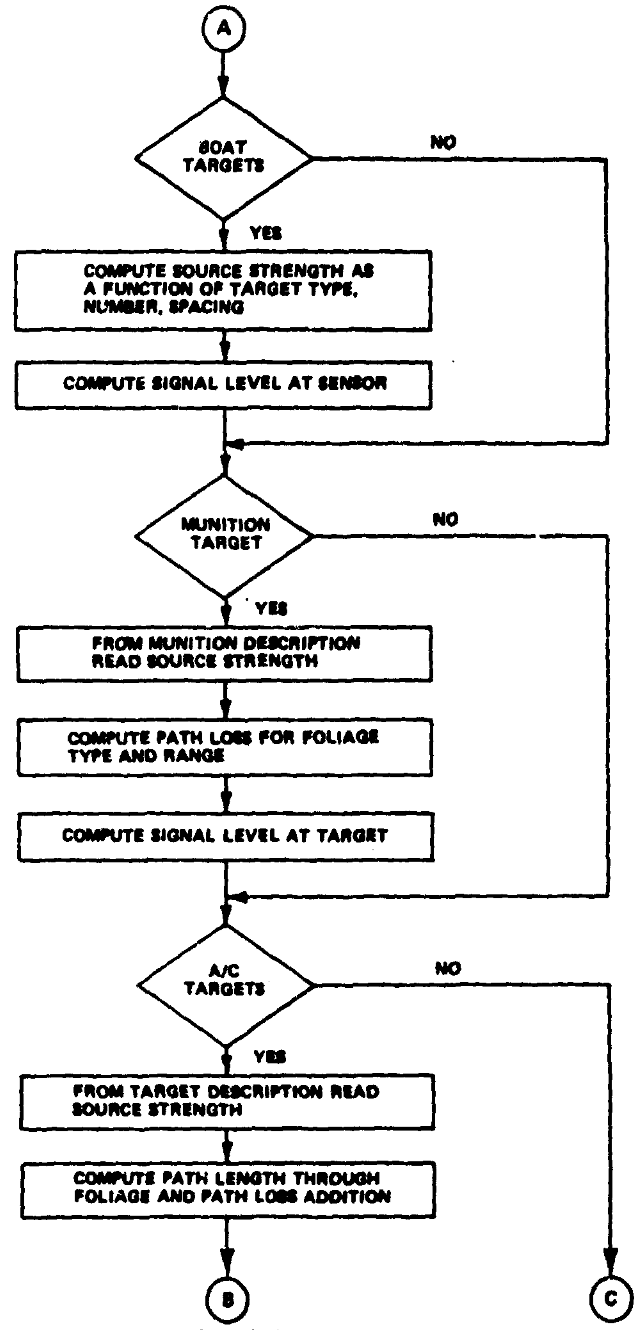


Figure 3.3-4 ACOUTG MACROFLOW
(Sheet 2 of 3)

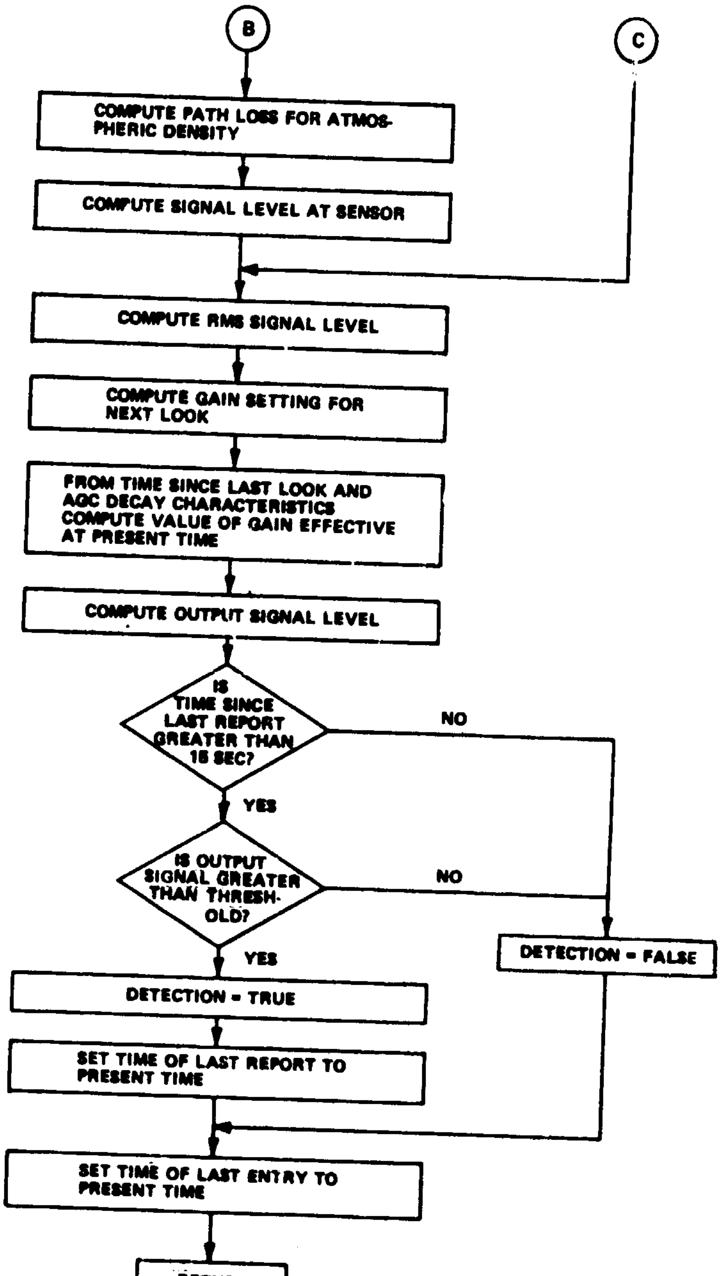


Figure 3.3-4 ACOUTG MACROFLOW
(Sheet 3 of 3)

River craft targets are assigned a source strength (DBWATR) according to the following statement:

$$DBWATR = f\ (\text{Boat Type}) + \text{Spacing Function} \times \text{Formation Function}$$

where the elemental boat signal source strength has assigned values as raft or motorless sampan, 82 dB; motor sampan, 88 dB; patrol craft, 94 dB. These estimates are based on very limited data and as first estimates would appear to be high.

Munition targets are treated as a simple acoustic source with path loss proportional to $1/R^2$, with added loss introduced by foliage. It is assumed that if foliage is present at the sensor, the acoustic signal is attenuated over its entire path from event to sensor by the associated foliage attenuation factor. Treatment of air and foliage refraction processes and determination of extent of foliage between target and sensor were beyond the present scope of the model, but it is to be noted that sufficient information is contained on which to base a much more detailed simulation for the acoustic path loss.

Aircraft targets are treated in a similar way with range through foliage being developed from similar triangle considerations as is done in SEISTG subroutine. Atmospheric attenuation based on water content and temperature is also included in the path loss term as in SEISTG and the same considerations of disregarding aircraft altitude and altitude dependent water density and temperature apply.

Having determined the sum of the signal voltages at the sensor input (VSIGSQ), the output is computed directly from the value of GAIN that applies. GAIN is developed in the same way as for the AGC system of SEISTG. Output detection logic and time indexing are identical to that described in the SEISTG subroutine.

Estimates of source strength and effects of environment on propagation were developed principally from References 1, 2, 4 and 5 although considerable additional literature was reviewed.

3.3.3 Subroutine PIRTG

3.3.3.1 Purpose

Using the background power estimates developed in PIRBK, this subroutine is employed to develop estimates of power due to target and background, and the signal to background ratio from which the probability of detection is determined. A test of the probability by comparison to a random number is then used to determine occurrence or absence of a detection event.

3.3.3.2 Glossary of Inputs, Computed Values and Outputs

Input Values

EXPAN	Area of Input Aperture x Field of View x Optical System Transmission Factor
FIELD	Solid Angle Field of View
IDSNSR	Sensor ID
IDTGT	Target ID
ITGTTP	Target Type
KSTRNG	Index on Strength Type of Target
TEMPEV	Background Temperature
VELTAR	Velocity of Target
WATTBK	Background Derived Power from PRERUN
XSEN	Sensor Position X
XTGT	Target Position X
YSEN	Sensor Position Y
YTGT	Target Position Y

Internally Stored Designer Input Values

AREABO	Area of Target for Boats (0.4,0.7,2.0 $meter^2$)
AREAMN	Area of Target for Man (3.0,10.0, 20.0, 50.0 m^2)
AREAVH	Area of Target for Vehicles (1.0, 5.0, 10.0)

Labelled Common Inputed Values

DELAZ	Angle Between Center Lines of the Two Beams
H2ODEN	Atmospheric, Water Content in Grams/cc
IPCODE	Precipitation Code Identifying Type of Precipitation
IPRINT	Output Data Device Designator = 6
ITIME	Game Running Time
LDUMP	True = Intermediate Calculations Printed, False = No Print
PRATE	Rain Fall Rate (mm/hr)
STEFK	Stefan-Boltzmann Constant/PI ($1.805455X10^{-8}$)
THRESH	Threshold
TIMMAX	Maximum Time Allowed for Detection(Common Senvar), 2.0 Seconds

Computed Values

AREREF	Reference Area Defined by Field of View and Range
ARTAR	Area of Target
BEAMRG	Distance Between Beam Centers at Target Range
BEAMTM	Time Required for Target to Travel Distance BEAMRG
DUM	Dummy Variable
FOGATN	Attenuation Modifier
INTER	Intermediate Index
PRATE1	PRATE x 0.1
PWRA	Fraction of Background Power Received, Target in Field
PWRAIR	Power Due to Radiance of Atmosphere
PWRBK	Background Derived Power
PWRINT	Received Power from Background
PWRTAR	Power at Sensor Aperture Due to Target
PWRTOT	Total Power at Sensor Aperture Due to Target and Background

Computed Values (continued)

RADAIR	Radiance of Atmosphere
RADTAR	Radiance of Target
RANGE	Distance Between Target and Sensor
RATIO	Ratio of Target Area to Area Subtended by the Beam
STOT	Signal to Threshold Ratio
TEMAMK	Ambient Air Temperature (Kelvin)
TEMPTG	Target Temperature
TEST	A Random Number Between 0 and 5

Output Values

LDET	Detection Decision (True or False)

3.3.3.3 Description of Subroutine Logic and Processing

The PIRID device (and other devices of this class) are passive infrared equipments operating in the 7 to 15 micron region. Two fields of view and two detectors are employed to provide both a reference system and a detection logic which potentially reduces false alarms to a very low rate. The two fields of view are offset from one another by a small angle which introduces a dead zone so that there will exist a null zone in target passage from one field to the second. For the PIRID device, azimuthal width for each field is 1.0 degree and the field center lines are separated by 3.0 degrees. A nulling type system is employed using feedback to the input stages of the two independent amplifier channels to reduce the output difference to zero for long-time average effects. When a target enters one field, the null balance is upset, a detection or threshold crossing takes place, and a timed logic sequence is initiated. If the target passes through the second beam, the balance is similarly upset and a second threshold crossing takes place. If the time between threshold crossings is within prescribed limits, a detection report is generated. The second beam must be intercepted within 3 seconds of the actuation of the first beam (TDMAX) and thus detection is a function of target speed. The signal to noise ratio is computed for one beam only and from it the probability of actuating each beam is determined and tested by selection of a random number. Timing requirements may, of course, be met by noise leading to false alarms as already discussed in the PIRBK subroutine.

On entry into this subroutine (Figure 3.3-5) a check is made to determine if target velocity is adequate to meet the logic requirements of the sensor. It is assumed that the device is positioned so that the centerline of the system will be perpendicular to the trail or road being monitored so that target vector will always be perpendicular to the centerline. For more general applications such as base defense when small fields of view and long ranges are employed, a more general solution may be required. Since the PIRID is a very limited range device, the approximation employed is considered to be satisfactory.

If the target velocity is satisfactory, target temperature is next established. Again, relatively little information on the average temperature of a target is available, so the set of data developed as part of the Warren Grove experiments (Ref. 6) were used. The data were plotted and straight line fits were applied as shown in Figure 3.3-6 (also see Ref. 7, page 103). Using those results, it is argued in the model that the target of cross sectional

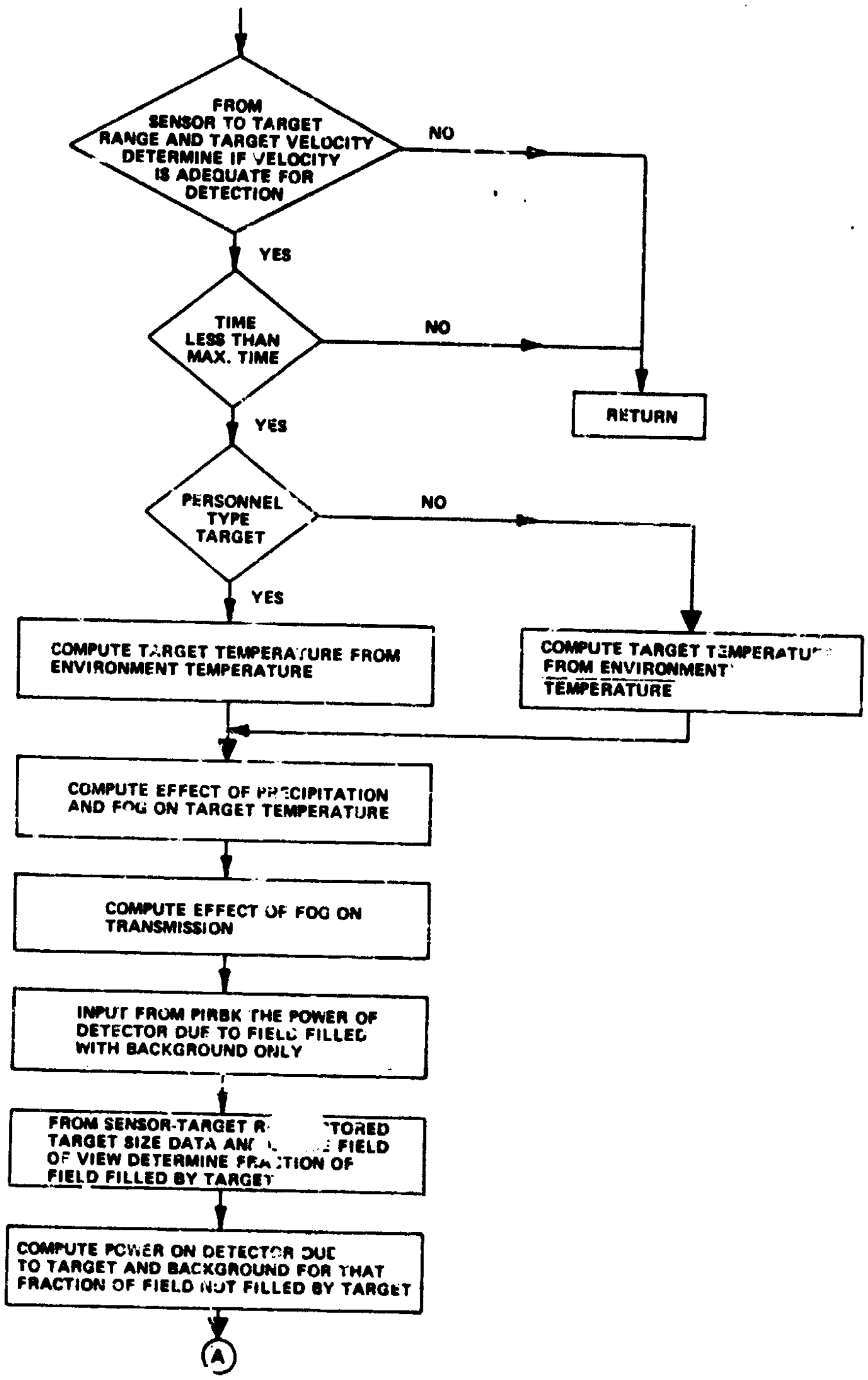


Figure 3.3-5 PIRTG MACROFLOW
(Sheet 1 of 2)

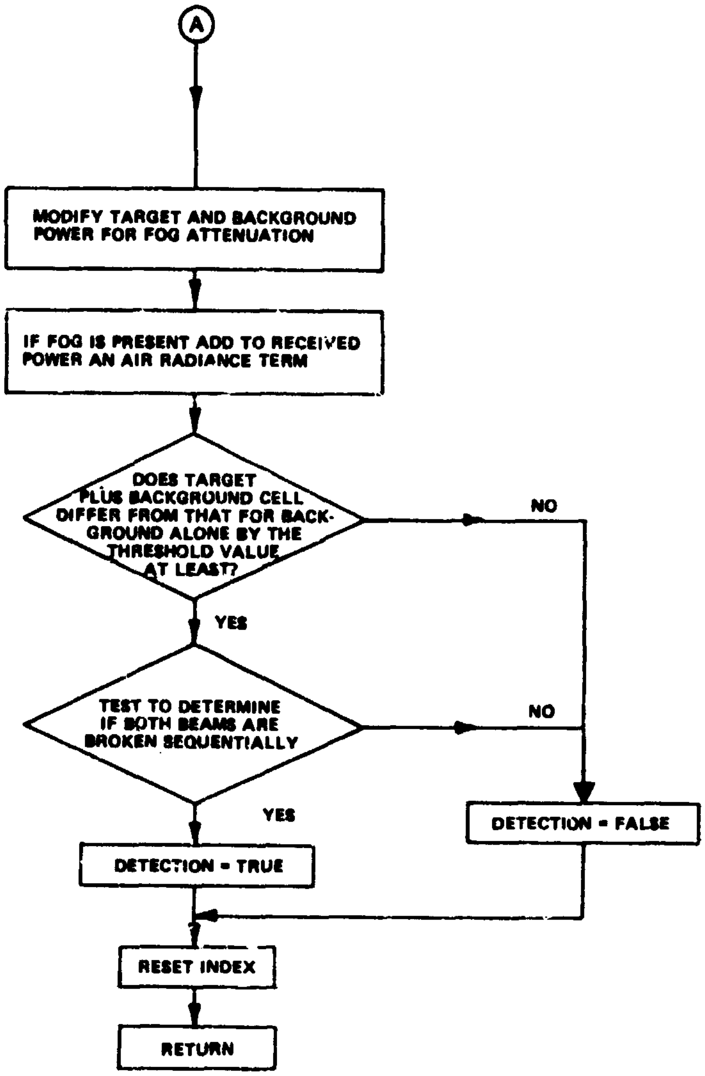


Figure 3.3-5 PIRTG MACROFLOW
(Sheet 2 of 2)

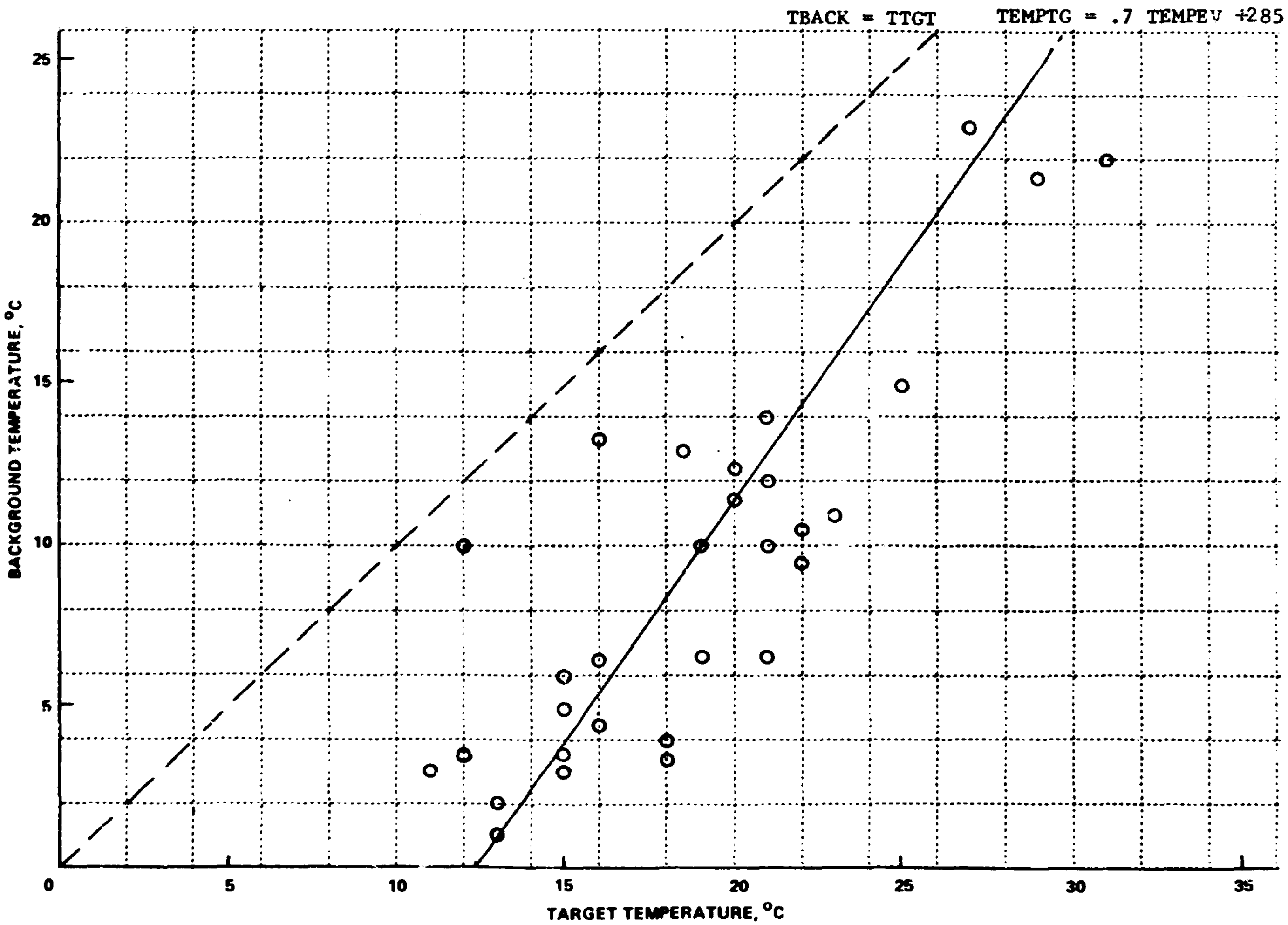


Figure 3.3-6 **TARGET TEMPERATURE VS. BACKGROUND TEMPERATURE FOR MAN TARGETS (FROM DATA IN REFERENCE 6)**

area given in the data statements of this subroutine will have temperatures defined as follows:

a. For military personnel type target, the maximum permissible temperature of the man and clothing will be 318° K, which will apply if the background temperature should be greater than 318° K. For lower background temperatures, the temperature of the target (TEMPTG) is related to the background temperature by the relation:

$$\text{Target Temperature} = 0.7(\text{Environmental Temperature}) + 285$$

where the environmental temperature (TEMPEV) is determined in the subroutine PIRBK and passed on to PIRTG through the SENVAR common area.

b. If the target is a large animal, the target temperature defined for a target of cross sectional area two meters square is given by:

$$\text{Target Temperature} = 0.7(\text{Environmental Temperature}) + 280$$

There is no support for animal temperatures in Reference 6, but the tentative equation is considered satisfactory for present purposes.

c. For inanimate objects the target temperature was found in Reference 6 to be approximately five degrees higher than the background temperature. Thus,

$$\text{Target Temperature} = \text{Environmental Temperature} + 278$$

while:

$$\text{Ambient Air Temperature} = \text{Environmental Temperature} + 273$$

d. Target temperature (TEMPTG) will be modified substantially by precipitation since both target and background temperatures are reduced by the convection cooling provided by precipitation. At high rates of precipitation the differences in these temperatures approach zero. This argument is introduced by the following statements:

$$\text{Target Temperature} = \text{TEMPTG} - 0.1(\text{Precipitation Rate})(\text{TEMPTG} - \text{Ambient Air Temperature})$$

$$\text{Target Temperature} = \text{TEMPTG} - \frac{\text{Water Density}}{0.3 \times 10^{-4}}(\text{TEMPTG} - \text{Ambient Air Temperature})$$

where it can be seen that target temperature is reduced by the difference between target and ambient air temperatures and by arguments related directly to precipitation rate or fog density. The ATMENV table is not specific on fog except to indicate that fog is present. In the sensor subroutines, fog density is assumed to be related to atmospheric water content, H2ODEN, as a temporary expediency. Again these arguments are unsupported by data but appear to be adequate for present purposes.

If the difference between target and background temperature is less than 0.25 degrees C., the target cannot be detected no matter how much of the field of view the target fills so that a direct step to the RETURN statement is made. This is a sensor design consideration which presents the minimum detectable temperature difference under optimum conditions.

The target also may not fill the field of view in which case it is necessary to determine the total power incident on one of the detectors where part of the power is derived from the target and the second component from that amount of background within the same field that is not obscured by the target. The target areas are selected on the basis of target type (ITGTTP) and classification (KSTRNG) from the data statements contained within the subroutine. If the area of the target is greater than the reference area (AREREF) which is the area subtended by the field of view at the range of the target, then the problem is simply that of comparing target with background power. If the target area is less than AREREF, then target plus background must be compared with background alone as noted above.

It is to be noted that in general the background will be distributed in range. The lower reaches of the field of view will be exposed to surface vegetation near to the sensor while at the upper limits, background may be located at a considerable range from the sensor. Since, however, all the background is taken to be at the same temperature, the field of view is taken to be filled by a black body radiating at a temperature defined as that of the background. This approximation will be somewhat in error for poor weather conditions where atmospheric attenuation, which is a function of range, is an important factor.

Having defined the target and background, power is determined from the subroutine PIRBK and the target component is computed by the expression:

$$\text{Radiance of Target} = 0.27\,\sigma(\text{Target Temperature})^4$$

where σ is the Stefan-Boltzmann constant/π. It can be seen that the Stefan Boltzmann equation for radiance is thus employed. The factor, 0.27, is introduced to assign the 7-15 micron interval fraction of total radiance to radiance of target.

The power on the detector due to the target is computed as:

PWRTAR = (Target Radiance)(EXPAN)(RATIO)

where EXPAN is the geometrical gain factor relating device characteristics (area, field of view, optical system transmission factor) to power incident on the detector as computed in PIRBK, and RATIO is the fraction of the field of view filled by the target. If attenuation due to fog is an important factor, the power due to target and background are both scaled to lower values as, for example:

$$PWRTAR_{Atten} = (PWRTAR)\, e^{-(Attenuation\ Modifier)(Range)}$$

In addition, the radiance attributable to the intervening atmosphere (between sensor and target) is added. (Attenuation due to precipitable water is not included because of the limited range of this sensor. A more general approach would include atmospheric attenuation as is done as part of the THERML subroutine.)

The signal to threshold ratio is next computed. The signal is taken to be the difference in power on the detector due to target plus background compared with the power on the detector due to background alone. Threshold is a designer input selected here to denote detections when the target filled field of view observes a temperature 0.25 degrees C. different from that of the background filled field of view.

Two tests are made in which the signal to threshold ratio is compared with a random number. If in both tests the signal to threshold ratio is the greater, a detection report is generated. For any other combination of outcomes, no report is made. This argument simply indicates that both beams must be actuated, and for low signal to threshold ratios, there is some probability that one or neither of the beams will be activated.

Finally, it is noted that the simulation described depicts the situation for one target element. For multi-element targets the entire process is repeated at those times that the MSM executive subroutine determines that a target element passes through the field.

3.3.4 Subroutine ARFTG

3.3.4.1 Purpose

This subroutine is provided to determine the probability that a target element or elements will excite an intrusion detection element (Button Bomblet) in a two-second interval for the specific targets within Button Bomblet distributions.

Input Values

AREADN	Area Density of the Bomblets in Number (Square Meters)
IDSNSR	Sensor ID
IDTGT	Target ID
IMAG	Index on Bomblet Type (0=Magnetic, 1=Noisless)
ITGTTP	Target Type
KSTRNC	Index on Strength Type
NOTFLD	Number of Targets in a Seeded Area
VELTAR	Velocity of the Target

Labelled Common Inputed Values

IPRINT	Output Data Device Designator = 6
ITIME	Game Running Time
LDUMP	True = Intermediate Calculations Printed, False = No Print

Internally Stored Designer Input Values

AREAM	Area Covered by Troop Target/Troop/Sec/Meter/Sec *
AREAMM	Area Covered by Troop Target/Troop/Sec/Meter/Sec (Mag. But.)
AREAV	Area Covered by Vehicle, Target/Vehicle/Sec/Meter/Sec
AREAVM	Area Covered by Vehicle, Target/Vehicle/Sec/Meter/Sec (Mag. But.)
SAMPTM	Time Over Which System is in a Single Sample Cycle (2 Sec.)

Computed Values

ARTAR	Area Covered by Target/Sec/Meter/Sec
DUM	Dummy Variable
PROB	Overall Probability of Excitation in a Sampling Interval
PROBUT	Probability of Actuation of Bomblet by a Single Target
TARNUM	Number of Targets in a Seeded Area

Output Values

LDET	Detection (True-False)

3.3.4.2 Description of the Subroutine Logic and Processing

In subroutine ARFBK, the area density of intrusion detection elements was computed from the planner input information regarding number of elements and the configuration and area size of the distribution. As a part of the MSM executive subroutine, computations are carried out which describe the number of target elements contained within the sensor field at any time (TARNUM). With this information and a set of stored response areas contained within the DATA statement sets of this subroutine, the probability of an actuation is determined. By testing against a random number, the success or failure of the detection test for the entry in question is determined.

In the subroutine the areas that are disturbed by individual targets are stored as data under the names AREAM, AREAV, AREAMM, and AREAVM (see glossary) for both trooper and vehicle type targets and for

* This area was estimated by measurement of the area swept by a man's feet in one second given a speed of 1 meter/sec. The area covered by all elements of the target is then a function of the number of elements, the target speed, and the observation time. Values are given in the program listing.

noiseless and magnetic type buttom bomblets. The areas are the areas per element per second per meter per second that each target would perturb. On entry into the subroutine (Figure 3.3-7), target specification is first assessed from the descriptors ITGTTP and KSTRNG, parameters that serve as indices for selection of the required areas from the data sets discussed above.

The average number of bomblet activations in a sampling period (AVGNBT) is given by the expression:

$$AVGNBT = ARTAR \cdot VELTAR \cdot SAMPTM \cdot AREADN \cdot TARNUM$$

where ARTAR is the target disturbance area discussed above, VELTAR is the target velocity, SAMPTM is the sample time defined by the data statement as two seconds, AREADN is the area density of bomblets computed in subroutine ARFBK, and TARNUM is the number of elements in the field as computed by the Executive subroutine. The probability of actuating a bomblet in any sampling period (PROB) is given by

$$PROB = 1 - e^{-AVGNBT}$$

Tests are then made against a random number, one on each entry to the subroutine. If the probability exceeds a random number selected from the uniform distribution lying between 0.0 and 1.0, a detection report is generated, otherwise the detection is declared to be false. Return to the calling routine is then made with the detection result being transmitted through the argument list in the sensor call statement.

3.3.5 Subroutine MAGTG

3.3.5.1 Purpose

This subroutine is employed to develop the report events and times that result from the interaction between targets and the magnetic, gradiometer type sensor as the targets pass in the vicinity of the sensor.

3.3.5.2 Glossary of Inputs, Outputs, and Computed Values

<u>Input Values</u>

IDSNSR	Sensor ID
IDTGT	Target ID
IREF	URN Reference Number
ITGTTP	Target Type
KSTRNG	Target Classifier
NCLOSE	Number of Closest Element to Sensor
NOELEM	Number of Elements in Target Group
THRESH	Threshold
TIME	Running Time in Half Second Intervals

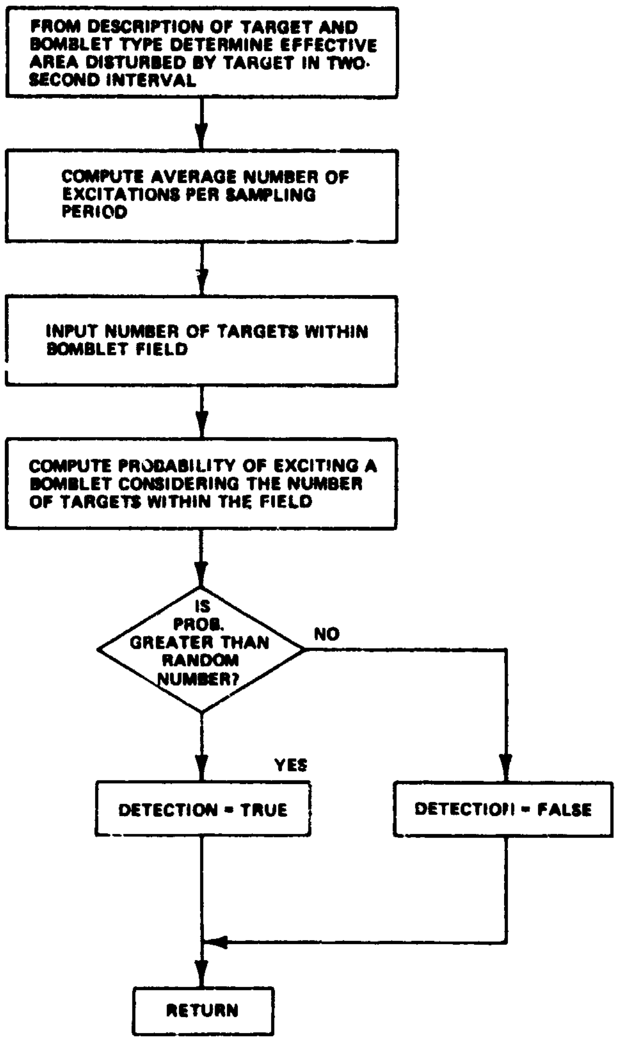


Figure 3.3-7 ARFTG MACROFLOW

Input Values (continued)

TSPACE	Target Spacing (Average)
VELTAR	Target Velocity
VX	Target Velocity X - Direction
VY	Target Velocity Y - Direction
XSENS	X-Coordinate of Sensor
XTGT	X-Coordinate of Target
YSENS	Y-Coordinate of Sensor
YTGT	Y-Coordinate of Target

Internally Stored Designer Input Values

PERM	Target Permeability
POLIND	Pole Induced Fact, Unit Poles/Gauss
POLRES	Residual Pole Strength, Unit Poles
RMAX	Maximum Range
TARLEN	Target Length (cm)
ZOTABL	Table of Heights Above Ground (cm)

Labelled Common Inputed Values

IPRINT	Output Data Device Designator = 6
ITIME	Game Running Time
LDUMP	True = Intermediate Calculations Printed, False = No Print
XFIELD	X Component of the Earth's Magnetic Field *
YFIELD	Y-Component of the Earth's Magnetic Field
ZFIELD	Z-Component of the Earth's Magnetic Field

Computed Values

CLOSE	Dummy Variable
DELTAT	Time Difference
DUM	Dummy Variable
EMOMX	Dipole Moment Due to Residual Poles, X Component
EMOMY	Dipole Moment Due to Residual Poles, Y Component
EMOMZ	Dipole Moment Due to Residual Poles, Z Component
HDOTZ	Time Derivative of Vertical Component of Field at Sensor
I	Dummy Index
INDEXH	One Space More Than Closest Element Index
INDEXL	One Space Less Than Closest Element Index
IREFP	URN Reference Number Changed Every 60 Seconds
ISAVE	Dummy Variable
ISKIP	Dummy Variable
ISIZE	Index on Target Length
J	Dummy Index
KI	Dummy Index
MULT	Dummy Variable
PART1	Random Component Associated with X for Carried Target
PART2	Random Component Associated with Y for Carried Target
PART3	Random Component Associated with Z for Carried Target
PARTSQ	Vector Length of X, Y, Z Random Components
POLL	Computational Variable

*Reference 4, Page 3-126

<u>Computed Values (continued)</u>

RANGE	Range from Target to Sensor (Meters)
RANGE5	Range to 5th Power
RANGE7	Range to 7th Power
RMOMX	Dipole Moment in X Due to Residual Poles
RMOMY	Dipole Moment in Y Due to Residual Poles
RMOMZ	Dipole Moment in Z Due to Residual Poles
SPACE	Distance from Closest Element
TLTC	Time of Last Threshold Crossing
VXCOMP	Intermediate Calculation
VYCOMP	Intermediate Calculation
XCOMP	Computational Variable
XLEN	Length of Target in X Direction
XMOM	Total Moment for X Dipole Component
XVAL	X Coordinate Position for ith Target Component
XYCOMP	Product of (XCOMP · YCOMP)
YCOMP	Computational Variable
YLEN	Length of Target in Y Direction
YMOM	Total Moment for Y Dipole Component
YVAL	Y Coordinate Position for ith Target Component
ZLEN	Length of Target in Z Direction
ZMOM	Total Moment for Z Dipole Component

<u>Output Values</u>

LDET	Detection Indicator (True-False)

3.3.5.3 Description of Subroutine Logic and Processing

The MAGID sensor which is the basis for this subroutine (Figure 3.3-8) consists of two, many turn, solenoids which are emplaced in the earth with longitudinal axis along the vertical so that voltages are produced in response to changes in the vertical or Z component of the magnetic field at the solenoids. The solenoids are placed so that one unit is near the trail or road being monitored while the second is emplaced as remote to the trail as possible serving as a background reference for the first. The solenoids are connected in opposition to one another so that for the same changes in field at both solenoids no output is produced. In the frequency range employed, earth's field fluctuations are area type fluctuations of such extent that the change in field at both solenoids is approximately the same to within the threshold setting. Thus the earth's field fluctuations are essentially removed and false alarm rates are held to a low level. When, however, a ferrous material is passed along the trail the change in field at the trail solenoid is much greater than that produced at the remote solenoid so that an imbalance exists which if large enough and of sufficient duration, will actuate the detection logic networks. While the difference in signal is the important signal component, it is assumed in this model that the target will always be much closer to the trail solenoid than to the reference unit so that only the field changes at the trail solenoid need be examined. Hence, the simulation is that for a single solenoid.

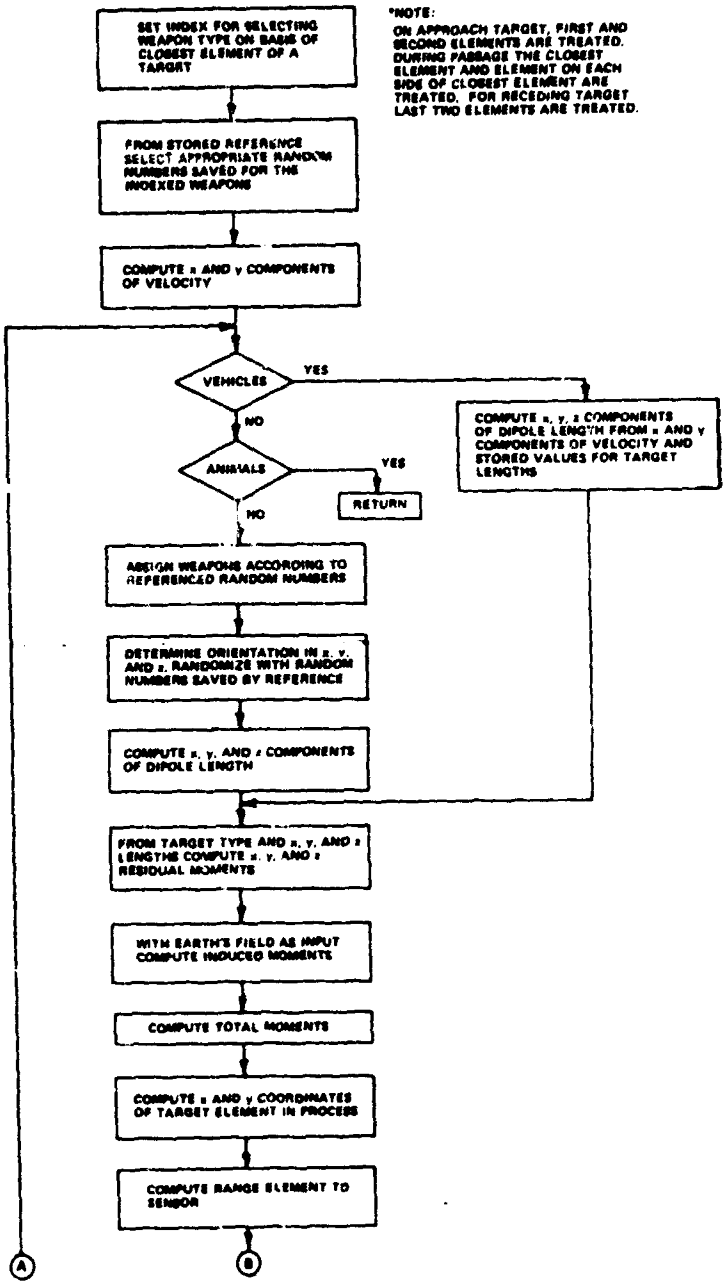


Figure 3.3-8 MAGTG MACROFLOW
(Sheet 1 of 2)

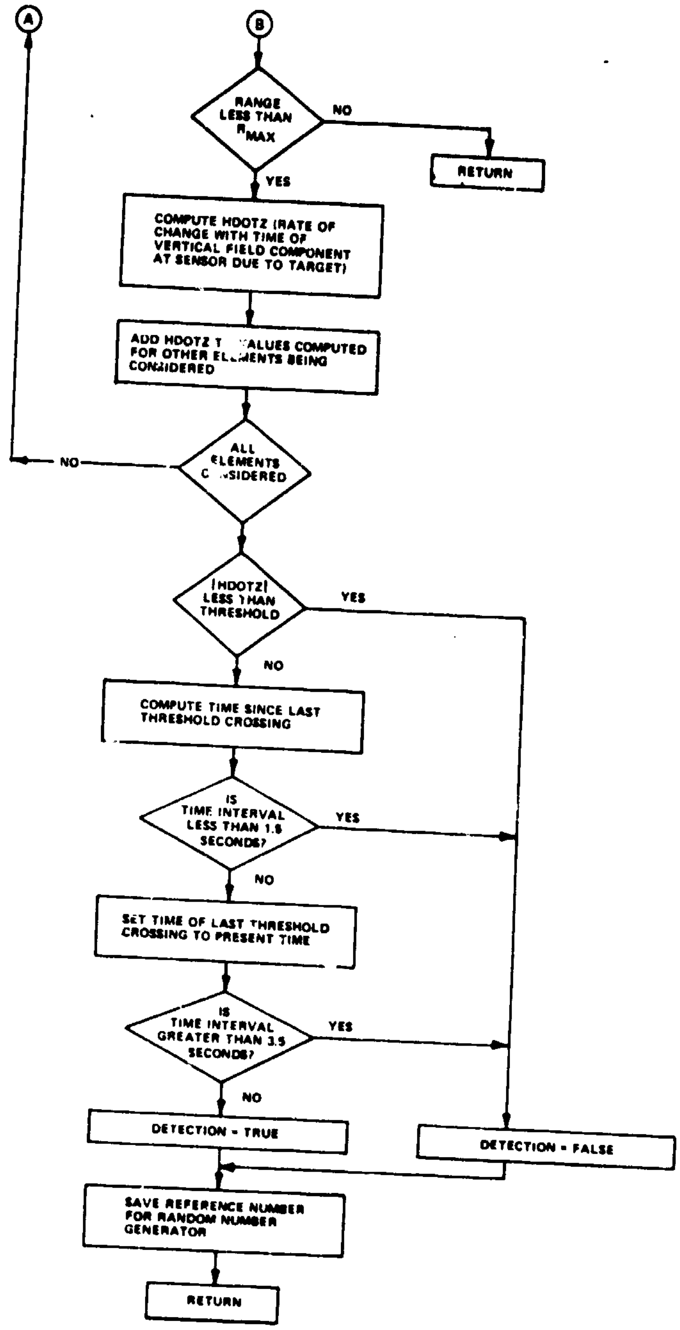


Figure 3.3-8 MAGTG MACROFLOW
(Sheet 2 of 2)

It is found that the rate of change of field strength at the solenoid is a function of the range to the ferrous object from the sensor, its size and orientation, prior induction history, earth's field components and rate of movement with respect to the sensor. It can be shown that the potential (V) at the sensor due to the magnetic dipole (M) described by the ferrous object is given by:

$$V = \frac{M}{r^3} = \frac{(x_o + v_x t) M_x + (y_o + v_y t) M_y + z_o M_z}{[(x_o + v_x t)^2 + (y_o + v_y t)^2 + z_o^2]^2}$$

where x_o, y_o, and z_o are initial or reference target positions with respect to the sensor and $(x_o + v_x t)$ is the X-axis position of the target with respect to the sensor at any time t. The vertical component of the magnetic field (H_z) is given by:

$$H_z = -\frac{\partial V}{\partial z_o} = -\frac{M_z}{r^3} + \frac{3z_o[(x_o + v_x t) M_x + (y_o + v_y t) M_y + z_o M_z]}{r^5}$$

$$r = ((x_o + v_x t)^2 + (y_o + v_y t)^2 + (z_o)^2)^{1/2}$$

Then the rate of change of H_z with respect to time is given by

$$\frac{dH_z}{dt} = 3\frac{M_z}{r^5}[(x_o + v_x t) v_x + (y_o + v_y t) v_y) + \frac{3z_o}{r^5}(v_x M_x + v_y M_y)]$$

$$- 15\frac{z_o}{r^7}[(x_o + v_x t) M_x + (y_o + v_y t) M_y + z_o M_z]$$

$$[(x_o + v_x t) v_x + (y_o + v_y t) v_y]$$

To carry out this computation it is necessary to know the dipole moments, M_x, M_y and M_z, as well as the geometrical factors and rates. The targets employed at the present level of simulation are described by target type (ITGTTP) and classification (KSTRNG). In order to limit the complexity of these descriptors, it was decided that ferrous objects would be assigned one to each target element for troop type targets, with definition of dipole moment limited to one value for each vehicle element. However, since entry must be made to this subroutine over an extended period of time in which time several sensors may actually be encountered, it became necessary to include a memory system so that assignment of weapons and their orientation can be retained on a one-to-one basis for the duration of a target-sensor engagement. Thus, each element of a troop target is assigned a weapon on entry to this subroutine and the weapon is oriented in a random manner but the orientation is held for a time corresponding to the duration of the engagement. Later, engagements of the same target with other magnetic sensors will find a redistribution and reorientation of the weapons among the elements. It is also to be noted that the residual and induced pole strength per unit field, the target lengths, and heights are stored as data statements in this subroutine.

On entry a call is made to subroutine URNASK to obtain the value of the dummy variable, ISAVE, which is an index on the random number generator. Then, since the sensor is applicable only to troop and vehicle type targets, the specification of target type greater than two will cause a branch to return to the calling routine with a detection equal false report. Next it is necessary to determine which element of a multi-element target is closest to the sensor (NCLOSE) so that the element and those adjacent to it can be treated as the target at the present time (Figure 3.3-9). Closest target element (NCLOSE) is computed by the MSM executive routine and passed into the subroutine through the calling sequence. The value of NCLOSE is used to determine the indices on the computing loop by which the several contributions to the time derivative of the vertical field component (HDOTZ) are determined. If the target is approaching, the closest element is the first element in the column and only the first and second elements are examined; thus, NCLOSE = 1 and NCLOSE + 1 = 2 and CLOSE, an index on the number of spaces between NCLOSE and the first target element is set to zero.

For a receding target in which the last element has passed the point of closest approach the targets considered are the last and second to last elements. In this case the second to last element = NOELEM-1 and the last element = NOELEM, where NOELEM is the number of elements in the target. For NCLOSE equal to any element from two to NOELEM-1, the indices run from NCLOSE-1 to NCLOSE+1; that is, three targets are considered to be effective. In this case, the first effective target (NCLOSE-1) will be at an X, Y position determined from the X and Y coordinates of the lead element and at a distance depending on average element spacing, TSPACE.

In the next set of statements, the appropriate references on the random number selector are established. The reference number is held constant for at least 60 seconds of game time. The integer variable is then used as index in the call to URNORG subroutine which initializes the random number generator index. Note that as game time changes by 60 seconds the reference number is indexed by one but a whole new sequence of random numbers results from this change of reference. Then, the proper number of random numbers must be skipped in order that assigned weapon and orientation or vehicle random variable may be properly assigned. Having determined the set required, a computing loop is employed to actually select the random numbers. This sequencing is required in order that the target orientation may be maintained fixed for the short intervals of time considered in target passage.

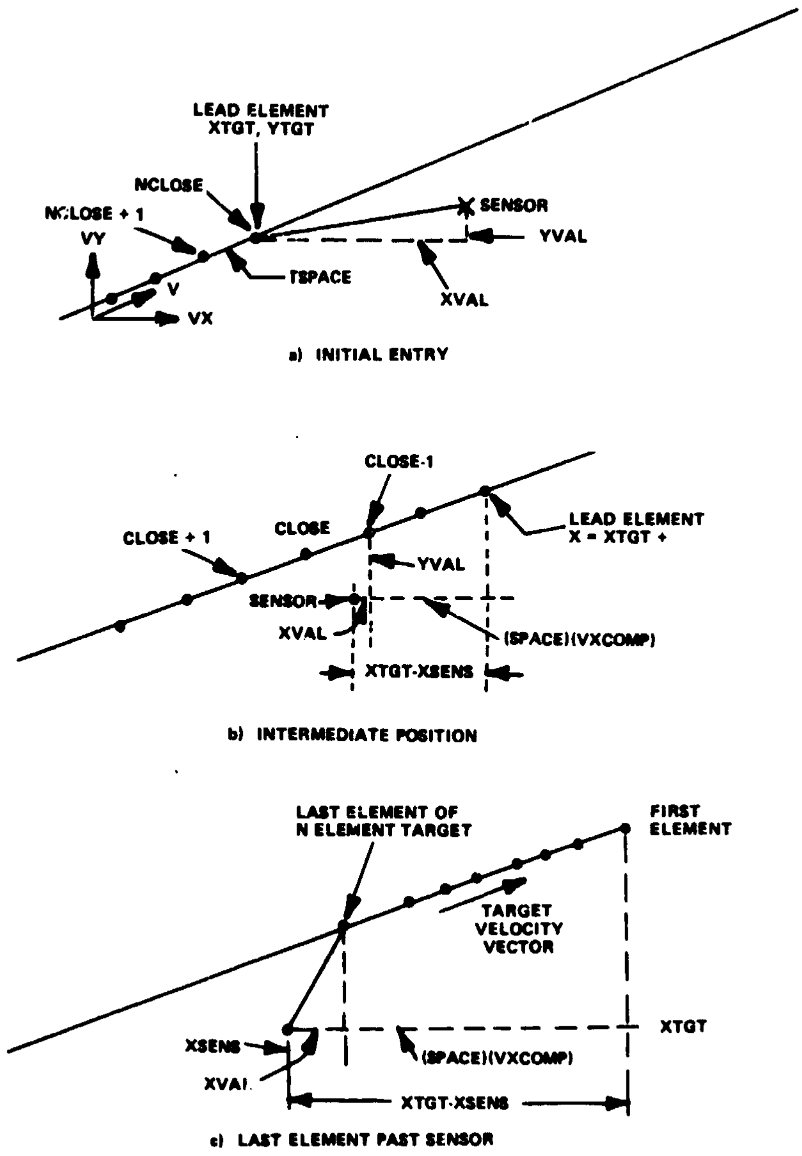


Figure 3.3-9 TARGET-SENSOR GEOMETRY

Before entering the major loop in which HDOTZ is computed, HDOTZ is initialized to 0.0 and the direction cosines of the velocity vector are determined as VXCOMP and VYCOMP since these values do not change with progress through the loop.

If the target is defined as a troop target, then a weapon will be assigned to each element within the limits of the closest element indices according to the value of target length index which will range from one to seven depending on the random number selected. Next the X, Y, and Z components of target length are computed. The three basic random components associated with carried targets (PART1, PART2, and PART3) will vary from -1 to +1 depending on the random numbers selected. The values of X, Y, and Z target length are then determined from the ratio of PART1, 2 or 3 to PARTSQ where

$$PARTSQ = \sqrt{PART1^2 + PART2^2 + PART3^2}$$

The reason that the range -1 to +1 is employed is to allow negative and positive values of X, Y, and Z target length components so that the dipole moments formed by the residual poles may be positive or negative, i.e., so that there will not be a biased orientation.

Vehicle targets are assumed to be oriented with the dipole axis along the direction of travel, that is, parallel to the trail or road. A random component is associated with the X, Y, and Z target length components for vehicles so that vehicles of the same size will not appear to be identical to the sensor because, in practice, vehicle to vehicle variations are observed.

The three components of residual dipole moment, that due to residual pulse, are determined, e.g., for X from:

$$(\text{Dipole Moment})_X = \text{Residual Pole Strength(ISIZE)} \cdot (\text{Length of Target})_X$$

where the index ISIZE provides the means for selecting the proper value from the data stored in this subroutine. Similarly the induced moments are found from

$$(\text{Residual Poles Dipole Moment})_X = \text{POLL} \cdot (\text{Earth Mag Field})_X \cdot |\text{X Length of Target}|$$

The absolute value of X target length is used because the polarity associated with the induced poles will depend only on the earth's components. The POLL computational term arises from the induced poles and a factor for target permeability (PERM) which is included to allow study of changes in relative permeability of the target material. The total moments are then computed as the sums of the induced and residual components.

Having defined M_x, M_y, and M_z for the particular element being considered, the position of that element with respect to the sensor is next determined. Since target position is given as the location of the lead element in the target, the position relative to the sensor for the element in question is determined by computing the X and Y distances from the lead element at which the element in question is located. This operation is carried out by the following computations:

$$\text{Distance From Closest Element} = \text{CLOSE} \cdot \text{TSPACE}$$

where CLOSE is the number of spaces of length, TSPACE, that the element lies behind. Then with lead element coordinates, sensor coordinates, and offset from the lead element for the element in question, the X and Y components of the horizontal projection of the target element-sensor line can be determined (XVAL and YVAL). With these components and z_o, the element-sensor range is determined.

After forming the remainder of the intermediate variables required, XCOMP, YCOMP, and XYCOMP, the time rate of change of vertical field at the sensor (HDOTZ) due to the target element is computed. CLOSE is then increased by one and return is made to the target element loop to introduce the second element in the same manner as for the first as described above. The value of HDOTZ for the second element is added vectorally to that due to the first. If a third element is to be considered, a third pass is made through the loop. In each case the random numbers are selected in appropriate sequence from the index provided.

When the elements have been processed for the particular instant of time in question, the detection logic simulation is examined. If the absolute value of the time derivative of vertical field (HDOTZ) is less than the threshold, there can be no detection event and control is returned to the executive subroutine. If, however, HDOTZ is greater in absolute value than the threshold a test of the logic circuitry is made. If the time since the last threshold crossing lies between 1.5 and 3.5 seconds, a detection is declared to be true, otherwise false. If the time interval (DELTAT) since last crossing is greater than 1.5 seconds, time of last threshold crossing is set equal to the present value of time so that if DELTAT is greater than 3.5 seconds, the present crossing serves as the initiator of the logic timing for reference in future entries for this particular sensor. In all cases, a reference number (ISAVE) is called to reset the random number generator before control is returned to the executive subroutine.

3.3.6 Subroutine RADAR

3.3.6.1 Purpose

This subroutine is employed as a simulation for MTI type radars to determine the probability of detecting targets of the type and range specified by the executive subroutine. Having determined the probability, a test is made against a uniform random number to decide whether or not a detection had occurred.

3.3.6.2 Glossary of Inputs, Computed Values, and Outputs

Input Values

ASPEED	Aircraft Velocity
BEAMAZ	Antenna Azimuth Beamwidth (Radians)
BEAMEL	Antenna Elevatation Beamwidth (Radians)
CLIMFR	Clutter Improvement Factor
FCUTHI	Upper Corner of Filter (Hz)
FCUTLO	Lower Corner of Filter (Hz)
FNKTB	Filtered Thermal Noise Level (dB)
HGTAC	Height of Aircraft (Meters)
HGTANT	Height of Antenna (Above Ground)
ICOH	= 0 Means Coherent, =1 Non Coherent
IDSNSR	Sensor ID
IFOL	= 0 Means No Foliage, = 1 Foliage
ITGTTP	Target Type
KSTRNG	Target Classifier
NOTFLD	Number of Elements in the Field
PFA	Probability of False Alarm
PRIMFR	Precipitation Improvement Factor
RADCAR	Radar Characteristic
RAMBDA	Radar Wave Length (Meters)
RAZANG	Radar Azimuth Angle Measured with Respect to Aircraft Axis
RGATE	Range Gate (Meters)
SCANRT	Scan Rate (Radians/Second)
SIGSTB	Standard Deviation of Clutter Spectrum (Radar Instability)
TARHGT	Target Height (Meters)
VRADL	Radial Velocity, Relative (Meters/Second)
XSENS	X Coordinate of Sensor
XTGT	X Coordinate of Target
YSENS	Y Coordinate of Sensor
YTGT	Y Coordinate of Target

Labelled Common Inputed Values

CHUL	Upper Limit Canopy or Vegetation Height (Meters)
IPCODE	Precipitation Code Identifying Type of Precipitation
IPRINT	Output Data Device Indicator = 6
IVCOV	Index of Vegetation Cover, 1=Heavy, 2=Medium, 3=Light, 4=Open, 5=Water
LDUMP	True = Intermediate Calculations Printed, False = No Print

Labelled Common Inputed Values (continued)

PI	Ratio of the Circumference of Circle to Its Diameter (3.14159?)
PRATE	Precipitation Rate (mm/hr)
SQ2	Square Root of 2.0
VISBLL	Lower Limit, Ground to Ground Visibility (Meters)
VISBUL	Upper Limit, Ground to Ground Visibility (Meters)
WSPEED	Wind Speed (km/ft)

Internally Stored Designer Input Values

ATATEN	Atmospheric Extinction Table (Two-Way) *
BSECTN	Clutter Cross-Section Per Unit Area Table
DCLEAR	Clear Distance to Foliage, Minimum Value
FPREC1	Precipitation Factor Table
FPREC2	Precipitation Factor Table
TARGHT	Target Height Set to 1.5
TLAM	Table of Wave Lengths
XSCMAN	Radar Cross Section of Target in Range Gate - Man
XSCBOT	Radar Cross Section of Target in Range Gate - Boat
XSCVEH	Radar Cross Section of Target in Range Gate - Vehicle

Computed Values

CLATEN	Attenuation for Clutter Signal
COSRDA	Cosine of Radar Depression Angle
CSECTN	Clutter Cross Section (Square Meters)
CSUV	Precipitation Cross Section Per Unit Volume
DBCLUT	Clutter Level in dB at Receiver
DBPREC	Precipitation Signal Value (dB)
DBSIG	Target Signal Level (dB)
DUM	Dummy Argument
FDOPLR	Doppler Frequency, Target (Hz)
FIPCL	Filtered Clutter Power (Watts)
FIPPRE	Filtered Power Precipitation (Watts)
FNKTBP	Filtered Thermal Noise Level (Watts)
FTGMAX	Maximum Usable Doppler Frequency (Hz)
FTGMIN	Minimum Usable Doppler Frequency (Hz)
GGVISB	Ground to Ground Visibility (Meters)
HTFOL	Height of Foliage
I	Index to Pick Up Tabular Values as a Function of Wavelength
IPRECP	Precipitation Code Modified for Radar
IUT	Index on Unit Terrain
IUTS	IUT of X, Y Sensor Position
PATEN	Attenuation Coefficient Due to Precipitation
PSECTN	Precipitation Cross Section
PWCLUT	Clutter Level in Watts at Receiver
PWPREC	Precipitation Signal Value (Watts)
PWSIG	Signal Level (Watts)
RANDOM	Random Number
RANGE	Slant Range, Radar to Target (Meters)

*Designer Input values are given in the program listing.

Computed Values (continued)

RANGEF	Range Through Foliage
RANGE4	Range to the 4th Power
SIGCAN	Standard Deviation of Clutter Spectrum Due to Scan Function
SIGPRE	Standard Deviation of Precipitation Signal
SIGTOT	Overall Standard Deviation
SIGTS2	Overall Standard Deviation Times Square Root of 2.0
SIGVEL	Standard Deviation of Clutter Spectrum Due to Aircraft Velocity
SIGWND	Standard Deviation of Clutter Spectrum Due to Wind
SINRAA	Sine of Radar Azimuth Angle
SINRDA	Sine of the Radar Depression Angle
SUMVAR	Sum of Variances
TARNUM	Number of Elements in Field
TATATN	Total Atmospheric Attenuation
TSECTN	Radar Cross Section of Targets in Range Gate

Output Values

LDET	Detection Indicator (True-False)
PD	Probability of Detection
STOCL	Signal to Clutter Ratio

3.3.6.3 Description of Subroutine Logic and Processing

Since this subroutine (Figure 3.3-10) is employed to simulate a number of radars, airborne and ground, fixed and scanning, VHF through microwave, the engagement problem is solved by external subroutines. Thus, through appropriate processing and subroutine calls the MSM executive subroutine establishes when a target is in a position in which some detection probability exists. For microwave radars, for example, the MSM executive subroutine must determine that target sensor line of sight exists for both terrain and foliage considerations when it has been established that the target is within range of the sensor. The major inputs to this subroutine consist of the target description and the description through parameter specification of the radar in question. Target description must include target type, number of elements within range gate and target radial velocity.

The major structure of the simulation consists of computing the received signal due to target and clutter, the development of the clutter spectral width, and subsequently the signal to clutter or signal to thermal noise ratios, the latter being computed only if thermal noise is the principal limiter. The signals are computed from sensor/target range information, radar parameters, target parameters, atmospheric losses due to precipitation and foliage losses for the foliage penetration radars. The clutter in all systems includes the power scattered by the background while, for microwave radars, the power scattered by precipitation within the range gate being examined is also included in the clutter spectrum. The effects of precipitation on attenuation and backscatter were described in Reference 8. Essentially, straight line fits are defined by a set of coefficients termed FPREC1 and FPREC2 in the data sets of this subroutine. Atmospheric attenuation

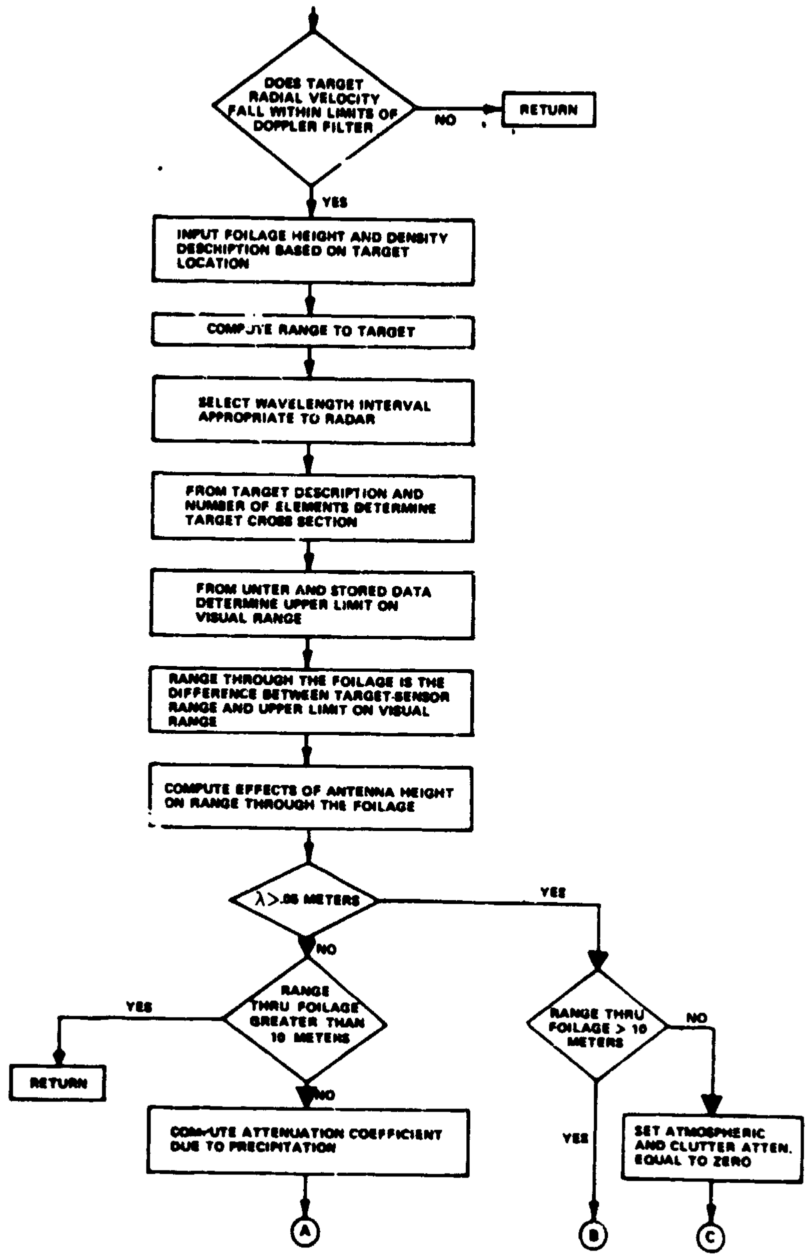


Figure 3.3-10 RADAR DESCRIPTIVE FLOW
(Sheet 1of 3)

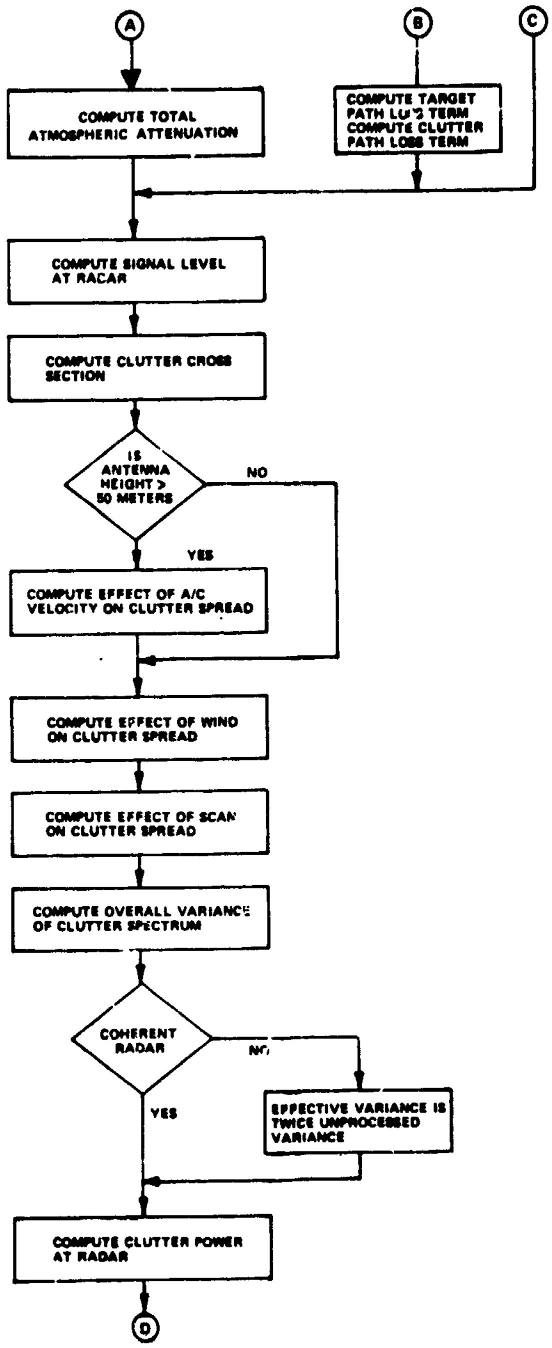


Figure 3.3-10 RADAR DESCRIPTIVE FLOW
(Sheet 2 of 3)

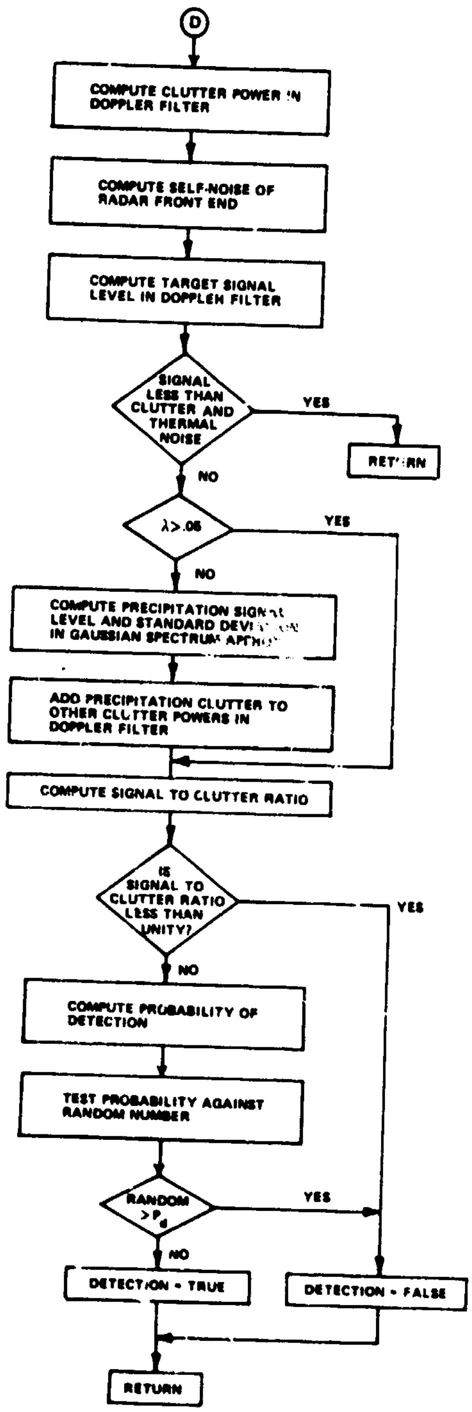


Figure 3.3-10 RADAR DESCRIPTIVE FLOW
(Sheet 3 of 3)

also taken from Reference 9 is described in the data set as ATATEN while background clutter cross section as a function of wavelength is given as BSECTN (Reference 10). Other data statements contain information on troop, vehicle and boat cross sections, XSCMAN, XSCVEH and XSCBOT. All target heights are taken to be 1.5 meters above the earth or water surface. It is also noted that a cleared area defined by a circle of 500 meters radius and centered on the radar is considered to apply in all cases, i.e., DCLEAR = 500. The actual line of sight and foliage masking determinations are made in other routines which proceed this routine in the calling sequence.

On entering the subroutine a check is first made to determine if the target radial velocity is adequate for detection. The doppler filter is assumed to be of bandpass type with a high pass section having a cutoff characteristic of 24 dB per octave and a low pass section with a slope of 12 dB per octave, as seen in Figure 3.3-11.

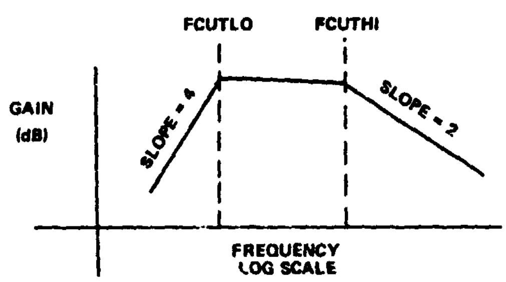


Figure 3.3-11 DOPPLER FILTER CHARACTERISTICS

If the target doppler, determined from

$$\text{doppler frequency} = \frac{2(\text{Radial Velocity})}{\text{Wave Length}}$$

is less than 1/4 of the lower doppler filter corner frequency or greater than four times the upper corner frequency the target will not be detected and return is made to the executive subroutine.

If the target velocity satisfies the requirements, the terrain features are inputed and the foliage index (IFOL) based on the UNTER table value of vegetation at the target is generated. IFOL = 1 means that heavy foliage will lie between the target and the radar and IFOL = 0 means no foliage. Following computation of geometrical factors slant range (RANGE), 4th power of range (RANGE4) and sine of radar azimuth angle (SINRAA), the wavelength index I is developed from the input value of wavelength, RAMBDA, and the stored table of wavelength values, TLAM. The index I is employed to select proper values of precipitation factors (FPREC1, FPREC2), atmospheric extinction (ATATEN) and clutter cross section (BSECTN). Then the index for precipitation (IPRECP) is generated from the ATMENV value of precipitation code.

The target cross section to be employed will be a function of target (ITGTTP) and target characteristics (KSTRNG), and on the number of elements within the range gate. A random variation is included to account for fluctuation and vector addition of randomly spaced target elements. (It can be shown that with two elements in the range gate, cross section will range between zero and four times the cross section of a single element, a cosine distribution about two times the cross section of a single element.) Then it is noted that if the height of the antenna, HGTANT, is greater than 50 meters above the surface, the radar is to be considered airborne.

In the computation of range through foliage a distinction is made between ground-based and airborne radars. For ground-based systems range through foliage (RANGEF) is given by

RANGEF =(Slant Range)-(Ground-to-Ground Visibility)

where ground-to-ground visibility will have a minimum value of 500 meters as specified by assumed clear distance to foliage. If the radar is airborne the depth through foliage is determined from similar triangles relating the height of aircraft/height of foliage ratio to the corresponding ratio on range. For airborne antennas, the assumption on clear distance to foliage from radar does not apply.

Next the effects of precipitation on attenuation are considered. If the radar wavelength is greater than 0.05 meters, precipitation effects are negligible and a bypass around the computations is taken. If for wavelengths less than 0.05 meters range through foliage should be greater than ten meters, then the target is obscured by foliage; all further computations are bypassed and a return to executive made. For those cases in which bypasses are not taken (wavelength less than 0.05 meters and range through foliage less than ten meters) attenuation due to precipitation (PATEN) is computed making use of one of three expressions depending on the value of the precipitation index (IPRECP) described above. Drawing the appropriate atmospheric loss factor (ATATEN) from the stored data set through use of index I, the total attenuation (TATATN) is computed. At this point, attenuation for the clutter return is also set equal to TATATN and is maintained at that value unless depth of foliage is greater than ten meters for the low frequency radars (wavelength greater than 0.05 meters).

If the latter conditions apply, a set of empirical equations are employed to determine the loss over free space for the target and the clutter. Since the clutter signal will be derived from the foliage top surface while the target is embedded in the foliage, the target path loss will be greater than that for clutter as may be seen in the relations:

$$\text{Attenuation} = \frac{41.3 \log_{10}(\text{Range Through Foliage}) - 0.12192(\text{Antenna Alt} + \text{Tgt Hgt})}{\text{Range}}$$

$$\text{Clutter Attenuation} = \frac{41.3 \log_{10}(\text{Range Through Foliage}) - 0.12192(\text{Antenna Alt} + \text{Foliage Hgt})}{\text{Range}}$$

However, if IFOL = 0, that is target not in foliage (although foliage lies between radar and target), then the two losses are set to be equal. These equations were developed from Reference 11.

Next the signal power at the receiver (DBSIG) is computed in the expression:

$$\text{DBSIG} = 10 \log_{10} \left(\frac{\text{RADCAR} \cdot \text{Target Cross Section}}{\text{Range}^4}\right) - (\text{Attenuation})(\text{Range})$$

In this expression the label RADCAR describes the radar characteristic, the combination of those parameters which do not change for a specific radar, namely:

$$\frac{P_T \, G_T \, G_R \, \lambda^2 \, L_T \, L_R}{(4\pi)^3}$$

where P_T is the peak power (or average power times compression factor for correlation type radars), G_T and G_R are the transmitter and receiver antenna gains and L_T and L_R are the transmitter and receiver losses. If the signal level computed is less than the thermal noise power in the doppler filter, FNKTB, the signal cannot be detected and return to the executive subroutine follows. FNKTB is found by multiplying the radar receiver thermal noise power by the ratio of the doppler filter width to the IF bandwidth and converting the result to decibels.

The clutter power in the doppler filter is determined by assuming that the clutter power spectrum has a gaussian shape with mean value at 0.0 Hz and a standard deviation which is a function of the several variables: platform motion, scan motion, wind speed, and radar instability. These variables are combined to produce sigma (SIGTOT) in one of two ways. For a coherent radar the expression employed is:

$$\text{SIGTOT} = \sqrt{\text{Sum of Variances of Individual Sigmas}}$$

while for an incoherent system, SIGTOT is given by:

$$\text{SIGTOT} = \sqrt{2(\text{Sum of Variances of Individual Sigmas})}$$

This approach follows the analysis of clutter spectra as given in Reference 12.

After computing the clutter power at the receiver and making use of the overall standard deviation of the clutter power the amount of power within the doppler filter (FIPCL) is computed by the expression:

$$\text{FIPCL} = \frac{0.5(\text{Rcvr Clutter})\left[\text{ERFC}\left(\frac{\text{Lower Filter Corner}}{\sqrt{2}\ \text{Std. Dev.}}\right) - \text{ERFC}\left(\frac{\text{Upper Filter Corner}}{\sqrt{2}\ \text{Std. Dev.}}\right)\right]}{\text{Clutter Improvement Factor}}$$

where ERFC is the complementary error function of the terms within parentheses. It is seen that the complementary error function is employed to determine the difference in areas in the tail of the gaussian spectrum, that applies to the value of sigma and the cutoff points which are defined by the corner frequencies of the doppler filter as shown in Figure 3.3-12. The parameter for clutter improvement factor (CLIMFR) applies to special processors of the KALMAS type as used in the CS-II radar. Clutter and precipitation (PRIMFR) factors are introduced to account for the improved performance of the KALMAS filter over the standard doppler method. The only known radar using this technique at the present time is the CS-II. A three decibel improvement in signal to clutter ratio over the standard doppler method is assumed. If the signal power exceeds the sum of the clutter and thermal components of noise in the doppler filter, computation continues.

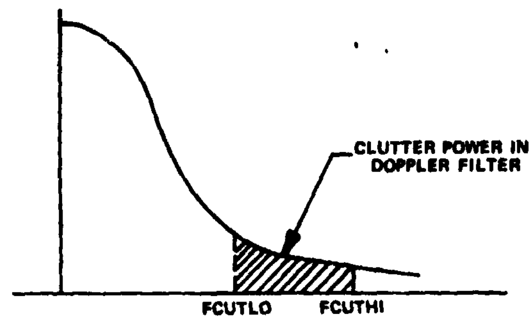


Figure 3.3-12 EFFECT OF DOPPLER FILTER ON CLUTTER POWER

To this point the contribution of precipitation return from precipitation within the range gate has been neglected but must be considered since it adds appreciable clutter noise in the microwave radar cases. If the wavelength is less than 0.05 meters, the precipitation cross section per unit volume, the total precipitation cross section and the signal level at the receiver due to precipitation are computed by one of three sets of expressions depending on the index of precipitation (IPRECP). The standard deviation for precipitation clutter power (SIGPRE) is taken to be

$$\text{SIGPRE} = \left(\frac{410}{\text{Wave Length}}\right)(1 \times 10^{-2})$$

from Reference 12. The overall variance on clutter power is then modified to include the precipitation component. The amount of precipitation clutter power found within the doppler filter is then computed again making use of the complementary error functions as described for clutter power above. This expression includes the use of the parameter PRIMFR which serves the same purpose as CLIMFR but for precipitation. In all likelihood CLIMFR and PRIMFR are the same for a radar. Note that if precipitation power is not defined (e.g., wavelength greater than 0.05 meters) or for no precipitation, the filtered precipitation power is set to 10^{-20} a value so low as to be of no consequence but adequate to avoid computer problems.

The signal to clutter ratio (STOCL) is then formed as

STOCL = Target Signal Level -10 $\log_{10}$(Filtered Precipitation Power + Filtered Clutter Power + Filtered Thermal Noise Level)

The probability of detection (PD) is then determined from the input value of probability of false alarm (PFA) and STOCL computed above as

$$PD = PFA^n$$
$$\text{where } n = \frac{1}{1 + 10^r} \quad \text{and } r = \frac{STOCL}{10}$$

Here PFA simply serves as a form of gain control by which the single pulse false alarm rate is set. The operator will not respond to single pulse events since these will be lost in the filtering provided. Only when PFA becomes quite large, 10^{-3} to 10^{-2}, will high noise false alarm rates be encountered. The actual false alarm rates for MTI radars are difficult to establish without careful consideration of operator performance.

The probability of detection is then tested against a random number. If the random number is greater than probability of detection, the logical decision on detection is LDET = False, but if the random number is less than probability of detection, the detection event is declared to be true.

One further comment is considered to be important. In the development of the standard deviations for clutter power, a value is assigned to the radar instability dependent factor (SIGSTB).* It is recognized that even under ideal conditions of stationary radar and near zero winds (conditions that reduce aircraft velocity clutter, scan clutter and wind clutter standard deviations to zero) there still remains an applicable clutter spectrum spread which can only be associated with the radar itself. However, values of SIGSTB are not published so this value has been selected by the designers to provide the range performance that matches field experience. The values of SIGSTB so derived were found to be consistent with radar experience, but should be substantiated by further investigation of source data.

3.3.7 Subroutine THERML

3.3.7.1 Purpose

This subroutine is included to provide simulation of sensors in which thermal imaging is employed to provide a target detection capability. Typical sensors include the hand-held thermal viewers and FLIR type equipments operating in the three to five micron region.

*Using the radar performance curves of Reference 13, the value of SIGSTB was adjusted until the model results agreed with the actual performance. The values obtained are consistent with estimates reported in Ref. 9 and Ref. 13.

3.3.7.2 Glossary of Inputs, Computed Values and Outputs

Input Values

FOCALL	Focal Length of the Optical System (Millimeters)
FNUMBR	F/Number, Focal Length to Diameter Ratio
IDSNSR	Identity Number of Sensor
ITGTTP	Target Type Index
IUT	Index on Unit Terrain
KSTRNG	Subindex on Target Type
RESOL	Resolution of Optical System (Radians)
TEMPEV	Temperature of Environment (Centigrade)
XSENS	Sensor X Coordinate
XTGT	Target X Coordinate
YSENS	Sensor Y Coordinate
YTGT	Target Y Coordinate

Labelled Common Inputed Values

ANEP	Noise Equivalent Power of Detector (Watts)
BANDTH	Sensor System Bandwidth (Hz)
HUMDTY	Relative Humidity (Percent)
H2ODEN	Atmospheric Water Content (Grams/cc)
IPCODE	Index on Precipitation
IPRINT	Output Data Device Designator = 6
ITIME	Game Running Time
LDUMP	True = Intermediate Calculations Printed, False = No Print
PRATE	Precipitation Rate (mm/hr)
STEFK	Stefan-Boltzmann Constant
XMNDEV	Optical System Transmission Factor

Internally Stored Designer Input Values

PIO4	PI/4 = 0.7854
SKYCON	Radiance of 250 K Atmosphere (Watts/M. Sq./Sterad)
TLENBT	Target (Boat) Length (Meters)
TLENMN	Target (Man) Length (Meters)
TLENVH	Target (Vehicle) Length (Meters)
TSIZBT	Target (Boat) Width (Meters)
TSIZMN	Target (Man) Width (Meters)
TSIZVH	Target (Vehicle) Width (Meters)

Computed Values

BACCON	Attenuated Radiance of Background (Watts/Square Meters/SR)
BACFAC	Fraction of Background Radiance in 3.5 to 5 Micron Interval
BAKCON	Unattenuated Radiance of Background (Watts/Square Meters/SR)
BAKGND	Power in Element Viewing Background (Watts)
BAREA	Background Area in Resolution Element Containing Target
DUM	Dummy Argument
ELEMTS	Number of Sensor Elemental Fields Subtended by Target
FACTOR	Intermediate Variable
FAREA	Area of Target Defined by Optical Resolution
FDIMEN	Diameter of Area Subtended by Optics at Target
FOGXMN	Fractional Transmittance Through Fog

* Designer values are given in program listing.

Computed Values (continued)

FOLXMN	Fractional Transmittance Through Foliage
GFACT	Solid Angle Subtended by Receiver Times Losses
H2OPRE	Precipitable Water (mm) In Path (Range)
H2OXMN	Fractional Transmittance Through Atmosphere
PRATE1	Precipitation Effect on Target Temperature
RADBAK	Radiant Intensity of Background in Target Resolved Element
RADSKY	Radiant Intensity of Atmosphere in Single Resolved Area
RADFLD	Radiant Intensity of Background (Watts/SR)
RADTAR	Radiant Intensity of Target
RANGE	Sensor Target Range (Meters)
SAREA	Area of Sensor Limiting Aperture (Meters)
SIGNAL	Power Incident on Element Containing Target
TARCON	Radiance of Target (3.5 to 5 Microns) (Watts/Square Meter/SR)
TAREA	Area of Target Within Resolution Cell
TARFAC	Fraction of Target Radiance in Sensor Bandwidth
TARLEN	Length of Target (Meters)
TARSIZ	Width of Target (Meters)
TEMAMK	Temperature of Environment (Kelvin)
TEMPTG	Target Temperature
TOTXMN	Total Fractional Transmittance (Fraction)
XNEP	Noise Power in System Bandwidth (Detector)

Output Values

LDET	Detection Decision
PDET	Probability of Detection
TRAST	Apparent Contrast at Detector Elements

3.3.7.3 Description of Subroutine Logic and Processing

The thermal imaging sensors are generally constructed in the following way. An array of detectors of the order of 50 or more are emplaced in an optical system so that the instantaneous field of view of each single detector is of the order of 0.2 to 2.0 milliradians. A scanning mechanism is provided so that the fields of view are swept across the total image plane at the rate of the order of 15 times per second. The total image plane will subtend angles of the order of 6° vertical by 12° horizontal at the sensor. Thus each instantaneous field is swept repetitively across the larger field of view defined by the 6° or 12° angle. In this scanning process the detectors respond to the energy incident lying between three and five microns.

In this simulation the following steps are carried out:

a. From target size and range, the angle subtended by the target at the sensor and hence the number of elemental fields of view containing the target are computed.

b. The powers received by an element field containing target and by an element containing background are computed from which apparent contrast at the sensor is computed.

c. From the number of elements containing the target and the apparent contrast, the probability of detection is determined.

d. Atmospheric effects on attenuation of 3 to 5 micron radiation are included in determination of target and background received powers.

e. The target and background radiances are limited to those derived from these sources acting as black body radiators at the temperatures assigned in this subroutine. Effects of solar or moonlight direct reflectance are not included. Since the crossover between solar and black body radiation as the predominant source of radiance occurs at three microns, this assumption is considered adequate. Since these devices are employed generally under night or foliage obscuration conditions, the assumption is further supported. However, actual measurements should be made to support this assumption or provide the data base for required modifications.

The data required to carry out the computations of this subroutine are provided through the labelled common areas SENSOR, ATMENV, BASICT, UNTER, CONST, and SENVAR, through the argument list of the calling statement, and through the set of data statements contained within this subroutine. That data relates particularly to the length and minimum width, TLEN-- and TSIZ-- respectively of the target elements.

On entering this subroutine (Figure 3.3-13) the target element length and minimum width are selected from the stored data based on the target descriptors (ITGTTP and KSTRNG) supplied through the subroutine argument list. These variables are labelled TARLEN and TARSIZ. Aircraft and artillery type targets cause a branch to the RETURN statement since they are not appropriate targets for the equipment considered. (The artillery piece itself was assumed not to be a target.*)

In general the equipments simulated have some small foliage penetration capability, that is, targets masked by small depths of foliage can be detected. Because of the coarseness of the unit terrain data (100 meter increments) depth of foliage masking a target will generally be excessive based only on target and sensor position and unit terrain data. Thus, it is assumed that line of sight exists when this subroutine is called and the foliage transmission factor (FOLXMN) is set equal to unity. Program modification to include estimate of foliage depth between target and sensor should be considered.

The other factors which contribute to attenuation of radiant energy in the three to five micron region are precipitable water in the sensor target path and scattering due to fog. Attenuation as a function of precipitable millimeters of water and wavelength have been extensively reported in tabular data form as for example in Reference 7. The expression relating attenuation

*Artillery pieces were not included in the initial list of targets considered. It and other targets can be included through proper modifications of the program.

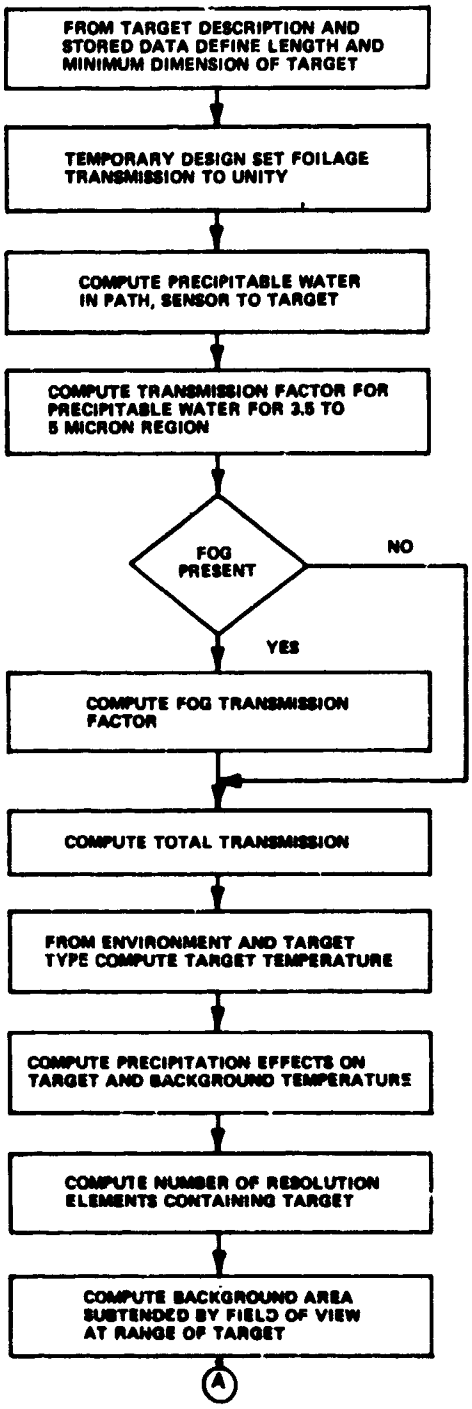


Figure 3.3-13 THERML MACROFLOW
(Sheet 1 of 2)

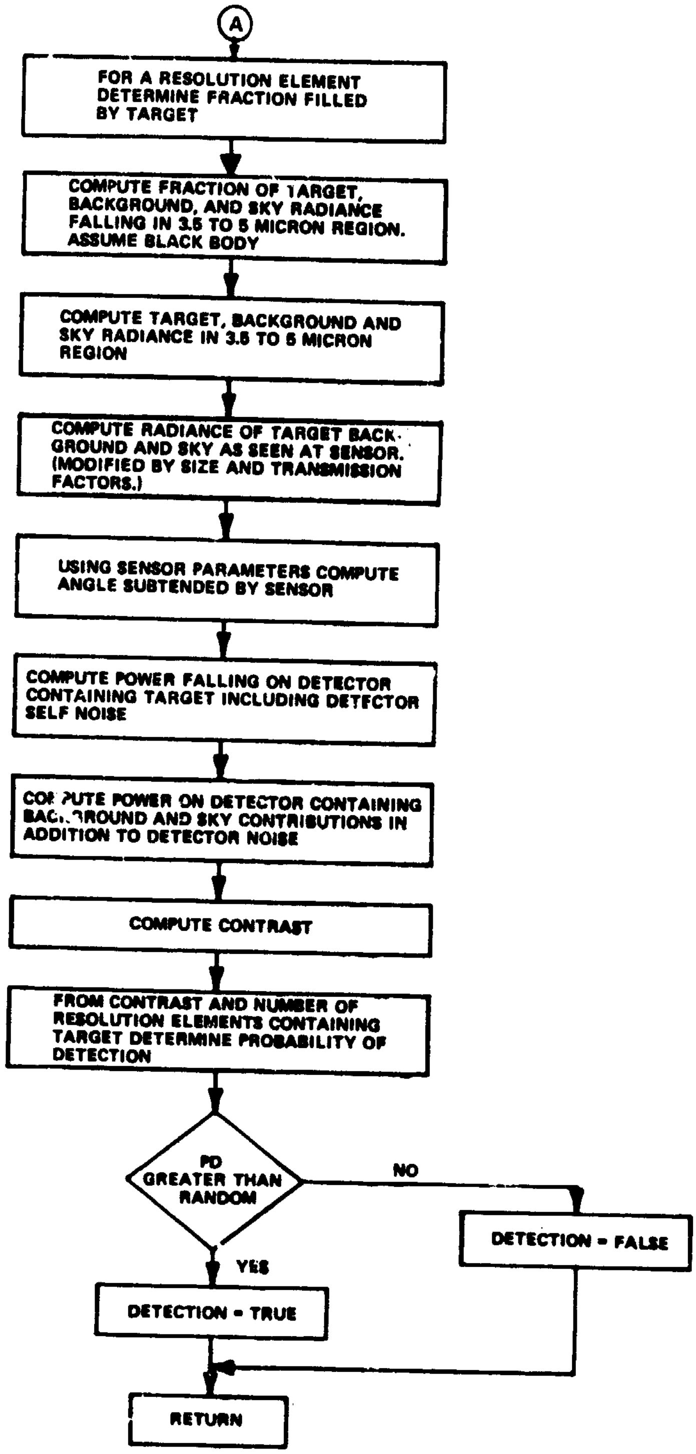


Figure 3.3-13 THERML MACROFLOW
(Sheet 2 of 2)

to precipitable water used in this simulation was developed by (1) averaging transmission over the 3.5 to 5 micron interval for the values of 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500 and 1000 millimeters of precipitable water. Then (2) a square law curve was fitted to average transmission versus precipitable millimeter values giving the expression:

$$\text{Fractional Transmittance Through Atmosphere} = [0.922 - \log_{10}(\text{H2ODEN})(1000)(\text{RANGE})][0.096 + 0.05211 \log_{10}(\text{H2ODEN})(1000)(\text{RANGE})]$$

where H2ODEN = Atmospheric Water Content in grams/cm^3 as taken from ATMENV table

To find the number of precipitable millimeters over the sensor/target range, multiply by 1000 to find precipitable millimeters/meter and then range to obtain millimeters.

The fog transmission factor was developed from observations relating the number of particles per cubic centimeter to the amount of water contained within the same volume. The amount of water in the fog fraction is found to be at least two orders of magnitude smaller than the total water content. It is reported in Reference 7 (page 161) that with 200 particles of five micron radius, transmission is reduced to a low level of the order of one percent in a 100 meter path. Using the equation:

$$T = e^{-\delta x}$$

it is found that $T = e^{-3.1} = 0.045$ when the total volume of water in the path is that for 200 particles/cubic centimeters and 100 meters of path. This result is approximated closely by the expression:

$$T = e^{-(\text{H2ODEN})(100)(1000)}$$

for values of H2ODEN near saturation, for example 30×10^{-6} grams/cubic centimeter. For that value and 100 meters range the result:

$T = e^{-3}$ is obtained.

Since the exponent given above is precipitable water (H2OPRE) as computed in the subroutine, fog transmission is approximated as:

$$\text{Fractional Transmittance Through Fog} = e^{-\text{precipitable water}}$$

Next the target and background temperatures are defined. Since this development is identical to that of the PIRTG subroutine, (para 3.3.3.3), it will not be discussed further at this point. The environment temperature (TEMPEV) is that computed by use of the subroutine ENVIR.

To compute the power incident on a detector in a single resolution cell it is necessary to determine the fraction of the cell that is filled by target and that by background, fractions that are functions of target size and range. The area at the target's range subtended by the resolution element (FAREA) is given by

$$FAREA = \frac{\pi}{4} D_F^2$$

where D_F is the diameter of that area given by resolution angle in radians times range. Several cases are considered as shown in Fig. 3.3-14 (1) If the target area is greater than the field area (FAREA) and the minimum target dimension is greater than the field diameter (D_F) then the target will fill the resolution cell and the number of cells filled by the target is given by the ratio of target area to field area; (2) If the target area is smaller than the field area, only one element contains the target and target area is given as TAREA; (3) If the target length is greater than the field dimensions, the number of cells containing a fraction of target and background is determined. The background area (BAREA) in a resolution element is then given as

$$BAREA = FAREA - TAREA$$

and the areas so developed are treated versus black bodies at the temperatures computed.

In order to employ the Stefan Boltzmann equation which gives the radiance over the entire wavelength interval it is necessary to determine the fraction that is to be found in the 3.5 to 5 micron region at a given temperature. That fraction is found from the Planck integral to be a function of temperature for which the following expression was developed:

$$F_{3.5-5} = [5.72 - (0.0623 - 0.0016T)T]0.01$$

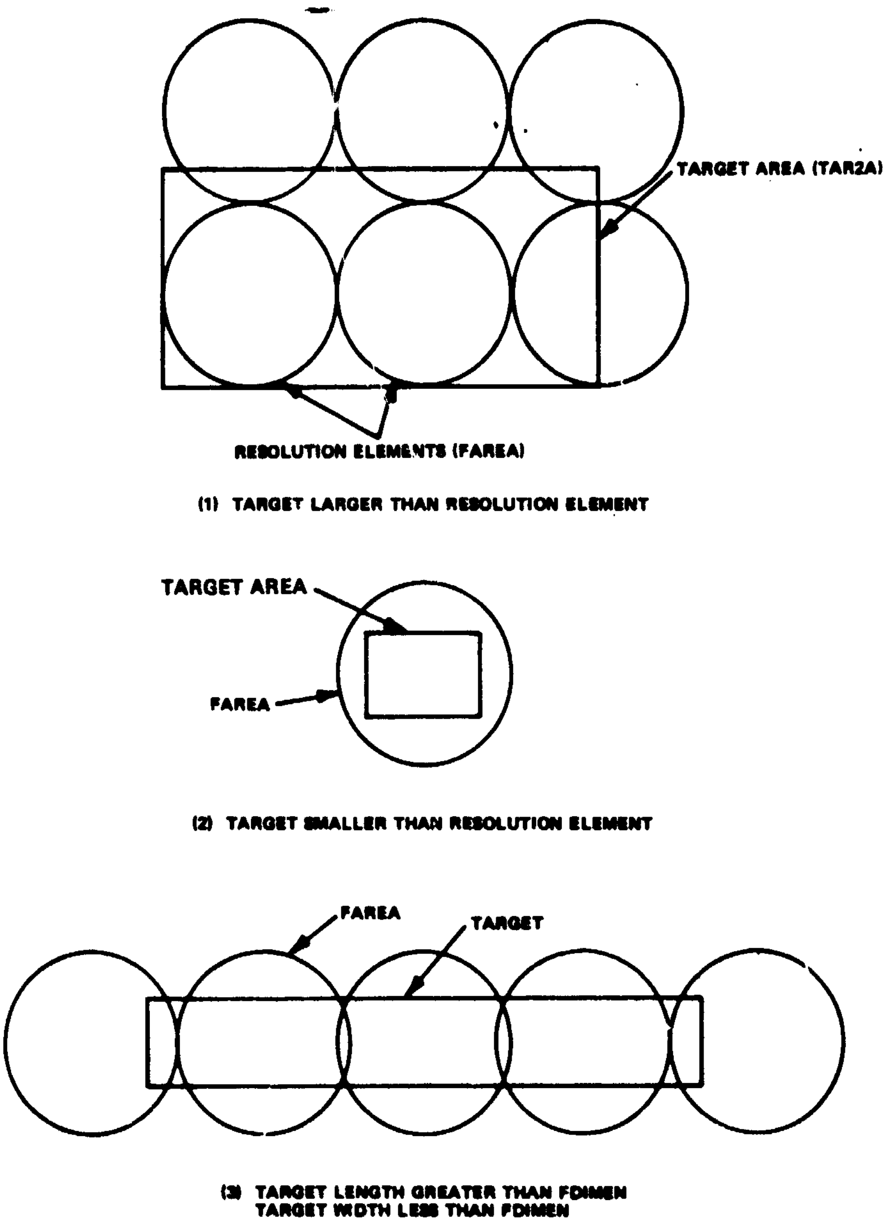


Figure 3.3-14 TARGET/RESOLUTION CELL GEOMETRY

Since the target (TEMPTG) and background (TEMAMK) temperature differ, this computation must be carried out for both fraction of target radiance (TARFAC) and fraction background radiance (BACFAC) in the subroutine. Then the radiant emittance of the target (TARCON) is given as:

$$TARCON = (TACFAC)(STEFK)(T_{TGT}^{4})$$

where STEFK is the Stefan Boltzmann constant divided by π, with a similar expression for the background.

The radiance of the target (RADTAR) is given by the radiant emittance and target area within a resolution element. The radiance is effectively reduced by atmospheric attenuation as expressed in the following:

Target Radiant Intensity = (Radiant Emittance)(Target Area)(Total Fractional Transmittance)
= (TARCON)(TAREA)(TOTXMN)

Again a similar expression applies for background except that background will not be reduced by foliage losses. For a resolution element containing only background, the radiance is found to be:

Background Radiant Intensity = (Background Unattenuation Radiance)(Atmosphere Fractional Transmittance)(Fog Fractional Transmittance)(Resolution Cell Area at Target)
= (BAKCON)(H2OXMN)(FOGXMN)(FAREA)

For the fractional component of background in a resolution element containing both target and background (RADBAK) the radiance is given by:

RADBAK = (Attenuated Background Radiance)[(FAREA-(TAREA)(1-Foliage Fractional Transmittance)]
= (BACCON)[(FAREA-(TAREA)(1-FOLXMN)]

Here it is noted that target area is reduced by and background area increased by foliage attenuation.

For conditions in which atmospheric attenuation is large (long ranges or high atmospheric water content or fog) the atmosphere itself will become an effective radiator, filling the field of view with a uniform background at a temperature of approximately 250° K (see Reference 7). Thus the sky radiance is also included as RADSKY.

Next the sensor parameters are employed to define the effective area of the sensor for each resolution element (SAREA) which is seen to be given by the expression:

$$SAREA = \left(\frac{FL}{FL/D}\right)^2 \left(\frac{\pi}{4} \times 10^{-6}\right) \text{ meters}^2$$

where focal length (FL) is given in millimeters and (Focal Length/Diameter(D)) is the f/number, FNUMBR. The angle subtended by SAREA at the target (GFACT) is given by:

$$GFACT = (XMNDEV)\left(\frac{SAREA}{RANGE^2}\right)$$

where XMNDEV, the device transmission factor, is included for convenience. Then the detector noise power is computed using the detector noise equivalent power (NEP) and the detector bandwidth. The bandwidth selected for checkout purposes was 2600 Hz determined from computing the number of resolution areas each resolution element would cover per second for a given resolution and overall field size, (6° x 12°).

The background power on a detector (BAKGND) is found to be:

BAKGND = (Background Radiant Intensity
+ Atmosphere Radiant Intensity) (GFACT) +
(Noise Power)
= (RADFLD + RADSKY) (GFACT) + XNEP

for a cell in which no target input is found. For a cell containing target the signal is found to be

SIGNAL = (RADTAR + RADBAK + RADSKY) (GFACT)
+ XNEP

From these two values the apparent contrast is then determined.

The probability of detection is argued to be a function of the apparent contrast (TRAST) and the number of resolution elements in which target radiance is contained by the following expression:

$$\text{Probability of Detection} = 1-e^{-(\text{No. Elements})(\text{TRAST})}$$

This expression produced detection results which were consistent with those of Reference 6. The final steps are then involved in the test of detection probability using a random number to produce a true or false conclusion.

3.3.8 Subroutine IMAGE

3.3.8.1 Purpose

This subroutine provides a simulation for imaging devices that operate in the 0.4 to 0.95 micron region. Equipment types simulated include passive night vision, low-light level TV, natural eyesight and binocular-aided vision. Illumination is provided by natural light due to sun, moon or skyglow and also by direct searchlight (aimed at the target), indirect searchlight (aimed at cloud base above target), and flares. The model of Reference 13, Carmonette IV considers many of the basic attributes required in the SAM simulation. The following discussion describes an extensive modification of that model.

3.3.8.2 Glossary of Inputs, Computed Values, and Outputs

Input Values

AINTNS	Intensity of Flare or Indirect Searchlight, Candlepower
AMAG	Magnification (Eye = 1), (7 x 50 Binoculars = 7)
ALPHA	Effective Lens Area of Natural Vision Sensors (Eyes, 0.5 Square Centimeter)(Binoculars, 33 Square Centimeters)
BWIDTH	Beamwidth of Sensor Assigned Searchlight (Radians)
CPOWER	Peak Candlepower of Sensor Assigned Searchlight
DEVCAL	Exponent Weighting Factor in Detection Probability Computation for Natural Vision Sensors (Eyes, 1.5) (Binoculars, 0.01)*
FLARHT	Height of Flare (Meters)
FNUMBR	Focal Length to Diameter Ratio
FOCALL	Focal Length of Optical System (Millimeters)
HTGAC	Height of Aircraft (Meters)
IAIR	Index on Sensor Usage (Negative Number = Airborne, 0 = Ground Moving Sensor, Positive Number = Ground Stationary Sensor)
IDSNSR	Identity Number of Sensor
ILXTRA	Index on External Illumination, 0 = No External Sources
ISERCH	Index, 0 = Natural Light, 1 = Searchlight, 2 = Searchlight with Pink Filter
ITGTTP	Index on Target Type
ITYPE	Index on Type of Electronic Aided Sensor(Daylight,0)(Night Vision, 1)
IUT	Index on Terrain Type

*Reported in private communication with N. W. Parsons of RAC to be an adjustment factor required to bring model and actual responses into agreement.

Input Values (continued)

IVISUL	Index on Sensor Class, Direct = 0, Electronic Aided = 1
KSTRNG	Subindex of Target Type Based on Target Formation for Personnel Targets, and Target Size for Vehicle and Boat Targets
MODE	Flare-Indirect Searchlight; 1 = Flare, 2 = Indirect Searchlight, 0.017 Rad. Beamwidth, 3 = Indirect Searchlight, 0.051 Rad Beamwidth, 4=Indirect Searchlight, 0.085 Rad Beamwidth, 5=Indirect Searchlight, 0.017 Rad Beamwidth, Pink Filter, 6=Indirect Searchlight, 0.051 Rad Beamwidth, Pink Filter, 7=Indirect Searchlight, 0.085 Rad Beamwidth, Pink Filter
RMAX	Maximum Detection Range (Meters)
TIMCON	Electronically Aided Sensor Time Constant, 0.1
XLITE	X Coordinate of Flare or Indirect Searchlight
XMTF	Area Under the Modulation Transfer Function Curve
XSENS	Sensor X Coordinate
XSRCH	X Coordinate of Sensor Assigned Searchlight
XTGT	Target X Coordinate
YLITE	Y Coordinate of Flare or Indirect Searchlight
YSENS	Sensor Y Coordinate
YSRCH	Y Coordinate of Sensor Assigned Searchlight
YTGT	Target Y Coordinate
NOTFLD	Number of Targets in Field of View

Labelled Common Inputed Values

ASID1	Amplitude Coefficient of Spectral Irradiance Due to Direct Sunlight or Moonlight (Watts/Square Meter)
ASID2	Amplitude Coefficient of Spectral Irradiance Due to Diffuse Sunlight or Moonlight (Watts/Square Meter)
ASID3	Amplitude Coefficient of Spectral Irradiance Due to Night Sky Glow (Watts/Square Meter)
CCOVER	Cloud Cover, Fractional
CEIL	Cloud Ceiling (Meters)
CCLL	Canopy Closure, Lower Limit (Percent)
CCUL	Canopy Closure, Upper Limit (Percent)
H2ODEN	Atmospheric Water Content (Grams/cc)
IBACK	Index on Background Type
IPCODE	Index on Precipitation
IPRINT	Output Data Device Designator = 6
ITIME	Game Running Time
LDUMP	True = Intermediate Calculations Printed, False = No Print
OPTXMN	Optical System Transmission Factor (Assumed Value = 0.8)
PI	Ratio of Circumference of Circle to Diameter (3.141593)
TCLOUD	Cloud Transmission Factor (Decimal Fraction)
VISIB	Meteorological Range (Meters)

Internally Stored Designer Input Values *

AGVISB	Air-to-Ground Visibility (Percent)
BEAMWD	Searchlight Beamwidth Keyed by Mode (Radians)
CAY3	Exponent Weighting Factor in Detection Computation for Electronic Aided Sensor (0.256)
FOTOPT	Spectral Weighting for Light Adapted Eye
RBACK1	Spectral Reflection Coefficient for Type 1 Background
RBACK2	Spectral Reflection Coefficient for Type 2 Background
RBACK3	Spectral Reflection Coefficient for Type 3 Background
RBACK4	Spectral Reflection Coefficient for Type 4 Background
RBACK5	Spectral Reflection Coefficient for Type 5 Background
RFOG	Height of Fog, (30 Meters)
RTGTBT	Spectral Reflection Coefficient for Boat Target
RTGTM1	Spectral Reflection Coefficient for Man 1 Target
RTGTM2	Spectral Reflection Coefficient for Man 2 Target
RTGTM3	Spectral Reflection Coefficient for Man 3 Target
RTGTVH	Spectral Reflection Coefficient for Vehicle Target
SCOPT	Spectral Weighting for Dark Adapted Eye
SEARCH	Spectral Distribution of Searchlight Power
SENSPH	Spectral Response of Extended S20 Photocathode
SID1	Spatial Irradiance Density Function for Direct Sunlight or Moonlight(Meter^{-1})
SID2	Spatial Irradiance Density Function for Diffuse Sunlight or Moonlight (Meter^{-1})
SID3	Spatial Irradiance Density Function for Night Sky Glow (Meter^{-1})
SQ2PI	Square Root of 2 x PI
SRVISB	Slant Range Visibility (Percent)
TAUO	Average Interval for Change in Cloud Cover
TSIZBT	Target Size (Minimum Dimension), Boat
TSIZMN	Target Size (Minimum Dimension), Man
TSIZVH	Target Size (Minimum Dimension), Vehicle

Computed Values

ALOSS	Loss Due to Scatter for Indirect Searchlight Mode
ANGLE	Minimum Resolvable Angle for Light Level and Contrast Available
AREASN	Area of Sensor (Alpha)
ATRANS	Atmospheric Loss for Searchlight to Cloud Path
ATRAST	Apparent Contrast, Target to Background, As Seen At Sensor
BKNOIS	Electronic Noise Component Due to Background And Sky
BTRANS	Atmospheric Extinction for Flare
CANDLE	Light Level Incident on Target and Background (Footcandles)
CAY1	Radiance of Sky Due to Scatter
CAY2	Fraction of Background Radiance Available at Sensor
CONST2	Area of Sensor Resolution Element x 16
CONTRA	Log Base 10 of Apparent Contrast
CPRIME	Fraction of Sky Clear of Clouds
DELRNG	Length of Volume Defined by Sensor and Slight Intersection
DELTAT	Time Since Last Call on Specific Sensor

*Designer Input Values are contained in the Program Listing

<u>Computed Values</u>

DELTAZ	Difference in Height Between Ceiling and Flare
DEVCON	Computational Variable with Appropriate DEVCAL Value
DEVMAG	Computational Variable with Appropriate AMAG Value
DUM	Dummy Argument
ECOMP	Component of Irradiance Due to Flare per 50 Micron Interval
EFACTR	Relative Flare Irradiance, Atmospheric and Geometric Losses Included
EFFECT	Effective Sensor Resolution
FACTOR	Number of Lines of Sensor Resolution Intercepted by Target
FCLOUD	Transmission Factor for Clouds
FNT	Float of Number of Target Elements (Not Field)
FNTR	Effective Number of Target Elements
FOLXMN	Transmission Factor of Light Through Vegetation Canopy
FSCAT	Light Scattering Function for Direct Searchlight
GAMMA	Extinction Coefficient (Meters)
I	Index on Wavelength Increment
IBACKP	Dummy Index Derived from IBACK
ICLASS	Type of Detector, 0=Natural, 1=Electronic Aided
ICLEAR	Computed Index on Clear or Cloud Shadow
INDX	Index Computed from Mode Referring to Indirect Searchlight Beamwidth
ITIMEP	Time of Previous Sensor Use
OFFSET	Length of Perpendicular from Sensor to Searchlight-Target Line
PCLEAR	Weighted Result of Cloud Cover Decay Computation
PROB	Probability of Change in Cloud Cover Condition
RADBAK	Radiant Intensity of Background
RADNCE	Irradiance of Horizontal Plane Due to Natural and Searchlight
RADSKY	Radiant Intensity of Sky
RADTAR	Radiant Intensity of Target
RANGE	Sensor to Target Range
RANGE1	Range, Searchlight to Target (Meters)
RANGE2	Range, Searchlight to Sensor (Meters)
RANGE4	Range, Flare to Target (Meters)
RANGE6	Range, Indirect Searchlight to Cloud Above Target (Meters)
REFLBF	Background Reflectance
REFLNF	Reflection Factor for Clouds, Indirect Searchlight
REFLTG	Target Reflectance
RELLUM	Weighting Function Due to SENSPH, FOTOPT, SCOPT, etc.
RESLEN	Length of Sensor Defined Resolution Element (millimeters)
RNGFOG	Range Through Fog
RNOISE	Receiver Noise Level
RSGRE	Square of Range, Sensor to Target (Square Meters)
R1SQRE	Square of Range, Searchlight to Target (Square Meters)
R2SQRE	Square of Range, Searchlight to Sensor (Square Meters)
R4SQRE	Square of Range, Flare to Target (Square Meters)
R6SQRE	Square of Range, Indirect Searchlight to Cloud Above Target
SCATTR	Atmospheric Scattering Function
SFACTR	Irradiance Loss Due to Atmosphere and Geometry
SIGNAL	Signal Level in Sensor
SGNOIS	Signal to Noise Ratio

Computed Values (continued)

SKYCUP	Fraction Coupling of Input Radiation to Atmosphere Scattered Light
SPCTRM	Spectral Distribution of Searchlight Power
TARSIZ	Minimum Dimension of Target
THETA	Angle Formed by Intersection of Sensor and Searchlight at Target
TLIGHT	Total Spectral Light, Natural Plus Searchlight
TRASTI	Inherent Contrast
VISANG	Log of Minimum Resolvable Angle
XMISSN	Fraction of Radient Intensity Available at Sensor
XMTC	Optical System Modulation Transfer Constant
YYY	Maximum Limit on Angle

Output Value

LDET	Detection Decision, True or False
PDET	Probability of Detection
RATIO	Ratio of Angle Subtended by Target to Minimum Resolvable Angle

3.3.8.3 Description of Subroutine Logic and Processing

Because of the fact that a broad range of sensors, sensor types, light sources, and methods of deployment are included in this model, this subroutine is the most extensive and complicated of the Systems Assessment Model sensor simulations. Sensors that may be employed are of natural vision types such as natural eyesight and binocular aided vision (IVISUL=0), or electronic aided types (IVISUL=1). This latter class is further subdivided into daylight (ITYPE=0) and night vision (ITYPE=1) devices. Further, the sensors may be airborne (IAIR= -Number), moving ground (IAIR=0), or stationary (IAIR= +Number).

Light levels are due to sunlight (ASID1), moonlight (ASID2), or sky glow (Starlight) (ASID3) or combinations of these levels. In addition light may be provided by direct searchlight (ISERCH=1), pink filtered direct searchlight (ISERCH=2), or by auxiliary sources (ILXTRA > 0) in which case the source may be a flare (MODE=1) or indirect searchlight (MODE=2 to 7, depending on filter type and beamwidth).

The subroutine is organized such that the light levels incident on target and background are computed. Using target and background reflectance data stored within the program, the radiant intensity of these two components and of the sky component are computed. The inherent contrast of the target as seen through the spectral response function of the sensor (SENSPH, FOTOPT, SCOPT) is determined and the degraded value (apparent contrast, degraded by scatter light and atmospheric attenuation) is determined. The size of the angle subtended by the target is compared with the minimum resolvable angle that can be observed by the sensor for the light level prevailing and from this comparison, the probability of detection is computed.

Under some conditions of natural and aided illumination, levels will be such that passive night vision devices would be saturated. In such a situation, the operator would most likely make use of natural vision. To indicate this course of action, under these conditions, the program causes a switch to be made to the natural vision routine and detection probability for the human observer is made. To indicate the fact that detection is due to natural vision when by input an electronic aided sight was employed, the ratio of angle subtended by the target to the minimum resolvable angle, a quantity that would always be positive is set to its negative for natural vision. Thus a key is provided to following routines to allow indication of the detection means. This parameter along with probability of detection, the decision on detection, and ratio, the parameter defined above, are subroutine output parameters.

On entering the subroutine (see Fig. 3.3-15) the index ICLASS is set to IVISUL. It will be seen later that the choice between natural vision and electronic aided devices is keyed on ICLASS, and this index may be set to 0 under high illumination conditions as noted in preceding paragraphs. Next the range to the target in the ground plane is computed and foliage transmission (FOLXMN) is set equal to unity. For ground-based sensors, the subroutine is

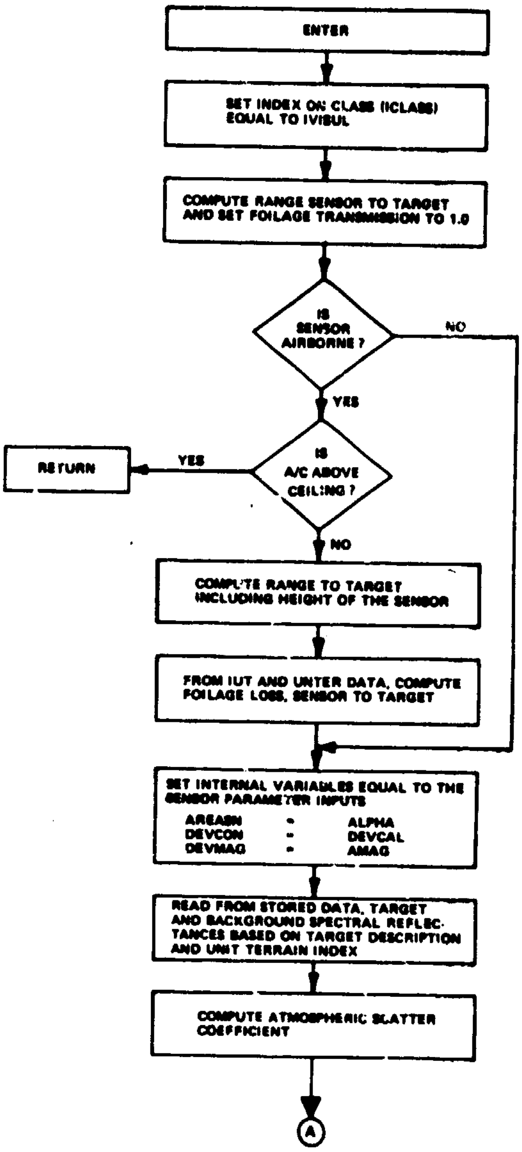


Figure 3.3-15 IMAGE SUBROUTINE MACROFLOW
(Sheet 1 of 7)

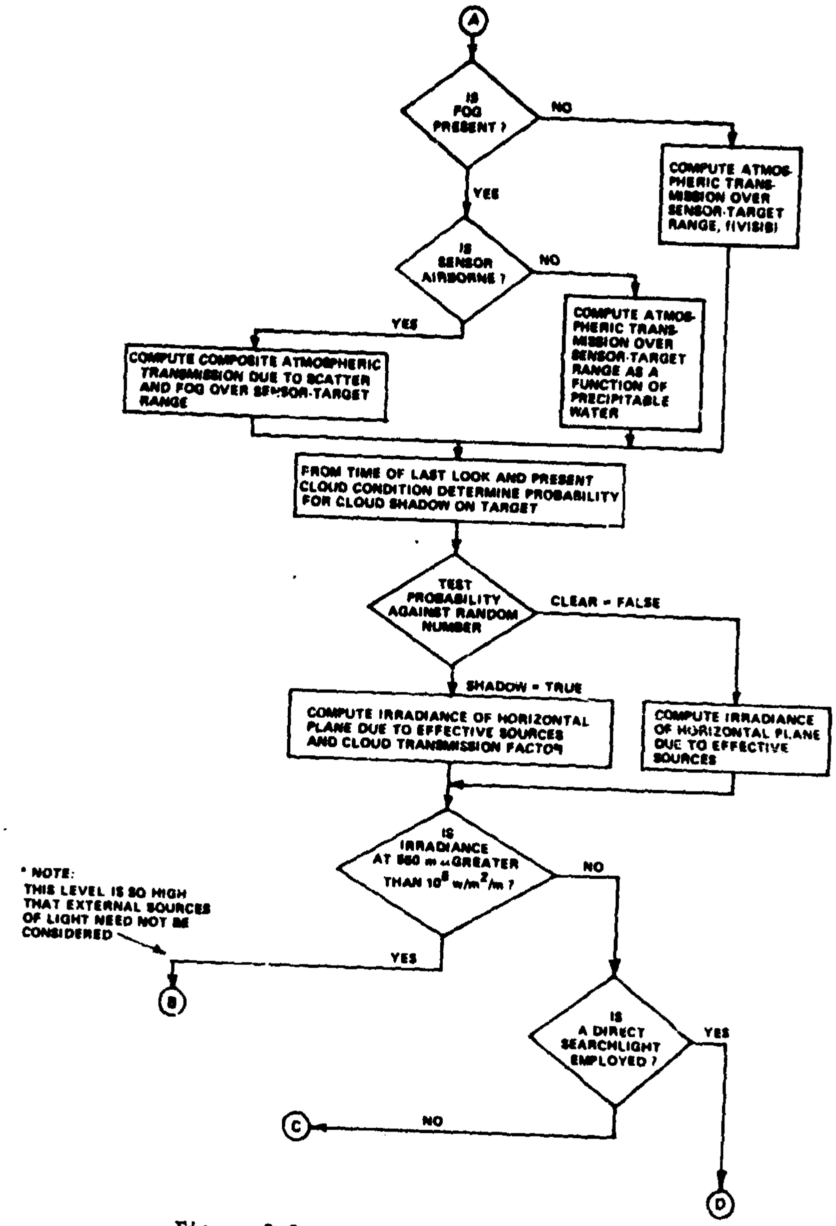


Figure 3.3-13 IMAGE SUBROUTINE MACROFLOW
(Sheet 2 of 7)

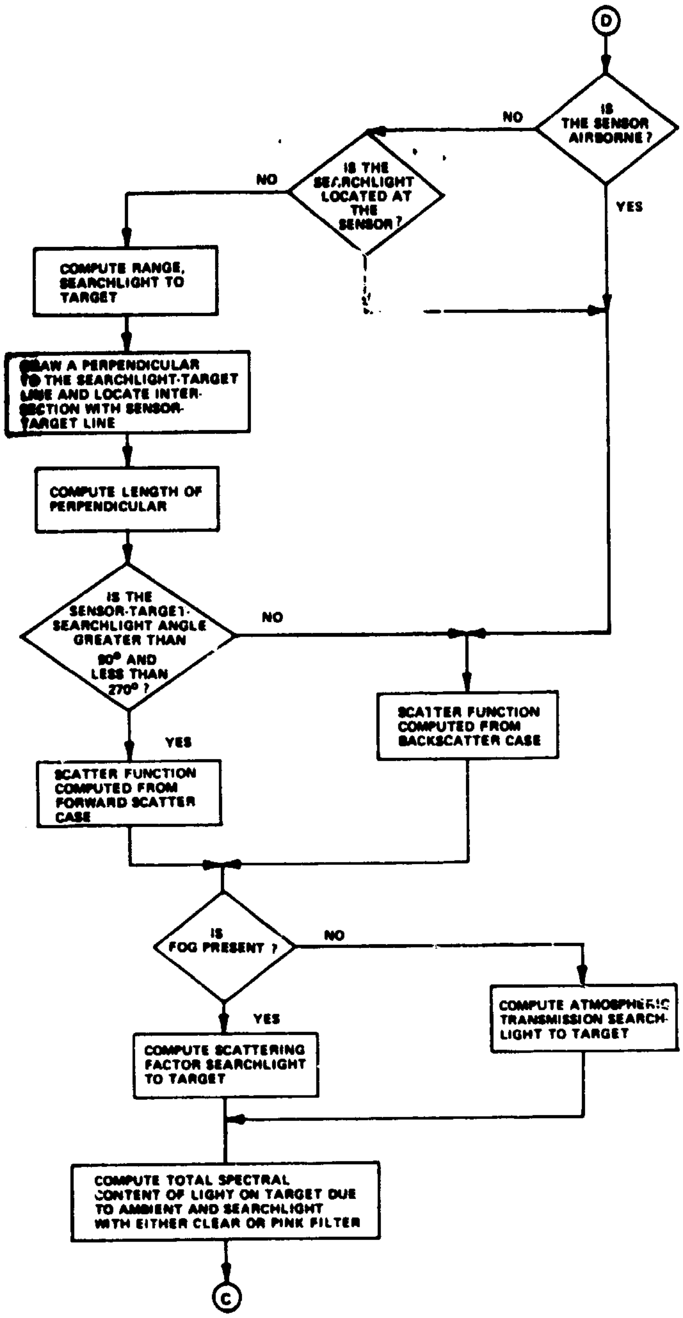


Figure 3.3-15 IMAGE SUBROUTINE MACROFLOW
(Sheet 3 of 7)

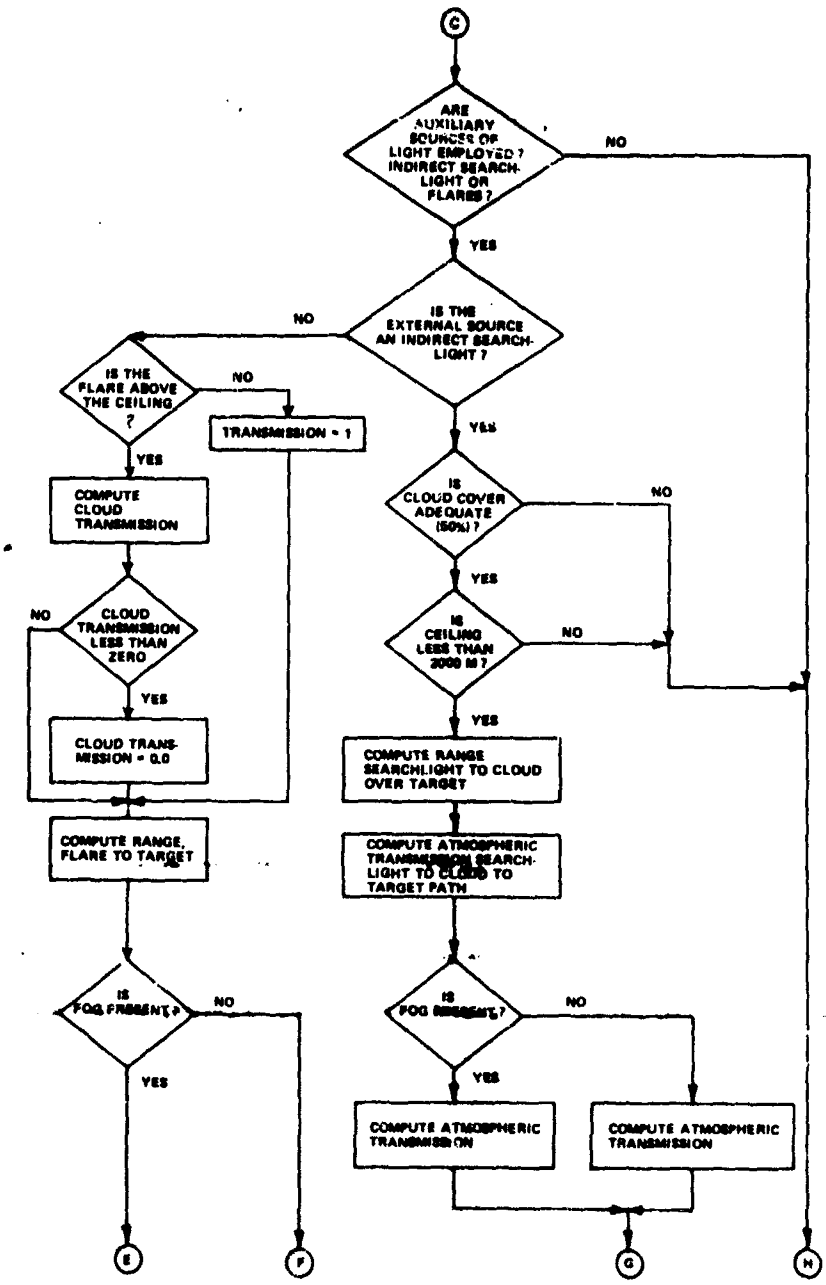


Figure 3.3-15 IMAGE SUBROUTINE MACROFLOW
(Sheet 4 of 7)

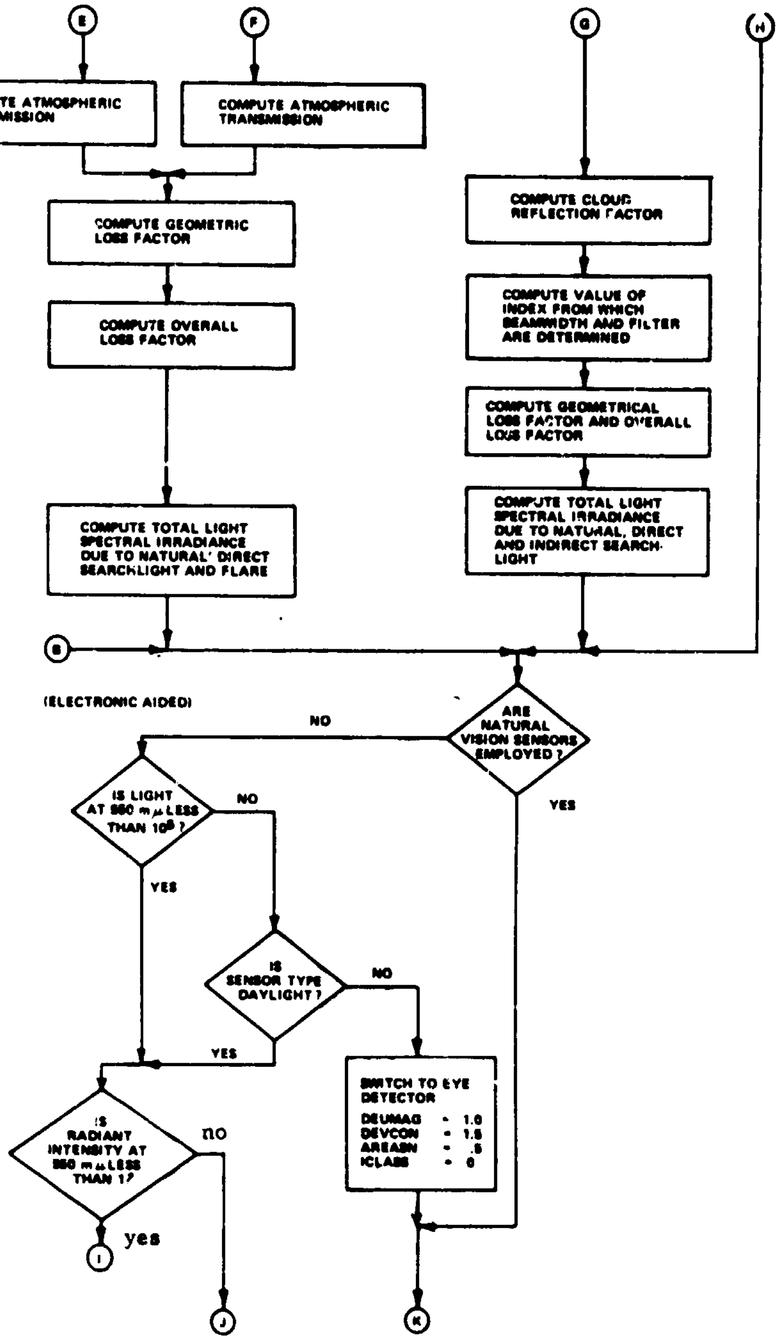


Figure 3.3-15 IMAGE SUBROUTINE MACROFLOW
(Sheet 5 of 7)

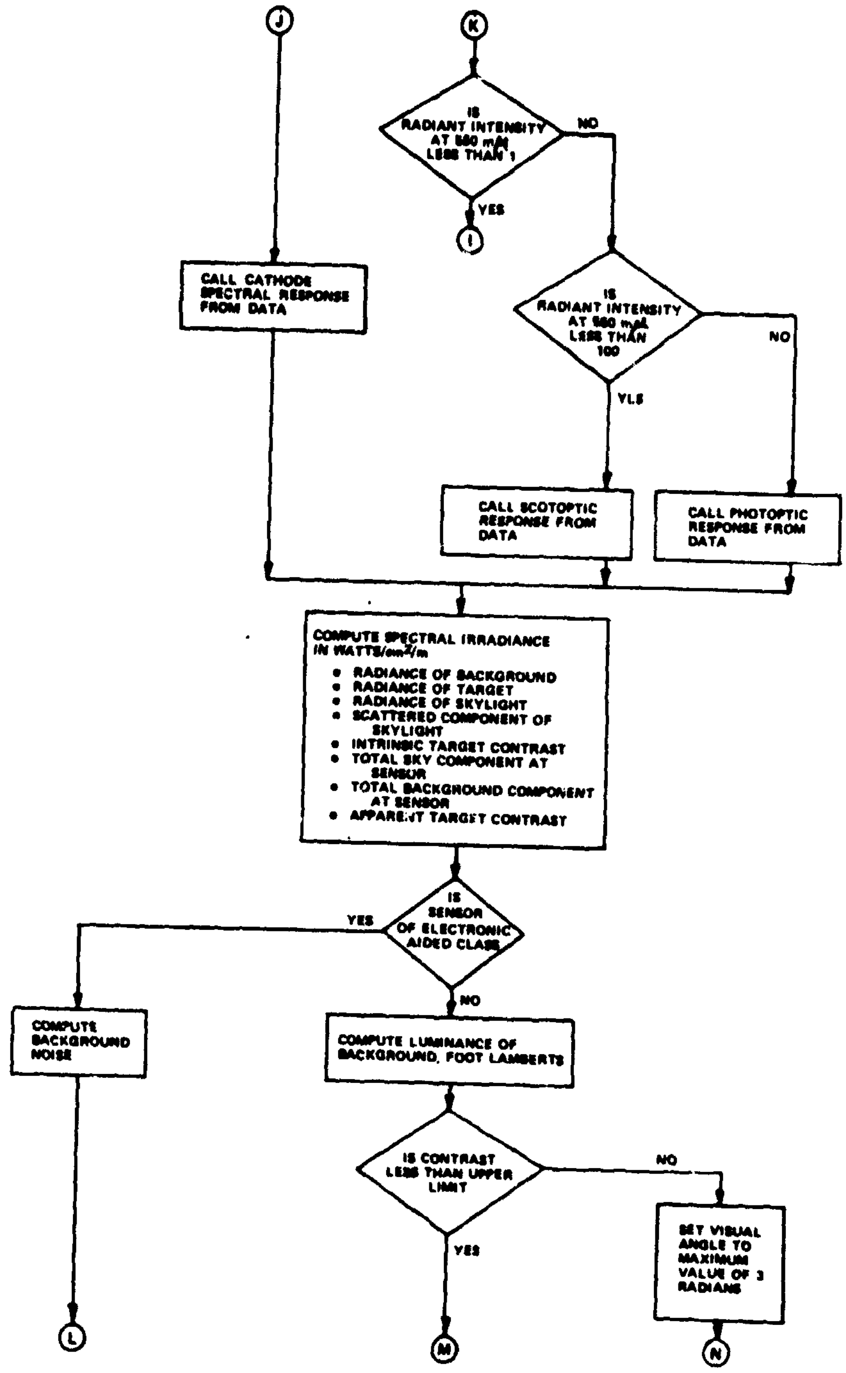


Figure 3.3-15 IMAGE SUBROUTINE MACROFLOW
(Sheet 6 of 7)

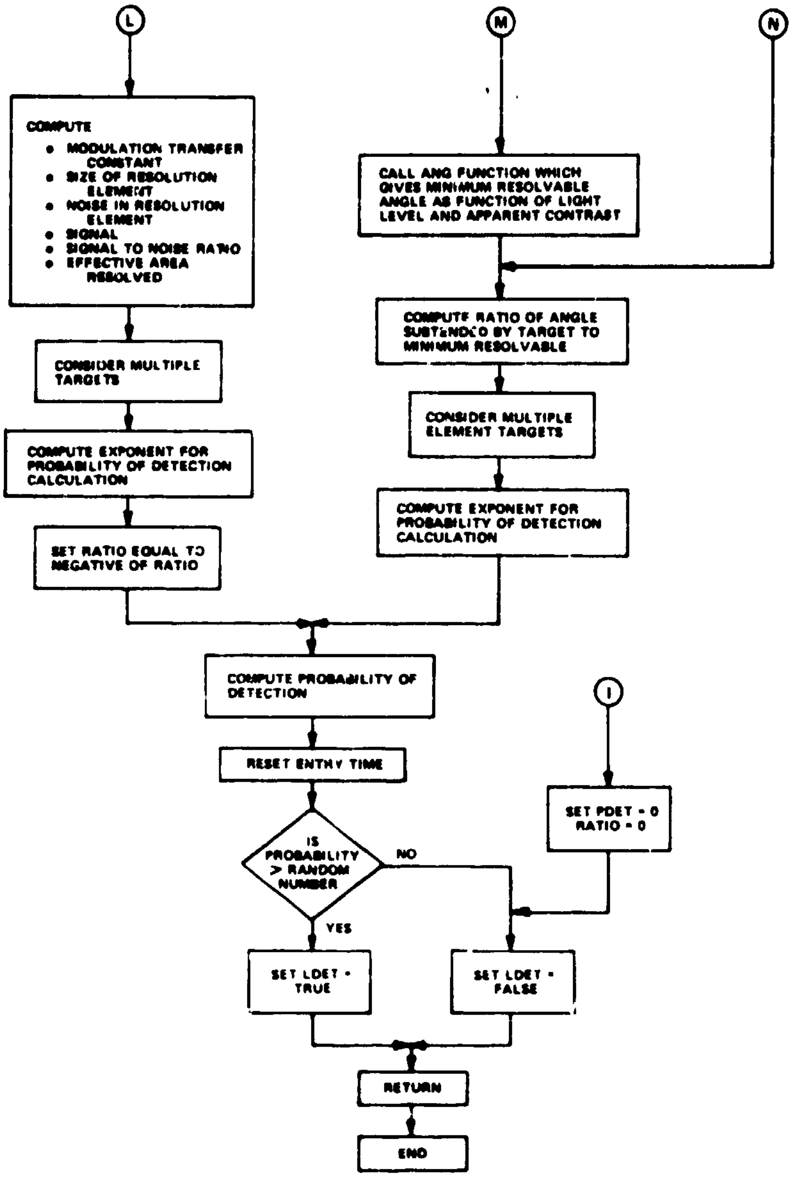


Figure 3.3-15 **IMAGE SUBROUTINE MACROFLOW (Sheet 7 of 7)**

called only if line of sight exists. If, however, the sensor is airborne (IAIR < 0) the slant range from sensor to target is computed and foliage transmission is computed making use of the upper and lower limits of foliage transmission (CHUL and CHLL) as derived from the unit terrain table. Should the sensor height for the airborne case be greater than the ceiling, no detections are allowed and an exit is made from the program. For ground or airborne sensors, the device parameters as input by the planner (sensor area, response factor, and magnification) are assigned to intermediate variables AREASN, DEVCON, and DEVMAG because these parameters may be set to new values should light level be above the threshold for use of night vision devices during simulation run.

Next the appropriate spectral reflectance characteristics and target size for the target are transferred into the active program based on the target descriptors, ITGTTP and KSTRNG. At this time only three spectral characteristics are provided: one for troop targets, one for vehicles, and one for river craft. Additional target spectral data may be introduced by expansion of the data set contained within the subroutine. Then the background spectral reflectance characteristic is transferred also, based on the background index (IBACK) contained in the unit terrain table description for the target location.

Atmospheric transmission factor for the sensor-target path is computed by the function, e^{-x}, where the form x depends on the situation. Several conditions must be considered including ground and airborne sensor situations and the presence of fog. The fog is assumed to be a uniform slab, 30 meters in vertical extent.

The amount of illumination due to natural sources at the ground will depend on cloud cover. In order to include some coherence in cloud cover from entry to entry into this subroutine particularly for those spaced closely in time, a brief set of statements is included that relates probability of cloud cover to percentage of cloud cover, time since last entry, and cloud cover conditions at the time of last entry. If, on test, it is concluded that clouds do not lie between target and source, one calculation for irradiance (RADNCE) is carried out using spectral irradiance amplitudes of light sources and wavelength weighting coefficients. For cloud cover the computation of irradiance includes fraction of cloud cover and cloud transmission factor considerations. The values of RADNCE, 12 in all, are given in units of watts/square meter/meter of spectral width.

External sources of illumination are next considered. If the spectral irradiance at 550 millimicrons is greater than 10^5 watts/square meter/meter,* **external sources are considered to be ineffective and the program progresses directly to detection by natural vision sensors. Sensor

* 10^5 watts/sq meter/meter = 10^{-8} watts/cm^2/millimicron

** Wavelength in units of meters is used to be consistent with the ATMENV subroutine (5.2.11) parameters are then modified by later statements to conform to those for detection by natural eyesight. If, however, the light level is less than 10^5, a test is made to determine if a direct searchlight is employed. By direct searchlight, we mean that the target is illuminated by direct rays of the searchlight and not by diffusely scattered light. If a direct searchlight is not employed the program bypasses to other program statements. Should a searchlight be employed, however, the following arguments are made. If the searchlight is located at the sensor, as will be the case for airborne systems, and as may be the case for ground systems, a minimum offset of ten meters between the sensor and searchlight is assumed. The offset is used in determining the length of the path common to both sensor and searchlight, i.e., path length over which backscatter light must be considered. The problem is shown in Figure 3.3-16.

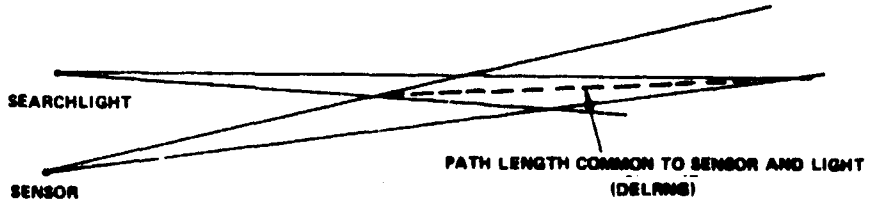


Figure 3.3-16 DEFINITION OF DELRNG

If, however, the direct searchlight is not colocated with the sensor. the geometry illustrated in Figure 3.3-17 must be solved where two possible positions are shown: (1) giving backscatter light while (2) produces forward scatter light from the aspect of the sensor.

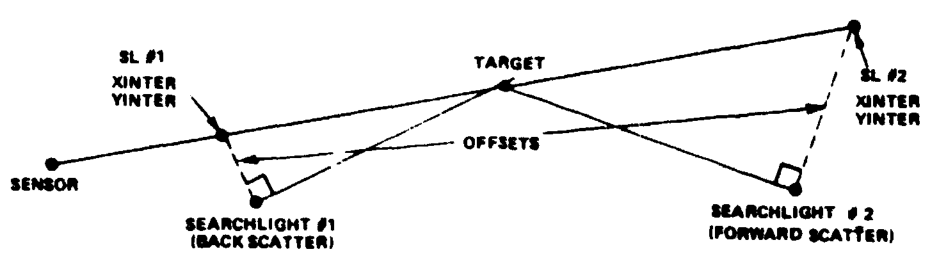


Figure 3.3-17 DEFINITION OF OFFSET AND SCATTERING FUNCTION

The dashed lines show the perpendiculars to the searchlight-target line of sight. The intersections of these lines with the sensor-target line or line extended produces the coordinates XINTER, YINTER. The lengths of the perpendiculars are taken to be the searchlight offset distances for computing the scatter length. The conclusion to this section is the determination of the scattering function (FSCAT):

FSCAT = 1.0 for backscatter case

or FSCAT = 1 + 2 cosine (angle between SL and sensor at target) for forward scatter

Next the atmospheric scattering factor is computed for the searchlight to target path, fog being taken into consideration in the same way as for the sensor-target path. The irradiance at the target due to the searchlight (SFACTOR) is then computed as:

$$\text{SFACTOR} = \frac{(\text{Foliage Transmission})\, e^{-\gamma \cdot \text{Range}_{SL \rightarrow TGT}}}{\text{Range}^2_{SL \rightarrow TGT}}$$

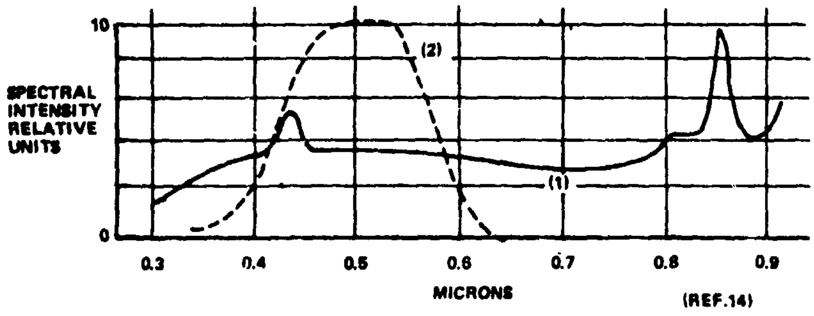
where γ = atmospheric scattering function (if clear)

or γ = atmospheric scattering function + 1000 (water content) (if in fog)

The total light level in each wavelength interval (I) at the target is then obtained by summing spectral components of natural light and searchlight in the expression:

$$\text{Total Light Level}_{(I)} = \text{RADNCE}_{(I)} + \text{SFACTOR} \cdot \text{CPOWER} \cdot .269 \cdot \frac{\text{SEARCH(I, ISERCH)}}{\text{BWIDTH}^2}$$

Here RADNCE is previously discussed natural light level; SFACTOR is irradiance due searchlight; SEARCH (I,ISERCH) is the relative spectral distribution of power in the searchlight for the particular filter employed (ISERCH=1=clear, ISERCH=2=pink filter); CPOWER is the peak candle power of the searchlight employed, and BWIDTH, the beamwidth of the searchlight. ISERCH, CPOWER, BWIDTH, and the X and Y coordinates of the searchlight are planner input parameters. The factor 0.269 is a conversion factor required to transform peak candlepower into spectral emittance (watts/steradian/meter) where meters is the unit of wavelength. The conversion problem is outlined in Figure 3.3-18 where the actual distribution of power in the searchlight is shown as (1) and that contained within the definition of candlepower, i.e., luminous efficiency by the indicator (2).



Ref .14

Figure 3.3-18 SPECTRAL DISTRIBUTION OF SEARCHLIGHT OUTPUT AND VISUAL RESPONSE

Flares and indirect searchlights can also be employed in the simulation for illumination. Flare position in X,Y, and height are provided to the subroutine through the sensor common area. So also are the parameters for the indirect searchlight, XLITE, YLITE, AINTNS, and for both the indicators ILXTRA and MODE. Either of these light sources may be used with a direct searchlight but both indirect searchlight and flare cannot be input at the same time. In this model it is assumed that the indirect searchlight illuminates the cloud base immediately above the target. The assumptions applying to flares are described below in the description of the subroutine.

A value for the index ILXTRA greater than 0 indicates that external sources are to be considered. If the index MODE is equal to 1, flares are to be treated, whereas, if MODE lies between 2 and 7 inclusive, indirect searchlights are employed. The MODE integers 2, 3, and 4 indicate a clear searchlight with beamwidths of 0.017, 0.051, 0.085 radians respectively, while the integers 5, 6, and 7 carry the same beamwidth connotation but for a pink filter searchlight.

Considering first the flare, determinations are first made to locate the flare with respect to cloud ceiling. If the flare height (FLARHT) is greater than the ceiling (CEIL), the flare intensity at the target must be reduced by the cloud transmission factor (FCLOUD). The range from flare to target is inputed and denoted by RANGE4. Fog effects are treated in the same manner described previously and atmospheric transmission is denoted by BTRANS. Thus the relative irrandiance is given by the following expression.

$$EFACTR = \frac{FCLOUD \cdot BTRANS}{(RANGE4)^2}$$

and the spectral irradiance (ECOMP) as:

$$ECOMP = (EFACTR)(AINTNS)(5 \times 10^3)$$

where AINTNS is the candlepower of the flare and 5×10^3 is a conversion factor to transform from candlepower to watts/steradian/meter. Here it is assumed that the flare radiates unformly over wavelength at a level given by that within the luminous efficiency curve. The luminous efficiency curve is assumed to be square in shape and 0.2 microns in width. The flare is also assumed to radiate uniformly over a solid angle of π steradians.

Flare irradiance is then added to the total irradiance in each wavelength increment (I) computed to this point as:

```
Total light level in wavelength increment I =
   Natural source component in wavelength increment I
 + Searchlight component in wavelength increment I
 + Flare light component in wavelength increment I
```

If a flare had been employed a branch in the program is now made to bypass the computations for indirect searchlight since both are not treated simultaneously.

If flares had not been considered, the index MODE had a value of 2 or higher, the program would have bypassed the flare routine and re-entered the subroutine for indirect searchlight simulation. First checks are made on cloud cover (CCOVER) and ceiling (CEIL) to insure the requirements are met. It is assumed that indirect searchlight will be ineffective if cloud cover is less than 50 percent or if ceiling is greater than 2000 meters and this segment of the subroutine would be bypassed if either assumption is not met.

Assuming conditions are proper, range from searchlight to cloud base above target is computed as RANGE6. Atmospheric transmission and fog effects are treated as previously described in previous discussion. The reflective factor (REFLNF) for the cloud is assumed to be

$$REFLNF = 0.05\ (1-TCLOUD)$$

where TCLOUD is the cloud transmission factor, a decimal function ranging from 0.0 to 1.0 and derived from the ATMENV table. It is assumed that the cloud can be treated as an extended area source that provides diffuse illumination to the ground below. The geometric loss factor (EFACTR) for the problem illustrated in Figure 3.3-19 is expressed as:

$$EFACTR = \frac{ALOSS \cdot REFLNF}{\pi(BEAMWD_{INDX}^{2} \cdot RANGE6^{3}) + CEIL^{3}} \cdot CEIL$$

where ALOSS is the total atmospheric transmission factor, BEAMWD(INDX) is the beamwidth of the searchlight as derived from the subroutine data statement keyed by INDX which is itself given by (INDX = MODE-1). The total function of irradiance incident on the target (ECOMP) is them computed as:

$$ECOMP = \frac{0.269(AINTNS)}{(BEAMWD_{INDX})^{2}} \cdot EFACTR$$

and after selecting the appropirate filter keyed on MODE as noted above, the total irradiance for each wavelength increment (I) is computed by:

$$\text{Total Light Level}_{I} = (\text{Total Light Level})_{Nat + SL} + (ECOMP)(\text{Spectral Distribution of Power})_{I}$$

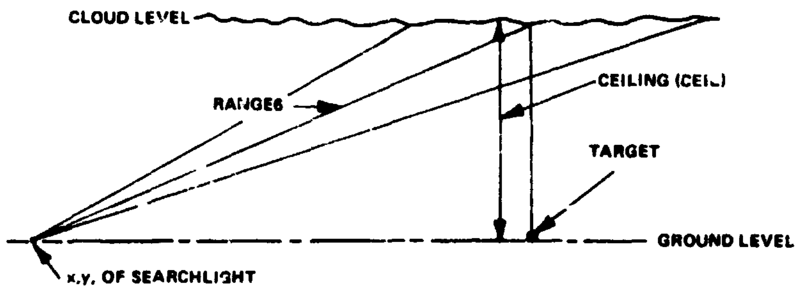


Figure 3.3-19 GEOMETRY FOR INDIRECT SEARCHLIGHT

At this point all sources of light have been examin . and computations for sensor performance are next executed. First, several checks are made. If the index ICLASS is zero, natural vision is implied. If, for natural eyesight, the light level at 550 millimicrons is less than one watt/square meter/meter,* probability of detection will be zero and an exit will be made from the subroutine. If the spectral irradiance lies between 1 and 100 watts/square meter/meter, scotoptic vision response values are employed, but for levels above 100 watts/square meter/meter fotoptic vision response is used. These responses are located in the data statements in the subroutine.

If ICLASS had been one and light level at 550 millimicrons less than 10^5, a night vision device can be applied and the program would have branched to transfer the photo cathode spectral response data into the subroutine. This course will also be taken if light level is greater than 10^5 but a daylight device (ITYPE=0) had been employed.

Having selected the appropriate sensor responses, light level (TLIGHT) is next converted from units of watts/square meter/meter to watts/square centimeter/millimicron by a 10^{-13} factor multiplication. Then by making use of an integrating function subroutine QUAD, computations of total background target, and skylight radiances (RADBAK, RADTAR, RADSKY) are carried out by introducing background and target spectral reflectivities (REFLBK and REFLTG). It should be noted that atmospheric scattering is considered to be insensitive to wavelength in this simulation, a factor that should be considered in further simulation development. The inherent contrast of the target with respect to the background (TRASTI) is then determined as

$$\text{TRASTI} = \frac{|\text{RADBAK} - \text{RADTAR}|}{\text{RADBAK}}$$

and can take on values from 0 (RADBAK = RADTAR) to ∞ (RADTAR $\neq$ 0, RADBAK = 0). The remainder of the subroutine treats the loss in inherent contrast due to atmosphere and sensor and subsequently the probability of detection.

The apparent contrast (ATRAST) is computed from the equation

$$\text{ATRAST} = \frac{\text{TRASTI}}{1 + \frac{\text{CAY1}}{\text{CAY2}}}$$

where the ratio CAY1/CAY2 is the ratio of the amount of power at the sensor due to atmospheric scattered light to that due to the background. The problem is illustrated in Figure 3.3-20.

*1 watt/sq. meter/meter = 10^{-13} watt/cm. sq/millimicron $\simeq 10^{-5}$ foot candles.

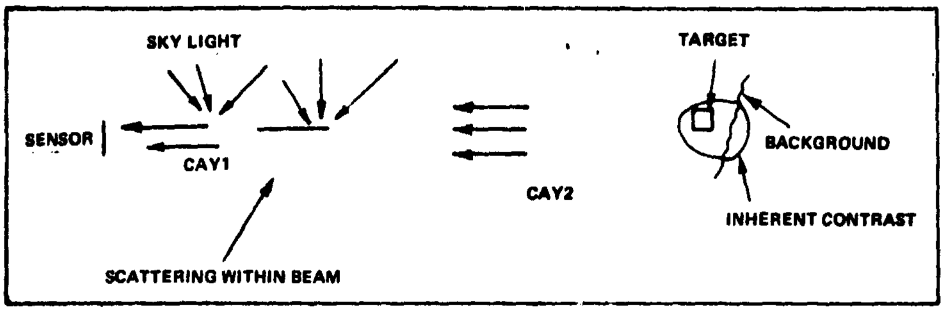


Figure 3.3-20 DEFINITION OF SCATTER COMPONENT, CAY1

Note that for searchlight problems, the atmospheric scatter component, CAY1, is increased by scattering of searchlight power over the sensor-searchlight intersection region.

If an electronic aided sensor is employed the total current on the sensor due to light falling on its aperture is computed as background noise (BKNOIS) as follows:

$$BKNOIS = \frac{(TIMCON)(OPTXMN)(\pi)(CAY1 + CAY2)}{4(FNUMBR)^2 \; (1.6 \times 10^{-17})}$$

The 10^{-17} factor includes the conversions from photons or charge/second to current (1.6×10^{-19}) and conversion from square millimeters to square centimeters. The response of the imaging device is given in terms of modulation transfer function which relates relative amplitude of output cyclic response to input cyclic forcing functions as shown in Fig. 3.3-21. The subroutine uses the modulation transfer constant, the area under the modulation transfer function curve. Thus two devices with the same area would show equal performance although in practice some differences would be observed. Compare the solid (1) and dashed (2) curves which have the same area. The present structure of the model is not adequate to distinguish between these two devices. However, the differences between the first two and that denoted as (3) in the figure are of significance in the simulation.

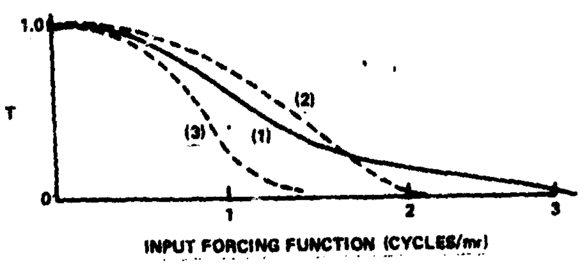


Figure 3.3-21 MODULATION TRANSFER FUNCTION

Ref. (See for example, Ref. 13)

Using the area under the modulation transfer function curve (XMTF), the minimum resolvable area on the sensor is computed as CONST2 by the following sequence of calculations.

$$XMTC = \frac{1000\ \ XMTF}{FOCALL}$$

where FOCALL is the focal length of the sensor in millimeters and XMTC has units of cycles/millimeter.

Then the minimum resolution length (RESLEN) is:

$$RESLEN = \frac{1}{2\sqrt{2\pi}(XMTC)}$$

and

$$CONST2 = 4\pi\ (RESLEN)^2$$

The noise per unit resolvable area is then computed as

$$RNOISE = \sqrt{BKNOIS \div CONST2}$$

The computation of SIGNAL, apparent contrast times background noise, and the signal to noise ratio are straightforward.

The resolution of the sensor will be a function of the light level and of the signal-to-noise ratio (SGNOIS). The effective sensor resolution (EFFECT) is shown in Figure 3.3-22.

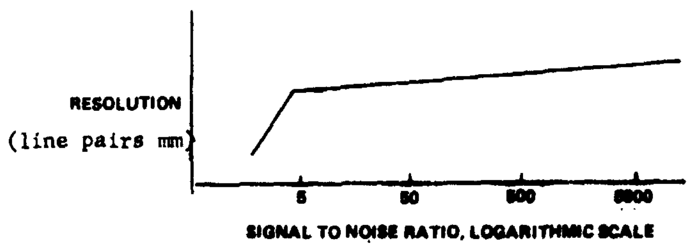


Figure 3.3-22 SENSOR RESOLUTION VS SIGNAL TO NOISE (Ref 13, 14)

If signal-to-noise ratio is greater than five, the response increases only slowly with signal-to-noise ratio itself a function of apparent contrast and an inverse function of the square root of the noise current, a function of light level. For signal-to-noise ratio greater than five the functional relation is given as

$$\text{EFFECT} = 2\ \text{XMTF}\ (0.883 + 0.166\ \log_{10} \text{SGNOIS})$$

The bracketed function is designed to be unity at a signal-to-noise ratio of five. For SGNOIS less than five the response EFFECT falls off at the rate given by

$$\text{EFFECT} = 0.4\ \text{SGNOIS} + \text{XMTF}$$

Then the ratio of the angle subtended by the target to the minimum resolvable angle (RATIO) is found for the conditions prevailing. Multiple element targets are introduced through the index FNTR which is a function of the number of elements in the field of view. The empirical relation for the exponent function EXPON is then determined as:

$$EXPON = CAY3\ (RATIO\ .\ FNTR)^2$$

where CAY3 is a weighting factor. The subroutine then proceeds to the probability of detection, PDET which is computed as:

$$PDET = 1 - e^{-(EXPON)}$$

PDET is subsequently tested against a random number to develop the logical output, LDET = True or False.

For natural vision, the subroutine is re-entered where the conversion from irradiance to illuminance is made. The light level incident on target and background (CANDLE) is converted to units of foot lamberts so that the data of Reference 15 may be employed directly. This data permits the logarithm of apparent contrast (CONTRA) and the light level in foot lamberts to be employed to determine logarithm of minimum resolvable angle subtended by a target for these conditions (VISANG). The functional relations between logarithm contrast, logarithm visual angle and light level are contained in a separate function subroutine ANG. In using function subroutine ANG, and by entering with CANDLE and CONTRA, VISANG is determined. VISANG is converted from its logarithmic basis to minutes by:

$$ANGLE = 10^{(VISANG)}$$

Then the ratio of the angle subtended by the target to the minimum resolvable angle is determined using a factor 3437.747 for the conversion from radians to minutes. As with electronic aided devices, the multiple-element target factor FNTR and RATIO are used to determine the exponent function

$$EXPON = DEVCON\ (RATIO\ .\ FNTR)^2$$

where DEVCON is a weighting factor with the appropriate [illegible] of DEVCAL. The probability of detection is determined as described above.

3.3.9 Subroutine BRKWIR

3.3.9.1 Purpose

This subroutine provides a simulation of breakwire devices for determining probability of actuating the sensor and to generate an output report, either a true or false detection.

3.3.9.2 Glossary of Inputs, Computed Values, and Outputs

Input Values

KSTRNG	Target Classifier
IDSNSR	Sensor ID
IDTGT	Target ID
ITGTTP	Target Type
IUT	Index on Unit Terrain
NOELEM	Number of Elements in Target Group

Internally Stored Designer Input Values

DISCOM	Detection Factor, A Function of Vegetation

Labelled Common Inputed Values

IPRINT	Output Data Device Designator = 6
ITIME	Game Running Time
ITOD	Time of Day
LDUMP	True = Intermediate Calculations Printed, False = No Print

Computed Values

DAYLIT	Detection Factor, A Function of Light Level
DISCOV	Detection Factor, A Function of Vegetation
DUM	Dummy Argument
EXPON	Effective Number of Elements
KFAUN	Index on Time of Day, 1=Daylight, 2=Night
PBRK	Probability for Single Element to Break Wire
PDET	Probability of Detecting Target
PDWIRE	Probability of Detecting Wire
PWDET	Probability of Detecting Wire, Best Conditions

Output Values

LDET	Detection (True-False)

3.3.9.3 Description of Subroutine Logic and Processing

The breakwire device consists of a thin cable consisting of two very fine wires (AWG44, for example) which is emplaced around a perimeter or along a line to be monitored. If an intrusion takes place, the wire is broken and continuity being checked at the monitor is lost, resulting in an alarm. The simulation of this type device, therefore, consists simply of a probability statement regarding the breakage of the wire given an intrusion event.

In this subroutine the probability of breaking the wire is given as a function of the number of elements in the target, the probability per element of breaking the wire, the probability per element of discovering the wire before breakage, and the target type. On entering the subroutine (Figure 3.3-23) processing is directed to one of seven sets of assignment statements depending on target type (ITGTTP) and character (KSTRNG). Each set of assignments contains an estimate of the probability for breaking the wire per target element (PBRK) and an estimate of the probability of the wire being detected by the first element in the target (PWDET). Thus for a troop type target PBRK is assumed to be 0.5 and PWDET to be 1.0. This latter estimate will be modified by foliage and light level at a later stage in the program.

Next a determination of time of day of significance to this sensor is computed through the index KFAUN. Daylight is considered to extend from 6AM to 6PM with KFAUN = 1 and a light level detection factor, DAYLIT = 1.0 for this condition. For night conditions DAYLIT is set equal to 0.1.

The vegetation characteristics in the vicinity of the sensor will also play a role in the detection capability of the intruder for the wire. A set of wire detection modifiers are contained in the data set labelled DISCOV which is keyed to the terrain index (IUT) number, (Section 5.3.2), The assignments based on intuitive argument only, are shown in the table below.

IUT Number	IUT Description	DISCOV
1	Rice Land	0.5
2	Single Canopy, Light Undergrowth	0.2
3	Brush Wood, Coffee Plantations	0.2
4	Brush Wood, Coffee Plantations	0.2
5	Multi-canopy, Dense Undergrowth	0.1
6	Multi-canopy, Dense Undergrowth	0.1
7	Single or Multiple Canopy, Bamboo	0.1
8	Dune Grass on Sand	0.7
9	Not Specified as Yet	0.0
10	Not Specified as Yet	0.0

The probability of detecting the wire (PDWIRE) is computed as:

PDWIRE = (PWDET)(DAYLIT)(DISCOV)

This result is tested against a random number and if detection probability is greater than the random number, the number of elements in the target is

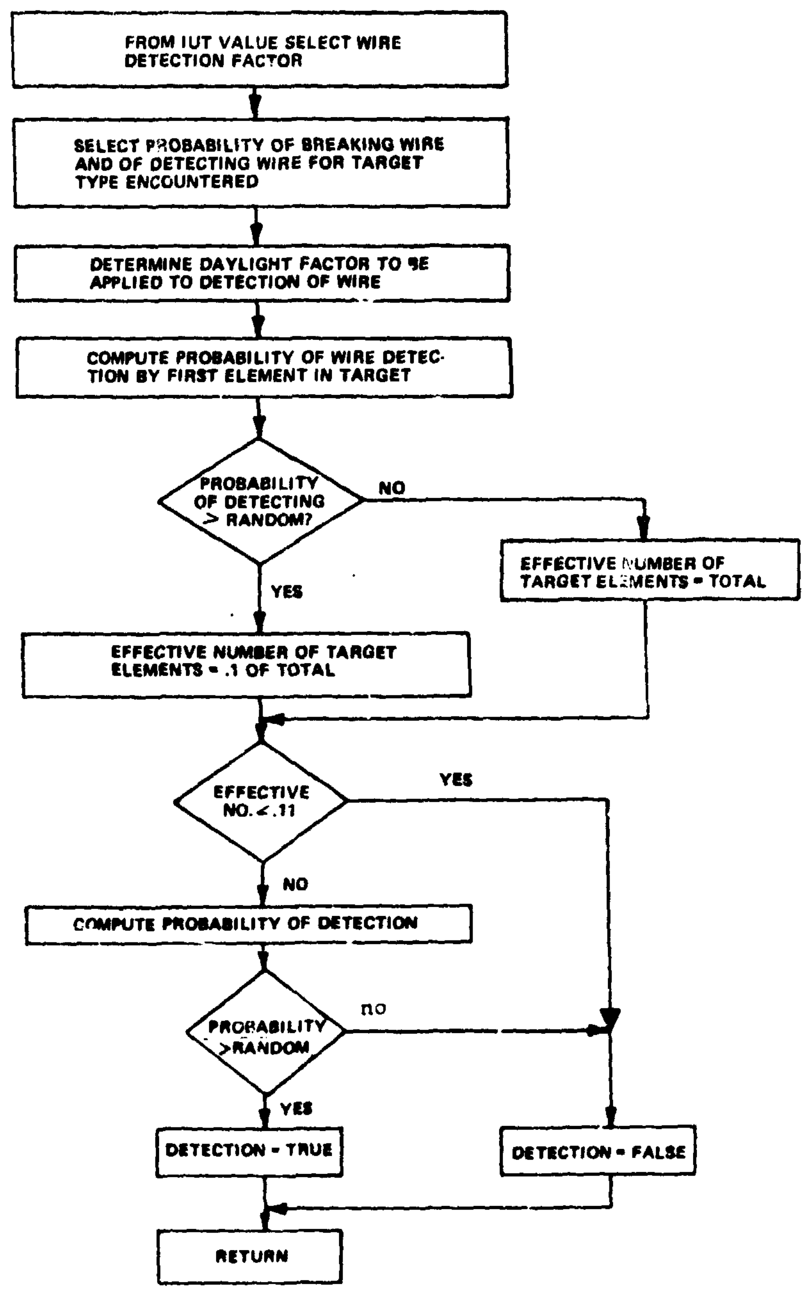


Figure 3.3-23 BRKWIR MACROFLOW

reduced by a factor of ten. Otherwise the effective number of target elements is the actual number. The variable for effective number of elements (EXPON) is thus developed by these selection rules. If EXPON is less than 0.11, however (which would be the case if the number of elements in the target was one and the discovery of the wire was found to be true), detection is declared to be false and control is returned to the executive subroutine. Otherwise probability of detection (PDET), probability of target breaking the wire is given by:

$$PDET = 1.0 - (1.0 - PBRK)^{EXPON}$$

A test is then made on PDET by comparing PDET with a random number. If the PDET is the greater, a detection is declared to take place, otherwise LDET = False.

It is to be noted that the probabilities associated with the target types for both breaking and discovering the wire are only suggested values at this time having no supporting field data for a basis. As such they must be regarded as tentative and highly subject to change as the field data base is developed.

Section 3

References

1. Atlantic Research Corp., Jansky and Bailey Engin. Dept. "Acoustic and Seismic Research", Final Rept. No. no. Jan 1967.

2. "Seismic and Meteorological Research Program"(U), Final Rept. Vol. 1,2 (1111-11-F(I), F(II) Oct. 1968, Stanford Research Institute and University of Michigan.

3. O'Brien, P.N.S., "The Efficient Use of Large Charges" in "Seismic Refraction Prospecting", Society of Exploration Geophysicists, Banta Co. Inc., Menasha, Wisc., 1967.

4. Handbook of Geophysics and Space Environments Air Force Cambridge Research Laboratories McGraw Hill Book Company LC CCN 65-28352, 1965.

5. Hooper, E.T. in Journal of Defense Research, Series B, Tactical Warfare, Fall 1969.

6. Acchione; L.J., Bliss,R., Hein,T., Johnston,D., "STANITEOPS Visonic Field Evaluation, Warren Grove Test", Final Report, 31 Jan 1969, Visionics Tech Area, Night Vision Lab., Army Electronics Command, Ft. Belvoir, Va.

7. Hudson, R.D., "Infrared Systems Engineering", Wiley & Sons, 1969.

8. Schneeberger, R.F., "Environmental Subroutine Contribution, Radar Precipitation Effects", CAL MASS Memo 501, 15 April 1970.

9. Barton, D.K., "Radar System Analysis". Prentice Hall, 1965.

10. Journal of Defense Research, Series B., Institute for Defense Analysis, Spring 1969.

11. "Tropical Propagation Research", Final Report, Vol. I, Atlantic Research Corp., Jansky & Bailey Engin. Dept., November, June 1966.

12. Berkowitz, R.S., "Modern Radar", Part IV, John Wiley &Sons, 1965.

13. Parsons, N.W., Bruce, C.A., Criste, R.E., "Use of Carmonette-IV in Assessing Combat Effectiveness of Small Units Equipped with Night Vision Devices", Research Analysis Corp. Draft (Nov. 1969), to be published.

14. Lockheed Missiles and Space Company, Report B 099021-1.

15. Blackwell, R.H., "Contrast Thresholds of the Human Eye", Journal of Optical Society of America, Vol. 36, No. 11, November 1946.

16. "Reference Data for Radio Engineers", Vol 5, International Telephone and Telegraph.

17. Handbook of Geophysics, Air Force Cambridge Research Center, McGraw Hill Book Company, 1960

VOLUME I, PART 1 INDEX

INDEX (Continued)

INDEX(Continued)

INDEX (Continued)