

AD-A101 860

INTEGRATED SCIENCES CORP SANTA MONICA CA

F/G 15/7

CONCEPTS FOR HIGHLY INTERACTIVE AIR STRIKE PLANNING SYSTEMS (AS--ETC(U)

MAY 81 D H WALSH, G J REBANE, L R LEVI

N00014-80-C-0812

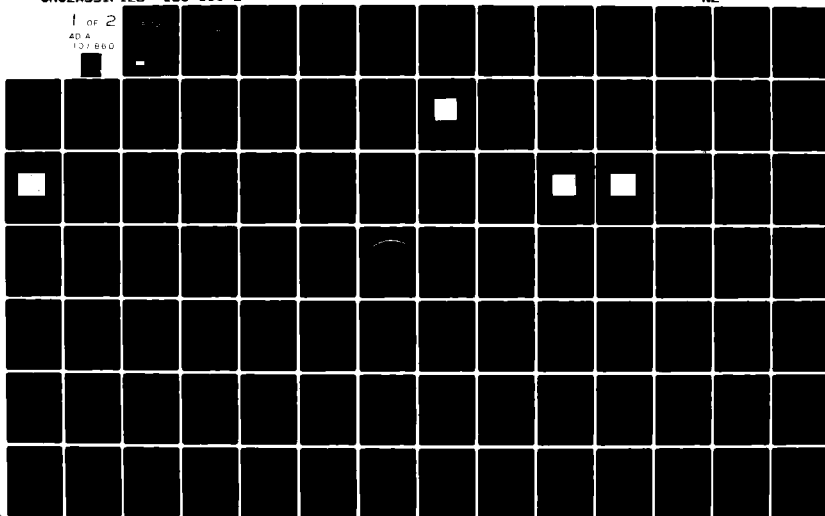
UNCLASSIFIED

ISC-330-2

NL

1 of 2

AD-A
137 860



AD A101000

DTIC FILE COPY

**CONCEPTS FOR HIGHLY
INTERACTIVE AIR STRIKE
PLANNING SYSTEMS (ASPS)**

REPORT NO. 330-2

PREPARED FOR:

CODE 455

DIRECTOR, ENGINEERING PSYCHOLOGY PROGRAM
PSYCHOLOGICAL SCIENCES DIVISION
OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
ARLINGTON, VIRGINIA 22217

MAY 1981

Reproduction in whole or in part is permitted for any purpose of
the United States Government.

This document has been approved
for public sale; its
distribution is unlimited.



INTEGRATED SCIENCES CORPORATION
Santa Monica, California

"Original contains color
plates: All DTIC reproductions
will be in black and
white."

81 7 22 101

LEVEL 4

(12)

DTIC

JUL 23 1981

A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
	AD-A702 860	
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED
CONCEPTS FOR HIGHLY INTERACTIVE AIR STRIKE PLANNING SYSTEMS		TECHNICAL REPORT
7. AUTHOR(s)		6. PERFORMING ORG. REPORT NUMBER
D.H. Walsh L.R. Levi G.J. Rebane R. Tash		330-2
9. PERFORMING ORGANIZATION NAME AND ADDRESS		8. CONTRACT OR GRANT NUMBER(s)
Integrated Sciences Corporation 1640 Fifth Street, Suite 204 Santa Monica, CA 90401		N00014-80-C-0812
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
ENGINEERING PSYCHOLOGY PROGRAM, CODE 455 OFFICE OF NAVAL RESEARCH ARLINGTON, VIRGINIA 22217		
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE
		May 1981
		13. NUMBER OF PAGES
		15. SECURITY CLASS. (of this report)
		UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
		N/A
16. DISTRIBUTION STATEMENT (of this Report)		
UNLIMITED		
This document has been approved for public release; its classification is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
Reproduction in whole or in part is permitted for any purpose of the United States Government.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Decision Aids Planning Aids Naval Tactics Interactive Graphics Man/Computer Functional Allocation Nonlinear Programming Optimization Command and Control Mission Planning		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
This is a report of work accomplished on one task in the project entitled Operational Decision Aids (ODA), which was initiated in 1974 by the Office of Naval Research to develop aids for Navy command and control functions and make them available for incorporation in the design of future systems. The first phase of the work produced a concept for a laboratory version of an Air Strike Planning System (ASPS) and about 2,000 lines of operational FORTRAN code. The purpose of the laboratory		

DD FORM 1473 1 JAN 73

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Abstract (continued)

ASPS was to experimentally test operator performances using variations of ASPS which differed in the amount of automation available to the air strike planner. Inputs to the concept formulation process about air strike planning needs and perspectives were obtained from A-7, A-6, and EA-6B pilots at Lemoore and Whidbey Naval Air Stations. The software implemented much of the user/machine interface for a computer and display system.

A concept for a fieldable version of ASPS that can be evaluated at the squadron level was developed during the second phase of the work. That concept and a recommendation for hardware to implement the concept are documented in the report.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

6)
CONCEPTS FOR HIGHLY INTERACTIVE
AIR STRIKE PLANNING SYSTEMS (ASPS)

12) 463
14) ISZ
Report No. -330-2

13) Contract No. N00014-80-C-0812

9) Technical Repts.

Prepared for:

Code 455
Director, Engineering Psychology Program
Psychological Sciences Division
Office of Naval Research
Department of the Navy
Arlington, Virginia

By:

10)
David H. / Walsh
George J. / Rebane
Lawrence R. / Levi
Richard / Tash

Integrated Sciences Corporation
1640 Fifth Street
Santa Monica, California 90401

11 May 1981

has been approved
in relation and under 200
to the

Distribution For	
GRA&I	✓
TAB	
Unpublished	
Classification	
Dist	
A	23

404569

ACKNOWLEDGMENTS

We would like to acknowledge the support of Dr. Martin A. Tolcott, Office of Naval Research. We especially appreciate the guidance and thoughtful suggestions by Dr. Tolcott and Cdr. Kent Hull of ONR. Cdr. Hull also facilitated our access to the many sources of information about air strike planning and use of digitized terrain data that were used in our work.

SUMMARY

This is a report of work accomplished on one task in the project entitled Operational Decision Aids, which was initiated in 1974 by the Office of Naval Research to develop aids for Navy command and control functions and make them available for incorporation in the design of future systems. In previous work on the ODA program, ISC developed and tested two interactive methods of selecting (a) an air strike path through a field of ten enemy sensors and (b) aircraft speeds on each leg of the path. Both methods were implemented on a computer-driven, four-color vector graphics display. Utility for each candidate strike path was computed according to a predetermined utility function; the utility function was nonlinear and multi-modal. With one method called Operator Aided Optimization (OAO), the operator guided an optimizing nonlinear programming algorithm toward a solution. With the other method called Iterative Manual Optimization (IMO), the operator selected a candidate solution and the computer acted as a calculator by calculating and displaying the solution's utility. The operator then modified the previous solution in light of what was learned from seeing its utility. The process was repeated until the operator was satisfied with the utility score achieved. An experiment using sixteen college students as subjects did not conclusively demonstrate that OAO was superior to IMO for problems having the relatively low level of complexity of the experimental problems.

The initial phase of the work reported in this document was a continuation of the previous work. ISC's aim was to develop and experimentally test OAO- and IMO-type aids that would account for most of the complexity of real-world problems by treating most of the problem factors, decision dimensions, and utility dimensions currently deemed important by air strike planners. Inputs about air strike planning needs and perspectives were obtained by visits to strike planners at Lemoore and Whidbey Island Naval Air Stations in October and November 1980 respectively. Strike planners from these Naval Air Stations also visited ISC in March 1981 and exercised on ISC's display system the operational software that had been developed up to that time. The report documents the concept and software that were developed for the laboratory version of the Air Strike Planning System (ASPS).

ODA funds for FY 81 and future years were reprogrammed in early 1981. A result of this action was that ISC received direction to devote its remaining contract effort to planning the modification and transfer of the developed technology to a fieldable desktop version of ASPS. Consequently, ISC developed during the second phase of its work a concept for a fieldable version of ASPS that can be evaluated at the squadron level. That concept is documented in this report.

During the latter stages of work, ISC concentrated on two key problem areas for a future effort to develop a fieldable planning aid for evaluation. The first of these involves the following:

1. Storing the digital terrain data that would be needed to represent real maps.
2. Development of an algorithm or modification of an existing algorithm for efficiently calculating terrain masking of hostile weapons/sensors at specified aircraft altitude levels above ground level.

ISC determined that digitized map products derived from the Defense Mapping Agency's Digital Landmass System (DLMS) Data Base are available and applicable to ASPS. The report describes several algorithms that have been developed for other programs for calculating terrain masking around a given point.

The second key problem is selecting the most cost-efficient computer and display system for implementing the system concept. ISC analyzed the capabilities of state-of-the-art graphics displays. A combination of a Megatek 7250 display and a Digital Equipment Corporation MINC 11/23 desktop computer was judged to be the best hardware available for a desktop version of ASPS.

Development of a highly interactive ASPS is recommended because ASPS will enable planners to use their time more efficiently than is possible with current manual procedures and because cost of ASPS development and purchase will be low. Low development cost is attributable to the fact that much of the design work has already been done. Low purchase cost is due to the recent arrival of exceptionally powerful, low cost desktop computers on the market.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.	ii
SUMMARY.	iii
1.0 INTRODUCTION	1
1.1 BACKGROUND.	1
1.2 CURRENT WORK.	3
2.0 CONCEPTS FOR AN AIR STRIKE PLANNING SYSTEM (ASPS).	7
2.1 CONCEPT FOR A LABORATORY VERSION OF ASPS.	7
2.2 CONCEPT FOR A FIELDABLE VERSION OF ASPS	30
3.0 INVESTIGATIONS OF CRITICAL ASPECTS OF ASPS DEVELOPMENT	40
3.1 DIGITIZED MAPS.	40
3.2 INVESTIGATION OF HARDWARE ALTERNATIVES FOR ASPS	44
4.0 RECOMMENDATION	54
REFERENCES.	56
APPENDIX A: SOFTWARE STRUCTURE	57
APPENDIX B: DATA FILE STRUCTURE.	62
APPENDIX C: SOFTWARE LISTINGS.	75
DISTRIBUTION LIST	150

LIST OF FIGURES

Figure		Page
1	A Map Recalled by the Planner.	10
2	Preliminary Screen Layout for the Air Strike Planning System (ASPS).	12
3	Function Key Setup	14
4	A Path Which Has Been Manipulated Using Change Path Mode . . .	17
5	Illustration of Range and Bearing Use During "Change Path" Operations	26
6	Defense Capability Contours for a Particular AGL	27
7	ASTAP Overview	35
8	ASTAP Display Elements	37
9	Illustration of a Masked Region Along a Radial from a Defense Site	43
10	DEC MINC 11/23	50
A-1	ASPS Functional Architecture	54
A-2	Function Key Setup	60
A-3	Contents of Admins Data File	61
A-4	Data Format of the Master Data File Header	65
A-5	Terrain Data Format.	66
A-6	Sensor, SAM, and Character String Data Formats	67
A-7	Data Format for Latitude/Longitude Position and Significant Points	68
A-8	Data Format of Path Information	69
A-9	Stores Data Format	70
A-10	Data Format of Sensor Contour Pointers	71
A-11	Data Format of Sensor Contour Points	72
A-12	Data Format for SAM Contours	73
A-13	Data Format for AA Contours and Locations.	74

LIST OF TABLES

Table		Page
1	Table of Symbols	13
2	Illustration of What Happens When a Waypoint is Added	19
3	Illustration of What Happens when a Waypoint is Deleted . . .	18
4	ASTAP Functions	38

1.0 INTRODUCTION

This report documents the results of a two-phase effort to field a desktop version of an interactive air strike planning system (ASPS) for evaluation at the squadron level. The initial phase of this work was a continuation of a multi-year participation by ISC in the Operational Decision Aids program of ONR. Effort was directed during the second phase toward a concept design that adapted a subset of the laboratory ASPS capability into the physical and computational constraints of a fieldable and relatively inexpensive desktop ASPS.

1.1 BACKGROUND

This is the sixth report by Integrated Sciences Corporation (ISC) as one of a group of contractors working on the Operational Decision Aids (ODA) program directed by the Office of Naval Research. The ODA program was initiated in 1974. Its intent is to develop a variety of decision aids and test and evaluate their usefulness to the Navy. Although the program is not tied to any specific command and control hardware system, it has focused on the functions of a Task Force Commander (TFC) and his staff. The role of ISC has been to find ways to improve user/machine communication by allocating functions between user and machine that take advantage of their respective strengths.

Under a previous contract ISC developed aids that were used to test the ability of humans to perceive complex functional relationships presented in geometric/graphical format (References 1, 2, and 3) on a computer-driven color display. The problem situation for which the aids were developed was the selection of (a) an air strike route through a field of multiple enemy sensors and (b) aircraft speeds on each leg. (Hereafter in this report, the selection of aircraft flight parameters is abbreviated to "route planning.")

One of the ISC-developed aids was called Operator Aided Optimization (OAO) using a Nonlinear Programming (NP) algorithm or OAO/NP. For the OAO/NP aid, the operator controls the algorithm by:

1. Choosing a starting point for the algorithm.
2. Stopping the optimization process of the algorithm when the utility calculated by a mathematical utility function shows diminishing returns versus time.
3. Selecting a new starting point in another region of the solution space.

Another mode of aiding the operator, Iterative Manual Optimization (IMO), was also developed. With IMO, an operator inputs a solution consisting of the route legs and speeds on each leg. The computer then calculates and displays the solution's utility and does nothing more until the next change to the route is input by the operator. Thus, with IMO, the computer acts as a calculator and all optimization is done by the operator.

An experiment was performed to determine how closely operator performance using IMO on a set of experimental problems would match performance using OAO/NP. The experiment was aimed at shedding light on the question of the importance of an optimizing algorithm as part of an aid to solving a problem having a multimodal, unsymmetric, nonlinear criterion function. The results showed that operators using IMO achieved an average score of 98.1 utility points by the end of the 15-minute experimental problems, whereas the average score was 99.4 using OAO/NP. The inference drawn from these results was that an optimizing algorithm does not greatly improve performance for problems having the level of complexity of the experimental problems. However, even for problems of this moderate complexity, use of an optimizing algorithm is worthwhile if its software cost is low and there is wide agreement among operators about the form and specific parameters of the utility function. The reason is that operators prefer using an algorithm because it makes the task simpler and less tiring.

In many situations where a planning aid will be useful, it will not be possible to pre-specify a universally acceptable form and parameters of a utility function because:

1. The importance of utility dimensions will vary from one tactical situation to another in the mind of a single decision maker.
2. Different decision makers will have different perceptions of what constitutes utility.

It is natural to ask whether optimizing algorithms have a place in aids for these situations. ISC believes they do. Our reasoning begins with the idea that, in these cases, the planning aid must be designed to do one of the following:

1. Allow the operator to select utility dimensions to be included in an overall criterion and to specify mathematical forms for combining dimensions and weights for each dimension. (It's difficult to do this well.)
2. Be free of machine calculation of an overall utility value. (The operator mentally combines the values along the utility dimensions.)

If (1) above is successfully implemented with mathematically tractable functions to the satisfaction of end users, then it is certainly reasonable to allow operator guided use of a general purpose optimizing algorithm. If (2) above is the more desirable approach, then it is reasonable to allow the operator to guide an optimizing algorithm along a single utility dimension at a time and let values along other dimensions fall where they may. In this case the operator can converge on the best solution across multiple utility dimensions through an iterative process.

1.2 CURRENT WORK

The real world of air strike planning is more complex than the problem used to compare the OAO and IMO concepts. Important real-world considerations are listed below:

Problem Factors

- capabilities and location of enemy sensors
- capabilities and locations of enemy missiles
- capabilities of enemy fighters and locations of their bases
- capabilities and locations of enemy anti-aircraft guns
- change in location of enemy defense forces during the flight time of the air strike
- terrain topography

Decision Dimensions

- x,y locations of path legs
- speed on each leg
- altitude on each leg
- when to use on-board jamming equipment
- where to direct use of electronic warfare aircraft
- weapon and fuel loading
- when and where to use standoff anti-radar weapons

Utility Dimensions

- expected damage to strike aircraft due to enemy defenses
- probability of strike mission success
- distance between target and strike aircraft when cumulative detection probability exceeds $x\%$

The IMO and OAO aids tested in the previous work did not account for all of these dimensions. Consequently, the following question arose:

Is the small magnitude of the performance difference between IMO and OAO observed in the study due to the relative simplicity of the problems to be solved?

This question may be put another way:

How will the difference in operator performance using IMO- and OAO-type aids change as the dimensions of the problem, decision, and utility function increase?

Performance data from aids and experiments designed to answer this question could be important evidence supporting a decision by R&D program managers to concentrate future efforts on developing one of the aiding concepts in preference to the other.

Consequently, the original aim of the current work was to develop and experimentally test OAO- and IMO-type aids that would account for most of the real-world problem factors, decision dimensions, and utility dimensions currently deemed important by air strike planners. ISC project members visited A-7 and A-6 strike planners at Lemoore and Whidbey Island Naval Air Stations in October and November 1980 in order to gain understanding of the route planning problem as the planners see it. We also obtained inputs concerning the planners' preferences for features and characteristics of an aid. This information provided much of the basis for our initial concept for an air strike planning aid (Reference 4).

ODA funds for FY 81 and future years were reprogrammed in early 1981. The ISC air strike planning concept and software are applicable to a current fleet need; consequently, ISC received the direction to devote its remaining contract effort to planning the modification and transfer of the developed technology to a fieldable desktop Air Strike Planning System.

Our initial concept for the aid was modified as a result of additional information obtained from strike planners in February and March. A major new input was received from Marine Corps aviators at Quantico in February. They stressed the need for an aid to planning and coordinating actions in the target area. This idea had been mentioned previously at Lemoore and Whidbey, but it had not been stressed. Additional inputs were received from strike planners who visited ISC in March, reviewed our concept, and exercised on our display system the operational software that had been developed up to that time. The reviewing officers were strike planners from Lemoore and an A-6 strike planner and an EA-6B crew member from Whidbey. The major new input from these visits was the desirability of having the aid show a dynamic replay of an alternative. As before, this feature had been mentioned earlier by strike planners, but it had not been stressed.

During the latter stages of work, ISC concentrated on two key problem areas for a future effort to develop a fieldable planning aid for evaluation. The first of these involves the following:

1. Storing the digital terrain data that would be needed to represent real maps.
2. Development of an algorithm or modification of an existing algorithm for efficiently calculating terrain masking of hostile weapons and sensors at specified aircraft altitude levels above ground level.

The second key problem is selecting the most cost-efficient computer and display system for implementing the system concept.

Section 2 of this report contains our concept for an Air Strike Planning System (ASPS) that embodies most of the features recommended by the Naval and Marine Corps air strike planners consulted by ISC. Elements of this concept have been translated into operational software; these elements are noted in Section 2. Section 3 contains the results of our investigations into the two key problems summarized in the previous paragraph. Our recommendation for development of a highly interactive Air Strike Planning System (ASPS) is in Section 4. Documentation of software developed during this project is presented in the appendices.

2.0 CONCEPTS FOR AN AIR STRIKE PLANNING SYSTEM (ASPS)

A concept and software for a laboratory version of an air strike planning system were developed during the initial phase of the contract. The laboratory version was to be used to test hypotheses about the performance of planners on problems that were virtually as complex as real planning problems. The most important results of the tests were expected to be the comparisons between performance using iterative manual optimization and operator aided optimization. The concept for the laboratory version of ASPS is documented in Subsection 2.1.

A fieldable ASPS for evaluation by an operational squadron requires capabilities that are not needed for the laboratory tests. The most important such capability is representation of actual terrain on the system's computer-driven display. Subsection 2.2 describes:

1. Capabilities that would be different in the fieldable version from the laboratory system described in Subsection 2.1.
2. Capabilities that strike planners desire but that are not needed to test hypotheses about OAO and IMO in a laboratory experiment.

2.1 CONCEPT FOR A LABORATORY VERSION OF ASPS

Elements of the concept which have been implemented in operational software at ISC are indicated by an asterisk within parentheses preceding the applicable concept elements, i.e., (*). The software structure and data file structure for these concept elements are discussed in the appendices. The data file structure for the completed ASPS system will be enhanced to allow for additional capabilities, but we expect its basic structure to remain the same.

The hardware for the laboratory version of ASPS includes a Sperry Univac V-73 minicomputer, a four-color vector graphics display, a 32-key special function keyboard, and a trackball. The special function keys and trackball are the instruments which allow the planner to communicate with the computer. Their specific uses are explained in detail below.

2.1.1 Problem Setup and Controls

(*) 2.1.1.1 Terrain Modeling; Modeling and Plotting Enemy Defenses. Scenario terrains will be developed off-line and stored in the computer. Terrain features will include:

1. Four Above-Ground-Level (AGL) values of altitudes
2. Cities
3. Forests
4. Lakes
5. Roads

(*) Locations of radar, missile, and anti-aircraft (AA) gun sites will also be stored with terrain. Contours of what can be seen for each AGL on a line-of-sight basis from each radar, gun, and missile site will be stored and displayed on operator request. The composite contours of what can be seen for each of the four AGL's will also be stored for the group of radars, the group of missile sites, and the group of AA gun sites.

(*) Each element of terrain and displayed information about defenses will have a display "priority" code attached to it. These are:

<u>PRIORITY I</u>	<u>PRIORITY II</u>	<u>PRIORITY III</u>
Terrain features	Visually significant points	Detectability contours for radars
Radar sites	Radar significant points	Reachability contours for missiles and AA weapons
Missile sites	(These are inputs by the operator. See Section 2.1.2 on Variable Background)	

Priority I information will always be displayed. Priority II and Priority III information will be displayed by operator command.

(*) 2.1.1.2 Beginning Operations. At the beginning of system operation, the following menu appears:

"Select MAP 1
2
3
4"

The operator indexes a cursor to the desired map and then pushes ACCEPT.¹ The screen shows Priority I information, namely, the radar and missile sites, and the physical terrain.

(*) After selecting the map from the menu, the operator uses the number pad to input the plan number.² (See Figure 1.) A message indicating the number of alternative plans already stored appears below the screen. For each map, Plan 1 is always the basic map without significant points or path. (If only the map exists without any plans, then the user will not be asked for a plan number but, instead, will be placed in the Director Mode). After inputting and accepting the plan number, the path and significant points will be displayed. The user is now in the Director Mode.

(*) 2.1.1.3 Director Mode. The Director Mode controls the flow of the program as specified by the operator. The active special function keys are lighted and await a user response. The key pressed specifies the next activity to be worked upon by the operator.

(*) Throughout all procedures, the latitude and longitude position of the cursor will be displayed whenever the cursor appears on the screen. Only those keys which are lighted will be active at any time during the flow of the program.

¹The number of maps available is limited only by computer memory.

²A plan is defined to be a path, speed, and altitude for each aircraft type in the strike plan. The plan for strikes including EA-6B aircraft includes the points in the path where electronic warfare equipment is turned on.

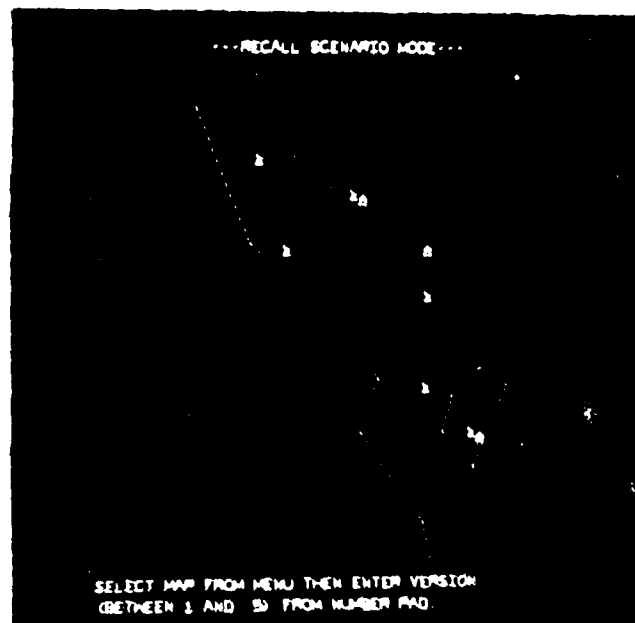


Figure 6. A Map Recalled by the Planner

2.1.1.4 Initialization. The user is given the opportunity to initialize certain constraints and inputs in the program such as fuel allocation, amount of fuel at recovery, and local time at beginning of the mission. There is an INIT key if he wishes to make changes; otherwise, default values will be used.

(*) 2.1.1.5 Screen Layout, Symbology, and Special Function Keys. Figure 2 shows the preliminary screen layout for the system. The display screen is laid out with the map in the center and miscellaneous information around the sides. The miscellaneous information includes menus, task and error messages, latitude and longitude position of the cursor, the current mode, a table of leg speeds and AGL's along the path, and other pertinent information. Table 1 describes the symbol and line conventions used in the ASPS program. Figure 3 shows the layout of the special function keyboard.

2.1.2 Variable Background

(*) The user is given the capacity to add elements which he considers important to the map. There are two types of these, namely, radar significant points (RSP) and visual significant points (VSP). The RSP's are points detectable by radar and VSP's are points that can be seen visually. These types of points might be relevant when selecting the path to and from the target area.

(*) When the operator presses SIGNIFICANT POINTS key, a cursor appears on the screen and is maneuvered with the trackball. All radar significant points are located, followed by visually significant points. The operator moves the cursor to the desired location and presses the ACCEPT key. An "R0" appears at that position. He moves the cursor to the location for the next radar significant point and ACCEPTs, which displays an "R1". This process continues up to a maximum of ten radar significant points. If the RETURN key is pressed before ten radar significant points are located, the user will then be asked to locate the visually significant points.

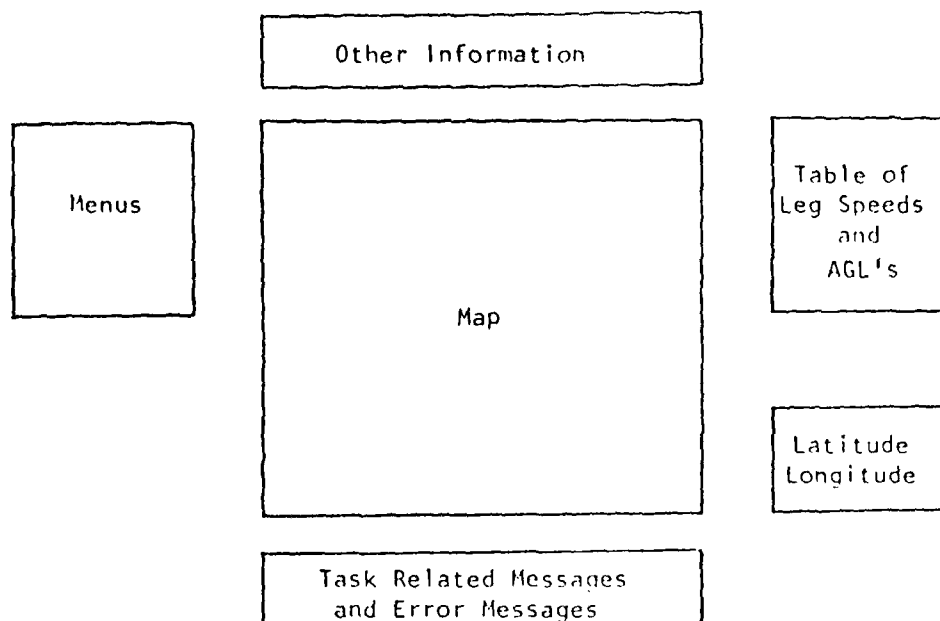


Figure 2. Preliminary Screen Layout for the Air Strike Planning System (ASPS).

Table 1. TABLE OF SYMBOLS

Symbol/Line Type	Description	Color
	Launch Point	Red
	Target point	Red
	Recovery Point	Red
	Way Point	Green
	SAM Site	Orange
	Sensor Site	Orange
Ri i = 0,...,9	Radar Significant Point	Yellow
Vi i = 0,...,9	Visually Significant Point	Yellow
	Cursor	Yellow
Solid Line	Path	Green
Dash-Dot Line	Line Connecting Range and Bearing Reference Point to Cursor	Red
Dashed Line	Sensor Contours	Red
Dash-Dot Line	Shoreline	Green
Dotted Line	Hills	Yellow
Dash-Dot Line	City	Yellow

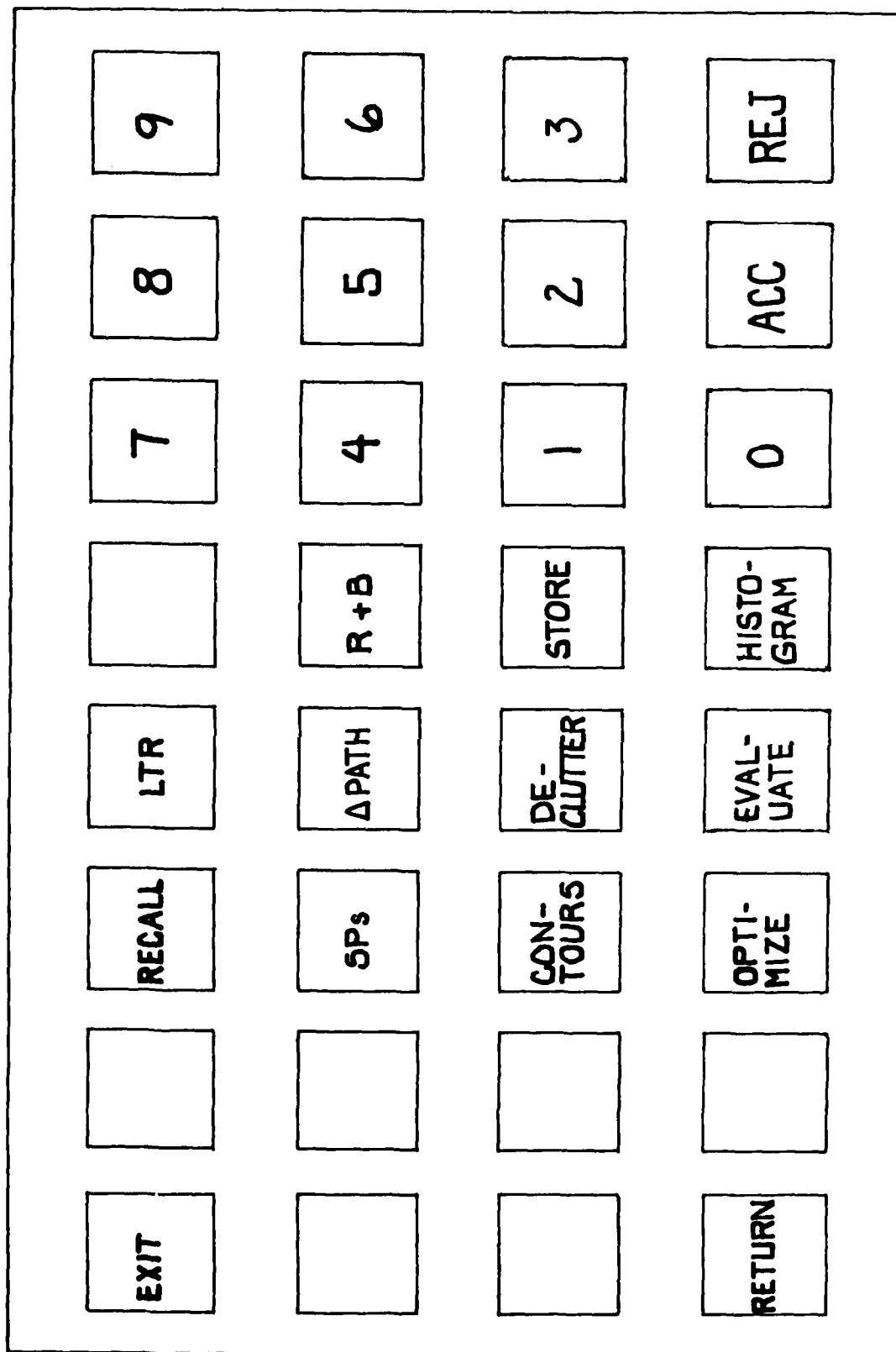


Figure 3. Function Key Setup.

(*) The process for visually significant points is the same as for radar significant points. A "V0", "V1", etc. appears on the screen identifying the locations of the visually significant points. Pressing the RETURN key will return control to the Director Mode.

2.1.3 Path Manipulation - IMO

This section describes how to change the path and path features. The basic idea is to find the path which yields the best value of the figure of merit (FOM) and satisfies the fuel constraint. The system enables the user to alter the route, speed, and AGL along the route until an acceptable FOM is determined. The FOM is calculated by use of the EVALUATE key.

(*) 2.1.3.1 Key "LTR". There is a key "LTR", which stands for Launch Point, Target Point, Recovery Point. When this key is pressed, a message appears below the map:

LOCATE LAUNCH POSITION WITH CURSOR.

PRESS 'ACCEPT' KEY WHEN CURSOR IS OVER DESIRED POSITION.

A cursor is moved to the desired launch point and accepted with the ACCEPT key. A symbol "L" will appear for the launch point.

(*) Similarly, this procedure is followed in turn for the target and recovery nodes with the appropriate change in the message. A "T" appears to identify the target point and an "R" for the recovery point.

(*) After locating the recovery point, a line connecting the route from launch to target to recovery is displayed. A Leg/Speed/AGL Table appears, indicating the speed and altitude for each respective leg. The default values are 360 knots at 500 feet.

(*) After the recovery position is identified, a menu appears that enables the user to specify the type and quantity of stores carried by the aircraft. The menu has the following options:

- "● 500 lb Bomb
- 1000 lb Bomb
- Missile
- Fuel tanks"

The user ACCEPTs the desired type of store from the menu and then uses the number pad to describe the quantity to be carried on the mission. The number selection then appears to the right of the menu item. When the user presses the RETURN key, the numbers of each desired store type are entered in the master data file. If the user wishes to change the quantity of a previously selected type, then the user indexes to the desired type and enters the desired quantity on the number pad. This overrides the previously stored number.

(*) 2.1.3.2 Key "CHANGE PATH". The CHANGE PATH key allows the user to select a waypoint that he intends to alter. The following message appears:

-- SELECT NODE --

USE TRACK BALL TO POSITION THE CURSOR OVER THE
DESIRED NODE AND PRESS THE 'ACCEPT' KEY.

Figure 4 shows a path which has been manipulated using the CHANGE PATH mode.

(*) A cursor which is moved by the trackball appears in the scenario. When the cursor is moved over one of the nodes, the node blinks and the ACCEPT key is activated. Pressing the ACCEPT key displays the menu:

- "● Add Node
- Delete Node
- Move Node
- Change Leg Speed
- Change Leg AGL
- Change Stores"

A menu cursor moves to the next option in the list each time the REJECT key is pressed. When the cursor is at the option desired, pressing the ACCEPT key will select it. The only option available for the recovery node is "Move Node." The CHANGE PATH key is active only if the launch, target, and recovery points exist.

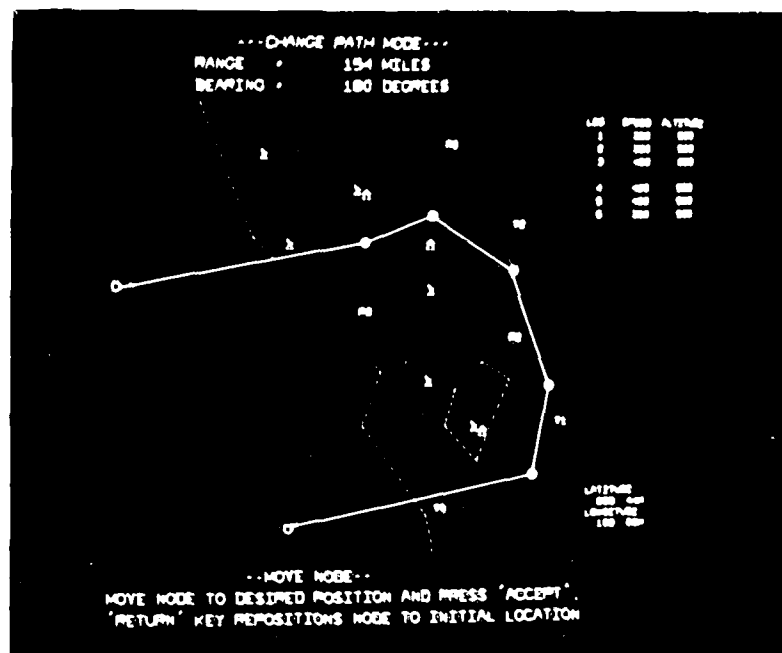


Figure 8. A Path Which Has Been Manipulated Using Change Path Mode

(*) Add Node

When this option is selected, a new waypoint appears halfway between the current node and the next along the flight path. Also, a position in the Leg/Speed/AGL Table appears for the leg following the new waypoint. (See Table 2 for an illustration.) In general, this is used in conjunction with Move Node to position the new node. After a node has been added, the user returns to CHANGE PATH.

(*) Deleting the last selected node can be accomplished by ACCEPTing the Delete Node option from the menu. This causes the node to be deleted, the path connected to its neighboring nodes, and the corresponding leg eliminated from the Leg/Speed/AGL Table. (See Table 3 for an illustration.) The Launch, Target, and Recovery points cannot be deleted. Immediately after Delete Node is ACCEPTed, the user returns to CHANGE PATH mode.

Table 3. Illustration of What Happens When a Waypoint is Deleted.

<u>OLD TABLE</u>				<u>NEW TABLE</u>			
(Node 3 is to be deleted.)							
Leg	Spanning Waypoints	Speed	AGL	Leg	Spanning Waypoints	Speed	AGL
1	$N_1 - N_2$	480	500	1	$N_1 - N_2$	480	500
2	$N_2 - N_3$	300	200	2	$N_2 - N_4$	300	200
3	$N_3 - N_4$	360	100	3	$N_4 - N_5$	420	500
4	$N_4 - N_5$	420	500				

(*) Move Node

When this item is accepted from the menu, the node to be moved remains blinking and a message below the scenario appears:

-- MOVE NODE --

MOVE NODE TO DESIRED POSITION AND PRESS 'ACCEPT.'
'RETURN' KEY REPOSITIONS NODE TO INITIAL LOCATION.

Table 2. Illustration of What Happens When a Waypoint is Added.

OLD TABLE (A new node is to be added between nodes 3 & 4.)

Leg	Spanning Waypoints	Speed	AGL
1	$N_1 - N_2$	360	1000
2	$N_2 - N_3$	420	500
3	$N_3 - N_4$	480	200
4	$N_4 - N_5$	540	1000

NEW TABLE

Leg	Spanning Waypoints	Speed	AGL
1	$N_1 - N_2$	360	1000
2	$N_2 - N_3$	360	1000
3	$N_3 - N_{3.5}$	480	200
4	$N_{3.5} - N_4$	480	200
5	$N_4 - N_5$	540	1000

(*) If an attempt to add a node too close to neighboring nodes is made, an error message appears:

-- ERROR --

WAYPOINT NOT ADDED SINCE IT LIES TOO CLOSE TO
SURROUNDING WAYPOINTS!

PRESS 'RETURN' KEY TO CONTINUE...

(*) If an attempt is made to add more than ten way points, the following error message appears:

-- ERROR --

THE MAXIMUM ALLOWABLE WAY POINTS -10- HAS
BEEN EXCEEDED.

PRESS 'RETURN' KEY TO CONTINUE...

If either of these messages appear, pressing RETURN automatically puts the user into CHANGE PATH mode.

The node can be moved around the map with the trackball. Pressing the ACCEPT key fixes its location and returns the user to the CHANGE PATH mode. If the user moves the node but wishes to return it to its original position, then the RETURN key should be pressed. The node will return to its starting position and the user will be in the CHANGE PATH mode.

(*) Change Leg Speed

When this option is ACCEPTed, the leg, preceding node, and corresponding entry in the Leg/Speed/AGL Table will blink and the following menu appears:¹

- "● SPEED I = 300 KNOTS
- II = 360
- III = 420
- IV = 480
- V = 540"

When the desired speed for that leg is ACCEPTed, the Leg/Speed/AGL Table will reflect the change in its corresponding entry, and the user is returned to the CHANGE PATH mode. If no change is desired, the RETURN key puts the user into the CHANGE PATH mode.

(*) Change Leg Altitude

This procedure is very similar to Change Leg Speed. The following menu appears:

- "● AGL I = 250 FEET
- II = 500
- III = 750
- IV = 1000"

¹ Pilots almost exclusively use integer multiples of 60 knots because time and distance calculations are simple to make. For example, 480 knots corresponds to eight nautical miles per minute. If a leg is 36 miles long, the time between turning points is 4 1/2 minutes. During the mission the pilot can therefore use his watch as an aid to determining when to turn. Thus, integer multiples of 60 knots are the only planning choices needed for the aid.

When an item is ACCEPTed, the corresponding altitude entry in the Leg/Speed/AGL Table is changed and the user returns to the CHANGE PATH mode. Pressing the RETURN key leaves the altitude unchanged and returns the user to the CHANGE PATH mode.

2.1.3.3 Inputs for Electronic Warfare (EW) Aircraft. Key "EW A/C" brings up the following menu:

"ROUTE

- Same Route as Attack a/c
- Different Route"

If the operator ACCEPTs "Same Route," then the program will check that speed constraints are not violated and store the same route for the EW aircraft as already planned for attack aircraft. If operator ACCEPTs "Different Path," then the operator defines the EW aircraft route in the same way as was done for the strike aircraft. At any point along the route the operator may activate or deactivate electronic warfare equipment aboard the EW aircraft. The program calculates the degradation effect of this equipment on each enemy defense system and the joint effect on overlapping systems.

2.1.3.4 Key "EVALUATE". There is an EVALUATE key. The user selects with the cursor, the beginning and ending node for which a Figure of Merit (FOM) will be calculated. The computer then calculates the value of the FOM for the user's Iterative Manual Optimization (IMO) solution and displays the local time of arrival at each waypoint, the fuel consumed for each leg, and the FOM for each leg.

2.1.4 Operator Aided Optimization (OAO)

2.1.4.1 Overview of OAO. Instead of using the IMO trial-and-error method of finding the best path, speed, and AGL to the target, the operator can have the computer use dynamic programming procedures to calculate the optimal route. First, the user defines a corridor boundary within which the optimization is to occur.

Next, a grid whose size is determined by the operator, is superimposed over the region. Finally, constraints on the speed and AGLs over the region being optimized are input by the operator. The optimal route between grid points for the given constraints is calculated and displayed by the computer. The operator can then modify the route (with the CHANGE PATH key) to make it practical, yet follow the guidelines set by the optimal path. This user/machine interaction has led to the name "operator aided optimization." The operator can use the Δ FOM key to ask the computer to define a region within which the routes are less than optimal, but still within an acceptable percentage of the optimal FOM.

2.1.4.2 Key "OPTIMIZE". The following, (a) - (d), happen in sequence:

(a) "Corridor Boundary"

The user first must select an entry and exit node on an already-defined route. A closed contour is drawn using the trackball and ACCEPT key. The entry and exit nodes exist along the boundary. When the contour is finished, the message, "Accept or Reject Corridor," appears. If the operator presses REJECT, the boundary is erased from the screen and the user must define a new corridor. If the ACCEPT key is pressed, then the corridor is set and the user then proceeds to the next step.

(b) "Grid Size"

The operator must now select a grid size from the following menu:

- "● Coarse
- Medium
- Fine"

The coarse size yields a solution which is obtained more quickly but is less precisely optimal than the others. Now the user must specify the speed constraints.

(c) "Speeds"

The following menu appears:

- "• 300
- 360
- 420
- 480
- 540"

All speeds ACCEPTed are considered by the dynamic programming algorithm when it is calculating the optimal solution, i.e., different legs may have different speeds, chosen from the list of ACCEPTed speeds.

(d) "AGL's"

A menu of AGL's now is displayed. All AGL's ACCEPTed by the operator are considered by the dynamic programming algorithm when it is calculating the optimal solution.

The computer now calculates and displays the optimal plan between the corridor entry and exit points for the set of constraints (corridor boundary, grid size, speeds, and AGL's). The display shows the route, AGL for each leg, speed for each leg, and the cumulative values of the Figure of Merit (FOM) at the end of each leg. The "optimal" plan will usually not be one that is practically navigable from a strike leader's point of view. Therefore, the planner must now input a modified plan. This is done by entering the CHANGE PATH mode and inputting a feasible approximation to the "optimal" plan.

2.1.4.3 " Δ FOM". Pressing this key brings up the prompt, "Use number pad to input acceptable degradation percentage in FOM." When the user has input the desired percentage, the computer calculates and draws the boundary around the region that contains solutions whose FOM values lie between (a) optimal and (b) optimal - Δ FOM.

2.1.5 Operator Aids

The ASPS program provides several operator aids to be used when needed. They don't affect the analysis but will be useful as aids for planning the airstrike. These aids can be categorized into two groups: modal and non-modal.

(*) The non-modal aids are the R&B key and DECLUTTER key. The R&B key allows the user to identify the distance and angle of the cursor from a reference point. The DECLUTTER key allows the user to blank out contours and/or significant points from the screen.

The modal aids are the Histogram Mode and Contours Mode. The Histogram Mode shows a histogram of the fuel consumed and FOM for each leg along the flight path. The Contours Mode allows the user to display contours that show for a desired AGL the outer limits of defense capability for hostile sensors, surface to air missiles (SAM's), and anti-aircraft (AA) guns.

(*) 2.1.5.1 Key "R&B". The RANGE/BEARING key is active whenever the operator is in the mode that uses the cursor. The following message appears:

Above the scenario:

RANGE = MILES
BEARING = DEGREES

Below the scenario:

-- RANGE AND BEARING --

SELECT REFERENCE POINT WITH CURSOR
PRESS 'ACCEPT' KEY TO FIX REFERENCE POINT
PRESS 'RETURN' KEY TO EXIT.

The cursor is moved with the trackball to the desired reference point and the ACCEPT key is pressed.

(*) As the cursor is moved, a line is attached from the reference point to the cursor. The range (in nautical miles) and bearing (from north) are measured from the cursor to the reference point and are continually updated and displayed.

This process continues as the user proceeds in the mode from which RANGE/BEARING was called. The range and bearing are turned off when the cursor position is ACCEPTed or the mode has been exited. If the user is in the CHANGE PATH mode and has selected the "Move Node" option, the displayed range and bearing figures are calculated from the moving node instead of the cursor. (See Figure 5.)

(*) 2.1.5.2 Key "DECLUTTER". There is a "DECLUTTER" key. This key allows the user to display selected contours and significant points. The "Declutter Key" works on a cycle of four presses. Each press will, in turn, perform the following functions:

1. Shows Terrain, Significant Points and Contours
2. Shows Terrain and Significant Points
3. Shows Terrain and Contours
4. Shows Terrain Only.

2.1.5.3 Key "HISTOGRAM". After the computer has calculated the value of the FOM for a plan, the user may choose the histogram option by pressing the key with this name. The histogram will graphically convey to the user the value of the FOM and the fuel consumption for each leg.

2.1.5.4 Key "CONTOURS". This key enables the operator to see displays of defense capability contours. The options are given on the following menu:

"Select Contour Option

- Sensors
- SAM/AA
- Both
- Composite
- Composite, SAM, AA"

After selecting the option, the operator uses the number pad to select an AGL between 1 and 4. Figure 6 shows contours for both SAM and sensor sites. If no contour exists for a particular option at a certain AGL, the program returns to the Director Mode.

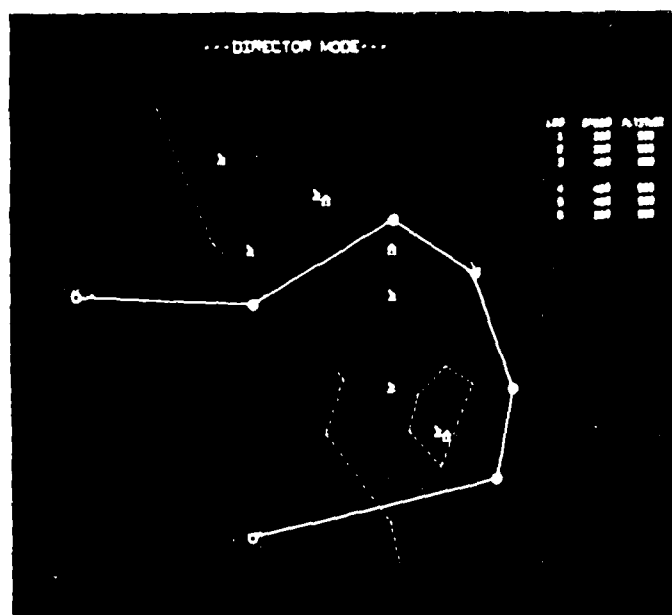


Figure 5. Illustration of Range and Bearing Use During 'Change Path' Operations

Sensors Option - This allows the user to select individual sensors to display or turn off their contours, or to turn on/off all the contours.

SAM/AA Option - The user sees all threat level contours for each of the SAM/AA sites at the given AGL.

Both - This is a combination of the above two options.

Composite - This shows the composite contours for all the sensors.

Composite, SAM, AA - This option displays the composite along with the SAM and AA contours.

2.1.6 Data File Communication

(*) The following keys are the only keys which interact with data files stored on disk. The STORE key takes the current plan and puts in onto disk, the RECALL key brings up a previously constructed plan from disk, and the EXIT key updates the Admin data file (see Appendix B).

(*) 2.1.6.1 Key "STORE". Pressing this key allows the user to save work done. When this key is pressed, a message appears below the screen indicating the available plans and the current plan number. "Plan" 1 is restricted to be the map only and is not available for storage. As many as five plans for each map can be saved. Caution: If a plan is stored in a plan number which is different from the one where it was recalled, then information previously residing in the second plan location will be lost. For example, if a plan is recalled from plan location #2, revised, and stored in plan location #4, then information previously stored in plan location #4 will be lost.

(*) 2.1.6.2 Key "RECALL". There is a RECALL PLAN key. When this key is pressed, the following message appears:

LAST PLAN IS NUMBER (e.g., 4)
BEST PLAN IS NUMBER (e.g., 3)
SELECT PLAN (1-N) USING NUMBER PAD

N is the number of plans available for recall. The operator uses the number pad to select the plan number he wants to see. The procedure is similar to that described in Subsection 2.1.1, Beginning Operations. If the RETURN key is pressed, the previous plan is displayed.

(*) 2.1.6.3 Key "EXIT". The EXIT key is used at the conclusion of the planning session. It serves two purposes:

1. To allow for normal termination of the ASPS program
2. To update the "Admin" data file.

It is important to use this key when finished, otherwise the next planning session may not properly reflect the current session's work.

2.1.7 System Calculation Capability

The system will have the following calculation capabilities:

1. Fuel consumption for a particular aircraft will be calculated as a function of speed, altitude and drag count for each leg of a route. (The model for this is obtainable from COMLATWING at Lemoore NAS.)
2. System will calculate time of arrival at each waypoint in terms of local time and time from launch.
3. The Figure of Merit (FOM) will be calculated for each user-input (IMO) solution. This will include times of arrival of waypoints, FOM for entire route, and elements of FOM (e.g., probability of survival) for each leg of a route.
4. System will compute the joint capability of enemy search radars at any point on the terrain.¹
5. System will compute the joint capability of enemy missile sites at any point on the terrain.¹
6. System will compute the joint capability of enemy AA guns at any point on the terrain.¹
7. System will compute degradations to radar and missile performance as a function of EW aircraft position over time of mission.

¹The dimensions of capability for each type of defense site are a matter to be decided in consultation with end users.

8. System will compute the dynamic programming solution between a user-designated pair of points after the user has input the constraints.

9. System will use the dynamic programming algorithm to compute and display the corridor containing solutions ΔFOM poorer than the "optimal" solutions.

10. System will store the four most recent solutions and the best solution to date.

2.2 CONCEPT FOR A FIELDABLE VERSION OF ASPS

As noted earlier, this subsection describes:

1. Capabilities that would be different in the fieldable version from the corresponding capabilities in the laboratory version described in Subsection 2.1. Consequently, descriptions of capabilities which would be the same in the fieldable version as in the laboratory version are not repeated here.

2. Capabilities that strike planners desire but that are not included in the laboratory version because they are not needed to test hypotheses about OAO and IMO in a laboratory experiment.

2.2.1 Problem Setup

2.2.1.1 Type of Scenario. At the beginning of a session, the planner will see a prompt on the display:

"Type of Scenario

- Land Target
- Sea Target"

If a land target is ACCEPTed via the function keyboard, then the planner proceeds to select the desired map and then performs the initialization step covered in Subsection 2.2.1.2.

If a sea target is ACCEPTed, the user must then establish an enemy order of battle. This is done by selecting an enemy ship type from a menu and then positioning the symbol for that type on the display with the display peripheral, e.g., trackball or joystick. The latitude/longitude readout of the symbol at the lower right of the display will enable the planner to position the ship. This process is repeated for each ship in the enemy formation or disposition.

The planner may now define single-ship movements or the movement of a multiple-ship formation. To define a single-ship movement, the planner locates nodes of the route with the trackball or joystick. When the route is completely drawn, the arrival time at each node is selected by inputting a speed of advance in knots on the number pad. To define a formation movement, the planner first designates the members of the formation with the trackball or joystick and then designates the formation guide. The formation route is drawn from the location of the formation guide in the same manner as for a single-ship movement and formation speed is also specified the same way. Routes parallel to the guide ship are automatically defined for the other ships in the formation.

2.2.1.2 Initialization. The locations of enemy airfields are added to the terrain data in the appropriate map file. Types of interceptors at each airfield are stored and associated with the airfield's location. All other elements of initialization remain the same as defined for the laboratory version of ASPS.

2.2.1.3 Digitized Maps. The Defense Mapping Agency (DMA) has digitized map products (Reference 5). Off-line processing of this digitized data would be done using an already-available algorithm to calculate and store topographic contours and coastline for a region of interest. Thus, when a planner requests a particular map and scale, the display shows the topographic contours for the region. (Subsection 3.1 has additional information about DMA map data and their algorithm that calculates topographic contours.)

2.2.1.4 Display of Terrain Features and Defense Capability Contours. The contours for roads, rivers, lakes, and cities in a region of interest will be drawn off-line prior to a planning session and the points defining those contours will be superimposed on the topographic contour data. Radar significant points (RSP's) and Visually Significant Points (VSP's) with user-defined labels such as { VSP3 } will be designated either off-line by an operations or intelligence { Tower 1 }

staff member or on-line by the planner. They will be superimposed on the other terrain data.

Calculations of each site's defense capability contours¹ will be based on the site's basic range envelope and the effects of terrain masking around the site. The calculation will be done on-line providing that the execution time does not cause a significant delay for the planner. However, calculation and display of joint capability contours for multiple sites will be done on-line. (Several algorithms for calculating terrain masking from a particular point using DMA data are available. See Subsection 3.1 for additional information.)

2.2.2 Variable Background

The variable background elements for the laboratory version of ASPS are:

- Visually significant points (Declutter priority II)
- Radar significant points (Declutter priority II)
- Detectability contours for radars (Declutter priority III)
- Reachability contours for missiles and AA weapons (Declutter priority III)

For land targets, the fieldable version of ASPS will have a display of a reachability circle around each enemy airfield that corresponds to the combat radius of the interceptor type designated at that airfield. For the war at sea scenario, the display will show the detectability contours for seaborne radars as a function of altitude and the reachability contours for seaborne guns and missiles. Both displays will have declutter priority III.

2.2.3 Route Planning

The first step in the route planning process will be designation of aircraft type used on the route. The alternatives available in a production version of ASPS would probably include all tactical aircraft in the Navy's inventory. A

¹The dimensions of capability for each type of defense site are a matter to be decided in consultation with end users.

version for field test and evaluation would probably include a subset of these, such as A-6, A-7, and EA-6B. Depending on the type selected, the user will next designate the mission, e.g., attack or tanker. (The mission is automatically electronic warfare when EA-6B is selected.) When the tanking mission is selected, the tanker aircraft automatically receives a standard fuel load and weapon stores (if any) for that mission.

If the attack mission is designated, then the next step is selection of stores. Again, a production version of ASPS would probably include all stores available for the aircraft types handled by the aid. A version for field test and evaluation would probably have a subset of these, such as 500- and 1000-pound bombs, an air-to-surface missile, external fuel tanks, and an anti-radar missile (ARM's). All of the above steps are performed using the function keyboard and displayed menus in the same manner as described in Subsection 2.1.

The planner will then plan the route of the designated aircraft type and mission using the basic procedures for interactive manual optimization (IMO) and/or operator aided optimization (OAO) as described in Subsection 2.1. Additional features in the fieldable version of ASPS are:

1. The planner may designate a point along the route where an anti-radar missile is fired.
2. The planner may designate a point where a refueling takes place. The computer transfers an amount of fuel to the attack aircraft which is lesser of (a) fuel needed to top off the attack aircraft or (b) fuel aboard tanker aircraft less fuel needed by tanker aircraft to return to carrier.

At any time after a complete route is established, the planner may elect to see a dynamic (movie-like) replay of the flight over mission time. This will include the flight of ARM's as well as aircraft. The planner may store a plan for one aircraft type and begin to plan a route for another aircraft that overlaps (in time) the mission of the previously planned route(s). The line segments displaying the geographic route of a previous plan remain on the screen while the next plan is being devised. When the planner elects the dynamic replay option

after the n^{th} mission has been planned, there will be an option to see the dynamic replay of only the n^{th} mission or the simultaneous replay of all missions planned for this map area thus far.

The planner will also have the option to designate a specific defense site which is to be deleted from the enemy's joint capability. A revised Figure of Merit (FOM) for the plan can then be calculated and compared to the FOM when the enemy has full capability. This feature will help in assessing need or desirability of making a particular site the object of an attack.

2.2.4 Air Strike Target Area Planning (ASTAP)

The basic elements of the ASTAP mode of ASPS are shown in Figure 7. The air strike planner brings to the problem the types of data indicated. These data are mission-specific and must be input to the system to begin the planning session. The synoptic data relevant to strike planning in general are already contained in the system bulk memory. It is accessed as needed by the specific planner-supplied mission parameters. During the problem initialization phase, the planner is concerned only with tactical variables important to the mission.

When the problem is initialized, the planner begins by asking a series of "what if?" questions in an arbitrary sequence. ASTAP does not force the planner to follow a canned sequence of procedures. Instead, the planner may work through the solution process using the analysis and simulation tools provided in a sequence that seems natural to him. Aspects of the problem that require more analysis may easily be given the required attention emphasis.

When a promising plan (or partial plan) is obtained, the planner labels and stores it for future recall. An ASTAP plan consists basically of the following:

- Aircraft (A/C) tracks
- Ordnance release points
- Ordnance effects
- Survivability data

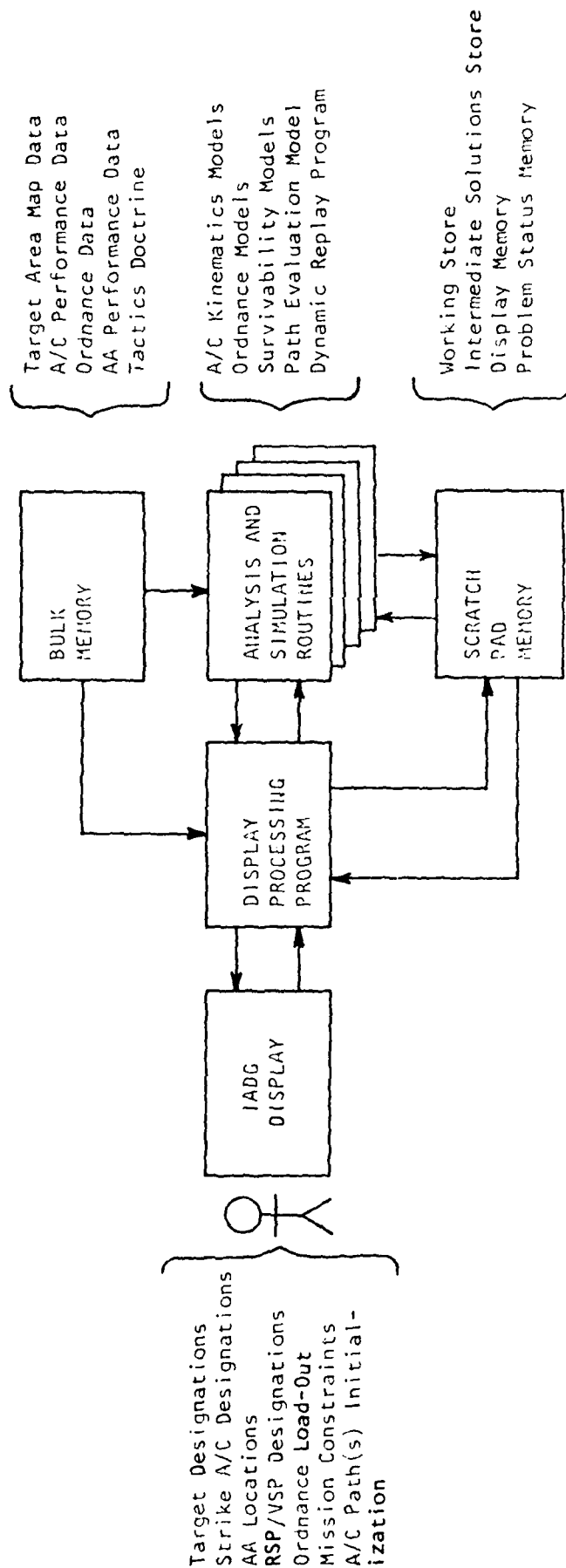


Figure 7. ASTAP Overview.

An example of the format is shown in Figure 8. Only a small number of display elements are presented to limit clutter. Various combinations of display elements (such as aircraft tracks, AA effectiveness contours, levels of map data) may be displayed by means of a special function key.

Dynamic replay of a plan shows the relative positions of the strike aircraft and their ordnance effects (blast debris, wind-driven smoke) in both time and space. The replay may be stopped at any point to examine relative aircraft positions, to assess safety aspects and examine potential problems due to arrival too early or late by designated aircraft.

Figure 8 is a representation of a strike involving three aircraft paths aimed at destroying buildings (T1) and a bridge (T2). The target area is defined by two AA installations whose radii of fire at the mission altitudes is shown. The double tracks indicate above-ground level (AGL) altitudes in excess of a designated threshold, e.g., 200 feet.

The basic plan calls for the aircraft along Path 1 to arrive first, staying outside the AA radii as shown. The predicted pop-up and ordnance release points (black squares) are computed by ASTAP using planner input/deployment maneuvers, aircraft speeds, ordnance type, and approach heading. The AA installations will have their attention diverted when aircraft along Path 2 do a pop-up along the ridge and first hit AA1. They continue along the shown path to neutralize the AA2 installation. The aircraft whose prime target is the bridge (T2) will follow closely behind. The intention is to time the strike so that the Path 3 aircraft are close enough to help swamp the air defenses, but not close enough to interfere or be exposed to interferences by the blast and smoke of Path 1 aircraft. The adequacy of the specific attributes of the plan are determined by the planner during dynamic display. Modifications to the plan are made to study the sensitivity of any part of the plan whose successful execution is dependent upon executing the plan within a timing or navigation accuracy window.

This brief description gives just a limited glimpse of the possibilities of this concept. A partial list of ASTAP functions and capabilities is in Table 4.

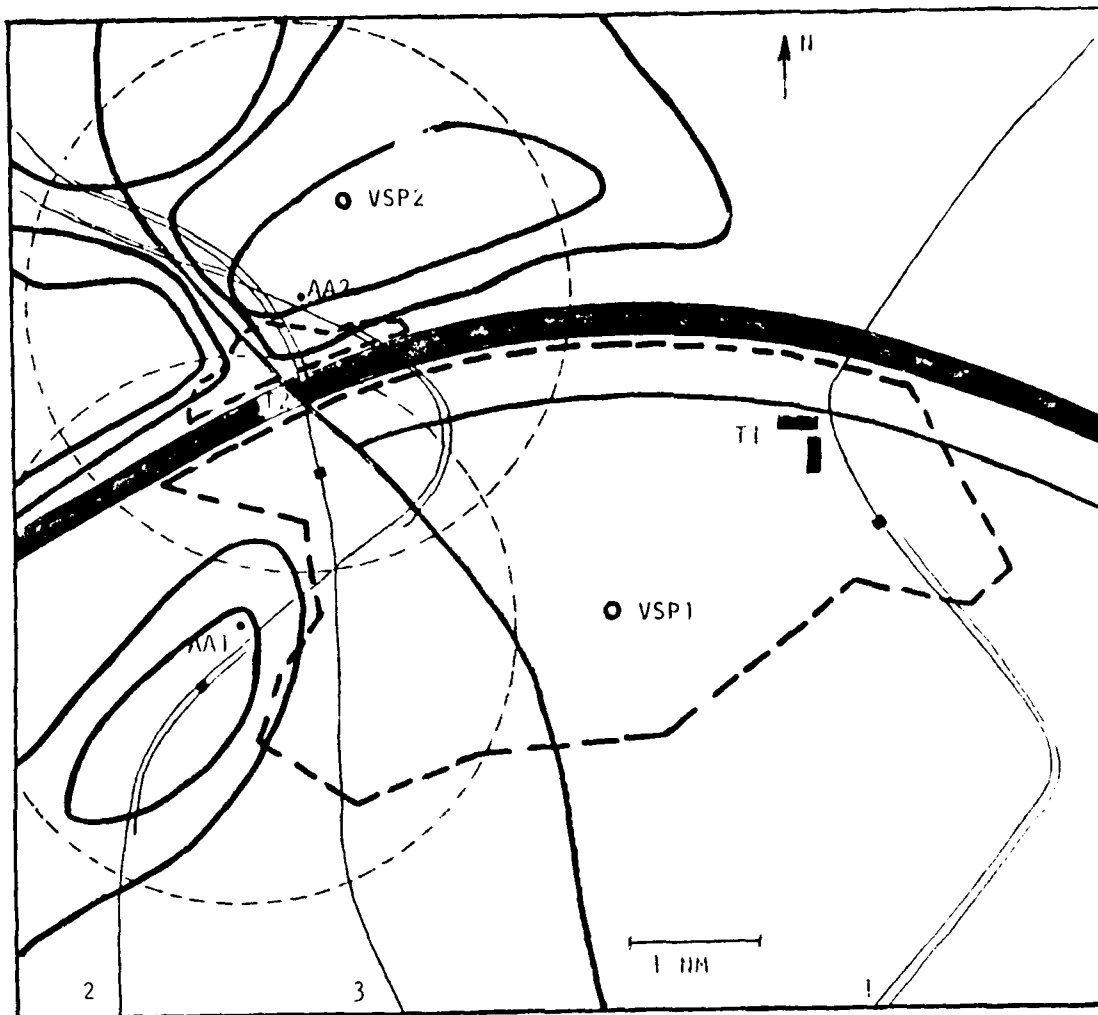
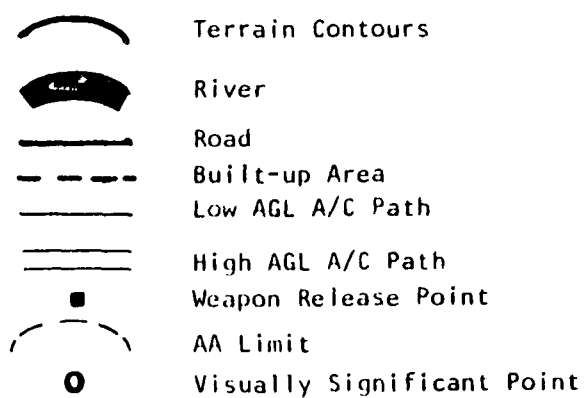


Figure 8. ASTAP Display Elements.



NOTE: This shows examples of all the types of displayed information. They would normally not all be displayed simultaneously.

Table 4. ASTAP Functions.

- 3D path simulation from "distilled" large scale models and test results.
- Ordnance release point computation
- Ordnance effects computation
- 2D A/C path designation input and replay
- Survivability computation as a function of A/C paths and AA types and locations
- Computation of sighting data (ranges and bearings) to operator-designated points to coordinate target area navigation
- Computation and display of AGL values as a function of designated A/C speeds, horizon plane tracks, and type of ordnance deployment maneuvers.
- Computation and display of ordnance CEP's as a function of attack parameters
- Computation of fuel consumption as a function of changing drag count and A/C kinematics
- Dynamic replay of current and stored alternative plans with stop-action and add-on analysis capability.
- Automatic monitoring of user input constraints and constraint violation alerting

2.2.6 Hard-Copy Outputs

The planner will choose the preferred plan from those analyzed at the conclusion of a session with ASPS. The user may request a hard copy of the plan elements in the standard format on a knee-pad-sized sheet of paper. The printed output will include at least the following:

- Geographic coordinates and time of arrival at each turning point on the path
- Speed and altitude for each path leg
- Geographic coordinates, time of arrival, and altitude at a refueling point
- Loadout of weapons and external fuel tanks
- Coordinates of ordnance release points

3.0 INVESTIGATIONS OF CRITICAL ASPECTS OF ASPS DEVELOPMENT

Creation of make-believe terrain is acceptable for a laboratory version of ASPS, but a fieldable ASPS must use terrain data from real maps. Consequently, ISC conducted an investigation to learn how digital terrain data might be obtained and what algorithms are available for performing certain calculations on such data. The results of this work are given in Subsection 3.1.

Our ONR sponsor and air strike planners consulted by ISC expressed a strong preference for low-cost, highly portable ASPS hardware. ISC investigated desk-top, stand-alone computer and display systems and systems consisting of separate, but highly portable computer and display subsystems. The results of this work are in Subsection 3.2.

3.1 DIGITIZED MAPS

There are basically two methods of obtaining digitized terrain data:

1. The do-it-yourself method.
2. Use of digitized map products from the Defense Mapping Agency (DMA).

The first method is simply manually recording the terrain properties of a region on a cell-by-cell basis according to a predetermined data format for each cell. This is the only method available if DMA does not have a digitized map for an area of interest and the end user needs the map before DMA can meet a request for it. The Marine Corps training unit that operates the Tactical Warfare Simulation, Evaluation, and Analysis System (TWSEAS) at Camp Pendleton occasionally does its own digitizing of maps which are used in TWSEAS.

DMA digitized map products are derived from the Digital Landmass System (DLMS) Data Base (Reference 5). This data base is collected at two different levels which are defined as follows:

Level 1

1. Terrain: Relief information in DMA standard digital format with each cell three seconds of latitude (303.81 feet) by three seconds of longitude.

2. Cultural features: A generalized description and portrayal, in DMA standard digital format, of planimetric features. The Level 1 data base is intended to cover large expanses of the earth's surface and has relatively large minimum size requirements for portrayal of planimetric features.

Level 2

1. Terrain: Relief information in DMA standard digital format with each cell one second of latitude (101.3 feet) by one second of longitude.

2. Cultural features: A highly detailed description and portrayal, in DMA standard digital format, of planimetric features. The Level 2 data base is intended to cover small areas of interest and has small minimum size requirements for planimetric features.

An important consideration for ASPS development is the availability of already-developed computer programs for calculating topographic contours and for calculating masking effects of terrain around a given point. DMA has two applicable programs. One program, filed as "ULB 427" at the DMA Aerospace Center, St. Louis, accepts the DMA standard format for digitized terrain elevation data and produces a topographic contour map of the area. The other is a subroutine called "RTMASK" within the program filed as "ULB 437." RTMASK uses the output of ULB 427 and produces a plot that, given an observer's geographic position, will indicate the visibility afforded to that observer.

Information supplied by the DMA Aerospace Center led to several other sources of algorithms for calculating masking effects of terrain. One of these sources is work done by Pattern Analysis and Recognition Corporation, Rome, New York, for the Rome Air Development Center. Reference 6 documents

their two FORTRAN algorithms for determining what areas on the ground are not detectable from an airborne radar due to terrain masking.¹ One of the algorithms uses a perspective projection approach and the other uses a ray tracing approach. DLMS Level 1 data from DMA are the input to their algorithms and the algorithms account for the effects of earth curvature and atmospheric refraction of the radar beam. Core requirements and execution speeds on an XDS Sigma V computer for several sizes of geographic areas are given in Reference 6.

Lincoln Laboratory at the Massachusetts Institute of Technology has a set of programs for calculating the effects of terrain masking on ground-based radars (Reference 7). These programs are written in PL/I for execution on an IBM 370 computer; they also use the DLMS Level 1 data as input. The terrain elevation data in the DLMS data base is converted from its matrix form to a polar form centered on the radar site. This is done for each elevation in the polar grid by interpolating from the four nearest elevations in the matrix grid. One drawback to the use of polar representations of data is the reduction in the density of data points as range increases from the origin. Compensation for this is made by increasing the azimuthal density of data points as range increases from the origin. The program then does a simple geometric calculation to determine those points in the polar grid for which the radar's line of sight is blocked. Inputs to the program include radar antenna height above ground and aircraft altitude above ground level. The program also accounts for earth curvature and atmospheric refraction of the radar beam. Program output indicates for each point on the polar grid *whether or not the aircraft is masked*.

Another terrain masking algorithm is available from a large-scale computer simulation of battles between ground based air defense systems and several weapons systems. This simulation is known as TACOS II (Reference 8).

¹ It is clear that, if a point on the ground is not observable from an aircraft, because of terrain feature obscuration, then the reverse is also true. Consequently, these algorithms are applicable to the problem of calculating terrain masking from ground defense sites. Radar refraction/ducting effects are not considered a significant advantage over visual reciprocity.

It was developed by the U.S. Army Combat Developments Command Air Defense Agency and Braddock, Dunn, and McDonald and it runs in FORTRAN on a CDC 6600 computer. The terrain masking algorithm is just one of many routines in TACOS II. Input to the algorithm is a matrix of terrain elevations taken at 500-meter intervals along each row and rows are spaced 500 meters apart. (This is not DLMS data.) The algorithm calculates the "Dominant Mask Function," i.e., the region of visibility within an air defense site's maximum range. On each radial from the site, the algorithm finds the elevation angle to each visible ridge and determines the ground distance beyond each ridge and along the radial where masking occurs. Figure 9 illustrates the idea. The algorithm accounts for curvature of the earth.

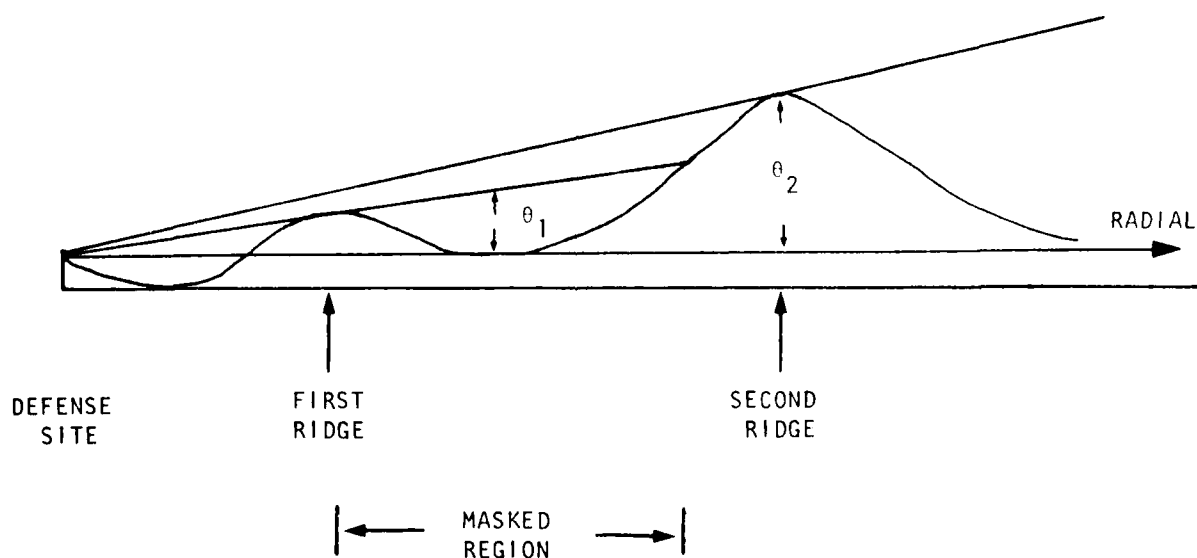


Figure 9. Illustration of a Masked Region Along a Radial from a Defense Site.

Thus, the results of our investigation of digitized maps show that:

1. There is a large and growing body of DLMS Level 1 digitized terrain data available from DMA which is suitable for use in ASPS.
2. There are several existing algorithms for calculating topographic contours and for calculating masking effects of terrain around a given point. It is reasonable to believe that algorithms suitable for ASPS can be devised by modifying ones already in existence.

3.2 INVESTIGATION OF HARDWARE ALTERNATIVES FOR ASPS

ISC investigated the products of 16 manufacturers of graphics terminals to find the best hardware available for the desktop version of ASPS. We restricted our investigation to the relatively large firms to better insure future availability of both service and parts. The major considerations in our analysis were:

- computation speed
- cost
- coding capabilities
- peripheral availabilities
- expandability of system
- ease of displaying and programming dynamics
- ease of entering map data

3.2.1 Computation Speed

The algorithms that are needed for the optimization modes of ASPS ("Rising Water" and other dynamic programming algorithms) involve a great deal of computation. Execution speed, therefore, is of great importance. Most systems which ran only the BASIC computer language were eliminated. This is because BASIC is an interpretive language, i.e., every line of code must be re-interpreted each time it is executed and this increases computation time considerably. FORTRAN, on the other hand, is a compiler language which is translated into machine language and, thus, each line of code is interpreted only once.

ISC previously developed defense penetration planning aids which used both "Rising Water" and other dynamic programming algorithms. These programs were implemented on a Varian V-73 computer. The execution speed of the V-73 was used as a benchmark to determine the relative speed of the various systems examined. We wrote a program which consisted of code similar to that used in the aforementioned algorithms. These programs were run on the V-73 and on the other systems examined. The V-73 executed the program in the fastest time (20 seconds). The other times are listed in the summary sheets of each system's characteristics which are shown below.

3.2.2 Raster vs. Vector

Two basic display technologies are available for the desktop versions of ASPS, namely, vector and raster graphics. Vector graphic images are produced by a "gun" which draws on a tube at very high speed. Raster graphics produces an image consisting of many points or pixels (currently as many as a million), each of which may be chosen to be a certain color. Using vector graphics, it is simple to create symbols and change their location frequently enough to provide smooth movement on the display. It is also an easy matter to blink these symbols or change their color. Animation is possible in raster graphics, but some fluidity of motion may be lost. Raster graphics facilitate colored background and shading where only lines and circles may be drawn in vector graphics. We principally examined raster graphic systems since this technology is more widely used in both the military and non-military sectors than is the vector-based technology. Also, raster systems currently sell for \$30,000 - \$160,000, while the vector-based systems generally begin at \$60,000. Color and shading provide an important coding dimension, and all raster systems supply this.

3.2.3 Peripherals

ASPS requires an input device which allows the user to designate a location on the screen with ease. A joystick or trackball is a good choice since either facilitates a continuous stream of updated locations on the screen to aid in the updating of a path. Special function keys would also be necessary to allow the user to change modes in ASPS. These keys would be clearly labeled, and would have lights associated with them which can be turned on when activated.

3.2.4 Map Data

We have explored various methods for displaying map data. One method requires digitizing map features and placing it in a data file. This file is subsequently read and displayed on the screen. This method can produce good terrain detail, but is slow to display and requires an enormous amount of storage. A second method employs a 35mm slide of the area which can be displayed through special equipment. This may be impractical, if not impossible, when the computer does not have the capability to accept the extra equipment required for this method. A third method is to store topographical contours in coded form in a data file. This seems to be the most practical and manageable method for displaying the region in question and is certainly adequate for ASPS. Using this last method, the terrain data can readily be held in a data file and displayed equally well on all systems under review.

3.2.5 Dynamics

The current ASPS work does not require a great deal of dynamics on the screen. The users need to alter a path which does not require an extensive amount of dynamic capability. However, the ability to easily program dynamics may well have major importance for any follow-on work for ASPS.

3.2.6 Comparisons Among Five Candidate Systems

With these considerations in mind, ISC compared five candidate systems:

- Tektronix 4054
 - Hewlett-Packard 9845C
 - Genisco GOT-3000
 - Megatek 6250
 - Megatek 7250
- } coupled with the Digital Equipment Corporation (DEC) MINC 11/23 as a host computer

3.2.6.1 Tektronix 4054. This system is a stand-alone, interactive graphics computer. The single, compact console consists of a 19-inch, monochromatic storage tube, a full ASCII keyboard, twenty special-function keys, two thumbwheels (dials which are used to location positions on the screen), and a tape drive.

EXECUTION SPEED: The slowest of the four¹ central processing units whose speeds were measured (more than twice as slow as the MINC 11/23). The speed, however, would be sufficient for ASPS. Benchmark program execution time: 121 seconds.

COST: The entire system with an additional two floppy discs costs approximately \$30,000.

CODING: The greatest weakness of the system. We found lack of color to be a great disadvantage. Textured lines (e.g., dot-dash, dot-dot-dash) do not have the capability of color-coding to convey complex "scenes" of information.

PERIPHERALS: The essential peripherals are built into the basic console. Floppy disc drives, graphics tablets, and printer can be easily added.

EXPANDABILITY: Since the Tektronix was designed as a stand-alone system, it will not accommodate more than one work station.

DYNAMICS: This vector graphics system provides fluid movement of objects. However, it is difficult to have simultaneous movement of a number of objects.

MISCELLANEOUS: Editing and programming features are cumbersome and would add to the programming cost of ASPS if this system were used.

¹The benchmark routine was timed on the Varian V-73, MINC 11/23, Tektronix 4054, and H.P. 9845-C.

3.2.6.2 Hewlett-Packard 9845-C. The Hewlett-Packard (H.P.) 9845-C is a compact, stand-alone desktop computer with a color graphics raster display. The standard system comes with built-in hard-copy device, two tape drives, a keyboard with 8 special function keys (with use of the shift keys, this is effectively raised to 32), and a lightpen. Additionally, there are 8 software programmable keys which are conveniently located just below the CRT to allow labeling on the screen. Although this computer may only be programmed in BASIC, its computation speed is adequate for the requirements of ASPS.

EXECUTION SPEED: Half as fast as the MINC 11/23, but adequate for ASPS.
Benchmark program execution time: 86 seconds.

COST: This system with two floppy discs costs approximately \$46,000.

CODING: There are virtually an unlimited number of colors available of which 16 may be used at any one time. The resolution of 560 X 455 provides more than sufficient clarity for ASPS. Shading and filling of polygons is easily accomplished in the software.

PERIPHERALS: Floppy disc is available for an additional \$5,000. A trackball is not available from Hewlett-Packard, but may be obtained from a second source for \$3,000-\$5,000. A graphics tablet is available from H.P. which can be used in place of a trackball or joystick for about \$3,000.

EXPANDABILITY: This desktop unit was not designed as an expandable system. Only one work station could be supported from each computer.

DYNAMICS: The lack of easily programmable dynamics is a disadvantage of this system. None of the demonstration programs by H.P. showed any dynamics. This capability was not of high priority in H.P.'s design and, hence, programming dynamics is rather difficult and, in some complicated cases, impossible.

MISCELLANEOUS: The lightpen does not provide a facile means of altering a path. The editing capabilities on the H.P. 9845 are excellent and will facilitate programming.

3.2.6.3 Systems Using the DEC MINC 11/23. The next three graphic systems use the DEC MINC 11/23 as a host computer (see Figure 10). The MINC 11/23 can support two display monitors working simultaneously. FORTRAN is programmable on this machine and good editing features are available. It gave the fastest result of the benchmark routine of all systems under consideration, about 45 seconds.¹ A fast floating-point processor will soon be available which will further increase its speed and efficiency. The MINC 11/23 is extremely lightweight and portable, coming standard on a mobile cart where the components reside. Adding components is very simple and designed in such a way so that the user can't make a mistake. Data files could be created on other DEC equipment, such as the VAX, and used on the on-board system, which increases its flexibility. Also, the software developed for the MINC is compatible with the VAX.

There are some disadvantages to a component system, such as the MINC 11/23 combined with display monitor. Having a separate host computer and display monitor would require separate maintenance contracts and cabling between the components. Cabling would add a slight cost and get in the way. The operating system for the MINC is not resident in core, but is read in via a floppy disk. Much precaution must be observed to protect these diskettes. The MINC only accepts a Q-Bus connector to its peripherals. Most of the display terminals reviewed require a Uni-bus connector, but converters are readily available. The problems of the MINC 11/23 are really quite minor and would not hamper the operation of the ASPS project.

Megatek 7250 and DEC MINC 11/23. This combination is well suited for current and add-on ASPS work. The ease of programming dynamics is a great advantage to this system. The 512 x 512 resolution is quite adequate

¹Varian V-73 is not under consideration because of its size, but was used as a comparison for the benchmark times.

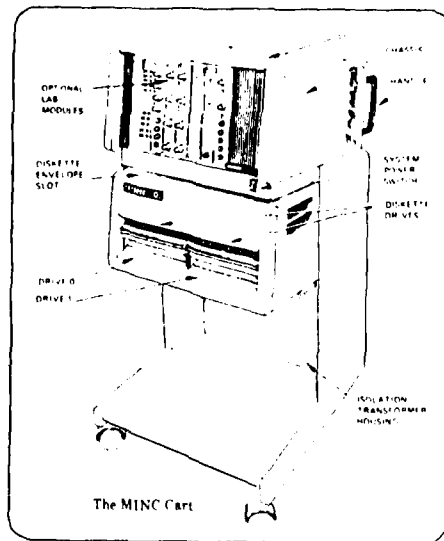


Figure 10. DEC Minc 11/23.

for the level of detail required for this project. This system has a pan and zoom capability built in the hardware. The analysis below is for the 7250 with a keyboard, joystick, 16 lighted function switches, and 8 control dials.

COST: MINC 11/23	Approximately \$18,000
MEGATEK 7250	<u>Approximately \$38,000</u>
	Approximately \$56,000

CODING: Up to 16 colors may be used at one time. Filling polygons is cumbersome and must be done in software. A hardware fill capability will soon be available.

PERIPHERALS: Hardcopy device (black and white), data tablet, and pick module (similar to a lightpen) are among the available options.

EXPANDABILITY: The heart of this system, the MINC 11/23, is well suited for expandability. Two work stations could be run from the computer.

DYNAMICS: The capability to easily program complex dynamics is the forte of the Megatek 7250. The data link from the MINC to this display is fast enough to support independent fluid movement of multiple symbols on a complex tactical decision aid.

Megatek 6250 and DEC MINC 11/23. This is a cheaper and less powerful version of the Megatek 7250 combination. The most noticeable loss is the complexity of dynamics which can be displayed. The MINC talks to Megatek 6250 over an RS-232 interface instead of the unibus. The result of this is that the complexity (number of vectors) which can be updated every frame is cut back drastically. However, the dynamic capabilities of the Megatek 6250 are adequate for the current version of the ASPS project although complex scenarios with multiple objects moving simultaneously might cause some difficulties.

The 6250 comes standard with a keyboard and joystick. The special function box is not an available option for this model. Another drawback of this system is that the only available display screen is just 13 inches across the diagonal.

COST: MINC 11/23
MEGATEK 625

Approximately \$18,000
Approximately \$20,000
Approximately \$38,000

Genisco 3000. This display system is a competitor of the two Megatek systems that have been previously reviewed. It would require a Q-Bus-to-Unibus converter to run off the MINC 11/23 computer. This is readily available. There are several monitors available with varying refresh rates and resolutions. The 512 X 512 resolution monitor has a maximum refresh rate of 30 Hz, whereas, the 640 X 480 resolution monitor can be increased to 40 Hz. There are other monitors available with better resolution and faster refresh rates, but the aforementioned monitors would be adequate for ASPS. If very complex movement is involved, then flickering or rough movement of objects might occur. Also, while moving an object around the screen, a ghosting effect (i.e., a slight trail of the image being moved) can occur. This is not envisaged to be a problem in ASPS.

The Genisco system can be programmed by either referencing pixels (used on raster systems) or specifying vector lines (used on vector systems). Genisco is best suited for image-processing, although dynamics can be easily performed, but not with the same clarity as a vector system. For the ASPS project, the dynamics have a major importance, while the image-producing capability has a smaller role and, hence, the Genisco 3000 would have more potential than what is required.

The Genisco 3000 has virtually unlimited color shading available on a single board which contains four bit planes. A trackball, joystick, and alphanumeric keyboard with sixteen lightable special function keys are easily

added. These special function keys can be used in conjunction with the shift and control keys to have 64 distinct functions. This system can be expanded to two monitors, if required.

COST: MINC 11/23	Approximately \$18,000
GENISCO 3000	<u>Approximately \$25,000</u>
	Approximately \$43,000

3.2.7 Recommendation

The Megatek displays are more suited to this project than is the Genisco because of the way dynamics are accommodated. We have not seen Genisco demonstrations of the type of dynamics which are required for ASPS and would be reluctant to recommend this display system.

Of the two Megatek systems, we recommend the 6250 if the current ASPS concept is to be implemented as now written. The 6250 could well support the requirements of the current ASPS system, but might not be suitable for follow-on work which involves more complex dynamics. The 7250, while more expensive, permits all the dynamics needed for a tactical decision aid. The special function key box which is available with this model is the input device most suited for this application. Of all the systems investigated, the Megatek 7250 most closely fits the needs of the current ASPS work, and would not restrict adding capabilities in the future.

All five of the systems examined are state-of-the-art systems. All except the Tektronix 4054 could be used for the on-board ASPS system. The Hewlett-Packard is a very impressive system. Its compactness and easy transportability is a true benefit as compared to multiple component systems linked to the MINC 11/23. However, difficulty of displaying dynamics, and the lack of a trackball or joystick make the H.P. a less desirable system than those driven by the DEC computer. The expandability of the DEC computer system may be important for ASPS follow-on work.

4.0 RECOMMENDATION

Procedures for planning Naval air strike missions are mainly done by manual processes at the present time. Planners spend much of the available planning time doing tasks such as:

1. Calculating fuel consumption for alternative routes and weapon loadings.
2. Measuring ranges and bearings from turning points to radar significant points or visually significant points.
3. Measuring courses and latitude and longitude at points on a map.
4. Recording a plan on a standard form.

Planners do not currently have the time to calculate terrain masking regions for different AGL's around defense sites nor are they able to calculate the effect of terrain masking on the probability of survival. All of the above tasks can be done by computer software whose use is controlled by the planner.

Development of a highly interactive ASPS is recommended because ASPS will enable planners to use their time more efficiently than is possible with current manual procedures and because cost of ASPS development and purchase will be low. The ASPS concept described in Subsection 2.2 will allow the planner to do all that is currently done in much less time per planning session. Alternatively, with ASPS the planner can evaluate many more alternative plans in the same amount of time that is currently used per session. Further, certain ASPS features will enable the planner to do analyses that are not practical at present. An example is the quantitative evaluation of terrain masking on alternative plans. Another reason why ASPS will be attractive to strike planners is that it is essentially a powerful calculation and display assistant. The planner's judgment and reasoning is in control throughout the planning process; ASPS is not configured to force any take-it-or-leave-it decisions on the planner.

Cost of ASPS development will be low for several reasons. The state of computer technology is advancing so rapidly that exceptionally powerful, low cost desktop computers are just coming on the market. (An example is a desktop VAX computer made by Digital Equipment Corporation; this model will be available early in 1982.) The Land Minefield Planning Aid System (LAMPAS) is a Marine Corps program which is taking advantage of this advance in technology by incorporating sophisticated models in a desktop computer and display system. Consequently, our judgment is that hardware cost for a production version of ASPS would be only about \$60,000.¹ There are two other reasons why cost of ASPS development would be low. One is that much of the design work has already been done. The other is that much of the software, including algorithms for determining terrain masking around defense sites, has already been developed.

¹ This is a non-GSA price for a single unit.

REFERENCES

1. Irving, G.W., et al. Experimental Investigation of Sketch Model Accuracy and Usefulness in a Simulated Tactical Decision Aiding Task. Report No. 215-3, Integrated Sciences Corporation, Santa Monica, California, May 1977.
2. Walsh, D.H. and Schechterman, M.D. Experimental Investigation of the Usefulness of Operator Aided Optimization in a Simulated Tactical Decision Aiding Task. Report No. 215-4, Integrated Sciences Corporation, Santa Monica, California, January 1978.
3. Schechterman, M.D. and Walsh, D.H. Comparison of Operator Aided Optimization With Iterative Manual Optimization in a Simulated Tactical Decision Aiding Task. Report No. 215-6, Integrated Sciences Corporation, Santa Monica, California, July 1980.
4. Rebane, G.J. and Walsh, D.H. Concept for an Air Strike Planning System. Technical Memorandum 330-1, Integrated Sciences Corporation, Santa Monica, California, February 1981.
5. Defense Mapping Agency. Produce Specifications for Digital Landmass System (DLMS) Data Base. PS/ICD/100; PS/ICE/100; PS/ICF/100; PS/ICG/100, Aerospace Center, St. Louis, July 1977.
6. Jackson, R. and Reed, C. Terrain Shadow Prediction. MRS³ Working Paper No. 29, Pattern Analysis and Recognition Corp., Rome, New York, 22 February 1977.
7. Delaney, J.R., et al. Description of Terrain Masking Prediction Software. Report CMT-4, Lincoln Laboratory, MIT, 20 March 1980.
8. Braddock, Dunn, and McDonald. TACOS II Documentation, Volume IIIA: FRAG1A, FRAG1B, and FRAG1C User/Planner Manual. BDM Report A-13-74-TR-R1, Albuquerque, New Mexico, 15 May 1974.

APPENDIX A:
SOFTWARE STRUCTURE

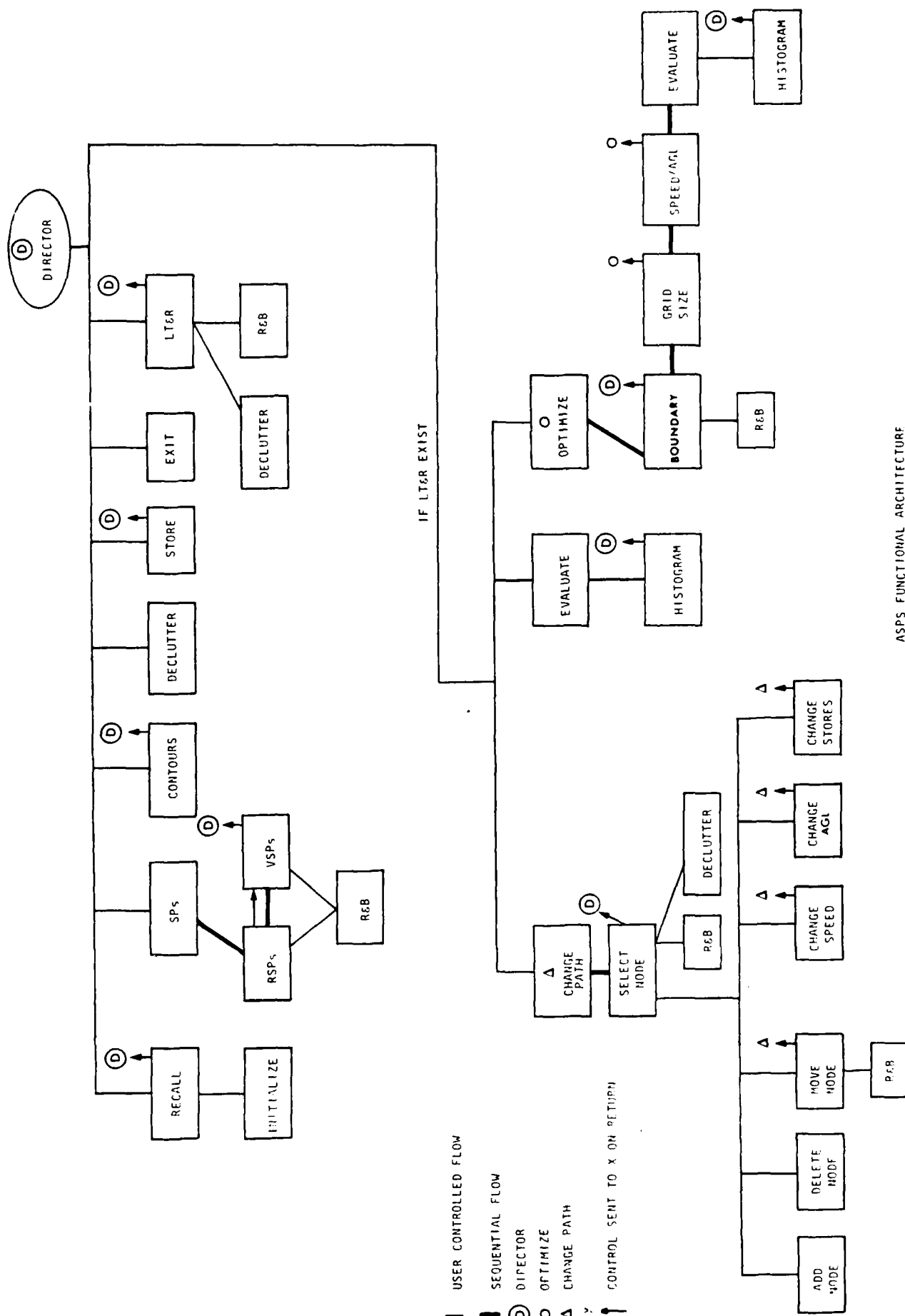
APPENDIX A: SOFTWARE STRUCTURE

A block diagram of ASPS's modular software structure is given in Figure A-1. For each box listed directly below the "Director" routine, there is a special function key as shown in Figure A-2. Routines indicated by boxes below the first tier boxes (special function keys) are manipulated through displayed menus and use of the ACCEPT and REJECT keys, number pad and/or trackball. The software implementing each function is self-contained to facilitate modifications, integration of new functions, and understanding the flow. The Director guides the top level flow of the program to the proper subroutines and ignores improper or untimely commands. Only those special function keys which are lighted at a given time are active. An example of an untimely command is attempting to change a path before a path exists. If no path exists, the " Δ CHANGE PATH" key is not activated (lighted).

The program consists of three overlays. When going between overlays, the flow always goes through the Director routine. The Director guides the flow to the proper overlay when the user presses an activated special function key. The three overlays contain modes which correspond to the following special function keys:

<u>Overlay 1</u>	<u>Overlay 2</u>	<u>Overlay 3</u>
Director	Change Path	Contours
Exit	Significant Points	Evaluate
Recall		Launch/Target/Recovery
Store		Optimize

Each mode has a specific task to perform and is entered via the Director. The user should know the purpose of each mode and how it is used. Instructions and a title for each mode are shown on the display to aid the operator. The subroutines "Evaluate" and "Optimize" do not currently perform specific mathematical computations. From each mode, the user can easily return to the Director Mode.



ASPS FUNCTIONAL ARCHITECTURE

Figure A-1.

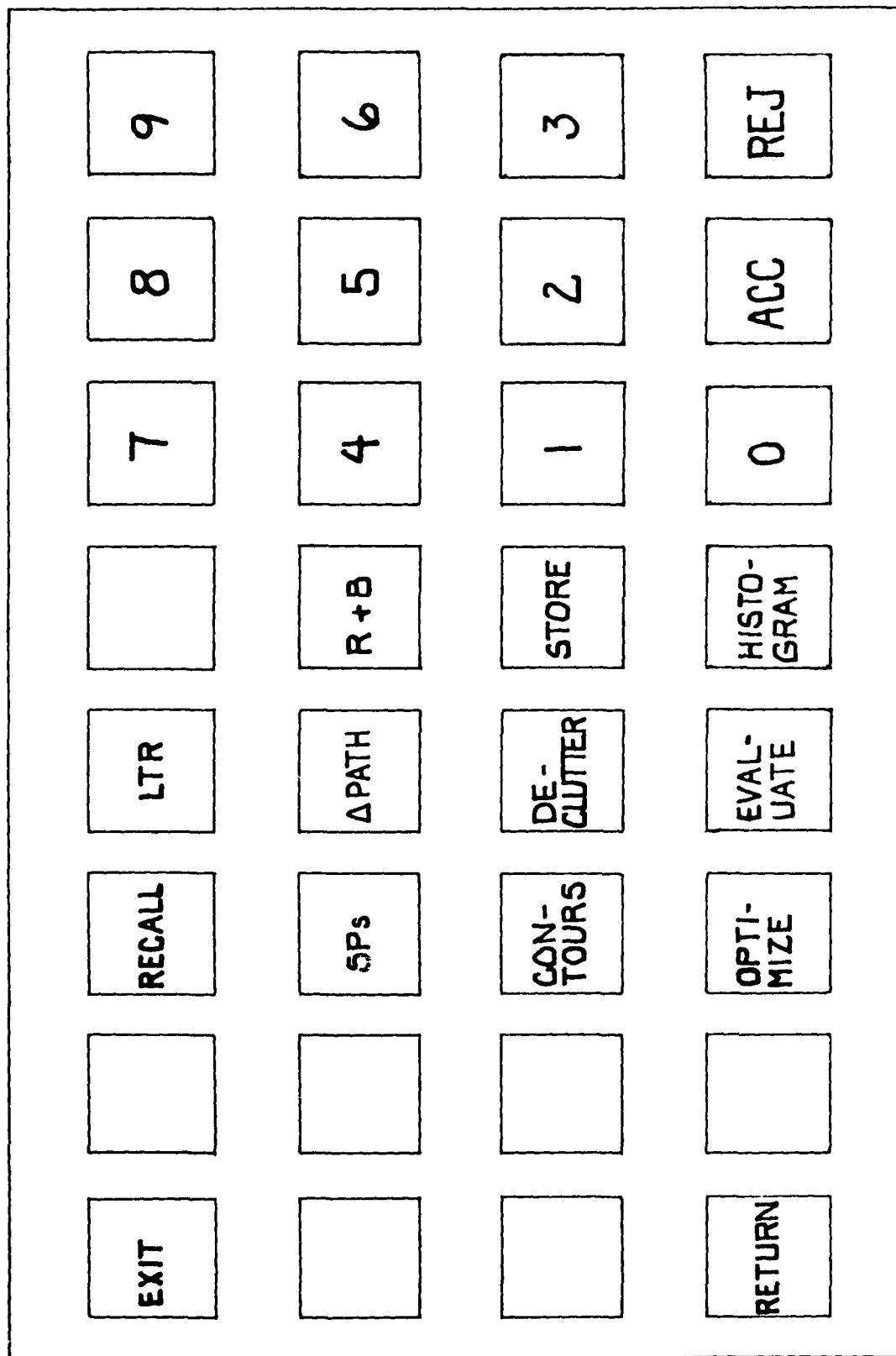


Figure A-2. Function Key Setup.

In addition to the modal keys, there are two special function keys used as aids for the operator, namely DECLUTTER and RANGE/BEARING. The DECLUTTER key allows the operator to call up or delete the display of significant points, defense capability contours, or both. This key is active while the program is in the Director, Change Path, Significant Points, and Launch/Target/Recovery Modes. The RANGE/BEARING key allows the user to get a range and bearing between two points. It is active while in the Change Path, Significant Points, and Launch/Target/Recovery Modes.

There is a bookmark routine which identifies and allows for the erasing of contour entities from the display file. Each time the Contours Mode is called, the previous entity numbers used for the contours are eliminated from the display file and then reused. The bookmark routine is not used when the DECLUTTER key is pressed to make contours disappear and reappear. The bookmark routine is only used when entering the Contours Mode. This allows a variable number of contours to be drawn and allows a variable length for each contour. The display file will not contain extraneous contours and, hence, the program runs more efficiently. All other entities have fixed length and are identified in the subroutine "Setup."

APPENDIX B:
DATA FILE STRUCTURE

APPENDIX B: DATA FILE STRUCTURE

The bookkeeping of the work done on ASPS is stored on disk in the data file "Admins." This is a data file which keeps track of the following information:

1. Number of maps and scenarios available
2. Last map and scenario recalled
3. Current scenario being worked on
4. Figure of Merit value for each scenario
5. Optimal scenario for each map thus far.

A description of the contents of the Admins file is given in Figure A-3.

For each scenario, there is a disk file with pertinent information to be read into the Master Data File (MDF). The MDF is set up such that the first fourteen words are pointers to their respective sections. This was done to allow maximum flexibility in the MDF array. Figure A-4 shows the MDF header with its set of pointers to specific locations within the MDF. Figures A-5 through A-13 describe each section of the MDF.

Some of the entries in the MDF are fixed, while others are supplied during the execution of the program by the operator. The permanent features include terrain, sensor positions, SAM site positions, AA site positions, origin and scale, elevations and character strings, and all defense capability contours. The information supplied by the user includes VSP/RSP positions, stores being carried, and the flight path.

The contents of the Admin file prevent user access to a Master Data file until a scenario has been stored in it. When a scenario is STORED by the operator, the Admin data is updated, but not stored onto disk until the operator presses EXIT.

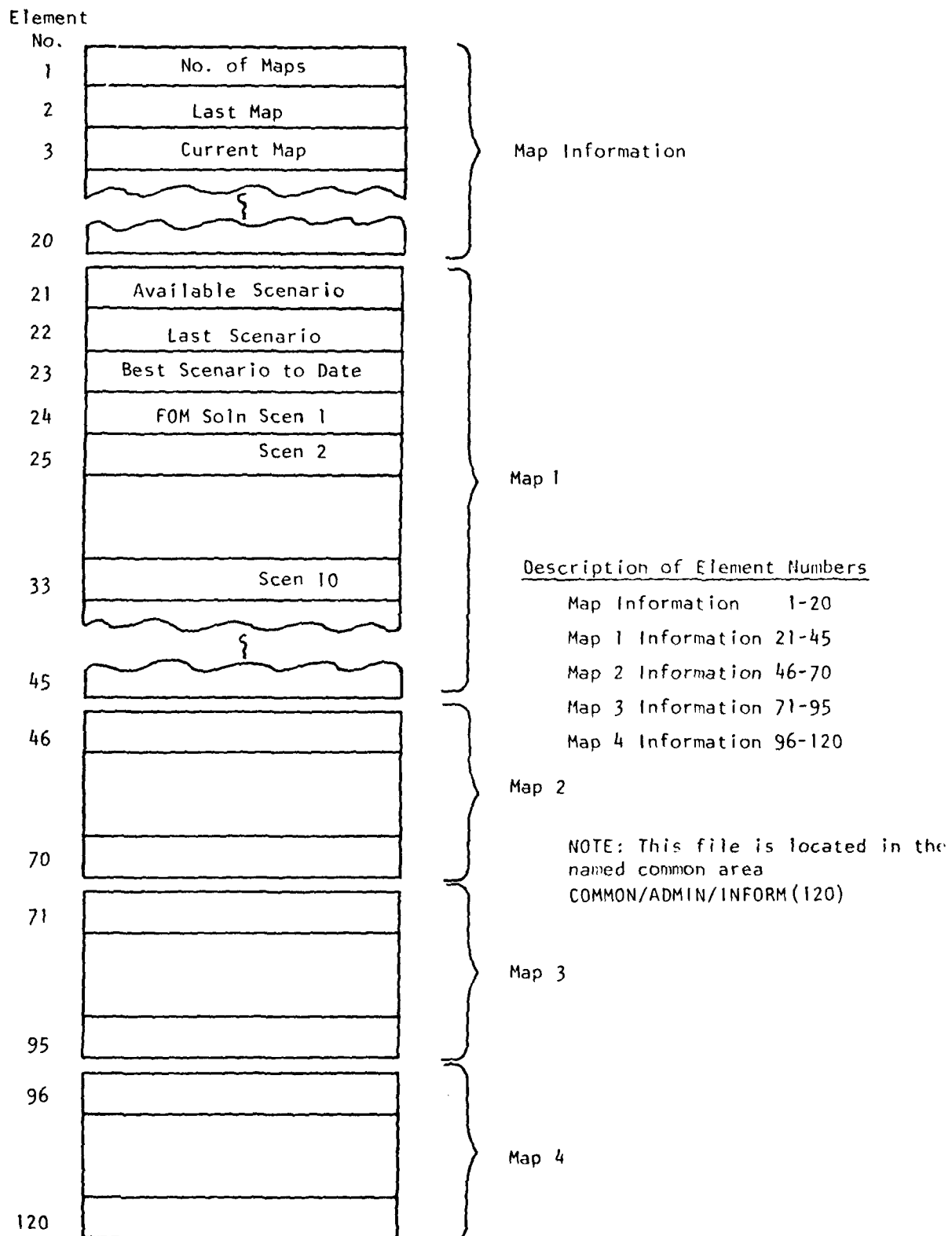


Figure A-3. Contents of Admins Data File.

1.	Terrain pointer
2.	Sensor pointer
3.	Surface to Air Missile (SAM) Pointer
4.	Character string and elevation pointer
5.	Lat., Long. and Scale pointer
6.	RSP pointer
7.	VSP pointer
8.	Path pointer
9.	Stores pointer
10.	Sensor Contour Pointer
11.	SAM Contour
12.	Anti-Aircraft (AA) Contour
13.	(AA) Pointer
14.	Scratch pointer

Figure A-4. Data Format of the Master Data File Header.

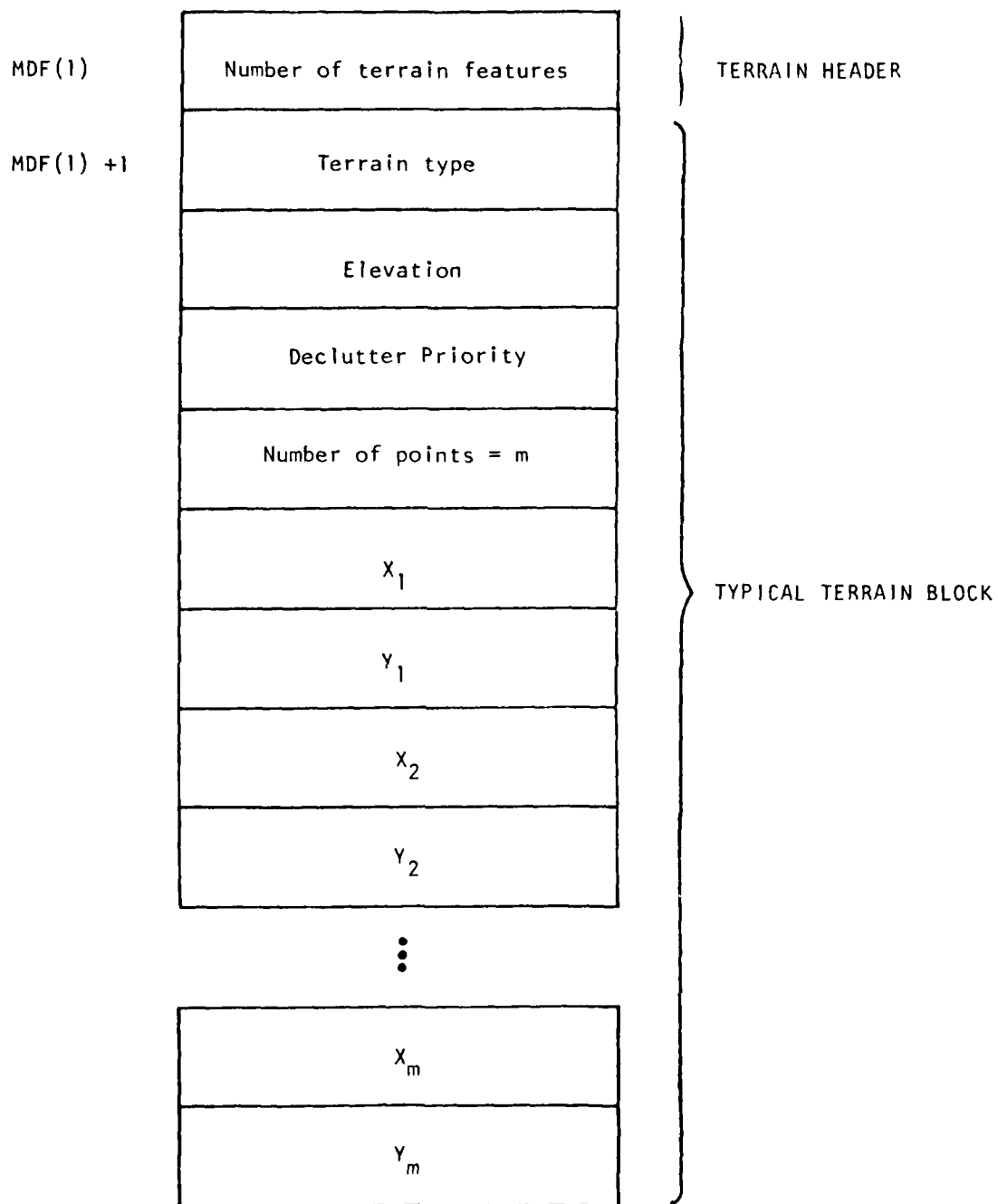


Figure A-5. Terrain Data Format.

MDF(2)

Number of Sensors = L
Bit code for active sensors**
X_i
Y_i

SENSOR HEADER

SENSOR BLOCK $i = 1$ to L

(a) Sensors

*100111₂ = 39 means sensors
1, 2, 3, 6 are active

MDF(3)

Number of SAMS = M
X_i
Y_i

SAM HEADER

SAM BLOCK $i = 1$ to M

(b) SAMs

MDF(4)

Number of character strings = N
X_i
Y_i
NUM

$i = 1$ to N

if NUM $\geq$ 0, display NUM

if NUM $<$ 0, display character
(NUM)

(c) Character Strings

Figure A-6. Sensor, SAM, and Character String Data Formats.

MDF(5)

lat. in minutes
long. in minutes
width of screen in nautical miles

(a) Latitude/Longitude Format

MDF(6)

Number of RSPs = N
X_i
Y_i

RSP header

$i = 1 \text{ to } N$

(b) RSP Format

MDF(7)

number of VSPs = M
X_i
Y_i

VSP header

$i = 1 \text{ to } M$

(c) VSP Format

Figure A-7. Data Format for Latitude/Longitude Position and Significant Points.

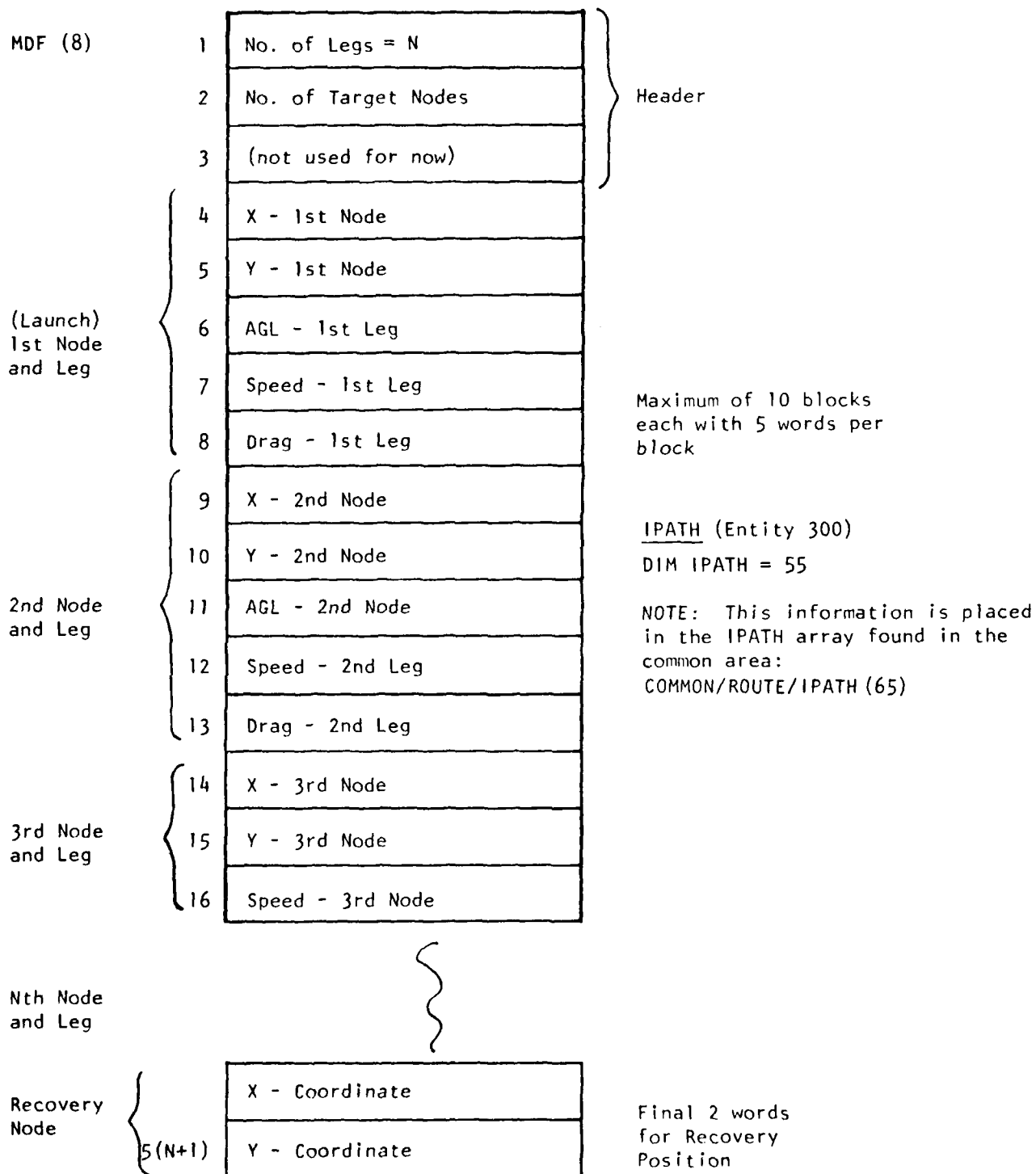


Figure A-8. Data Format of Path Information.

MDF (9)

No. of 500 lb. Bombs
1000 lb. Bombs
Missiles
Fuel Tanks

Figure A-9. Stores Data Format.

Each word is a pointer for each AGL and for each sensor and composite.

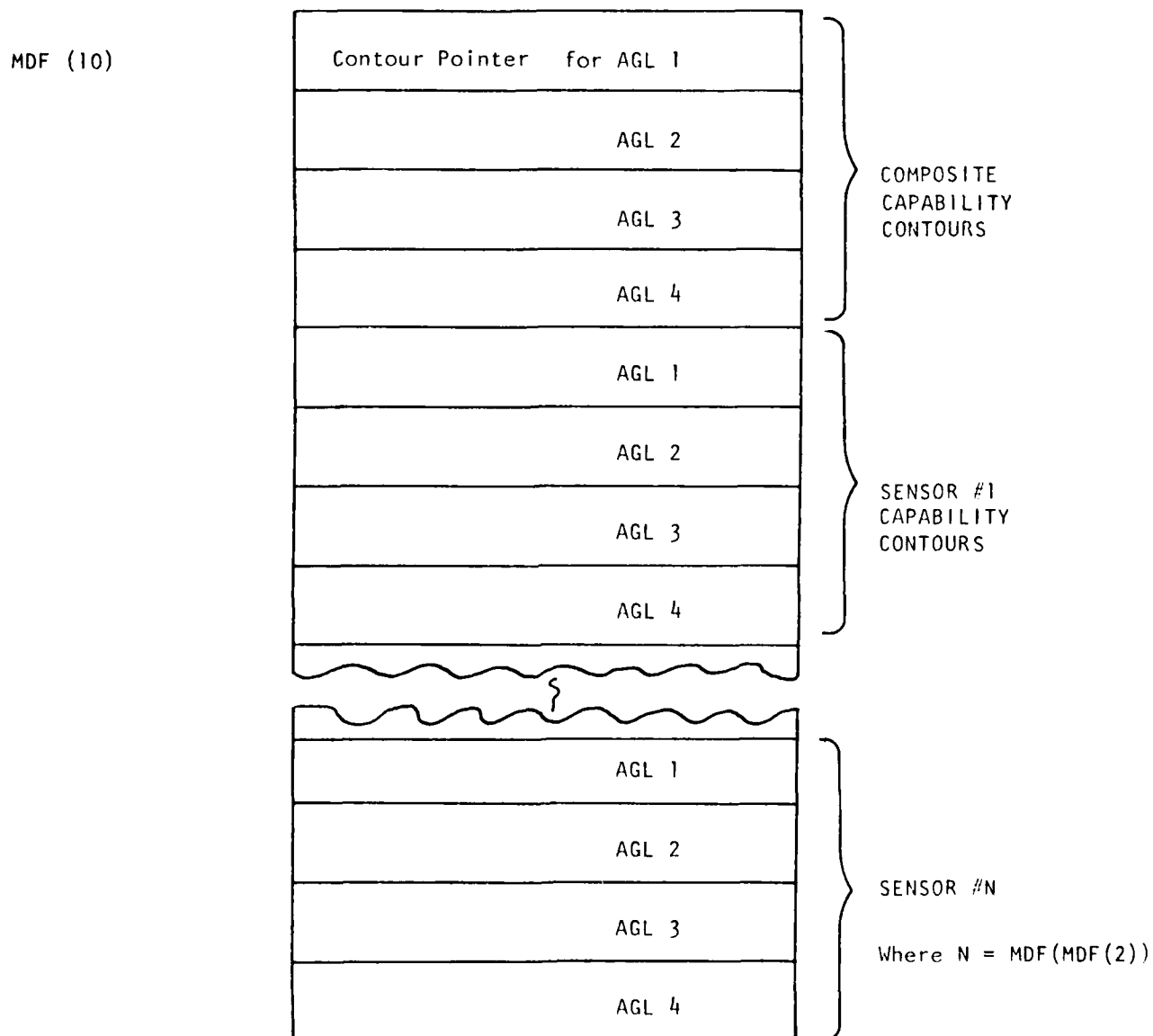


Figure A-10. Data Format of Sensor Contour Pointers.

For each AGL and each Sensor (including the Composite) the format is as follows:

No. of disjoint figures = M
No. of points in figure 1 = N_1 X_1 Y_1 $\vdots$ X_{N_1} Y_{N_1}
No. of points in figure 2 = N_2 X_1 Y_1 $\vdots$ X_{N_2} Y_{N_2}
}
No. of points in figure M = N_m X_1 Y_1 $\vdots$ X_{N_m} Y_{N_m}

Figure A-11. Data Format of Sensor Contour Points.

MDF (11)

SAM capability is modeled as a wide-angle cone with apex at the site.

"Radius" means horizon-plane radius of SAM capability at AGL elevation.

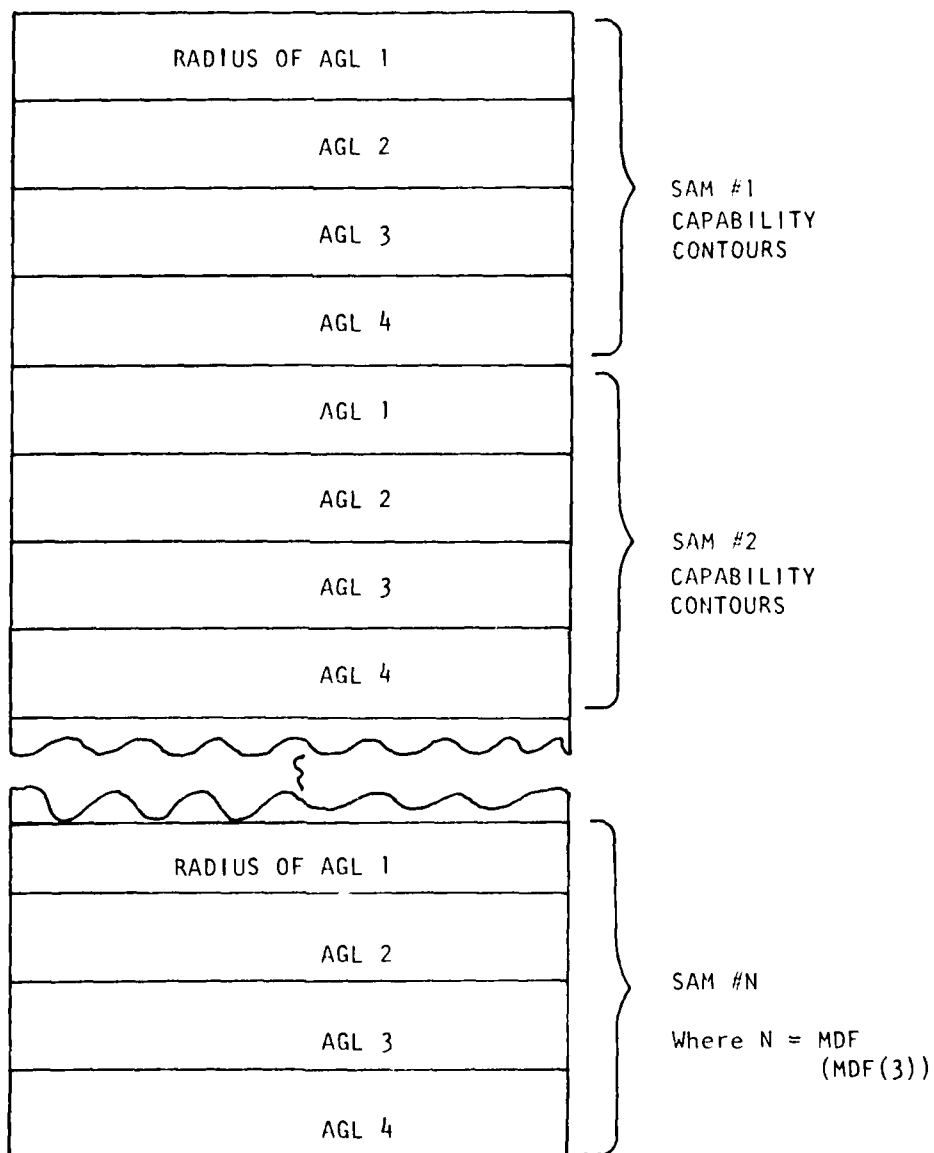


Figure A-12. Data Format for SAM Contours.

MDF(12)

Radius of AGL 1	}	AA #1
AGL 2		
AGL 3		
AGL 4		
AGL 1	}	AA #2
AGL 2		
AGL 3		
AGL 4		
		
		
Radius of AGL 1	}	AA #N
AGL 2		
AGL 3		
AGL 4		

Where N =
MDF(MDF(13))

Where N =
MDF(MDF(13))

(a) AA Contours

MDF(13)

Number of AAs = N	}	AA Header
X ₁		AA #1 Location
Y ₁		
⋮		
X _N	}	AA #N Location
Y _N		

where N = MDF(MDF(13))

(b) AA Locations

Figure A-13. Data Format for AA Contours and Locations.

APPENDIX C:
SOFTWARE LISTINGS

ASPS MAIN DRIVER

PAGE 1

MPLEVI VORTXII FTH IV(G)

0022 HOURS

```

1  COMMON /ADUIN/ INFORM(120)
2  COMMON /HISTEP/ HEP(2400)
3  COMMON /HTT/ IATT(120), UIR
4  UIR = 0
5  UIR = UIR + 1
6  COMMON /HISTEP/ HEP(2400)
7  UIR = UIR + 1
8
9  DEFINITION IOWLAY(3,40)
10 DATA IOWLAY(1,1), IOWLAY(2,1), IOWLAY(3,1)/'D0', '05', '01'
11 DATA IOWLAY(1,2), IOWLAY(2,2), IOWLAY(3,2)/'D0', '05', '02'
12 DATA IOWLAY(1,3), IOWLAY(2,3), IOWLAY(3,3)/'D0', '05', '03'
13 DATA IOWLAY(1,4), IOWLAY(2,4), IOWLAY(3,4)/'D0', '05', '04'
14 DATA IOWLAY(1,5), IOWLAY(2,5), IOWLAY(3,5)/'D0', '05', '05'
15
16 CALL QUIT(HEP, 4000, IATT, IERR)
17 IERR(1) = 0
18 IERR(2) = 0
19 IERR(3) = 0
20 CALL QUIT(0,1,IOWLAY(1,1))
21
22 10 CALL QUIT(0,1,IOWLAY(1,IFORM(1)))
23 GO TO 10
24 END
0 ERRORS COMPILATION COMPLETE

```

```

1      SUBROUTINE ADDNOD(NODE)
2 C    SUBROUTINE ADDNOD ALLOWS FOR THE ADDITION OF NEW WAY POINTS. A
3 C    MAXIMUM OF 10 WAY POINTS ARE ALLOWED AND A NODE CANNOT BE ADDED
4 C    BEYOND THE RECOVERY POINT.
5 C
6 C    NODE - THE NUMBER OF THE NODE TO PRECEED THE NEW NODE
7 C
8      COMMON /ROUTE/ IPATH(65)
9      IF (IPATH(1) .EQ. 10) GO TO 500
10
11 C    CALCULATE THE STARTING COORDINATES OF THE NEW NODE
12      IFOS = 6/NODE-2
13      IXNEW = FLOAT(IPATH(IFOS)+IPATH(IFOS+6))/2.
14      IYNEW = FLOAT(IPATH(IFOS+1)+IPATH(IFOS+7))/2.
15
16 C    CHECK IF THE DISTANCE OF THE NEW NODE TO THE REFERENCE NODE IS WITHIN
17 C    10 UNITS. IF TRUE, A NEW NODE WILL NOT BE ADDED AND A MESSAGE WILL APPEAR.
18      IF (IABS(IXNEW-IPATH(IFOS)) .GT. 12) GO TO 50
19      IF (IABS(IYNEW-IPATH(IFOS+1)) .LE. 12) GO TO 400
20 C
21 C    EXPAND IPATH ARRAY TO MAKE ROOM FOR THE NEW NODE
22      IPOS = (IPATH(1)-NODE)*6+2
23      IEND = 6*IPATH(1)+6
24      DO 100 J=1,IPOS
25          IPATH(IEND+6-J)=IPATH(IEND-J)
26      100 CONTINUE
27 C
28 C    INCREASE THE TOTAL NUMBER OF NODES COUNTER AND (IF NECESSARY)
29 C    THE POSITION OF THE TARGET NODE
30      IPATH(1) = IPATH(1) + 1
31      IF (IPATH(1) .GT. 100) IPATH(2) = IPATH(2)+1
32 C
33 C    FILL IN NEW NODE POSITION AND SET DEFAULT VALUES FOR AGL AND SPEED
34      IPATH(100+5) = IXNEW
35      IPATH(100+6) = IYNEW
36      IPATH(100+3) = IPATH(IFOS+2)
37      IPATH(100+9) = IPATH(IFOS+3)
38      GO TO 510
39
40 C    TRYING TO ADD A NODE TOO CLOSE TO SURROUNDING NODES
41      400 CALL ERRMSG (2)
42      GO TO 510
43
44 C    TRYING TO ADD MORE THAN 10 NODES.
45      500 CALL ERRMSG (1)
46      510 CALL DEPTH
47      CALL GHLT
48      CALL GENT(0)
49      CALL GENT(0,0)
50      RETURN
51      END
0 ERRORS COMPILATION COMPLETE

```

PAGE	1	NOLEVI	NOFTMII	FTN IVCG	0000 INITS
1		SUBROUTINE GMMIT			
2		COMMON /GMMT/IGMT(12),IERR			
3					
4	0	SUBROUTINE GMMITS FOR A DISPLAY INTERRUPT			
5					
6		IGMT(1)=0			
7		CALL GMMT(0,0)			
8	1	CONTINUE			
9		IF(IGMT(1).EQ.0)GO TO 1			
10		RETURN			
11		END			
0 ERRORS COMPILATION COMPLETE					

```

1      SUBROUTINE BLAST
2 C    SUBROUTINE BLAST ALLOWS THE USER TO DESCRIBE THE TYPE AND
3 C    AMOUNT OF STORES THE AIRCRAFT IS CARRYING. A MENU OF THE
4 C    TYPES OF STORES APPEARS AND A NUMBER PAD IS USED TO DESCRIBE
5 C    THE AMOUNT.
6
7      DIMENSION MAX(4)
8      COMMON /MASTER/ NDF(2400)
9 C    MAX IS GIVEN THE MAXIMUM POSSIBLE VALUE FOR NOW. TO BE CHANGED
10     DATA MAX /999,999,999,999/
11
12     CALL GET (1,4,0)
13     CALL GET (1,7,1)
14     IF IR - INT(9)
15     PRINTS = 4
16     CALL OPTIMU (5)
17
18 C    PUT CURRENT STORES VALUE INTO MENU
19 100   DO 120 K=0,3
20       CALL GET (1,10,K)
21       CALL GET (24,NDF(IPTRK),4)
22 120   CONTINUE
23       CALL SELECT (NDF(0),ICPT)
24       IF (ICPT .EQ. 0) GO TO 150
25       CALL CONTINU(5)
26       CALL GET (1,10,ICPT)
27       CALL SELECT (NDF(0),110+ICPT,24,N0)
28       IF (N0 .EQ. 0) N0 = (ICPT-1) * N0
29       GO TO 100
30
31 150   RETURN
32       END

```

0 ERRORS COMPILATION COMPLETE

PAGE 1 KLEVI MORTAL ETH IVCD CDDO MIES

```

1      SUPPORTIVE PLTCH
2      INFLIGHT INTEROP (0-2)
3
4      C  XXYX  FILLS CHARACTER AREAS AT LEFT SIDE OF SCREEN WITH PLTCH
5
6      CALL GMLT
7
8      DO 1 ENTITY=110,120
9      CALL GECHENTITY(6)
10     WRITE(15,1000)
11 1000  FORMATTED( )
12 1     CONTINUE
13
14     CALL GCHC(10,35)
15     WRITE(15,1000)
16     CALL GSHC(0,0)
17
18     RETURN
19     END
0 ERRORS COMPILATION COMPLETE

```

0 LEADS COMPLETED COMPLETE

```

1  SUBROUTINE CNRPTH
2  COMMON /ROUTE/ IPATH(65)
3
4  C THIS IS THE DRIVING ROUTINE TO ALTER A ROUTE
5
6  DIMENSION IONVRT(3)
7  INTEGER TARNOD
8  DATA IONVRT/1,4,5/
9
10 C WRITE WHEREABOUTS
11 LCHNOD=1
12
13 10 CALL SELNOD(NODNUM)
14 C NODNUM=0 IS USED TO EXIT THIS ROUTINE
15 IF (NODNUM.EQ. 0) GO TO 100
16
17 TARNOD=IPATH(2)
18 NOD=1
19 IF (NODNUM.EQ. LCHNOD. OR. NODNUM.EQ. TARNOD) NOD=2
20
21 C WRITE OPTIONS IN THE MENU
22 CALL CPTHMU(NODE)
23
24 C IF THE LAUNCH NODE OR TARGET NODE IS CHODEN USE A DIFFERENT MENU
25 C SELECT FROM MENU
26 NHOPTS=5
27 IF (NODE.EQ.2) NHOPTS=3
28 CALL SELECT(NHOPTS,IOPT)
29 C IOPT=0 INDICATES THAT THE RETURN KEY WAS HIT IN SELECT
30 IF (IOPT.EQ. 0) GO TO 50
31 IF (IOPT.EQ. 1) CALL FORTN(IOPT)
32
33 50 IF (IOPT.EQ. 3) CALL HOVNOD(NODNUM)
34 CALL ARGE (100)
35 CALL TCTNIG (2)
36 IF (IOPT.EQ. 4) CALL CNGSED(NODNUM)
37 IF (IOPT.EQ. 5) CALL CHGHLT(NODNUM)
38 CALL UNBLNK
39 IF (IOPT.EQ. 1) CALL ADDNOD(NODNUM)
40 IF (IOPT.EQ. 2) CALL DELNOD(NODNUM)
41 GO TO 10
42
43 100 CALL GUP (1,-1,0)
44 CALL UNBLNK
45 CALL TORNIG (1)
46 CALL GCNT (0)
47 CALL GSTT (0,0)
48 CALL GEOP(10)
49 CALL GEOP(193)
50 RETURN
51 END
0 ERRORS COMPILATION COMPLETE

```

```

1  SUBROUTINE CHGSPD(NODE)
2  C SUBROUTINE CHGSPD ALLOWS THE USER TO CHANGE THE CURRENT
3  C SPEED OF A GIVEN LEG.
4  C NODE - THE NODE NUMBER THAT IMMEDIATELY PRECEEDS THE LEG TO
5  C BE CHANGED
6
7  DIMENSION ISPD(4)
8  COMMON /ROUTE/ IPATH(65)
9  DATA ISPD /300,350,420,420/
10
11 C BLINK LEG (IBLINK = ELEMENT # THAT BLINKS LEG)
12 IFLINK = GNODE+3
13 CALL GENT (G00)
14 CALL GENT (G100,150,3,1)
15 CALL GENT (G100,150,3,1)
16 IF (IBLINK .EQ. 1) THEN
17   CALL GENT (G100,150,3,1)
18   CALL GENT (G100,150,3,1)
19
20 C DISPLAY CURRENTS, GET NEW ALTITUDE, AND PUT INTO IPATH
21 IPATH% = 4
22 CALL GENT (G100)
23 CALL GENT (G100,150,3,1)
24 IF (ALTITUDE .EQ. 0) THEN
25   IPATH% = 4
26
27 C DISPLAY NEW VALUES AND RETURN
28 CALL GENT (G100)
29 100 CALL GENT (G100,150,3,1)
30 CALL GENT (G100,150,3,1)
31 CALL GENT (G100,150,3,1)
32
33 RETURN
34 END

```

0 ERRORS COMPILATION COMPLETE

```

1      SUBROUTINE CNTOUR
2      COMMON /DISP/IDPL(4000),MARKR(12)
3      COMMON /MASTER/ NDF(2400)
4
5      C ALLOW USER TO SELECT CONTOUR AND AGL
6      IF (MARKR(1) .NE. 500) CALL GFEST(IDPL,MARKR(2),MARKR(3),MARKR(4))
7      CALL GEOME (IDPL,MARKR(2),MARKR(3),MARKR(4))
8      IENT = 500
9      CALL TORHOG(9)
10     CALL CPTHNU(7)
11     CALL SELECT (5,IWHCH)
12     CALL CPTHNU(7)
13     CALL GEOM (IWHCH+110)
14     CALL SELNUM (4,IWHCH+110,24,IAGL)
15     IF (IAGL .LE. 0) GO TO 500
16
17     C DISPLAY SENSOR CONTOURS
18     IPTR = NDF(10)
19     IF (IWHCH .EQ. 2 .OR. IWHCH .EQ. 5) GO TO 200
20     IF (IWHCH .EQ. 4) GO TO 400
21     NO = NDF(NDF(2))
22     DO 120 J=1,NO
23         NKTPTP = NDF(IPTR+IAGL+4*(J-1))
24         IF (NKTPTP .EQ. 0) GO TO 120
25         CALL SENCNT (NKTPTP,IENT)
26 120    CONTINUE
27     IF (IWHCH .NE. 3) GO TO 500
28     C DISPLAY SRA AA CONTOURS
29 200    IPTR = NDF(3)
30        IAGLPT = NDF(11)
31        NPTS = NDF(12)
32        DO 220 J=1,NPTS
33            IX = NDF(IPTR+IAGLPT+4*(J-1))
34            IY = NDF(IPTR+IAGLPT+4*(J-1)+1)
35            CALL CPTH (NKTPTP,IENT,IX,IY)
36            CALL CPTHNU (2,IENT)
37            IDPD = IAGLPT+IAGL+4*(J-5)
38            CALL GEUT (6,IPTR,IDPD,0)
39 220    CONTINUE
40        IF (IWHCH .NE. 5) GO TO 500
41
42     C DISPLAY COMPOSITE SENSOR CONTOUR
43 400    IPTR = NDF(10)
44        NKTPTP = NDF(IPTR+IAGL-1)
45        IF (NKTPTP .EQ. 0) GO TO 500
46        CALL SENCNT (NKTPTP,IENT)
47 500    MARKR(1) = IENT
48        CALL TORHOG(1)
49        CALL ALPHNU
50        CALL GEOM (110+IWHCH)
51        RETURN
52    END
53
0 ERRORS COMPILE COMPLETE

```

```

1 SUBROUTINE OPTIMUM(MODE)
2 C CHANGE PATH MENU
3 C IF MODE=1 THEN LIST ALL S OPTIONS IF MODE=2 THEN LIST
4 C ONLY THE THREE OPTIONS AVAILABLE FOR LAUNCH AND TARGET NODES
5
6 CALL GHLT
7 GO TO (100,100,600,700,200,300,350),MODE
8
9 C FOR NODES 1 AND 2 DISPLAY PATH OPTIONS
10 100 CALL GSCH(111,6)
11 WRITE(15,111)
12 IF(MODE.EQ.2)GO TO 500
13
14 CALL GSCH(112,6)
15 WRITE(15,112)
16 CALL GSCH(113,6)
17 WRITE(15,113)
18 CALL GSCH(114,6)
19 WRITE(15,114)
20 CALL GSCH(115,6)
21 WRITE(15,115)
22 111 FORMAT('ADD NODE')
23 112 FORMAT('DELETE NODE')
24 113 FORMAT('MOVE NODE')
25 114 FORMAT('CHANGE LEG SPEED')
26 115 FORMAT('CHANGE LEG ALTITUDE')
27 GO TO 900
28
29 500 CALL GSCH(112,6)
30 WRITE(15,114)
31 CALL GSCH(113,6)
32 WRITE(15,115)
33 GO TO 900
34
35 C FOR NODE 3 DISPLAY SPEED OPTIONS
36 600 CALL GSCH(111,6)
37 WRITE(15,111)
38 CALL GSCH(112,6)
39 WRITE(15,1001)
40 CALL GSCH(113,6)
41 WRITE(15,1002)
42 CALL GSCH(114,6)
43 WRITE(15,1003)
44 1600 FORMAT('SPEED I = 300 KNOTS')
45 1601 FORMAT(' II = 360')
46 1602 FORMAT(' III = 420')
47 1603 FORMAT(' IV = 480')
48 GO TO 900
49
50 C FOR NODE 4 DISPLAY AGL OPTIONS
51 700 CALL GSCH(111,6)
52 WRITE(15,1700)
53 CALL GSCH(112,6)
54 WRITE(15,1701)
55 CALL GSCH(113,6)
56 WRITE(15,1702)
57 CALL GSCH(114,6)
58 WRITE(15,1703)

```

PAGE 2 NRLEVI VORTXII FTN IV(G) 0001 HOURS

```

59 1700 FORMAT (' AGL I = 200 FEET')
60 1701 FORMAT (' II = 500')
61 1702 FORMAT (' III = 750')
62 1703 FORMAT (' IV = 1000')
63 GO TO 900
64
65 C MODE = 5 DISPLAY STORE OPTIONS
66 200 CALL GSCH (110,6)
67 WRITE (15,1200)
68 CALL GSCH (111,6)
69 WRITE (15,1201)
70 CALL GSCH (112,6)
71 WRITE (15,1202)
72 CALL GSCH (113,6)
73 WRITE (15,1203)
74 CALL GSCH (114,6)
75 WRITE (15,1204)
76 1200 FORMAT ('SELECT STORES OPTION')
77 1201 FORMAT (' 500 LB BOMB')
78 1202 FORMAT (' 1000 LB BOMB')
79 1203 FORMAT (' MISSILES')
80 1204 FORMAT (' FUEL TANKS')
81 GO TO 900
82
83 C DISPLAY MIP MENU
84 300 CALL GSCH (110,6)
85 WRITE (15,1300)
86 CALL GSCH (111,6)
87 WRITE (15,1301)
88 CALL GSCH (112,6)
89 WRITE (15,1302)
90 CALL GSCH (113,6)
91 WRITE (15,1303)
92 CALL GSCH (114,6)
93 WRITE (15,1304)
94 1300 FORMAT ('SELECT MIP')
95 1301 FORMAT (' 1. 1000 LB BOMB')
96 1302 FORMAT (' 2. 1000 LB BOMB')
97 1303 FORMAT (' 3. 1000 LB BOMB')
98 1304 FORMAT (' 4. 1000 LB BOMB')
99 GO TO 900
100
101 C CONTOUR'S MENU
102 350 CALL GSCH (110,6)
103 WRITE (15,1350)
104 CALL GSCH (111,6)
105 WRITE (15,1351)
106 CALL GSCH (112,6)
107 WRITE (15,1352)
108 CALL GSCH (113,6)
109 WRITE (15,1353)
110 CALL GSCH (114,6)
111 WRITE (15,1354)
112 CALL GSCH (115,6)
113 WRITE (15,1355)
114 1350 FORMAT ('SELECT CONTOUR'S OPTION')
115 1351 FORMAT (' 1. 1000 LB BOMB')
116 1352 FORMAT (' 2. 1000 LB BOMB')

```

AD-A101 860

INTEGRATED SCIENCES CORP SANTA MONICA CA
CONCEPTS FOR HIGHLY INTERACTIVE AIR STRIKE PLANNING SYSTEMS (AS--ETC(U)
MAY 81 D H WALSH, G J REBANE, L R LEVI

F/G 15/7

N00014-80-C-0812

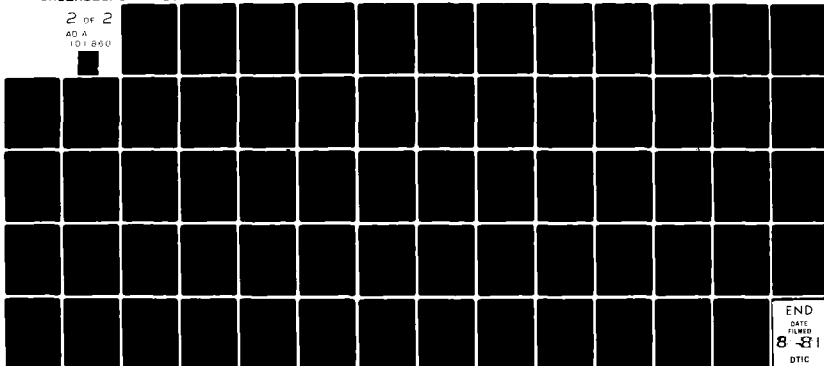
UNCLASSIFIED

ISC-330-2

NL

2 OF 2

AD A
101 860



END

DATE

FILED

8-81

DTIC

NO.	Q	DEPT	ASST	PER IVCO	DATE
117	1253	PAINT CO	PAINT		
118	1254	PAINT CO	PAINT		
119	1255	PAINT CO	PAINT		
120		PAINT CO	PAINT		
121		PAINT CO	PAINT		
122	960	PAINT CO	PAINT		
123		PAINT CO	PAINT		
124		PAINT CO	PAINT		
OVERHEAD CONSULTATION OUTLET					

```

1 SUBROUTINE DECLUT
2 COMMON /FLAG/ LMODE(9),RXPT(9),LSEICH(9),IFLOW(5)
3
4 C LEVEL 1 TERRAIN SFS CONTOURS
5 C LEVEL 2 TERRAIN SFS
6 C LEVEL 3 TERRAIN CONTOURS
7 C LEVEL 4 TERRAIN
8
9 LMODE(1)=0
10 TERRAIN(1)=0
11 LMODE(2)=0
12 CONTOURS(1)=0
13
14 100 CALL SUB1(0)
15 CALL SUB1(100)
16 GO TO 110
17
18 200 CALL SUB1(200)
19 CALL SUB1(300)
20 GO TO 110
21
22 300 CALL SUB1(400)
23 CALL SUB1(500)
24 GO TO 110
25
26 400 CALL SUB1(600)
27 CALL SUB1(700)
28 GO TO 110
29
30 500 CALL SUB1(800)
31 CALL SUB1(900)
32 GO TO 110
33
34 600 CALL SUB1(1000)
35 CALL SUB1(1100)
36 GO TO 110
37
38 700 CALL SUB1(1200)
39 CALL SUB1(1300)
40 GO TO 110
41
42 800 CALL SUB1(1400)
43 CALL SUB1(1500)
44 GO TO 110
45
46 900 CALL SUB1(1600)
47 CALL SUB1(1700)
48 GO TO 110
49
50 1000 CALL SUB1(1800)
51 CALL SUB1(1900)
52 GO TO 110
53
54 1100 CALL SUB1(2000)
55 CALL SUB1(2100)
56 GO TO 110
57
58 1200 CALL SUB1(2200)
59 CALL SUB1(2300)
60 GO TO 110
61
62 1300 CALL SUB1(2400)
63 CALL SUB1(2500)
64 GO TO 110
65
66 1400 CALL SUB1(2600)
67 CALL SUB1(2700)
68 GO TO 110
69
70 1500 CALL SUB1(2800)
71 CALL SUB1(2900)
72 GO TO 110
73
74 1600 CALL SUB1(3000)
75 CALL SUB1(3100)
76 GO TO 110
77
78 1700 CALL SUB1(3200)
79 CALL SUB1(3300)
80 GO TO 110
81
82 1800 CALL SUB1(3400)
83 CALL SUB1(3500)
84 GO TO 110
85
86 1900 CALL SUB1(3600)
87 CALL SUB1(3700)
88 GO TO 110
89
90 2000 CALL SUB1(3800)
91 CALL SUB1(3900)
92 GO TO 110
93
94 2100 CALL SUB1(4000)
95 CALL SUB1(4100)
96 GO TO 110
97
98 2200 CALL SUB1(4200)
99 CALL SUB1(4300)
100 GO TO 110
101
102 2300 CALL SUB1(4400)
103 CALL SUB1(4500)
104 GO TO 110
105
106 2400 CALL SUB1(4600)
107 CALL SUB1(4700)
108 GO TO 110
109
110 2500 CALL SUB1(4800)
111 CALL SUB1(4900)
112 GO TO 110
113
114 2600 CALL SUB1(5000)
115 CALL SUB1(5100)
116 GO TO 110
117
118 2700 CALL SUB1(5200)
119 CALL SUB1(5300)
120 GO TO 110
121
122 2800 CALL SUB1(5400)
123 CALL SUB1(5500)
124 GO TO 110
125
126 2900 CALL SUB1(5600)
127 CALL SUB1(5700)
128 GO TO 110
129
130 3000 CALL SUB1(5800)
131 CALL SUB1(5900)
132 GO TO 110
133
134 3100 CALL SUB1(6000)
135 CALL SUB1(6100)
136 GO TO 110
137
138 3200 CALL SUB1(6200)
139 CALL SUB1(6300)
140 GO TO 110
141
142 3300 CALL SUB1(6400)
143 CALL SUB1(6500)
144 GO TO 110
145
146 3400 CALL SUB1(6600)
147 CALL SUB1(6700)
148 GO TO 110
149
150 3500 CALL SUB1(6800)
151 CALL SUB1(6900)
152 GO TO 110
153
154 3600 CALL SUB1(7000)
155 CALL SUB1(7100)
156 GO TO 110
157
158 3700 CALL SUB1(7200)
159 CALL SUB1(7300)
160 GO TO 110
161
162 3800 CALL SUB1(7400)
163 CALL SUB1(7500)
164 GO TO 110
165
166 3900 CALL SUB1(7600)
167 CALL SUB1(7700)
168 GO TO 110
169
170 4000 CALL SUB1(7800)
171 CALL SUB1(7900)
172 GO TO 110
173
174 4100 CALL SUB1(8000)
175 CALL SUB1(8100)
176 GO TO 110
177
178 4200 CALL SUB1(8200)
179 CALL SUB1(8300)
180 GO TO 110
181
182 4300 CALL SUB1(8400)
183 CALL SUB1(8500)
184 GO TO 110
185
186 4400 CALL SUB1(8600)
187 CALL SUB1(8700)
188 GO TO 110
189
190 4500 CALL SUB1(8800)
191 CALL SUB1(8900)
192 GO TO 110
193
194 4600 CALL SUB1(9000)
195 CALL SUB1(9100)
196 GO TO 110
197
198 4700 CALL SUB1(9200)
199 CALL SUB1(9300)
200 GO TO 110
201
202 4800 CALL SUB1(9400)
203 CALL SUB1(9500)
204 GO TO 110
205
206 4900 CALL SUB1(9600)
207 CALL SUB1(9700)
208 GO TO 110
209
210 5000 CALL SUB1(9800)
211 CALL SUB1(9900)
212 GO TO 110
213
214 5100 CALL SUB1(10000)
215 CALL SUB1(10100)
216 GO TO 110
217
218 5200 CALL SUB1(10200)
219 CALL SUB1(10300)
220 GO TO 110
221
222 5300 CALL SUB1(10400)
223 CALL SUB1(10500)
224 GO TO 110
225
226 5400 CALL SUB1(10600)
227 CALL SUB1(10700)
228 GO TO 110
229
230 5500 CALL SUB1(10800)
231 CALL SUB1(10900)
232 GO TO 110
233
234 5600 CALL SUB1(11000)
235 CALL SUB1(11100)
236 GO TO 110
237
238 5700 CALL SUB1(11200)
239 CALL SUB1(11300)
240 GO TO 110
241
242 5800 CALL SUB1(11400)
243 CALL SUB1(11500)
244 GO TO 110
245
246 5900 CALL SUB1(11600)
247 CALL SUB1(11700)
248 GO TO 110
249
250 6000 CALL SUB1(11800)
251 CALL SUB1(11900)
252 GO TO 110
253
254 6100 CALL SUB1(12000)
255 CALL SUB1(12100)
256 GO TO 110
257
258 6200 CALL SUB1(12200)
259 CALL SUB1(12300)
260 GO TO 110
261
262 6300 CALL SUB1(12400)
263 CALL SUB1(12500)
264 GO TO 110
265
266 6400 CALL SUB1(12600)
267 CALL SUB1(12700)
268 GO TO 110
269
270 6500 CALL SUB1(12800)
271 CALL SUB1(12900)
272 GO TO 110
273
274 6600 CALL SUB1(13000)
275 CALL SUB1(13100)
276 GO TO 110
277
278 6700 CALL SUB1(13200)
279 CALL SUB1(13300)
280 GO TO 110
281
282 6800 CALL SUB1(13400)
283 CALL SUB1(13500)
284 GO TO 110
285
286 6900 CALL SUB1(13600)
287 CALL SUB1(13700)
288 GO TO 110
289
290 7000 CALL SUB1(13800)
291 CALL SUB1(13900)
292 GO TO 110
293
294 7100 CALL SUB1(14000)
295 CALL SUB1(14100)
296 GO TO 110
297
298 7200 CALL SUB1(14200)
299 CALL SUB1(14300)
300 GO TO 110
301
302 7300 CALL SUB1(14400)
303 CALL SUB1(14500)
304 GO TO 110
305
306 7400 CALL SUB1(14600)
307 CALL SUB1(14700)
308 GO TO 110
309
310 7500 CALL SUB1(14800)
311 CALL SUB1(14900)
312 GO TO 110
313
314 7600 CALL SUB1(15000)
315 CALL SUB1(15100)
316 GO TO 110
317
318 7700 CALL SUB1(15200)
319 CALL SUB1(15300)
320 GO TO 110
321
322 7800 CALL SUB1(15400)
323 CALL SUB1(15500)
324 GO TO 110
325
326 7900 CALL SUB1(15600)
327 CALL SUB1(15700)
328 GO TO 110
329
330 8000 CALL SUB1(15800)
331 CALL SUB1(15900)
332 GO TO 110
333
334 8100 CALL SUB1(16000)
335 CALL SUB1(16100)
336 GO TO 110
337
338 8200 CALL SUB1(16200)
339 CALL SUB1(16300)
340 GO TO 110
341
342 8300 CALL SUB1(16400)
343 CALL SUB1(16500)
344 GO TO 110
345
346 8400 CALL SUB1(16600)
347 CALL SUB1(16700)
348 GO TO 110
349
350 8500 CALL SUB1(16800)
351 CALL SUB1(16900)
352 GO TO 110
353
354 8600 CALL SUB1(17000)
355 CALL SUB1(17100)
356 GO TO 110
357
358 8700 CALL SUB1(17200)
359 CALL SUB1(17300)
360 GO TO 110
361
362 8800 CALL SUB1(17400)
363 CALL SUB1(17500)
364 GO TO 110
365
366 8900 CALL SUB1(17600)
367 CALL SUB1(17700)
368 GO TO 110
369
370 9000 CALL SUB1(17800)
371 CALL SUB1(17900)
372 GO TO 110
373
374 9100 CALL SUB1(18000)
375 CALL SUB1(18100)
376 GO TO 110
377
378 9200 CALL SUB1(18200)
379 CALL SUB1(18300)
380 GO TO 110
381
382 9300 CALL SUB1(18400)
383 CALL SUB1(18500)
384 GO TO 110
385
386 9400 CALL SUB1(18600)
387 CALL SUB1(18700)
388 GO TO 110
389
390 9500 CALL SUB1(18800)
391 CALL SUB1(18900)
392 GO TO 110
393
394 9600 CALL SUB1(19000)
395 CALL SUB1(19100)
396 GO TO 110
397
398 9700 CALL SUB1(19200)
399 CALL SUB1(19300)
400 GO TO 110
401
402 9800 CALL SUB1(19400)
403 CALL SUB1(19500)
404 GO TO 110
405
406 9900 CALL SUB1(19600)
407 CALL SUB1(19700)
408 GO TO 110
409
410 10000 CALL SUB1(19800)
411 CALL SUB1(19900)
412 GO TO 110
413
414 10100 CALL SUB1(20000)
415 CALL SUB1(20100)
416 GO TO 110
417
418 10200 CALL SUB1(20200)
419 CALL SUB1(20300)
420 GO TO 110
421
422 10300 CALL SUB1(20400)
423 CALL SUB1(20500)
424 GO TO 110
425
426 10400 CALL SUB1(20600)
427 CALL SUB1(20700)
428 GO TO 110
429
430 10500 CALL SUB1(20800)
431 CALL SUB1(20900)
432 GO TO 110
433
434 10600 CALL SUB1(21000)
435 CALL SUB1(21100)
436 GO TO 110
437
438 10700 CALL SUB1(21200)
439 CALL SUB1(21300)
440 GO TO 110
441
442 10800 CALL SUB1(21400)
443 CALL SUB1(21500)
444 GO TO 110
445
446 10900 CALL SUB1(21600)
447 CALL SUB1(21700)
448 GO TO 110
449
450 11000 CALL SUB1(21800)
451 CALL SUB1(21900)
452 GO TO 110
453
454 11100 CALL SUB1(22000)
455 CALL SUB1(22100)
456 GO TO 110
457
458 11200 CALL SUB1(22200)
459 CALL SUB1(22300)
460 GO TO 110
461
462 11300 CALL SUB1(22400)
463 CALL SUB1(22500)
464 GO TO 110
465
466 11400 CALL SUB1(22600)
467 CALL SUB1(22700)
468 GO TO 110
469
470 11500 CALL SUB1(22800)
471 CALL SUB1(22900)
472 GO TO 110
473
474 11600 CALL SUB1(23000)
475 CALL SUB1(23100)
476 GO TO 110
477
478 11700 CALL SUB1(23200)
479 CALL SUB1(23300)
480 GO TO 110
481
482 11800 CALL SUB1(23400)
483 CALL SUB1(23500)
484 GO TO 110
485
486 11900 CALL SUB1(23600)
487 CALL SUB1(23700)
488 GO TO 110
489
490 12000 CALL SUB1(23800)
491 CALL SUB1(23900)
492 GO TO 110
493
494 12100 CALL SUB1(24000)
495 CALL SUB1(24100)
496 GO TO 110
497
498 12200 CALL SUB1(24200)
499 CALL SUB1(24300)
500 GO TO 110
501
502 12300 CALL SUB1(24400)
503 CALL SUB1(24500)
504 GO TO 110
505
506 12400 CALL SUB1(24600)
507 CALL SUB1(24700)
508 GO TO 110
509
510 12500 CALL SUB1(24800)
511 CALL SUB1(24900)
512 GO TO 110
513
514 12600 CALL SUB1(25000)
515 CALL SUB1(25100)
516 GO TO 110
517
518 12700 CALL SUB1(25200)
519 CALL SUB1(25300)
520 GO TO 110
521
522 12800 CALL SUB1(25400)
523 CALL SUB1(25500)
524 GO TO 110
525
526 12900 CALL SUB1(25600)
527 CALL SUB1(25700)
528 GO TO 110
529
530 13000 CALL SUB1(25800)
531 CALL SUB1(25900)
532 GO TO 110
533
534 13100 CALL SUB1(26000)
535 CALL SUB1(26100)
536 GO TO 110
537
538 13200 CALL SUB1(26200)
539 CALL SUB1(26300)
540 GO TO 110
541
542 13300 CALL SUB1(26400)
543 CALL SUB1(26500)
544 GO TO 110
545
546 13400 CALL SUB1(26600)
547 CALL SUB1(26700)
548 GO TO 110
549
550 13500 CALL SUB1(26800)
551 CALL SUB1(26900)
552 GO TO 110
553
554 13600 CALL SUB1(27000)
555 CALL SUB1(27100)
556 GO TO 110
557
558 13700 CALL SUB1(27200)
559 CALL SUB1(27300)
560 GO TO 110
561
562 13800 CALL SUB1(27400)
563 CALL SUB1(27500)
564 GO TO 110
565
566 13900 CALL SUB1(27600)
567 CALL SUB1(27700)
568 GO TO 110
569
570 14000 CALL SUB1(27800)
571 CALL SUB1(27900)
572 GO TO 110
573
574 14100 CALL SUB1(28000)
575 CALL SUB1(28100)
576 GO TO 110
577
578 14200 CALL SUB1(28200)
579 CALL SUB1(28300)
580 GO TO 110
581
582 14300 CALL SUB1(28400)
583 CALL SUB1(28500)
584 GO TO 110
585
586 14400 CALL SUB1(28600)
587 CALL SUB1(28700)
588 GO TO 110
589
590 14500 CALL SUB1(28800)
591 CALL SUB1(28900)
592 GO TO 110
593
594 14600 CALL SUB1(29000)
595 CALL SUB1(29100)
596 GO TO 110
597
598 14700 CALL SUB1(29200)
599 CALL SUB1(29300)
600 GO TO 110
601
602 14800 CALL SUB1(29400)
603 CALL SUB1(29500)
604 GO TO 110
605
606 14900 CALL SUB1(29600)
607 CALL SUB1(29700)
608 GO TO 110
609
610 15000 CALL SUB1(29800)
611 CALL SUB1(29900)
612 GO TO 110
613
614 15100 CALL SUB1(30000)
615 CALL SUB1(30100)
616 GO TO 110
617
618 15200 CALL SUB1(30200)
619 CALL SUB1(30300)
620 GO TO 110
621
622 15300 CALL SUB1(30400)
623 CALL SUB1(30500)
624 GO TO 110
625
626 15400 CALL SUB1(30600)
627 CALL SUB1(30700)
628 GO TO 110
629
630 15500 CALL SUB1(30800)
631 CALL SUB1(30900)
632 GO TO 110
633
634 15600 CALL SUB1(31000)
635 CALL SUB1(31100)
636 GO TO 110
637
638 15700 CALL SUB1(31200)
639 CALL SUB1(31300)
640 GO TO 110
641
642 15800 CALL SUB1(31400)
643 CALL SUB1(31500)
644 GO TO 110
645
646 15900 CALL SUB1(31600)
647 CALL SUB1(31700)
648 GO TO 110
649
650 16000 CALL SUB1(31800)
651 CALL SUB1(31900)
652 GO TO 110
653
654 16100 CALL SUB1(32000)
655 CALL SUB1(32100)
656 GO TO 110
657
658 16200 CALL SUB1(32200)
659 CALL SUB1(32300)
660 GO TO 110
661
662 16300 CALL SUB1(32400)
663 CALL SUB1(32500)
664 GO TO 110
665
666 16400 CALL SUB1(32600)
667 CALL SUB1(32700)
668 GO TO 110
669
670 16500 CALL SUB1(32800)
671 CALL SUB1(32900)
672 GO TO 110
673
674 16600 CALL SUB1(33000)
675 CALL SUB1(33100)
676 GO TO 110
677
678 16700 CALL SUB1(33200)
679 CALL SUB1(33300)
680 GO TO 110
681
682 16800 CALL SUB1(33400)
683 CALL SUB1(33500)
684 GO TO 110
685
686 16900 CALL SUB1(33600)
687 CALL SUB1(33700)
688 GO TO 110
689
690 17000 CALL SUB1(33800)
691 CALL SUB1(33900)
692 GO TO 110
693
694 17100 CALL SUB1(34000)
695 CALL SUB1(34100)
696 GO TO 110
697
698 17200 CALL SUB1(34200)
699 CALL SUB1(34300)
700 GO TO 110
701
702 17300 CALL SUB1(34400)
703 CALL SUB1(34500)
704 GO TO 110
705
706 17400 CALL SUB1(34600)
707 CALL SUB1(34700)
708 GO TO 110
709
710 17500 CALL SUB1(34800)
711 CALL SUB1(34900)
712 GO TO 110
713
714 17600 CALL SUB1(35000)
715 CALL SUB1(35100)
716 GO TO 110
717
718 17700 CALL SUB1(35200)
719 CALL SUB1(35300)
720 GO TO 110
721
722 17800 CALL SUB1(35400)
723 CALL SUB1(35500)
724 GO TO 110
725
726 17900 CALL SUB1(35600)
727 CALL SUB1(35700)
728 GO TO 110
729
730 18000 CALL SUB1(35800)
731 CALL SUB1(35900)
732 GO TO 110
733
734 18100 CALL SUB1(36000)
735 CALL SUB1(36100)
736 GO TO 110
737
738 18200 CALL SUB1(36200)
739 CALL SUB1(36300)
740 GO TO 110
741
742 18300 CALL SUB1(36400)
743 CALL SUB1(36500)
744 GO TO 110
745
746 18400 CALL SUB1(36600)
747 CALL SUB1(36700)
748 GO TO 110
749
750 18500 CALL SUB1(36800)
751 CALL SUB1(36900)
752 GO TO 110
753
754 18600 CALL SUB1(37000)
755 CALL SUB1(37100)
756 GO TO 110
757
758 18700 CALL SUB1(37200)
759 CALL SUB1(37300)
760 GO TO 110
761
762 18800 CALL SUB1(37400)
763 CALL SUB1(37500)
764 GO TO 110
765
766 18900 CALL SUB1(37600)
767 CALL SUB1(37700)
768 GO TO 110
769
770 19000 CALL SUB1(37800)
771 CALL SUB1(37900)
772 GO TO 110
773
774 19100 CALL SUB1(38000)
775 CALL SUB1(38100)
776 GO TO 110
777
778 19200 CALL SUB1(38200)
779 CALL SUB1(38300)
780 GO TO 110
781
782 19300 CALL SUB1(38400)
783 CALL SUB1(38500)
784 GO TO 110
785
786 19400 CALL SUB1(38600)
787 CALL SUB1(38700)
788 GO TO 110
789
790 19500 CALL SUB1(38800)
791 CALL SUB1(38900)
792 GO TO 110
793
794 19600 CALL SUB1(39000)
795 CALL SUB1(39100)
796 GO TO 110
797
798 19700 CALL SUB1(39200)
799 CALL SUB1(39300)
800 GO TO 110
801
802 19800 CALL SUB1(39400)
803 CALL SUB1(39500)
804 GO TO 110
805
806 19900 CALL SUB1(39600)
807 CALL SUB1(39700)
808 GO TO 110
809
810 20000 CALL SUB1(39800)
811 CALL SUB1(39900)
812 GO TO 110
813
814 20100 CALL SUB1(40000)
815 CALL SUB1(40100)
816 GO TO 110
817
818 20200 CALL SUB1(40200)
819 CALL SUB1(40300)
820 GO TO 110
821
822 20300 CALL SUB1(40400)
823 CALL SUB1(40500)
824 GO TO 110
825
826 20400 CALL SUB1(40600)
827 CALL SUB1(40700)
828 GO TO 110
829
830 20500 CALL SUB1(40800)
831 CALL SUB1(40900)
832 GO TO 110
833
834 20600 CALL SUB1(41000)
835 CALL SUB1(41100)
836 GO TO 110
837
838 20700 CALL SUB1(41200)
839 CALL SUB1(41300)
840 GO TO 110
841
842 20800 CALL SUB1(41400)
843 CALL SUB1(41500)
844 GO TO 110
845
846 20900 CALL SUB1(41600)
847 CALL SUB1(41700)
848 GO TO 110
849
850 21000 CALL SUB1(41800)
851 CALL SUB1(41900)
852 GO TO 110
853
854 21100 CALL SUB1(42000)
855 CALL SUB1(42100)
856 GO TO 110
857
858 21200 CALL SUB1(42200)
859 CALL SUB1(42300)
860 GO TO 110
861
862 21300 CALL SUB1(42400)
863 CALL SUB1(42500)
864 GO TO 110
865
866 21400 CALL SUB1(42600)
867 CALL SUB1(42700)
868 GO TO 110
869
870 21500 CALL SUB1(42800)
871 CALL SUB1(42900)
872 GO TO 110
873
874 21600 CALL SUB1(43000)
875 CALL SUB1(43100)
876 GO TO 110
877
878 21700 CALL SUB1(43200)
879 CALL SUB1(43300)
880 GO TO 110
881
882 21800 CALL SUB1(43400)
883 CALL SUB1(43500)
884 GO TO 110
885
886 21900 CALL SUB1(43600)
887 CALL SUB1(43700)
888 GO TO 110
889
890 22000 CALL SUB1(43800)
891 CALL SUB1(43900)
892 GO TO 110
893
894 22100 CALL SUB1(44000)
895 CALL SUB1(44100)
896 GO TO 110
897
898 22200 CALL SUB1(44200)
899 CALL SUB1(44300)
900 GO TO 110
901
902 22300 CALL SUB1(44400)
903 CALL SUB1(44500)
904 GO TO 110
905
906 22400 CALL SUB1(44600)
907 CALL SUB1(44700)
908 GO TO 110
909
910 22500 CALL SUB1(44800)
911 CALL SUB1(44900)
912 GO TO 110
913
914 22600 CALL SUB1(45000)
915 CALL SUB1(45100)
916 GO TO 110
917
918 22700 CALL SUB1(45200)
919 CALL SUB1(45300)
920 GO TO 110
921
922 22800 CALL SUB1(45400)
923 CALL SUB1(45500)
924 GO TO 110
925
926 22900 CALL SUB1(45600)
927 CALL SUB1(45700)
928 GO TO 110
929
930 23000 CALL SUB1(45800)
931 CALL SUB1(45900)
932 GO TO 110
933
934 23100 CALL SUB1(46000)
935 CALL SUB1(46100)
936 GO TO 110
937
938 23200 CALL SUB1(46200)
939 CALL SUB1(46300)
940 GO TO 110
941
942 23300 CALL SUB1(46400)
943 CALL SUB1(46500)
944 GO TO 110
945
946 23400 CALL SUB1(46600)
947 CALL SUB1(46700)
948 GO TO 110
949
950 23500 CALL SUB1(46800)
951 CALL SUB1(46900)
952 GO TO 110
953
954 23600 CALL SUB1(47000)
955 CALL SUB1(47100)
956 GO TO 110
957
958 23700 CALL SUB1(47200)
959 CALL SUB1(47300)
960 GO TO 110
961
962 23800 CALL SUB1(47400)
963 CALL SUB1(47500)
964 GO TO 110
965
966 23900 CALL SUB1(47600)
967 CALL SUB1(47700)
968 GO TO 110
969
970 24000 CALL SUB1(47800)
971 CALL SUB1(47900)
972 GO TO 110
973
974 24100 CALL SUB1(48000)
975 CALL SUB1(48100)
976 GO TO 110
977
978 24200 CALL SUB1(48200)
979 CALL SUB1(48300)
980 GO TO 110
981
982 24300 CALL SUB1(48400)
983 CALL SUB1(48500)
984 GO TO 110
985
986 24400 CALL SUB1(48600)
987 CALL SUB1(48700)
988 GO TO 110
989
990 24500 CALL SUB1(48800)
991 CALL SUB1(48900)
992 GO TO 110
993
994 24600 CALL SUB1(49000)
995 CALL SUB1(49100)
996 GO TO 110
997
998 24700 CALL SUB1(49200)
999 CALL SUB1(49300)
1000 GO TO 110
1001
1002 24800 CALL SUB1(49400)
1003 CALL SUB1(49500)
1004 GO TO 110
1005
1006 24900 CALL SUB1(49600)
1007 CALL SUB1(49700)
1008 GO TO 110
1009
1010 25000 CALL SUB1(49800)
1011 CALL SUB1(49900)
1012 GO TO 110
1013
1014 25100 CALL SUB1(50000)
1015 CALL SUB1(50100)
1016 GO TO 110
1017
1018 25200 CALL SUB1(50200)
1019 CALL SUB1(50300)
1020 GO TO 110
1021
1022 25300 CALL SUB1(50400)
1023 CALL SUB1(50500)
1024 GO TO 110
1025
1026 25400 CALL SUB1(50600)
1027 CALL SUB1(50700)
1028 GO TO 110
1029
1030 25500 CALL SUB1(50800)
1031 CALL SUB1(50900)
1032 GO TO 110
1033
1034 25600 CALL SUB1(51000)
1035 CALL SUB1(51100)
1036 GO TO 110
1037
1038 25700 CALL SUB1(51200)
1039 CALL SUB1(51300)
1040 GO TO 110
1041
1042 25800 CALL SUB1(51400)
1043 CALL SUB1(51500)
1044 GO TO 110
1045
1046 25900 CALL SUB1(51600)
1047 CALL SUB1(51700)
1048 GO TO 110
1049
10
```

PAGE 1

MRLEVI VORTXII FTH IV(G)

0002 HOURS

```
1  SUBROUTINE DIR1
2  CONTINUE  IFLAGS= LEADE(9),ENPT(9),LSMCH(3),IFLCH(5)
3  LEAD=2,ENPT=1
4  INFO=0
5
6  IF (IFLAGS(1).EQ.0) GO TO 100
7  CALL FLECH
8  CALL FENPT
9
10 100  CALL FLECH
11      IF (IFLAGS(1).EQ.1000 TO 2000
12          GO TO 1000
13      IF (IFLAGS(1).EQ.3000) GO TO 1000
14      IF (IFLAGS(1).EQ.4000) GO TO 1000
15      IF (IFLAGS(1).EQ.5000) GO TO 1000
16      IF (IFLAGS(1).EQ.6000) GO TO 1000
17
18      IF (IFLAGS(1).EQ.1)
19          IF (IFLAGS(2).EQ.0)
20              GO TO 1000
21
22 200  RETURN
23  END
24 0 ERRORS CALCULATION COMPLETE
```

PAGE 1 PULVER MORTIMER FTH IV-10 01 01 1961

1 SUPPLEMENTAL FILE
2 CONTROL FILE-1 DATE (9), DATE (20), DATE (30), DATE (60)
3 CONTROL FILE-1
4 CONTROL FILE-1
5 CONTROL FILE-1
6 CONTROL FILE-1
7 CONTROL FILE-1
8 CONTROL FILE-1
9 CONTROL FILE-1
10 CONTROL FILE-1
11 CONTROL FILE-1
12 CONTROL FILE-1
13 CONTROL FILE-1
0 LEADS CONTAINING CONTROL FILE

1

407019

FTH 12000

1. *Phragmites* (common)

01 TOPS COMBINATION OUTLINE
/ 110
/ POSITION, SI, OF

```

1      SUBROUTINE DRECTR
2
3      C      THIS ROUTINE INFORMS THE USER OF THE OPTIONS AVAILABLE
4      C      BY TURNING ON THE LIGHTS ON THE KEYS. WHEN AN OPTION HAS BEEN
5      C      SELECTED IFLOW(1) IS SET TO THE OVERLAY NUMBER AND IFLOW(2)
6      C      IS SET TO THE ROUTINE NUMBER
7
8      IMPLICIT INTEGER(A-Z)
9      COMMON /MASTER/ NDF(2400)
10     COMMON /FLAGS/ LNODE(9),RXPT(9),LSUCH(3),IFLOW(5)
11     REAL RXPT
12     LOGICAL LSUCH
13     LOGICAL NOPATH
14
15     CALL TORMSG(13)
16     VDATE=4
17     OPTIND=5
18     STOFF=11
19     DCLTER=12
20     CNTOUR=13
21     CHGPTH=20
22     GETSPS=21
23     GETLTP=23
24     RECALL=29
25     EXIT=31
26
27     NOPATH=.FALSE.
28     IF(NDF(NDF(8)).EQ.0)NOPATH=.TRUE.
29
30     IF(NOPATH)CALL LAMP5(176,32,56,0)
31     IF(.NOT.NOPATH)CALL LAMP5(176,48,56,48)
32
33 10    CALL CKINT(KEY)
34     IF(KEY.EQ.-1)GO TO 10
35
36     IF(NOPATH)GO TO 120
37 110   IF(PRESINT)GO TO 120
38       IFLOW(1)=3
39       IFLOW(2)=5
40       GO TO 900
41
42 120   IF(KEY.NE.OPTIND)GO TO 130
43       IF(NOPATH)GO TO 130
44       IFLOW(1)=3
45       IFLOW(2)=5
46       GO TO 900
47
48 130   IF(KEY.NE.STOFF)GO TO 140
49       IFLOW(1)=1
50       IFLOW(2)=2
51       GO TO 900
52
53 140   IF(KEY.NE.DCLTER)GO TO 150
54       IFLOW(1)=1
55       IFLOW(2)=6
56       GO TO 900
57
58 150   IF(KEY.NE.CNTOUR)GO TO 160

```

PAGE	2	MRLEVI	VORTNII	FTN IV(G)	0003 HOURS
59		IFLOW(1)=3			
60		IFLOW(2)=4			
61		GO TO 900			
62					
63	160	IF(KEY.IE.CINPTH)GO TO 170			
64		IF(NCFATH)GO TO 170			
65		IFLOW(1)=2			
66		IFLOW(2)=2			
67		GO TO 900			
68					
69	170	IF(KEY.IE.GETSPS)GO TO 180			
70		IFLOW(1)=2			
71		IFLOW(2)=3			
72		GO TO 900			
73					
74	180	IF(STATE.EQUAL)GO TO 200			
75		IFLOW(1)=3			
76		IFLOW(2)=1			
77		GO TO 900			
78					
79	200	IF(STATE.EQUAL)GO TO 210			
80		IFLOW(1)=1			
81		IFLOW(2)=1			
82		GO TO 900			
83					
84	210	IF(STATE.EQUAL)GO TO 220			
85		IFLOW(1)=1			
86		IFLOW(2)=5			
87		GO TO 900			
88					
89	220	GO TO 10			
90					
91	900	CALL SUBR(1,-1,0)			
92		CALL SUBR(0)			
93		RETURN			
94		END			

0 ERRORS COMPILATION COMPLETE

```

1 SUBROUTINE IPATH
2 C SUBROUTINE IPATH DRAWS THE PATH IN THE COMMON AREA /ROUTE/
3 C IT EXPECTS TO FIND AN ARRAY OF LENGTH 50 CALLED IPATH IN THE
4 C COMMON AREA. THE LAUNCH, TARGET, AND TERMINAL NODES WILL BE
5 C COLORED DIFFERENTLY THAN THE OTHER NODES.
6
7 COMMON /ROUTE/ IPATH(50)
8 COMMON /FLAGS/ LNODE(9),RXPT(9),LSWCH(3),IFLOW(5)
9
10 C PRINT CIRCLE FOR LAUNCH NODE
11 L1=IPATH(1)
12 L2 = IPATH(2)-1
13 IEL = 7
14 ICOL = 1
15 CALL GENT(300)
16 CALL GPLOT(1,100,IPATH(4),0)
17 CALL GPLOT(2,110,IPATH(5),0)
18 CALL GPLOT(5,1700,0,0)
19 CALL GPLOT(6,140,6,0)
20
21 C ON FIRST PASS, SET UP ENTITY ONLY
22 IF (L1 .NE. 0) GO TO 90
23 ICHT = 1
24 CALL GPLOT(7,1774,255,0)
25 GO TO 105
26
27 90 CALL GPLOT(7,1000,10,0)
28 C DISPLAY LEGS AND CIRCLES FOR REMAINING NODES
29 DO 100 ICHT = 1,L1
30 J = 6*ICHT+4
31 CALL GPLOT (INT(IEL),140,6,ICOL)
32 CALL GPLOT (INT(IEL),130,3,0)
33 CALL GPLOT (INT(IEL),53,IPATH(J),IPATH(J+1))
34 IF (ICHT .EQ. 10) GO TO 105
35 ICOL = 1
36 CALL GPLOT (INT(IEL),110,3,0)
37 CALL GPLOT (INT(IEL),1000,10,0)
38 ICOL = 1
39 100 CONTINUE
40
41 C FILL IN NO DATA TO LEAVE SPACE IN IDPL ARRAY FOR POSSIBLE
42 C FUTURE ADDED NODES
43 IF (ICHT .GT. 10) RETURN
44 105 DO 110 J=ICHT,10
45 CALL GPLOT (INT(IEL),1774,255,0)
46 CALL GPLOT (INT(IEL),1774,255,0)
47 CALL GPLOT (INT(IEL),73,0,0)
48 CALL GPLOT (INT(IEL),1774,255,0)
49 CALL GPLOT (INT(IEL),1774,255,0)
50 CALL GPLOT (INT(IEL),1774,255,0)
51 110 CONTINUE
52
53 IF (L1 .NE. 0 .AND. LNODE(2) .NE. 1) CALL LEGTAB
54 RETURN
55 END
0 ERRORS COMPILE COMPLETE

```

0004 HOURS

0 111005 00 110111 0011 0111

PAGE 1

NOLEVI VERTEX ETH IVCO

0004 10000

1 SUPD AT THE 10000 10000 10000
2 C THIS 10000 10000 10000 10000
3
4 CALL 10000 10000 10000
5 CALL 10000 10000
6 CALL 10000 10000 10000 10000
7 CALL 10000 10000 10000
8 CALL 10000 10000 10000
9 CALL 10000 10000 10000
10 CALL 10000 10000 10000
11 CALL 10000 10000 10000
12 CALL 10000 10000 10000
13 CALL 10000 10000 10000
14 CALL 10000 10000 10000
15 RETURN
16
O 10000 10000 10000 10000

1 SURROUNDING ELEVATION HEIGHT, IN, TWO
 2 C THIS ELEVATION DOES NOT RELY ON SITE
 3
 4 CALL OF ELEVATION, IN, TWO
 5 CALL OF ELEVATION
 6 CALL OF ELEVATION, IN, TWO
 7 CALL OF ELEVATION, IN, TWO
 8 CALL OF ELEVATION, IN, TWO
 9 CALL OF ELEVATION, IN, TWO
 10 CALL OF ELEVATION, IN, TWO
 11 CALL OF ELEVATION, IN, TWO
 12 CALL OF ELEVATION, IN, TWO
 13 CALL OF ELEVATION, IN, TWO
 14 CALL OF ELEVATION, IN, TWO
 15 CALL OF ELEVATION, IN, TWO
 16 CALL OF ELEVATION, IN, TWO
 17 ELEVATION
 18 ELEVATION

0 ELEVATION CORRELATION QUALITY

```

1  SUBROUTINE DPLNERS
2  C THIS SUBROUTINE TURNS ON SIGNIFICANT POINTS TO BE DISPLAYED
3
4  COMMON /MASTER/ NDF(2400)
5
6  C DISPLAY RADAR SIGNIFICANT POINTS
7  IPTR = NDF(6)
8  IEND = NDF(IPTR)
9  IF (IEND.EQ.0) GO TO 30
10 DO 30 J=1,IEND
11     IF (NDF(J).EQ.0) GO TO 30
12     CALL GOUT (1,1,0,NDF(IPTR+IP00),0)
13     CALL GOUT (2,1,0,NDF(IPTR+IP00+1),0)
14     CALL GOUT (3,1,0,NDF(IPTR+IP00+1),0)
15     CALL GOUT (4,1,0,NDF(IPTR+IP00+1),0)
16 20  CONTINUE
17
18  C DISPLAY VISUAL SIGNIFICANT POINTS
19 30  IEND = NDF(7)
20  IEND = IEND - 1
21  IF (IEND.EQ.0) GO TO 50
22  DO 40 J=1,IEND
23     IF (NDF(J).EQ.0) GO TO 40
24     CALL GOUT (1,1,0,NDF(IPTR+IP00),0)
25     CALL GOUT (2,1,0,NDF(IPTR+IP00+1),0)
26     CALL GOUT (3,1,0,NDF(IPTR+IP00+1),0)
27     CALL GOUT (4,1,0,NDF(IPTR+IP00+1),0)
28 40  CONTINUE
29
30 50  RETURN
31  END

```

0 ERRORS CONTILATION OCCURE

```

1      SUBROUTINE DEFSOON
2      C SUBROUTINE DEFSOON DISPLAYS THE SCENARIO, RADAR, AND SAM SITES. IT
3      C EXPECTS TO FIND THE POINTS FOR THE TERRAIN AND THE COORDINATES FOR
4      C THE SAM AND RADAR SITES IN AN ARRAY MDF WHICH IS LOCATED IN THE
5      C COMMON AREA INCTER. THE SCENARIO WILL BE FOUND IN ENTITIES 200 TO 299.
6      C SEE DOCUMENTATION AS TO THE FORM OF THE MDF ARRAY.
7
8      COMMON /MASTER/ MDF(2400)
9      IENT = 129
10
11      C DRAW IN ORDER: (1) TERRAIN, (2) RADAR SITES, (3) SAM SITES
12      DO 100 J=1,3
13          IPTR = MDF(J)
14          N = NDF(IPTR)
15          IF (N.EQ.0) GO TO 100
16          IPTD = IPTD+1
17          IF (J.GT.1) GO TO 75
18
19      C DISPLAY THE SCENARIO AND TERRAIN FEATURES.
20      DO 50 I=1,N
21          CALL GETLOC(TERRAIN,IPTR,IPTR+4,NDF(IPTR+3))
22          CALL GETLOC(MDF(IPTD),IPTD)
23          CALL GETLOC(MDF(IPTD+3))
24      50      CONTINUE
25      GO TO 100
26
27      C DRAW THE RADAR AND SAM SITES
28      75      IF (J.EQ.2) CALL GETLOC(IPTD,IPTR,IPTR+4,NDF(IPTR+3))
29              IF (J.EQ.3) CALL GETLOC(IPTD,IPTR,IPTR+4,NDF(IPTR+3))
30              IF (J.EQ.1) GO TO 100
31              HOLD=IPTR
32              IPTD=IPTR+4
33              IPTD=IPTD+3
34              CALL GETLOC(IPTD,HOLD,NDF(IPTD),NDF(IPTD+3))
35              CALL GETLOC(IPTD)
36      90      CONTINUE
37      100     CONTINUE
38      RETURN
39      END
0 ERRORS COMPILATION COMPLETE

```

```

1      SUBROUTINE ERRMSG(N0)
2      COMMON DATA IAT(12),IERP
3      C SUBROUTINE ERRMSG PRINTS ALL ERROR MESSAGES BELOW THE SCENARIO. THIS
4      C ROUTINE EXPECTS ENTITY 191 TO HAVE BEEN SET UP TO ALLOW FOUR LINES OF
5      C EACH LINE WITH A MAXIMUM OF 50 CHARACTERS. AFTER THE MESSAGE IS DISPLAYED,
6      C THE ROUTINE SITS IN A LOOP UNTIL THE RETURN KEY IS PRESSED.
7
8      CALL GLWP (1,-1,0)
9      CALL GLWP (1,7,1)
10
11     C BLINK "ERROR" MESSAGE
12     CALL GENT(191)
13     CALL GFUT (3,130,3,1)
14     CALL GFUT (167,140,6,0)
15     CALL GGOH (191,6)
16     WRITE (15,500)
17     500 FORMAT (20X,'-- ERROR --')
18     GO TO (1,2,3),N0
19
20     1 CALL GGOH (191,60)
21     WRITE (15,10)
22     10 FORMAT (5X,'THE MAXIMUM NUMBER OF ALLOWABLE WAY POINTS')
23     CALL GGOH(191,114)
24     WRITE (15,11)
25     11 FORMAT (5X,'-10- HAS BEEN EXCEEDED!')
26     GO TO 505
27
28     2 CALL GGOH (191,60)
29     WRITE (15,20)
30     20 FORMAT (5X,'WAY POINT NOT ADDED SINCE IT LIES TOO ')
31     CALL GGOH(191,114)
32     WRITE (15,21)
33     21 FORMAT (5X,'CLOSE TO THE SURROUNDING WAY POINTS!')
34     GO TO 505
35
36     C NUMBER BAD ROUTINE ERROR. NUMBER ENTERED GREATER THAN MAX
37     3
38     CALL GGOH (191,60)
39     WRITE (15,30)
40     CALL GGOH(191,114)
41     WRITE (15,30)
42     3000 FORMAT (2X,'INPUTTED NUMBER IS GREATER THAN THE MAXIMUM.')
43     3005 FORMAT (2X,'CHECK ON THE RANGE AND INPUT THE NUMBER AGAIN')
44     GO TO 505
45
46     505 CALL GGOH (191,168)
47     WRITE (15,510)
48     510 FORMAT (10X,'PRESS "RETURN" TO CONTINUE . . .')
49     CALL GGOH (191)
50
51     C UNIT FOR WHICH NEW INTEREST TO CONTINUE WITH THE SCENARIO.
52     CALL GENT(191)
53     CALL GFUT (3,130,3,1)
54     610 CALL GFUT (167,140,6,0)
55     IF (IERP) GO TO 610
56     CALL GGOH (191,6)
57     IF (IERP) GO TO 610
58     CALL GFUT (3,130,3,1)
59     CALL GFUT (167,140,6,0)

```

PAGE

2

RELAY

MODIFY

FIN 1400

0004 11000

50

CALL 0000 00, 100, 0, 00

60

CALL 0000 00, 100, 0, 00

61

CALL 0000 000

62

00000

63

000

0 00000 000 000000 000000

PAGE 1 NOLEVT NOFTML FTH INCG 0125 19475

1 SUMMATION UNIT
 2 DIFFERENTIAL UNIT
 3 CONTROL UNIT - 100000000
 4 DATA BUFFER, 100000, 100000, 100000, 100000, 100000, 100000, 100000, 100000
 5 BUFFER 100000
 6 BUFFER 100000
 7 BUFFER 100000
 8 BUFFER 100000
 9 BUFFER 100000
 10 BUFFER 100000
 11 BUFFER 100000
 12 BUFFER 100000
 0 LARORS CONJUNCTION CONTROL

```

1      SUBROUTINE GETLTP
2      C SUBROUTINE GETLTP ALLOWS THE USER TO SPECIFY THE POSITION OF THE LAUNCH,
3      C TARGET, AND RECOVERY NODES. A CURSOR APPEARS DESCRIBED BY ENTITY 10.
4      C THE USER MOVES THE CURSOR TO THE DESIRED POSITION AND PASSES THE
5      C ACCEPT KEY TO SPECIFY A NODE POSITION.
6
7      COMMON /ATT/ IATT(12), IERR
8      COMMON /ROUTE/ IPATH(65)
9      COMMON /MASTER/ MEF(2400)
10     COMMON /FLAGS/ LMODE(9), RXPT(9), LSWCH(3), IFLOW(5)
11
12     C TURN OFF OLD ROUTE
13     CALL GEOF(300)
14     CALL GEOF(310)
15     CALL GEOF(311)
16     CALL GEOF(312)
17     DO 3 J=1,11
18         CALL GEOF (J+149)
19     3 CONTINUE
20     DO 4 J=1,4
21         CALL GENT(110+J)
22         CALL GRUT (23,90,1H ,0)
23         CALL GRUT (24,90,1H ,0)
24         CALL GRUT (25,90,1H0,0)
25         MEF(MEF(9)-1+J) = 0
26     4 CONTINUE
27
28     C INITIALIZE PATH ARRAY AND TURN ON CYCLE TIMER
29     ISWCH = 0
30     IPTR = MEF(3)
31     IPOS = 0
32     IPATH (1) = 2
33     IPATH (2) = 2
34     CALL TOTMSG(6)
35     CALL GULT
36     CALL GETT(1)
37
38     C SET CURSOR TO CURRENT TRACKBALL POSITION
39     CALL GRPCK (IX,IY)
40     CALL GENT(10)
41     CALL GRUT (1,100,IX,0)
42     CALL GRUT (2,110,IY,0)
43     CALL GEON (10)
44
45     C GET IN ORDER LAUNCH, TARGET, AND RECOVERY NODES
46     DO 100 J = 4,16,6
47         K = FLOAT(J+2)/6.+309.
48         CALL MSGAG (K-306)
49         LSWCH(1) = .FALSE.
50         CALL GEOF (100)
51         CALL TOTMSG(2)
52         CALL LMF3 (0,3,16,130)
53     5 IATT(1) = 0
54       CALL GETT(0,0)
55
56     C CHECK ON INTERUPTS
57     10 CALL CRINT(KEY)
58       IF (KEY .EQ. -1) GO TO 10

```

```

59 IF (KEY .EQ. 1) GO TO 50
60 IF (KEY .EQ. 19) GO TO 75
61 IF (KEY .EQ. 7) GO TO 200
62 IF (KEY .EQ. 12) CALL DECLUT
63 IF (KEY .NE. 40) GO TO 5
64
65 C CYCLE TIMER INTERRUPT! MOVE CURSOR TO CURRENT POSITION
66 CALL GTRACK (IX,IY)
67 ISUCH = 1-ISUCH
68 CALL GENT (10)
69 CALL GPUT (1,100,IX,0)
70 CALL GPUT (2,110,IY,0)
71 IF (ISUCH .EQ. 0) GO TO 30
72 CALL LATLON (IX,IY)
73 GO TO 10
74
75 C RANGE AND BEARING UPDATE
76 30 IF (.NOT. LSMCH(1)) GO TO 10
77 CALL RANGE (IX,IY)
78 CALL LAMP (0,0,0,130)
79 GO TO 10
80
81 C KEYBOARD INTERRUPT!
82 C SET APPROPRIATE ELEMENTS IN IPATH ARRAY TO LAUNCH, TARGET, RECOVERY POSITIONS
83 50 IPATH(J) = IX
84 IPATH(J+1) = IY
85 NDC(IPTR+IPOS)=IX
86 NDC(IPTR+IPOS+1) = IY
87 IPOS = IPOS+2
88
89 C SET ON FIGURE OVER THE APPROPRIATE NODE. K GETS THE ENTITY NUMBER.
90 CALL GENT (K)
91 CALL GPUT (1,100,IX,0)
92 CALL GPUT (2,110,IY,0)
93 CALL GENT (0)
94 GO TO 10
95
96 C RANGE AND BEARING ROUTINE - GET REFERENCE POINT
97 75 IF (LSUCH(1)) GO TO 30
98 IX=100
99 IY = 0
100 CALL RANGE (IX,IY)
101 CALL MESSAG (M=306)
102 CALL GLMP (1,1,1)
103 IF (.NOT.LSMCH(1)) CALL GLMP (1,19,1)
104 GO TO 10
105 100 CONTINUE
106
107 C SET AGL AND SPEED DEFAULT VALUES
108 IPATH(6)=500
109 IPATH(7)=350
110 IPATH(12) = 500
111 IPATH(13) = 350
112 CALL DEPTH
113 CALL GEOM(300)
114
115 C TURN OFF LIGHTS, HALT CYCLE TIMER THEN RETURN
116 150 CALL GEOP(10)

```

ENDS HERE

- 105 -

```

1  SUBROUTINE GETSPS
2  C SUBROUTINE GETSPS ALLOWS THE USER TO IDENTIFY RADAR SIGNIFICANT
3  C POINTS AND VISUALLY S.P. A MAXIMUM OF 10 EACH IS ALLOWED.
4
5      COMMON /ATT/ IATT(12),IEFR
6      COMMON /MASTER/ IDEF(2400)
7      COMMON /FLAGS/ LNODE(9),RXPT(9),LSMCH(3),IFLOW(5)
8
9  C      TURN OFF OLD SPS
10     DO 3 IENT=370,389
11         CALL GEOP(IENT)
12 3     CONTINUE
13
14  C  INITIALIZE
15     ISMCH = 0
16     IPTR=INTF(6)
17     CALL GHLT
18     CALL GCYT(1)
19     IENT = 389
20     CALL TOFMSG(5)
21
22  C  SET CURSOR TO CURRENT TRACKBALL POSITION.
23     CALL GENT(10)
24     CALL GTRACK (IX,IY)
25     CALL GPUT (1,100,IX,0)
26     CALL GPUT (2,110,IY,0)
27     CALL GEON (10)
28
29  C  GET UP TO A MAXIMUM OF 10 RADAR AND VISUALLY SIGNIFICANT POINTS
30     DO 200 K=1,2
31         IDEF (IPTR) = 0
32         CALL MESSAG (K+6)
33         DO 100 J=1,10
34             LSMCH(J) = .FALSE.
35             CALL GEOP (100)
36             CALL TOFMSG(3)
37             CALL LATLON (1,0,130)
38 5         IATT(J) = 0
39             CALL GSTT(0,0)
40
41  C  CHECK ON INTERRUPTS
42 10     CALL CKINT(KEY)
43         IF (KEY .EQ. -1) GO TO 10
44         IF (KEY .EQ. 7) GO TO 150
45         IF (KEY .EQ. 1) GO TO 50
46         IF (KEY .EQ. 19) GO TO 40
47         IF (KEY .NE. 40) GO TO 5
48
49  C  CYCLE TIMER INTERRUPT, DISPLAY CURRENT TRACKBALL POSITION.
50     CALL GTRACK (IX,IY)
51     ISMCH = 1-ISMCH
52     CALL GENT (10)
53     CALL GPUT (1,100,IX,0)
54     CALL GPUT (2,110,IY,0)
55     IF (ISMCH .EQ. 0) GO TO 25
56     CALL LATLON (IX,IY)
57     GO TO 10
58

```

```

59 C UPDATE RANGE AND BEARING
60 25 IF (.NOT. LSMCH(1)) GO TO 10
61 CALL FRNGE (IX,IY)
62 CALL LAPS (0,0,0,130)
63 GO TO 10
64
65 C RANGE AND BEARING ROUTINE
66 40 IF (LSMCH(1)) GO TO 25
67 IX=-100
68 IY = 0
69 CALL FRNGE (IX,IY)
70 CALL HESSNG (K+6)
71 CALL LAPS (0,0,0,130)
72 IF (.NOT. LSMCH(1)) CALL GLMP (1,19,1)
73 GO TO 10
74
75 C ACCEPT KEY INTERPUOT. PLACE SIGNIFICANT POINTS IN MDP
76 C AND DRAW SYMBOL IN PROPER LOCATION ON THE SCREEN.
77 50 MDP(IPTR+2*J-1) = IX
78 MDP(IPTR+2*J) = IY
79
80 C TURN ON SIGNIFICANT POINT ENTITY
81 CALL GENT (ERR=0,0,0)
82 CALL GENT (ERR=0,0,0)
83 CALL GENT (ERR=0,0,0)
84 CALL GENT (ERR=0,0,0)
85
86 100 CONTINUE
87 150 IF (TIME) = 1.0
88 IF (TIME) = 1.0
89
90 200 CONTINUE
91
92 C PREPARE TO RETURN TO CALLING PROGRAM
93 CALL GENT (0,-1,0)
94 CALL GENT (0,0,0)
95 CALL GENT (0,0,0)
96 CALL GENT (0,0,0)
97 CALL GENT (0,0,0)
98 CALL GENT (0,0,0)
99 IATT(1) = 0
100 CALL GSTT(0,0)
101 RETURN
102 END
0 ERRORS COMPILATION COMPLETE

```

PAGE 1

PLANT CURRENT FIN IN

00000000

1 SUPERVISE STING AND TWO
2 C SMOKE AND TWO STINGING THE OUTSIDE POSITION OF THE STING BALL
3 C AND TWO STINGING THE OUTSIDE POSITION OF THE STING BALL
4 C IV - THE STINGING POSITION
5 C IV - THE STINGING POSITION
6 C THE STINGING POSITION OF THE STING BALL TWO
7 C IV - THE STINGING POSITION
8 C IV - THE STINGING POSITION
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C

0 LINES CONTAINING CORRECTIONS


```

1 SUBROUTINE LATLON (IN,IV)
2 C SUBROUTINE LATLON CALCULATES THE CURRENT POSITION OF THE
3 C CARRIER FROM MINUTES TO DEGREES LATITUDE AND MINUTES AND
4 C DEGREES LONGITUDE AND MINUTES. THE POSITION IS DISPLAYED
5 C ON THE SCREEN IN ENTRY NUMBER 193.
6 C THE 4TH ELEMENT OF THE INT ARRAY POINTS TO THE ORIGIN
7 C LAT AND LONG POSITION EXPRESSED IN MINUTES AND THE
8 C WIDTH OF THE SCENARIO IN KILOMETERS.
9
10 COMMON MASTER, HLF(2400)
11 CALL GENT (193)
12 CALL GENT (193)
13 IPTR = HLF(5)
14 CVFACT = FLOAT(INT(IPTR*2))/1024.
15
16 C GET CURRENT LATITUDE
17 IIPR = INT(REAL(INT(IPTR*CVFACT*YD)*.5))
18 LATMIN = IIPR/60
19 LATSEC = IIPR - LATMIN*60
20 IF (LATMIN .LT. 0) LATMIN = -LATMIN
21
22 C GET CURRENT LONGITUDE FROM LATITUDE
23 LATVAL = INT(REAL(INT(IPTR*CVFACT*YD)*.5))
24 LONGMIN = INT(REAL(INT(IPTR*CVFACT*XD)*.5))
25 LONGSEC = LONGMIN*60
26 LONGVAL = LONGMIN + LONGSEC/60
27 IF (LONGMIN .LT. 0) LONGMIN = -LONGMIN
28 IF (LONGSEC .LT. 0) LONGSEC = -LONGSEC
29 IF (LONGMIN .LT. 0) LONGVAL = -LONGVAL
30
31 C DISPLAY LATITUDE AND LONGITUDE ON THE SCREEN
32 CALL GENT (193)
33 CALL GENT (193)
34 CALL GENT (193)
35 CALL GENT (193)
36 CALL GENT (193)
37 CALL GENT (193)
38 END

```

0 ERRORS COMPIATION COMPLETE

0 LPROCS COMPILETIME COMPLETE

```

1      SUBROUTINE LINTNP (NO,IENT)
2      C THIS SUBROUTINE TAKES FOR A GIVEN ENTITY ITS LINES IN
3      C VARIOUS COLORS AND FORMS. CONTROL REGISTER (ELEMENT 3)
4      C CONTROLS THE LINE TYPE. DISPLAY REGISTER (ELEMENT 4) CONTROLS COLOR.
5      DATA IRED, IYELLOW, IORANG, IGREEN /0,1,2,3/
6
7      CALL GENT (IENT)
8      GO TO (1,2,3,4,5,6,7,8,9,10),NO
9      C LINE TYPE 1 - RED DASHED LINE
10     1  CALL COLOR (IRED)
11         CALL GPRT (3,130,1,2)
12         RETURN
13     C TERRAIN TYPE 2 - YELLOW DASHED LINE
14     2  CALL COLOR (IYELLOW)
15         CALL GPRT (3,130,1,2)
16         RETURN
17     C LINE TYPE 3 - ORANGE DASHED LINE
18     3  CALL COLOR (IORANG)
19         CALL GPRT (3,130,1,2)
20         RETURN
21     C LINE TYPE 4 - ORANGE DOTTED LINE
22     4  CALL COLOR (IORANG)
23         CALL GPRT (3,130,1,1)
24         RETURN
25     C LINE TYPE 5 - GREEN DOTTED LINE
26     5  CALL COLOR (IGREEN)
27         CALL GPRT (3,130,1,1)
28
29     C LINE TYPE 6 - RED DASHED-DOT
30     6  CALL COLOR (IRED)
31         CALL GPRT (3,130,1,1)
32         RETURN
33     C LINE TYPE 7 - ORANGE DASHED-DOT
34     7  CALL COLOR (IORANG)
35         CALL GPRT (3,130,1,1)
36         RETURN
37     C LINE TYPE 8 - YELLOW DASH-DOT
38     8  CALL COLOR (IYELLOW)
39         CALL GPRT (3,130,1,3)
40         RETURN
41     C LINE TYPE 9 - GREEN DASH-DOT
42     9  CALL COLOR (IGREEN)
43         CALL GPRT (3,130,1,3)
44         RETURN
45     C LINE TYPE 10 - YELLOW SOLID
46     10 CALL COLOR (IYELLOW)
47         RETURN
48     END
0 ERRORS COMPILATION COMPLETE

```

PAGE

1

1000000

1000000

```

1  SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE
2  SUBROUTINE SUBROUTINE
3  SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
4  SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
5  SUBROUTINE SUBROUTINE
6  SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
7  SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
8  SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
9  SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
10 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
11 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
12 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
13 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
14 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
15 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
16 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
17 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
18 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
19 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE
20 SUBROUTINE SUBROUTINE, SUBROUTINE, SUBROUTINE, SUBROUTINE

```

0 1000000 SUBROUTINE SUBROUTINE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

10

[illegible]

[illegible]

```

1      SUBROUTINE MESSAGING
2      C SUBROUTINE MESSAGE DISPLAYS BELOW THE SCENARIO BORDER A MESSAGE
3      C DESCRIBING AN INSTRUCTION OR AN ERROR MESSAGE. THE AREA USED FOR
4      C THE MESSAGES IS SET UP IN ENTITY 191.
5
6      C BLANK OUT MESSAGE AREAS
7      CALL GSCN (191,6)
8      WRITE (15,1110)
9      CALL GSCN (191,60)
10     WRITE (15,1110)
11     CALL GSCN (191,114)
12     WRITE (15,1110)
13     CALL GSCN (191,168)
14     WRITE (15,1110)
15 1110 FORMAT (' ')
16     GO TO (100,200,300,350,400,450,500,550,600,650),ND
17
18      C MOVE NODE MESSAGE
19 100  CALL GSCN (191,6)
20     WRITE (15,110)
21 110  FORMAT (15,'--MOVE NODE--')
22     CALL GSCN(191,60)
23     WRITE (15,115)
24 115  FORMAT (' MOVE NODE TO DESIRED POSITION AND PRESS "ACCEPT".')
25     CALL GSCN (191,114)
26     WRITE (15,120)
27 120  FORMAT (' "RETURN" KEY REPOSITIONS NODE TO INITIAL LOCATION. ')
28     GO TO 999
29
30      C SELECT NODE MESSAGE
31 200  CALL GSCN(191,6)
32     WRITE (15,210)
33 210  FORMAT(14X,'--SELECT NODE--')
34     CALL GSCN (191,60)
35     WRITE (15,215)
36 215  FORMAT (' ENTER MESSAGEID TO POSITION THE CURSOR OVER THE ')
37     CALL GSCN (191,114)
38     WRITE (15,220)
39 220  FORMAT (' DESIRED NODE AND PRESS THE "ACCEPT" KEY')
40     GO TO 999
41
42      C RANGE AND BEARING MESSAGES
43 300  CALL GSCN (191,6)
44     WRITE (15,1310)
45 1310 FORMAT (10X,'-- RANGE AND BEARING --')
46     CALL GSCN (191,60)
47     WRITE(15,1320)
48 1322 FORMAT (' SELECT REFERENCE POINT WITH CURSOR')
49     CALL GSCN (191,114)
50     WRITE (15,1325)
51 1325 FORMAT (' PRESS "ACCEPT" KEY TO FIX REFERENCE POINT')
52     CALL GSCN (191,168)
53     WRITE (15,1345)
54 1345 FORMAT (' PRESS "RETURN" KEY TO EXIT . . .')
55     GO TO 999
56
57      C LAUNCH NODE MESSAGE
58 350  CALL GSCN (191,60)

```

PAGE 2 MRLEVI VORTXII FTH IV(G) 0009 HOURS

```

59      WRITE (15,1350)
60 1350  FORMAT (' LOCATE ''LAUNCH'' POSITION WITH CURSOR.  PRESS')
61      GO TO 475
62
63 C TARGET NODE MESSAGE
64 400   CALL GSCH (191,60)
65      WRITE (15,1400)
66 1400  FORMAT (' LOCATE ''TARGET'' POSITION WITH CURSOR.  PRESS')
67      GO TO 475
68
69 C RECOVERY NODE MESSAGE
70 450   CALL GSCH (191,60)
71      WRITE (15,1450)
72 1450  FORMAT (' LOCATE ''RECOVERY'' POSITION WITH CURSOR.  PRESS')
73 475   CALL GSCH (191,114)
74      WRITE (15,1375)
75 1375  FORMAT (' ''ACCEPT'' KEY WHEN CURSOR IS OVER DESIRED POSITION')
76      GO TO 999
77
78 C RADAR SIGNIFICANT POINTS
79 500   CALL GSCH (191,60)
80      WRITE (15,1500)
81 1500  FORMAT ('MOVE CURSOR TO POSITION OF RADAR SIGNIFICANT')
82      GO TO 525
83
84 C VISUAL SIGNIFICANT POINTS
85 550   CALL GSCH (191,60)
86      WRITE (15,1550)
87 1550  FORMAT ('MOVE CURSOR TO POSITION OF VISUAL SIGNIFICANT')
88 525   CALL GSCH (191,114)
89      WRITE (15,1505)
90 1555  FORMAT ('POINTS AND PRESS ''ACCEPT'' KEY.  WHEN COMPLETED')
91      CALL GSCH (191,163)
92      WRITE (15,1560)
93 1560  FORMAT ('PRESS ''RETURN'' KEY.  ')
94      GO TO 999
95
96 C RECALL SCENARIO MESSAGE
97 600   CALL GSCH (191,60)
98      WRITE (15,1600)
99      CALL GSCH (191,60)
100     WRITE (15,1610)
101 1600  FORMAT ('SELECT MAP FROM MENU THEN ENTER VERSION')
102 1610  FORMAT ('(BETWEEN 1 AND  ) FROM NUMBER PAD.')
103     GO TO 999
104
105 C STORE SCENARIO MESSAGE
106 1000  CALL GSCH (191,60)
107     WRITE (15,1700)
108     CALL GSCH (191,60)
109     WRITE (15,1600)
110 1650  FORMAT ('ENTER NUMBER BETWEEN 2 AND  FROM')
111 1660  FORMAT ('THE NUMBER PAD TO IDENTIFY THE SCENARIO.')
112     GO TO 999
113
114 999   CALL GSCH (191)
115     RETURN
116     END

```

```

1  SUBROUTINE MOVE(NODE)
2  COMMON/PATH/INTT(12),IEPR
3  COMMON/ROUTE/ IPATH(65)
4  COMMON/FLAGS/ LNODE(9),RXPT(9),LSWCH(3),IFLOW(5)
5  CALL MESSAGE(1)
6
7  C  SUBROUTINE MOVE(NODE) ALLOWS THE USER TO MOVE THE POSITION OF A
8  C  NODE ALONG THE STRIKE PATH. RETURN KEY #7 WILL REPLACE THE NODE
9  C  TO ITS ORIGINAL POSITION.
10 C  NODE - THE NODE THAT IS TO CHANGE.
11
12 C  GET THE CURRENT POSITION OF THE TRACKBALL
13  CALL GETPRC(IX,IY)
14  CALL GETPRC(INOLD,INOLD)
15  ISWCH = .TRUE.
16  I = GNODE-2
17  CALL GETI(300)
18
19 C  GET ELEMENT NUMBER OF THE NODE
20  IEL = 1+GNODE
21  CALL LUTPS(0,0,10,130)
22  CALL GULT
23  CALL GOUT(1)
24  5  IATT(1)=0
25  CALL GSTT(0,0)
26  10  CALL CHIT(KEY)
27  IF (KEY.EQ. -1) GO TO 10
28
29  IF (KEY.EQ. 1) GO TO 20
30  IF (KEY.EQ. 7) GO TO 25
31  IF (KEY.EQ. 12) CALL DECLUT
32  IF (KEY.NE. 40) GO TO 5
33
34  C  CYCLE TIMER INTERRUPT:  DISPLAY NEW NODE POSITION.
35  CALL GETPRC(IX,IY)
36  CALL GETPRC(IX,IY,5)
37  CALL GOUT(1)
38  CALL GOUT(12,IEPR,IX,IY)
39  IF (.NOT. INTT) CALL LUTLON(IX,IY)
40  IF (RXPT(IX).EQ. 1) CALL PRNGE (IX,IY)
41  ISWCH = .NOT. ISWCH
42  GO TO 10
43
44  C  KEY INTERRUPT:
45  20  IPATH(1)=IX
46  IPATH(I+1)=IY
47
48  C  SET OFF CYCLE TIMER AND PREPARE TO EXIT FROM SUBROUTINE
49  25  CALL GULT
50  CALL GOUT(1)
51  CALL GOUT(12,IEPR)
52  CALL LUTLON(IX,IY)
53  CALL LUTLON(IX,IY)
54  CALL LUTLON(IX,IY)
55  CALL LUTLON(IX,IY)
56  CALL LUTLON(IX,IY)
57  CALL LUTLON(IX,IY)
58  CALL LUTLON(IX,IY)
59  CALL LUTLON(IX,IY)
60  CALL LUTLON(IX,IY)
61  CALL LUTLON(IX,IY)
62  CALL LUTLON(IX,IY)
63  CALL LUTLON(IX,IY)
64  CALL LUTLON(IX,IY)
65  CALL LUTLON(IX,IY)
66  CALL LUTLON(IX,IY)
67  CALL LUTLON(IX,IY)
68  CALL LUTLON(IX,IY)
69  CALL LUTLON(IX,IY)
70  CALL LUTLON(IX,IY)
71  CALL LUTLON(IX,IY)
72  CALL LUTLON(IX,IY)
73  CALL LUTLON(IX,IY)
74  CALL LUTLON(IX,IY)
75  CALL LUTLON(IX,IY)
76  CALL LUTLON(IX,IY)
77  CALL LUTLON(IX,IY)
78  CALL LUTLON(IX,IY)
79  CALL LUTLON(IX,IY)
80  CALL LUTLON(IX,IY)
81  CALL LUTLON(IX,IY)
82  CALL LUTLON(IX,IY)
83  CALL LUTLON(IX,IY)
84  CALL LUTLON(IX,IY)
85  CALL LUTLON(IX,IY)
86  CALL LUTLON(IX,IY)
87  CALL LUTLON(IX,IY)
88  CALL LUTLON(IX,IY)
89  CALL LUTLON(IX,IY)
90  CALL LUTLON(IX,IY)
91  CALL LUTLON(IX,IY)
92  CALL LUTLON(IX,IY)
93  CALL LUTLON(IX,IY)
94  CALL LUTLON(IX,IY)
95  CALL LUTLON(IX,IY)
96  CALL LUTLON(IX,IY)
97  CALL LUTLON(IX,IY)
98  CALL LUTLON(IX,IY)
99  CALL LUTLON(IX,IY)
100 CALL LUTLON(IX,IY)

```

PAGE 1 RELEASE DATE: 1971-10-01

1. SUBJECT: [REDACTED]
2. [REDACTED]
3. [REDACTED]
4. [REDACTED]
5. [REDACTED]
6. [REDACTED]
7. [REDACTED]

```

1 SUBROUTINE OUTSEN (IVERS)
2 COMMON /MASTER/ NDF(2400)
3 COMMON /ADMIN/ INFOFH(120)
4 INTEGER SCENE(3,20), IFCB(13)
5 DATA SCENE(1, 1), SCENE(2, 1), SCENE(3, 1) /2H5C,2HEN,2H12/
6 DATA SCENE(1, 2), SCENE(2, 2), SCENE(3, 2) /2H5C,2HEN,2H22/
7 DATA SCENE(1, 3), SCENE(2, 3), SCENE(3, 3) /2H5C,2HEN,2H13/
8 DATA SCENE(1, 4), SCENE(2, 4), SCENE(3, 4) /2H5C,2HEN,2H42/
9 DATA SCENE(1, 5), SCENE(2, 5), SCENE(3, 5) /2H5C,2HEN,2H12/
10 DATA SCENE(1, 6), SCENE(2, 6), SCENE(3, 6) /2H5C,2HEN,2H23/
11 DATA SCENE(1, 7), SCENE(2, 7), SCENE(3, 7) /2H5C,2HEN,2H24/
12 DATA SCENE(1, 8), SCENE(2, 8), SCENE(3, 8) /2H5C,2HEN,2H13/
13 DATA SCENE(1, 9), SCENE(2, 9), SCENE(3, 9) /2H5C,2HEN,2H14/
14 DATA SCENE(1,10), SCENE(2,10), SCENE(3,10) /2H5C,2HEN,2H24/
15 DATA SCENE(1,11), SCENE(2,11), SCENE(3,11) /2H5C,2HEN,2H24/
16 DATA SCENE(1,12), SCENE(2,12), SCENE(3,12) /2H5C,2HEN,2H24/
17 DATA SCENE(1,13), SCENE(2,13), SCENE(3,13) /2H5C,2HEN,2H24/
18 DATA SCENE(1,14), SCENE(2,14), SCENE(3,14) /2H5C,2HEN,2H24/
19 DATA SCENE(1,15), SCENE(2,15), SCENE(3,15) /2H5C,2HEN,2H24/
20 DATA SCENE(1,16), SCENE(2,16), SCENE(3,16) /2H5C,2HEN,2H24/
21 DATA SCENE(1,17), SCENE(2,17), SCENE(3,17) /2H5C,2HEN,2H24/
22 DATA SCENE(1,18), SCENE(2,18), SCENE(3,18) /2H5C,2HEN,2H24/
23 DATA SCENE(1,19), SCENE(2,19), SCENE(3,19) /2H5C,2HEN,2H24/
24 DATA SCENE(1,20), SCENE(2,20), SCENE(3,20) /2H5C,2HEN,2H24/
25
26 INFOFH = 0
27 INFOFH = 1
28 NDFCN = NDFCN+INFOFH-2
29 NDFCN = NDFCN+INFOFH
30 INFOFH = SCENE(1,INFOFH)
31 INFOFH = SCENE(2,INFOFH)
32 CALL VTCN(1,15,15,1,2400)
33 CALL VTCN(1,15,15,1,2400)
34 CALL VTCN(1,15,15,1,2400)
35 IFCB = 0
36
37 O LAPORS COMPILE COMPLETED

```

PAGE 1

RELEASED UNDER E.O. 12958

DATE 11/15/05

1 SUBPOENA DUCASOL
2 GILL TAPPI
3 DUCASOL
4 TAPPI
5 DUCASOL TAPPI CONFERENCE

PAGE 1

RELEVANT

WORTHY

FIN 1900

0-10-1-100

1. SUPPLEMENTAL RECORDS
 2. GALL. 1011
 3. 1011
 4. 1011
0. RECORDS OF PLANTATION COMPANY

PAGE 1

RELEVANT

EXCLUDED

FIN. INFO.

CONFIDENTIAL

1 SUBJ: JAMES EARL RAY
2 CUBA, HAITI
3 EL SALVADOR
4 ECUADOR
O LACERS CO. (LATERAL OUTLINE)

```

1 SUBROUTINE RANGE (IX,IY)
2 C SUBROUTINE RANGE CALCULATES AND DISPLAYS THE RANGE AND
3 C BEARING FROM THE CURSOR TO A REFERENCE POINT. THE FIRST
4 C PASS THROUGH RANGE SELECTS A REFERENCE POINT. ALL SUBSEQUENT
5 C CALLS CALCULATE AND DISPLAY THE RANGE AND BEARING. THE
6 C RANGE IS CALCULATED IN NAUTICAL MILES AND BEARING IS MEASURED
7 C FROM NORTH FOR 360 DEGREES.
8
9 C PARAMETERS
10
11 C FIRST PASS -
12 C IX - NEGATIVE NUMBER LESS THAN -10 TO IDENTIFY FIRST CALL
13 C IY - IGNORED
14 C THE CURSOR COORDINATES ARE PASSED BACK TO THE CALLING ROUTINE.
15
16 C AFTER FIRST CALL -
17 C IX - X POSITION OF CURSOR
18 C IY - Y POSITION OF CURSOR
19 C NOTHING IS ALTERED IN THE SUBROUTINE.
20
21 COMMON /MASTER/ MDE(2400)
22 COMMON /ATT/ IATT(12),IERR
23 COMMON /FLAGS/ LINDX(9),REPT(9),LSWCH(3),IFLOW(5)
24 IF (IX .GT. -10) GO TO 500
25
26 C FIRST PASS SECTION
27 C PRELIMINARIES
28 CALL MERRAS(3)
29 CALL LINDX(0,0,0,130)
30 CALL TAIL(0)
31 CALL GENT(1,100,IY,0)
32 CALL GENT(2,110,IY,0)
33
34 C CHECK ON INTERRUPTS
35 S IATT(1) = 0
36 CALL GENT(10,0)
37 10 CALL GENT(11,0)
38 IF (KEY .EQ. -1) GO TO 10
39 IF (KEY .EQ. 1) GO TO 100
40 IF (KEY .EQ. 7) GO TO 200
41 IF (KEY .NE. 40) GO TO 5
42
43 C CYCLE TIMER INTERRUPT
44 CALL GTPACK (IX,IY)
45 CALL TOLDS (IYOLD,IYOLD,IX,IY,5)
46 CALL GENT (10)
47 CALL GPUT (1,100,IX,0)
48 CALL GPUT (2,110,IY,0)
49 CALL LATON (IX,IY)
50 GO TO 10
51
52 C ACCEPT KEY INTERRUPT
53 C REFERENCE POINT HAS BEEN IDENTIFIED
54 100 CALL GTPACK (IX,IY)
55 RZ = FLOAT(IX)
56 RY = FLOAT(IY)
57 CALL GENT (150)
58 CALL GPUT (1,100,IX,0)

```

```

59 CALL GPUT (2,110,IY,0)
60 LSWCH(1) = .TRUE.
61 RETURN
62
63 300 CALL TPTNCG(2)
64 CALL GEOF (100)
65 RETURN
66
67 C SUBSEQUENT PASSES AFTER THE FIRST
68
69 C RUBBER BAND LINE BETWEEN 2 POINTS
70 500 CALL GENT (100)
71 CALL GPUT (7,53,IX,IY)
72 CALL GENT (100)
73 ISTR = ISTR(5)
74 ISTR = ISTR(5) + 1
75
76 C CALCULATE DISTANCE AND ANGLE TO SECOND POINT
77 DIST = SQRT( (X2-X1)**2 + (Y2-Y1)**2 )
78 ANGLE = ATN2( Y2-Y1, X2-X1 )
79 ISTR = ISTR(5) + 1
80
81 C CHECK ON ANGLE
82 IF (ANGLE .GT. 1.0) GO TO 550
83 ISTR = ISTR(5)
84 IF (ISTR .EQ. 0.0) ISTR = ISTR(5) + 1
85 GO TO 500
86
87 550 ISTR = ISTR(5) + 1
88 IF (ISTR .EQ. 0.0) ISTR = ISTR(5) + 1
89
90 C DISTANCE AND ANGLE
91 500 CALL GENT (100)
92 CALL GPUT (7,53,IX,IY)
93 CALL GENT (100)
94 RETURN
95
96 C ERRORS COMPILATION COMPLETE

```

1 SUPPLEMENTAL PAGE 10

[illegible][illegible]

A C F P U S A

THE FIVE FINGERED HAND

5. The following are the results of the regression analysis:

DATE: 11/11/1964

100-443887-100

9. *By an Hon. Member, Mr. G. M. G. M.*

ALL INFORMATION CONTAINED
HEREIN IS UNCLASSIFIED

11. 1900-1901 (H¹ 9540, J 1, 170)

54. 2.16. 1920

1991

1. The first group of people who are not in the labor force are those who are not in the labor force because they are not in the labor force.

OUTLINES OF LECTURE ONE

```

1      SUBROUTINE RECALL
2  C   SUBROUTINE RECALL ALLOWS THE USER TO RECALL A MAP FROM DISK
3  C   VIA A MENU AND SELECT A VERSION WITH THE NUMBER PAD. IT
4  C   WILL DRAW A PATH IF ONE EXISTS AND DISPLAY SIGNIFICANT POINTS
5
6      COMMON /ADIMIN/ INFOCN(120)
7      COMMON /MSTEP/ MEF(2400)
8      COMMON /AID/ IATT(12), IERR
9      COMMON /DISP/ IDFL(4000), MARKR(12)
10     COMMON /ROUTE/ IPATH(65)
11     COMMON /FLAGZ/ LNODE(9), PXPT(9), LSWCH(3), IFLOW(5)
12
13     CALL GINI(IDFL,4000,IATT,IERR)
14     CALL SETUP
15     CALL TOPDISG(10)
16     CALL MESSAGE(9)
17     LNODE(1)=1
18
19  C   DISPLAY MAP OPTIONS
20  C   CHECK IF ONLY ONE MAP EXISTS, IF SO NO SELECTION TAKES PLACE
21     NMPT = INFOCN(1)
22     IF (NMPT .NE. 1) GO TO 25
23     MAP = 1
24     GO TO 30
25
26 25     CALL OPTIMU(6)
27     CALL SELECT (NMPT,MAP)
28     IF (MAP .EQ. 0) GO TO 35
29     CALL RESET
30 30     CALL INSEN (MAP,0)
31     CALL DEFSCH
32     INFOCN(3) = MAP
33     IFUS = 35*MAP-4
34     IF (INFOCN(IFUS) .LE. 1) GO TO 50
35
36  C   USE NUMBER PAD TO INPUT VERSION
37     CALL GETVER(1)
38     CALL GINI(1,INFOCN(IFUS),2)
39     CALL GETVER(1)
40     CALL GETVER(1)
41     CALL SELECT (INFOCN(IFUS),110*MAP,24,IVERS)
42     IF (IVERS .LE. 0) IVERS = 1
43     CALL INSEN (MAP,IVERS)
44
45  C   PUT PATH INTO IPATH ARRAY FROM MDF
46     IPTR = MEF(3)-1
47     DO 20 J=1,65
48         IPATH(J)=MDF(IPTR+J)
49 20     CONTINUE
50     IF (IPATH(1) .EQ. 0) GO TO 45
51
52  C   DRAW PATH AND SWHEELS OVER LAUNCH, TARGET AND RECOVERY
53 35     IF (MAP .EQ. 0) CALL DEFSCH
54     CALL INPATH
55     CALL GEUT (310)
56     CALL GEUT (1,100,IPATH(1),0)
57     CALL GEUT (2,110,IPATH(5),0)
58     IFUS = IPATH(2)*6-2

```

50 CALL TYPED (TYP)
 51 CALL TYPED (TYP, H015, H015, 0, 0)
 52 CALL TYPED (TYP, H015, H015, 0, 0)
 53 CALL TYPED (TYP, H015, H015, 0, 0)
 54 CALL TYPED (TYP, H015, H015, 0, 0)
 55 CALL TYPED (TYP, H015, H015, 0, 0)
 56 CALL TYPED (TYP)
 57 CALL TYPED (TYP)
 58 CALL TYPED (TYP)
 59 CALL TYPED (TYP)
 60
 61 45 CALL TYPED
 62 IF (TYP) GOTO 100
 63
 64 50 CALL TYPED
 65 CALL TYPED
 66 CALL TYPED
 67 CALL TYPED (TYP)
 68 TYPED
 69 END
 0 ERRORS COMPILATION COMPLETE

[illegible]

```

1  SUBROUTINE SELECT(OPTION,CHOICE)
2  IMPLICIT INTEGER (A-C,E-Z)
3  DATA IPED,IVLUM,IGRUS,IGREEN/0,1,2,3/
4  C **** ROUTINE ENABLES USER TO SELECT ONE OF A LIST OF OPTIONS.
5  C **** OPTIONS ARE DISPLAYED IN RED ON THE LEFT SIDE OF THE SCREEN.
6  C **** USER MANIPULATES A POINTING CURSOR THAT PRECEDES THE GREEN
7  C **** CHARACTERS OF THE POINTED-AT OPTION.
8
9  C ****      OPTION      NUMBER OF MEMBERS IN LIST
10 C ****      CHOICE      RELATIVE POSITION IN LIST OF CHOSEN OPTION
11
12
13 C      TURN ON MENU AND LIGHT LAMPS
14 C      CALL MENUON
15 C      CALL LAMPS(0,0,0,131)
16
17 C **** POINTER 'INC' IS 0 AND POINTS TO THE FIRST OPTION IN THE LIST
18 C      INC=0
19
20 C **** POSITION THE CURSOR POINTER
21 C      CALL GENT(11)
22 C      CALL GFUT(1,100,-329,0)
23 C      CALL GFUT(2,110,957,0)
24 C      CALL GEON(11)
25
26 C **** COLOR THE FIRST ITEM IN THE LIST GREEN
27 C      CALL GENT(11)
28 C      CALL COLOR(IGREEN)
29
30 C **** POLL THE FUNC. KEYBOARD FOR AN ACCEPT/REJECT
31 C      1  CALL CRINT(KEY)
32
33 C      IF(KEY.NE.0)GO TO 2
34
35 C      REJECT KEY HIT
36 C      IF KEY=1
37 C      IF (KEY.EQ.1)OPTION=INC+1
38 C      IF (KEY.EQ.2)OPTION=INC-1
39
40 C      COLOR LAST ITEM RED AGAIN AND REPOSITION CURSOR
41 C      CALL COLOR(IPED)
42 C      CALL GENT(11)
43 C      CALL GFUT(2,110,Y,0)
44
45 C **** COLOR NEW ITEM GREEN
46 C      CALL GENT(11+INC)
47 C      CALL COLOR(IGREEN)
48 C      CALL DELAY(30)
49 C      GO TO 1
50
51 C      2  IF(KEY.NE.1)AND(KEY.NE.7)GO TO 1
52
53 C      ACCEPT KEY HIT
54 C      IF (KEY.EQ.1)OPTION=INC+1
55
56 C      REJECT KEY HIT
57 C      IF (KEY.EQ.2)OPTION=INC-1
58

```



```

1  SUBROUTINE SELNOD(NODE)
2  C SELECT NODE SUBROUTINE
3  C THIS SUBROUTINE ALLOWS THE USER TO MOVE A BLINKING FIGURE (CURSOR)
4  C AROUND THE DISPLAY TO SELECT A NODE. WHEN THE CURSOR GETS WITHIN
5  C 10 UNITS OF A NODE THE NODE WILL BLINK. AS THE CURSOR MOVES AWAY, THE
6  C NODE STOPS BLINKING. WHEN A NODE IS CHOSEN, THE NODE NUMBER IS RETURNED.
7
8  COMMON /ATT/ IATT(12), IERR
9  COMMON /ROUTE/ IPATH(65)
10 COMMON /FLAGS/ LNODE(9), RXP(9), LSWCH(3), IFLOW(5)
11 CALL TOPNG (4)
12 CALL MESSNG (2)
13 ISWCH = 0
14 C SET OFF ALL BUT THE ACCEPT LAMP
15 CALL LAMP(0, 8, 16, 128)
16
17 C NODCHK CONTAINS THE NODE NUMBER OF A BLINKING NODE
18 NODCHK = 0
19 LSWCH(1) = .FALSE.
20 IEND = IPATH(1)
21
22 C MOVE CURSOR TO CURRENT NODE
23 CALL GTRCK (IX, IY)
24 CALL GTRCK (IXOLD, IYOLD)
25 CALL GENT(10)
26 CALL GRUT (1, 100, IX, 0)
27 CALL GRUT (2, 110, IY, 0)
28 CALL GENT(10)
29
30 C SET UP CYCLE TIMER AND CHECK INTERRUPTS
31 CALL GALT
32 CALL GENT(1)
33 5 IATT (1) = 0
34 CALL GETT(0, 0)
35 10 CALL CHNTOLEV
36 IF (KEY .EQ. -1) GO TO 10
37
38 IF (KEY .EQ. 19) GO TO 45
39 IF (KEY .EQ. 1) GO TO 50
40 IF (KEY .EQ. 12) CALL DELUT
41 IF (KEY .NE. 7) GO TO 15
42 NODCHK = 0
43 CALL GENT (120)
44 CALL TOPNG (2)
45 GO TO 55
46 15 IF (KEY .NE. 40) GO TO 5
47
48 C CYCLE TIMER INTERRUPT!
49 C MOVE CURSOR AND CHECK TO SEE IF ITS CLOSE TO A NODE
50 CALL GTRCK (IX, IY)
51 CALL TOLOS (IXOLD, IYOLD, IX, IY, 5)
52 ISWCH = 1-ISWCH
53 CALL GENT(10)
54 CALL GRUT (1, 100, IX, 0)
55 CALL GRUT (2, 110, IY, 0)
56 IF (ISWCH .EQ. 0) GO TO 16
57 CALL LATLOH (IX, IY)
58 GO TO 13

```

- 134 -

```

1      SUBROUTINE SELNUM (MAX,IENT,IEL,NUMBER)
2 C    SUBROUTINE SELNUM ALLOWS THE USER TO ENTER A NUMBER FROM THE
3 C    NUMBER PAD ON THE SPECIAL FUNCTION UNIT. IT ALLOWS UP TO
4 C    FOUR DIGITS TO BE ENTERED AND DISPLAYS EACH DIGIT AS ENTERED
5 C    IN A SPECIFIED LOCATION.
6
7 C    PARAMETERS
8 C      MAX - THE MAXIMUM ALLOWABLE NUMBER.
9 C      IENT - ENTITY WHERE NUMBER IS TO BE DISPLAYED.
10 C      IEL - ELEMENT IN CHARACTER AREA WHERE NUMBER ENDS (UNITS
11 C          POSITION). FOUR CHARACTER SPACES MUST BE AVAILABLE.
12 C      NUMBER - VALUE OF NUMBER ENTERED TO BE RETURNED TO CALLING PROG
13 C          RETURNS -9999 IF NO NUMBER IS DESIRED.
14
15      DIMENSION NO(10),INT(4)
16      COMMON /ATTX/ INT(12),IERR
17      DATA NO /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
18
19      CALL GENT (IENT)
20      CALL GEON (IENT)
21
22      4      NUMBER = 0
23      CALL LAMP'S (7,7,7,132)
24
25      DO 100 IDEC = 1,4
26      5      INT(1) = 0
27      CALL GSTT(0,0)
28
29 C    CHECK ON INTERPRET!
30 10      CALL CRINT (KEY)
31          IF (KEY.EQ. -1) GO TO 10
32          IF (KEY.GT. 26) GO TO 10
33          IF (KEY.EQ. 7) GO TO 400
34          IF (KEY=1) 300,200,100
35 C    GET INTEGER AND DISPLAY IT
36 100      IF (KEY.EQ. -1) GO TO 110
37          INT(1) = KEY
38          GO TO 100
39
40 110      IF (KEY.GT. 10) GO TO 120
41          IF (KEY.LT. 7) GO TO 10
42          INT(1AC) = 11-KEY
43          GO TO 150
44
45 120      IF (KEY.GT. 12) GO TO 130
46          IF (KEY.LT. 10) GO TO 10
47          INT(1DEC) = 22-KEY
48          GO TO 150
49
50 130      INT(1DEC) = 33-KEY
51
52 C    DISPLAY DIGIT RIGHT JUSTIFIED
53 150      DO 170 K=1,1DEC
54          CALL GAUT(IEL+2-K,20,NO(INT(1DEC-K+1)+1),0)
55 170      CONTINUE
56          CALL CLMP (1,0,1)
57          CALL CLMP (1,1,1)
58 180      CONTINUE

```

```

50
60 C ACCEPT KEY PRESSED, CHECK IF NUMBER IS OVER MAXIMUM
61 000 IF CURR - MAX THEN
62     GOTO 001
63     IF CURR - MAX THEN
64         GOTO 001
65     CONTINUE
66     IF CURR - MAX THEN
67         GOTO 001
68     CONTINUE
69     IF CURR - MAX THEN
70         GOTO 001
71 C ACCEPT KEY PRESSED, TURN OUT DISPLAY AREA AND START OVER
72 000 IF CURR - MAX THEN
73     GOTO 001
74 000 CONTINUE
75 000 CONTINUE
76 C ACCEPT KEY PRESSED
77 000 IF CURR - MAX THEN
78 400 GOTO 001
79 000 CONTINUE
80 000 CONTINUE
81 000 CONTINUE
000000 CONTINUE

```

-137-

```

1 SUBROUTINE SETUP
2 C THIS ROUTINE SETS UP THE BORDER AND CHARACTER AREAS FOR THE MENU
3 COMMON /ATTN/ IATT(12), IERR
4 COMMON /FLWS/ LFLOW(9), RMPT(9), LSMCH(3), IFLOW(5)
5 DIMENSION NUMBER(10)
6 INTEGER ENTITY
7 DATA NUMBER /1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9/
8 DATA IRED, IYELLOW, IORNG, IGREEN/0, 1, 2, 3/
9
10 C SET UP DOTTED
11 CALL GREG(2, 0, 0)
12 CALL GPUT(3, 10, 2, 3)
13 CALL COLOR(IRED)
14 CALL GPUT(5, 170, 0, 1)
15 CALL GPUT(6, 51, 1003, 0)
16 CALL GPUT(7, 53, 1003, 0)
17 CALL GPUT(8, 51, -1003, 0)
18 CALL GPUT(9, 52, -1003, 0)
19
20 C SET UP TAYLORBALL CURSOR IN ENTITY (10)
21 CALL GREG(10, 0, 0)
22 CALL COLOR(IYELLOW)
23 CALL GPUT(5, 170, 0, 1)
24 CALL GPUT(6, 160, 10, 0)
25 CALL GPUT(7, 71, 15, 0)
26 CALL GPUT(8, 51, -30, 0)
27 CALL GPUT(9, 70, 15, 15)
28 CALL GPUT(10, 52, -30, 0)
29 CALL GPUT(11, 72, 15, 0)
30 CALL GEOF(10)
31
32 C SET UP MENU CURSOR AS ENTITY 11
33 CALL GREG(11, 0, 0)
34 CALL GEOF(11)
35 CALL GPUT(3, 170, 2, 0)
36 CALL GPUT(4, 160, 10, 0)
37 CALL GPUT(5, 170, 0, 1)
38 CALL GPUT(6, 160, 10, 0)
39 CALL GPUT(7, 170, 0, 1)
40
41 C SET UP CHARACTER AREAS
42
43 DO 10 I=0, 10
44 IENT=110+I
45 IY=1000-50*I
46 CALL GREG(IENT, -330, IY)
47 CALL GEOF(IENT)
48 CALL COLOR(IRED)
49 CALL GPUT(5, 1770, 1, 1)
50 CALL GCH(IENT, 6, 0, 1, 25)
51 10 CONTINUE
52
53 CALL GPUT(110)
54 CALL GPUT(3, 130, 2, 1)
55 CALL GPUT(4, 140, 5, 0)
56 CALL GPUT(4, 140, 0, 1)
57 CALL GPUT(32, 100, 100, 0)
58 CALL GPUT(33, 110, 1000, 0)

```

```

59 CALL GRUT(34,114,50,0)
60 CALL GRHA(110,35,0,2,25)
61
62 C SET UP LEG SPEED/ALT TABLE IN ENTITIES 150-160
63 DO 30 F=1,11
64 DO 30 K=1,18
65     IENT=1401K
66     CALL GDLG(IENT,1023,1000-30WK)
67     CALL GDTOT(IENT)
68     CALL GRUT (6,104,20,0)
69     CALL GDSACIENT,7,0.1,21)
70 30 CONTINUE
71
72 CALL GRCH (150,7)
73 WRITE (15,1050)
74 1050 FORMAT ('LEG SPEED ALTITUDE')
75
76 C SET UP LINE TO CONNECT TWO POINTS FOR RANGE AND BEARING IN 100
77 CALL GREG (150,0,0)
78 CALL LINEP (7,100)
79 CALL GRUT (6,100,0,0)
80 CALL GRUT (7,73,0,0)
81
82 C SET UP TOP OF SCREEN MESSAGE AREA IN 190
83 CALL GSEA (190,0,1023)
84 CALL GSEF (190)
85 CALL GRUT (3,170,2,1)
86 CALL GCLCP (190,0)
87 CALL GRUT (6,114,20,0)
88 CALL GRHA (190,7,0,2,43)
89 CALL GRUT (56,100,-1002,0)
90 CALL GRUT (57,114,50,0)
91 CALL GRHA (190,53,0,2,42)
92 CALL GRUT (107,104,-1003,0)
93 CALL GRHA(190,114,50,0)
94 CALL GRUT (110,100,2,50)
95
96 C SET UP ALTITUDE MESSAGE IN ENTITY 191
97 CALL GREG (191,0,100)
98 CALL GSEF (191)
99 CALL GRUT (3,170,2,1)
100 CALL GCLCP (191,0)
101 CALL GRHA (191,5,0,2,50)
102 CALL GRUT (57,100,0,0)
103 CALL GRUT (58,110,-100,0)
104 CALL GRUT (59,120,3,0)
105 CALL GRHA (191,50,0,2,50)
106 CALL GRUT (111,120,3,0)
107 CALL GRUT (112,100,0,0)
108 CALL GRUT (113,110,-150,0)
109 CALL GRHA (191,114,0,2,50)
110 CALL GRUT (170,100,0,0)
111 CALL GRUT (100,110,-700,0)
112 CALL GRUT (107,100,5,0)
113 CALL GRUT (107,100,6,1)
114 CALL GRHA (191,100,0,2,50)
115
116 C SET UP LONGITUDE AND LATITUDE DISPLAY IN ENTITY 193

```

```

117 CALL GPRG (193,100,140)
118 CALL GPRG (193)
119 CALL GRUT (6,104,24,0)
120 CALL GPRG (193,7,0,1,8)
121 WRITE (15,100)
122 100 FORMAT ('LATITUDE')
123 CALL GRUT (16,104,-113,0)
124 CALL GRUT (17,114,-25,0)
125 CALL GPRG (193,15,0,1,4)
126 CALL GRUT (23,90,100,0)
127 CALL GPRG (193,24,0,1,4)
128 CALL GRUT (29,90,100,0)
129 CALL GRUT (30,104,-100,0)
130 CALL GRUT (31,114,-25,0)
131 CALL GPRG (193,32,0,1,9)
132 WRITE (15,100)
133 120 FORMAT ('LATITUDE')
134 CALL GRUT (48,104,-126,0)
135 CALL GRUT (49,114,-25,0)
136 CALL GPRG (193,44,0,1,4)
137 CALL GRUT (49,90,100,0)
138 CALL GPRG (193,50,0,1,4)
139 CALL GRUT (55,90,100,0)
140
141 C SET UP PATH ENTITY IN 300
142 CALL GPRG (300,0,0)
143 CALL GPRG (300)
144 LIDDE(2)=1
145 CALL LIDDETH
146 LIDDE(2)=0
147
148 C FIGURE TO CIRCUMPIRE LAUNCH NODE IN ENTITY 310.
149 CALL GPRG (310,0,0)
150 CALL GPRG (310)
151 CALL GPRG (310)
152 CALL GRUT (6,73,20,-15)
153 CALL GRUT (7,53,-50,0)
154 CALL GRUT (8,73,15,20)
155 CALL GRUT (9,53,30,-50)
156 CALL GRUT (10,73,-20,0)
157 CALL GRUT (11,53,20,50)
158
159 C FIGURE TO CIRCUMPIRE TARGET NODE IN ENTITY 311.
160 CALL GPRG (311,0,0)
161 CALL GPRG (311)
162 CALL GPRG (311)
163 CALL GRUT (6,73,20,0)
164 CALL GRUT (7,53,-50,0)
165 CALL GRUT (8,73,15,20)
166 CALL GRUT (9,53,30,-50)
167 CALL GRUT (10,73,-20,0)
168 CALL GRUT (11,53,20,50)
169
170 C FIGURE TO CIRCUMPIRE RECOVERY NODE IN ENTITY 312.
171 CALL GPRG (312,0,0)
172 CALL GPRG (312)
173 CALL GPRG (312)
174 CALL GRUT (6,73,20,-15)

```

```

175      CALL GPUT(7,53,0,0)
176      CALL GPUT(2,53,-40,0)
177      CALL GPUT(2,53, 30,-15)
178      CALL GPUT(10,53,30,-15)
179      CALL GPUT(11,53,40,0)
180
181      C SET UP RADIC SIGNIFICANT POINTS IN 370 - 379
182      DO 375 I=1,10
183          CALL GPUT(I,53,0,0)
184      END DO
185
186      CALL GPUT(1,53,0,0)
187      CALL GPUT(1,53,0,1,0)
188      CALL GPUT(1,53,0,1,0)
189      CALL GPUT(1,53,0,1,0)
190      CALL GPUT(1,53,0,1,0)
191
192      375      CONTINUE
193
194      C SET UP RADIC SIGNIFICANT POINTS IN 380 - 389
195      DO 385 I=1,10
196          CALL GPUT(I,53,0,0)
197      END DO
198
199      CALL GPUT(1,53,0,0)
200      CALL GPUT(1,53,0,1,0)
201      CALL GPUT(1,53,0,1,0)
202      CALL GPUT(1,53,0,1,0)
203      CALL GPUT(1,53,0,1,0)
204
205      385      CONTINUE
206
207      CALL GPUT(1,53,0,0)
208      CALL GPUT(1,53,0,0)
209      CALL GPUT(1,53,0,0)
210      CALL GPUT(1,53,0,0)
211
212      389      CONTINUE

```



```

1
2 SUBROUTINE STAGE
3 C THIS SUBROUTINE PROMPTS THE USER TO SELECT WHICH VERSION ON
4 C DISK THE SOURCE FILE WILL BE STORED. IT CAN'T GO INTO SECTION 1
5 C SINCE THIS IS THE FIRST ONLY. IT ALSO CAN'T BE GREATER THAN
6 C THE LAST VERSION NUMBER UP TO A MAXIMUM OF 6. IT CAN BE
7 C STORED ON TOP OF OTHER VERSIONS. THE NUMBER PAD IS USED TO
8 C IDENTIFY THE VERSION NUMBER.
9
10 CONTINUE /MESSAGE / INFO(100)
11 CONTINUE /MESSAGE / NET (5000)
12 CONTINUE /ROUTE / INFO(65)
13
14 C PUT PATH FROM INTO THE MASTER DATA FILE
15 IFIP = INFO(5)-1
16 DO 10 J=1,65
17     INFO(IPTR+J) = IPATH(J)
18 10 CONTINUE
19 MAP = INFO(103)
20 CALL MESSAGE(10)
21
22 IEINT = 25000-4
23 INAX = INFO(103)+1,6)
24 CALL GEOT (191)
25 CALL GEOT (33,INFO,2)
26 CALL GEOT(6)
27 CALL GEOT (110+MAP)
28
29 C GET VERSION NUMBER AND WRITE TO DISK. IF VERSION LIMITED
30 C RETURN TO 1000. DO NOTHING ON DISK WILL TAKE PLACE.
31 DO 20 J=1,65
32     INFO(IPTR+J) = IPATH(J)
33 20 CALL GEOT(191)
34
35 C RESET LAST VER. IN THE ADMINISTRATOR (ADMIN CONTROL 1000)
36 INFO(103) = 0
37 INFO(103) = INFO(103)
38 IF (INFO(103) .EQ. 0) (INFO(103)+1) = INAX
39
40 C SET OFF LOG CREDITVOLT TABLE
41 IEINT = 142
42 DO 35 J=1,11
43     CALL GEOT (IEINT+J)
44 35 CONTINUE
45
46 C TIME TO LEAVE
47 40 CALL GEOT (110+MAP)
48 CALL GEOT(191)
49 CALL GEOT(191)
50 RETURN
51 END
0 ERRORS COMPILATION COMPLETE

```

-145-

1931

-147-

NAME COUNT VALUE PERCENT

1 CONTAINER 1000
2 CONTAINER 1000
3 CONTAINER 1000
4 CONTAINER 1000
5 CONTAINER 1000
6 CONTAINER 1000

7 CONTAINER 1000
8 CONTAINER 1000

9 CONTAINER 1000

DISTRIBUTION LIST

Director, Engineering Psychology
Programs (Code 455)
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (5 cys)

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 cys)

CDR Paul Chatelier
Office of the Deputy Under Secretary
of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Dr. Stuart Starr
Office of the Under Secretary of Defense
(C31)
Pentagon, Room 3C200
Washington, D.C. 20301

Dr. Craig Fields
Director, Cybernetics Technology Office
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Office of the Chief of Naval
Operations, OP987H
Personnel Logistics Plans
Washington, D.C. 20350

Dr. A.L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Commanding Officer
ONR Eastern/Central Regional Office
ATTN: Dr. J. Lester
Bldg. 114, Section D
666 Summer Street
Boston, MA 02210

Commanding Officer
ONR Branch Office
ATTN: Dr. Charles Davis
536 South Clark Street
Chicago, IL 60605

Commanding Officer
ONR Western Regional Office
ATTN: Dr. E. Gloye
1030 East Green Street
Pasadena, CA 91106

Commanding Officer
ONR Western Regional Office
ATTN: Mr. R. Lawson
1030 East Green Street
Pasadena, CA 91106

Tactical Development & Evaluation
Support
Code 230
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Operations Research Program
Code 434
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Statistics and Probability Program
Code 436
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Information Systems Program
Code 437
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

CDR Kent Hull
Code 430B
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

CDR P.M. Curran
Code 270
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Dr. James McGrath
Code 311
Navy Personnel Research and
Development Center
San Diego, CA 92152

Management Support Department
Code 210
Navy Personnel Research and
Development Center
San Diego, CA 92152

Naval Electronics Systems Command
Human Factors Engineering Branch
Code 4701
Washington, D.C. 20360

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375 (6 cys)

Commander, Naval Electronics
Systems Command
Command and Control Division
Code 530
Washington, D.C. 20360

Dr. John Silva
Head, Human Factors Division
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Jesse Orlansky
Institute for Defense Analyses
400 Army-Navy Drive
Arlington, VA 22202

Human Factors Department
Code N215
Naval Training Equipment Center
Orlando, FL 32813

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 92940

Dr. Joseph Zeidner
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. H.W. Sinaiko
Smithsonian Institution
801 N. Pitt Street
Alexandria, VA 22314

Office of the Chief of Naval
Operations OP942
Pentagon
Washington, D.C. 20350

Office of the Chief of Naval
Operations OP987C
R&D Plans Division
Washington, D.C. 20350

Commander
Naval Electronics Systems Command
C3 Project Office
PME 10801
Washington, D.C. 20360

Dr. Julie Hopson
Human Factors Engineering Division
Code 604
Naval Air Development Center
Warminster, PA 18974

M.L. Metersky
Naval Air Development Center
Code 5424
Warminster, PA 18974

Dr. Edgar M. Johnson, Director
Organizations & Systems Research
Laboratory
U.S. Army Research Laboratory
5001 Eisenhower Avenue
Alexandria, VA 22333

Mr. Victor Monteleon
Naval Ocean Systems Center
San Diego, CA 91252

Commander, Naval Electronics
Systems Command
ELEX-03
Washington, D.C. 20360

CDR Richard Schlaff
NIPSSA
Hoffman Bldg. #1
2461 Eisenhower Avenue
Alexandria, VA 22331

Dr. Chantee Lewis
Management Department
Naval War College
Newport, RI 02840

Dr. John Shore
Naval Research Laboratory
Code 5403
Communications Sciences Division
Washington, D.C. 20375

Dr. Meredith Crawford
American Psychological Association
Office of Educational Affairs
1200 17th Street, N.W.
Washington, D.C. 20036

Dr. William Dejka
ACCAT
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Merlin Malehorn
Office of the Chief of Naval
Operations (Op 102)
Washington, D.C. 20350

Dr. S.D. Epstein
Analytics
2500 Maryland Road
Willow Grove, PA 19090

Dr. Amos Freedy
Perceptronics, Inc.
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. G. Hurst
University of Pennsylvania
Wharton School
Philadelphia, PA 19174

Dr. Miley Merkhofer
Stanford Research Institute
Decision Analysis Group
Menlo Park, CA 94025

Mr. George Pugh
Decision Science Applications, Inc.
1500 Wilson Boulevard
Arlington, VA 22209

Dr. Arthur Siegel
Applied Psychological Services
Science Center
404 E. Lancaster Street
Wayne, PA 19087

LCDR J.A. Sears
Department of MIS
College of Business Administration
University of Arizona
Tucson, AZ 85721

I.R. Mirman
Asst for Special Projects
HQ AFSC-DL
Andrews AFB, MD 20334

Mr. Joseph Wohl
MITRE Corporation
Box 208
Bedford, MA 01730

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Technology
Establishment
Teddington, Middlesex TW11 0LN
ENGLAND

Mr. Tim Gilbert
MITRE Corporation
1820 Dolly Madison Blvd.
McLean, VA 22102

Mr. Leslie Innes
Defense & Civil Institute of
Environmental Medicine
P.O. Box 2000
Downsview, Ontario M3M 3B9
Canada

Dr. Rex Brown
Decision Science Consortium
Suite 421
7700 Leesburg Pike
Falls Church, VA 22043

Dr. A.C. Miller, III
Applied Decision Analysis
3000 Sand Hill Road
Menlo Park, CA 94025

DATE
FILMED
-8