NASA CONTRACTOR REPORT 166293

Real-Time Simulation of an Airborne Radar for Overwater Approaches

J. Karmarkar
D. Clark

Systems Control Technology, Inc.

CONTRACT NAS2-10479
March 1982

NASA

NASA CONTRACTOR REPORT 166293

Real-Time Simulation of an Airborne Radar for Overwater Approaches

J. Karmarkar
D. Clark

Systems Control Technology, Inc.
1801 Page Mill Road
Palo Alto, California 94306

Prepared for
Ames Research Center
under Contract NAS2-10479

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

N82-26262#

This Page Intentionally Left Blank

SUMMARY

This report documents the real-time algorithm used to generate the necessary video interface signals, for each of the selected oil rig/helicopter relative geometry scenarios, to drive an airborne color radar display. Typical display results are documented. Unique features of the real-time algorithm include (1) a circular list approach to target sorting and ray generation, (2) target shape and multiple target merging effect generation, and (3) sea clutter generation.

Gratitude is expressed to Mr. George Clary of the NASA-Ames Research Center for his efforts on the project. In particular, his technical guidance on the radar math model and his work to provide in-flight radar video data for simulation validation was much appreciated.

TABLE OF CONTENTS

I. INTRODUCTION

1.1 BACKGROUND

This report documents software developed by Systems Control Technology, Inc. (SCT) to provide a real-time simulation of an airborne radar for overwater approaches to oil rig platforms. This software was developed for NASA-Ames Research Center as part of the Rotorcraft All-Weather Operations Research Program. The simulation is being used to study advanced concepts for enhancement of airborne radar approaches (ARA) in order to reduce crew workload, improve approach tracking precision, and reduce weather minimums. ARA's are currently used for off-shore helicopter operations to and from oil rigs.

With this purpose in mind, NASA-Ames needed a weather radar simulator for use in an existing fixed-base helicopter simulator. They awarded a contract to Computer Avionics Corporation (CAC) to develop the necessary hardware interface between the Sperry/RCA Primus 500 color weather radar and a Xerox Data Systems Sigma 9 simulation computer currently used by NASA-Ames. The radar interface hardware provides for two-way communication between the Sigma 9 and the radar display. Information obtained from the radar display through the interface includes antenna position, sweep direction, range setting, mode settings (weather/map and radar/beacon/both), radar and beacon gain settings, tilt, and sector scan (120 or 60 degrees). The radar interface receives from the simulation software encoded data defining the current target (oil rig) to be displayed. The ray definition consists of a sequence of intensity levels and ranges. The range defines where the radar is to display the new intensity level.

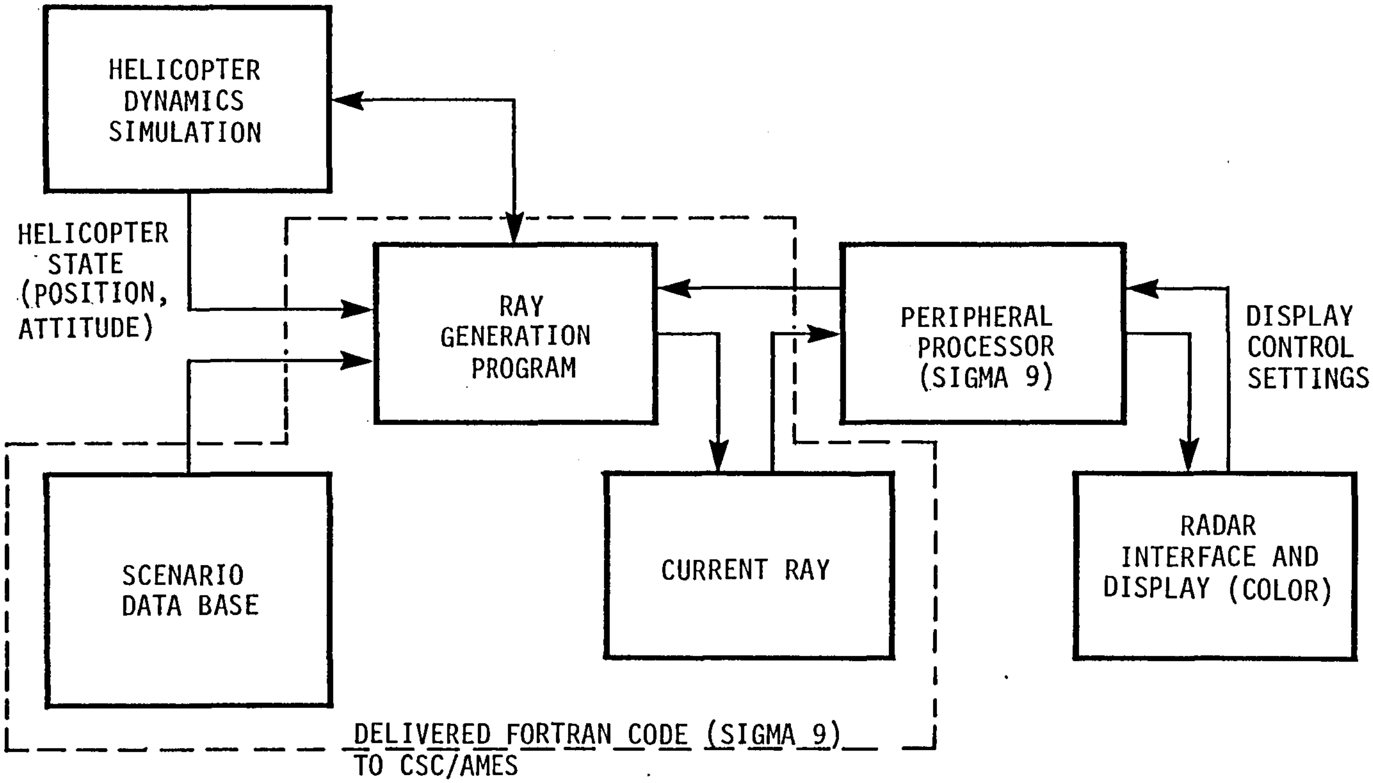


Figure 1.1 Overview of Simulation

1.2 OBJECTIVES

SCT was awarded a contract with the objective of developing the software necessary to simulate overwater radar displays. The real-time simulation requirement for the software was a constraint requiring trade-off studies to achieve a reasonable degree of simulation fidelity using simplified math models and computationally efficient algorithms. Also, due to the projected research uses for the software modular programming, the use of a high-level language (FORTRAN) was required to achieve research flexibility.

1.3 APPROACH

An overview block diagram of the radar simulation software is depicted in Figure 1.1. The software consists of three basic sections. The first section is the initialization section which allows the operator to select a block of targets and to execute this code to define all necessary parameters before actual simulation. This code is only executed once before the simulation starts. The second block of code deals primarily with target manipulation. It is responsible for maintaining the target list in a circular linked list, maintaining pointers into the target list for targets in view of the radar and deciding what targets should be updated. The third area of code is responsible for modeling visible radar targets and sea clutter by inserting intensity level changes into a linked list. At the end of each pass, this list is then used to merge all targets and output the sequence of intensity levels and ranges to the radar interface.

There are six subroutines in the radar simulation software. The first two deal only with initialization.

INIRDR - Called by the simulation executive to initialize the radar simulation. Contains the X, Y locations of all targets to be simulated. Operator must define NBRTGT to indicate the first target in the block and NUMTGT to indicate the number of targets before INIRDR is called.

INSTGT - Called by INIRDR to install a particular target into the linked list of targets. Used only for initialization.

The remaining four subroutines are used when the simulation is running, and they perform the target manipulation and display generation tasks.

CREATE - The main subroutine called by the simulation executive to generate a radar vector. It inputs data from the radar interface and determines the antenna heading. It then processes targets starting with a target known to be behind the antenna sweep and continuing until it encounters a target beyond the antenna sweep. For each target visible, it generates a set of intensity level changes and inserts them into a linked list by a call to subroutine INSERT. It also determines which target should be updated with new relative position calculations and updates it with a call to subroutine UPDATE. It also inserts any intensity level changes caused by the sea clutter model. Finally, it uses the linked list of intensity level changes to create the encoded data for the radar display.

INSERT - Insert intensity level changes into the linked list. Each call inserts an increased level change at the two ranges supplied with the call.

UPDATE - Updates a radar target position by a call to HDGDST. It then makes sure that the target is still in its correct position in the circular linked list. If not, it is moved to its correct position.

HDGDST - Calculates the relative bearing, relative tilt and distance for a target. It is called by UPDATE and also INSTGT to calculate the target position.

The constraint for real-time operation requires the radar simulation software to execute in 13 milliseconds or less for each radar ray generated. It was desired to keep the simulation cycle time under 45 milliseconds; the helicopter simulation with visual system and instrument panel outputs requires 32 milliseconds per cycle, leaving up to 13 milliseconds of cycle time for the radar software. Using the 45 millisecond cycle time, the radr antenna moves 1-1/4 to 1-1/2 degrees per cycle. Therefore, each software-generted ray is repeated until a new one is sent, and the software resolution is 1-1/4 to 1-1/2 degrees. This has not limited the display realism since the weather radar resolution is 6 to 8 degrees beamwidth. The information used to model the radar returns from point targets, beacons, and the sea was obtained from George Clary of NASA-Ames Research Center. Also, information regarding modeling of the threshold levels for the Sperry/RCA Primus 500 color radar and validation of the actual display were obtained with his assistance. Information regarding the radar theory and general description of the RCA Primus 500 radar is being documented in a NASA Technical Memo, "Simulation of a Weather Radar Display for Overwater Airborne Radar Approaches."

The rest of this report is organized as follows. Chapter II reviews the key problems and issues of the radar simulation and describes the selected approach. Chapter III documents the high-level flow of the radar simulation. Chapter IV shows the radar display output from the simulation for various radar parameters and attempts to show typical results from varying one parameter at a time. Chapter V presents concluding remarks. Appendix A contains a complete source listing of the simulation software as well as documentation for each subroutine describing key variables and common areas.

This Page Intentionally Left Blank

II. PROBLEM DISCUSSION AND APPROACH

This section discusses the key problems and issues and then proceeds to describe the specific approach used in resolving each of them within the scope of this effort.

2.1 OVERVIEW OF KEY PROBLEMS AND ISSUES

Several major problems had to be resolved in order to meet the requirements of the real-time radar simulation. These included the following.

(1) Devising a method of maintaining the lists of targets in sorted order so as to minimize the number of targets processed per pass. This was deemed necessary because of the real-time requirements.

(2) Determining a method to update a target's position in a timely manner in order to give an accurate position and to prevent the target from being updated while it is being displayed. This will prevent a target from being split.

(3) Determining a time-efficient method for modeling the target shape. This method should accurately account for known radar theory. It must also reflect changes in the target range, range setting, radar mode, gain settings, beacon targets, and antenna tilt.

(4) Devising a method to model sea clutter. Since little is known about the effects of sea clutter, an experimental model should be developed which will be time-efficient and will model some of the known or anticipated effects of sea clutter. It should take into account both the velocity and direction of the wind, thereby defining the "sea state."

(5) Implementing a simple method to merge multiple targets and/or sea clutter. Since each target will be independently processed and the output of this processing will consist of

intensity changes at calculated ranges, then a sorted list of these intensity changes will need to be maintained. Before output to the actual hardware, those targets which overlap in range will have to be merged.

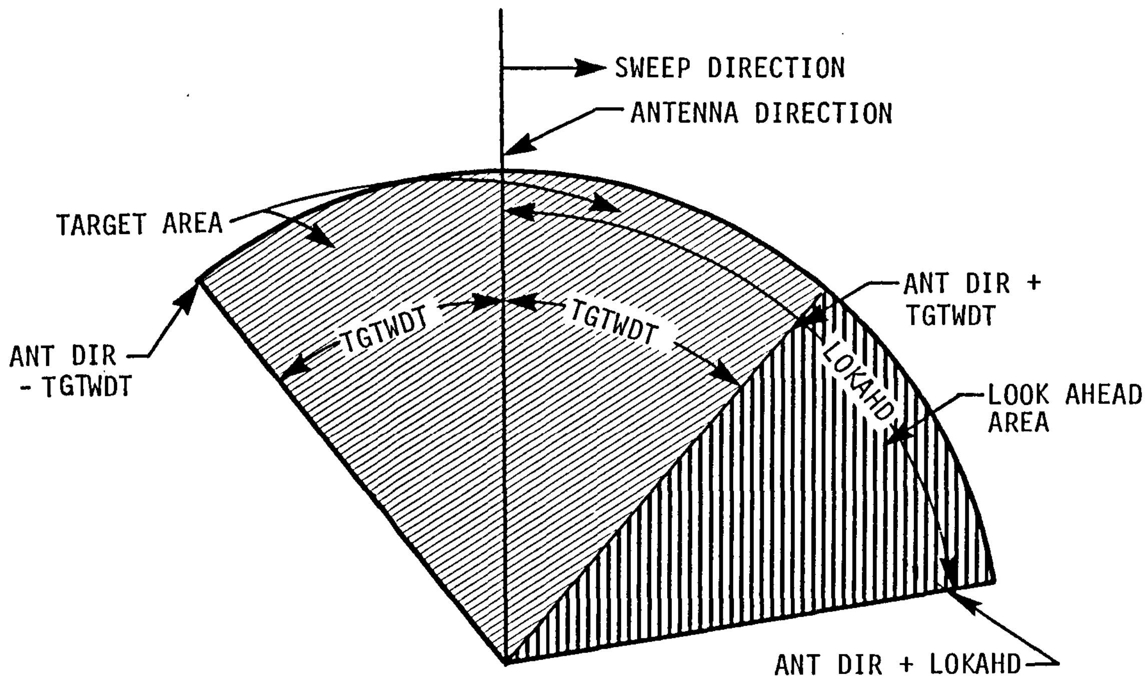
2.2 TARGET TRACKING

To satisfy requirements (1) and (2), it was decided to maintain the targets in a circular linked list in order of magnetic bearing from the helicopter's current position. Since the antenna sweeps in both the clockwise and counterclockwise direction, two pointers for each target were maintained. One is used for the clockwise direction and the other for the counterclockwise direction. Magnetic bearing was used as opposed to heading relative to the helicopter since magnetic bearing changes slowly with the helicopter's position while relative heading can change quickly as the helicopter changes heading. These pointers are stored in the TGTPTR array.

In order to process only the targets pointed to by the antenna, two angular areas were defined. Both areas are defined relative to the current antenna direction. The first is the target area which defines the area in which targets may be visible and should be processed. It is defined by the TGTWDT parameter which specifies the maximum difference in direction allowed between the target and the antenna direction. It is currently about 12 degrees, and therefore the target area extends from the antenna heading minus 12 degrees to the antenna heading plus 12 degrees. Once a target is in the target area, it can no longer have its position updated. This prevents the target from splitting since its position remains constant throughout the sweep past the target direction.

The second angular area is the look-ahead area. It is defined in order to assure that the target's positions are up to date before entering the target area. This area extends from the forward edge of the target area about 6 degrees. This angle is defined by the LOKAHD parameter, but the actual size of the look-ahead area is determined by LOKAHD - TGTWDT. This area is always in front of the antenna sweep direction. When the antenna changes direction, the look-ahead area changes to the other side of the target area. When a target enters the look-ahead area, it becomes a primary candidate for updating its position. The proper size of the look-ahead area is determined by the number of targets that may enter the look-ahead area at the same time. Once a target enters the look-ahead area, it has about five passes to update its position before it enters the target area and becomes "frozen." If six targets came into the look-ahead area at once, then only five would have a chance to be updated before they entered the target area. The current value of LOKAHD seems appropriate since targets never enter the look-ahead area at that rate and the target's position changes very little in .6 seconds; this is the time from when the target gets updated to the time the antenna sweeps by the target. A diagram of the target and look-ahead areas is contained in Figure 2.1.

Targets have their position updated by two different priorities. Only one target is updated on each pass of the algorithm. First priority goes to targets that have entered into the look-ahead area. If no targets have entered the look-ahead area, a target is picked that is outside both the target and look-ahead areas. These secondary priority targets are selected by traversing clockwise through the target list. When the target area is encountered, it is skipped until a target is found past the target area. The actual updating is done by subroutine UPDATE. This routine updates the true heading and distance from the helicopter to the target. The distance computed is actually the slant range computed from the differences in X position, Y position, and altitude between the helicopter and the target. It also updates the tilt angle to the target and the log 10 value of the target's distance. If the target heading has changed so that



TGTWDT = .2 RADIANS = 11.46 DEGREES
LOKAHD = .3 RADIANS = 17.19 DEGREES

Figure 2.1 Target and Look-Ahead Areas

it no longer lies between its clockwise and counterclockwise neighbor, the target pointers are changed by removing the updated target, finding its new position in the linked list, and then inserting it.

2.3 TARGET MODELING AND SIMULATION

Targets are simulated only if they are in the current target area. Then target modeling starts by computing the signal strength returned from the current target. The equations for calculating signal strength are derived from basic radar theory. The NASA technical memo, which is currently in work, will contain a detailed discussion of the radar theory involved in the simulation as well as discussions on the radar-specific values used in the radar equation. This memo will define the returned signal strength equatons for both normal and beacon targets, and the threshold levels for all three intensity levels.

The basic radar equation for signal strength returned from a non-beacon oil rig target is:

Return Power (dB) = Transmit Power - 30 * log (4*Pi)
- 40 * log (Range) + 2 * Antenna Gain (in dB)
+ 20 * log (Lambda) + 10 * log (Sigma)
- TR Loss

Several of these values are constants for purposes of this simulation. These include

Transmit Power = 7000 watts = 68.45 dbm
Lambda = .0319 meters = radar wavelength
Sigma = 1000 square meters = target reflectivity
TR Loss = 3.6 dB = SH-3 transmit/receive losses

The antenna gain is computed from the maximum antenna gain minus antenna pointing error loss. The maximum antenna gain depends on the antenna size and can be computed by 6.4 + 200 log (antenna diameter in inches). For an 18" antenna, this gives a value of

31.5 db. The pointing error is calculated from the direction error and the tilt error. The gain loss is then calculated using the angular antenna error as an index into a table of antenna losses. The range is obtained from the target position information which was computed during the last update of the target. The range value is in nautical miles while the radar equation expects it in meters. The log of the conversion factor is therefore added in as a constant. The final radar equation for return power after all constants have been removed is

Return Power (dB) = - 95.13 - 40* log (Range in NM)
+ 2.00* antenna gain - TR loss

The returned signal strength is then compared with the threshold levels. These define the minimum level for which the target is visible. The actual threshold levels depend on several factors which include pulse width, radar mode (map or weather), target distance and maximum antenna gain. Figures 2.2 and 2.3 show the threshold levels for weather and map mode, respectively, as obtained from Sperry/RCA avionics for the RCA Primus 500 radar. The MDS (minimum detectable signal) depends on the pulse width. For weather mode or map mode with a range of 50 miles or greater a long pulse width (2.35 s) is used and the MDS is lower. This is also used if the radar is in beacon or both modes. For map mode less than 50 miles in range, the shorter pulse width (0.6 s) is used; this gives a higher MDS.

The 28 dB maximum antenna gain is the correct value for the 12" antenna. If an 18" antenna is installed, the maximum antenna gain increases to 31.5 dB. This causes the returned signal strength to be increased by 7 dB. To correct for this, the threshold levels for the simulated radar are also increased whereas an actual hardware adjustment to the radar receiver is required on an aircraft.

The target is modeled independently for each color intensity. The threshold level for the color intensity being modeled is subtracted from the returned signal strength. If this

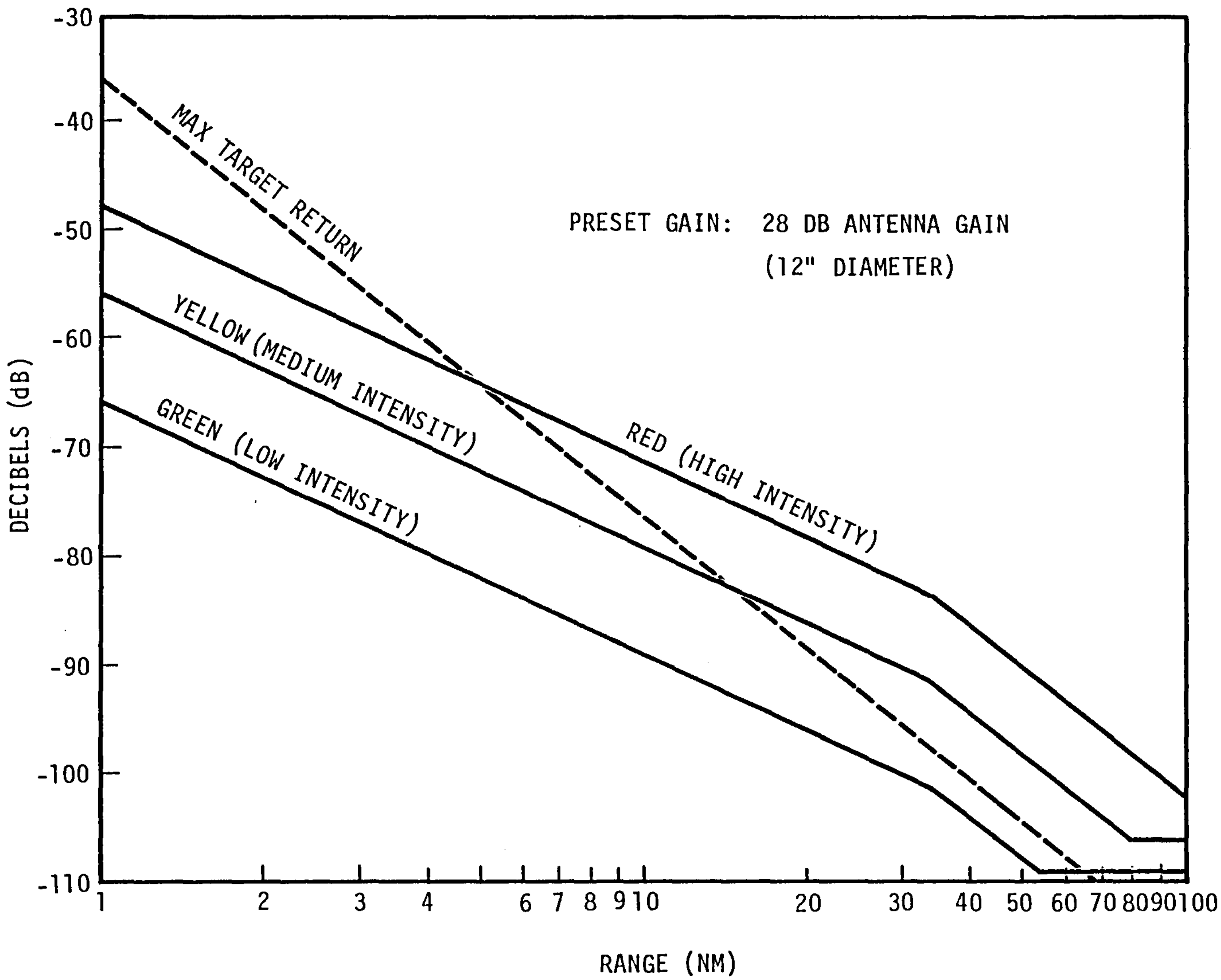


Figure 2.2 Weather Mode

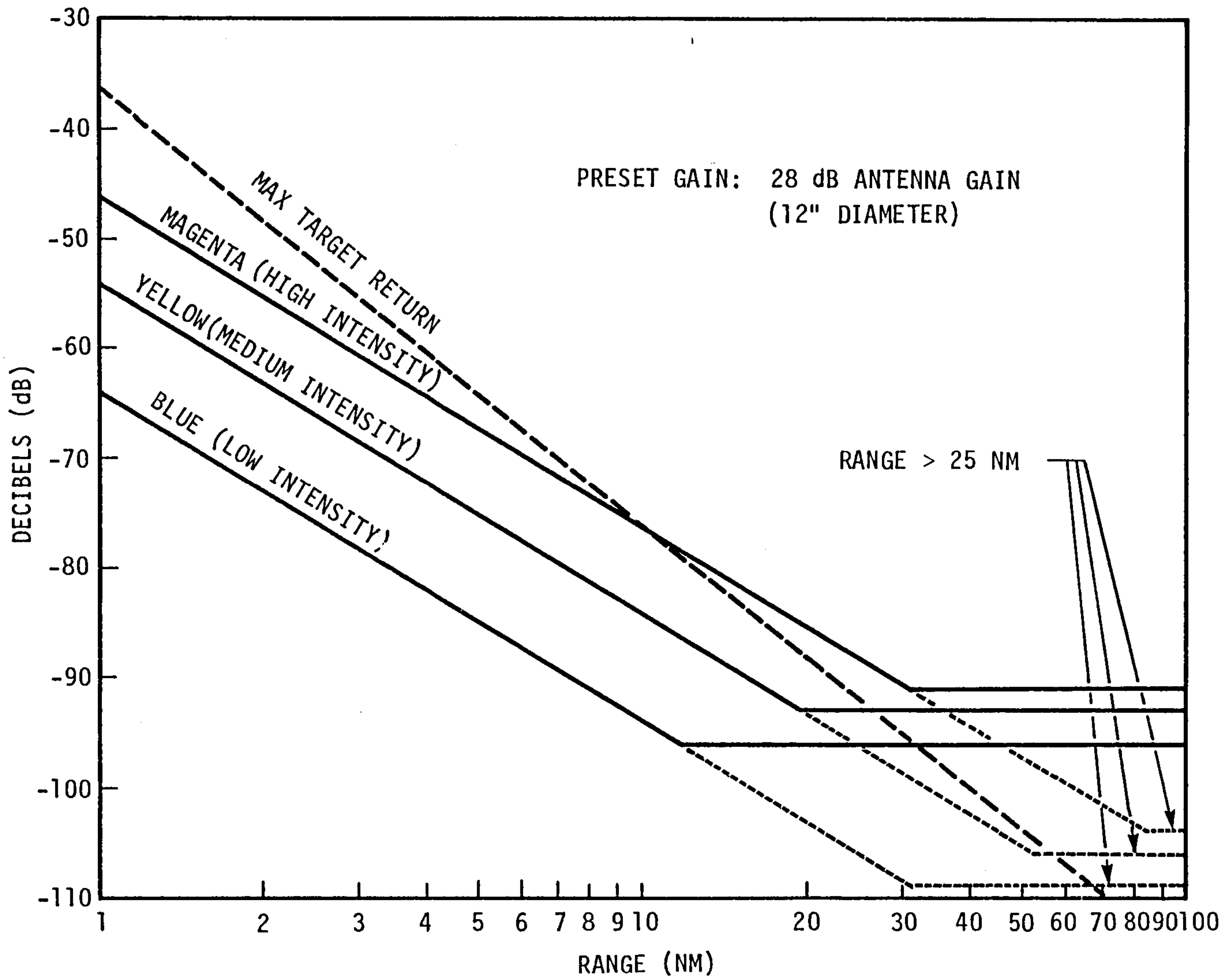


Figure 2.3 Map Mode

yields a positive dB level, then the target is visible at that intensity. The length of the target for this intensity level is obtained from parameters defining the minimum length and a factor multipled by the dB level. The actual length is a function of the range selected, the color intensity, and the pulse width. Parameters also define the percentage of the target ahead and behind the actual target distance. Target shape and the parameters that define it are given in Figure 2.4.

Targets with beacons on them are also modeled. Return power from beacons is defined as

Return power (dB) = beacon transmit power
- 20 log (4*PI)
- 20 log (range in meters)
+ radar antenna gain
+ beacon antenna gain
+ 20 log (lambda)
- one-way loss

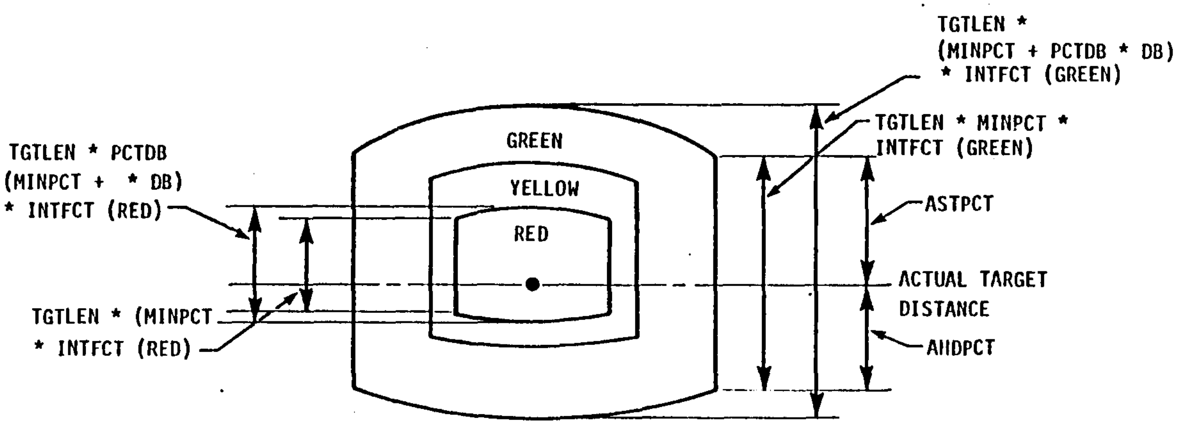
Several of these values are constants for purposes of this simulation. These include:

Beacon transmit power = 400 watts = 56.02 dBm
Radar beacon antenna gain = 5 dB
Lambda = 0.0322 meters

The range was again converted to nautical miles and the radar antenna gain is computed as before. This yields the return power as

Return power (dB) = -56.16 - 20* log (range in nm)
+ antenna gain - one-way loss

The threshold level for beacon returns is modeled as in Figure 2.5. There is no MDS defined. The dB level is obtained by subtracting the threshold level from the return power. If it is positive then the beacon target is visible and is modeled as a fixed length target. This length is the same as TGTLEN described in Figure 2.4. The beacon target appears at the maximum intensity while other non-beacon targets are prevented from



ACTUAL PARAMETERS

RANGE SETTING	TARGET LENGTH
2.5 NM	.15 NM
5.0 NM	.15 NM
10.0 NM	.30 NM
25.0 NM	.40 NM
50.0 NM	.80 NM
100.0 NM	1.60 NM

PULSE WIDTH	FACTOR
2.35 μs	2.0
0.6 μs	1.0

INTENSITY	FACTOR
GREEN (LOW)	1.0
YELLOW (MED)	0.65
RED (HIGH)	0.25

MINPCT = MINIMUM LENGTH (DB=0) = 0.7
PCTDB = ADDITIONAL LENGTH PER DB = 0.015
ASTPCT = PERCENT OF TARGET BEHIND ACTUAL DISTANCE = 0.85
AHDPCT = PERCENT OF TARGET AHEAD OF ACTUAL DISTANCE = 0.15

Figure 2.4 Target Modeling Parameters

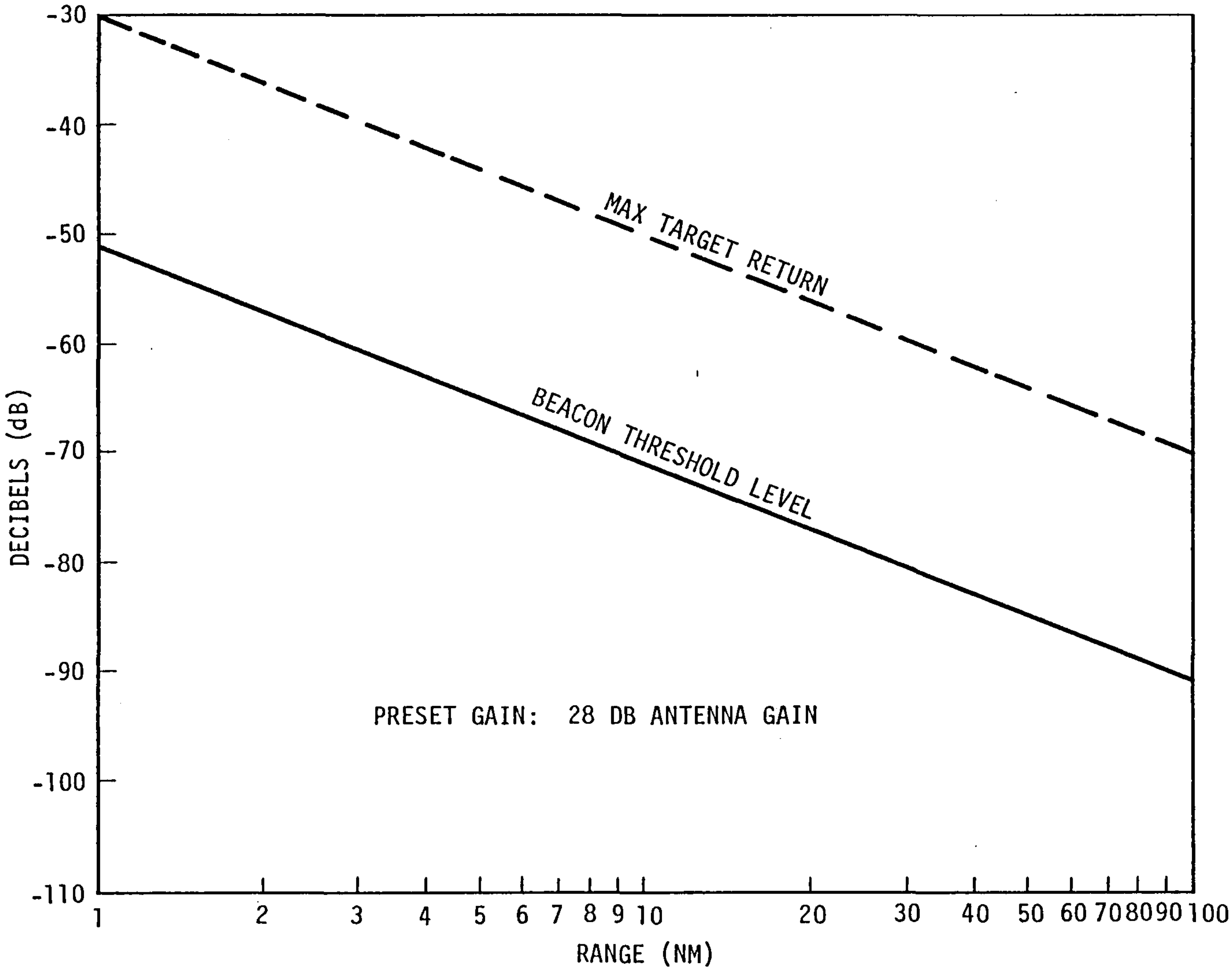


Figure 2.5 Beacon Mode

reaching maximum intensity. Because of the time delay between the beacon receiving an interrogation signal and the beam responding, the beacon target appears 0.6 miles behind the actual target. Thus, when the display mode is "both" (radar and beacon targets), the radar target with the beacon will appear at the correct distance, while the beacon target will appear 0.6 miles behind it.

2.4 INITIALIZATION

Several requirements were necessary for the initialization task. Initialization occurs only once prior to executing the simulation when subroutine INIRDR is executed. Many different target patterns were stored in the 100 loctions reserved for target descriptions. The operator must set two parameters (first target to be displayed and the total number of targets). The initialization procedure then calculates the position and bearing and then inserts it into the target pointers which maintain the clockwise and counterclockwise lists. This is repeated for the total number of targets. This procedure provides a means for the operator to define new target locations interactively and then to display them without the need for recompilation.

2.5 SEA CLUTTER MODEL

The sea clutter model is intended to model the effects of radar return from the sea. The sea clutter effect on actual radar is very complicated. There can exist very random returns from the sea with the amount of clutter ranging from almost solid with a few holes to an almost clear screen with a few solid patches. Because of the execution time constraints, several simplifying constraints were made. Once such constraint is that no holes would exist in the sea clutter. This allowed sea clutter to be defined as a single length of sea "noise."

Furthermore, it was decided that the sea clutter would only exist in the lowest intensity level. A separate pass through the algorithm would have to be made to generate a different intensity level and it was decided that the amount of actual sea clutter at a higher intensity level was not sufficient to justify more execution time.

Therefore, the sea clutter model consists of calculating a return level from the sea and comparing it with the threshold level. The threshold level is calculated as in the radar target with the range being defined as the slant range between the radar antenna and the sea. The level of return from the sea accounts for several factors, which include the following.

(1) The increasing reflectance of the sea as tilt decreases (pointing more directly at the sea);

(2) The increasing reflectance as sea state increases. The sea state was defined in terms of wind velocity for purposes of this simulation.

(3) The changes in reflectance a a function of heading relative to the sea.

Other factors affecting the sea return include radar gain, pulse width, slant range, antenna gain, and miscellaneous transmission losses.

After the sea return level is calculated, it is then compared with the threshold level. If the return level is greater than the threshold level, then an area of sea clutter is generated. This area is centered around the sea slant range. To provide more realism, a random noise factor was inserted in the calculation of sea clutter starting and ending ranges.

Further discussion on the derivation of the sea clutter model is to be included in a NASA Technical Memo to be published later this year.

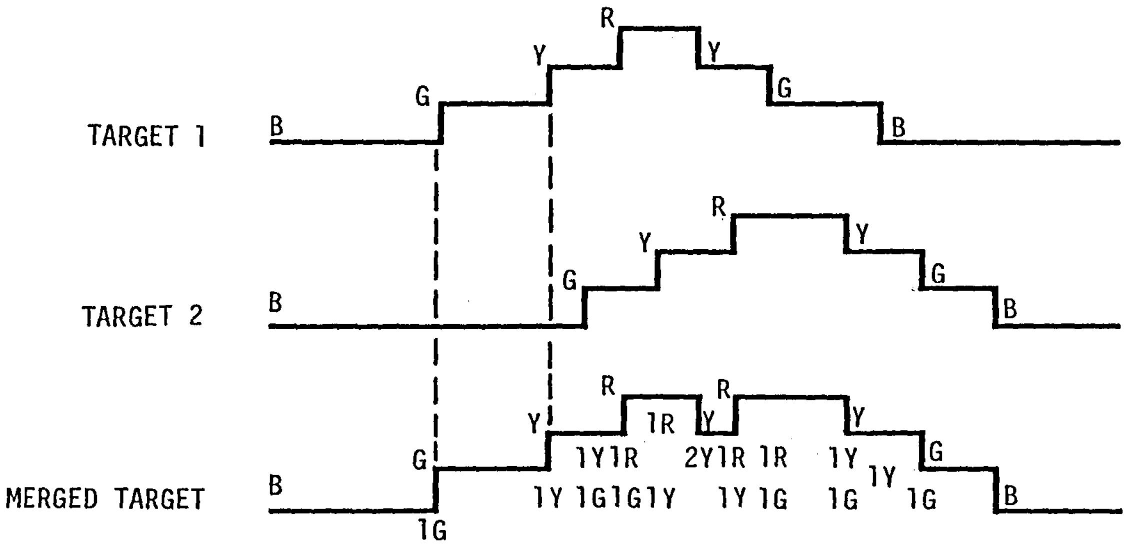
2.6 TARGET MERGING

The final problem remaining was how to merge the independently generated targets when two or more targets overlap. The solution was to maintain a range sorted list of intensity changes. The intensity change defines both the previous intensity level and the new intensity level traversed. At every range, a count of each intensity level is maintained. If the intensity level produced is the same as the current intensity, then no change is necessary and the point is eliminated. An example of the merging of two targets is in Figure 2.6.

The rules for merging intensity are as follows:

2 level 1	= level 1 (low intensity)
3-5 level 1	= level 2 (medium intensity)
6 or more level 1	= level 3 (high intensity)
1 level 2 + 3 or more level 1	= level 3
1 level 2 + 1 or more level 2	= level 3

These rules were designed to produce realistic merging at large ranges (multiple targets in close proximity) and also merging of targets at short range and high intensities.



B = BLACK; R = RED; Y = YELLOW; G = GREEN

Figure 2.6 Target Merging Effects

This Page Intentionally Left Blank

III. PROGRAM DOCUMENTATION

The methodology used in developing the real-time software was first to implement it on a system consisting of a PDP 11/34 computer driving a Megatek graphics display which, in turn, was used to simulate the RCA radar. This allowed substantial development and debugging to be performed without access to the heavily used NASA simulation computer. After a baseline version had been developed and optimized for real-time execution, the program was converted to the NASA Sigma 9 computer. After several revisions of this baseline code on both the above-mentioned computers the final software, documented below, resulted. The work on the Sigma 9 facility was supported by Computer Sciences Corporation. Consequently, on-site personnel are acquainted with the code developed at a working level and are qualified to make future revisions.

The following three Figures 3.1, 3.2 and 3.3 show the high level flow of the CREATE subroutine including target processing, modeling and updating. Further documentation including program description, listings and a list of variables is contained in Appendix A. This is the current software existing at NASA-Ames Research Center on the Sigma 9.

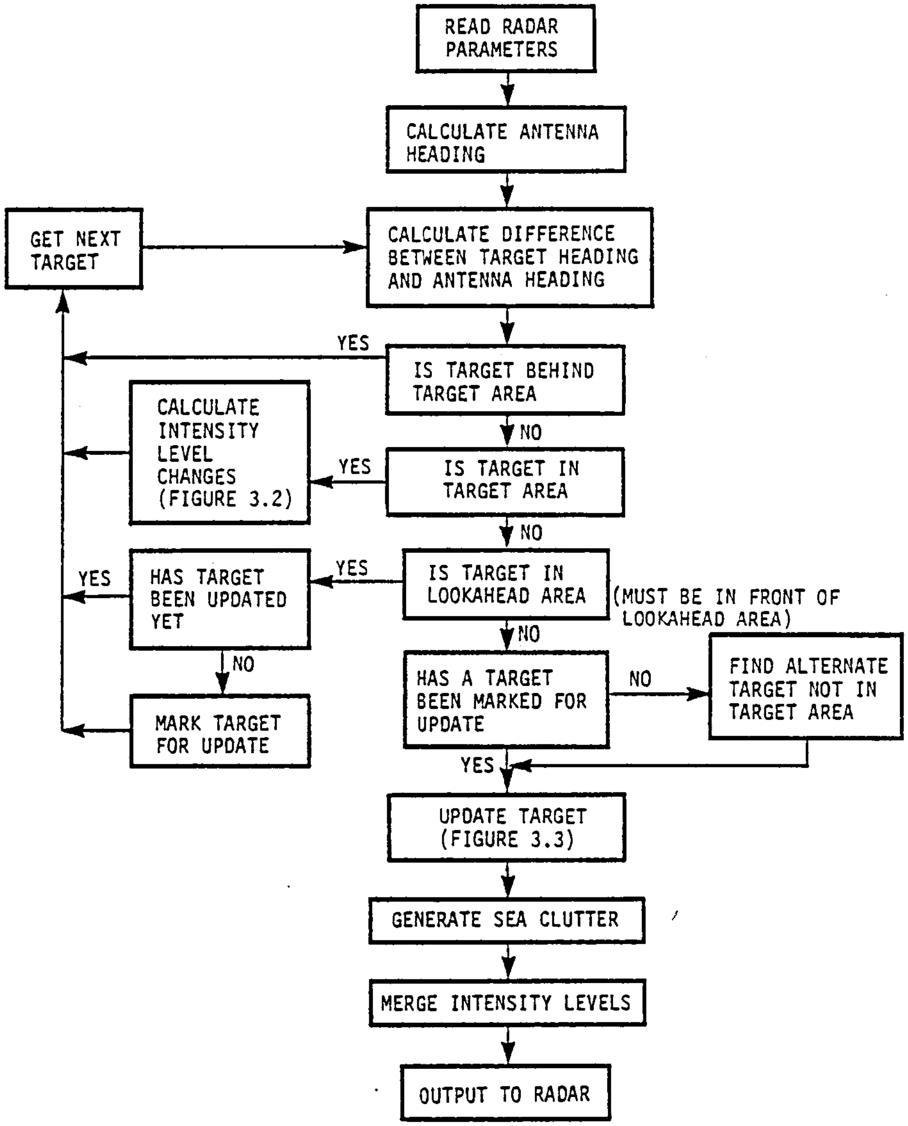


Figure 3.1 Target Processing

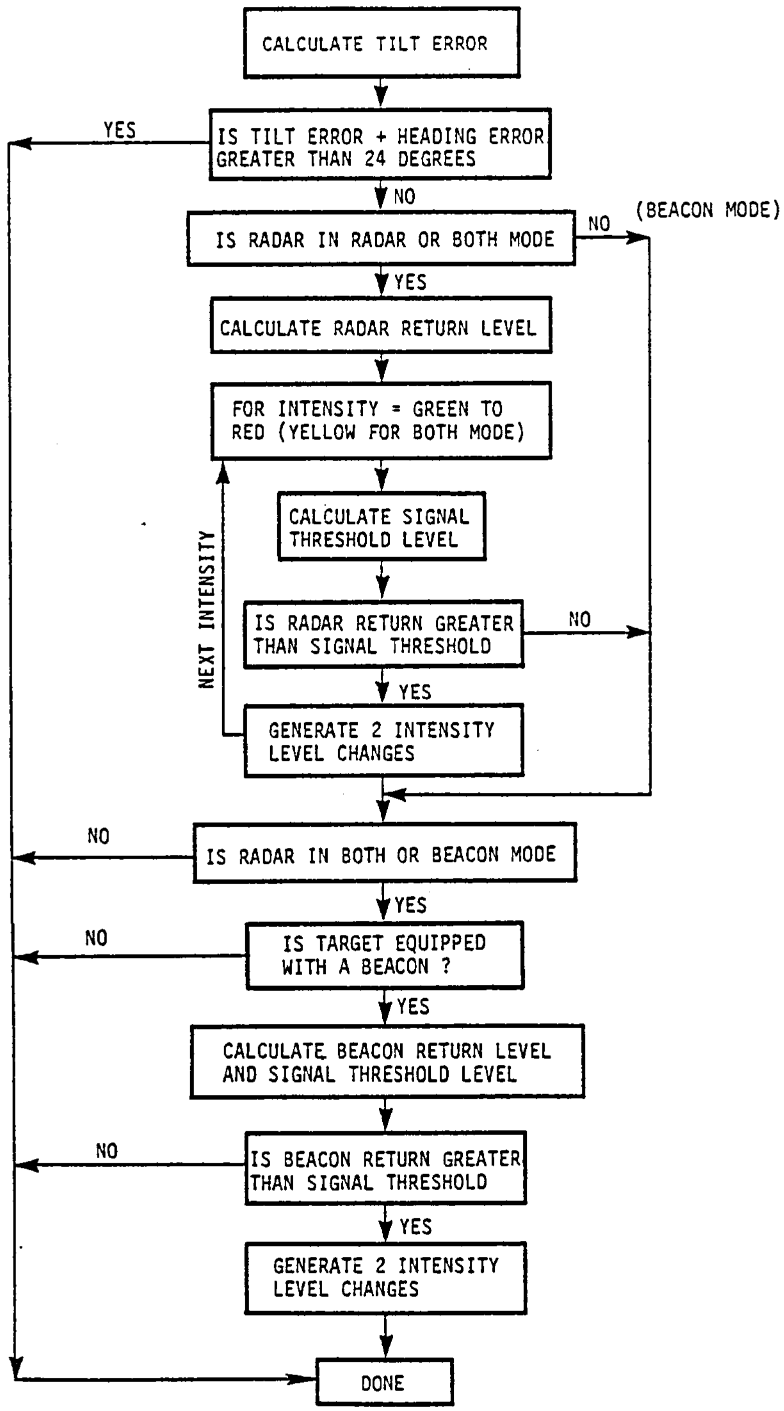


Figure 3.2 Target Modeling

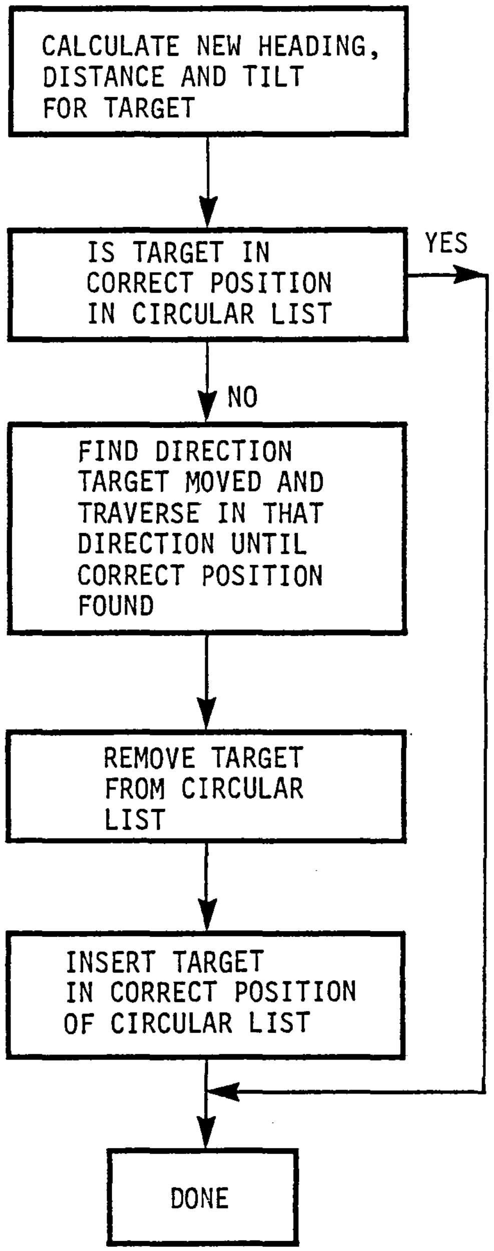


Figure 3.3 Target Update

IV. DISPLAY OUTPUT RESULTS

The objective of this effort was to produce visually realistic displays for typical overwater approaches to oil rig clusters and platforms. In order to allow project engineers to perform this activity it was necessary to (1) review currently available data on such approaches and (2) characterize the quality of the displays by providing hard copy examples of the display outputs for typical sets of key system parameters.

As the work progressed, it turned out that very few flight test data sources were available; in fact, NASA-Ames project engineers appeared to have the best available visual description of the target returns. A flight was made in one of the NASA aircraft specifically for obtaining visual display data from oil rigs, and the NASA TM in preparation will show comparisons between photos of the simulation display and airborne display.

As a mechanism to characterize the quality of the display generated, a series of runs wre made at the SCT facility and the results were drawn on a plotter. Table 4.1 summarizes the various runs. It should be noted that these runs were generated with SCT's software version of the radar simulation. The Sigma 9 listing in Appendix A has modified some of the parameters to improve target shaping and sea clutter models. Therefore, these plots are not exactly as in the Sigma 9 version although the basic characteristics of the simulation are accurately represented.

Figure 4.1 details the parameter values used for the simulation runs shown in Figures 4.2 through 4.45. Several observations should be made about the trial runs. Figures 4.2 to 4.7 show the effects of target merging as the target distance goes from 1.75 to 70 miles. Also apparent is the pulse width change as the range in map mode is changed from 25 to 50 miles. As the pulse width changes from .6 μs to 2.35 μs the target

Table 4.1

Display Characterization Cases

FIGURE NO.	COMMENTS	
4.2 - 4.7	Effects of range on MAP mode	
4.8 - 4.12	Effects of range on WEATHER mode	
4.13	Both radar and beacon	
4.14	Beacon only	
4.15 - 4.18	Effects of radar gain	
4.19 - 4.22	Effect of beacon gain	
4.23 - 4.29	Effects of tilt 0-knot wind	2000 ft
4.30 - 4.34	Effects of tilt 10-knot wind	2000 ft
4.35 - 4.39	Effects of tilt 10-knot wind	2000 ft
4.40 - 4.45	Effects of tilt 10- 30-knot wind	500 ft

FIGURE NUMBER	NORTH POS (NM)	ALTITUDE (FT)	WIND DIR. (DEGS)	WIND VEL. (KTS)	RANGE (NM)	RADAR/BEACON	TILT (DEGS)	RADAR GAIN (VOLTS)	BEACON GAIN (VOLTS)	WEATHER/ MAP
4.2	1.75	0	0	0	2.5	RADAR	0	PRESET	PRESET	MAP
4.3	3.5				5			(4.625)	(4.625)	
4.4	7				10					
4.5	17.5				25					
4.6	35				50					
4.7	70				100					
4.8	1.75	0	0	0	2.5	RADAR	0	PRESET	PRESET	WEATHER
4.9	3.5				5.0					
4.10	7				10					
4.11	17.5				25					
4.12	35				50					
4.13	3.5	0	0	0	5	BOTH	0	PRESET	PRESET	WEATHER
4.14						BEACON		PRESET		
4.15						RADAR		5.0		
4.16								4.0		
4.17								3.0		
4.18								2.0		
4.19	3.5	0	0	0	5	BOTH	0	PRESET	5.0	WEATHER
4.20									4.0	
4.21									3.0	
4.22									2.0	

Figure 4.1 Simulation Runs

FIGURE NUMBER	NORTH POS (NM)	ALTITUDE (FT)	WIND DIR. (DEGS)	WIND VEL. (KTS)	RANGE (NM)	RADAR/BEACON	TILT (DEGS)	RADAR GAIN (VOLTS)	BEACON GAIN (VOLTS)	WEATHER/ MAP
4.23	3.5	2000	0	0	5	RADAR	3	PRESET	PRESET	WEATHER
4.24							-1			
4.25							-3			
4.26							-6			
4.27							-9			
4.28							-12			
4.29							-15			
4.30	3.5	2000	0	10	10	RADAR	-1	PRESET	PRESET	WEATHER
4.31							-3			
4.32							-6			
4.33							-9			
4.34							-12			
4.35	3.5	2000	0	20	50	RADAR	-1	PRESET	PRESET	WEATHER
4.36							-3			
4.37							-6			
4.38							-9			
4.39					10		-12			
4.40	3.5	500	0	10	10	RADAR	-1	PRESET	PRESET	WEATHER
4.41							-3			
4.42							-6			
4.43				20			-3			
4.44				30						
4.45			270	20						

Figure 4.1 (Continued)

length increases. Figures 4.8 to 4.12 show the same runs in weather mode. The major difference between map and weather mode is the use of the 2.35 μs pulse width at all ranges in the weather mode. This causes the targets in weather mode to be thicker at ranges less than 50 miles than targets in map mode. Figure 4.13 shows the effect of placing the radar in both mode which displays both beacon and radar returns. All radar returns are suppressed from the highest intensity (red) while the beacon return is a constant thickness at the highest intensity. The beacon delay of .6 nautical miles places the beacon return behind the primary target. Figure 4.14 shows the beacon only display. Figures 4.15 to 4.18 shows the effects of the radar gain. As the gain voltage is lowered from 5 to 2 volts the targets shrink. This represents a change in gain from +3 to -21 dB compared to the preset gain equal to 4.625 volts. The targets disappear completely at radar gains of less than 2 volts.

Figures 4.19 to 4.22 show similar effects on the beacon target when the beacon gain is reduced from 5 volts to 2 volts. Figures 4.23 to 4.29 show the effects of tilt on target appearance. When the radar tilt deviates from the optimal the target size diminishes. Also the first effects of sea clutter is shown in Figures 4.28 and 4.29. As the tilt decreases the angular width of the sea clutter increases while the range of the clutter decreases. Figures 4.30 to 4.39 show sea clutter effects as the wind velocity increases to 10 and 20 knots with tilts ranging from -1 to -12 degrees. Maximum range of the sea clutter is with a wind velocity of 20 knots and -1 degree tilt. The sea clutter in this case extends up to 50 miles. The last set of figures are sea clutter effects at 500 feet altitude. This gives shorter ranges of the sea clutter. Figure 4.45 shows sea clutter when the wind direction is perpendicular to the flight path. This causes the waves to be parallel to the flight path and less sea clutter return. The width of the sea clutter is maximum at the edges and minimal straight ahead as opposed to the other way around with a wind direction of 360 degrees.

A few typical cases of interest were also documented by photographing the RCA color display as simulated at NASA Ames Research Center. These photos are shown in Figures 4.46 through 4.53.

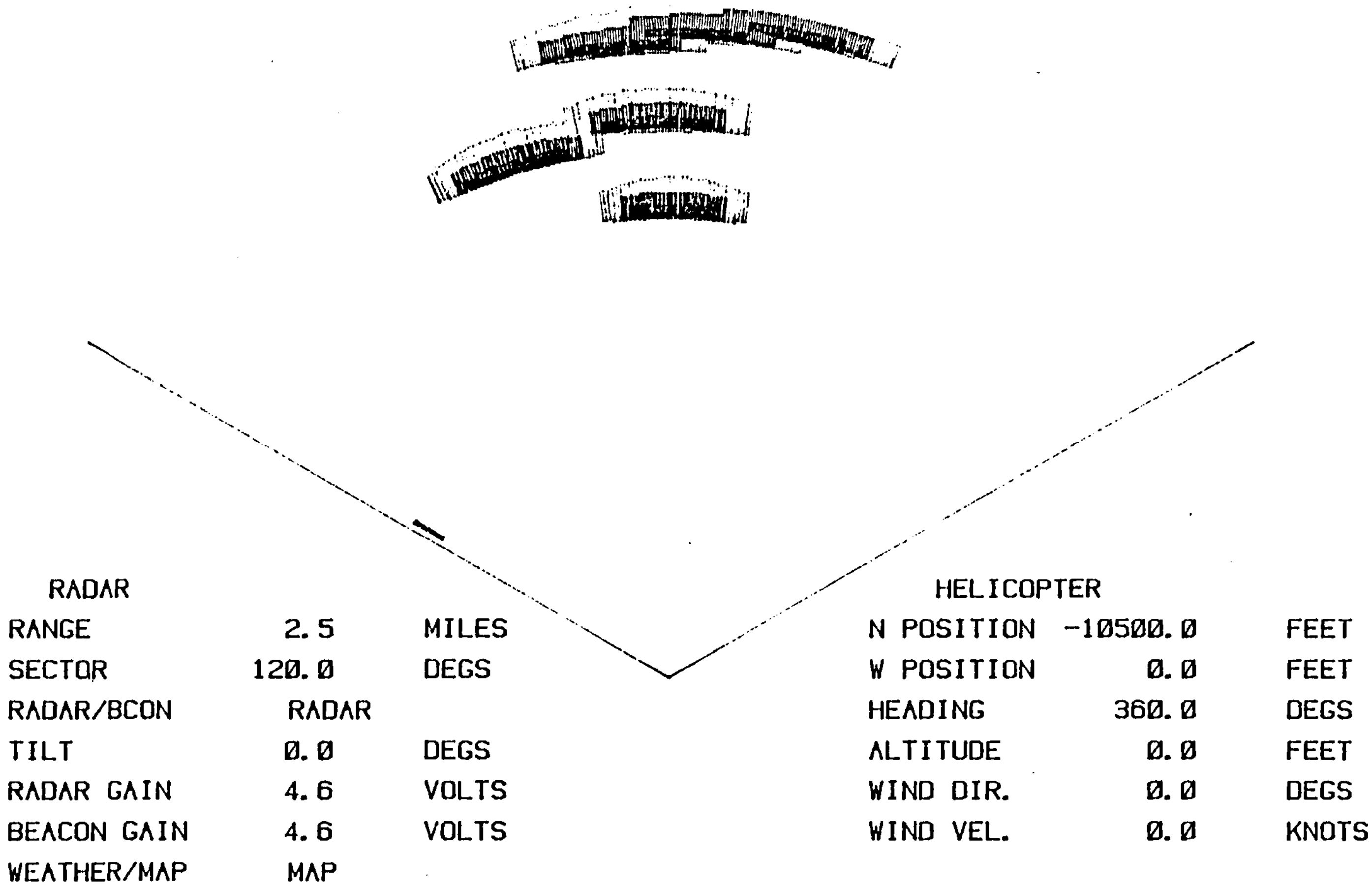

RADAR
RANGE 2.5 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT 0.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP MAP
HELICOPTER
N POSITION -10500.0 FEET
W POSITION 0.0 FEET
HEADING 360.0 DEGS
ALTITUDE 0.0 FEET
WIND DIR. 0.0 DEGS
WIND VEL. 0.0 KNOTS

Figure 4.2

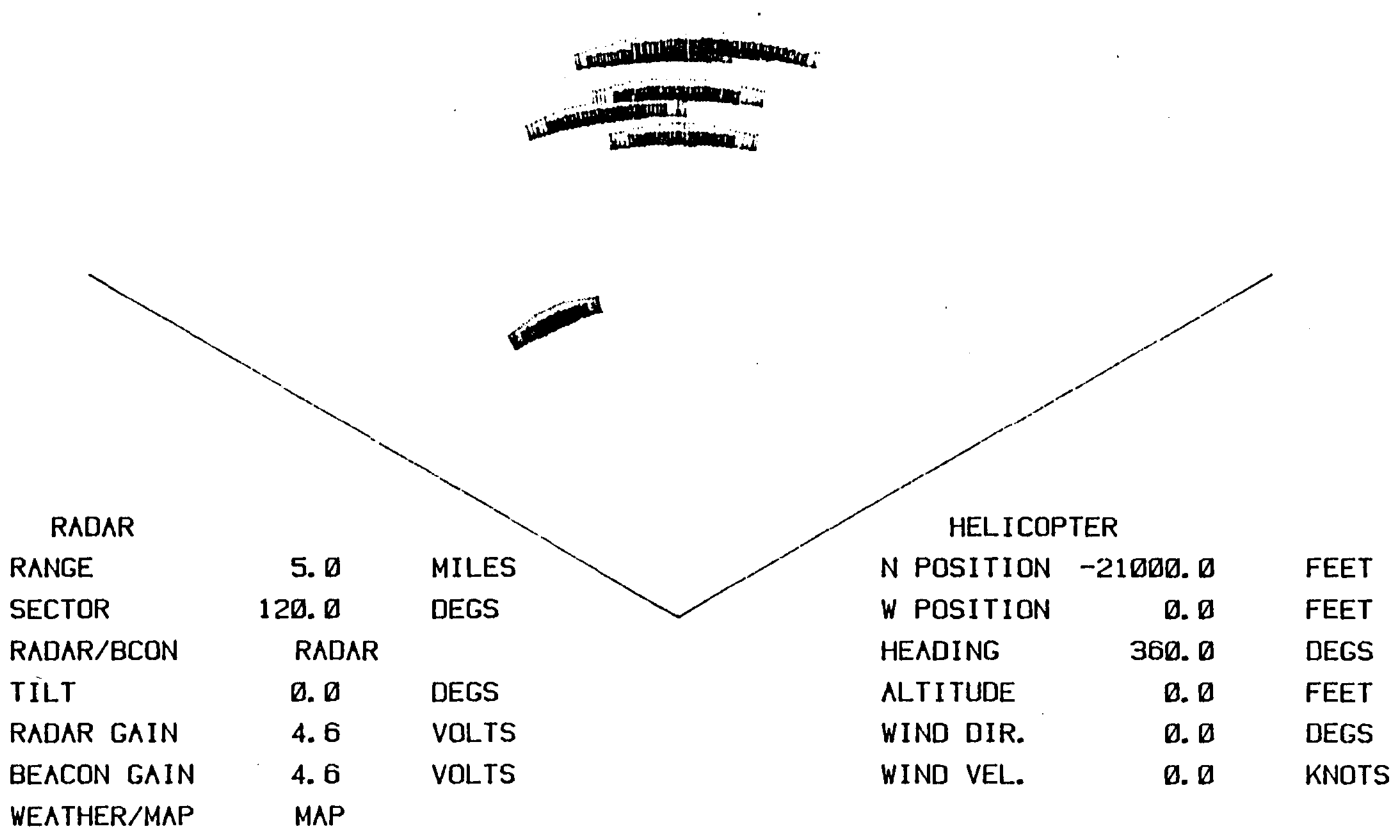
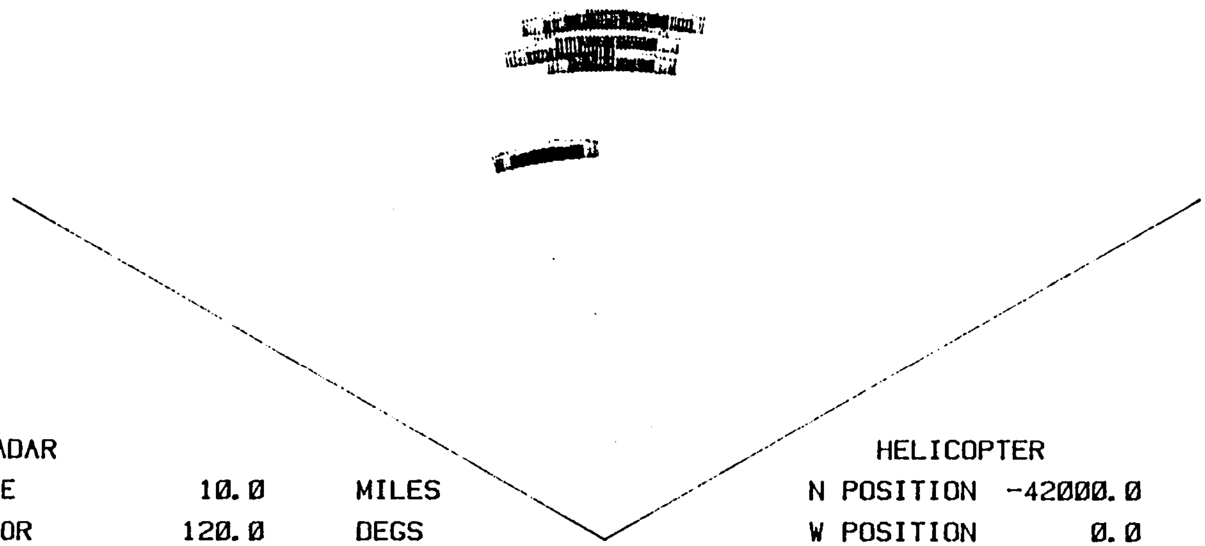

RADAR
RANGE 5.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT 0.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP MAP
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 360.0 DEGS
ALTITUDE 0.0 FEET
WIND DIR. 0.0 DEGS
WIND VEL. 0.0 KNOTS

Figure 4.3



RADAR		
RANGE	10.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	RADAR	
TILT	0.0	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	MAP	

HELICOPTER		
N POSITION	-42000.0	FEET
W POSITION	0.0	FEET
HEADING	360.0	DEGS
ALTITUDE	0.0	FEET
WIND DIR.	0.0	DEGS
WIND VEL.	0.0	KNOTS

Figure 4.4

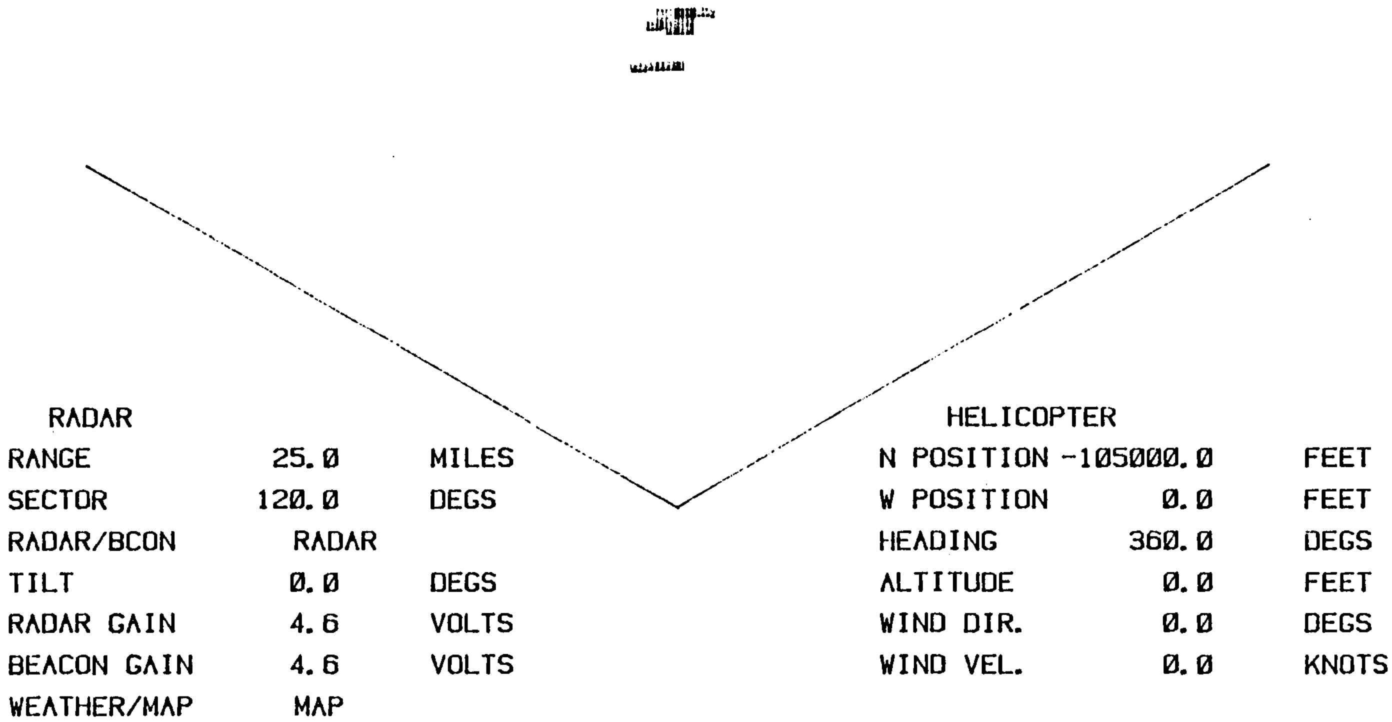

RADAR
RANGE 25.Ø MILES
SECTOR 12Ø.Ø DEGS
RADAR/BCON RADAR
TILT Ø.Ø DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP MAP
HELICOPTER
N POSITION -1Ø5ØØØ.Ø FEET
W POSITION Ø.Ø FEET
HEADING 36Ø.Ø DEGS
ALTITUDE Ø.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

Figure 4.5

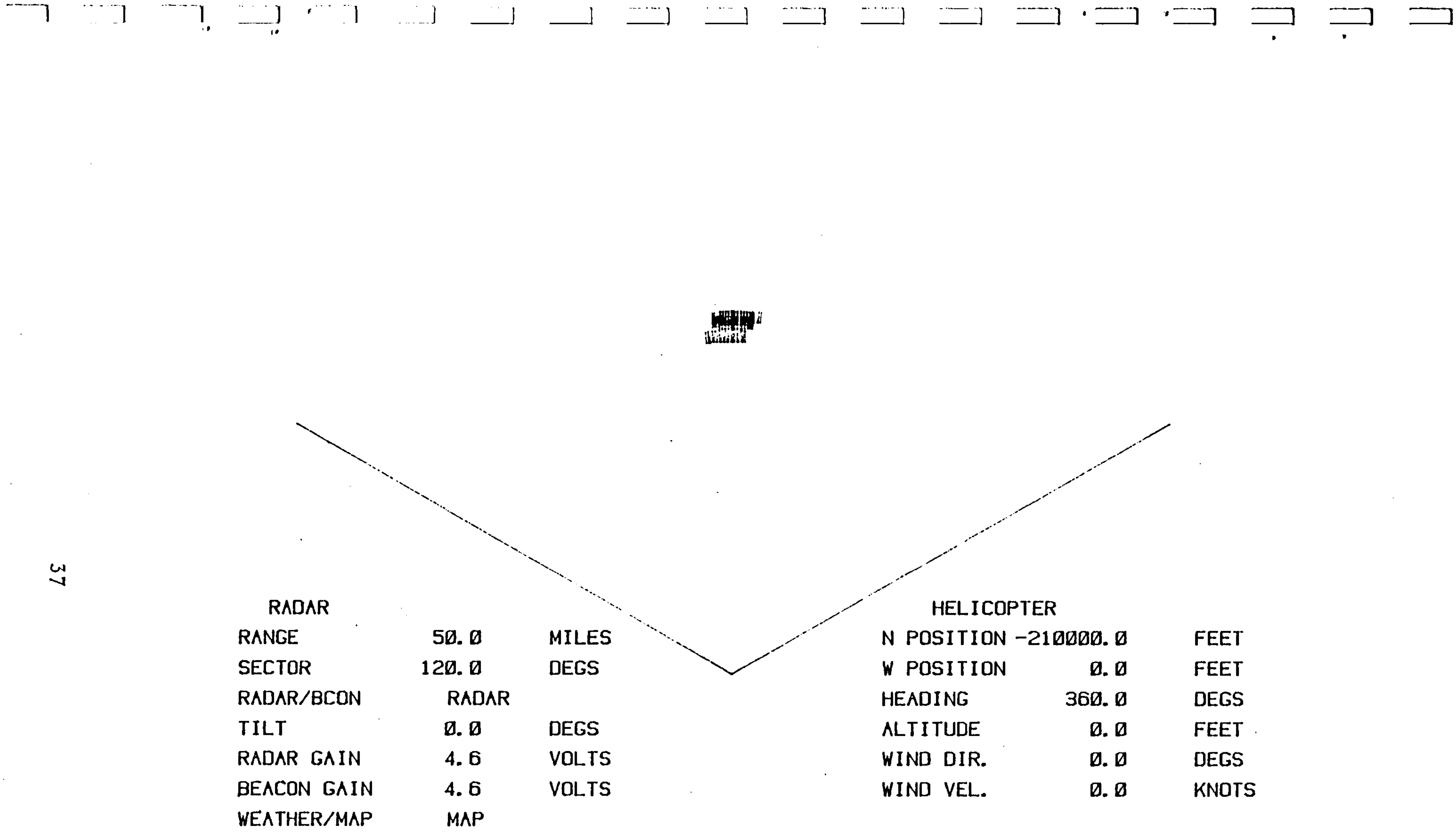

RADAR
RANGE 50.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT 0.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP MAP
HELICOPTER
N POSITION -210000.0 FEET
W POSITION 0.0 FEET
HEADING 360.0 DEGS
ALTITUDE 0.0 FEET
WIND DIR. 0.0 DEGS
WIND VEL. 0.0 KNOTS

Figure 4.6

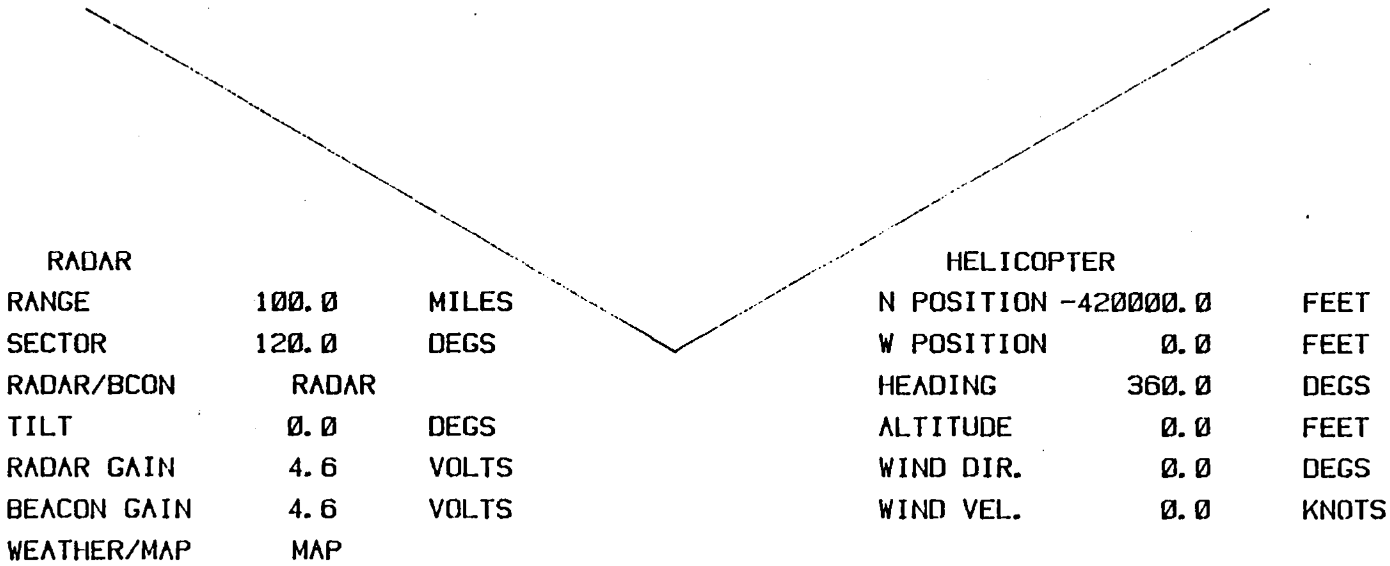

RADAR
RANGE 100.Ø MILES
SECTOR 120.Ø DEGS
RADAR/BCON RADAR
TILT Ø.Ø DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP MAP
HELICOPTER
N POSITION -420000.Ø FEET
W POSITION Ø.Ø FEET
HEADING 360.Ø DEGS
ALTITUDE Ø.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

Figure 4.7

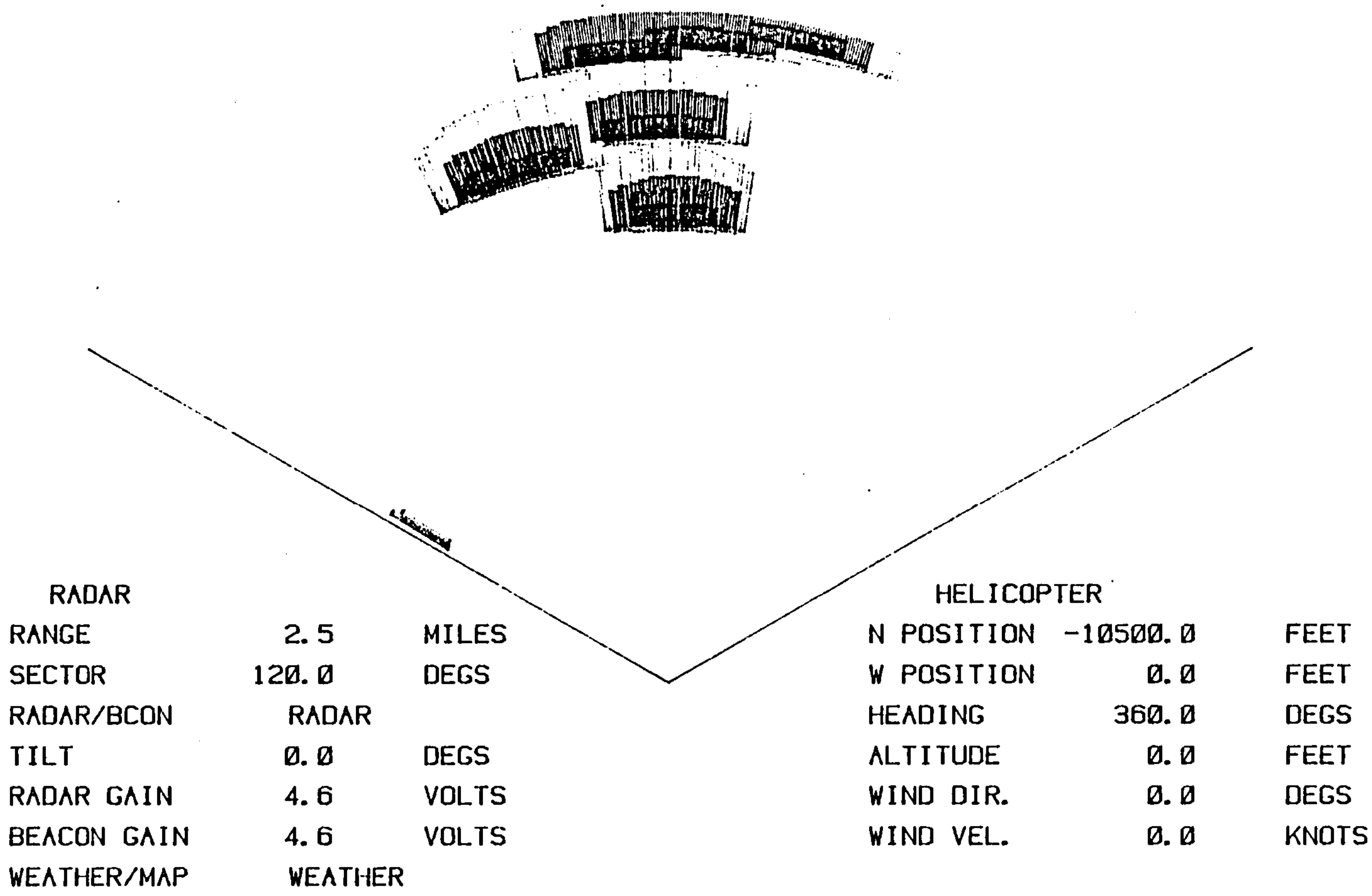
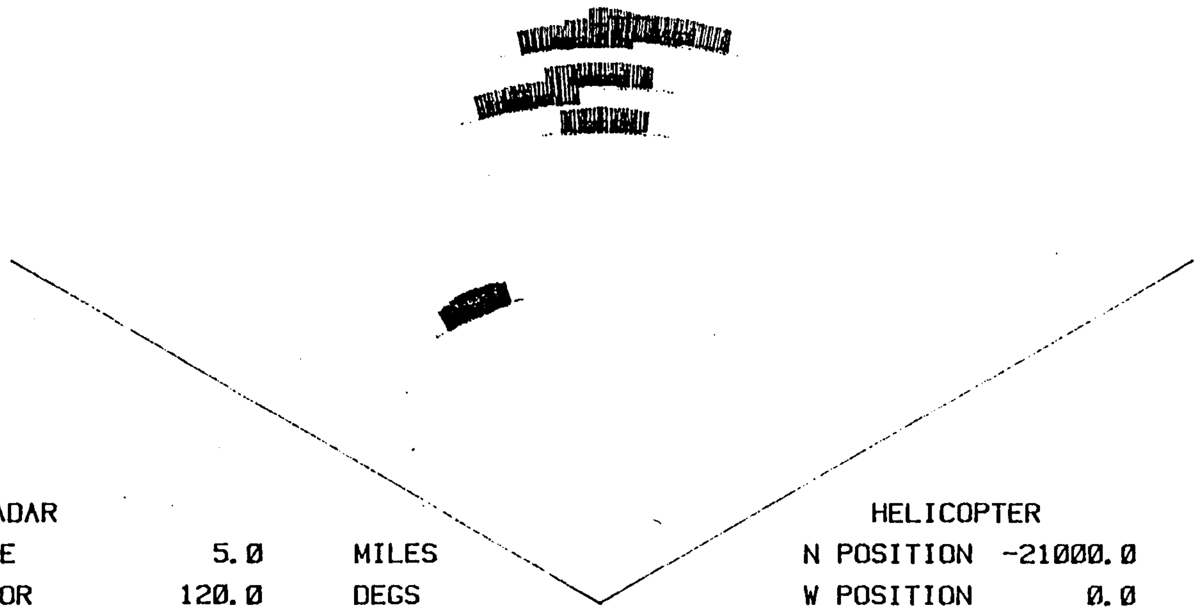

RADAR
RANGE 2.5 MILES
SECTOR 120.Ø DEGS
RADAR/BCON RADAR
TILT Ø.Ø DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -105ØØ.Ø FEET
W POSITION Ø.Ø FEET
HEADING 36Ø.Ø DEGS
ALTITUDE Ø.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

Figure 4.8



RADAR		
RANGE	5.Ø	MILES
SECTOR	120.Ø	DEGS
RADAR/BCON	RADAR	
TILT	Ø.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	36Ø.Ø	DEGS
ALTITUDE	Ø.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	Ø.Ø	KNOTS

Figure 4.9

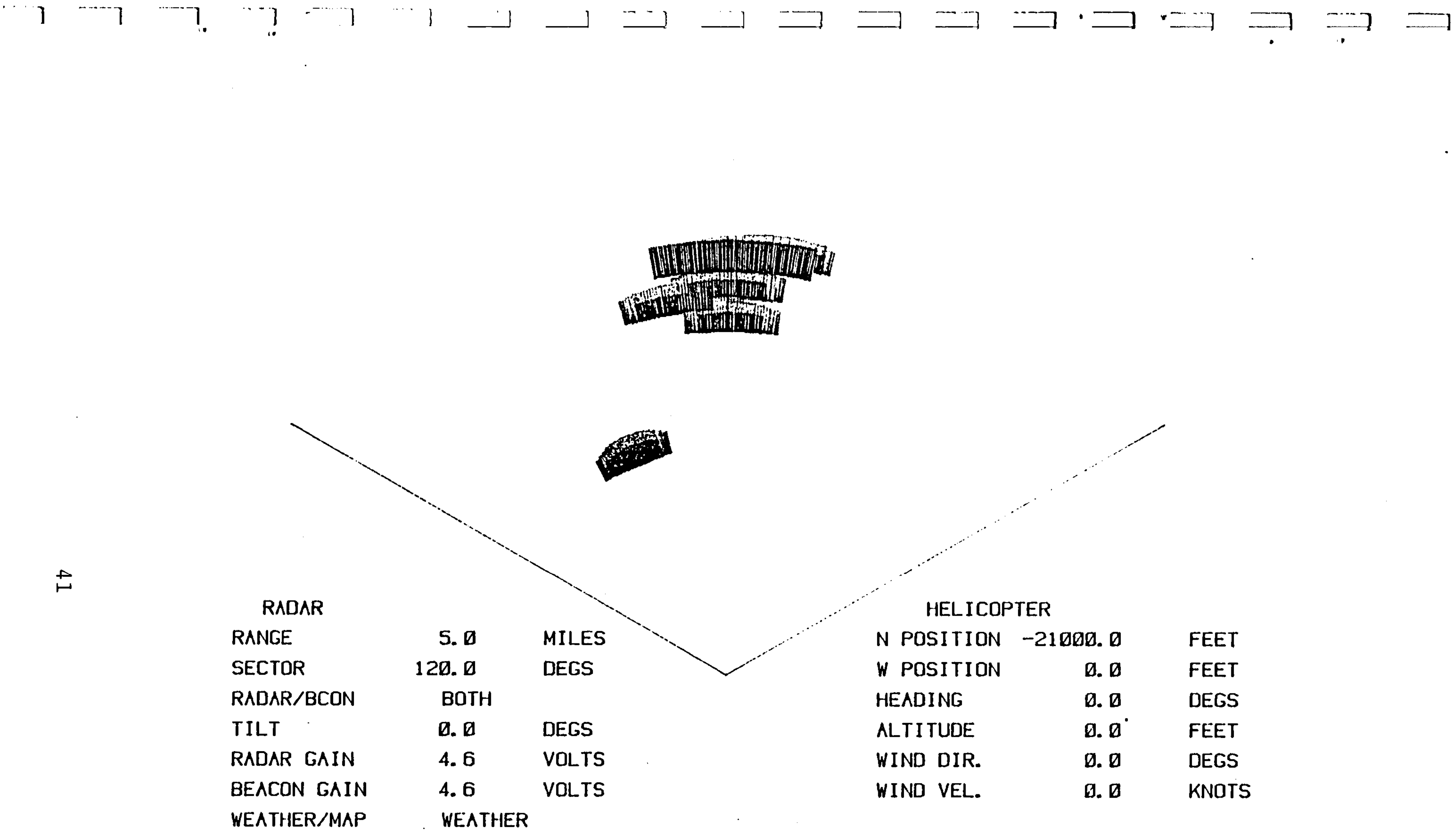

RADAR
RANGE 5.Ø MILES
SECTOR 120.Ø DEGS
RADAR/BCON BOTH
TILT Ø.Ø DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21ØØØ.Ø FEET
W POSITION Ø.Ø FEET
HEADING Ø.Ø DEGS
ALTITUDE Ø.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

Figure 4.10

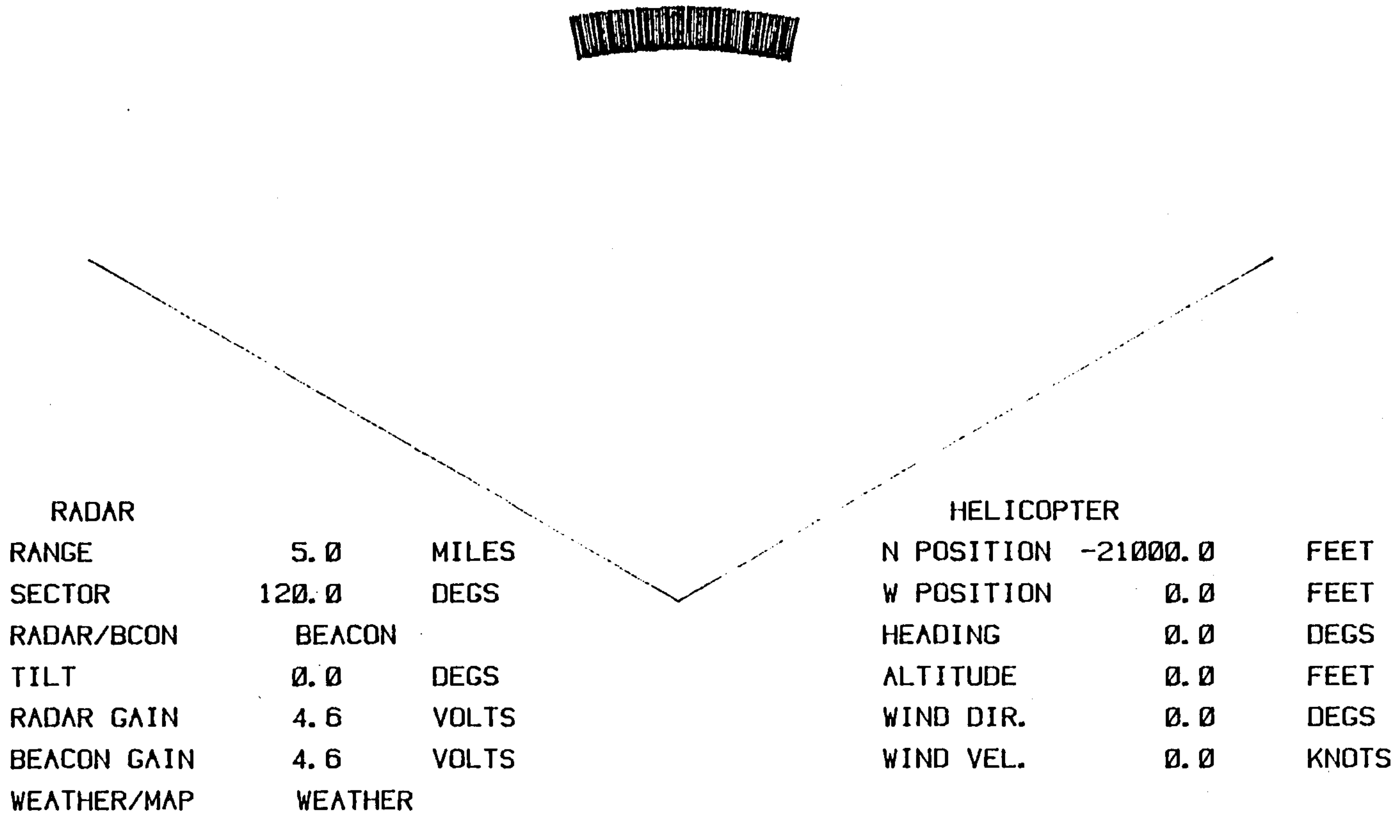
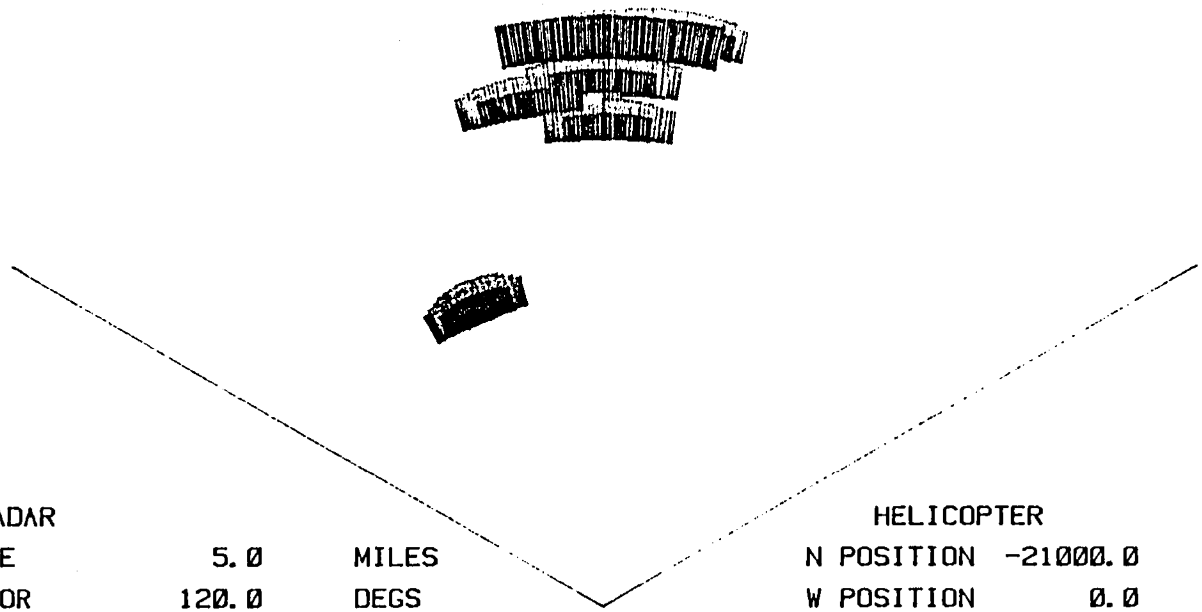

RADAR
RANGE 5.Ø MILES
SECTOR 12Ø.Ø DEGS
RADAR/BCON BEACON
TILT Ø.Ø DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21ØØØ.Ø FEET
W POSITION Ø.Ø FEET
HEADING Ø.Ø DEGS
ALTITUDE Ø.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

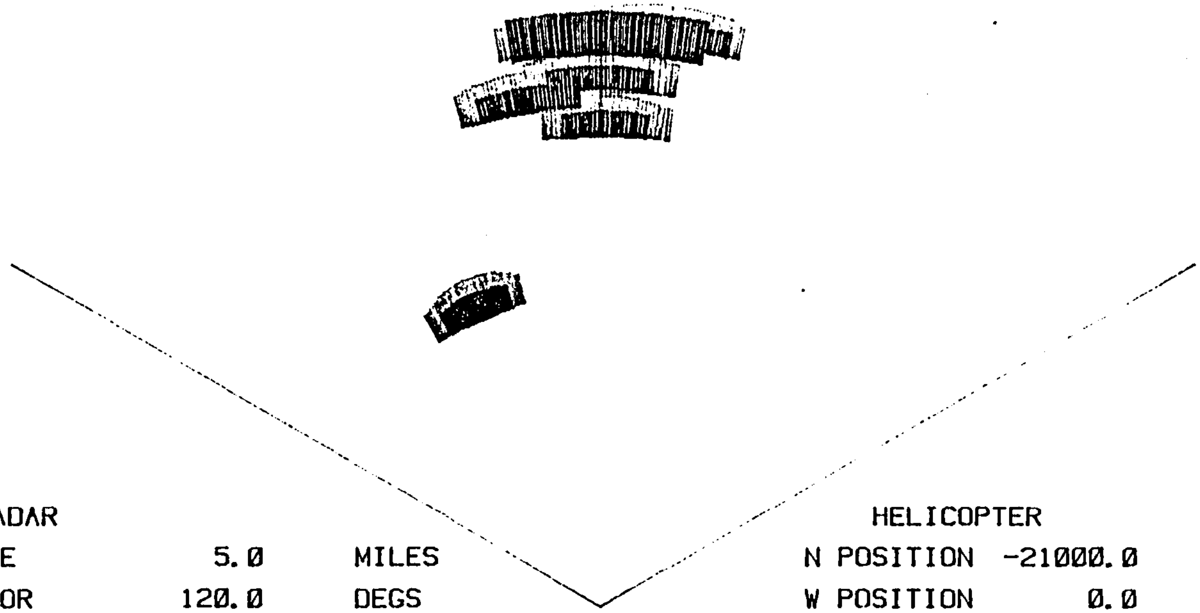
Figure 4.11



RADAR		
RANGE	5.Ø	MILES
SECTOR	12Ø.Ø	DEGS
RADAR/BCON	BOTH	
TILT	Ø.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	5.Ø	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	Ø.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	Ø.Ø	KNOTS

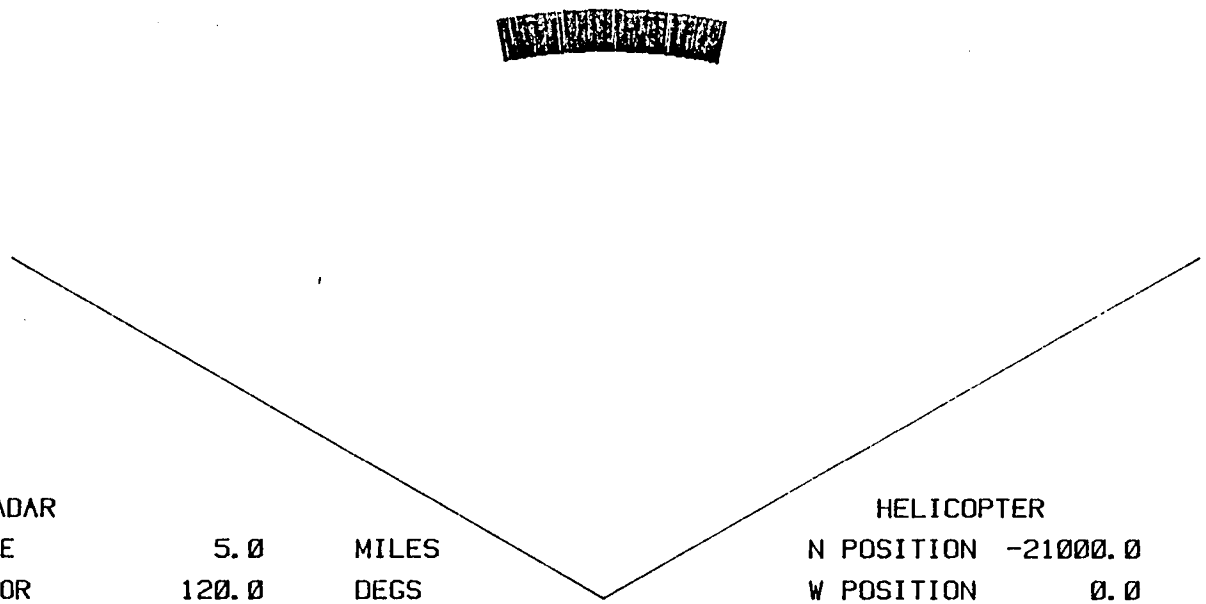
Figure 4.12



RADAR		
RANGE	5.Ø	MILES
SECTOR	120.Ø	DEGS
RADAR/BCON	BOTH	
TILT	Ø.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.Ø	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	Ø.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	Ø.Ø	KNOTS

Figure 4.13



RADAR		
RANGE	5.Ø	MILES
SECTOR	12Ø.Ø	DEGS
RADAR/BCON	BEACON	
TILT	Ø.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	Ø.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	Ø.Ø	KNOTS

Figure 4.14

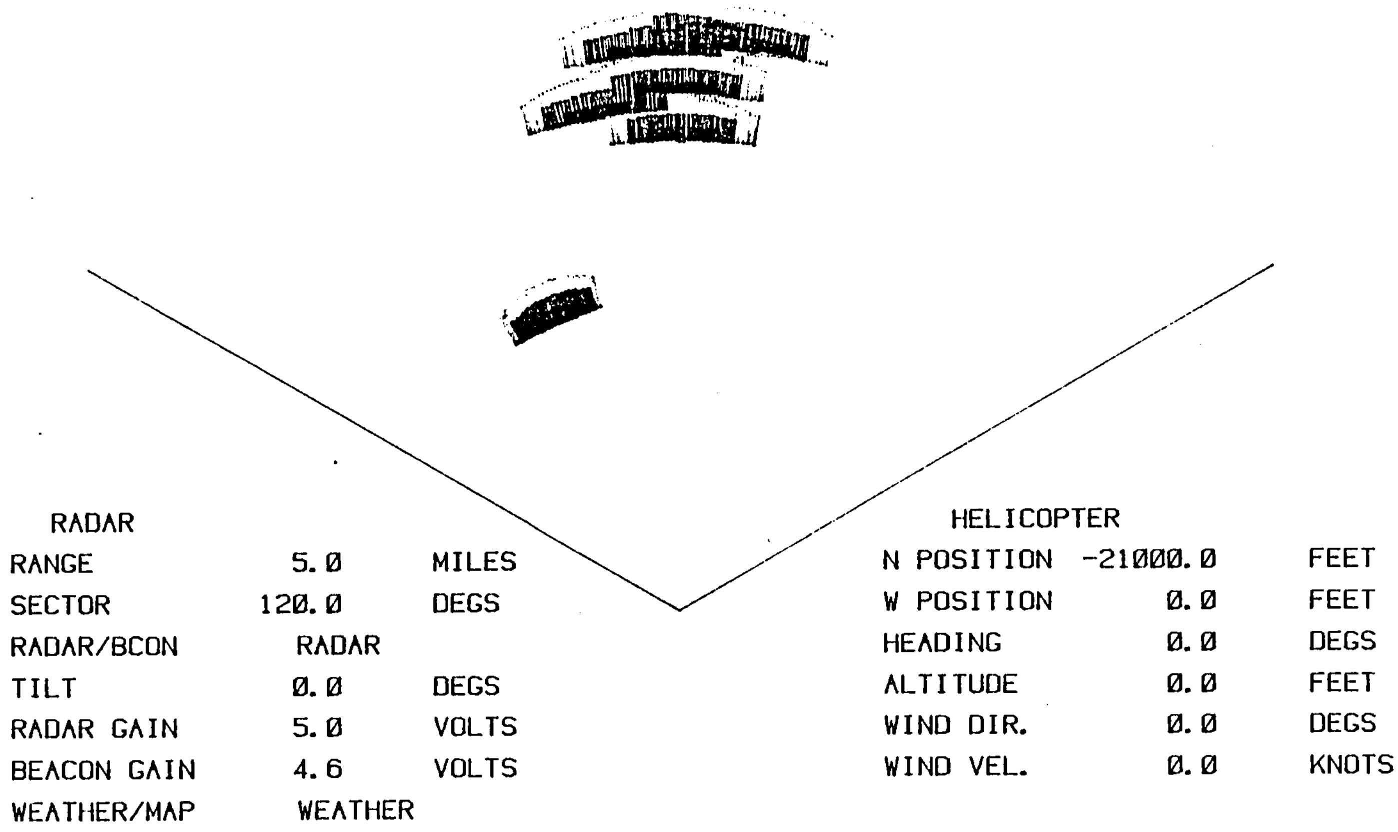

RADAR
RANGE 5.Ø MILES
SECTOR 12Ø.Ø DEGS
RADAR/BCON RADAR
TILT Ø.Ø DEGS
RADAR GAIN 5.Ø VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21ØØØ.Ø FEET
W POSITION Ø.Ø FEET
HEADING Ø.Ø DEGS
ALTITUDE Ø.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

Figure 4.15

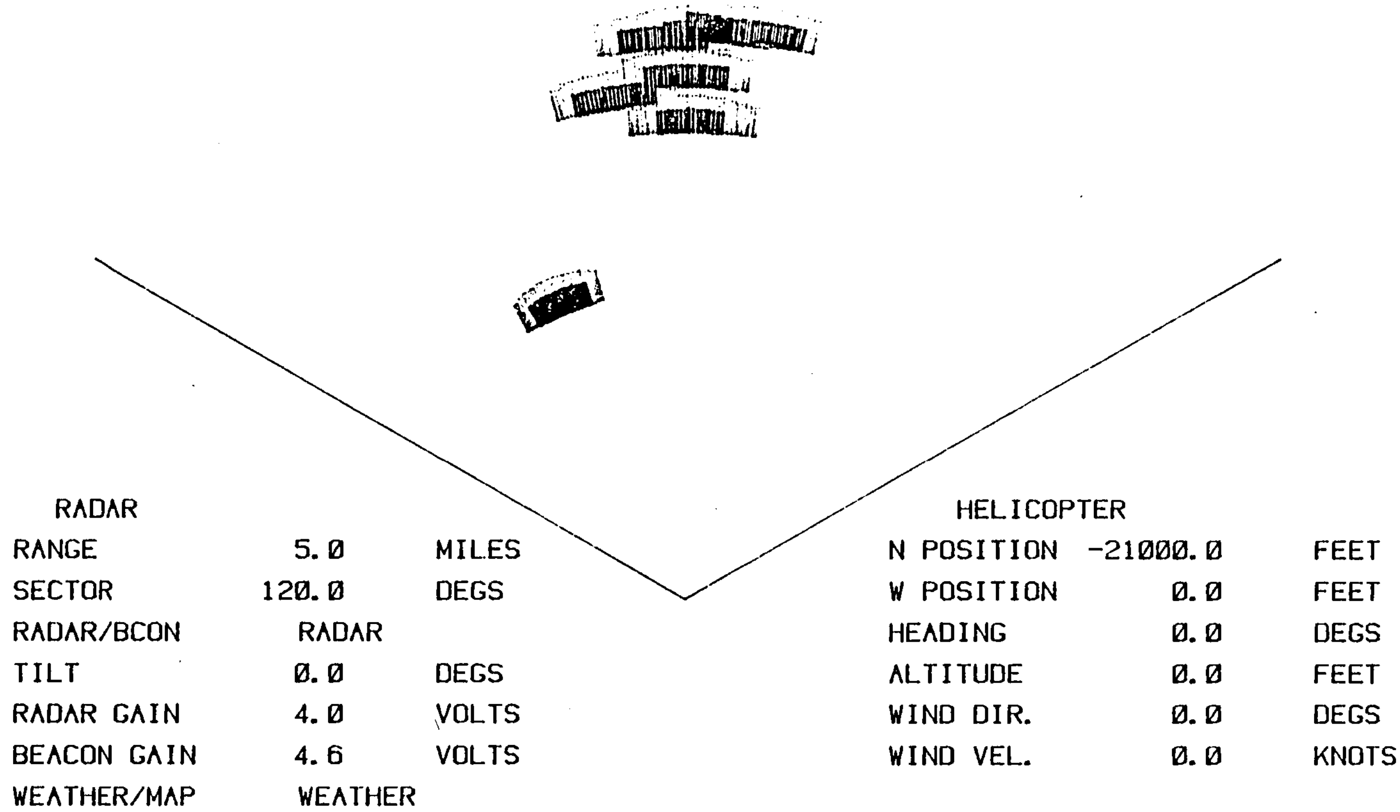
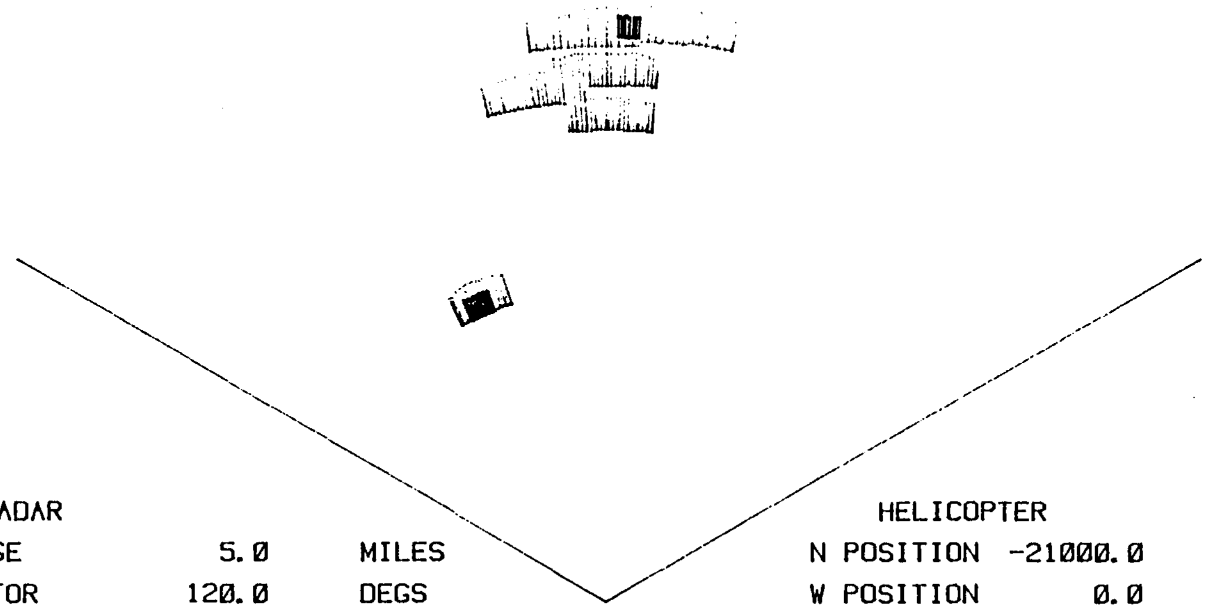

RADAR
RANGE 5.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT 0.0 DEGS
RADAR GAIN 4.0 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 0.0 FEET
WIND DIR. 0.0 DEGS
WIND VEL. 0.0 KNOTS

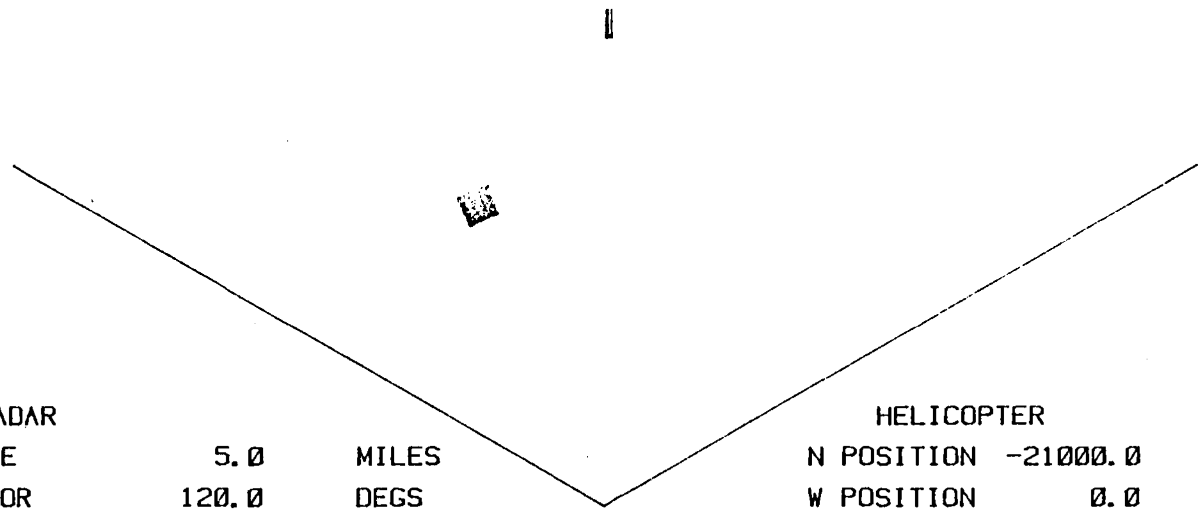
Figure 4.16



RADAR		
RANGE	5.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	RADAR	
TILT	0.0	DEGS
RADAR GAIN	3.0	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21000.0	FEET
W POSITION	0.0	FEET
HEADING	0.0	DEGS
ALTITUDE	0.0	FEET
WIND DIR.	0.0	DEGS
WIND VEL.	0.0	KNOTS

Figure 4.17



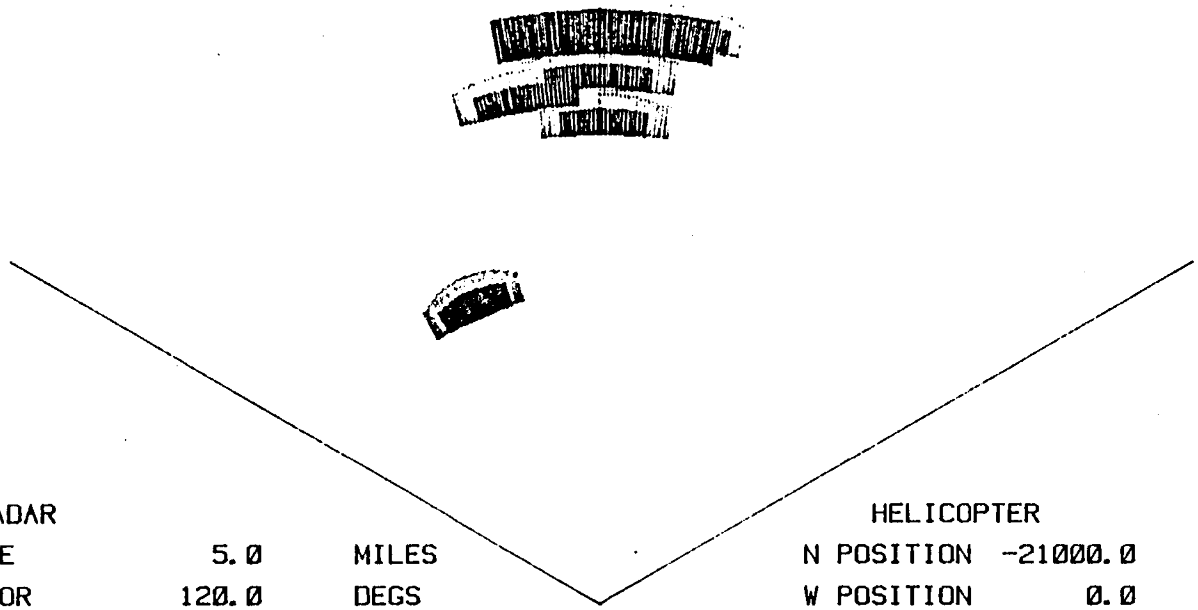
RADAR

RANGE	5.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	RADAR	
TILT	0.0	DEGS
RADAR GAIN	2.0	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER

N POSITION	-21000.0	FEET
W POSITION	0.0	FEET
HEADING	0.0	DEGS
ALTITUDE	0.0	FEET
WIND DIR.	0.0	DEGS
WIND VEL.	0.0	KNOTS

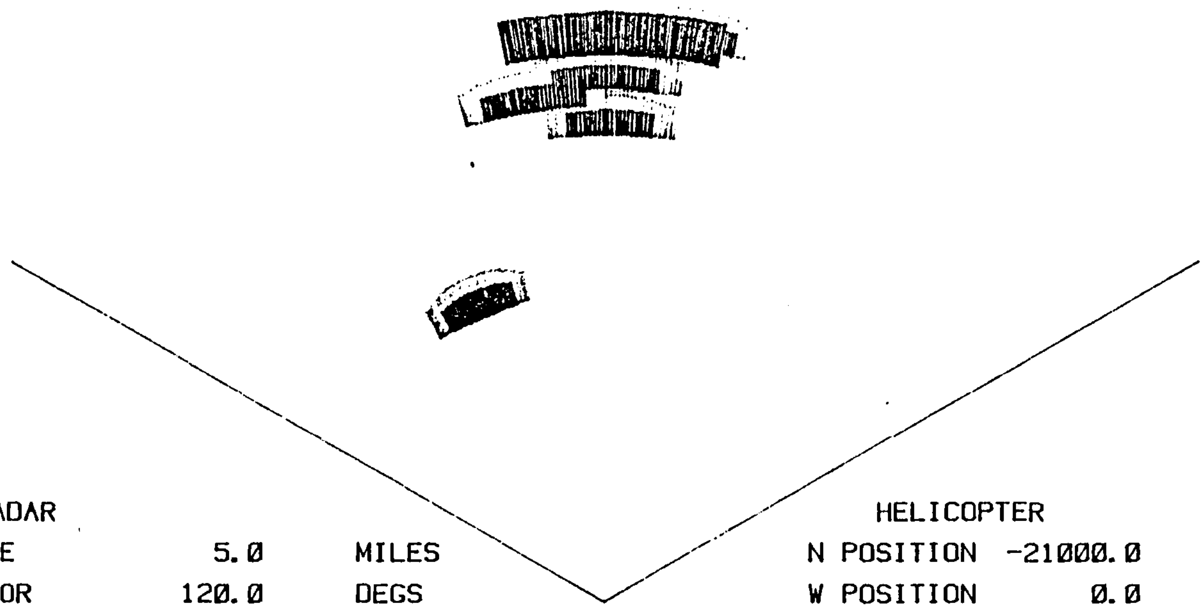
Figure 4.18



RADAR		
RANGE	5.Ø	MILES
SECTOR	120.Ø	DEGS
RADAR/BCON	BOTH	
TILT	Ø.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	5.Ø	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	Ø.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	Ø.Ø	KNOTS

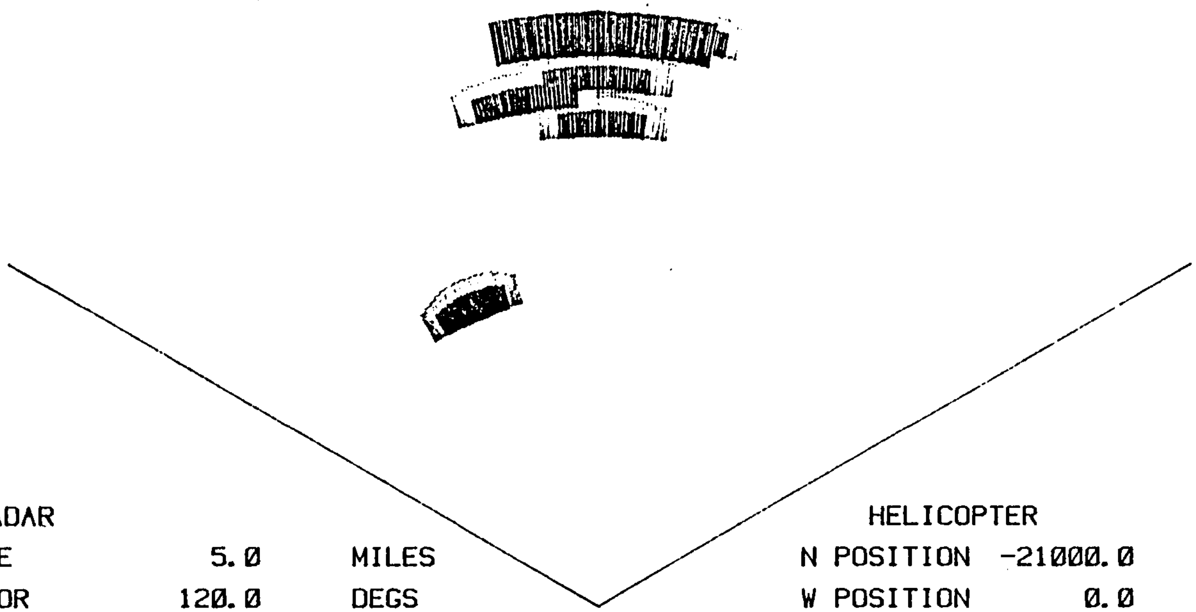
Figure 4.19



RADAR		
RANGE	5.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	BOTH	
TILT	0.0	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.0	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21000.0	FEET
W POSITION	0.0	FEET
HEADING	0.0	DEGS
ALTITUDE	0.0	FEET
WIND DIR.	0.0	DEGS
WIND VEL.	0.0	KNOTS

Figure 4.20



RADAR		
RANGE	5.Ø	MILES
SECTOR	12Ø.Ø	DEGS
RADAR/BCON	BOTH	
TILT	Ø.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	3.Ø	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	Ø.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	Ø.Ø	KNOTS

Figure 4.21

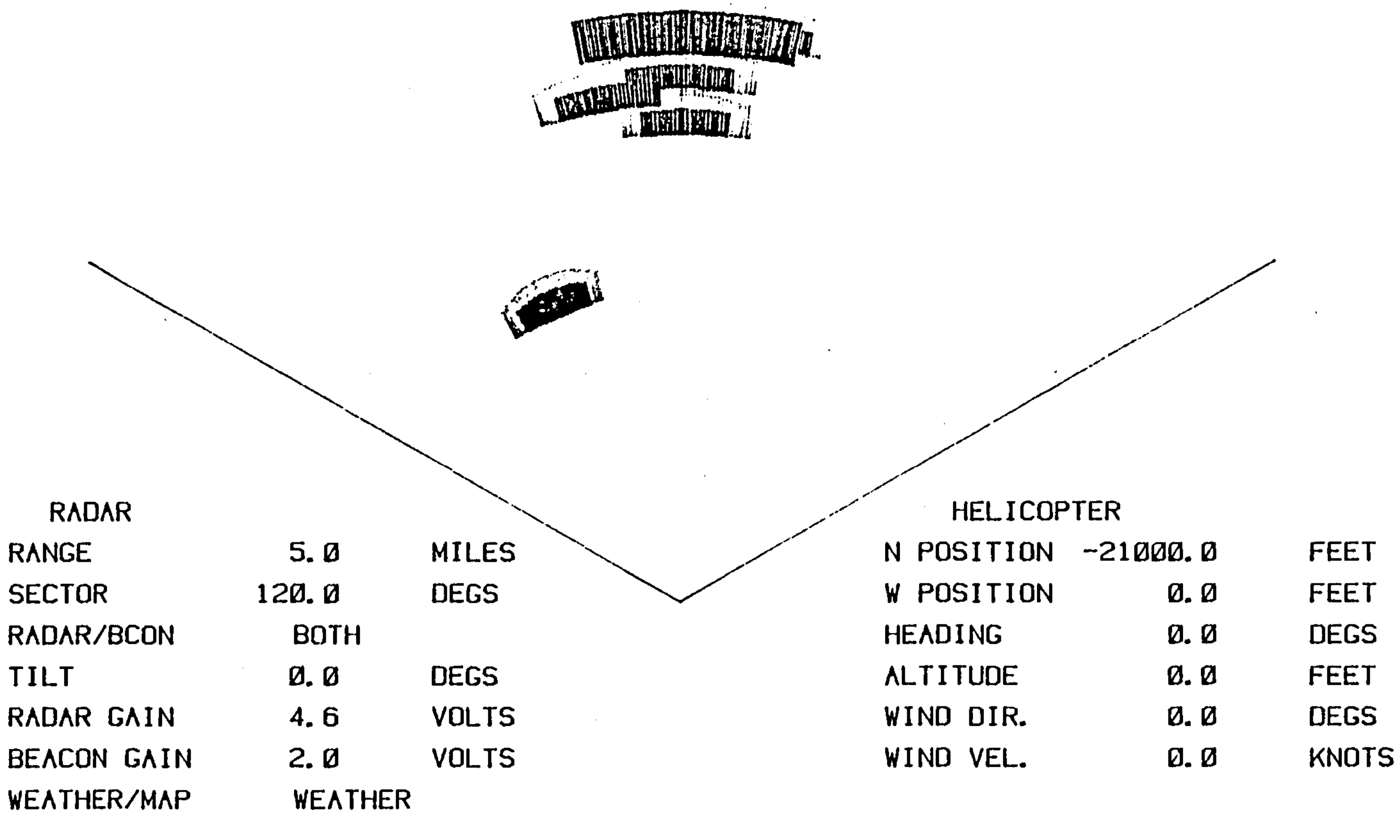
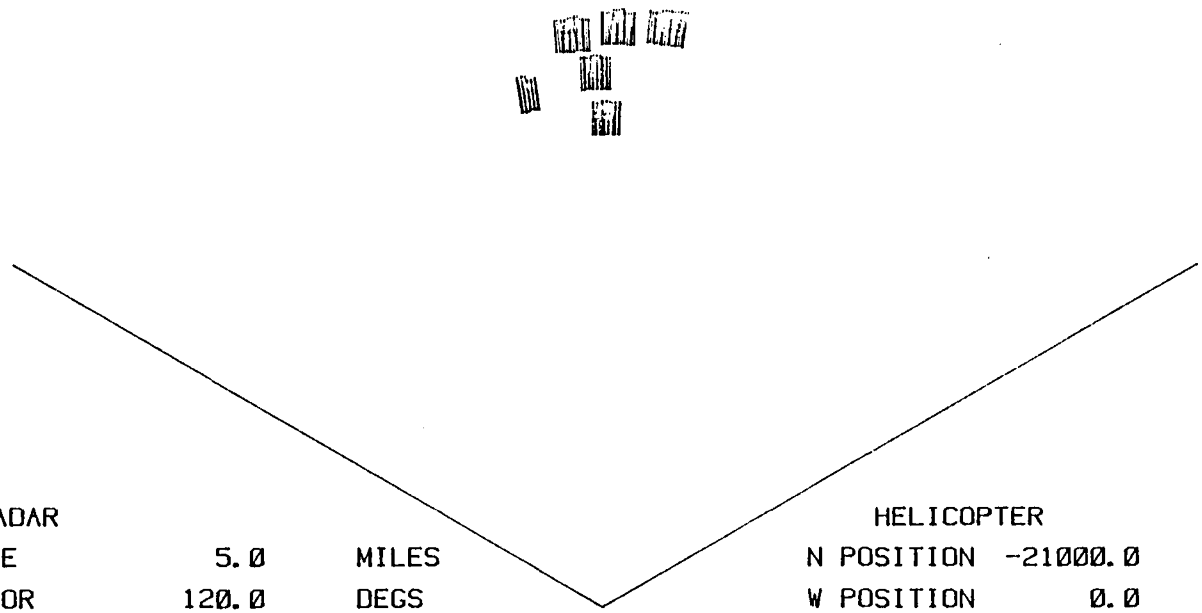

RADAR
RANGE 5.Ø MILES
SECTOR 12Ø.Ø DEGS
RADAR/BCON BOTH
TILT Ø.Ø DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 2.Ø VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21ØØØ.Ø FEET
W POSITION Ø.Ø FEET
HEADING Ø.Ø DEGS
ALTITUDE Ø.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

Figure 4.22



RADAR		
RANGE	5.Ø	MILES
SECTOR	120.Ø	DEGS
RADAR/BCON	RADAR	
TILT	3.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	2ØØØ.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	Ø.Ø	KNOTS

Figure 4.23

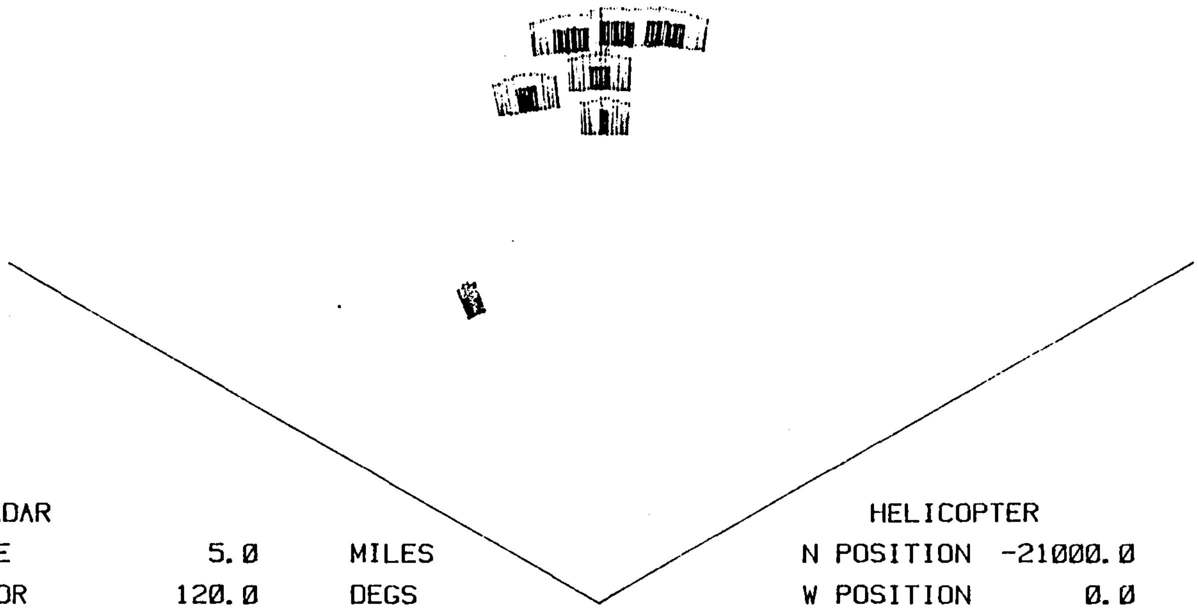
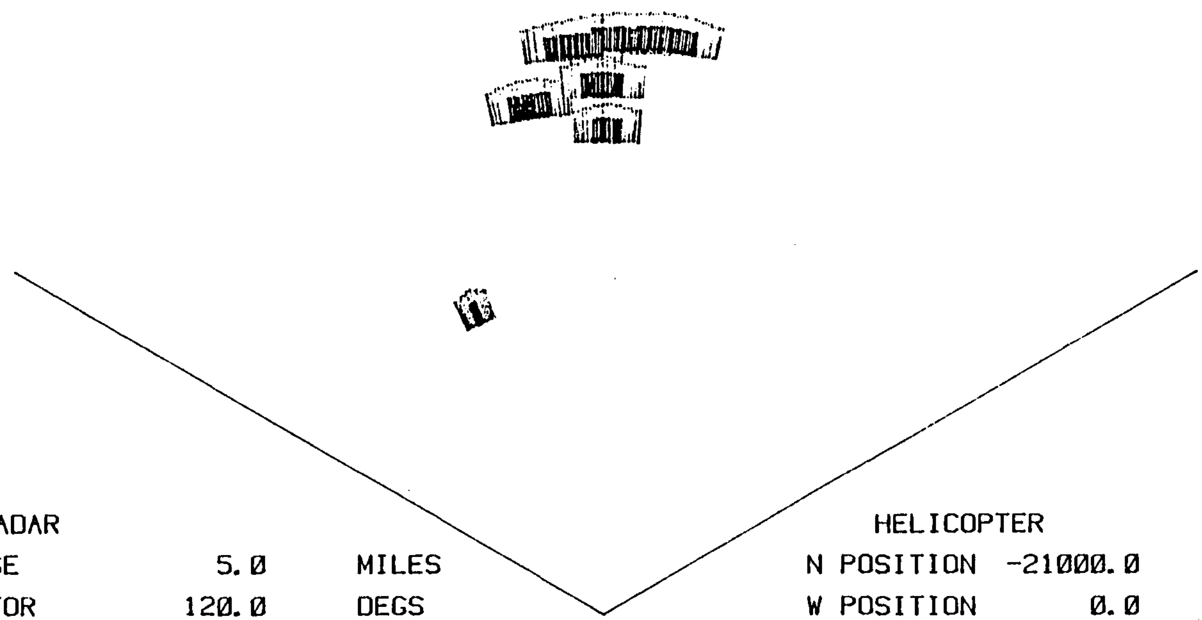

RADAR
RANGE 5.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -1.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 2000.0 FEET
WIND DIR. 0.0 DEGS
WIND VEL. 0.0 KNOTS

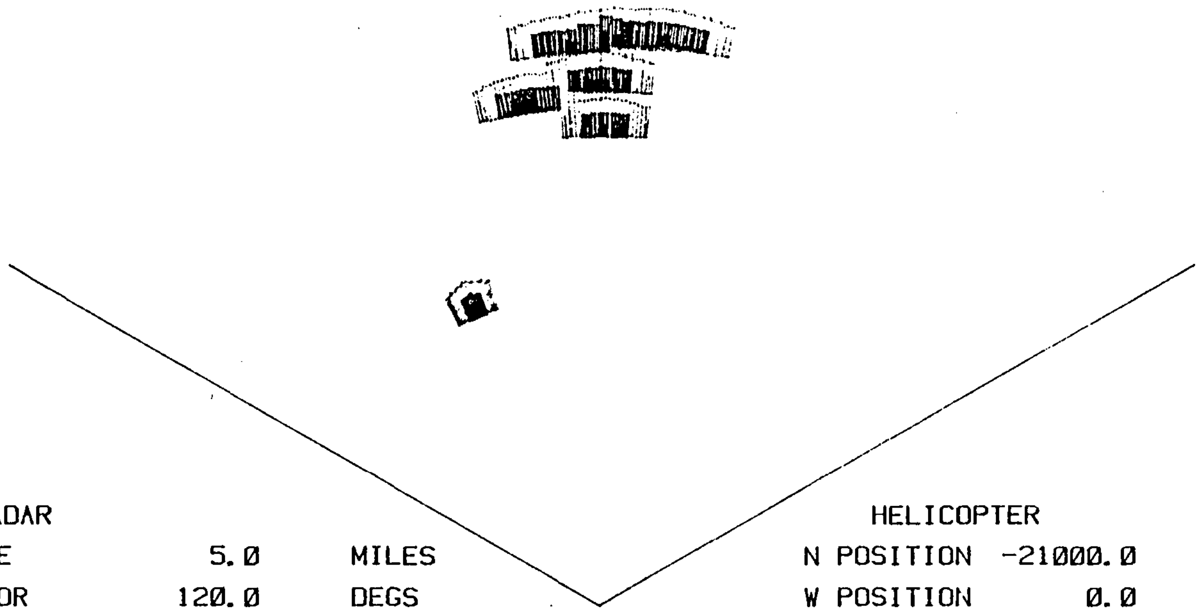
Figure 4.24



RADAR		
RANGE	5.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	RADAR	
TILT	-3.0	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21000.0	FEET
W POSITION	0.0	FEET
HEADING	0.0	DEGS
ALTITUDE	2000.0	FEET
WIND DIR.	0.0	DEGS
WIND VEL.	0.0	KNOTS

Figure 4.25



RADAR		
RANGE	5.Ø	MILES
SECTOR	120.Ø	DEGS
RADAR/BCON	RADAR	
TILT	-6.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	2ØØØ.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	Ø.Ø	KNOTS

Figure 4.26

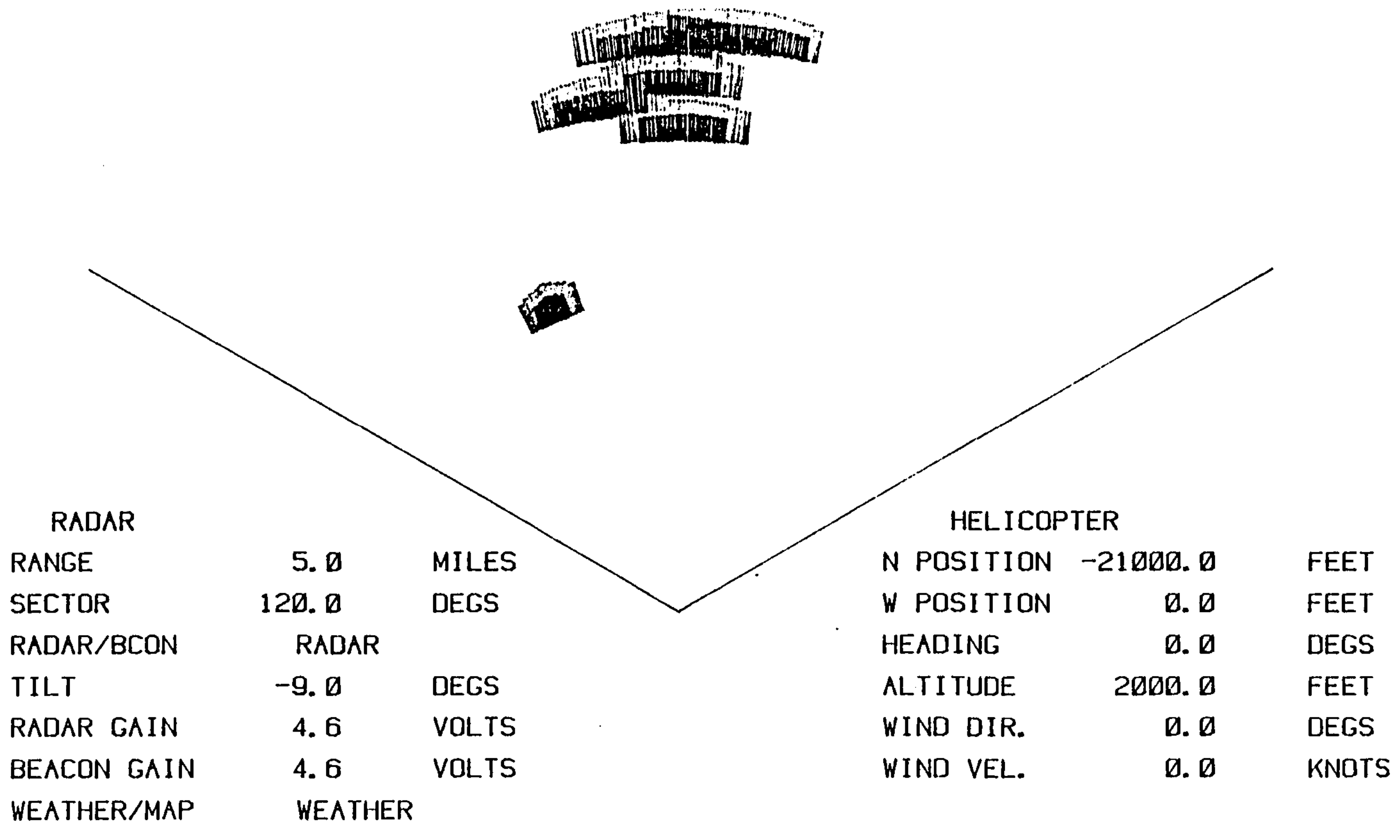
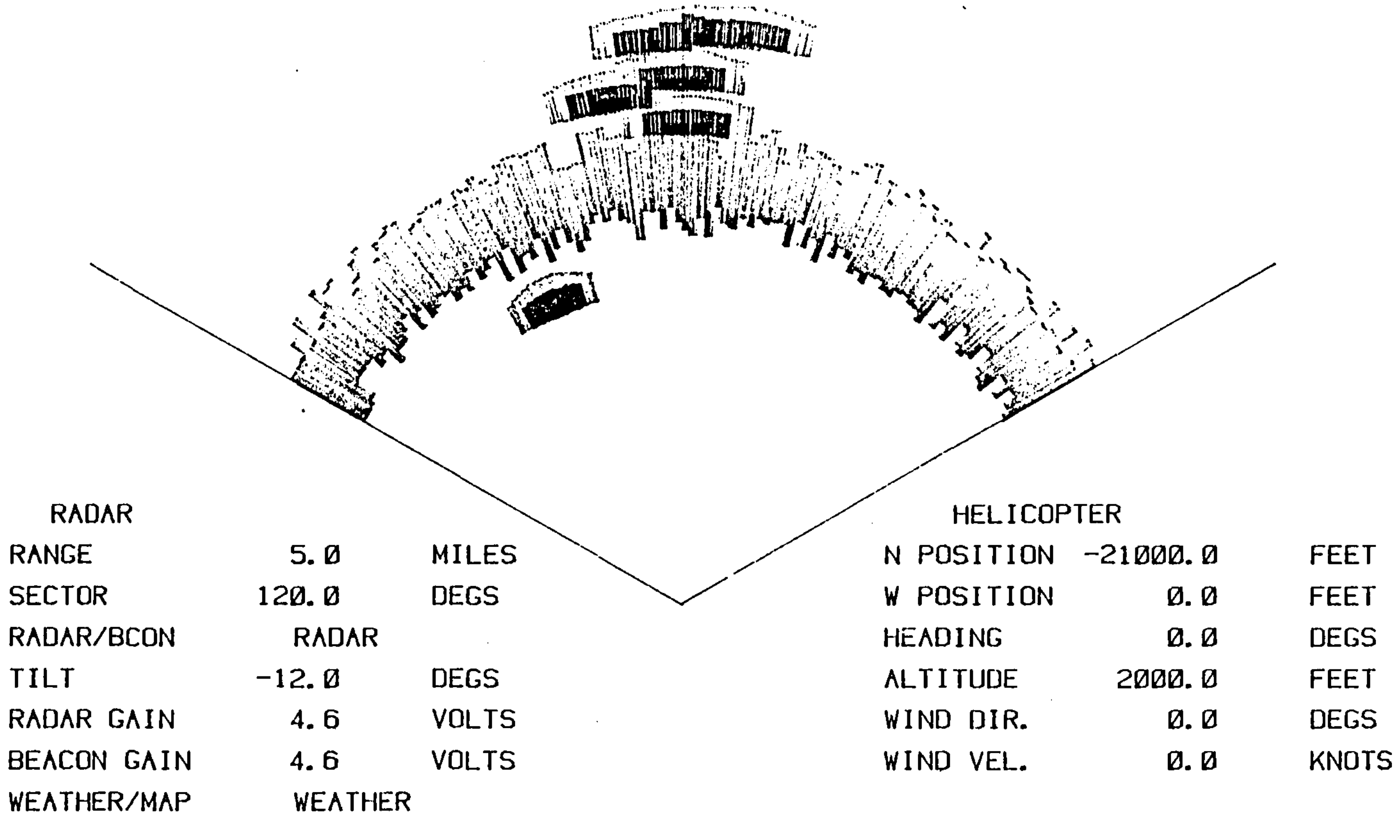

RADAR
RANGE 5.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -9.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 2000.0 FEET
WIND DIR. 0.0 DEGS
WIND VEL. 0.0 KNOTS

Figure 4.27



RADAR
RANGE 5.Ø MILES
SECTOR 120.Ø DEGS
RADAR/BCON RADAR
TILT -12.Ø DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21ØØØ.Ø FEET
W POSITION Ø.Ø FEET
HEADING Ø.Ø DEGS
ALTITUDE 2ØØØ.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

Figure 4.28

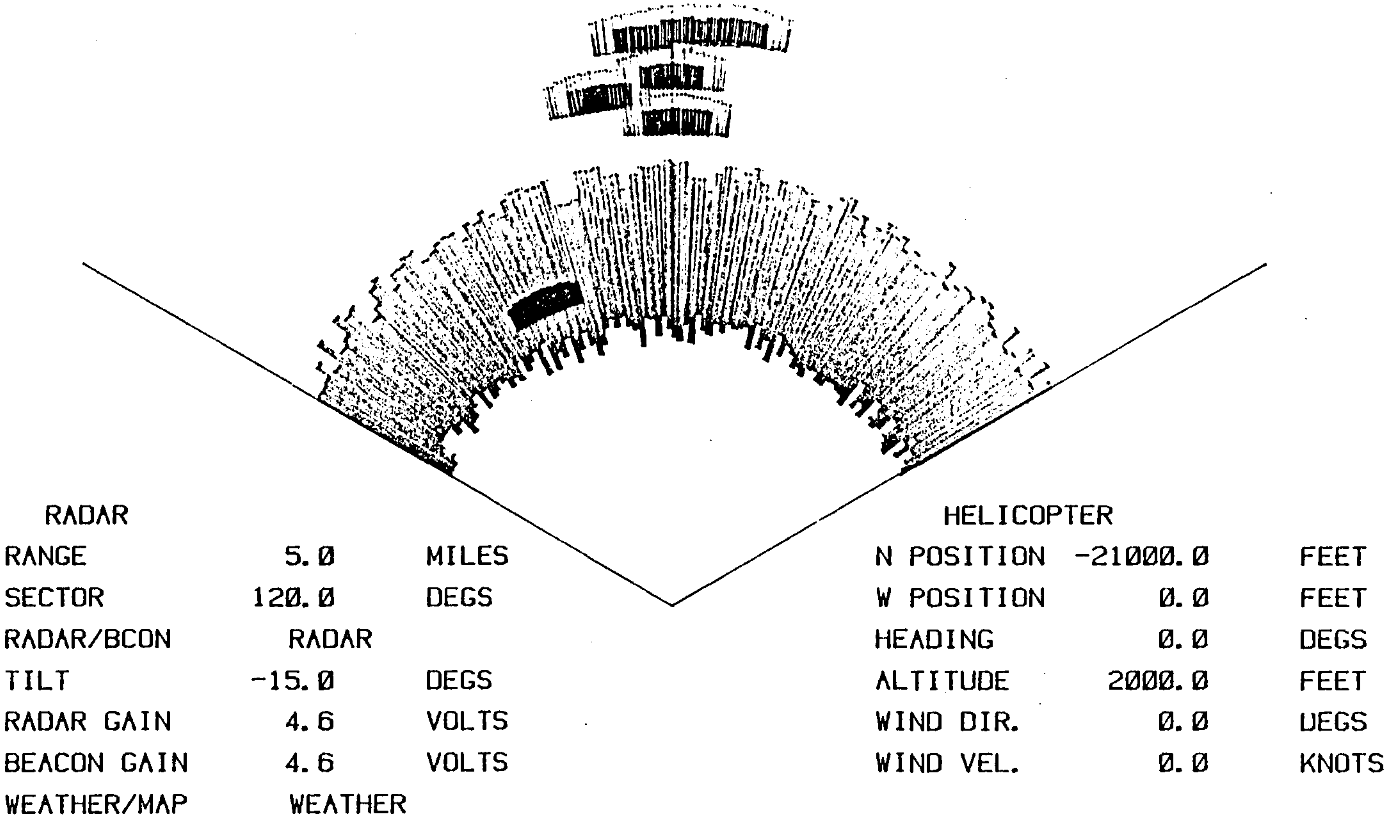

RADAR
RANGE 5.Ø MILES
SECTOR 120.Ø DEGS
RADAR/BCON RADAR
TILT -15.Ø DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.Ø FEET
W POSITION Ø.Ø FEET
HEADING Ø.Ø DEGS
ALTITUDE 2000.Ø FEET
WIND DIR. Ø.Ø DEGS
WIND VEL. Ø.Ø KNOTS

Figure 4.29

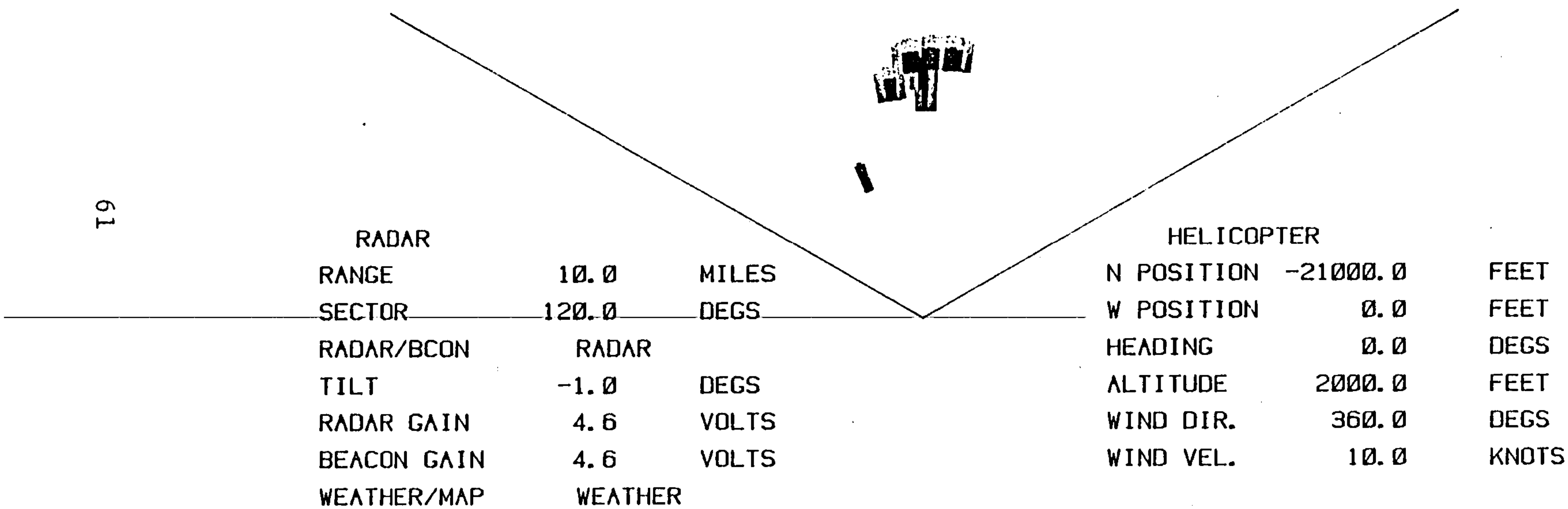
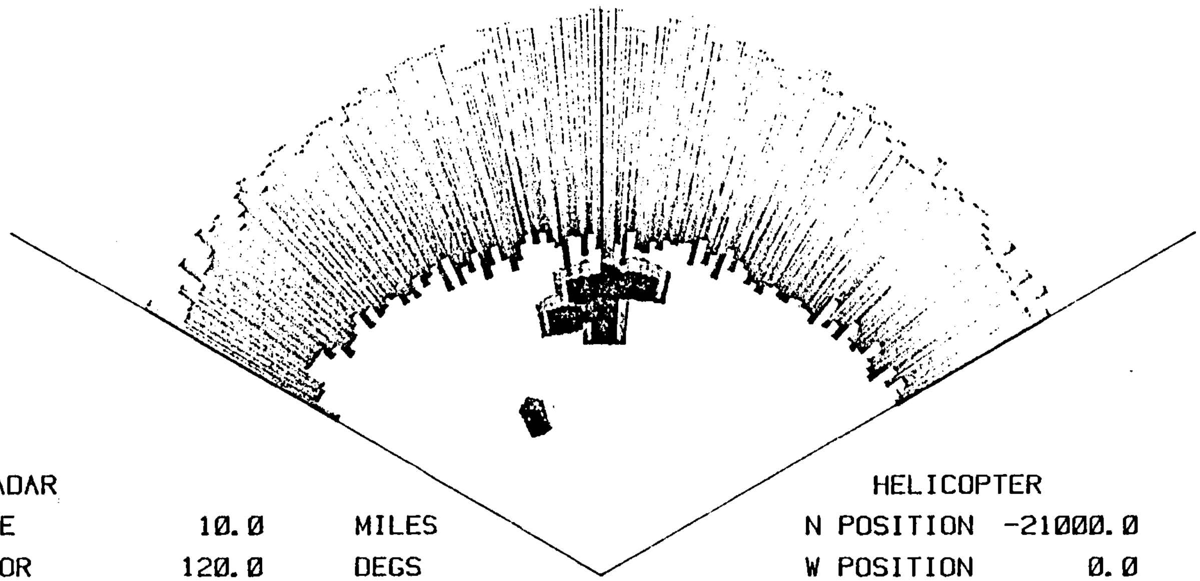

RADAR
RANGE 10.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -1.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 2000.0 FEET
WIND DIR. 360.0 DEGS
WIND VEL. 10.0 KNOTS

Figure 4.30



RADAR		
RANGE	10.Ø	MILES
SECTOR	120.Ø	DEGS
RADAR/BCON	RADAR	
TILT	-6.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	2ØØØ.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	1Ø.Ø	KNOTS

Figure 4.32

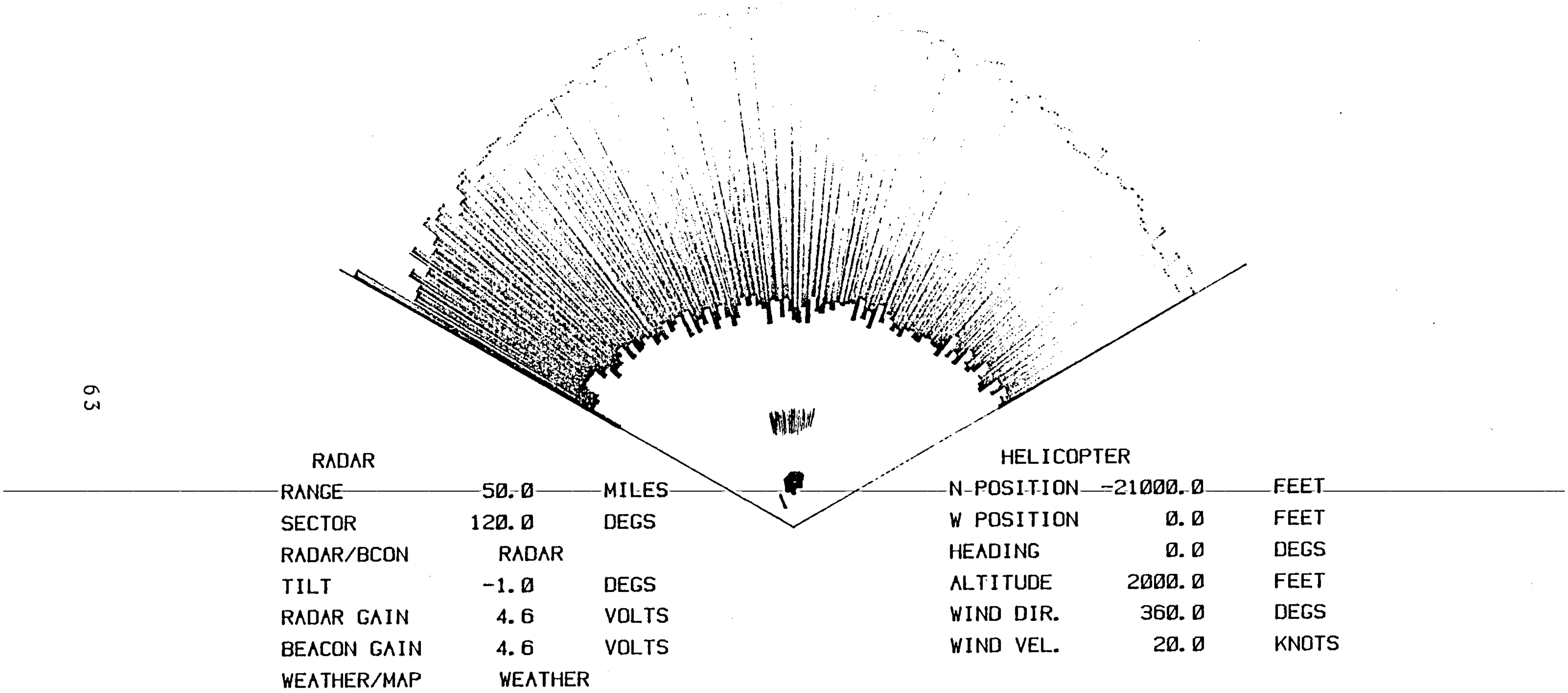
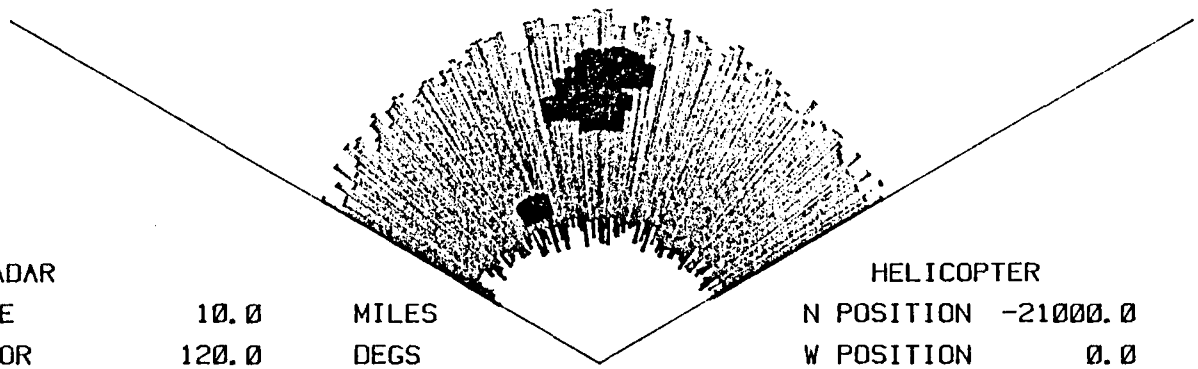

RADAR
RANGE 50.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -1.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 2000.0 FEET
WIND DIR. 360.0 DEGS
WIND VEL. 20.0 KNOTS

Figure 4.35



RADAR		
RANGE	10.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	RADAR	
TILT	-12.0	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21000.0	FEET
W POSITION	0.0	FEET
HEADING	0.0	DEGS
ALTITUDE	2000.0	FEET
WIND DIR.	0.0	DEGS
WIND VEL.	10.0	KNOTS

Figure 4.34

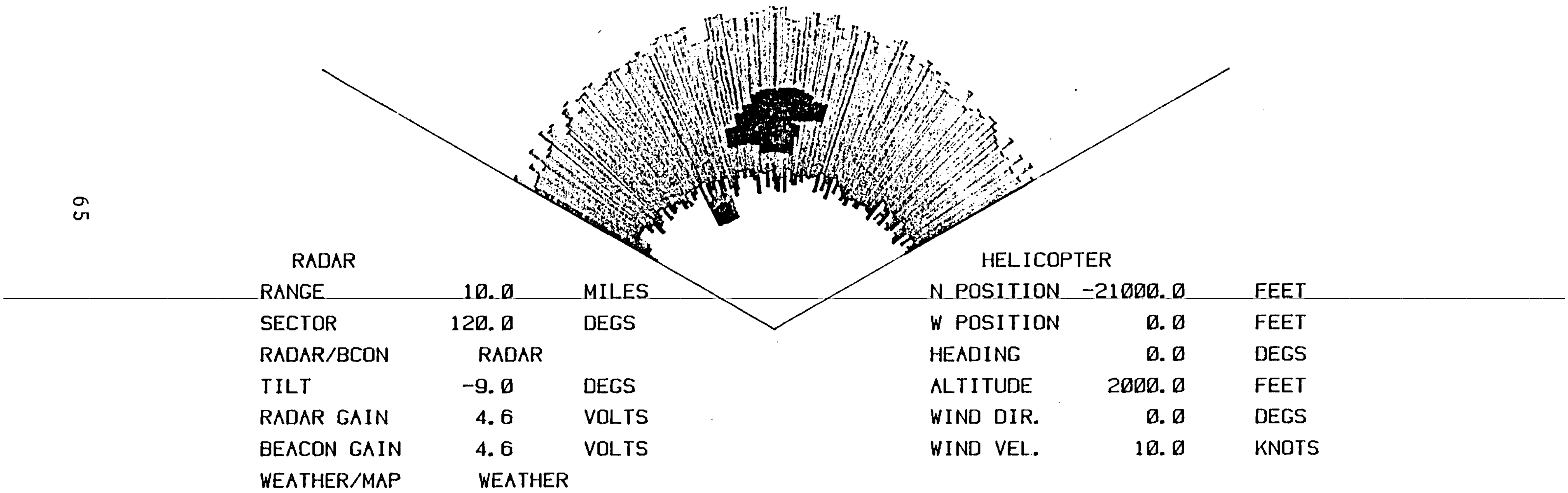

RADAR
RANGE 10.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -9.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 2000.0 FEET
WIND DIR. 0.0 DEGS
WIND VEL. 10.0 KNOTS

Figure 4.33

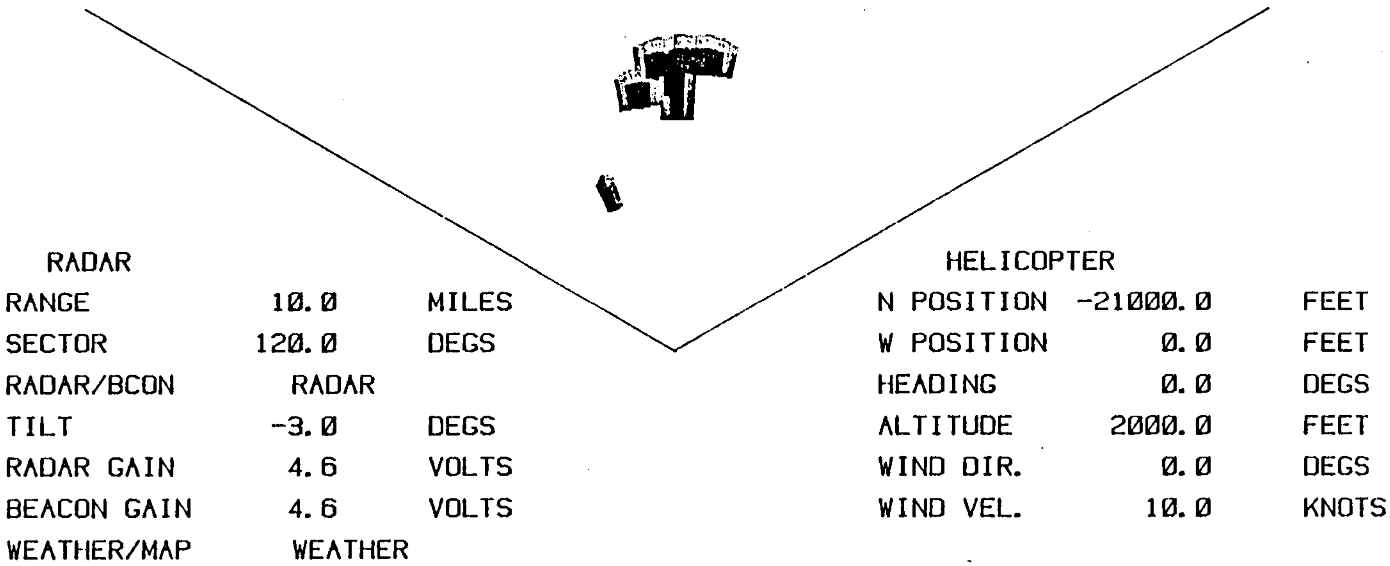

RADAR
RANGE 10.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -3.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 2000.0 FEET
WIND DIR. 0.0 DEGS
WIND VEL. 10.0 KNOTS

Figure 4.31

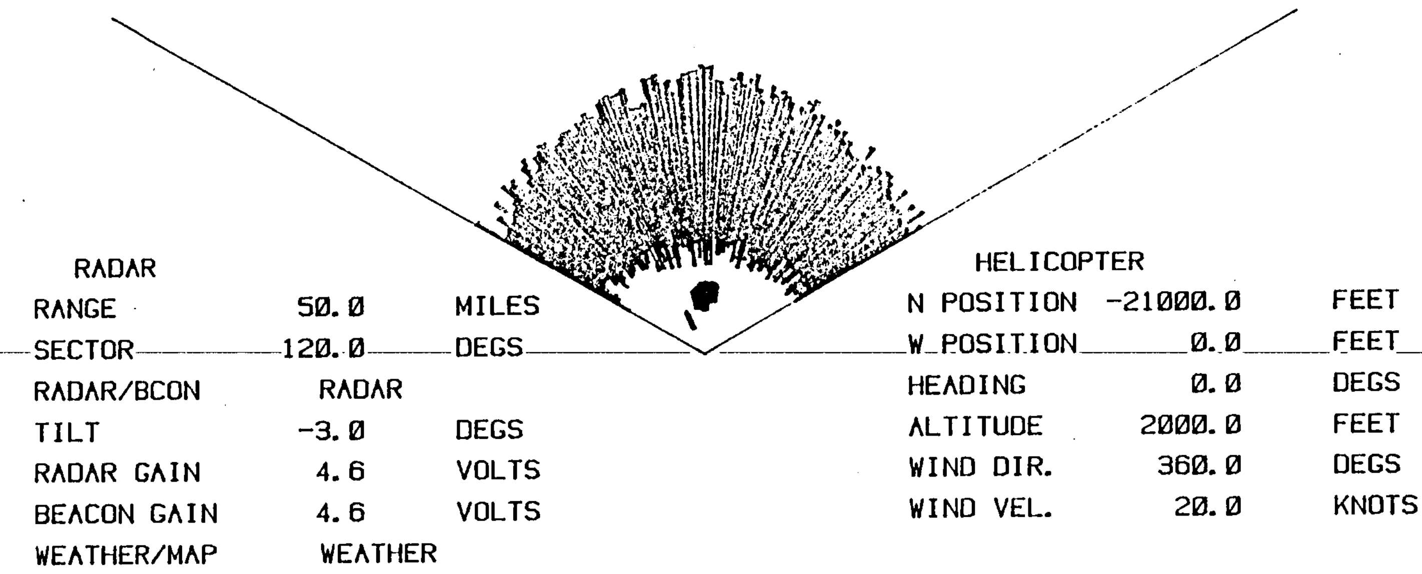
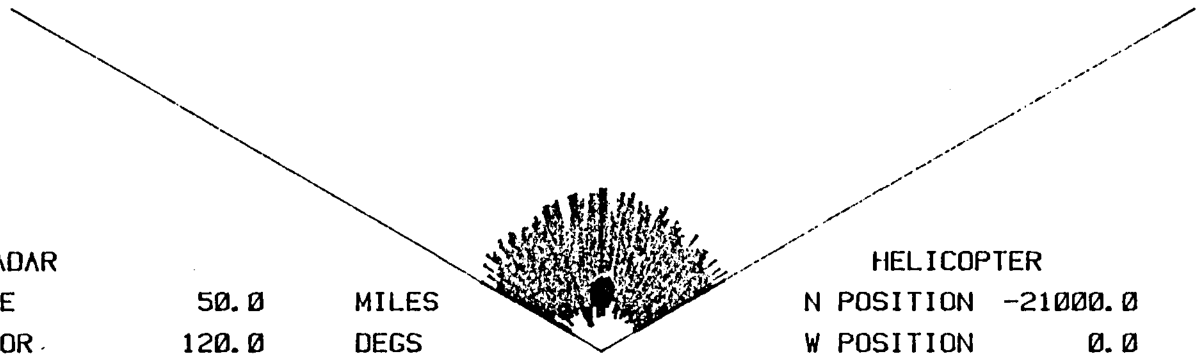

RADAR
RANGE 50.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -3.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 2000.0 FEET
WIND DIR. 360.0 DEGS
WIND VEL. 20.0 KNOTS

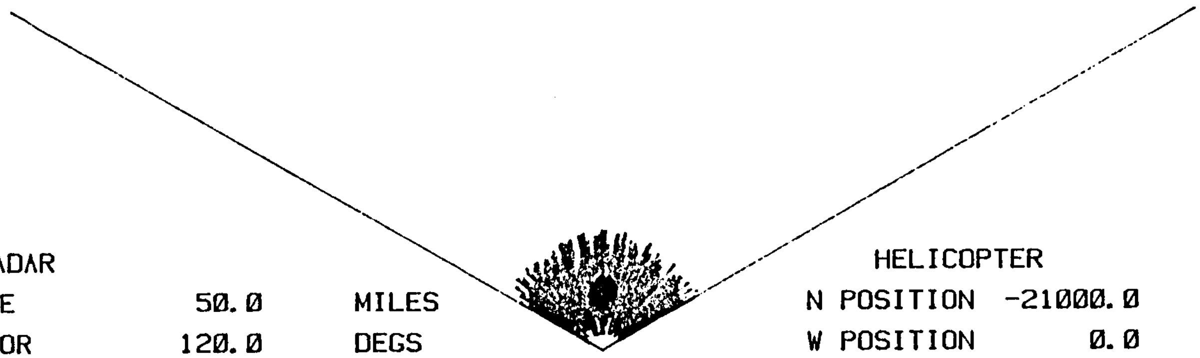
Figure 4.36



RADAR		
RANGE	50.Ø	MILES
SECTOR	120.Ø	DEGS
RADAR/BCON	RADAR	
TILT	-6.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	2ØØØ.Ø	FEET
WIND DIR.	36Ø.Ø	DEGS
WIND VEL.	2Ø.Ø	KNOTS

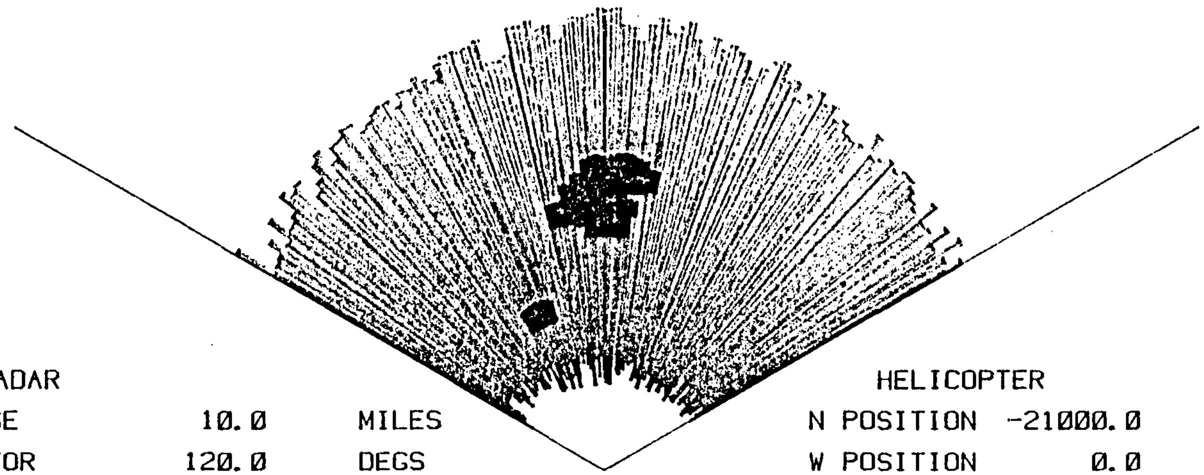
Figure 4.37



RADAR		
RANGE	50.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	RADAR	
TILT	-9.0	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21000.0	FEET
W POSITION	0.0	FEET
HEADING	0.0	DEGS
ALTITUDE	2000.0	FEET
WIND DIR.	360.0	DEGS
WIND VEL.	20.0	KNOTS

Figure 4.38



RADAR		
RANGE	10.Ø	MILES
SECTOR	120.Ø	DEGS
RADAR/BCON	RADAR	
TILT	-12.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21000.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	2000.Ø	FEET
WIND DIR.	Ø.Ø	DEGS
WIND VEL.	20.Ø	KNOTS

Figure 4.39

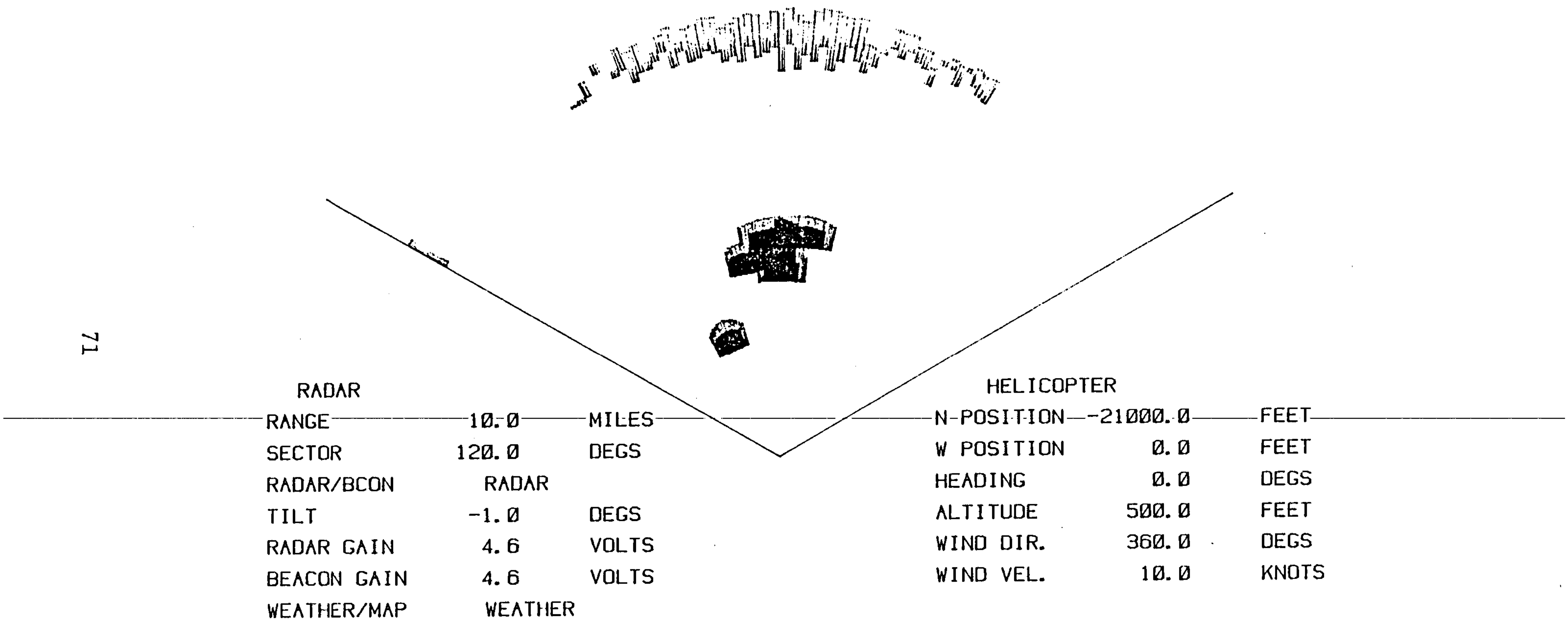

RADAR
RANGE 10.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -1.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N-POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 500.0 FEET
WIND DIR. 360.0 DEGS
WIND VEL. 10.0 KNOTS

Figure 4.40

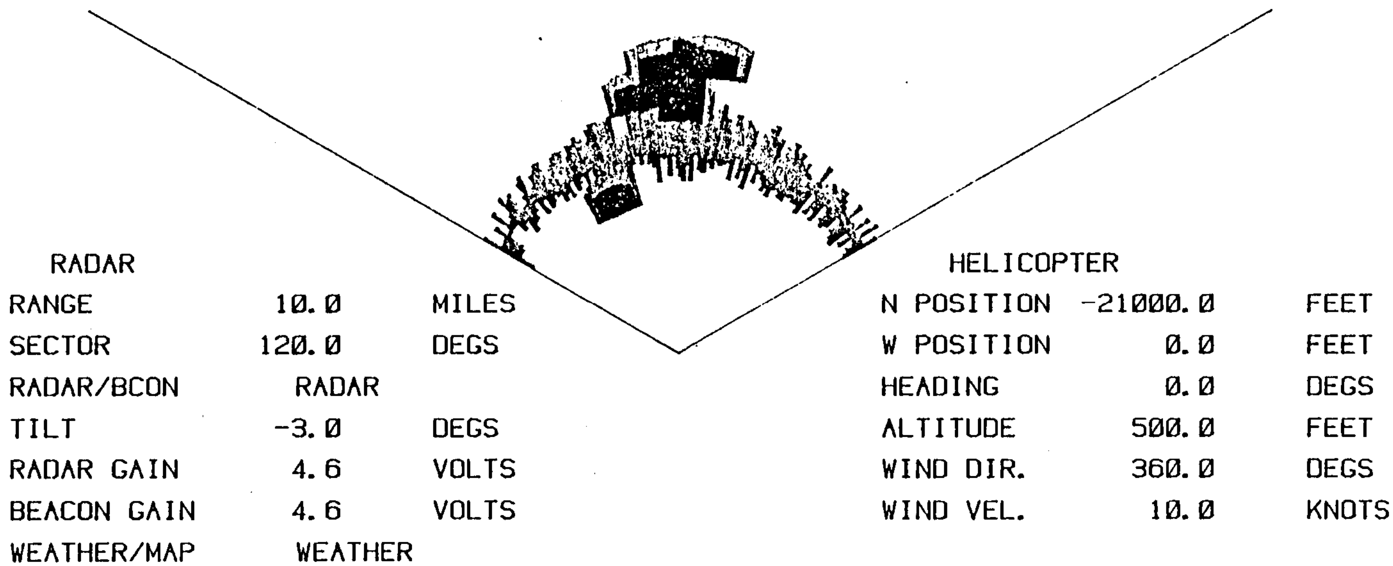
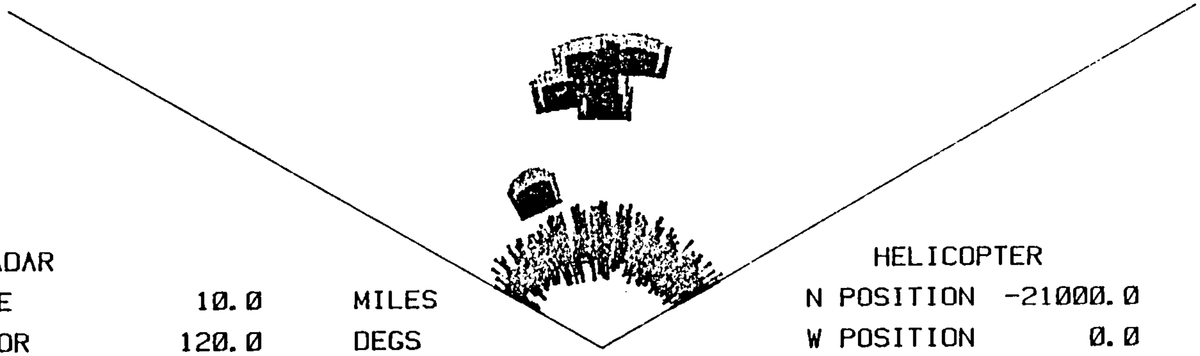

RADAR
RANGE 10.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -3.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 500.0 FEET
WIND DIR. 360.0 DEGS
WIND VEL. 10.0 KNOTS

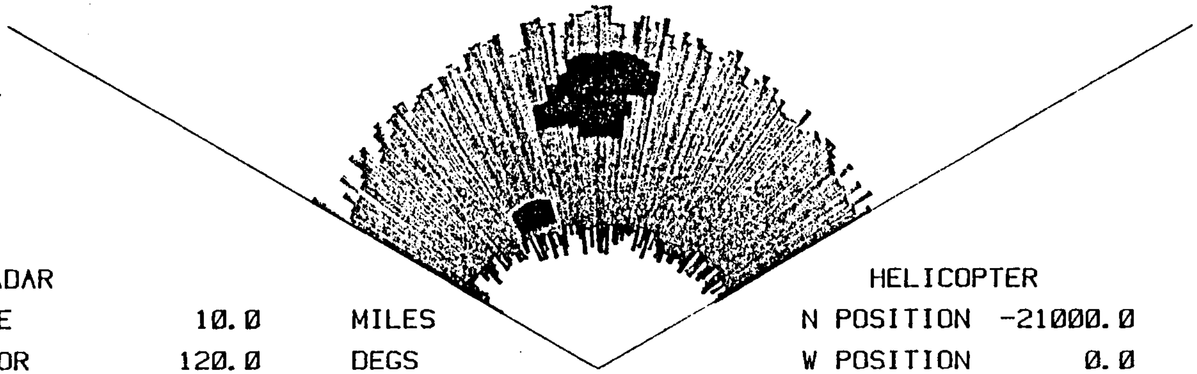
Figure 4.41



RADAR		
RANGE	10.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	RADAR	
TILT	-6.0	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21000.0	FEET
W POSITION	0.0	FEET
HEADING	0.0	DEGS
ALTITUDE	500.0	FEET
WIND DIR.	360.0	DEGS
WIND VEL.	10.0	KNOTS

Figure 4.42



RADAR		
RANGE	10.Ø	MILES
SECTOR	12Ø.Ø	DEGS
RADAR/BCON	RADAR	
TILT	-3.Ø	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21ØØØ.Ø	FEET
W POSITION	Ø.Ø	FEET
HEADING	Ø.Ø	DEGS
ALTITUDE	5ØØ.Ø	FEET
WIND DIR.	36Ø.Ø	DEGS
WIND VEL.	2Ø.Ø	KNOTS

Figure 4.43

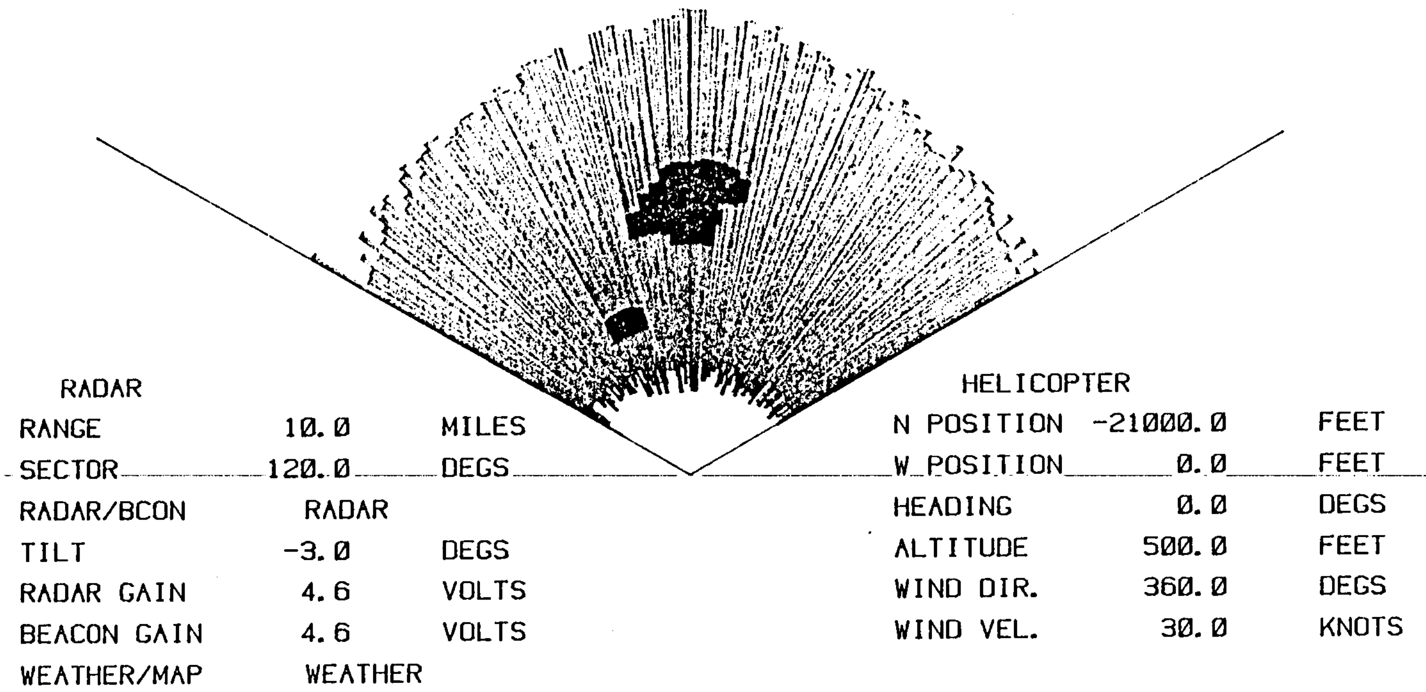
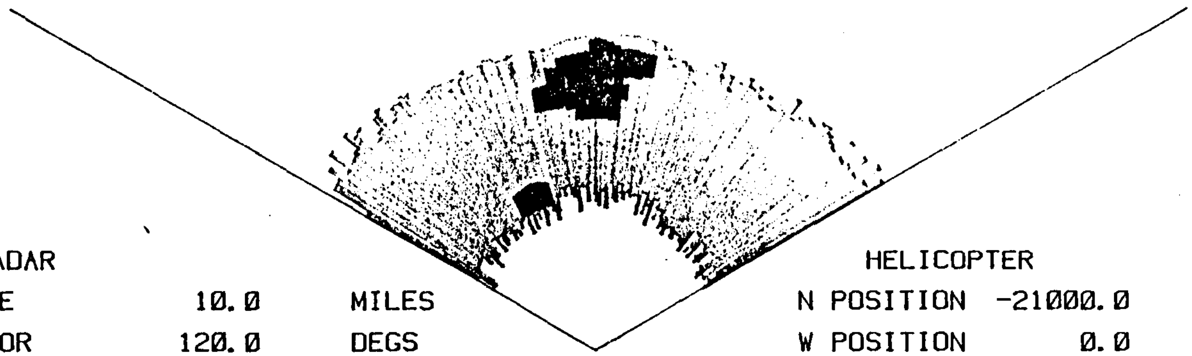

RADAR
RANGE 10.0 MILES
SECTOR 120.0 DEGS
RADAR/BCON RADAR
TILT -3.0 DEGS
RADAR GAIN 4.6 VOLTS
BEACON GAIN 4.6 VOLTS
WEATHER/MAP WEATHER
HELICOPTER
N POSITION -21000.0 FEET
W POSITION 0.0 FEET
HEADING 0.0 DEGS
ALTITUDE 500.0 FEET
WIND DIR. 360.0 DEGS
WIND VEL. 30.0 KNOTS

Figure 4.44



RADAR		
RANGE	10.0	MILES
SECTOR	120.0	DEGS
RADAR/BCON	RADAR	
TILT	-3.0	DEGS
RADAR GAIN	4.6	VOLTS
BEACON GAIN	4.6	VOLTS
WEATHER/MAP	WEATHER	

HELICOPTER		
N POSITION	-21000.0	FEET
W POSITION	0.0	FEET
HEADING	0.0	DEGS
ALTITUDE	500.0	FEET
WIND DIR.	270.0	DEGS
WIND VEL.	20.0	KNOTS

Figure 4.45

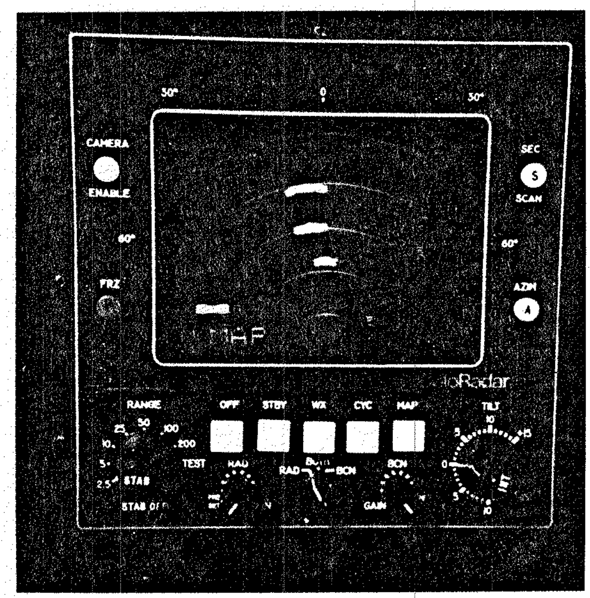


Figure 4.46 Map Mode for Three-Target Situation

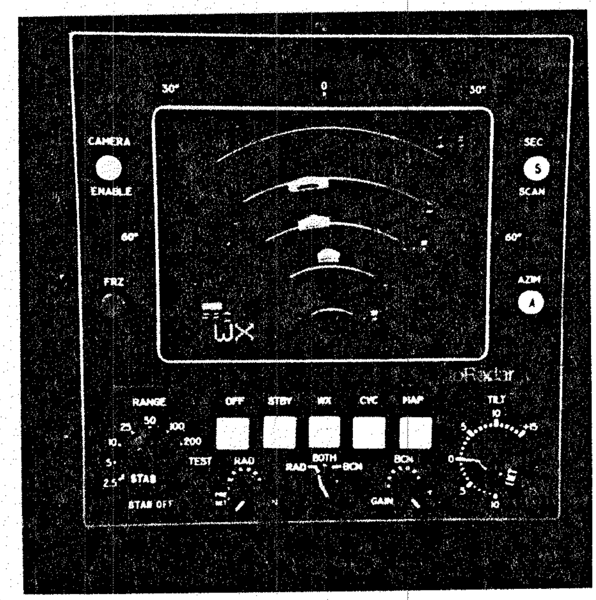


Figure 4.47 Weather Mode for Three-Target Situation

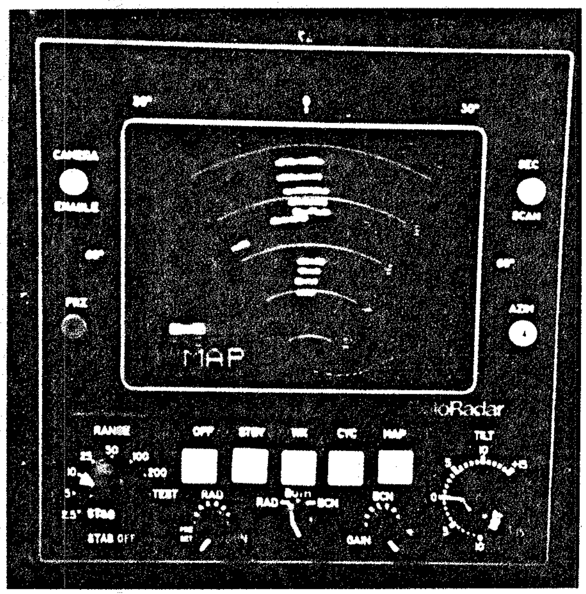


Figure 4.48 Map Mode of Multiple Targets

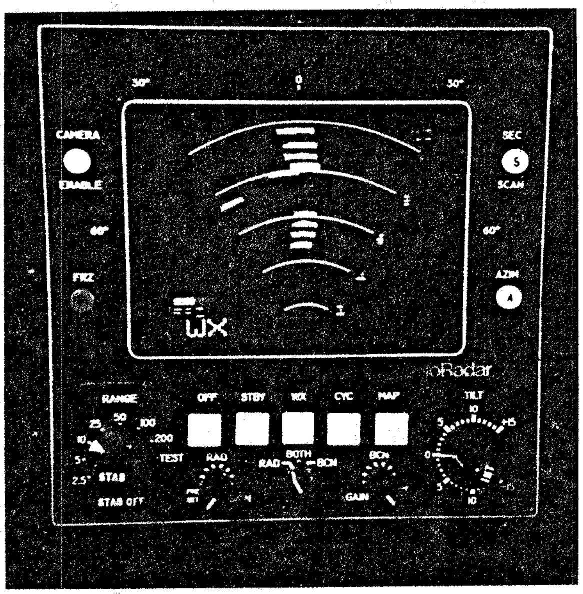


Figure 4.49 Weather Mode of Multiple Targets

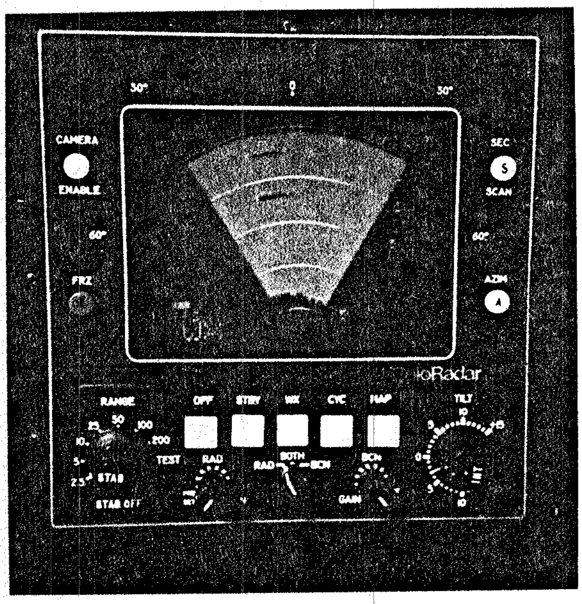


Figure 4.50 Two-and-a-Half Mile Range Under Wind Conditions

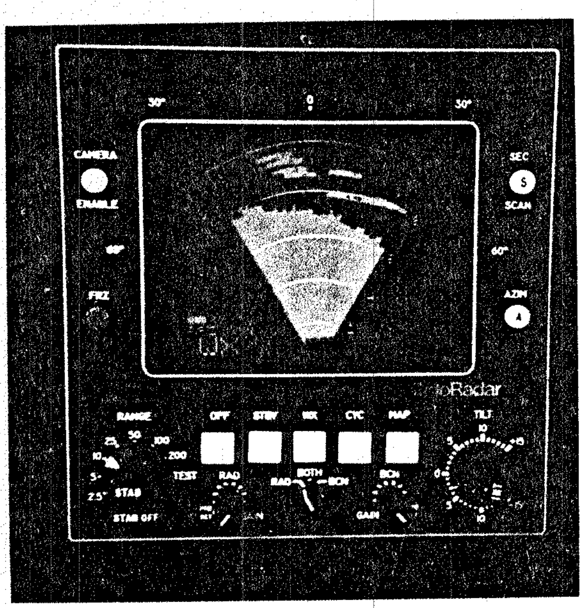


Figure 4.51 Ten Mile Range Under Wind Conditions

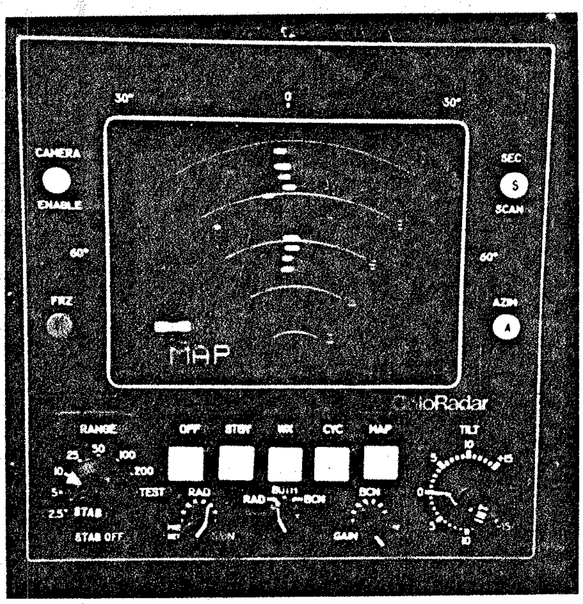


Figure 4.52 Beacon Mode

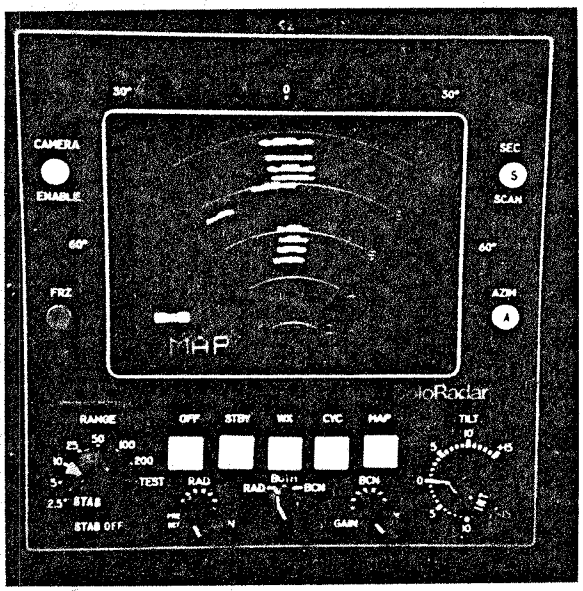


Figure 4.53 Radar Mode

V. CONCLUDING REMARKS

This report documents the software developed to drive a color radar display for overwater helicopter approaches to off-shore oil rig platforms. A variety of off-shore scenarios representing airborne radar approach conditions were defined in a software library and snapshots of the resulting output for interesting situations were documented in the report.

Unique features of the real-time algorithm are (1) a circular list approach to target sorting and ray generation, (2) target shape and multiple target merging effect generation and (3) sea clutter generation.

This Page Intentionally Left Blank

APPENDIX A

HELICOPTER RADAR DISPLAY SIMULATION PROGRAM LISTING

This Page Intentionally Left Blank

SUBROUTINE CREATE

DESCRIPTION:

THIS PROGRAM WAS WRITTEN FOR NASA-AMES TO SIMULATE RADAR RETURNS IN A RCA PRIMUS 500 RADAR SCREEN.

THIS SUBROUTINE CREATES A RADAR VECTOR IN A DIGITAL ENCODED FORM. THIS VECTOR IS IN A FORM COMPATIBLE WITH THE RADAR TARGET SIMULATOR WHICH DRIVES A PRIMUS 500 RADAR SCOPE. THIS PROGRAM CREATES THESE VECTORS FOR SEA MODE ONLY.

CREATE IS CALLED EACH TIME A NEW VECTOR IS TO BE PRODUCED. IT INPUTS DATA FROM THE RADAR HARDWARE TO DETERMINE THE ANTENNA POSITION, DIRECTION OF TRAVEL AND RANGE SETTING. IT ALSO KNOWS THE CURRENT HELICOPTER LOCATION AND HEADING THRU COMMON /XFLOAT/. SUBROUTINE INIRDR WAS CALLED AT INITIALIZATION TIME TO CREATE A LIST OF TARGET DATA WHICH IS MAINTAINED IN A CIRCULAR LIST AND IS STORED IN COMMON /TARGET/. ALSO MAINTAINED IS A SET OF POINTERS INTO THIS TARGET LIST. TGTSTR POINTS THE FIRST TARGET BEHIND THE RADARSWEEP AND NOT VISABLE. TGTEND POINTS TO THE LAST TARGET WHICH IS AHEAD OF THE RADAR SWEEP AND AHEAD OF THE LOOKAHEAD AREA. A LOOKAHEAD AREA IS DEFINED AHEAD OF THE SWEEP TO INSURE THAT ALL TARGETS ABOUT TO BE VISABLE TO THE RADAR HAVE BEEN UPDATED WITH REGARD TO DISTANCE AND HEADING. THESE TARGETS ARE UPDATED BY PRIMARY UPDATE WHENEVER THEY HAVE ENTERED THE LOOKAHEAD AREA. IF ALL TARGETS IN THE LOOKAHEAD AREA HAVE BEEN UPDATED THEN AN ALTERNATE TARGET NOT IN THE RADAR SWEEP IS UPDATED. ONLY ONE TARGET IS UPDATED AT A TIME AND ONCE A TARGET IS VISABLE TO THE RADAR ITS POSITION WILL NOT BE UPDATED UNTIL THE RADAR SWEEP IS PAST IT. A TARGET UPDATE IS PERFORMED BY A CALL TO SUBROUTINE UPDATE. THIS CORRECTS THE HEADING AND DISTANCE INFORMATION AND INSURES THAT THE CIRCULAR LIST IS MAINTAINED IN THE PROPER ORDER.

THE ALGORITHM STARTS BY CALCULATING THE ANTENNA HEADING AND THE HEADING DIFFERNCE OF THE TARGET POINTED TO BY TGTSTR. THIS INITIAL HEADING DIFFERENCE IS ALWAYS NEGATIVE AND HEADING DIFFERENCE IS CORRECTED FOR THE ANTENNA DIRECTION SO THAT A NEGATIVE HEADING DIFFERENCE MEANS THE TARGET IS BEHIND THE ANTENNA SWEEP. THE ALGORITHM PROCESSES EACH TARGET BY TRAVERSING THE CIRCULAR LIST IN ORDER OF ANTENNA DIRECTION. THIS PROCEEDS UNTIL A TARGET IS ENCOUNTERED PAST THE LOOKAHEAD AREA. EACH TARGET PROCESSED THAT IS VISABLE TO THE RADAR (IE. -TGTWDT < HDGDIF < TGTWDT) GENERATES A NUMBER OF INTENSITY TRANSITIONS AT OFFSETS FROM THE TARGET DISTANCE. THESE CHANGES ARE STORED IN A LINKED LIST BY SUBROUTINE INSERT AFTER ALL TARGETS HAVE BEEN PROCESSED THE LIST IS TRAVERSED TO GENERATE A LIST OF INTENSITY CHANGES IN ORDER OF DISTANCE TO BE SENT TO THE RADAR HARDWARE. THIS OUTPUT DATA IS SENT BY A CALL TO SUBROUTINE SRDI.

THIS ENDS THE PROCESSING DONE BY SUBROUTINE CREATE.

```
C       COMMON DEFINITIONS:
C       /TARGET/ INDIVIDUAL TARGET DATA
C               NUMTGT - NUMBER OF TARGETS DEFINED
C               TGTLOC(2,TGTID) - X AND Y LOCATION OF TARGET IN FEET
C                       FROM PRIMARY TARGET
C               TGTHDG(TGTID) - TARGET HEADING RELATIVE TO TRUE NORTH
C                       IN RADIANS
C               TGTDST(TGTID) - TARGET DISTANCE FROM HELICOPTER IN
C                       NAUTICAL MILES
C               TGTPTR(2,TGTID) - INTEGER POINTERS TO THE NEXT
C                       COUNTERCLOCKWISE AND CLOCKWISE TARGETS
C               TGTTYP(TGTID) - TYPE OF TARGET (0 = NON-BEACON, 1 = BEACON)
C               TGTLOG(TGTID) - LOG 10 OF TARGET DISTANCE FROM HELICOPTER
C               TGTTLT(TGTID) - ANGLE OF TILT FROM HELICOPTER TO TARGET
C                       IN RADIANS

C       /STATUS/ STATUS OF POINTERS INTO CIRCULAR LIST OF TARGETS
C               TGTSTR - POINTER TO THE FIRST TARGET BEHIND THE RADAR SWEEP
C                       AND NOT VISABLE
C               TGTEND - POINTER TO THE FIRST TARGET AHEAD OF THE SWEEP
C                       AND BEYOND THE LOOKAHEAD AREA
C               LSTDIR - LAST DIRECTION OF RADAR SWEEP (0=CCW,1=CW)
C               NXTUPD - NEXT PRIMARY TARGET TO BE UPDATED. ALWAYS IN
C                       THE LOOKAHEAD AREA OR EQUAL TO TGTEND
C               NXTALT - NEXT ALTERNATE TARGET TO BE UPDATED. ALWAYS
C                       OUTSIDE OF THE VISABLE AND LOOKAHEAD AREAS. USED
C                       WHENEVER NO PRIMARY TARGET IS READY FOR UPDATING

C       /BINTRE/ LINKED LIST OF INTENSITY CHANGES
C               BINLST(1,NODE) - DISTANCE ALONG VECTOR OF THIS INTENSITY
C                       CHANGE IN RADAR TARGET POSITIONS
C               BINLST(2,NODE) - INTENSITY CHANGES OF LOWEST COLOR
C               BINLST(3,NODE) - INTENSITY CHANGES OF MIDDLE COLOR
C               BINLST(4,NODE) - INTENSITY CHANGES OF HIGHEST COLOR
C               BINLST(5,NODE) - POINTER TO NEXT NODE (LONGER DISTANCE)
C               HDBLST - NODE CLOSEST IN DISTANCE
C               NXTNOD - NEXT NODE AVAILABLE FOR INSERT

C       /ANTENA/ ANTENNA PARAMETERS
C               ANTDIM - DIAMETER OF RADAR ANTENNA IN INCHES
C               RADOME - BEAM SPREADING FACTOR OF ANTENNA
C               ANGAIN - MAXIMUM GAIN OF ANTENNA IN DB'S
C               BRKPNT - DISTANCE AT WHICH STC CURVE CHANGES TO
C                       40 * LOG(DISTANCE) IN MILES
C               ANTLOG - LOG 10 OF ANTDIM

C       /XFLOAT/ SIMULATION DATA
C               A(6) - PSIR - HELICOPTER HEADING IN RADIANS
C               A(103) - XPR - HELICOPTER X LOCATION RELATIVE TO
C                       PRIMARY TARGET (IN FEET)
C               A(104) - YPR - HELICOPTER Y LOCATION RELATIVE TO
C                       PRIMARY TARGET (IN FEET)
C               A(105) - HPR - HELICOPTER ALTITUDE RELATIVE TO
C                       PRIMARY TARGET (IN FEET)
C               A(358) - D2R - DEGREE TO RADIAN CONVERSION
C               A(359) - R2D - RADIAN TO DEGREE CONVERSION
C               A(460) - PI - 3.14159
C               A(461) - TWOPI - 2 * PI
C               A(464) - VWIND - VELOCITY OF WIND IN KNOTS
C               A(466) - PSWR - DIRECTION OF WIND IN RADIANS
```

```
PROGRAM VARIABLE DEFINITIONS
        AHDLEN - COMPUTED LENGTH OF TARGET AHEAD OF ACTUAL DISTANCE
                 IN MILES
        AHDPCT - PERCENT OF TARGET IN FRONT OF ACTUAL POSITION
        ANTERR - ERROR BETWEEN ANTENNA HEADING AND TARGET HEADING
                 IN DEGREES
        ANTGAN - ACTUAL ANTENNA GAIN IN DB CONSIDERING ANTENNA
                 POINTING ERROR
        ANTHDG - HEADING OF ANTENNA IN RADIANS RELATIVE TO TRUE NORTH
        ANTINC - CHANGE IN HEADING (IN RADIANS) FOR EACH
                 ANTENNA POSITION INCREMENT
        ANTLOS - ANTENNA LOSS IN DB DUE TO ANTENNA ERROR
        ANTTBL(25) - TABLE OF ANTENNA LOSSES IN DB DUE TO ANGULAR ERROR
                     ONE ENTRY FOR EACH DEGREE OF ERROR
        ASTLEN - COMPUTED LENGTH OF TARGET BEHIND ACTUAL DISTANCE IN
                 MILES
        ASTPCT - PERCENT OF TARGET BEHIND ACTUAL TARGET POSITION
        ATK    - ANTENNA TILT CONVERSION CONSTANT FROM RADAR
                 UNITS TO DEGREES
        BCNGAN - BEACON GAIN SETTING IN DB. RANGES FROM -37 TO +3 DB.
        BECDEL - BEACON TRANSMITTING DELAY IN MILES
        BGK    - BEACON GAIN CONVERSION CONSTANT FROM RADAR
                 UNITS TO VOLTS
        COEF1(2) - CORRECTION FACTOR FOR TARGET LENGTH CONSIDERING
                   PULSE WIDTH.
                   (1) = 2.35 USEC PULSE WIDTH
                   (2) =  .6  USEC PULSE WIDTH
        COEF2(3) - PERCENT OF TARGET LENGTH FOR THE THREE INTENSITY
                   LEVELS.
                   (1) = PERCENT LENGTH OF LOWEST INTENSITY
                   (2) = PERCENT LENGTH OF INTERMEDIATE INTENSITY
                   (3) = PERCENT LENGTH OF HIGHEST INTENSITY
        CONVRT(6) - ARRAY TO CONVERT DISTANCE (IN MILES) TO ENCODED
                    RADAR DISTANCE FOR EACH RANGE SETTING
        CURTGT - TARGET ID OF CURRENT TARGET BEING PROCESSED
        DB - COMPUTED DB LEVEL FOR EACH INTENSITY LEVEL
        DBLVL - COMPUTED RETURN SIGNAL STRENGTH IN DBM
        DBZERO - TRIGGER LEVEL FOR SEA CLUTTER MODEL
        FIXFCT - CONVERTS FIXED SEA CLUTTER LEVEL TO VALUE USED TO
                 COMPUTE MINRANGE AND MAXRANGE (I.E. CONTROLS
                 CONTROLS CLUTTER LENGTH)
        FT2NM  - FEET TO NAUTICAL MILE CONVERSION FACTOR
        HDGDIF - DIFFERENCE BETWEEN TARGET AND ANTENNA HEADING
                 FOR CURRENT TARGET
        HDGFCT - FACTOR APPLIED TO SEA REFLECTIVITY FOR
                 HEADING REALATIVE TO THE SEA
        I - WORK INDEX
        IANT - ENCODED ANTENNA TILT
        IAWPCI - ENCODED ANALOG WORD RECEIVED FROM RADAR
        IAZM - ENCODED ANTENNA DIRECTION
        IBBR - ENCODED RADAR MODE
               1 = RADAR ONLY
               2 = BEACON ONLY
               3 = RADAR AND BEACON
        IBGN - ENCODED BEACON GAIN
        ICOMP - COMPLETE BIT VALUE
        IDATA(1-100) - ARRAY FOR OUTPUT DATA FOR RADAR HARDWARE
        IDIRF - CURRENT DIRECTION OF RADAR SWEEP (0=CCW, 1=CW)
        IDWPCI - ENCODED DISCRETE WORD RECEIVED FROM RADAR
        IMAPS - WEATHER OR MAP MODE
```

```
          0 = WEATHER
          1 = MAP
INTADD - INTENSITY ADDED FOR STRONGEST TARGET
INTCHG(4) - INTENSITY CHANGES FOR EACH LEVEL
             ALSO USED AS INTENSITY TOTALS FOR VECTOR CREATION
INTENT(4) - ENCODED VALUE FOR ALL 4 INTENSITY LEVELS
INTOUT - CURRENT OUTPUT INTENSITY LEVEL
INTTOT - TOTAL INTENSITY LEVEL
         (0-1) = BLACK
         (2-3) = LOWEST INTENSITY
         (3-6) = INTERMEDIATE INTENSITY
         (7- ) = HIGHEST INTENSITY
IPGR - PRESET RADAR INTENSITY IF = 1
IPULSE - PULSE WIDTH
         1 = 2.35 US
         2 =  .6  US
IRGN - ENCODED RADAR GAIN SETTING
IRGSET - CURRENT RANGE SETTING
         (1-6) = (2.5, 5.0, 10.0, 25.0, 50.0, 100.0) MILES
IRNG1  - RANGE AT WHICH INTENSITY LEVEL RISES (ENCODED)
IRNG2  - RANGE AT WHICH INTENSITY LEVEL DROPS (ENCODED)
IRX    - SEED FOR RANDOM NUMBER GENERATOR
IRY    - SEED FOR RANDOM NUMBER GENERATOR
LOKAHD - SIZE OF LOOKAHEAD AREA (IN RADIANS FROM CURRENT
         ANTENNA HEADING)
LOOKUP - INDEX INTO ANTENNA TABLE
LSTINT - LAST INTENSITY LEVEL OUTPUT
MANT   - MASK FOR ANTENNA TILT (ANALOG WORD)
MAX - MAXIMUM ENCODED RANGE FOR CURRENT RANGE SETTING
MAXINT - MAXIMUM INTENSITY LEVEL FOR CURRENT RADAR MODE
MAXRNG(6) - MAXIMUM ENCODED RANGE FOR EACH RANGE SETTING
MBBR   - MASK FOR RADAR/BOTH/BEACON MODE (DIGITAL WORD)
MBGN   - MASK FOR BEACON GAIN (ANALOG WORD)
MDIRF  - MASK FOR ANTENNA DIRECTION (DIGITAL WORD)
MDS(3,2) - MINIMUM DETECTABLE SIGNAL IN DB FOR 3 INTENSITY
           LEVELS AND 2 PULSE WIDTHS
MIN - MINIMUM ENCODED RANGE FOR CURRENT RANGE SETTING
MINRNG(6) - MINIMUM ENCODED RANGE FOR EACH RANGE SETTING
MMAPS - MASK FOR MAP/WEATHER MODE (DIGITAL WORD)
MPGR  - MASK FOR PRESET GAIN (DIGITAL WORD)
MRGN  - MASK FOR RADAR GAIN (ANALOG WORD)
MRGSET - MASK FOR RANGE SETTING (DIGITAL WORD)
MSECT - MASK FOR SECTOR TYPE 60/120 DEGS (DIGITAL WORD)
NODE - CURRENT NODE FOR LINKED LIST TRAVERSAL
NPT - NUMBER OF OUTPUT WORDS FOR SRDO
NXTPOS - NEXT ANTENNA POSITION AZIMUTH
NXTTGT - NEXT TARGET IN DIRECTION OF RADAR SWEEP
         FROM CURRENT TARGET
OHDDIF - HEADING DIFFERENCE OF LAST TARGET PROCESSED
PCT - PERCENT OF TARGET LENGTH CONSIDERING DB LEVEL AND
      INTENSITY LEVEL
PCTDB - FACTOR MULTIPLIED BY DB TO INCREASE TARGET LENGTH
PCTMIN - MINIMUM TARGET LENGTH (AT 0 DB) IN PERCENT
PRSEA  - RADAR POWER RECEIVED FROM THE SEA
PULSE  - RADAR PULSE WIDTH IN USEC
RANFIX - FACTOR FOR RANDOM EFFECT OF SEA CLUTTER
RANGE1 - RANGE AT WHICH INTENSITY LEVEL INCREASES (IN MILES)
RANGE2 - RANGE AT WHICH INTENSITY LEVEL DECREASES (IN MILES)
RANGES(6) - MAXIMUM RANGE (IN MILES) FOR EACH RANGE SETTING
RANGR - CURRENT RANGE OF TARGET (OR SEA CLUTTER) IN MILES
RDRGAN - RADAR GAIN IN DB (FROM -37 TO +3)
```

```
RGK     - RADAR GAIN CONVERSION CONSTANT FROM RADAR
          UNITS TO VOLTS
RHDG    - ANTENNA DIRECTION RELATIVE TO THE SEA
          (0 = HEAD SEA, 1 = DOWNSEA)
RNCOEF  - TOTAL RANDOM EFFECT OF SEA CLUTTER
RNGLOG  - LOG 10 OF CURRENT TARGET RANGE
SEADB   - SEA RETURN POWER RELATIVE TO LEVEL 1
          THRESHOLD VALUE
SEAFCT  - FACTOR WHICH MULTIPLIES WIND SPEED TO
          INCREASE SEA REFLECTIVITY IN HIGHER SEA STATES
SFACT   - ANTENNA HEADING CONVERSION CONSTANT TO CHANGE
          RADAR UNITS TO DEGREES
SIGSEA - RADAR REFLECTIVITY OF SEA PER UNIT AREA
STC - STC LEVEL FOR CURRENT INTENSITY LEVEL
STCLVL - STC LEVEL OF HIGHEST INTENSITY SETTING
STCORG(3) - OFFSETS ADDED TO STCLVL TO OBTAIN STC LEVEL FOR
            EACH INTENSITY LEVEL
STHDDF - HEADING DIFFERENCE OF STARTING TARGET ID (TGTSTR)
TGTLEN(6) - LENGTH OF TARGET FOR EACH RABGE SETTING (MILES)
TGTWDT - HALF OF THE WIDTH OF TARGET (RADIANS)
TILT - RADAR ANTENNA TILT (IN RADIANS)
TLTDIF - TILT ERROR BETWEEN TARGET AND ANTENNA (IN RADIANS)
TLTFCT - FACTOR MULTIPLIED BY TILT FOR SEA CLUTTER MODEL
UPDTGT - TARGET TO BE UPDATED NEXT
WNCOEF - FACTOR MUTIPLIED BY WIND FOR SEA CLUTTER MODEL
WNDB - TOTAL FACTOR OF WIND, TILT AND WIND DIRECTION USED
       FOR SEA CLUTTER MODEL
XANT - ANTENNA TILT IN DEGREES
XBGN - BEACON GAIN IN VOLTS
XRGN - RADAR GAIN IN VOLTS
```

NAME = S:CREATX.HELRAD

```
C*********************************************************************
C*********************************************************************
C*****            TITLE...            S:CREATX, HELRAD          ******
C*****          LAST MØDIFIED:    8 APRIL, '82: VDBEN           ******
C*****          TECHNICAL MØNITØR: G. CLARY                     ******
C*********************************************************************
C*********************************************************************
C***
C***  THIS SUBRØUTINE CREATES THE SIMULATED RADAR DATA FØR ØUT-
C***   PUT TØ A MØDIFIED RADAR DISPLAY, RCA MØDEL PRIMUS 500...
C***
C*********************************************************************
C***
C***   SUBRØUTINES CALLED:
C***
C***      SRDI    - GETS THE ANALØG AND DIGITAL DATA FRØM THE
C***                RADAR DISPLAY UNIT...
C***      INSERT  - SETS THE INTENSITY LEVEL (CØLØR) CHANGES IN
C***                THE BINARY TREE...
C***      SCIUPD  - UPDATES THE TARGETS' HEADING AND DISTANCE...
C***      RANDUM  - RETURNS A FLØATING-PØINT RANDØM NUMBER BASED
C***                ØN THE PØWER RESIDUE METHØD (IBM 020-3011)...
C***      SRDØ    - ØUTPUTS THE VECTØR DATA TØ THE RADAR UNIT FØR
C***                DISPLAY...
C***
C*********************************************************************
C
      SUBRØUTINE CREATE
C
C*********************************************************************
C*****                       CØMMØN BLØCKS...                   ******
C*********************************************************************
C
      CØMMØN
     *     /XFLØAT/ A(500)/IFIXED/IA(250)
     *     /HRCØM /RH( 50),IH(50),KC( 25)
      CØMMØN
     *     /BINTRE/BINLST(5,150), HDBLST, NXTNØD
```

```
     *     /TARGET/TGTLØC(2,100), TGTDST(100), TGTHDG(100),
     *             TGTPTR(2,100), TGTLØG(100), TGTTLT(100),
     *                                         TGTTYP(100),
     *             NBRTGT, NUMTGT,  TGTID
     *     /STATUS/LSTDIR, NXTALT, NXTUPD, TGTEND, TGTSTR, LAZM
     *     /ANTENA/ANTDIM, RADØME, ANGAIN, BRKPNT, RNGBRK, ANTLØG
C
C*****************************************************************
C*****                    EQUIVALENCES...                   ******
C*****************************************************************
C
      EQUIVALENCE
     *     ( A(  6),  PSIR), ( A(105),   HPR), ( A(358),   D2R),
     *     ( A(359),   R2D), ( A(460),    PI), ( A(461), TWØPI),
     *     ( A(464), VWIND),                   ( A(466),  PSWR)
      EQUIVALENCE
     *     (RH( 1),  XANT), (RH( 2),  XBGN), (RH( 3),  XRGN),
     *                      (RH( 5), XMAPS),
     *     (RH( 7),  XBBR), (RH( 8),XRANGE),
     *                                       (RH(15),  XAZM),
     *     (RH(31),   ATK), (RH(32),   BGK), (RH(33),   RGK),
     *                      (RH(35), SFACT)
      EQUIVALENCE
     *     (IH( 1),  IANT), (IH( 2),  IBGN), (IH( 3),  IRGN),
     *     (IH( 4), ICAMS), (IH( 5), IMAPS), (IH( 6),  IPGR),
     *     (IH( 7),  IBBR), (IH( 8),IRGSET), (IH( 9),  IAZF),
     *     (IH(10),IRNGCF), (IH(11),IUNRUN), (IH(12),ISYNCE),
     *     (IH(13), ISECT), (IH(14), IDIRF), (IH(15),  IAZM),
     *     (IH(31),  MANT), (IH(32),  MBGN), (IH(33),  MRGN),
     *     (IH(34), MCAMS), (IH(35), MMAPS), (IH(36),  MPGR),
     *     (IH(37),  MBBR), (IH(38),MRGSET), (IH(39),  MAZF),
     *     (IH(40),MRNGCF), (IH(41),MUNRUN), (IH(42),MSYNCE),
     *     (IH(43), MSECT), (IH(44), MDIRF), (IH(45),  MAZM)
      EQUIVALENCE
     *     (KC( 1),IAWPCI), (KC( 2),IDWPCI), (KC( 6), ISCAN),
     *     (KC(20),ANTHDG)
C
C*****************************************************************
C*****                    DECLARATIØNS...                   ******
C*****************************************************************
C
```

```
      REAL     ANTTBL(25), CBNVRT( 6), RANGES( 6),  MDS(3,2),
     *          CBEF1( 2),  CBEF2( 3), STCBRG( 3), TGTLEN(6),
     *         LBKAHD
C    *
      INTEGER     INTENT(4),  MINRNG(6),  MAXRNG(6), IDATA(100),
     *            INTCHG(4)
C    *
      INTEGER     BINLST, CURTGT, HDBLST, TGTEND, TGTID, TGTSTR,
     *            TGTPTR, TGTTYP, UPDTGT
C
C*********************************************************************
C*****                       DATA STATEMENTS...                ******
C*********************************************************************
C
      DATA       ANTTBL/0.00, 0.20, 0.90, 2.00, 3.30, 5.50, 7.20,
     *                  11.0, 14.8, 19.0, 24.0, 29.0, 36.0, 45.0,
     *                  34.5, 29.0, 26.7, 26.5, 27.0, 29.0, 31.0,
     *                  32.8, 34.0, 35.0, 37.0/
C    *
      DATA         ATK/0.1176470/,  BGK,  RGK/2*0.01961/
     *            SFACT/0.2343750/
C    *
      DATA       ANTINC/4.0906E-3/, LBKAHD/0.30/, TGTWDT/0.230/
C    *
      DATA      CBNVRT/124.0, 123.8, 123.7, 61.84, 30.92, 15.46/,
     *          ANGAIN/31.5/, ANTDIM/18.0/,
     *          FT2NM/1.6447368E-04/,
     *          RADBME/1.03/, TRLBSS/3.60/,
     *          INTENT/12288, 8192,0,4096/, ICBMP/4Z7003/
C    *
      DATA       PCTMIN/0.6/, PCTDB/0.015/, ZGWN/0.5/
C    *
      DATA       UPDTGT/1/, IRX/137462873/, IRY/1073741823/
C    *
C*****   BIT MASKS FBR THE RADAR 'ANALBG'  STATUS WBRD...
C    *
      DATA         MANT/6ZFF0000/, MBGN/4ZFF00/, MRGN/2ZFF/
C    *
C*****   BIT MASKS FBR THE RADAR 'DIGITAL' STATUS WBRD...
C    *
      DATA      MCAMS /6Z800000/, MMAPS /6Z400000/,
```

```
     *          MPGR  /6Z200000/, MBBR  /6Z180000/,
     *          MRGSET/6Z070000/, MAZF  /6Z008000/,
     *          MRNGCF/6Z004000/, MUNRUN/6Z002000/,
     *          MSYNCE/6Z001000/, MSECT /6Z000800/,
     *          MDIRF /6Z000400/, MAZM  /6Z0003FF/
C    *
      DATA      MDS/-109.0, -106.0, -104.0, -96.0, -93.0, -91.0/,
     *        BECDEL/0.5/, CØEF1/2.2,1.0/, CØEF2/1.0,0.65,0.25/
C    *
      DATA    MAXRNG/310,619,1237,1546,1546,1546/,
     *        MINRNG/16,16,16,8,4,2/
C    *
      DATA    TGTLEN/0.15, 0.15, 0.30, 0.30, 0.30, 1.60/
     *        AHDPCT/0.15/, ASTPCT/0.85/
     *        STCØRG/-17.0,-7.0,0.0/
C    *
       DATA   FIXFCT/0.15/, RANFIX/0.2/, TLTFCT/1.0/,
     *        HDGFCT/10.0/,SEAFCT/0.15/,
     *        RANGES/2.500, 5.000, 10.00, 25.00, 50.00, 100.0/
C
C*****************************************************************
C*****************************************************************
C
C*****   GET THE ANALØG AND DIGITAL RADAR STATUS WØRDS...
C
      CALL SRDI(IAWPCI,IDWPCI)
C
C*****   PREVENT INVALID STATUS WØRDS...
C
      IF (IAWPCI .EQ. 0)  IAWPCI = 1338212
      IF (IDWPCI .EQ. 0)  IDWPCI =  533502
C
                YAZM = IAZM = IAND(MAZM,IDWPCI)
                XAZM = YAZM*SFACT
C
C***   MASK ØUT THE DATA FRØM THE RADAR ANALØG WØRD...
C
                YANT = IANT = ISL(IAND(MANT ,IAWPCI),-16) - 6
                YBGN = IBGN = ISL(IAND(MBGN ,IAWPCI), -8)
                YRGN = IRGN =     IAND(MRGN ,IAWPCI)
C
```

```
C***   COMPUTE THE ANTENNA TILT ANGLE (DEGREES) AND THE BEACON
C***     AND RADAR GAINS (VOLTS)...
C
             XANT = YANT*ATK - 15.0
             XBGN = YBGN*BGK
             XRGN = 5.0 - YRGN*RGK
C
C***   MASK OUT THE DATA FROM THE RADAR DIGITAL WORD...
C
X           ICAMS  = ISL(IAND(MCAMS ,IDWPCI),-23)
            IMAPS  = ISL(IAND(MMAPS ,IDWPCI),-22)
            IPGR   = ISL(IAND(MPGR  ,IDWPCI),-21)
            IBBR   = ISL(IAND(MBBR  ,IDWPCI),-19)
            IRGSET = ISL(IAND(MRGSET,IDWPCI),-16) + 1
X           IAZF   = ISL(IAND(MAZF  ,IDWPCI),-15)
X           IRNGCF = ISL(IAND(MRNGCF,IDWPCI),-14)
X           IUNRUN = ISL(IAND(MUNRUN,IDWPCI),-13)
X           ISYNCE = ISL(IAND(MSYNCE,IDWPCI),-12)
            ISECT  = ISL(IAND(MSECT ,IDWPCI),-11)
            IDIRF  = ISL(IAND(MDIRF ,IDWPCI),-10)
C
            XMAPS  = FLOAT(IMAPS)
            XBBR   = FLOAT(IBBR)
            XRANGE = FLOAT(IRGSET)
C
C*****   COMPUTE THE ANTENNA TILT, AND THE BEACON AND RADAR GAIN.
C
              TILT = XANT*D2R
            BCNGAN = XBGN*8.0 - 37.0
            RDRGAN = XRGN*9.0 - 42.0
C
          IF (IPGR   .EQ. 1)  RDRGAN = 0.0
          IF (IRGSET .GT. 6)  IRGSET = 6
C
            IPULSE = 1
          IF (IMAPS .EQ. 1 .AND. IRGSET .LE. 4 .AND. IBBR .EQ. 1)
     *      IPULSE = 2
C
            AHDLEN = TGTLEN(IRGSET)*COEF1(IPULSE)*AHDPCT
            ASTLEN = TGTLEN(IRGSET)*COEF1(IPULSE)*ASTPCT
C
```

```
                 MAX = MAXRNG(IRGSET)
                 MIN = MINRNG(IRGSET)
C
C*****   INITIALIZE THE PØINTERS AND THE BINARY TREE...
C
                 HDBLST = UPDTGT = 0
                 NXTNØD = 1
                 NXTTGT = TGTPTR(IDIRF + 1, TGTSTR)
C
             IF (LSTDIR .EQ. IDIRF) GØ TØ 8
C
                 TGTSTR = TGTEND
                 NXTTGT = TGTPTR(IDIRF + 1, TGTSTR)
                 TGTEND = NXTTGT
C
             IF (NXTUPD .EQ. TGTSTR)  NXTUPD = NXTTGT
C
      8          CURTGT = TGTSTR
C
C*****   CØMPUTE THE ANTENNA HEADING...
C
             IF (ABS(IAZM - LAZM) .GT. 50)  LSECT = ISECT
                                             LAZM = IAZM
                                           NXTPØS = IAZM + 1
             IF (IDIRF .EQ. 0)             NXTPØS = IAZM - 1
                                            ISCAN = 255
             IF (LSECT .EQ. 1)              ISCAN = 127
C
                 ANTHDG = PSIR + ANTINC*(NXTPØS - ISCAN)
C
      2      IF (ANTHDG .LT. TWØPI)         GØ TØ 4
                 ANTHDG = ANTHDG - TWØPI;   GØ TØ 2
      4      IF (ANTHDG .GE. 0.0)           GØ TØ 6
                 ANTHDG = ANTHDG + TWØPI;   GØ TØ 4
C
C*****   CØMPUTE INITIAL HEADING DIFFERENCE LESS THAN ZERØ...
C
      6          ØHDDIF = TGTHDG(CURTGT) - ANTHDG
C
             IF (IDIRF  .EQ. 0)          ØHDDIF = -ØHDDIF
             IF (ØHDDIF .GT. -TGTWDT)  ØHDDIF =  ØHDDIF - TWØPI
```

```
      C
      C*****   STØRE THE HEADING DIFFERENCE & GET THE FIRST TARGET...
      C
                     STHDDF = ØHDDIF
                     CURTGT = NXTTGT
      C
      C*****   CØMPUTE THE TARGET HEADING DIFFERENCE...
      C
           10        HDGDIF = TGTHDG(CURTGT) - ANTHDG
      C
                 IF (IDIRF  .EQ. 0)        HDGDIF = -HDGDIF
                 IF (HDGDIF .GT. -TGTWDT) HDGDIF =  HDGDIF - TWØPI
           11    IF (HDGDIF .GE. ØHDDIF)        GØ TØ 12
                     HDGDIF = HDGDIF + TWØPI;  GØ TØ 11
           12        ØHDDIF = HDGDIF
      C
      C*****   CHECK HEADING DIFFERENCES...
      C
                     NXTTGT = TGTPTR(IDIRF + 1, CURTGT)
      C
                 IF (HDGDIF .GT. LØKAHD)  GØ TØ 45
                 IF (NXTALT .EQ. CURTGT)  NXTALT = NXTTGT
                 IF (HDGDIF .GE. TGTWDT)  GØ TØ 35
      C
                 IF((TGTEND .EQ. CURTGT) .AND. (LSTDIR .NE. IDIRF))
              *      TGTEND = NXTTGT
                 IF((NXTUPD .EQ. CURTGT) .AND. (NXTUPD .NE. TGTEND))
              *      NXTUPD = NXTTGT
                 IF (HDGDIF .GT.-TGTWDT)  GØ TØ 20
      C
      C*****   TARGET IS BEHIND THE RADAR (NØT VISIBLE)...
      C
                     TGTSTR = CURTGT
                     STHDDF = HDGDIF
                     GØ TØ 40
      C
      C*****   TARGET MAY BE VISIBLE;  CHECK THE ANTENNA TILT...
      C
           20        TLTDIF = TILT - TGTTLT(CURTGT)
                      RANGR = TGTDST(CURTGT)
                     RNGLØG = TGTLØG(CURTGT)
```

```
                 ANTERR =   (ABS(HDGDIF) + ABS(TLTDIF))/D2R
                 ANTERR = (ANTERR/RADOME * ANTDIM/12.0)
                 LOOKUP =   IFIX(ANTERR) + 1
C
           IF (LOOKUP .GT. 25)  GO TO 40
C
                 ANTLOS = ANTTBL(LOOKUP)
                 ANTGAN = ANGAIN - ANTLOS
C
C*****************************************************************
C***
C***    RADAR TARGETS...
C***
C***                              RETURN POWER (DB) = TRANSMIT POWER -
C***     30*LOG10(4*PI) - 40*LOG10(RANGE IN METERS) + 2*(ANTENNA
C***               GAIN) + 20*LOG10(LAMBDA) + 10*LOG10(SIGMA)
C***                        - T/R LOSS...
C***    WHERE:
C***
C***     TRANSMIT POWER = 7000 WATTS = 68.45 DECIBELS...
C***       ANTENNA GAIN = 6.4 + 20LOG10HHHHH*LOG10(ANTDIM) - POINTING
C***             LAMBDA = 0.0319 METERS...                    LOSS
C***              SIGMA = 1000 SQUARE METERS REFLECTIVITY...
C***           T/R LOSS = TWO-WAY LOSS (USER INPUT)...
C***
C***    THIS REDUCES TO:
C***
C***      DBLVL = -95.13 - 40*LOG10(RANGE IN NAUTICAL MILES)
C***                        +  2*(ANTENNA GAIN)...
C***
C*****************************************************************
C
           IF (IBBR .EQ. 2)  GO TO 25
C
         DBLVL = -95.13 + 2.0*ANTGAN - 40.0*RNGLOG - TRLOSS
        STCLVL = -85.40 +      ANGAIN - 23.0*RNGLOG - RDRGAN
C
           IF (RANGR .GE. RNGBRK)
      *STCLVL = -87.80 + 2.0000*ANGAIN - 39.86*RNGLOG - RDRGAN
C
           IF (IMAPS .EQ. 1)
```

```
     *STCLVL = -92.33 + 1.3333*ANGAIN - 29.90*RNGLØG - RDRGAN
C
                          MAXINT = 3
        IF (IBBR .EQ. 3)  MAXINT = 2
C
       DØ 25, J = 1,MAXINT
                                     STC = STCLVL + STCØRG(J) .
          IF (STC .LT. MDS(J,IPULSE))  STC = MDS(J,IPULSE)
                                      DB = DBLVL - STC
          IF ( DB .LT. 0.0)  GØ TØ 25
C
                   PCT =  (PCTDB*DB + PCTMIN)*CØEF2(J)
                RANGE1 =       RANGR - PCT*AHDLEN
                 IRNG1 = IFIX(RANGE1*CØNVRT(IRGSET))
C
          IF (IRNG1 .GE. MAX)  GØ TØ 25
          IF (IRNG1 .LT. MIN)  IRNG1 = MIN
C
                RANGE2 = PCT*ASTLEN + RANGR
                 IRNG2 = IFIX(RANGE2*CØNVRT(IRGSET))
C
          IF (IRNG2 .LT. MIN)  GØ TØ 25
          IF (IRNG2 .GE. MAX)  IRNG2 = MAX - 1
C
            INTCHG(2)    = INTCHG(3) = INTCHG(4) = 0
            INTCHG(J)    =-1
            INTCHG(J+1)  = 1
C
          CALL INSERT(IRNG1,IRNG2,INTCHG)
C
    25 CØNTINUE
C
C**********************************************************************
C***
C***   BEACØN TARGETS...
C***
C***                            RETURN PØWER (DB) = TRANSMIT PØWER -
C***         20*LØG10(4*PI) - 20*LØG10(RANGE IN METERS) +
C***           ANTENNA GAIN + 20*LØG10(LAMBDA) - ØNE-WAY LØSS
C***                          + BEACØN ANTENNA GAIN...
C***   WHERE;
```

```
C***
C***         TRANSMIT POWER = 400 WATTS = 56.02 DBM...
C***           ANTENNA GAIN = 6.4 + 20*LOG10(ANTDIM) - POINTING LOSS
C***                 LAMBDA = 0.0322 METERS (BEACON RETURN PULSE
C***           ONE-WAY LOSS = 0.5* T/R LOSS              FREQUENCY)...
C***   BEACON ANTENNA GAIN = 5 DECIBELS...
C***
C***    THIS REDUCES TO:
C***
C***        DBLVL = -56.16 - 20*LOG10(RANGE IN NAUTICAL MILES)
C***                       + ANTENNA GAIN - 0.5*(T/R LOSS))...
C***       STCLVL = -51.00 - 20*RNGLOG - BCNGAN
C***           DB =  DBLVL - STCLVL
C***           DB =  -5.16 + ANTGAN - 0.5*(T/R LOSS)) + BCNGAN
C***
C*************************************************************************
C
            IF ((IBBR .EQ. 1) .OR. (TGTTYP(CURTGT) .EQ. 0))
     *                       GO TO 40
C
               DB = -5.16 + ANTGAN - 0.5*TRLOSS + BCNGAN
C
            IF (DB .LT. 0.0)  GO TO 40
C
             RANGE1 = RANGR + BECDEL - AHDLEN
              IRNG1 = IFIX(RANGE1*CONVRT(IRGSET))
C
            IF (IRNG1 .GE. MAX)  GO TO 40
            IF (IRNG1 .LT. MIN)  IRNG1 = MIN
C
             RANGE2 = RANGR + BECDEL + ASTLEN
              IRNG2 = IFIX(RANGE2*CONVRT(IRGSET))
C
            IF (IRNG2 .LT. MIN)  GO TO 40
            IF (IRNG2 .GE. MAX)  IRNG2 = MAX - 1
C
          INTCHG(2) = INTCHG(3) = 0
          INTCHG(4) = 1
C
          CALL INSERT(IRNG1,IRNG2,INTCHG)
C
```

```
C*****   TARGET IS IN THE LOOK-AHEAD AREA...
C
    35   IF (TGTEND .EQ. CURTGT)  TGTEND = NXTTGT
         IF((NXTUPD .NE. CURTGT) .OR. (UPDTGT .NE. 0))
     *      GO TO 40
C
          UPDTGT = CURTGT
          NXTUPD = NXTTGT
C
C*****    GET THE NEXT TARGET...
C
    40    CURTGT = NXTTGT
             GO TO 10
C
C*****    TARGET IS IN FRONT OF THE LOOK-AHEAD AREA...
C
    45    LSTDIR = IDIRF
C
         IF (CURTGT .EQ. TGTEND)  GO TO 50
         IF (NXTUPD .EQ. TGTEND)  NXTUPD = CURTGT
C
          TGTEND = CURTGT
C
C*****    FIND AN ALTERNATE TARGET...
C
    50   IF (UPDTGT .NE. 0)  GO TO 55
C
         IF((TGTEND .EQ. TGTPTR(IDIRF + 1, TGTSTR)) .AND.
     *      (STHDDF .LE. LOKAHD - TWOPI))  GO TO 57
C
          UPDTGT = NXTALT
          NXTALT = TGTPTR(2, UPDTGT)
C
         IF((IDIRF .EQ. 1) .AND. (UPDTGT .EQ. TGTSTR))
     *    NXTALT = TGTEND
         IF((IDIRF .EQ. 0) .AND. (UPDTGT .EQ. TGTEND))
     *    NXTALT = TGTSTR
         IF (UPDTGT .EQ. TGTSTR) TGTSTR = TGTPTR(2 - IDIRF,UPDTGT)
         IF (UPDTGT .EQ. TGTEND) TGTEND = TGTPTR(1 + IDIRF,UPDTGT)
C
C*****    UPDATE THE TARGET...
```

```
C
   55     TGTID = UPDTGT
C
      CALL UPDATE
C
C*************************************************************************
C*****
C*****   SEA CLUTTER MODEL (IF 'TILT' IS GREATER THAN + 2.0
C*****                DEGREES, NO SEA CLUTTER IS DISPLAYED)...
C
   57    IF ((TILT .GT. 0.04) .OR. (IBBR .EQ. 2))  GO TO 60
C
                                     TILTD = TILT*R2D
         IF (TILTD .GT. -1.0)        TILTD = -1.0
C
          RANGR = (-HPR*FT2NM)/(TILTD*D2R)
         RNGLOG = LOG10(RANGR)
         STCLVL = -85.4 + ANGAIN - 23.0*RNGLOG - RDRGAN
C
         IF (RANGR .GE. RNGBRK)
     *    STCLVL = -87.80 + 2.0000*ANGAIN - 39.86*RNGLOG
     *                    - RDRGAN
         IF (IMAPS .EQ. 1)
     *    STCLVL = -92.33 + 1.3333*ANGAIN - 29.90*RNGLOG
     *                    - RDRGAN
                                             STC = STCLVL + STCSRG(1)
         IF (STC .LT. MDS(1, IPULSE))  STC = MDS(1,IPULSE)
C
                                  PULSE = 2.35
         IF (IPULSE .EQ.  2)  PULSE = 0.60
C
         IF (ANTHDG .GT. PI) ANTHDG = ANTHDG - TWOPI
                             PSWNEW = PSWR
         IF (PSWNEW .GT.PI)  PSWNEW = PSWNEW - TWOPI
C
                                   RHDG = ABS(ANTHDG - PSWNEW)/PI
         IF (VWIND  .LE. 5.0)  RHDG = 0.0
C
          SIGSEA = -40.00 - TLTFCT*TILTD - HDGFCT*RHDG + SEAFCT*VWIN
          PWRSEA = -66.72 +   2.0*ANGAIN +  10.0*(LOG10(PULSE))
     *                    -  30.0*RNGLOG - TRLOSS - 10.0*ANTLOG
```

```
     *                              +  SIGSEA
        SEADB = PWRSEA -  STC
C
      IF (SEADB .LE. 0.0)  GØ TØ 60
C
         WNDB =  SEADB*FIXFCT
       RNCØEF = (RANFIX + RANDB*WNDB)*RANGR
C
C***   GET A SAMPLE ØF UNIFØRMLY DISTRIBUTED WHITE NØISE...
C
      CALL RANDUM(IRX,IRY,ZGWN)
C
       RANGE1 = RANGR/(1.0 + WNDB) + (ZGWN - 0.5)*RNCØEF
C
C***   GET A SAMPLE ØF UNIFØRMLY DISTRIBUTED WHITE NØISE...
C
      CALL RANDUM(IRX,IRY,ZGWN)
C
       RANGE2 = RANGR*(1.0 + WNDB) + (ZGWN - 0.5)*RNCØEF
C
      IF (RANGE1 .GT. RANGE2)  GØ TØ 60
C
        IRNG1 = RANGE1*CØNVRT(IRGSET)
        IRNG2 = RANGE2*CØNVRT(IRGSET)
C
      IF (IRNG1 .GE. MAX)  GØ TØ 60
      IF (IRNG1 .LT. MIN)  IRNG1 = MIN
      IF (IRNG2 .LT. MIN)  GØ TØ 60
      IF (IRNG2 .GE. MAX)  IRNG2 = MAX - 1
C
      INTCHG(2) = 1
      INTCHG(3) = 0
      INTCHG(4) = 0
C
      CALL INSERT(IRNG1,IRNG2,INTCHG)
C
C*****  CREATE THE VECTØR...
C
   60    NØDE     = HDBLST
       LSTINT     = NPT = 1
       IDATA(1)   = INTENT(LSTINT)
```

```
      INTCHG(1) = INTCHG(2) = INTCHG(3) = 0
      GO TO 75
C
C*****   CREATE THE VECTOR...
C
   70 INTCHG(1) = INTCHG(1) + BINLST(2,NODE)
      INTCHG(2) = INTCHG(2) + BINLST(3,NODE)
      INTCHG(3) = INTCHG(3) + BINLST(4,NODE)
         INTTOT = INTCHG(1) + 2*INTCHG(2) + 3*INTCHG(3)
C
                                          INTADD = 1
      IF (INTCHG(2) .NE. 0)  INTADD = 2
      IF (INTCHG(3) .NE. 0)  INTADD = 4
C
        INTTOT = INTTOT   + INTADD
        INTOUT = INTTOT/2 + 1
C
      IF (INTTOT .EQ. 6)  INTOUT = 3
C
        MAXINT = 4
C
      IF ((IBBR  .EQ. 3) .AND. (INTCHG(3) .EQ. 0))  MAXINT = 3
      IF (INTOUT .GT. MAXINT)  INTOUT = MAXINT
      IF (INTOUT .EQ. LSTINT)  GO TO 74
C
           NPT   = NPT + 1
    IDATA(NPT) = IOR(INTENT(INTOUT),BINLST(1,NODE))
       LSTINT   = INTOUT
   74     NODE   = BINLST(5,NODE)
C
   75 IF (NODE .NE. 0)  GO TO 70
C
C*****   CREATE THE COMPLETE VECTOR AND OUTPUT THE DATA...
C
           NPT   = NPT + 1
    IDATA(NPT) = IOR(ICOMP, MAXRNG(IRGSET))
C
    CALL SRDO(IDATA,NPT)
C
  999 RETURN
C
C**********************************************************************
C*****                   END OF SUBROUTINE 'CREATE'...          ******
C**********************************************************************
C
      END
```

SUBROUTINE INSERT

DESCRIPTION:

INSERTS INTENSITY CHANGES INTO LINKED LIST LIST ARRANGED IN ORDER OF RANGE. THE SUBROUTINE IS PASSED AN ARRAY OF INTENSITY CHANGES FOR LEVELS 0, 1, 2, and 3 RESPECTIVELY. ALSO PASSED ARE 2 RANGES. THE FIRST IS WHERE THE INTENSITY CHANGES OCCUR WHILE THE SECOND IS WHERE THE INTENSITY LEVEL REVERTS BACK. FOR A CHANGE OF LEVEL 1 TO LEVEL 2, THE ARRAY WOULD CONTAIN A -1 FOR LEVEL 1 AND A +1 FOR LEVEL 2. THIS INDICATES 1 LESS TARGET AT LEVEL 1 AND 1 MORE AT LEVEL 2. IF INTENSITY LEVELS ARE INSERTED AT A PREVIOUSLY INSERTED RANGE, THE INTENSITY LEVELS ARE ADDED. OTHERWISE, A NEW MODE IS ADDED.

COMMON DESCRIPTION:

SEE SUBROUTINE CREATE

VARIABLE DESCRIPTION:

JHEAD - CURRENT NODE IN LINKED LIST
INSNEG - SET TO 1 AFTER NEGATIVE INTENSITY CHANGES HAVE BEEN PROCESSED
INTCH(4) - INTENSITY CHANGES BEING INSERTED
INTENC(4) - LOCAL ARRAY TO HOLD INTENSITY CHANGES
IRNG - CURRENT ENCODED RANGE BEING INSERTED
IRNG1 - ENCODED RANGE FOR INTENSITY CHANGES
IRNG2 - ENCODED RANGE FOR REMOVAL OF INTENSITY CHANGES
NEWNOD - INSERTED NODE
OLDHD - PREVIOUS NODE IN BINARY TREE

SUBROUTINE CALLED:

NONE.

NAME = S:INSERT.HELRAD

```
C**********************************************************************
C**********************************************************************
C*****             TITLE...            S:INSERT, HELRAD          ******
C*****         LAST MODIFIED:    20 JAN, '82:  VOBEN             ******
C**********************************************************************
C**********************************************************************
C
C***     THIS SUBROUTINE INSERTS INTENSITY (COLOR) CHANGES INTO
C***   THE BINARY TREE ARRANGED BY RANGE VALUES...
C
C**********************************************************************
C
C***   SUBROUTINES CALLED:  NONE...
C
C**********************************************************************
C
      SUBROUTINE INSERT(IRNG1,IRNG2,INTCHG)
C
C**********************************************************************
C*****                         COMMON BLOCKS...                  ******
C**********************************************************************
C
      COMMON
     *     /BINTRE/BINLST(5,150), HDBLST, NXTNOD
     *     /TARGET/TGTLOC(2,100), TGTDST(100), TGTHDG(100),
     *             TGTPTR(2,100), TGTLOG(100), TGTTLT(100),
     *                                         TGTTYP(100),
     *             NBRTGT, NUMTGT,  TGTID
     *     /STATUS/LSTDIR, NXTALT, NXTUPD, TGTEND, TGTSTR, LAZM
C
C**********************************************************************
C*****                        DECLARATIONS...                    ******
C**********************************************************************
C
      INTEGER
     *           BINLST, HDBLST, OLDHD, TGTTYP
      INTEGER
     *           INTCHG(4), INTENC(4)
```

```
C
C*********************************************************************
C*********************************************************************
C
C***  INITIALIZE AND CØNVERT THE RANGE...
C
            ØLDHD = 0
            JHEAD = HDBLST
             IRNG = IRNG1
C
        INTENC(2) = INTCHG(2)
        INTENC(3) = INTCHG(3)
        INTENC(4) = INTCHG(4)
        INSNEG    = 0
C
C***  CHECK NØDE...
C
   10   IF (JHEAD .EQ. 0)      GØ TØ 50
        IF (IRNG - BINLST(1,JHEAD)) 50, 30, 40
C
C***  SAME RANGE - ADD INTENSITY...
C
   30   BINLST(2,JHEAD) = BINLST(2,JHEAD) + INTENC(2)
        BINLST(3,JHEAD) = BINLST(3,JHEAD) + INTENC(3)
        BINLST(4,JHEAD) = BINLST(4,JHEAD) + INTENC(4)
        GØ TØ 60
C
C***  GET THE NEXT PØINT...
C
   40       ØLDHD = JHEAD
            JHEAD = BINLST(5,ØLDHD);   GØ TØ 10
C
C***  INSERT A NEW NØDE...
C
   50     NEWNØD = NXTNØD
          NXTNØD = NXTNØD  + 1
          BINLST(1,NEWNØD) = IRNG
          BINLST(2,NEWNØD) = INTENC(2)
          BINLST(3,NEWNØD) = INTENC(3)
          BINLST(4,NEWNØD) = INTENC(4)
          BINLST(5,NEWNØD) = JHEAD
```

```
C
C***  UPDATE THE P0INTERS...
C
      IF (0LDHD .EQ. 0)  G0 T0 55
C
         BINLST(5,0LDHD)  = NEWN0D;  G0 T0 56
C
   55    HDBLST = NEWN0D
   56     JHEAD = NEWN0D
C
C***  INSERT NEGATIVE P0INT...
C
   60   IF (INSNEG .NE. 0)  G0 T0 999
C
         INTENC(2) = -INTENC(2)
         INTENC(3) = -INTENC(3)
         INTENC(4) = -INTENC(4)
         INSNEG    = 1
           IRNG    = IRNG2; G0 T0 10
C
  999 RETURN
C
C*************************************************************************
C*****                   END 0F SUBR0UTINE 'INSERT'...              ******
C*************************************************************************
C
      END
```

SUBROUTINE UPDATE

DESCRIPTION:
UPDATE UPDATES ONE TARGET'S DISTANCE AND BEARING AND MOVES IT TO THE CORRECT LOCATION IN THE CIRCULAR TARGET LIST

COMMON DESCRIPTION:
SEE SUBROUTINE CREATE.

VARIABLE DESCRIPTION:
CURTGT - CURRENT TARGET FOR CIRCULAR LIST TRAVERSAL
HDGCHG - CHANGE IN HEADING FROM LAST UPDATE
IDCCW - TARGET ID OF COUNTERCLOCKWISE TARGET
IDCW - TARGET ID OF CLOCKWISE TARGET
LENTRV - NUMBER OF TARGETS TRAVERSED
NEWHDG - NEW HEADING OF TARGET
NXTHDG - HEADING OF CURRENT TARGET BEING TRAVERSED
OLDHDG - HEADING OF TARGET LAST UPDATE
TGTID - ID OF TARGET BEING UPDATED

SUBROUTINES CALLED:
HDGDST - COMPUTES HEADING AND DISTANCE OF TARGET

NAME = S:SCIUPD.HELRAD

```
C*****************************************************************
C*****************************************************************
C*****           TITLE...            S:SCIUPD, HELRAD        ******
C*****        LAST MODIFIED:    20 JAN, '82:  VDBEN          ******
C*****************************************************************
C*****************************************************************
C***
C***      THIS SUBROUTINE UPDATES ONE TARGET'S DISTANCE AND
C***    BEARING AND MOVES IT TO THE CORRECT LOCATION IN THE
C***    CIRCULAR TARGET LIST...
C***
C*****************************************************************
C***
C***      SUBROUTINES CALLED:
C***
C***          HDGDST - COMPUTES THE HEADING AND DISTANCE OF EACH
C***                   TARGET...
C***
C*****************************************************************
C
      SUBROUTINE UPDATE
C
C*****************************************************************
C*****                       COMMON BLOCKS...                ******
C*****************************************************************
C
      COMMON
     *     /XFLOAT/ A(500)/IFIXED/IA(250)
     *     /HRCOM / RH(50),IH(50),KC(25)
      COMMON
     *     /BINTRE/BINLST(5,150), HDBLST, NXTNOD
     *     /TARGET/TGTLOC(2,100), TGTDST(100), TGTHDG(100),
     *             TGTPTR(2,100), TGTLOG(100), TGTTLT(100),
     *                                         TGTTYP(100),
     *             NBRTGT, NUMTGT,  TGTID
     *     /STATUS/LSTDIR, NXTALT, NXTUPD, TGTEND, TGTSTR, LAZM
C
C*****************************************************************
```

```
C*****                       EQUIVALENCES...                    ******
C**********************************************************************
C
      EQUIVALENCE
     *    ( A(358),   D2R), ( A(460),    PI), ( A(461), TWØPI)
C
C**********************************************************************
C*****                       DECLARATIØNS...                    ******
C**********************************************************************
C
        REAL
     *           NEWHDG, NXTHDG
      INTEGER
     *           CNTTGT, TGTEND,  TGTID, TGTPTR, TGTSTR, TGTTYP
C
C**********************************************************************
C**********************************************************************
C
C***  CØMPUTE DISTANCE AND BEARING...
C
            ØLDHDG = TGTHDG(TGTID)
C
      CALL HDGDST
                               NEWHDG = TGTHDG(TGTID)
                               HDGCHG = NEWHDG - ØLDHDG
        IF (HDGCHG .GT.  PI)  HDGCHG = HDGCHG - TWØPI
        IF (HDGCHG .LT. -PI)  HDGCHG = HDGCHG + TWØPI
                               NEWHDG = ØLDHDG + HDGCHG
C
C***  CHECK FØR PRØPER PLACE IN LIST...
C
        IF (HDGCHG .LT. 0.0)  GØ TØ 40
C
C***  TRAVERSE CLØCKWISE...
C
            CNTTGT = TGTID
            LENTRV = 0
C
   30       LENTRV = LENTRV + 1
            CNTTGT = TGTPTR(2,CNTTGT)
C
```

```
          IF (CNTTGT .EQ. TGTID)  RETURN
                                  NXTHDG = TGTHDG(CNTTGT)
          IF (ØLDHDG .GT. NXTHDG) NXTHDG = NXTHDG + TWØPI
                                  ØLDHDG = NXTHDG
          IF (NEWHDG .GT. NXTHDG) GØ TØ 30
C
C***  CHECK IF MØVE NECESSARY...
C
          IF (LENTRV .NE. 1)  GØ TØ 35
          RETURN
C
C***  REMØVE THE TARGET...
C
   35                    IDCCW  = TGTPTR(1,TGTID)
                         IDCW   = TGTPTR(2,TGTID)
            TGTPTR(2,IDCCW) = IDCW
            TGTPTR(1,IDCW ) = IDCCW
C
C***  INSERT THE TARGET...
C
                        IDCCW    = TGTPTR(1,CNTTGT)
           TGTPTR(2,IDCCW)   = TGTID
           TGTPTR(1,TGTID)   = IDCCW
           TGTPTR(2,TGTID)   = CNTTGT
           TGTPTR(1,CNTTGT)  = TGTID
           RETURN
C
C***  TRAVERSE CØUNTERCLØCKWISE...
C
   40       CNTTGT = TGTID
            LENTRV = 0
C
   45       LENTRV = LENTRV + 1
            CNTTGT = TGTPTR(1,CNTTGT)
C
          IF (CNTTGT .EQ. TGTID)  RETURN
C
            NXTHDG = TGTHDG(CNTTGT)
C
          IF (ØLDHDG .LT. NXTHDG) NXTHDG = NXTHDG - TWØPI
                                  ØLDHDG = NXTHDG
```

```
      IF (NEWHDG .LT. NXTHDG) GO TO 45
C
C***  CHECK IF MOVE NECESSARY...
C
      IF (LENTRY .NE. 1)  GO TO 50
      RETURN
C
C***  REMOVE THE TARGET...
C
   50                   IDCCW  = TGTPTR(1,TGTID)
                        IDCW   = TGTPTR(2,TGTID)
            TGTPTR(2,IDCCW) = IDCW
            TGTPTR(1,IDCW ) = IDCCW
C
C***  INSERT THE TARGET...
C
                        IDCW     = TGTPTR(2,CNTTGT)
            TGTPTR(1,IDCW)     = TGTID
            TGTPTR(1,TGTID)    = CNTTGT
            TGTPTR(2,TGTID)    = IDCW
            TGTPTR(2,CNTTGT)   = TGTID
            RETURN
C
C************************************************************************
C*****                  END OF SUBROUTINE 'SCIUPD'...              ******
C************************************************************************
C
      END
```

```
C       SUBROUTINE HDGDST

C       DESCRIPTION:
C               COMPUTES HEADING AND DISTANCE OF GIVEN TARGET USING
C               THE DISTANCE OF THE HELICOPTER FROM THE PRIMARY TARGET AND
C               THE DISTANCE OF THE TARGET FROM THE PRIMARY TARGET

C       COMMON DESCRIPTION:
C               SEE SUBROUTINE CREATE.

C       VARIABLE DESCRIPTION:
C               DELX - DIFFERENCE IN X LOCATION
C               DELY - DIFFERENCE IN Y LOCATION
C               FT2NM - FEET TO NAUTICAL MILE CONVERSION
C               NEWHDG - NEW HEADING OF TARGET
C               TGTID - ID OF TARGET BEING UPDATED

C       SUBROUTINE CALLED:
C               NONE.
```

NAME = S:HDGDST.HELRAD

```
C*****************************************************************
C*****************************************************************
C*****           TITLE...          S:HDGDST, HELRAD         ******
C*****        LAST MØDIFIED:    20 JAN, '82:  VDHEN         ******
C*****************************************************************
C*****************************************************************
C***
C***  THIS SUBRØUTINE CØMPUTES THE HEADING AND DISTANCE ØF
C***    GIVEN TARGETS USING THE X- AND Y- CØØRDINATES ØF THE
C***    TARGET AND HELICØPTER...  NØ ELLIPTICAL CØRRECTIØNS
C***    ARE USED...
C***
C*****************************************************************
C***   SUBRØUTINES CALLED:  NØNE...
C*****************************************************************
C
      SUBRØUTINE HDGDST
C
C*****************************************************************
C*****                    CØMMØN BLØCKS...                  ******
C*****************************************************************
C
      CØMMØN
     *     /XFLØAT/ A(500)/IFIXED/IA(250)
     *     /HRCØM /RH( 50),IH(50),KC( 25)
      CØMMØN
     *     /BINTRE/BINLST(5,150), HDBLST, NXTNØD
     *     /TARGET/TGTLØC(2,100), TGTDST(100), TGTHDG(100),
     *             TGTPTR(2,100), TGTLØG(100), TGTTLT(100),
     *                                         TGTTYP(100),
     *             NBRTGT, NUMTGT,  TGTID
     *     /STATUS/LSTDIR, NXTALT, NXTUPD, TGTEND, TGTSTR, LAZM
C
C*****************************************************************
C*****                    EQUIVALENCES...                   ******
C*****************************************************************
C
      EQUIVALENCE
```

```
      *     ( A(103),    XPR), ( A(104),    YPR), ( A(105),    HPR),
      *     ( A(358),    D2R), ( A(460),     PI), ( A(461), TWOPI)
C
C*********************************************************************
C*****                         DECLARATIONS...                  *****
C*********************************************************************
C
            REAL    NEWHDG
C
        INTEGER   TGTEND,  TGTID, TGTPTR, TGTSTR, TGTTYP
C
C*********************************************************************
C*****                        DATA STATEMENTS...                *****
C*********************************************************************
C
        DATA  FT2NM/1.6447368E-4/, XPR/-12160.0/, YPR/0.0/
C
C*********************************************************************
C*********************************************************************
C
C*****   COMPUTE THE DISTANCE...
C
                 DELX = TGTLOC(1,TGTID) - XPR
                 DELY = TGTLOC(2,TGTID) - YPR
C
        TGTDST(TGTID) =  SQRT(DELX**2 + DELY**2 + HPR**2)*FT2NM
        TGTLOG(TGTID) =  LOG10(TGTDST(TGTID))
               ALTNM  =  -HPR*FT2NM
        TGTTLT(TGTID) =  ASIN(ALTNM/TGTDST(TGTID))
C
        IF (DELX .EQ. 0.0) RETURN
C
C*****   COMPUTE THE HEADING...
C
                              NEWHDG = ARCTAN(DELY,DELX)
        IF (NEWHDG .LT. 0.0)  NEWHDG = NEWHDG + TWOPI
C
        TGTHDG(TGTID) = NEWHDG
C
        RETURN
C
```

```
C*********************************************************************
C*********************************************************************
C
      REAL FUNCTIØN ARCTAN(DX,DY)
C
      DATA  HALFPI/1.5707963/, PI/3.1415927/, TWØPI/6.2831853/
C
      IF (ABS(DX) .GT. ABS(DY))  GØ TØ 10
          IF (DY  .EQ.     0.0)  ARCTAN = 0.0; RETURN
C
                                 ARCTAN = ATAN(DX/DY)
      IF      (DY .LT. 0.0)  ARCTAN = ARCTAN + PI
      IF (ARCTAN .GT.  PI)  ARCTAN = ARCTAN - TWØPI
      RETURN
C
   10                        ARCTAN = HALFPI - ATAN(DY/DX)
      IF (DX .LT. 0.0)  ARCTAN = ARCTAN - PI
      RETURN
C
C*********************************************************************
C*****                 END ØF SUBRØUTINE 'HDGDST'...            ******
C*********************************************************************
C
      END
```

SUBROUTINE INIRDR

DESCRIPTION:

INIRDR INITIALIZES THE TARGET DATA NECESSARY FOR THE RADAR SIMULATION. THE HELICOPTER LOCATION MUST BE INITIALIZED IN COMMON XFLOAT PRIOR TO CALLING THIS ROUTINE. ALL THE TARGETS ARE DATA INITIALIZED IN COMMON /TARGET/. IN ORDER TO CHOOSE THE BLOCK OF TARGETS, THE OPERATOR MUST FIRST SET NURTGT TO THE FIRST TARGET IN THE BLOCK. THEN A CALL TO INIRDR WILL INITIALIZE THE TARGET DATA FOR THAT BLOCK OF TARGETS.

COMMON DESCRIPTION:

SEE SUBROUTINE CREATE.

VARIABLE DESCRIPTION:

NONE.

SUBROUTINE CALLED:

INSTGT - INSTALL TARGETS INTO TARGET LIST.

NAME = S:INIRDR.HELRAD

```
C************************************************************************
C************************************************************************
C*****           TITLE...           S:INIRDR, HELRAD               *****
C*****      LAST MODIFIED:     20 JAN, '82:  VDBEN                 *****
C************************************************************************
C************************************************************************
C***
C***     THIS SUBROUTINE INITIALIZES THE TARGET DATA NECESSARY
C***  FOR THE HELICOPTER RADAR SIMULATION...  ALL TARGETS ARE
C***  INSERTED VIA A CALL TO 'INSTGT'...  THE PARAMETERS ARE
C***  THE X- AND Y- COORDINATES OF THE TARGET (IN FEET)...
C***     THE HELICOPTER SIMULATION MUST BE INITIALIZED IN
C***  'XFLOAT' COMMON PRIOR TO CALLING THIS SUBROUTINE...
C***
C************************************************************************
C***
C***   SUBROUTINES CALLED:
C***
C***     INSTGT - INSTALL TARGET DATA INTO THE TARGET LIST...
C***
C************************************************************************
C
      SUBROUTINE INIRDR
C
C************************************************************************
C*****                       COMMON BLOCKS...                      *****
C************************************************************************
C
      COMMON
     *     /XFLOAT/ A(500) /IFIXED/IA(250)
     *     /HRCOM / RH(50), IH(50), KC(25)
      COMMON
     *     /BINTRE/BINLST(5,150), HDBLST, NXTNOD
     *     /TARGET/TGTLOC(2,100), TGTDST(100), TGTHDG(100),
     *               TGTPTR(2,100), TGTLOG(100), TGTTLT(100),
     *                                            TGTTYP(100),
     *               NBRTGT, NUMTGT,  TGTID
     *     /STATUS/LSTDIR, NXTALT, NXTUPD, TGTEND, TGTSTR, LAZM
```

```
     *     /ANTENA/ANTDIM, RADØME, ANGAIN, BRKPNT, RNGBRK, ANTLØG
C
C*****************************************************************
C*****                    DECLARATIØNS...                   *****
C*****************************************************************
C
      INTEGER   TGTEND,  TGTID, TGTPTR, TGTSTR, TGTTYP
C
C*****************************************************************
C*****                   DATA STATEMENTS...                 *****
C*****************************************************************
C
      DATA    ANGAIN/28.0/, ANTDIM/12.0/, TGTPTR/200*0/,
     *         XRGS/500.0/, XRLØC/1500./,
     *         YRGS/-100./, YRLØC/0.000/
C
      DATA        NBRTGT/3/,    NUMTGT/7/,  TGTTYP(3)/1/
C
C*****************************************************************
C*****                     TARGET DATA...                   *****
C*****************************************************************
C
C***  ILS RADAR TARGETS...  SET 'NBRTGT' = 1; 'NUMTGT' = 2...
C
      DATA     TGTLØC(1,  1)/  500.0/, TGTLØC(2,  1)/  -100.0/,
     *         TGTLØC(1,  2)/ 1500.0/, TGTLØC(2,  2)/     0.0/
C
C***  ØIL RIG TARGETS...    SET 'NBRTGT' = 3; 'NUMTGT' = 7...
C
      DATA     TGTLØC(1,  3)/    0.0/, TGTLØC(2,  3)/     0.0/,
     *         TGTLØC(1,  4)/ 1986.0/, TGTLØC(2,  4)/  -304.0/,
     *         TGTLØC(1,  5)/ 3972.0/, TGTLØC(2,  5)/  2723.0/,
     *         TGTLØC(1,  6)/ 3986.0/, TGTLØC(2,  6)/   561.0/,
     *         TGTLØC(1,  7)/ 3669.0/, TGTLØC(2,  7)/ -1430.0/,
     *         TGTLØC(1,  8)/  992.0/, TGTLØC(2,  8)/ -3491.0/,
     *         TGTLØC(1,  9)/-8094.0/, TGTLØC(2,  9)/ -5610.0/
C
C***  SANTA BARBARA CHANNEL (9 TARGETS)...
C***                           SET 'NBRTGT' = 10; 'NUMTGT' = 9...
C
      DATA     TGTLØC(1, 10)/-21400.0/, TGTLØC(2, 10)/-1000.0/,
```

```
      *        TGTLØC(1, 11)/-18500.0/, TGTLØC(2, 11)/ -900.0/,
      *        TGTLØC(1, 12)/-15800.0/, TGTLØC(2, 12)/ -200.0/,
      *        TGTLØC(1, 13)/-13100.0/, TGTLØC(2, 13)/ -160.0/,
      *        TGTLØC(1, 14)/     0.0/, TGTLØC(2, 14)/    0.0/,
      *        TGTLØC(1, 15)/  2380.0/, TGTLØC(2, 15)/ -950.0/,
      *        TGTLØC(1, 16)/  4990.0/, TGTLØC(2, 16)/-1300.0/,
      *        TGTLØC(1, 17)/  8480.0/, TGTLØC(2, 17)/-2400.0/,
      *        TGTLØC(1, 18)/ 12120.0/, TGTLØC(2, 18)/-3200.0/
C
C***   1 NAUTICAL MILE ARC...  SET 'NBRTGT' = 19; 'NUMTGT' = 8...
C
       DATA    TGTLØC(1, 19)/ 5364.9/, TGTLØC(2, 19)/ -2852.6/,
      *        TGTLØC(1, 20)/ 5709.7/, TGTLØC(2, 20)/ -2078.1/,
      *        TGTLØC(1, 21)/ 5943.3/, TGTLØC(2, 21)/ -1263.3/,
      *        TGTLØC(1, 22)/ 6061.3/, TGTLØC(2, 22)/  -423.8/,
      *        TGTLØC(1, 23)/ 6061.3/, TGTLØC(2, 23)/   423.8/,
      *        TGTLØC(1, 24)/ 5943.3/, TGTLØC(2, 24)/  1263.3/,
      *        TGTLØC(1, 25)/ 5709.7/, TGTLØC(2, 25)/  2078.1/,
      *        TGTLØC(1, 26)/ 5364.9/, TGTLØC(2, 26)/  2852.6/
C
C***   2 NAUTICAL MILE ARC...  SET 'NBRTGT' = 27; 'NUMTGT' = 8...
C
       DATA    TGTLØC(1, 27)/10729.8/, TGTLØC(2, 27)/ -5705.1/,
      *        TGTLØC(1, 28)/11419.3/, TGTLØC(2, 28)/ -4156.3/,
      *        TGTLØC(1, 29)/11886.6/, TGTLØC(2, 29)/ -2526.6/,
      *        TGTLØC(1, 30)/12122.6/, TGTLØC(2, 30)/  -847.7/,
      *        TGTLØC(1, 31)/12122.6/, TGTLØC(2, 31)/   847.7/,
      *        TGTLØC(1, 32)/11886.6/, TGTLØC(2, 32)/  2526.6/,
      *        TGTLØC(1, 33)/11419.3/, TGTLØC(2, 33)/  4156.3/,
      *        TGTLØC(1, 34)/10729.8/, TGTLØC(2, 34)/  5705.1/
C
C***   3 NAUTICAL MILE ARC...  SET 'NBRTGT' = 35; 'NUMTGT' = 8...
C
       DATA    TGTLØC(1, 35)/16094.6/, TGTLØC(2, 35)/ -8557.7/,
      *        TGTLØC(1, 36)/17129.0/, TGTLØC(2, 36)/ -6234.4/,
      *        TGTLØC(1, 37)/17830.0/, TGTLØC(2, 37)/ -3789.9/,
      *        TGTLØC(1, 38)/18183.9/, TGTLØC(2, 38)/ -1271.5/,
      *        TGTLØC(1, 39)/18183.9/, TGTLØC(2, 39)/  1271.5/,
      *        TGTLØC(1, 40)/17830.0/, TGTLØC(2, 40)/  3789.9/,
      *        TGTLØC(1, 41)/17129.0/, TGTLØC(2, 41)/  6234.4/,
      *        TGTLØC(1, 42)/16094.6/, TGTLØC(2, 42)/  8557.7/
```

```
C
C***    4 NAUTICAL MILE ARC...   SET 'NBR   TGT' = 43; 'NUMTGT' = 8...
C
        DATA      TGTLOC(1, 43)/21459.5/, TGTLOC(2, 43)/-11410.2/,
     *            TGTLOC(1, 44)/22838.7/, TGTLOC(2, 44)/ -8312.6/,
     *            TGTLOC(1, 45)/23773.3/, TGTLOC(2, 45)/ -5053.2/,
     *            TGTLOC(1, 46)/24245.2/, TGTLOC(2, 46)/ -1695.4/,
     *            TGTLOC(1, 47)/24245.2/, TGTLOC(2, 47)/  1695.4/,
     *            TGTLOC(1, 48)/23773.3/, TGTLOC(2, 48)/  5053.2/,
     *            TGTLOC(1, 49)/22838.7/, TGTLOC(2, 49)/  8312.6/,
     *            TGTLOC(1, 50)/21459.5/, TGTLOC(2, 50)/ 11410.2/
C
C***    5 NAUTICAL MILE ARC...   SET 'NBR   TGT' = 51; 'NUMTGT' = 8...
C
        DATA      TGTLOC(1, 51)/26824.4/, TGTLOC(2, 51)/-14262.8/,
     *            TGTLOC(1, 52)/28548.3/, TGTLOC(2, 52)/-10390.7/,
     *            TGTLOC(1, 53)/29716.6/, TGTLOC(2, 53)/ -6316.5/,
     *            TGTLOC(1, 54)/30306.5/, TGTLOC(2, 54)/ -2119.2/,
     *            TGTLOC(1, 55)/30306.5/, TGTLOC(2, 55)/  2119.2/,
     *            TGTLOC(1, 56)/29716.6/, TGTLOC(2, 56)/  6316.5/,
     *            TGTLOC(1, 57)/28548.3/, TGTLOC(2, 57)/ 10390.7/,
     *            TGTLOC(1, 58)/26824.4/, TGTLOC(2, 58)/ 14262.8/
C
C***   10 NAUTICAL MILE ARC...   SET 'NBR   TGT' = 59; 'NUMTGT' = 8...
C
        DATA      TGTLOC(1, 59)/53649.0/, TGTLOC(2, 59)/-28526.0/,
     *            TGTLOC(1, 60)/57097.0/, TGTLOC(2, 60)/-20781.0/,
     *            TGTLOC(1, 61)/59433.0/, TGTLOC(2, 61)/-12633.0/,
     *            TGTLOC(1, 62)/60613.0/, TGTLOC(2, 62)/ -4238.0/,
     *            TGTLOC(1, 63)/60613.0/, TGTLOC(2, 63)/  4238.0/,
     *            TGTLOC(1, 64)/59433.0/, TGTLOC(2, 64)/ 12633.0/,
     *            TGTLOC(1, 65)/57097.0/, TGTLOC(2, 65)/ 20781.0/,
     *            TGTLOC(1, 66)/53649.0/, TGTLOC(2, 66)/ 28526.0/
C
C***    SANTA BARBARA CHANNEL (11 TARGETS)...
C***                                     SET 'NBR   TGT' = 67; 'NUMTGT' = 11..
C
        DATA      TGTLOC(1, 67)/-21400.0/, TGTLOC(2, 67)/-1000.0/,
     *            TGTLOC(1, 68)/-18500.0/, TGTLOC(2, 68)/ -900.0/,
     *            TGTLOC(1, 69)/-15800.0/, TGTLOC(2, 69)/ -200.0/,
     *            TGTLOC(1, 70)/-13100.0/, TGTLOC(2, 70)/ -160.0/,
```

```
     *           TGTLØC(1, 71)/     0.0/, TGTLØC(2, 71)/     0.0/,
     *           TGTLØC(1, 72)/  2380.0/, TGTLØC(2, 72)/  -950.0/,
     *           TGTLØC(1, 73)/  4990.0/, TGTLØC(2, 73)/ -1300.0/,
     *           TGTLØC(1, 74)/  8480.0/, TGTLØC(2, 74)/ -2400.0/,
     *           TGTLØC(1, 75)/ 12120.0/, TGTLØC(2, 75)/ -3200.0/,
     *           TGTLØC(1, 76)/-10450.0/, TGTLØC(2, 76)/-20200./,
     *           TGTLØC(1, 77)/ -2130.0/, TGTLØC(2, 77)/-18300.0/
C
C*********************************************************************
C*********************************************************************
C
C***  EQUATE THE RILS CØØRDINATES WITH THEIR RESPECTIVE LØCAT-
C***  IØNS IN ARRAY 'TGTLØC'...
C
            TGTLØC(1,  1) = XRGS ; TGTLØC(2,  1) = YRGS
            TGTLØC(1,  2) = XRLØC; TGTLØC(2,  2) = YRLØC
C
C***  INITIALIZE PARAMETERS AND SET NUMBER ØF TARGETS TØ ZERØ...
C
            LSTDIR =  1
            NXTALT =  1
              LAZM = -100
C
            ANTLØG = LØG10(ANTDIM)
            BRKPNT = 10.0**((-2.4 + ANGAIN)/16.86)
            RNGBRK = 10.0**(( 4.0 + 20.0*ANTLØG)/16.86)
C
            CALL INSTGT
C
      RETURN
C
C*********************************************************************
C*****                END ØF SUBRØUTINE 'INIRDR'...           ******
C*********************************************************************
C
      END
```

SUBROUTINES INSTGT

DESCRIPTION:

INSTGT INSERTS INTO THE TARGET COMMON AREA THE COMPUTED BEARING AND DISTANCE FROM THE HELICOPTER AND MAINTAINS THE TARGET POINTERS SO THAT THE TARGETS ARE IN A CIRCULAR LIST IN ORDER OF BEARING. ALSO DEFINED IS THE STARTING VALUES OF VARIOUS POINTERS, INTO THE TARGET LIST.

COMMON DESCRIPTION:

SEE SUBROUTINE CREATE.

VARIABLE DESCRIPTION:

FSTVIS - FIRST VISIBLE TARGET IN TARGET LIST
HDGDIF - HEADING DIFFERENCE BETWEEN TARGET AND HELICOPTER
HDGEND - HEADING DIFFERENCE OF LAST VISIBLE TARGET
HDGSTR - HEADING DIFFERENCE OF FIRST VISIBLE TARGET
HEAD - HEAD OF TARGET LIST (SMALLEST HEADING)
IDCCW - COUNTERCLOCKWISE TARGET I.D.
IDCW - CLOCKWISE TARGET I.D.
LSTVIS - TARGET I.D. OF LAST VISIBLE TARGET
NTGHDG - HEADING OF TARGET BEING INSERTED
TAIL - TAIL OF TARGET LIST (LARGEST NUMBER)
TGTID - I.D. OF TARGET BEING INSERTED
XLOC - X LOCATION OF TARGET
YLOC - Y LOCATION OF TARGET

SUBROUTINE CALLED:

HDGDST - COMPUTES HEADING AND DISTANCE OF TARGET

NAME = S:INSTGT.HELRAD

```
C*******************************************************************
C*******************************************************************
C*****            TITLE...             S:INSTGT, HELRAD         ******
C*****        LAST MODIFIED:     20 JAN, '82:  VDBEN            ******
C*******************************************************************
C*******************************************************************
C***
C***     THIS SUBROUTINE INSERTS INTO THE TARGET COMMON AREA THE
C***   X- AND Y- COORDINATES OF THE TARGETS; COMPUTES THE BEAR-
C***   ING AND DISTANCE FROM THE HELICOPTER AND MAINTAINS THE
C***   TARGET POINTERS SO THAT THE TARGETS ARE IN A CIRCULAR
C***   LIST IN ORDER OF THE BEARING VALUES...
C***
C*******************************************************************
C***
C***    SUBROUTINES CALLED:
C***
C***      HDGDST - COMPUTES THE HEADING AND DISTANCE OF EACH
C***               TARGET...
C***
C*******************************************************************
C
      SUBROUTINE INSTGT
C
C*******************************************************************
C*****                        COMMON BLOCKS...                 ******
C*******************************************************************
C
      COMMON
     *     /XFLOAT/ A(500)/IFIXED/IA(250)
     *     /HRCOM /RH( 50),IH(50),KC( 25)
      COMMON
     *     /BINTRE/BINLST(5,150), HDBLST, NXTNOD
     *     /TARGET/TGTLOC(2,100), TGTDST(100), TGTHDG(100),
     *             TGTPTR(2,100), TGTLOG(100), TGTTLT(100),
     *                                        TGTTYP(100),
     *             NBRTGT, NUMTGT,  TGTID
     *     /STATUS/LSTDIR, NXTALT, NXTUPD, TGTEND, TGTSTR, LAZM
```

```
C
C***********************************************************************
C*****                      EQUIVALENCES...                       ******
C***********************************************************************
C
      EQUIVALENCE
     *         ( A(   6),  PSIR), ( A(358),   D2R),
     *         ( A(460),    PI), ( A(461), TWØPI)
C
C***********************************************************************
C*****                      DECLARATIØNS...                       ******
C***********************************************************************
C
       REAL    NTGHDG
C
      INTEGER
     *         FSTVIS,   HEAD,   TAIL, TGTEND,
     *          TGTID, TGTPTR, TGTSTR, TGTTYP
C
C***********************************************************************
C*****                    DATA STATEMENTS...                      ******
C***********************************************************************
C
      DATA
     *        PI/3.1415927/,  TWØPI/6.2831853/, MAXTGT/100/
C
C***********************************************************************
C***********************************************************************
C
C***  GET NEW TARGET ID...
C
      IF  (NUMTGT .GT. (MAXTGT-NBRTGT))  NUMTGT = MAXTGT-NBRTGT
C
      IF ((NBRTGT .LE. 0) .ØR. (NBRTGT+NUMTGT-1 .GT. MAXTGT))
     *     RETURN
C
C*****   INITIALIZE THE TARGET PØINTER ARRAY TØ ZERØ...
C
         DØ 2, I  = 1,2
         DØ 1, J  = 1,100
       TGTPTR(I,J) = 0
```

```
      1     CØNTINUE
      2     CØNTINUE
C
C*****
                        INDEX = 0
      REPEAT 100, WHILE INDEX < NUMTGT
C
            TGTID = NBRTGT + INDEX
C
C*****    CØMPUTE HEADING AND DISTANCE...
C
            CALL HDGDST
C
C*****    FIX PØINTERS...
C
            IF (TGTID .EQ. NBRTGT)  GØ TØ 30
C
            IDCCW = TAIL
            IDCW  = HEAD
            NTGHDG = TGTHDG(TGTID)
C
      5     IF (NTGHDG - TGTHDG(IDCW)) 10, 20, 20
C
C*****    INSERT INTØ LIST...
C
     10     TGTPTR(2,IDCCW) = TGTID
            TGTPTR(1,IDCW ) = TGTID
            TGTPTR(1,TGTID) = IDCCW
            TGTPTR(2,TGTID) = IDCW
C
            IF ((IDCW .EQ. HEAD) .AND. (TAIL .NE. TGTID))
     *                     HEAD = TGTID
            GØ TØ 50
C
C*****    GET THE NEXT TARGET...
C
     20     IDCCW = IDCW
            IDCW  = TGTPTR(2,IDCCW)
C
            IF (IDCW .NE. HEAD)  GØ TØ 5
C
```

```
              TAIL = TGTID
              GØ TØ 10
C
C*****   FIRST TARGET...
C
   30           HEAD =   TAIL  = NBRTGT
              TGTPTR(1,NBRTGT) = NBRTGT
              TGTPTR(2,NBRTGT) = NBRTGT
                      HDGSTR  =  1.5
                      HDGEND  = -1.5
              FSTVIS = LSTVIS  = NBRTGT
C
C*****   UPDATE TARGET PØINTERS...
C
   50                             HDGDIF = TGTHDG(TGTID) - PSIR
              IF (HDGDIF .LE. -PI) HDGDIF = HDGDIF + TWØPI
              IF (HDGDIF .GT.  PI) HDGDIF = HDGDIF - TWØPI
              IF (ABS(HDGDIF) .GT. 1.5)  GØ TØ 60
              IF (HDGDIF .LE. HDGEND)    GØ TØ 55
C
              HDGEND = HDGDIF
              LSTVIS = TGTID
C
   55         IF (HDGDIF .GE. HDGSTR)  GØ TØ 60
C
              HDGSTR = HDGDIF
              FSTVIS = TGTID
C
   60         TGTSTR = TGTPTR(1,FSTVIS)
              TGTEND = TGTPTR(2,LSTVIS)
              NXTUPD = TGTEND
               INDEX =  INDEX + 1
C
  100 CØNTINUE
  999 RETURN
C
C*****************************************************************
C*****                  END ØF SUBRØUTINE 'INSTGT'...       ******
C*****************************************************************
C
      END
```

NAME = S:RANDUM,HELRAD

```
C***********************************************************************
C***********************************************************************
C*****              TITLE...            S:RANDUM, HELRAD          ******
C*****          LAST MODIFIED:     20 JAN, '82:  VDREN            ******
C***********************************************************************
C***********************************************************************
C***
C***
C***   PURPOSE:
C***               COMPUTES UNIFORMLY DISTRIBUTED RANDOM REAL
C***   NUMBERS BETWEEN 0.0 AND 1.0 AND RANDOM INTEGERS BETWEEN
C***   ZERO AND 2**31...   EACH ENTRY USES AS INPUT AN INTEGER
C***   RANDOM NUMBER AND PRODUCES A NEW INTEGER AND REAL RANDOM
C***   NUMBER...
C***
C***
C***   UTILIZATION:  CALL RANDUM(IRX,IRY,ZFLP)
C***
C***
C***   DESCRIPTION OF PARAMETERS:
C***
C***    'INTX' - FOR THE FIRST ENTRY THIS MUST CONTAIN ANY ODD
C***             INTEGER NUMBER WITH NINE OR LESS DIGITS...
C***             AFTER THE FIRST ENTRY, 'IRX' SHOULD BE THE
C***             PREVIOUS VALUE OF 'IRY' COMPUTED BY THIS SUB-
C***             ROUTINE...
C***
C***    'INTY' - A RESULTANT INTEGER RANDOM NUMBER REQUIRED FOR
C***             THE NEXT ENTRY TO THIS SUBROUTINE...
C***             THE RANGE OF THIS NUMBER IS BETWEEN ZERO AND
C***             2**31...
C***
C***    'WFLP' - THE RESULTANT, UNIFORMLY DISTRIBUTED, FLOATING
C***             POINT, RANDOM NUMBER WITH RANGE 0.0 TO 1.0...
C***
C***
C***   METHOD:
C***             POWER RESIDUE METHOD DISCUSSED IN IBM MANUAL
```

```
C***                       020-8011, RANDOM NUMBER GENERATION AND TESTING
C***
C*****************************************************************************
C
      SUBROUTINE RANDUM(INTX,INTY,WFLP)
C
         REAL  WFLP
C
      INTEGER  INTX, INTY
C
               INTY = INTX*65539
               INTX = INTY
C
      IF (INTY), 1,1,2
C
    1          INTY = INTY + 2147483647 + 1
    2          WFLP = INTY
C
               WFLP = WFLP*0.4656613E-9
C
      RETURN
C
C*****************************************************************************
C*****                 END OF SUBROUTINE 'RANDUM'...                    *****
C*****************************************************************************
C
      END
```

Technical Report Documentation Page

1. Report No.
NASA CR-166293

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

REAL-TIME SIMULATION OF AN AIRBORNE RADAR FOR OVERWATER APPROACHES

5. Report Date
MARCH 1982

6. Performing Organization Code

8. Performing Organization Report No.

7. Author(s)
J.S. Karmarkar and D. Clark

9. Performing Organization Name and Address
SYSTEMS CONTROL TECHNOLOGY, INC.
(formerly Systems Control, Inc. (Vt))
1801 Page Mill Road
Palo Alto, CA 94303

10. Work Unit No. (TRAIS)
T-3771Y

11. Contract or Grant No.
NAS2-10479

12. Sponsoring Agency Name and Address
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546

13. Type of Report and Period Covered
FINAL REPORT
November 1980 — November 1981

14. Sponsoring Agency Code
532-01-11

15. Supplementary Notes
Technical Monitor: George R. Clary, Mail Stop 210-9, NASA-Ames Research Center, Moffett Center, Moffett Field, CA 94035, (415) 965-5452 or FTS 448-5452.

16. Abstract

This report documents the real-time algorithm used to generate the necessary video interface signals, for each of the selected oil rig/helicopter relative geometry scenarios, to drive an airborne color radar display. Typical display results are documented.

Unique features of the real-time algorithm include: (1) a circular-list approach to target sorting and ray generation, (2) target shape and multiple target merging effect generation, and (3) sea clutter generation.

17. Key Words
Color radar display simulation
Real-time
Helicopter display
Overwater radar
Airborne radar approaches

18. Distribution Statement
Unclassified - Unlimited

STAR Category - 04

19. Security Classif. (of this report)
UNCLASSIFIED

20. Security Classif. (of this page)
UNCLASSIFIED

21. No. of Pages
131

22. Price

End of Document