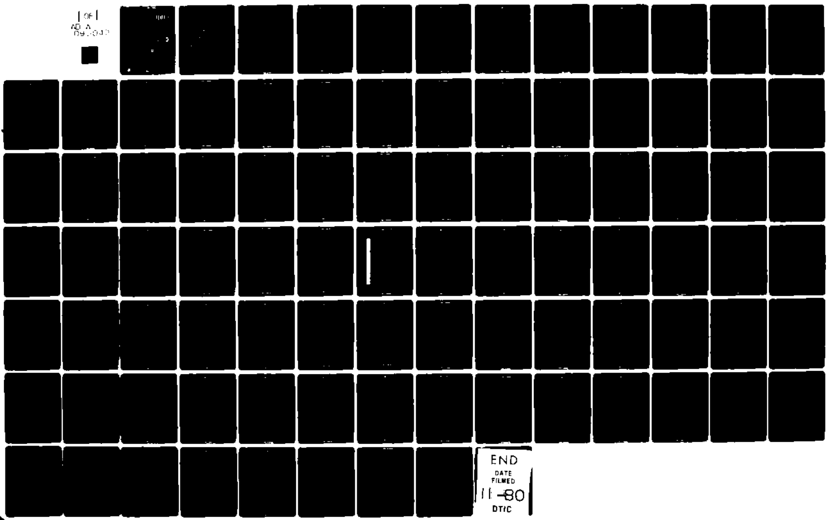
AD-A090 040 GENERAL RESEARCH CORP HUNTSVILLE AL F/6 19/5
FIRE CONTROL SIMULATION REQUIREMENTS. VOLUME III.(U)
MAY 80 R R PARKER, W V BOULDIN, C P MARKS DAAK40-78-C-0054
UNCLASSIFIED GRC-CR-6-824 NL



CR-6-824

FINAL REPORT

FIRE CONTROL SIMULATION REQUIREMENTS (U)

Volume III

Contract No. DAAK40-78-C-0054

R. R. Parker
W. V. Bouldin
C. P. Marks
G. K. Warmbrod

May 1980

SYSTEMS TECHNOLOGIES GROUP
GENERAL RESEARCH CORPORATION
A SUBSIDIARY OF FLOW GENERAL INC.
307 Wynn Drive, Huntsville, Alabama 35805

DISCLAIMER NOTICE

UNCLASSIFIED

TIO-2006

14 GRC-CR-6-824

9 FINAL REPORT. Jan-Dec 79,

6

FIRE CONTROL SIMULATION REQUIREMENTS

Volume III.

Contract No. 15 DAAK40-78-C-0054

10 R. R. Parker
W. V. Bouldin
C. P. Marks
G. K. Warmbrod

11 May 1980

12 83

Accession For	
NTIS GRA&I	☑
DDC TAB	☐
Unannounced	☐
Justification: Per Letter on File	
By	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
A	23

GENERAL RESEARCH CORPORATION
307 Wynn Drive
Huntsville, Alabama 35805

407116

UNCLASSIFIED

Security Classification of Abstract: UNCLASSIFIED

Date: May 1980

PROGRAM SUMMARY/STATUS

Short Program Title: Fire Control Simulation Requirements	*Report Security Classification:* Unclassified
Contract No. DAAK40-78C-0054 *Eff. Date:* *Exp. Date:* 1 Feb 80	*Project Mgr. Preparing Report:* G. K. Warmbrod *Contract Monitor:* N. Powell

Type of Report: Final *Report No.* CR-6-824 *No. of Pages in Rpt* 82

Inclusive Date of Reports January 1979 to December 1979

Key Words: Simulation, Laser, Fire Control, IFTRAN

ABSTRACT:

(U) A set of requirements for a HEL weapon fire control system simulation were generated in this study. A functional approach, taking advantage of the decoupling of implementation decisions from values of performance parameters, is recommended.

(U) A functional breakdown of the HELWS appears in the report, described in terms of inputs, sequencing, processing, and output of each function and subfunction.

(U) A demonstration version of the simulator has been implemented on the CDC 7600 at the ARC, using that facility's graphics capabilities as a primary input output device. A description of the capabilities of the simulation executive is given, and a test case example, along with input data and listings.

(U) Schedules for the simulation development are given and task descriptions appear as the last section of the report.

THIS PAGE LEFT BLANK INTENTIONALLY

(U) CONTENTS

THIS PAGE LEFT BLANK INTENTIONALLY

(U) FIGURES

TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1 (U) SIMULATOR DEFINITION

(U) The system engineering process, as set forth in MIL -STD 449A, [1] is a "logical sequence of activities and decisions transforming an operational need into a description of system performance parameters and a preferred system configuration." Similar definitions exist in other government documents and in the literature. Although sources differ in the terminology they use for describing these activities, all agree that the system engineering process should determine what actions a system is to perform, and how well, before deciding how it is to be implemented.

(U) The underlying requirement on the system simulator discussed in this chapter is that it will be used to support the study and analysis of what actions a HELWS system will perform, and how well it must perform these actions. Additionally, the simulator will later be used to validate candidate engineering implementations (by either hardware or software) to perform these actions.

(U) The sequence of activities in the system engineering process,which we assume for purposes of devising the requirements on the HELWS system simulator, are described in many sources. For instance, MIL - STD 499A defines three activities for establishing the performance and design requirements: (1) "mission requirements analysis", (2) "functional analysis" and (3) "function allocation". The logical sequence starts with setting system wide requirements based on mission objectives, proceeds to derive detailed performance and design requirements in terms of the system functions to be performed, and then allocates the functional requirements to subsystems. In a more theoretical context, Hall [2] sets forth a similar approach. One of his seven problem solving steps, problem definition, establishes relationships between overall goals and specific system requirements. Another, value system design, is concerned with the derivation of requirements in terms of functional characteristics of the system.

(U) This functional approach to requirements engineering has the advantage that it provides a means to analyze system behavior and set performance requirements, while postponing certain design decisions. The analysis proceeds by: (1) identifying system functions, (2) resolving functions to subfunctions, (3) establishing relationships among functions, and (4) building a functional model of the system. A function, as the word is used here, is a mathematical specification of a group of related actions in system behavior. The advantage of functional modeling is that it offers a high degree of generality and versatility. Systems may have significantly different physical components, but have essentially the same functional breakdown. We advocate using a functional approach in establishing performance and design requirements for systems involving technology programs such as HELWS because the result will be a decoupling of implementation decisions from choosing values of performance parameters. For example, selecting a type of acquisition radar is decoupled from selecting the value of performance parameters such as search range, sampling rate, and probability of detection. Consequently, the requirements can apply to numerous candidate radar configurations and modes of operation..

(U) In many cases the functional breakdown of systems are similar if they are solving the same problem, although the physical configurations may differ significantly. A simulation built along the lines of the functional breakdown is an excellent tool to help guide a technology program. The near term steps in the HELWS technology program are to:

- (U) Generate mission requirements
- (U) Define alternative constructs
- (U) Select the promising technology areas
- (U) Evaluate the technologies
- (U) Select a viable construct or constructs
- (U) Generate appropriate technology requirements, so that the system engineering process can proceed to design and prototype testing of the most promising concept.

(U) GRC has designed and used several flexible modular simulators that would be appropriate in the HELWS technology program. The present task uses this experience base, and an existing simulation framework, to meet the objectives of the study. These objectives were to:

- (U) define a detailed functional breakdown of HELWS
- (U) realize a simulator framework
- (U) define the requirements and interfaces for a threat driver
- (U) demonstrate a representative subset of the simulation
- (U) prepare a plan for further development of the simulator.

Subsequent sections of this chapter present our progress in meeting these objectives.

(U) In this reporting period GRC developed a preliminary hierarchical structure of HELWS functions, which is presented in Section 2. As high level HELWS functions we selected groups of system actions that have minimal coupling to implementation decisions. For instance, with this structure we can model search operation and derive inherent performance requirements, without concern for whether search and target track are performed by the same or different sensors. The functions discussed in Section 2 are applicable to all of the HELWS configurations that have been put forward. Alternative configurations can be realized by altering the mathematical forms of the functions. The functional breakdown is incomplete in areas such as the Weapon Operation function. Further work here must await the results of more detailed modeling of high powered lasers in a weapon configuration.

(U) Section 3 reports on the adaptation of a previously developed simulator that reflects the functions of the HELWS problem and has a high degree of independence from specific HELWS hardware configurations. We made use of an operational, GRC developed, functional simulator to provide the facilities of a simulation executive that has a HEL weapon system orientation and that utilized the interactive facility at the ARC computer as a primary communication vehicle with the analysts using the HELWS simulation. Although only a demonstration subset of the HELWS functions are implemented in the simulator, this subset with the executive software and the graphics capability for a full simulator are operational. Section 4 describes the development plan for functional simulation of the HELWS engagement process which would use the simulator to aid the establishment of performance requirements for each function in the engagement process. (U)

2 (U) FUNCTIONAL BREAKDOWN OF HELWS

(U) We analyzed generic characteristics of HELWS and chose the following functions as the principal first-order functions for modeling HELWS tactical operation:

1. (U) Threat Acquisition
2. (U) Precommit Track
3. (U) Threat Assessment
4. (U) Target Acquisition
5. (U) Post-commit Track
6. (U) Weapon Operation
7. (U) Kill Assessment
8. (U) HELWS Control

The first three determine the HELWS handling of the total threat and environment. The next four characterize the HELWS behavior in engaging a specific target. HELWS Control is a management function that coordinates the performance of all other functions so as to effectively neutralize the threat.

(U) The principal reason for selecting this group of functions is the degree to which they are decoupled from implementation decisions. For instance, we can model the search function and derive inherent performance requirements, without concern for whether search and track are supported by the same or different sensors. The functions in the above list are applicable to all of the HELWS configurations that have been put forward. We used them as the basis for defining the structure of a flexible simulator that can model the operation of various HELWS possibilities. Alternative configurations can be realized by altering the mathematical form of the functions. Variables in the mathematical expressions provide a means for studying the dependence of performance upon system parameters.

(U) Figure 2.1 is a schematic block diagram showing the first level functional structure of a HELWS. Of course, at this level of detail the particulars of system operation are not apparent. In subsequent paragraphs we elaborate our functional model by describing each of the functions in Figure 2.1 in more detail. In particular, we identify the input and output of each function in the process, and in many cases, the next level of functions.

2.1 (U) INITIAL THREAT ACQUISITION

(U) The initial acquisition function that we considered is based on a radar technology for realizing the sensor subfunctions. It has the following inputs and outputs:

Inputs (U)

- search returns
- system control (including timing)

Outputs (U)

- search pulses
- detection reports
- status

At this first level of decomposition, Initial Threat Acquisition has the structure shown in Figure 2.2. Acquisition Radar Sensing consists of the functions that define the operation of the radar as an externally controlled sensor. Signal Processing defines the electrical processes that detect search returns, extract their signal properties and estimate measurement parameters. Acquisition Processing accounts for the computational processes that control search radar operation, schedule search and verify actions and report the detection of targets.

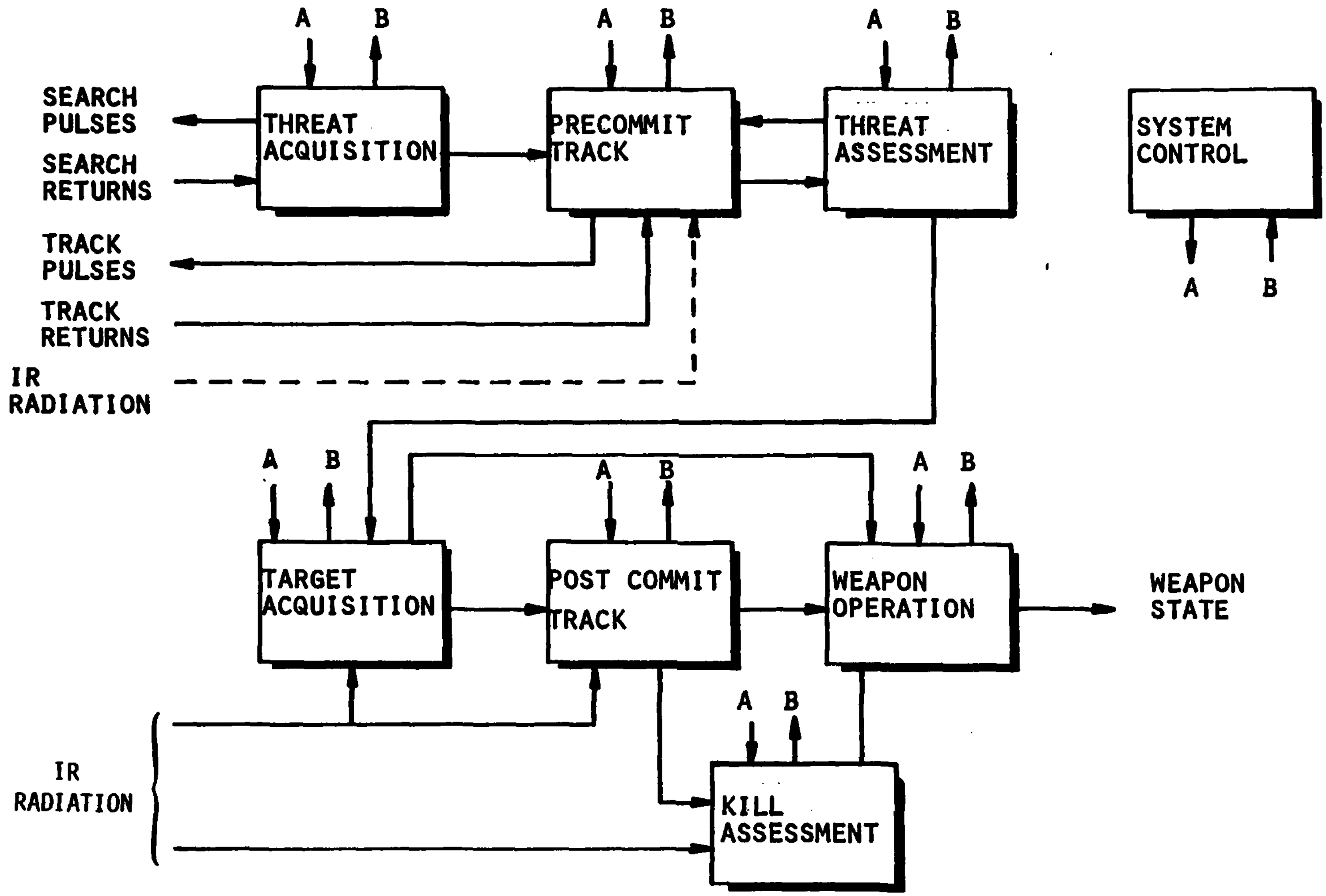


FIGURE 2.1 (U) FIRST LEVEL FUNCTIONAL BREAKDOWN FOR HELWS

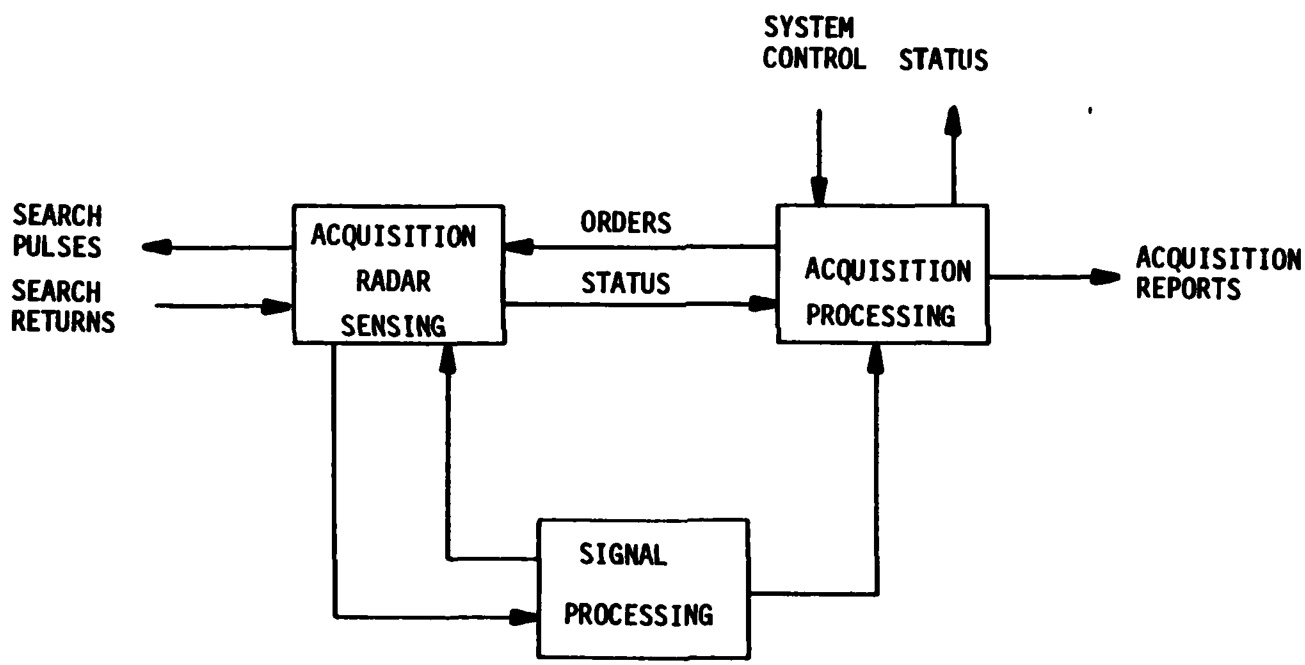


Figure 2.2 (U) First Level Schematic Block Diagram of Initial Acquisition

2.2 (U) PRECOMMIT TRACK

(U) In our initial HELWS functional breakdown the precommit track function can have either of two forms. One is based on accurate radar sensing only. The other is based on using less accurate radar sensing to track targets after acquisition, but using IR sensing to improve track accuracy before producing a handover report. They are distinguished by suffixing "I" designates radar only, and "II" optics augmentation. The inputs and outputs for Precommit Track are:

Inputs (U)

- Acquisition Reports (I, II)
- Track Returns (I, II)
- IR Radiation (II)
- Precision Track Command (I, II)
- System Control (I, II)

Outputs (U)

- Radar Pulses (I, II)
- Handover Messages (I, II)
- Status (I, II)

At the first level of decomposition for Precommit Track the structure has the form shown in Figure 2.3, where both alternatives (I and II) appear. In HELWS operation a track is established for each acquisition report. Track Processing acquires the data for initiating and maintaining tracks by issuing orders for radar pulses. It reports the track state vector as an output and also makes an association between acquisition reports and the state of all tracks to avoid establishing multiple tracks on the same target. Prior to target handover, upon command, Track Processing increases the data rate on the designated target to reduce handover errors, if needed.

(U) In alternative II, radar sensing is augmented by IR sensing. Track Signal Processing outputs pointing commands for the IR Sensing functions. IR Signal Processing produces angle and intensity measurement data as input to Track Processing.

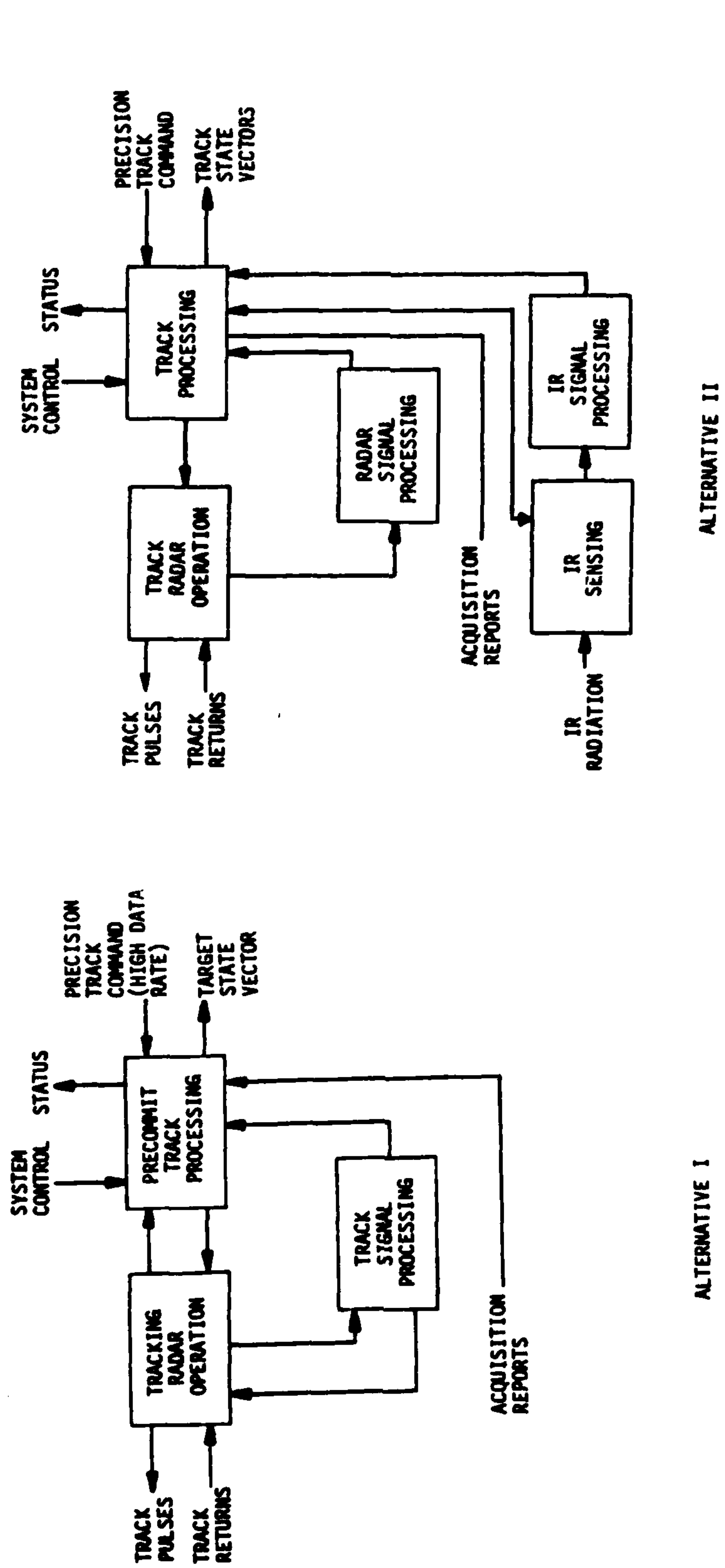


Figure 2.3 (U) First Level Schematic Block Diagram for the Precommit Track Function

2.3 (U) THREAT ASSESSMENT

(U) Threat Assessment is a coordination function that provides the logic for designating a specific target to engage with a laser weapon. The inputs and outputs are:

<u>Inputs</u> (U)

- State vectors of targets in precommit track
- System control

<u>Outputs</u> (U)

- Precision track message
- Handover message
- Status

The breakdown of Threat Assessment is shown in Figure 2.4.

(U) The Target Prioritization component monitors the state and conditions of targets as produced by Precommit Track. Target range and flight direction are factors in assessing the severity of a threat. Three other subfunctions play a supporting role. Target Identification makes use of signature data in the target state to estimate the target type. Kill Time Prediction assigns each target an expected duration for attack with the laser weapon. Impact Point Prediction estimates where the target could impact, data which may be used in Target Priortization to estimate which defended assets are threatened. Target Prioritization designates targets for precision track (either high data rate for a radar sensor or direct viewing by a passive IR or an electro-optical sensor). When all track error and discriminations criteria are met, a handover message is sent to the Post-commit Track and HELWS Control function.

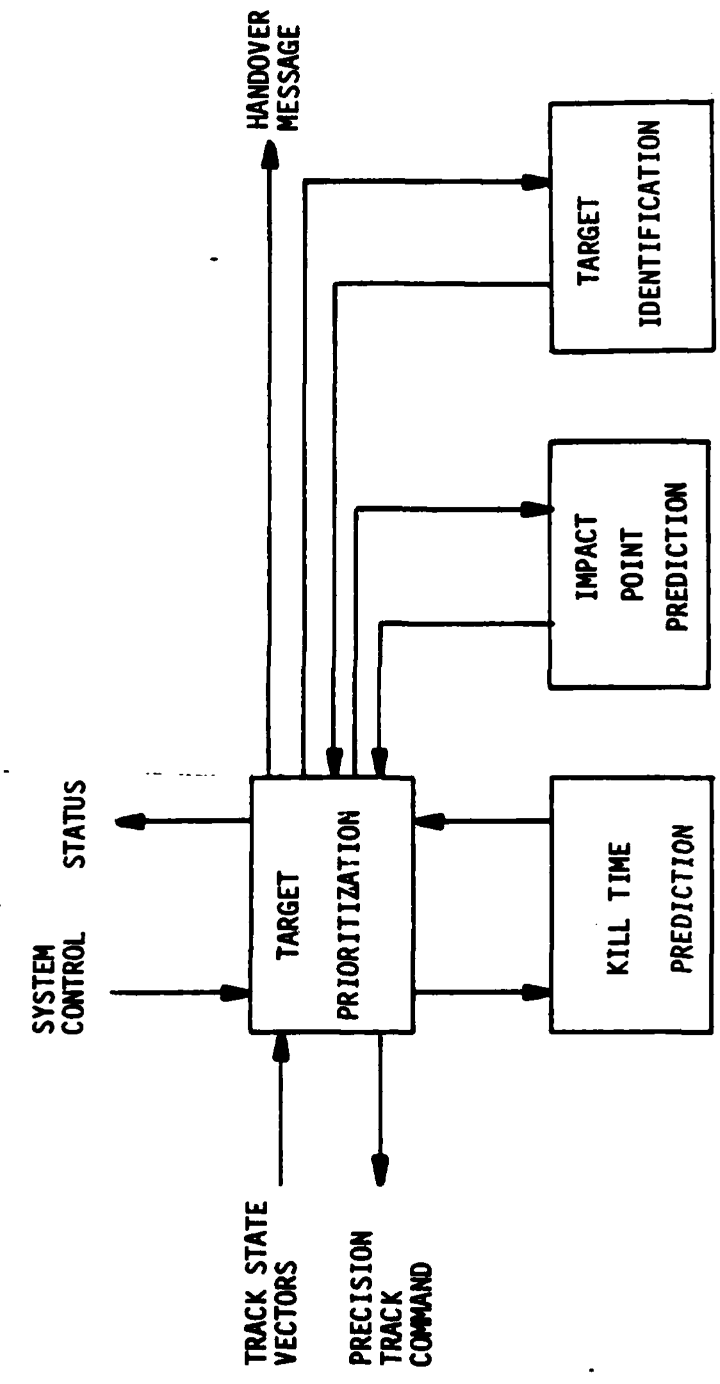


Figure 2.4 (U) Schematic Block Diagram of Threat Assessment Function

2.4 (U) TARGET ACQUISITION

(U) Using a handover message from Threat Assessment, Target Acquisition comprises the HELWS actions to point the weapon aiming sensors at the designed target and detect its radiation. The inputs and outputs are:

<u>Inputs</u> (U)

- Handover messages
- Radiation intensity

<u>Outputs</u> (U)

- Detection messages
- Aiming directions

Target Acquisition consists of three subfunctions; Search Control, Slew Control, and Detection, coupled as shown in Figure 2.5.

(U) Using information in the handover messages from Threat Assessment, Search Control generates an aimpoint for input to Slew Control. The detection subfunction processes sensor inputs and indicates target detection to Slew Control and to the Postcommit Track function when the target enters the field-of-view of the principal track sensor. Slew Control has an interface with the Weapon Control function, which directly controls the weapon.

2.5 (U) POSTCOMMIT TRACKING

(U) The aiming of a laser weapon is a cooperative process involving selecting and tracking of an aimpoint on the target, track of the actual point of irradiation by the laser, and correlation of the two track states to derive aiming signals for the laser. We define this process with the Postcommit Tracking function. Its inputs and outputs are:

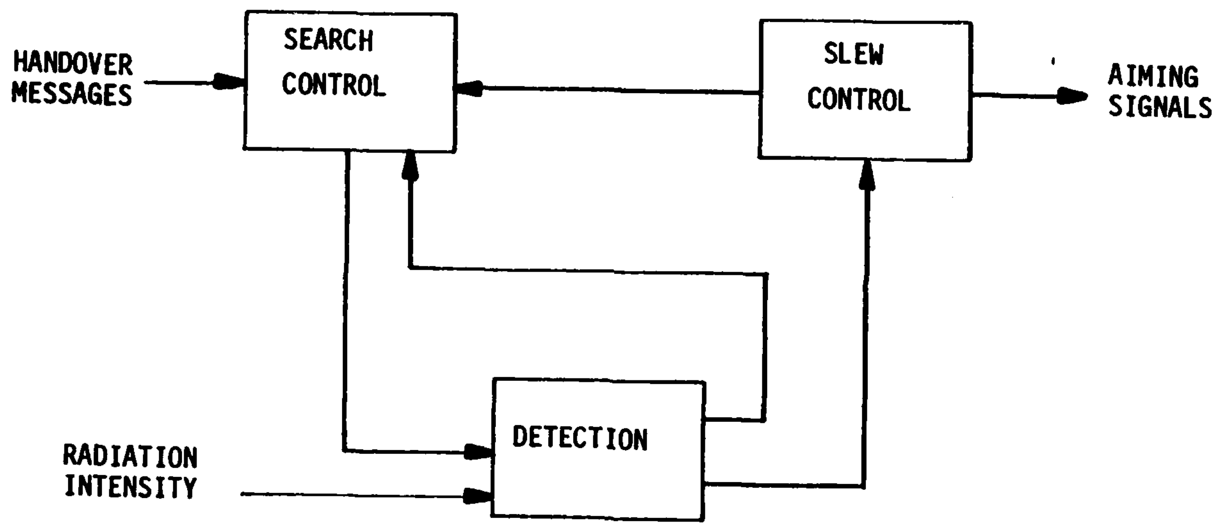


Figure 2.5 (U) Schematic Block Diagram of the Target Acquisition Function

<u>Inputs</u> (U)

- Target Radiation
- Hot Spot Radiation
- Detection Message
- System Control

<u>Outputs</u> (U)

- Target Illumination (optional)
- Weapon Aiming Signals
- Status

Postcommit Tracking uses passive radiation from the target (or optionally reflected energy from illumination) to locate and track an aimpoint on the target (and, optionally, to determine the far-field beam irradiance at the target). In our formulation of this function a sensor detects the hot spot created by the laser as an indication of the point being irradiated by the laser. An alternative or additional sensor may be employed that detects laser energy back-scattered from the target.

(U) The subfunctions of Postcommit Tracking and their interconnection is shown in Figure 2.6. The track processing subfunctions use inputs from the track sensing function to cooperatively produce an error signal for transmission to the Weapon Control function. Target Track Processing is supported by Aim Point Selection, a subfunction that designates the angular coordinates of the most lethal target region in view of the weapon. A hot spot tracking loop gives the angular coordinates of the actual dwell point of the weapon. Correlation Track includes tha algorithm for estimating the deviation between the two track vectors, and may be supported by additional algorithms which derive far-field data to be used to optimize the beam irradiance at the target. It also provides coordination among the subfunctions of Postcommit Tracking and interfaces these functions with System Control.

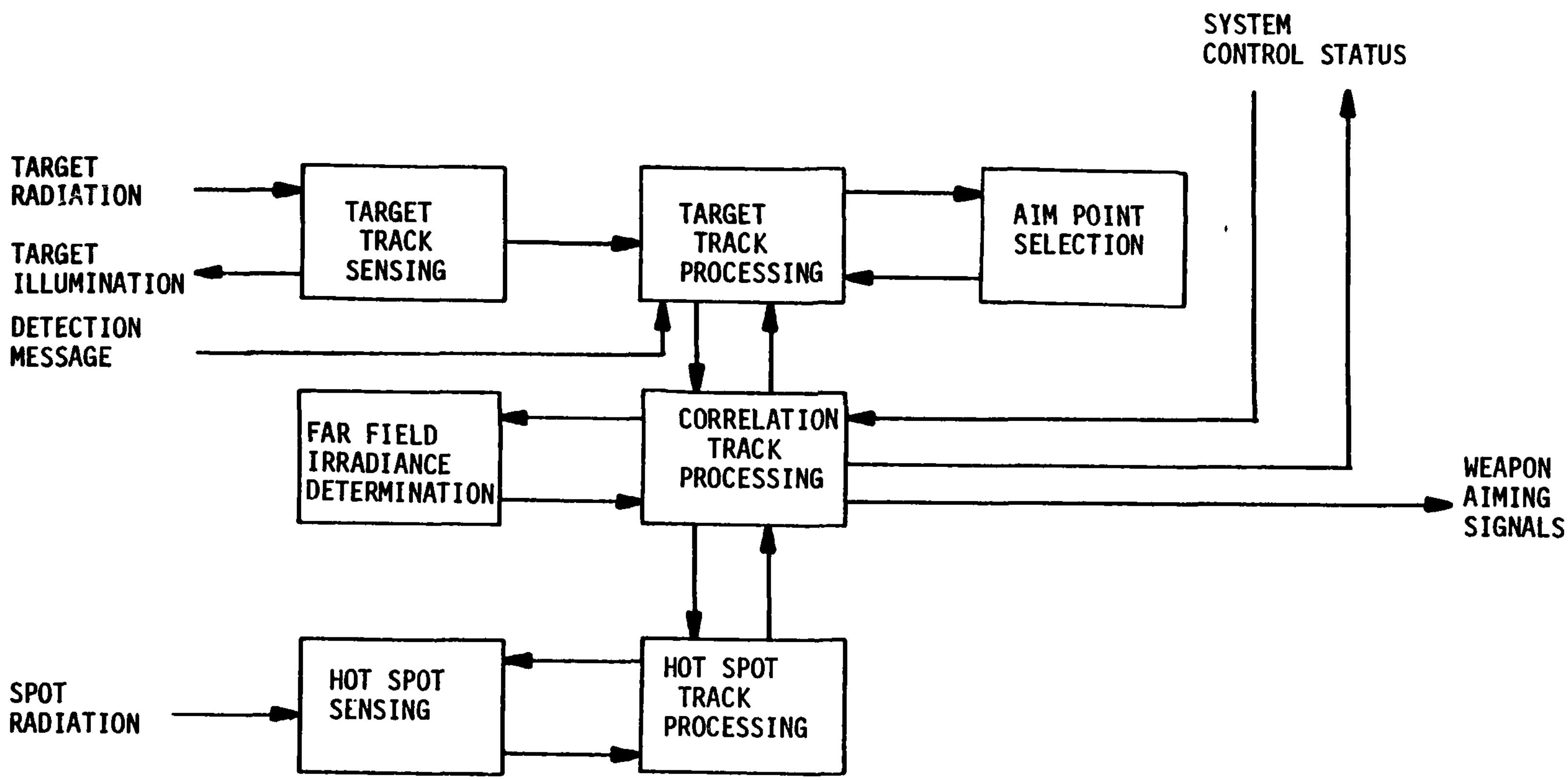


Figure 2.6 (U) Schematic Block Diagram for Pcst Commit Tracking

2.6 (U) WEAPON OPERATION

(U) Weapon Operation specifies the computational and physical processes that define the operation of the laser weapon and its support systems. The inputs and outputs for this function are:

Inputs (U)

- Start Firing Command
- Stop-Firing Command
- Aiming Signals
- HELWS Control

Outputs (U)

- Weapon State
- Status

Weapon Operation accounts for the closed loop positioning of the weapon and models the actual firing of the laser. Depending on the level of detail to which it is specified, this function can be very specific to a particular weapon. In our functional analysis we considered only very general properties. Consequently, the initial configuration of Weapon Control, as shown in Figure 2.7 is quite simple.

(U) Open Fire commands, which originate in Postcommit Track, are inputs to the Open Fire subfunction. Open Fire specifies the logic for assuring that all conditions (including safety) are met for triggering the weapon. Conversely, Cease Fire is a subfunction that can interlock the trigger and can halt a firing. It accepts inputs from the Kill Assessment function and from HELWS control. All modeling of the physical processes of the weapon are in the subfunction Weapon Dynamics, including adaptive optics. This subfunction also includes the computational and electronic processes for optimal beam control. The Laser phenomenology sub function accounts for the power generating properties of the laser and for the interaction of laser energy with the atmosphere. Much further work is needed for the Weapon Operation function, but must await the results of more detailed modeling of high powered lasers in a weapon configuration.

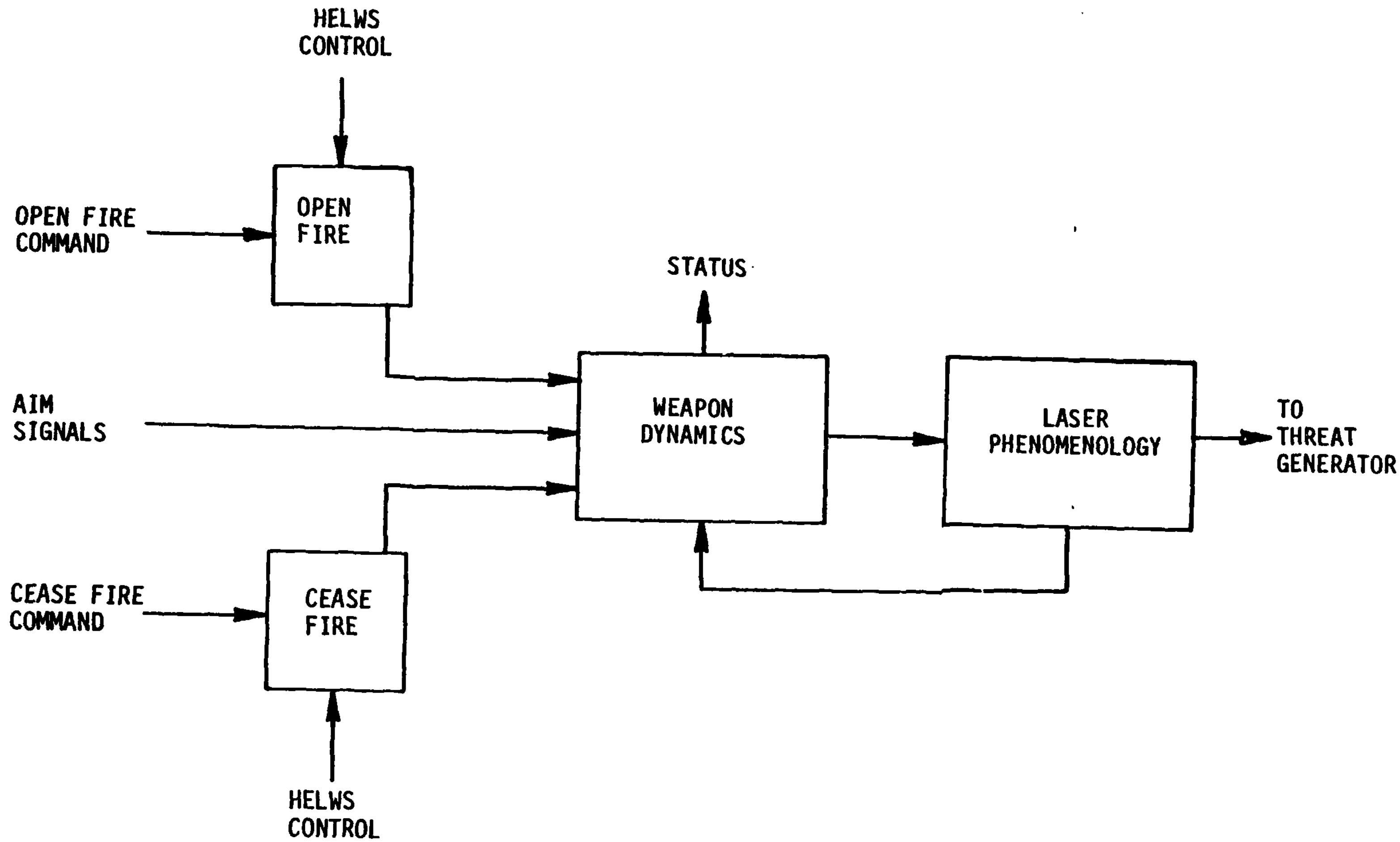


Figure 2.7 (U) Schematic Block Diagram of Weapon Operation Function

2.7 (U) KILL ASSESSMENT

(U) In many situations the tactical effectiveness of HELWS may depend critically upon recognizing that hits by the weapon have made a target nonthreatening. If a priori knowledge of target vulnerability to lasers is uncertain, the HELWS must be capable of recognizing those changes in the state of a target that indicate it is disabled. The functional breakdown includes Kill Assessment as a function that specifies the processes for causing the HELWS to disengage a target. The Kill Assessment inputs and outputs are:

Inputs (U)

- Targer radiation
- Target State Vector
- System Control

Outputs (U)

- Status

Our formulation of Kill Assessment recognizes two possible means for detecting a change in target states: (1) sensing a change in radiation from the target and (2) recognizing a perturbation in the target's trajectory. Figure 2.8 shows two subfunctions, Damage Sensing and Trajectory Computation; that account for the actions to detect these changes. Damage Assessment consolidates information for evaluating the benefit in continuing to fire at a target.

2.8 (U) HELWS CONTROL

(U) HELWS Control is not yet defined as a function. We expect to make it the resultant of such subfunctions as:

- (U) Battle Management
- (U) Engagement Logic
- (U) Multiple Battery and Intersystem C^3

Before proceeding to specify HELWS Control, we require a better definition of HELWS mission requirements and the operational procedures for engaging targets.

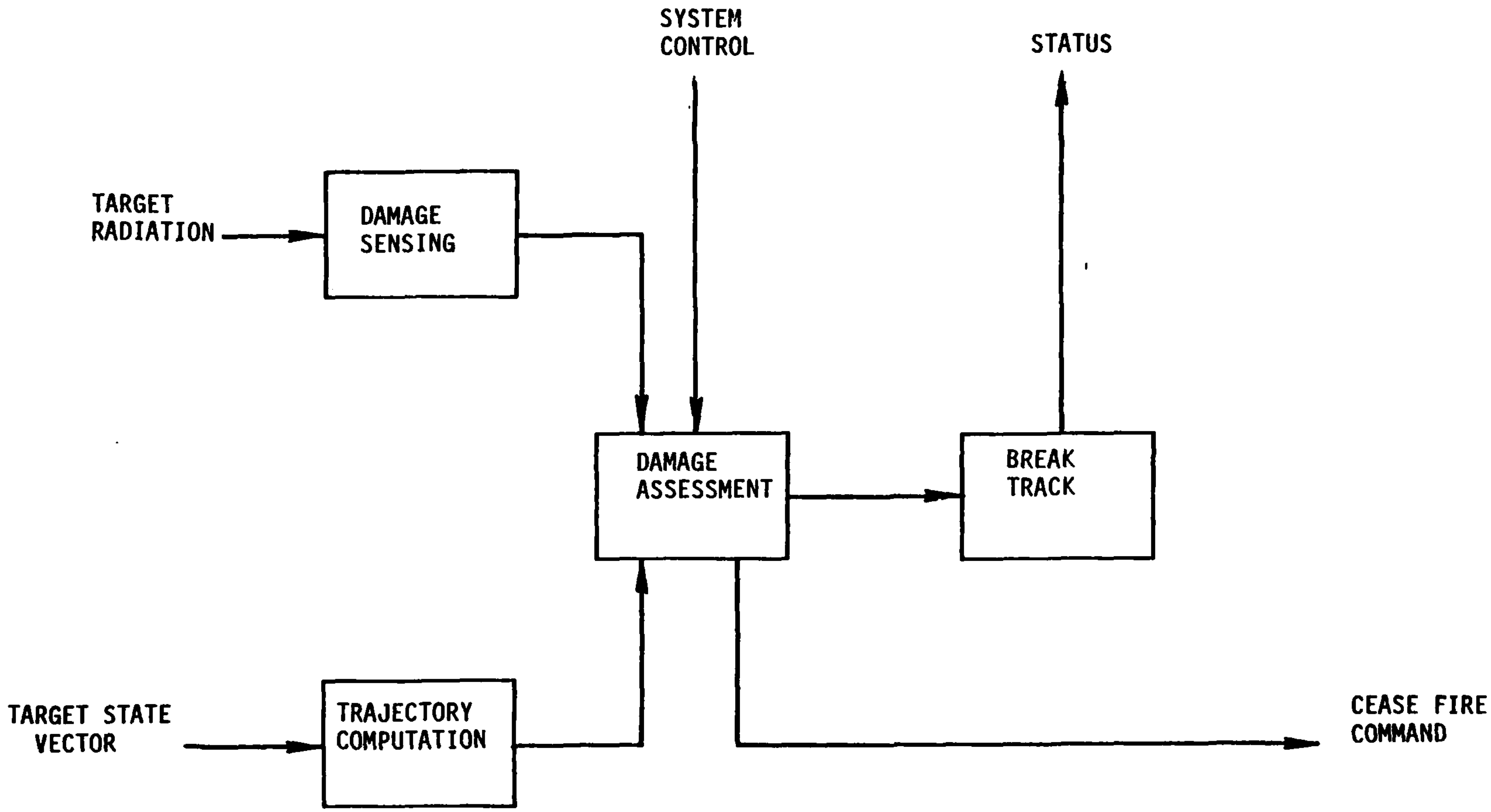


Figure 2.8 (U) Schematic Block Diagram of Kill Assessment

3 (U) SIMULATION FRAMEWORK AND FUNCTIONAL BREAKDOWN

3.1 (U) REQUIREMENTS

(U) The system engineering process we assume for HELWS, as for all weapon systems satisfying MIL STD 449A, includes functional analysis activities for deriving system/subsystem performance and design requirements. An important tool for performing this analysis is a simulation that allows HELWS to be modeled as the configuration of functions it performs, and which will support the determination of how well the functions collectively perform, given each functions' individual performance characteristics. The functional analysis will include both:

(1) (U) Changing the performance characteristics of the functions (i.e., - modifying the data or software algorithms representing that function in the simulation), and

(2) (U) Changing the configuration of functions (i.e., - modifying the sequencing, enablement, or flow of data between functions).

Since the analysts will want to evaluate the performance of both the individual functions and the ensemble of all functions, the simulation structure must allow the analyst to set measurement points between functions and/or data to be gathered within functions. Other desirable goals include modularity of functions and data so that processes that are in different stages of design can be accomodated, and applicability of the simulation at more equipment-specific levels of designs (such as the real-time data processing system). Additionally, the simulation software should aid the analysis effort by providing interactive graphics for I/O, providing an easy-to-use programming language that is FORTRAN compatible, and providing automatic data plotting software for the variables of interest to the analyst.

(U) The HELWS functional simulation discussed here consists of two parts: the collection of functions in the simulation, and the simulator structure and executive software. The structure and executive software of the simulator are an adaption of the Modular Missile-Borne Computer (MMBC) Simulation-Emulation Driver (SED) which was previously developed by GRC.

3.2 (U) SIMULATION-EMULATION DRIVER (SED)

(U) The basic goal for SED was that it be applicable at all of the design levels through which the MMBC real-time data processing system would evolve. Specifically it was designed to accommodate stages of evolution that begin with high-level functional definitions of the MMBC system's process where only input/output characteristics of the first-order functions in the process are known, and progresses all the way to a real-time software/hardware realization of the final system. Secondary goals for SED included that it be capable of accomodating simultaneously, processes in different stages of design.

(U) SED is resident in the Advanced Research Center's CDC 7600. It employs that facility's interactive color graphics (Anagraph System) to enable an analyst to control and examine the flow of processing of data that characterizes the behavior of the real-time system for which data processing requirements and specifications are sought. The SED software includes standarized plot packages so that performance data required by the analyst will be monitored by the SED during an exercise, and then be available for automatic plotting when called for.

(U) To achieve the goals of SED, the software structure requires; the analyst to decompose the target real-time processes into logically independent subprocesses (e.g. the hierarchical structure for HELWS described in Section 1.1), being careful to clearly separate computational elements within a subprocess from data transfer elements which transfer data between subprocesses. SED aids and encourages this decomposition by employing a multi-module overlay structure in which the analyst defines module boundaries at those points where he desires to modify and/or examine data. For large processes such as MMBC, the overlay structure provides the additional benefits accrued from economical memory utilization. The SED employs GRC developed support software including IFTRAN (a structured-programming language which is FORTRAN compatible, and Dynamic Storage Allocation (DSA). A detailed description of the basic SED program as configured for the MMBC signal processing functions is provided in Reference 3.

3.3 (U) FUNCTIONAL BREAKDOWN FOR HELWS DEMONSTRATION

(U) The deomonstration version of the HELWS functional simulator does not incorporate the hierarchical structure presented in Section 2. Instead, it was configured with certain subfunctions as the principal component functions. These were known to some degree at the outset of our study. The analysis leading to the integration into first level functions was conducted in parallel with the initial implementation of a simulation capability. Figure 3.1 is a flow diagram for the initial version of the flexible simulator. Incidentally, Figure 3.1 is an output on a graphics terminal, used by the analyst to define to the simulation executive software the sequencing of functions.

(U) At present, many of the functions in the demonstration simulation are implemented as dummy routines which require time, but functionally perform perfectly. The detailed requirements for a HELWS functional model were not part of this investigation. Nevertheless, the structure we have developed makes provisions for incorporating system functions as their descriptions become better defined. The major achievement during this reporting period was to modify the executive and display software of a previously developed simulator to interface with routines that represent HELWS functions. Subsequent paragraphs summarize the capability and features of this software.

3.4 (U) HELWS SIMULATOR PHYSICAL DESCRIPTION

(U) The HELWS simulator has the configuration shown in Figure 3.2. It resides on the CDC 7600 computer at the BMDATC Advanced Research Center. Code for the executive is written in IFTRAN, a version of FORTRAN that incorporates structured software. It consists of a main overlay containing commonly used routines and secondary overlays to perform specific independent executive functions. It interfaces with the threat generator that drives the simulator input and manages communications with the Anagraph Display System. The Anagraph is an interactive terminal set consisting of a 19-inch color CRT, keyboard and trackball positioned cursor. The terminal includes a black and white hard-copy device.

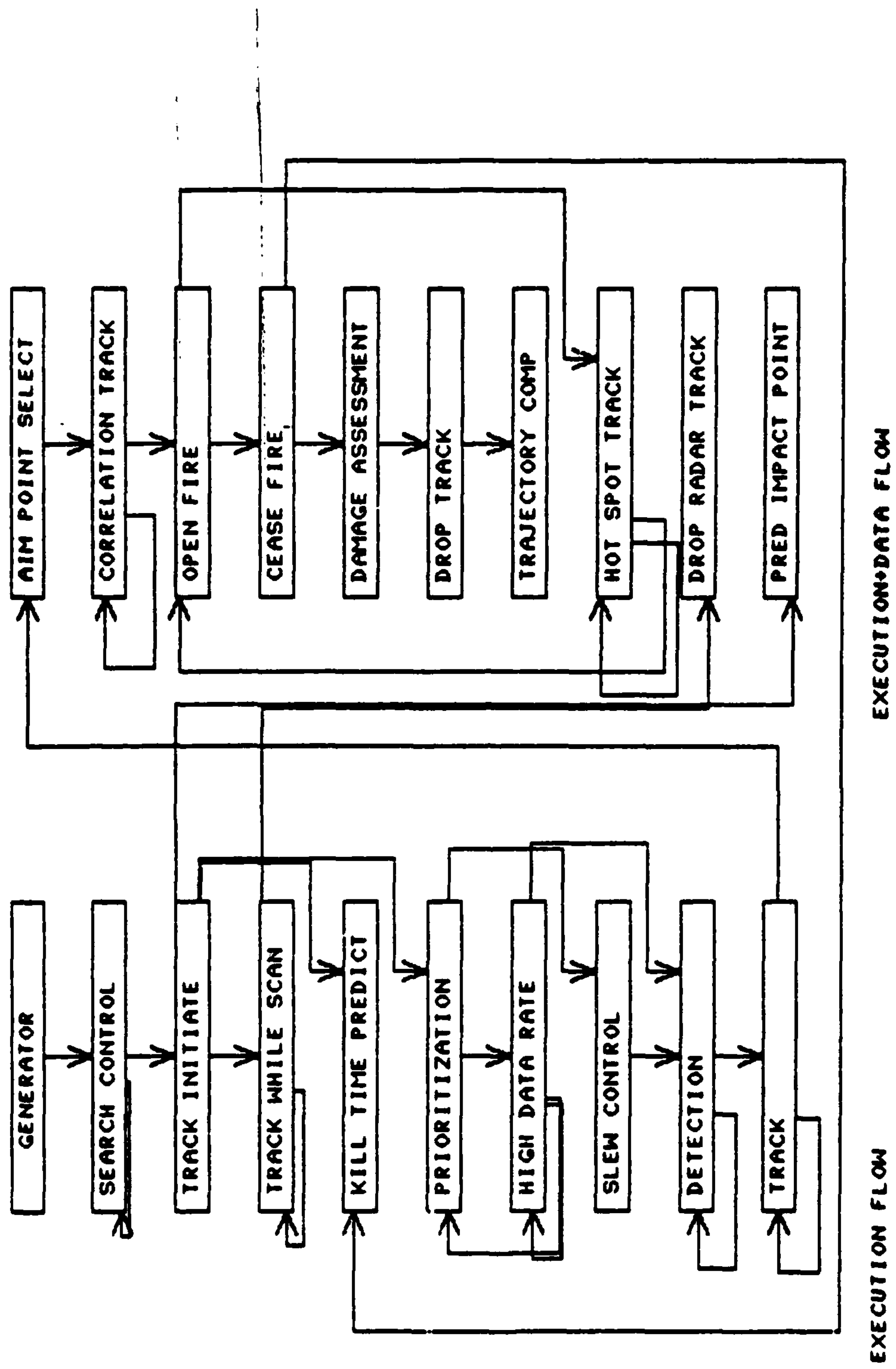


Figure 3.1 (U) Flow Diagram of Initial Version of a HELWS Functional Simulator

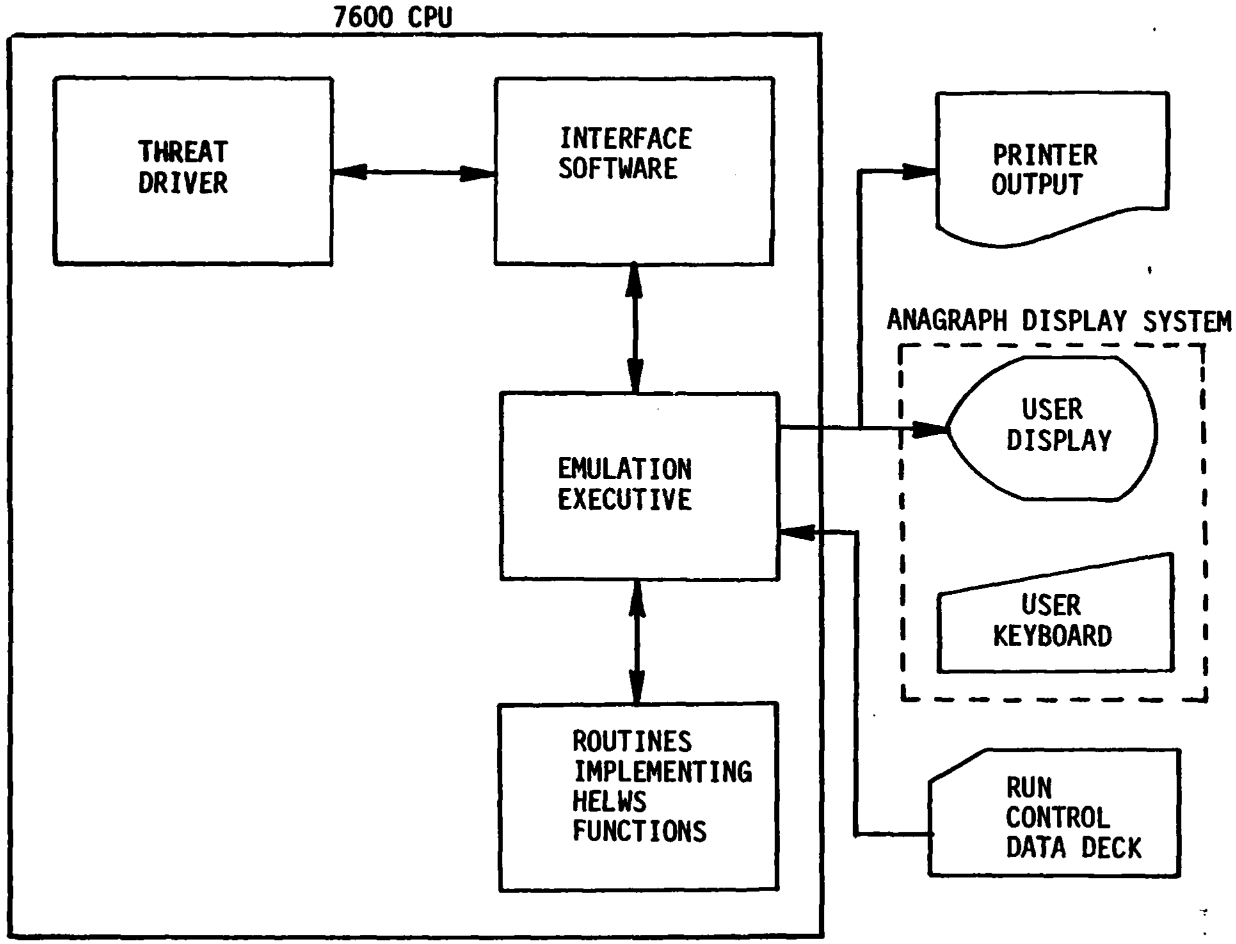


Figure 3.2 (U) Physical Configuration of HELWS Functional Simulator

3.5 (U) EXECUTIVE DESCRIPTION

(U) The HELWS simulator executive implements a variety of functions that help the analyst coordinate operation of the simulator. These include:

- (U) Interpreting user commands
- (U) Reading in data from the threat driver
- (U) Controlling flow of data through simulator
- (U) Bookkeeping
- (U) Performance monitoring

Having most service routines in the executive enhances the flexibility of the simulator. Changes in input-output formats do not impact the routines that implement HELWS functions. Conversely, altering or replacing a HELWS function has minimal impact on executive routines. A listing of the main driver control logic may be found in Appendix A.

(U) The analyst will have at his command various instructions which allow him to control and monitor the execution simulator. The following is a list of the most powerful commands:

- (U) Modify Input - The command allows the user to change any of the inputs of a functional module, any functional module constant or executive constant.
- (U) Change Model - Allows the user to dynamically change which version of a functional module will be executed. For example, a user could choose a track while scan function for Precommit Track or, alternatively, a dedicated tracking function.
- (U) Display Input - This command allows the user to examine all of the inputs to a functional module prior to its execution.
- (U) Display Output - This command allows the user to display the modules output, or to request a printed or plotted performance monitor output. It can also be used to request a display of the executives summary statistics.
- (U) Set Breakdown - This command lets the user regain control of the executive at the end of a designated functional modules execution. The user may then use any of the other commands to examine or change information.

(U) • Next Scan - This command causes the next scan data set to be read.

(U) • Continue - Signals the executive that execution may resume. This is the command that gives control back to the executive after stopping for a breakpoint.

(U) • Stop - Causes run termination.

(U) • Read Threat Generator Outputs - The Emulation Executive will read in the data from the next scan of a sensor. The program will assign space for the data, and prepare it for the first functional module; for instance, "Search Control."

(U) • Control Flow of Data Through System - The Emulation Executive will control the flow of data through the system as it passes from one functional module to the next.

(U) • Bookkeeping - The Executive will keep track of the status of each functional module. Among the item will be:

1) The current module type

2) Whether it is active or idle, and if active, the wait queue length for the module

3) The amount of data transferred between modules, and

4) Simulate Clocks for each module, along with active/idle time ratios

(U) • Performance Monitor Data - Performance Monitor Data will be collected for each functional module and printed on the normal output file. Upon user request, the data may also be displayed in printed or plotted form on the graphic terminal.

(U) The Simulator Program is submitted over the counter at the ARC as a batch job requiring a graphic display terminal. The run deck consists of the control cards specifying the load modules and output files for the run, along with the input data defining the initial configuration and nominal values for the functional modules. The user at the graphics terminal then controls the execution of the simulator. He can cause the next scan to be read from threat driver output files, change input value, dynamically change which functional module is to be used, display overall status, or examine the performing monitor data for any of the modules. Each scan data set produced by the threat driver will be processed by acquisition and track modules. As the data sets pass through the system, if the next functional module encountered is a simulation, it will be executed immediately by calling in the appropriate overlay. (U)

(U) After all return data sets are processed, the data set furthest along is taken off of the queue and the simulation overlay for it is brought in and executed.

3.6 (U) THREAT DRIVER

(U) The role of the threat driver in the simulator is to generate a record of target dynamics and characteristics at each instant in time when the inputs to the sensors in the system are to be updated. The threat driver "flys" all of the objects in the threat and produces a "shot" of the aspect-dependent scene in the field of view of each sensor. In a detailed functional simulator, depending on the sensor, it accounts for radar cross section, IR emmissive properties, and the emmissive and reflective characteristics in the passband of other electro-optical sensors.

(U) The simulation makes data requests by sending a message to the threat driver. The message identifies the sensor type, location and field of view and, if applicable, the illumination beam description. The threat driver returns a data record consisting of the radiation from the targets within the sensor field of view. The executive sends the record to the routine that models operation of the sensor. In the demo, the threat driver is an algorithm which determines the positions of all targets $(X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}, \ddot{X}, \ddot{Y}, \ddot{Z})$ as a function of time. It is represented in Figure 1.9 by the box, "GENERATOR".

(U) The complexity of the threat driver will depend in large measure upon the fidelity with which targets must be modeled. For instance, the trajectory of an ARM might be approximated by a series of constant velocity flight segments, or it might be generated by a 3 DOF simulation, or even a 6 DOF simulation. Eventually, the driver should operate at several levels of fidelity. For example, the multiple target input data for the Initial Acquisition Function can be of relatively low fidelity, whereas, the single target data for cross correlation track must come from a very high fidelity model. Generating the input signals for imaging sensors is expected to present the most stressing signature modeling requirements for the threat driver.

3.7 (U) HELWS FUNCTIONAL MODELS

(U) Figure 3.1 shows a display produced by the SED executive giving the execution and/or data flows between the HELWS functional models currently defined. Each of these functional models consist of two parts: 1) model control, and 2) model algorithm.

3.7.1 (U) Model Control

(U) Each individual function model is an overlay in the SED design, and the main program of the overlay is in charge of executing the proper code for the model under test. The Integer variable comtype, in common/control/, is used to determine the purpose of this call, as shown in Figure 3.3.

3.7.2 (U) Model Algorithm

(U) The algorithms for the functional models of HELWS have, in some cases, been implemented as time delays, in others, as bookkeeping devices, while the rest are derived from the algorithms used in the COMO simulation of a HELWS.

COMTYPE	RESPONSE OF CONTROL
-1	Read in namelist data for model
0	Perform model initialization
1	Execute model algorithm
2	Display the model output data
3	Display the model input data
4	Modify the model input data
5	Destroy the input to model
6	Reformat the input data for hardware/emulation versions of model (as appropriate)
7	Reformat the output data from hardware/emulation versions of model (as appropriate)
8	Record model input so that it may be rerun at a later time

Figure 3.3 (U) Response of Model Control to COMTYPE

(U) An example of the logic implemented for one of the functional algorithms is prioritization. A logic flow chart for this model can be found in Figure 3.4 , and a listing of this functional model, along with its control can be found in Appendix B.

3.7.3 (U) Test Case

(U) In order to verify the logic and algorithms for this version of HELWS, a test case consisting of 4 objects flying straight line trajectories with no intermediate maneuvers was selected. All of the objects were to impact near the laser, but only one of them to impact in the laser defended region. The actual input cards for the test case are shown in Figure 3.5 , and this information, along with other SED Executive and system wide parameters output is shown in Figure 3.6.

(U) The output from the simulation for this test case, 50 seconds in duration with all debug output turned on, will be a printout several inches thick. Instead of presenting the entire output, selected interesting portions of the output may be found in Appendix C. In addition, a summary of the status of each functional model at the end of execution of the test case including the number of times executed and the number of words of data transferred between models may be found in Figure 3.7 . The same display as produced for the user as the interactive graphics terminal is shown in Figure 3.8 .

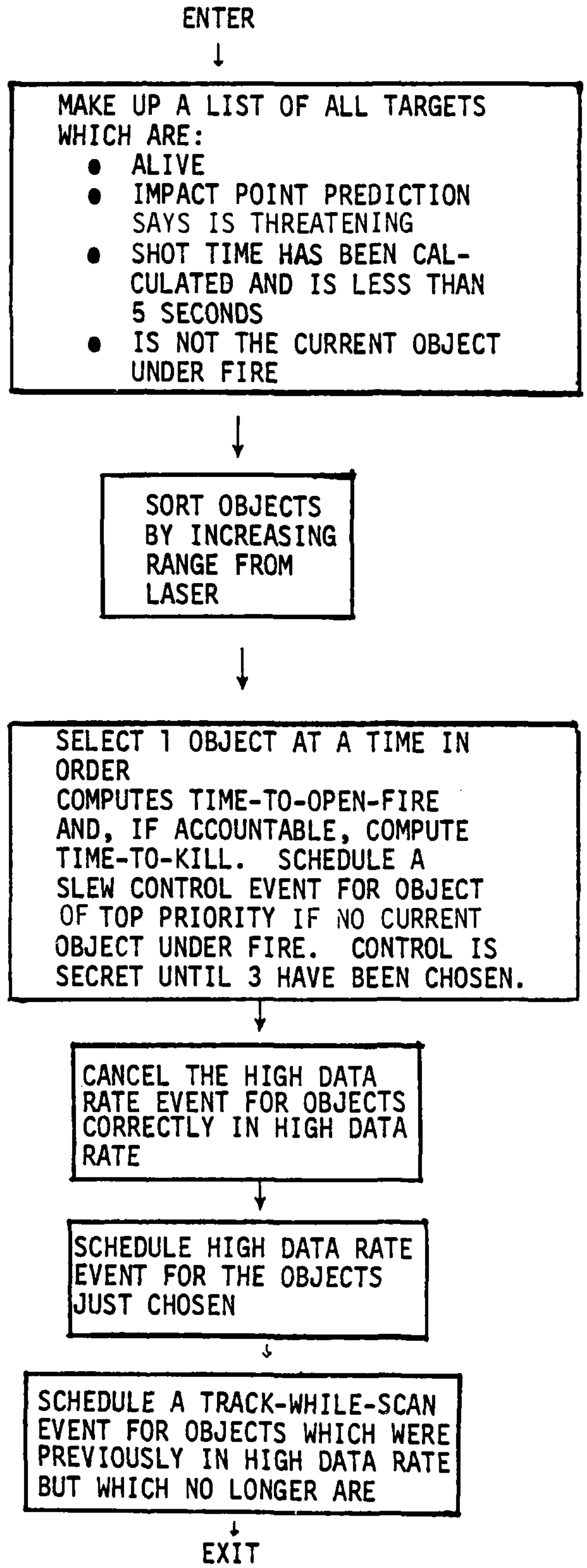


Figure 3.4 (U) Logic of Prioritization Functional Modes

```
THIS IS A STEP IN THE ADAPTION OF MMBS TO FIRE CONTROL
 $EXEC     IPRINT(1)=20*2,TIMLAST=51.,
  NOTARS=4, XPOSI(1)=5000., -5000., 5000., -5000.,
   YPOSI(1)=2*5000., -5000., -5000., ZPOSI(1)=4*5000.,
   XVELI(1)=-160., 163., -158., 165., YVELI(1)=-99., -95., 94., 106.,
    ZVELI(1)=-100., -98., -101., -99.   $
EXECUTE
```

Figure 3.5 (U) Test Case Input

```
$EXEC
INPUIN  = 30,
NCON    = 9,
XCON    = 9*.5,
YCON    = 8.399999999999998, 7.5, 6.599999999999999, 5.699999999999999, 4.800000000000001, 3.900000000000001, 3., 2.099999999999999, 1.2,
NSYS    = 20,
XSYS    = 20*9.5,
YSYS    = 8.5, 8.199999999999999, 7.900000000000001, 7.599999999999999, 7.300000000000001, 7., 6.699999999999999, 6.400000000000001,
          6.099999999999999, 5.800000000000001, 5.5, 5.199999999999999, 4.900000000000001, 4.599999999999999, 4.300000000000001, 4., 3.7,
          3.400000000000001, 3.099999999999999, 2.8,
MODULE  = 20*1,
MAIN    = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
NLED    = 5,
XLED    = 5*9.,
YLED    = 2., 1.5, 1., .5, 0.,
LCOLOR  = 0, 4, 6, 2, 5,
ISTOP   = 2,
ICTOP   = 0,
ISSIDE  = 3,
COLOR   = 1, 2, 0, 4, 3,
ISCON   = 7,
ICCON   = 1,
ICAFOV  = 1,
IPRINT  = 20*2, 10*1.
MAXLCM  = 10000,
MAXSCM  = 10000,
IFLAGS  = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
IDSPLA  = 20*1,
IBAKPER = 6,
ILFTPER = 1,
```

Figure 3.6 (U) SED Executive and System Wide Parameters

```
IBAKSUM = 3,
ILETSUM = 0,
OUTTASK = 0,
TIMLAST = 51.,
INFEAD  = 0,
DT1     = 1.,
DT2     = .5,
DT3     = 1.,
DT4     = .1,
DT5     = .5E-01,
DT6     = .1,
DT7     = .25,
DT8     = .25,
DT9     = .25,
DT10    = 0.,
DT11    = 1.,
DT12    = .1,
DT26    = .1,
XRADAF  = 200.,
YRADAF  = 0.,
ZRADAF  = 0.,
RACRAD  = 100.,
NLASER  = 1,
RACLAS  = 100.,
XLASER  = 1000.,
YLASER  = 1000.,
ZLASER  = 1000.,
NOTAPS  = 4,
XPOSI   = 8000., -6000., 5000., -9000., 5600.,
```

Figure 3.6 (U) (Cont.) SED Executive and System Wide Parameters

```
YPOSI   = 2*5000., 2*-5000., 36*0.,
ZPOSI   = 4*5000., 36*0.,
XVELI   = -160., 163., -158., 165., 36*0.,
YVELI   = -99., -95., 94., 106., 36*0.,
ZVELI   = -100., -98., -101., -59., 36*0.,
NOMANS  = 0,
TMAN    = 100*0.,
NTMAN   = 100*0,
XVMAN   = 100*0.,
YVMAN   = 100*0.,
ZVMAN   = 100*0.,
$END
```

Figure 3.6 (U) (Cont.) SED Executive and System Wide Parameters

DATA WORDS TRANSFERED	PERCENT IDLE TIME	CLOCK TIME (MIPS)	STATUS	EXECUTION CYCLE	QUEUE LENGTH	FUNCTIONAL MODULE
52	0.000	.026	IDLE	52	0	SEARCH CONTROL
103	0.000	.003	IDLE	3	0	TRACK INITIATE
364	0.000	.010	IDLE	14	0	TRACK WHILE SCAN
104	0.000	.005	IDLE	4	0	KILL TIME PREDICT
28	0.000	.006	IDLE	5	0	PRIORITIZATION
61138	0.000	1.219	IDLE	838	0	HIGH DATA RATE
78	0.000	.062	IDLE	3	0	SLEW CONTROL
26	0.000	.001	IDLE	1	0	DETECTION
78	0.000	.002	IDLE	3	0	TRACK
26	0.000	.001	IDLE	1	0	AIM POINT SELECT
10244	0.000	.207	IDLE	334	0	CORRELATION TRACK
26	0.000	.001	IDLE	1	0	OPEN FIRE
26	0.000	.001	IDLE	1	0	CEASE FIRE
26	0.000	.001	IDLE	1	0	DAMAGE ASSESSMENT
26	0.000	.001	IDLE	1	0	DROP TRACK
26	0.000	.001	IDLE	1	0	TRAJECTORY COMP
754	0.000	.021	IDLE	29	0	HOT SPOT TRACK
8	0.000	.001	IDLE	2	0	DROP RADAR TRACK
104	0.000	.002	IDLE	4	0	PRED IMPACT POINT

Figure 3.7 (U) Summary of Functional Model Execution for Test Case

FIRE CONTROL ENGINEERING SIMULATION

DATA WORDS	PERCENT	CLOCK TIME	STATUS	EXECUTION	QUEUE	
TRANSFERED	IDLE TIME	(MIPS)		CYCLE	LENGTH	EXECUTIVE
52	0.000	.026	IDLE	52	0	SEARCH CONTROL
100	0.000	.003	IDLE	3	0	TRACK INITIATE
364	0.000	.010	IDLE	14	0	TRACK WHILE SCAN
104	0.000	.003	IDLE	4	0	KILL TIME PREDICT
20	0.000	.006	IDLE	5	0	PRIORITIZATION
61138	0.000	1.219	IDLE	880	0	HIGH DATA RATE
78	0.000	.002	IDLE	3	0	SLEW CONTROL
26	0.000	.001	IDLE	1	0	DETECTION
78	0.000	.002	IDLE	3	0	TRACK
26	0.000	.001	IDLE	1	0	AIM POINT SELECT
10244	0.000	.287	IDLE	394	0	CORRELATION TRACK
26	0.000	.001	IDLE	1	0	OPEN FIRE
26	0.000	.001	IDLE	1	0	CEASE FIRE
26	0.000	.001	IDLE	1	0	DAMAGE ASSESSMENT
26	0.000	.001	IDLE	1	0	DROP TRACK
26	0.000	.001	IDLE	1	0	TRAJECTORY COMP
754	0.000	.021	IDLE	29	0	HOT SPOT TRACK
8	0.000	.001	IDLE	2	0	DROP RADAR TRACK
104	0.000	.002	IDLE	4	0	PRED IMPACT POINT

CDC 7600 CPU TIME = 5.233

MODULE LEGEND

SIMULATION

EMULATION

HARDWARE

NOT IN SYSTEM

Figure 3.8 (U) Screen Display of Test Case Summary

4 (U) SIMULATOR DEVELOPMENT PLAN

(U) A flexible functional simulator and an associated threat generator would be of great benefit to the HELWS system engineering process. They would be effective tools for deriving the performance and design requirements for a HELWS. Presently, however, only the executive software and the graphics capabilities are operational; only a demonstration subset of the HELWS functions are implemented in the simulator. Further, the threat generator now provides only trajectory information for point targets To be fully useful for driving a HELWS simulator, it requires the capability to model the finite extent, shape and radiation properties of targets and to represent susceptibility to damage by a laser.

(U) In this reporting period we prepared a plan for continued developement of the simulator and threat generator. Our analysis also considered the tasks that must proceed hand-in-hand with work on the simulator and the threat generator. We recognize that continued development requires:

- (U) establishing detailed Mission Requirements for HELWS
- (U) modeling the properties and characteristics of likely targets
- (U) formulating specific threat sensors
- (U) preparing algorithms for battle management of engagements with laser weapons

The total plan is costly in time and effort. Consequently, we have phased activities so as to produce an inital operational capability quickly and at modest cost. Subsequent phases of the program provide more advanced and complete capability.

(U) The simulation plan consists of two parts. Section 4.1 presents a work breakdown in the form of task descriptions. Section 4.2 presents a schedule for achieving increasing levels of capability in a flexible HELWS simulation and threat generator.

4.1 (U) TASKS FOR SIMULATOR DEVELOPMENT

(U) The work breakdown for continued simulator development has five major tasks. Two of these account for the effort to build a flexible simulator and a threat generator. Two others continue the modeling activities to produce a functional breakdown of HELWS and to generate models of specific elements in the threat. The fifth task is the mission analysis to identify typical battlefield deployments for laser weapons and strategies for using them and to establish the threat environment. As illustrated in Figure 4.1, the work breakdown has a hierarchial structure if viewed in terms of where each task gets its inputs. Subsequent paragraphs of this section present additional detail about the objective of each task.

(U) The mission requirements analysis task will establish the scope of alternative missions for laser weapons. This effort will define the threat that each mission is to meet and identify the criteria for neutralizing the threat with a laser weapon. The output of this task is a statement of the system objectives, and criteria for mission success. Using the mission analysis output, alternative constructs are formulated in terms of functions to be performed and their sequencing. Note that this need not involve a decision about which laser technology, repetitively pulsed or chemical, is in the construct. At this point, the simulation tools developed become part of the mainstream of the technology program: the framework would be ready at this point in the program to accept models of the functions which have been developed by other technology contractors (e.g. beam control).

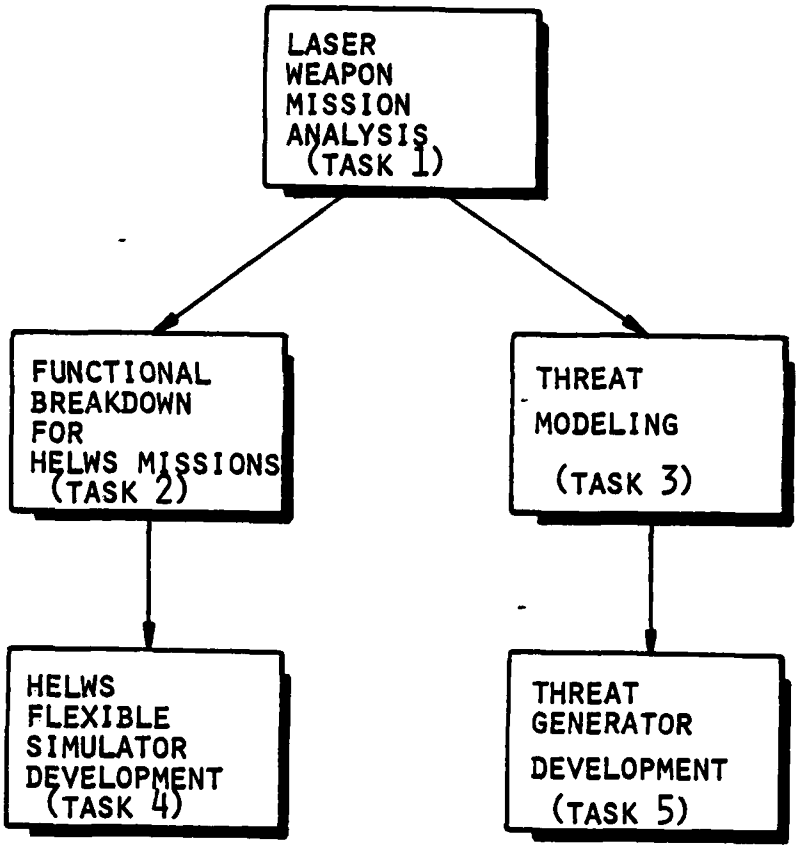


Figure 4.1 (U) Work Breakdown for HELWS Simulation Program

(U) At this point the level of fidelity in the simulation is important to the goals of the analysis. We will use the terms and definitions for the models of the functions presented in Table 4.1. It is clear from the table that we envision the simulation effort as an evolving, continuously useful tool. Outputs from the function performance level are necessary inputs to concept formulators who choose and describe generic components of possible constructs. Some concepts can be quickly eliminated on the basis of the performance requirements being well beyond current projected capabilities.

(U) The survivors of the first phase of simulation at the function performance level are modeled at the component performance level. These candidates are optimized by tradeoffs at this level of detail and compared. Much significant data is generated at this time that will greatly aid in the selection of a preferred construct which can be designed and evaluated using the final component simulation/emulation version.

(U) Note that it is at the last stage of the effort that one complex, component simulation is built. Such models are difficult to change, long running, and require extensive memory. Such a simulation is built only when the program is mature, and the specifics of the design are known.

(U) The approach to using simulation in a technology program outlined above initially allows a broad hack at many possible solutions, a more detailed analysis of a few solutions, and thorough testing and evaluation of components and software via simulation of preferred constructs during development. Feedback between levels is likely, as problems are uncovered, and the framework and previously used models survive for such use for the life of the program.

TABLE 4.1

(U) MODEL LEVELS DEFINITIONS

Level	Definition	Results of Analysis
Function Performance	Calculates what happens to an object as a function of its kinematics, and what happened in previous functions. Largely stochastic in nature, i.e. draw from probability of detection vs. range for an object entering field of view.	Determine performance requirements for each function, independent of any component, generate approximate timelines, error (leakage) budgets. Minimal interaction between functions, except in terms of timing, and leakage.
Component Performance	Functions are represented as components such as sensors, that have certain capabilities as a function of the engagement state; i.e. an optical sensor is represented by a set of sensitivity parameters, a scan pattern, and a noise model. which interacts with an object to determine S/N, and thus probability of detection.	Determine whether the performance requirements of the components play together to give overall construct performance stipulated. Uncover logical and interface problems, and do performance tradeoffs.
Component Simulation or Emulation	Calculates at a subcomponent level, as a function of the engagement state, what the signals and messages are passed through each component. Models based on design parameters of components. Interfaces between components, and subcomponents very accurately modeled.	Detailed evaluation of hardware and software design, evolving to hardware in the loop testing and software verification in a mature program.

(U) Although we constructed a functional breakdown for HELWS, as reported in Section 1.1, the detail of function specifications is presently inadequate for realization in a simulator. Task 2 provides functional analysis to extend and detail the functional breakdown. Initially, emphasis will be given to complete specification of the basic sensor and sensor processing functions. Subsequently, emphasis will shift to control and coordination functions and subfunctions.

(U) The threat modeling task (Task 3) will upgrade models for targets that are contained in the threat. It will account for dynamic behavior, radar and optical scattering characteristics, and IR radiation properties of the targets. Further, this effort will produce the detailed specification of specific threat scenarios that are anticipated for HELWS missions.

(U) Current work has produced the execution software and graphics capability for a HELWS simulator. We have as yet implemented only a few system functions in computer code. Task 4 includes the effort to design and implement software routines that realize the functions identified in the HELWS functional breakdown. Our objective is to build these routines so that they can be readily replaced by alternative specifications of a function.

(U) Task 5 produces the threat generator for driving the HELWS functional simulator. It is a program that responds to simulation commands for the radar signal or optical/IR radiation from the direction that a sensor is pointing. It implements models of specific targets and moves targets according to the specification of threat scenarios and the actions of the weapon system.

4.2 (U) DEVELOPMENT SCHEDULE

(U) As shown in Figure 4.2 the development schedule covers three years, and has three distinct phases. This program plan provides a simulation capability that increases incrementally. The product of Phase I is an initial version of the flexible simulator and the threat generator. The HELWS fire control problem is modeled at a high level with limited fidelity. The output of Phase II is a baseline simulation capability that models targets and HELWS functions with greater fidelity. Phase III produces an advanced version of the flexible simulator and the threat generator. This advanced simulation capability includes representing battle management and responsive threats.

(U) The initial version of the flexible simulator will contain representations of the major HELWS function, as identified in the functional breakdown. Some of the subfunctions, however, will be realized only in simple form. For instance, Laser Operation will model ideal beam control and Battle Management will take account of only simple threat scenarios. Similarly, the threat generator will model a target as an object consisting of several point scatterers or radiation sources. The initial simulation capability will support the development of requirements at the system level (i.e., firing rate, acquisition range, engagement ranges, laser power, etc.)

(U) The baseline version of HELWS simulation capability will be an upgrade of the initial version. It will have more detailed representations of system functions and subfunctions in the flexible simulator and the threat generator will be based on target models of greater fidelity. Battle management will handle complex threat senarios and the beam control functions will be accurately modeled for a class of laser weapons. The threat generator will contain target models that account for the special form and shape of targets. Further, the baseline version will accomodate more complicated target trajectories than the baseline version.

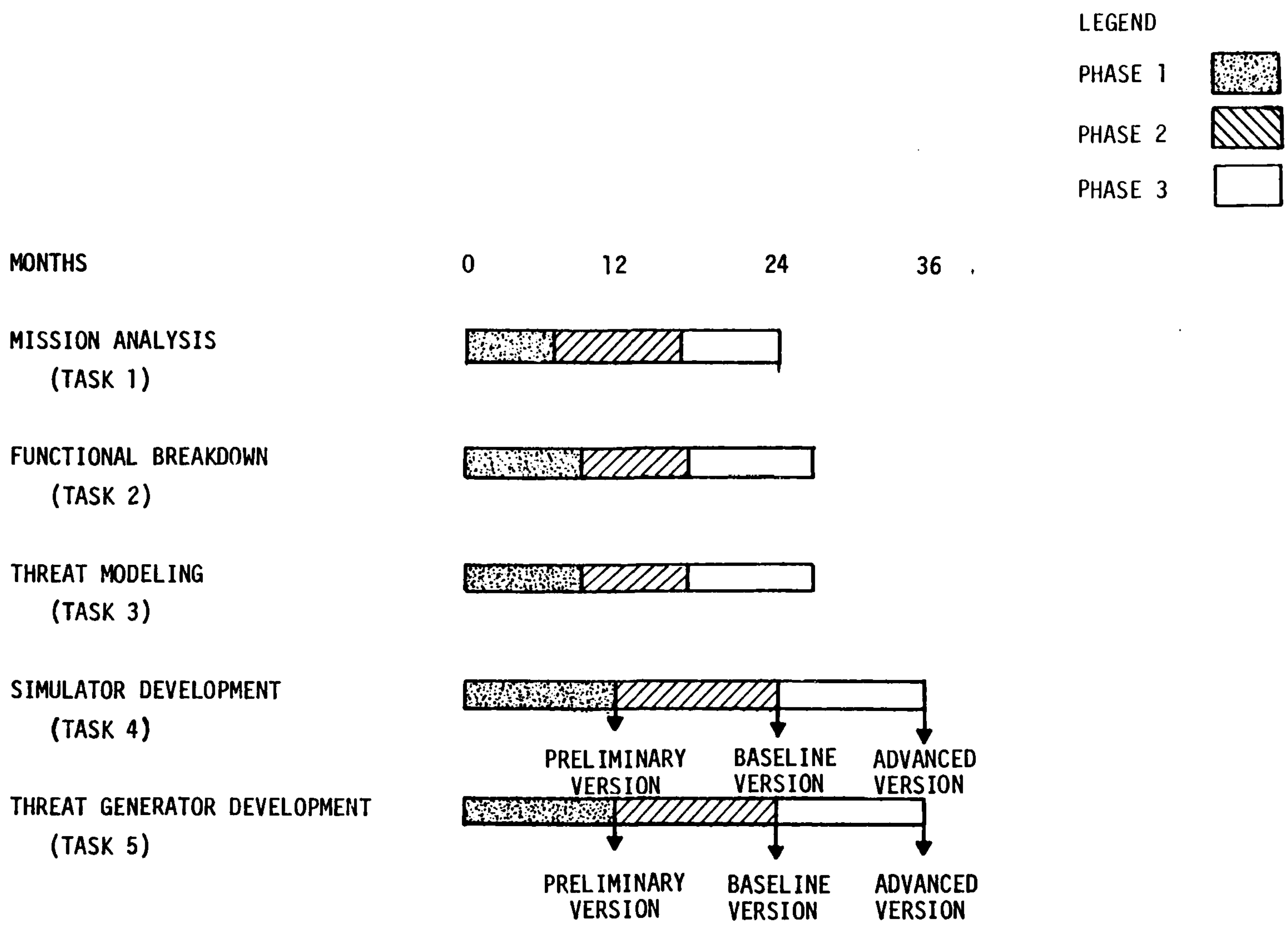


Figure 4.2 (U) Development Schedule for HELWS Simulation Capability

(U) We included a third phase in the development cycle, because laser concepts and HELWS missions will surely change as development proceeds in the earlier phases. The result of these changes will be the need for even greater flexibility than can be forseen initially. For example, some considerable mission analysis must be performed before we can fully anticipate the many alternatives for Battle Management - more analysis than is possible before committing to the design for the initial version of the simulation capability.

THIS PAGE LEFT BLANK INTENTIONALLY

5 (U) REFERENCES

1. Military Standard; Engineering Management, MIL-STD-499A (USAF), Department of Defense, 1 May 1974.

2. Hall, A. D., "Three-Dimensional Morphology of Systems Engineering," IEEE Transaction on Systems Man and Cybernetics, Vol. SMC-5, No. 2, pp 156-160, April 1969.

3. Simulation-Emulation Driver (SED) Users Guide for MMBC, C. P. Marks, General Research Corporation, June 1978.

THIS PAGE LEFT BLANK INTENTIONALLY

APPENDIX A

(U) LISTING OF SED DRIVER CONTROL LOGIC

```
SEQ NEST SOURCE                OVERLAY(SEFILE, 0, 0)

  1          OVERLAY(SEFILE, 0, 0)
  2          PROGRAM DRIVER(TAPE5, OUTPUT, TAPE6=OUTPUT, TAPE1, TAPE10, TAPE11,
            1 TAPE8, TAPE14, TAPE13, TAPE39)

  3      C
  4      C     THIS PROGRAM IS THE CONTROL FOR THE SIMULATION-EMULATION DRIVER
  5      C     (SED).
  6      C
  7      C**** CONTROL (RUN CONTROL COMMON)
  8      C

  9          COMMON/CONTROL/NCH, CONTYPE, INPUTN, INDEX, SQUEUE, FINISHQ.LASTE,
            X NCON, XCON(9), YCON(9), TITLC(2, 9), NSYS,
            1 ISRSKE(20), MODULE(20), XSYS(20), YSYS(20), TITLS(2, 20),MAIN(20)
            2,XLED(5), YLED(5), TITLL(2, 5), LCOLOR(5), NLED,
            3 ISTOP, ICTOP, ISSIDE,          COLOR(5), ISCON, ICCON, ICARDW,
            X IBAKPER, ILFTPER, IBAKSUM, ILFTSUM,
            4 IPRINT(30), CLOCKS(20), MAXLCM, NORJN, STARTT, MAXSCM, NEWSCRN,
            5 INFILE, FILE(20), IFLAGS(30)
            6, SYSTEM, ISTARTP, IDSPLA(20), MTITLE(4), NAMESN(20)
            7, OUTFILE. OUTTASK, LSTFUNC, TIMLAST, LSTFLOW, IPREAD

 10      C

 11          INTEGER COLOR, CONTYPE, SQUEUE, FINISHQ
            1, OUTFILE, OUTTASK

 12      C

 13          LOGICAL NEWSCRN

 14      C

 15          EQUIVALENCE(IFLAGS(22), IDSPQ), (IFLAGS(23), ICSPI), (IFLAGS(24),
            1 IMODE), (IFLAGS(25), I22K), (IFLAGS(26), ICHNG), (IFLAGS(27),
            2 INXTSCN), (IFLAGS(28), IPERI), (IFLAGS(29), IPERC), (IFLAGS(30),
            3 IDSP)

 16      C
 17      C**** BLANK COMMON (DSA AREA)
 18      C

 19          COMMON
            1 Q(1)

 20      C

 21          DIMENSION IQQ(1, 1), QQ(1, 1), IQ(1)

 22      C

 23          EQUIVALENCE (Q, IQ, QQ, IQQ)

 24      C
 25      C**** GENLST - GENERAL LIST EQUIVALENCE BLOCK
 26      C
```

SEQ NEST SOURCE   OVERLAY(TEST, 1, 0)

```
         DIMENSION NEWFUNC(1), NFUNC(1), TIMEFCN(1), INDIM(1), LISTIN(1),
        1 INPOUT(1), LISTOUT(1), NCOM(1)
        2, TIMESCN(1)
   C
         EQUIVALENCE (Q, NEWFUNC, NCOM), (Q(2), NFUNC), (Q(3), TIMEFCN),
        1 (Q(2), INDIN, INDOUT, LISTIN, LISTOUT)
        2, (Q(3), TIMESCN)
   C
   C
   C***** MODELNO (CURRENT MODEL NUMBER)
   C
         COMMON/MODELNO/MODEL
   C
         LOGICAL BREAKPT, BRAKNOW
   C
         DIMENSION LABEL(4)
   C
         DATA LABEL/10HPLEASE CAL, 10HL C. MARKS, 10H AT GRP #6, 10H37-7980
        1/  /
   C
   C     RESERVE THE CONSOLE AND CLEAR THE SCREEN
   C
         CALL CORES(1)
         CALL DOCOM(1)
         CALL GRCEM(1, SYSTEM, LABEL)
         CALL DOSOT(1)
         LOOP
1        .   CALL DOEKW(5.5, 0., 9HSTART JOB, 10, 1., 1., IANS)
         EXIT IF(IANS .EQ. ISTART)
         END LOOP
   C
   C     READ IN THE NOMINAL VALUES FOR THIS TEST
   C
         CALL READNOM
   C
   C     WE WILL SET UP A LOOP FOR MULTIPLE CASES
   C
         LOOP
```

```
SEQ NEST SOURCE                 OVERLAY(AFILE, 0, 0)

     1    C    *
     1    C    *   INITIALIZE FOR THIS CASE
     1    C    *
     1         .   CALL INITAL
     1    C    *
     1    C    *   LOOP ON TYPE OF COMMAND
     1    C    *
     1         .   LOOP
     2         .   .   LOOP
     3         .   .   .   CALL WATCMD(COMTYPE)
     3    C    *   *   *
     3    C    *   *   *   LEAVE THE LOOP FOR CONTINUE, REINITIALIZE OR TERMINATE
     3    C    *   *   *
     2         .   .   EXIT IF(COMTYPE .EQ. 1 .OR. COMTYPE .GE. 8)
     3    C    *   *   *
     3    C    *   *   *   IF THIS IS DISPLAY INPUT, DISPLAY OUTPUT, OR MODIFY INPUT, CALL
     3    C    *   *   *   IN THE REQUIRED OVERLAY
     3    C    *   *   *
     3         .   .   .   IF(COMTYPE .LE. 4)
     4         .   .   .   .   CALL WHICH1(MODEL)
     4         .   .   .   .   CALL SETNOW(MODEL)
     4         .   .   .   .   CALL OVERLAY(FILE(MODEL), MAIN(MODEL), 0, 6HRECALL)
     4         .   .   .   .   NEWSCRN = .TRUE.
     4         .   .   .   .   CALL SETNOW(1)
     4    C    *   *   *   *
     4    C    *   *   *   *   IS THIS SET BREAKPOINT
     4    C    *   *   *   *
     3         .   .   .   OR IF(COMTYPE .EQ. 5)
     4         .   .   .   .   CALL SETBRK
     4    C    *   *   *   *
     4    C    *   *   *   *   OR WE CHANGE MODEL
     4    C    *   *   *   *
     3         .   .   .   OR IF(COMTYPE .EQ. 6)
     4         .   .   .   .   CALL CHANGE
     4    C    *   *   *   *
     4    C    *   *   *   *   THIS MUST BE A REQUEST FOR THE NEXT INPUT DATASET
     4    C    *   *   *   *
     3         .   .   .   ELSE
     4         .   .   .   .   CALL OVERLAY(INFILE, INPUTN, 0, 6HRECALL)
     3         .   .   .   END IF
     3    C    *   *   *
```

```
SEQ NEST SOURCE                OVERLAY(AFILE, 0, 0)

 2          .  .  END LOOP

 2   C      *  *
 2   C      *  *  IF THIS IS TERMINATE OR REINITIALIZE, WE HAVE TO GET OUT OF
 2   C      *  *  ANOTHER LOOP
 2   C      *  *

 1          .  EXIT IF(COMTYPE .GE. 6)

 2   C      *  *
 2   C      *  *  WIPEOUT ANYTHING LEAVING THE SYSTEM
 2   C      *  *

 2          .  .  IF(FINISHQ .NE. 0)
 3          .  .  .   CALL BLASTIT(FINISHQ)
 3          .  .  .   NEWSCPM = .TRUE.
 2          .  .  END IF

 2   C      *  *
 2   C      *  *  LOOP FOR READING THE BUFFER FROM EMULATION AND HARDWARE MODULES,
 2   C      *  *  AND THEN PICKING UP THE FUNCTION WHICH IS TO OCCUR NEXT IN TIME
 2   C      *  *

 2          .  .  LOOP
 3          .  .  .   CALL BUFRIN(MLSEC, 1, 999KEPT)
 2          .  .  EXIT IF (BRAKEPT)
 3          .  .  .   LINK = LSTFUNC
 3          .  .  .   CALL NEXT(LINK, INDEX)
 3          .  .  .   IF(INDEX .NE. 0)
 4          .  .  .   .   CALL REMOVE(LSTFUNC, 0, INDEX)
 4          .  .  .   .   IPOINT = MFUNC(INDEX)
 4          .  .  .   .   CALL REMOVE(IQQ(IPOINT, SQUEUE), 0, INDEX)
 4          .  .  .   .   BRAKEPT = .FALSE.

 4   C      *  *  *   *
 4   C      *  *  *   *   WE HAVE ONE. CALL IN THE OVERLAY TO PROCESS IT
 4   C      *  *  *   *

 4          .  .  .   .   CALL SETNOW(IPOINT)
 4          .  .  .   .   COMTYPE = 1
 4          .  .  .   .   CALL OVERLAY(FILE(IPOINT), MAIN(IPOINT),0, 6HEFCALL)

 4   C      *  *  *   *
 4   C      *  *  *   *   DO WE WRITE OUT THIS TASKS FILE
 4   C      *  *  *   *

 4          .  .  .   .   IF(IPOINT .EQ. OUTTASK .AND. INDEX .NE. 0)
 5          .  .  .   .   .   COMTYPE = 8
 5          .  .  .   .   .   IPP = MFUNC(INDEX)
 5          .  .  .   .   .   CALL OVERLAY(FILE(IPP), MAIN(IPP), 0, 0)
 4          .  .  .   .   END IF
 4          .  .  .   .   IF(BRAKNOW(IPOINT))
 5          .  .  .   .   .   BRAKEPT = .TRUE.
 4          .  .  .   .   END IF
 3          .  .  .   END IF
 2          .  .  EXIT IF(BRAKEPT)
```

```
SEQ NEST SOURCE                    OVERLAY(FETLE, 0, 0)

     2          .    .    EXIT IF(LSTFUNC .EQ. 0)
     2          .    .    END LOOP
     1          .    END LOOP

     1    C     *
     1    C     *    IF THIS IS TERMINATE, WE MUST GET OUT OF ONE MORE LOOP
     1    C     *

                EXIT IF(COMTYPE .EQ. 9)
                END LOOP

          C
          C     SIGN OFF
          C

                CALL DDSDI(0)
                CALL GPSF1(2; SYSTEM)
                CALL DDREL
                WRITE(6, 10)
             10 FORMAT(1H1,10X,*NORMAL TERMINATION FOR THIS RUN THROUGH THE SEQ PR
               1OGRAM.*//)
                IF(OUTTASK .GT. 0)
     1          .    WRITE(OUTFILE)-99; 0; 0; 0
                END IF
                STOP 1234

          C

                END
```

THIS PAGE LEFT BLANK INTENTIONALLY

APPENDIX B
(U) LISTING OF FUNCTION MODEL PRIORITIZATION

```
SEQ NEST SOURCE                 OVERLAY(FILE6, 6, 0, C37777)

           OVERLAY(FILE6, 6, 0, C37777)
           PROGRAM PRIORTY
      C
      C    ADAPT THIS OVERLAY TO PRIORITIZATION
      C
      C**** BLANK COMMON (OSA AREA)
      C
           COMMON
          1 Q(1)
      C
           DIMENSION IQQ(1, 1), QQ(1, 1), IQ(1)
      C
           EQUIVALENCE (Q, IQ, QQ, IQQ)
      C
      C**** CONTROL (RUN CONTROL COMMON)
      C
           COMMON/CONTROL/NCM, CONTYPE, INPUTN, INDEX, SQUEUE, FINISHQ,LASTF,
          X NCON, XCOM(9), YCOM(9), TITLC(2, 9), NSYS,
          1 IBRIKE(20), MODULE(20), XSYS(20), YSYS(20), TITLS(2, 20),MAIN(20)
          2,XLED(5), YLED(5), TITLL(2, 5), LCOLOR(5), NLED,
          3 ISTOP, ICTOP, ISSTOP,           COLOR(5), ISCON, ICCON, ICARCH,
          X IBAKPER, ILFTPER, IBAKSUM, ILFTSUM,
          4 IPRINT(30), CLOCKS(20), MAXLCM, NORUN, STARTT, MAXSCM, NEWCORN,
          5 INFILE, FILE(20), IFLAGS(30)
          6, SYSTEM, ISTARTF, IDSFLA(20), MTITLE(4), NAMESN(20)
          7, OUTFILE, OUTTASK, LSTFUNC, TIMLAST, LSTFLOW, INREAD
      C
           INTEGER COLOR, CONTYPE, SQUEUE, FINISHQ
          1, OUTFILE, OUTTASK
      C
           LOGICAL NEWCORN
      C
           EQUIVALENCE (IFLAGS(22), IDEPO), (IFLAGS(23), IDSPI), (IFLAGS(24),
          1 IMODE), (IFLAGS(25), IBRK), (IFLAGS(26), ICHKS), (IFLAGS(27),
          2 INXTSCN), (IFLAGS(28), IPERI), (IFLAGS(29), IPREC), (IFLAGS(30),
          3 IOSP)
      C
      C**** GENLST - GENERAL LIST EQUIVALENCE BLOCK
      C
           DIMENSION NPRFUNC(1), NFUNC(1), TIMEFCN(1), INPIN(1), LISTIN(1),
```

```
SEQ NEST SOURCE                     OVERLAY(FILE6, 6, 0, 037777)
              1 INDOUT(1), LISTOUT(1), NSCAN(1)
              2, TIMESCN(1)

           C

                EQUIVALENCE (Q, NERFUNC, NSCAN), (Q(2), NFUNC), (Q(3), TIMESCN),
              1 (Q(4), INDTN, INDOUT, LISTIN, LISTOUT)
              2, (Q(3), TIMESCN)

           C
           C

                DIMENSION NAME(3), TYPE(3), IND1(3), IND2(3), IND3(3), IPOINT(3),
              1 LTYPE(3), IBOTH(2)

           C

                INTEGER TYPE

           C

                EQUIVALENCE (IND2, IND3)

           C

                DATA NAME/9HFROM FUNC, 7HTO FUNC, 9HTIME FUNC/, TYPE/2*1, 2/,
              1 IND1/3*1/, IND2/3*0/, IPOINT/1, 2, 3/, LENGTH/3/, LSTART/1/

           C

                DATA IBOTH/7, 8/

           C

                CALL COMOUT(TEL, TST, 5)

           C
           C    SEE WHICH ENTRY WE HAVE
           C

                IF(COMTYPE .EQ. 1)

       1   C    *
       1   C    *   READ IN THE NOMINAL DATA
       1   C    *
       1   C    *
       1   C    *   IS THIS THE INITIALIZATION LOOP
       1   C    *

                OR IF(COMTYPE .EQ. 3)

       1   C    *

       1        .   CALL DEPATH(6, 0, 0, 0, 0, 2, IBOTH)

       1   C    *
```

```
SEQ NEST SOURCE                 OVERLAY(FTLF6, 6, 0, C37777)

     1   C       *   DO WE EXECUTE THE MODULE
     1   C       *------------------------------------------------------------

                 OR IF(COMTYPE .EQ. 1)
     1           .   CALL COUNT(INDEX, LENGTH, LSTART, LTYPE, IPOINT, NWORDS)
     1           .   CALL PRIOR  (IPOINT(IFL), TST, NWORDS)

     1   C       *------------------------------------------------------------
     1   C       *   DO WE DISPLAY THE OUTPUT
     1   C       *

                 OR IF(COMTYPE .EQ. 2)

     1   C       *------------------------------------------------------------
     1   C       *   THE PERFORMANCE MONITOR
     1   C       *
     1   C       *------------------------------------------------------------
     1   C       *   DO WE RELEASE THE SPACE
     1   C       *

                 OR IF(COMTYPE .EQ. 5)
     1           .   CALL RELEASE(INDEX, LENGTH, LSTART, LTYPE, IPOINT)

     1   C       *
     1   C       *   DISPLAY THE INPUT OR MODIFY IT
     1   C       *------------------------------------------------------------

                 ELSE
     1           .   IF(COMTYPE .EQ. 3)
     2           .   .   IPATH = 0
     1           .   ELSE
     2           .   .   IPATH = 1
     1           .   END IF

     1   C       *
     1   C       *   SEE WHICH ONE
     1   C       *

     1           .   LOOP
     2           .   .   CALL DISOUT(NUMB)
     1           .   EXIT IF (NUMB .EQ. 0)
     2           .   .   IF(NUMB .EQ. 1)

     3   C       *---*---*
     3   C       *   *   *   THE MODULE INPUT
     3   C       *   *   *

     3           .   .   .   LINK = IOO(MON, COQUEUE)
     3           .   .   .   CALL NEXT(LINK, INDEX)
     3           .   .   .   IF(INDEX .NE. 0)
     4           .   .   .   .   CALL DSPLAYIN(NAME, TYPE, IND1, IND2, IND3, IPOINT, LTYPE, LENGTH,
               1.    .   .   .       LSTART, INDEX, IPATH)
     3           .   .   .   END IF

     3   C       *   *   *
     3   C       *---*---*   OR THE INITIALIZATION VARIABLES
```

```
SEQ NEST SOURCE                    OVERLAY(FILE6, 6, 0, C77777)

 3    C     *   *   *

 2          .   .   OR IF(NUMB .EQ. 2)

 3    C     *   *   *

 2          .   .   END IF
 1          .   END LOOP
            END IF

      C
      C     FINISH BOOKKEEPING
      C

            IF(COMTYPE .NE. 1)
 1          .   CALL BKEEP(0, TST)
            END IF

      C

            RETURN
            END
```

SEQ NEST SOURCE SUBROUTINE PRIOR(IPRNT, TST, NWORDS)

```
      SUBROUTINE PRIOR(IPRNT, TST, NWORDS)
C
C     THIS ROUTINE PERFORMS THE PRIORITIZATION OF THE TARGETS
C
C**** BLANK COMMON (DSA AREA)
C
      COMMON
     1 Q(1)
C
      DIMENSION IQQ(1, 1), QQ(1, 1), IQ(1)
C
      EQUIVALENCE (Q, IQ, QQ, IQQ)
C
C**** GENLST - GENERAL LIST EQUIVALENCE BLOCK
C
      DIMENSION NEFFUNC(1), NFUNC(1), TIMEFCN(1), INDIN(1), LISTIN(1),
     1 INDOUT(1), LISTOUT(1), NSCAN(1)
     2, TIMESCN(1)
C
      EQUIVALENCE (Q, NEFFUNC, NSCAN), (Q(2), NFUNC), (Q(3), TIMEFCN),
     1 (Q(4), INDIN, INDOUT, LISTIN, LISTOUT)
     2, (Q(3), TIMESCN)
C
C
C**** CONTROL (RUN CONTROL COMMON)
C
      COMMON/CONTROL/NOW, CONTYPE, INPUTN, INDEX, SQUEUE, FINISHQ,LASTF,
     X NCON, XCON(9), YCON(9), TITLC(2, 9), NSYS,
     1 IBRAKE(20), MODULE(20), XSYS(20), YSYS(20), TITLS(2, 20),MAIN(20)
     2,XLED(5), YLED(5), TITLL(2, 5), LCOLOR(5), NLED,
     3 ISTOP, ICTOF, ISSIDE,         COLOR(5), ISCON, ICCON, ICSHOW,
     X IBRKPER, ILETPER, IBRKSUM, ILETSUM,
     4 IPRINT(30), CLOCKS(20), MAXLCM, NORUN, STARTT, MAXSCM, NEWCORN,
     5 INFILE, FILE(20), IFLAGS(30)
     6, SYSTEM, ISTARTP, IRSELA(20), NTITLE(4), NAMSEN(20)
     7, OUTFILE, OUTTASK, LSTFUNC, TIMLAST, LSTFLOW, INFEAR
C
      INTEGER COLOR, CONTYPE, SQUEUE, FINISHQ
     1, OUTFILE, OUTTASK
C
```

SEQ NEST SOURCE   SUBROUTINE PRIOR(IFRNT, IST, NWORDS)

```
      LOGICAL NEWSCON
C
      EQUIVALENCE (IFLAGS(22), INCPO), (IFLAGS(23), ICSPT), (IFLAGS(24),
     1 INOTF), (IFLAGS(25), INRK), (IFLAGS(26), ICHNG), (IFLAGS(27),
     2 INXTSON), (IFLAGS(29), TPRPI), (IFLAGS(29), IEFKC), (IFLAGS(30),
     3 IDS2)
C
C***** INITSIM - INITIALIZATION FOR SIMULATION
C
      COMMON/INITSIM/DT1, DT2, DT3, DT4, DT5, DT6, DT7, DT8, DT9, DT10,
     1 DT11, DT12, DT26, XRADAR, YRADAR, ZRADAR, RADRAC, NLASER, RADLAS,
     2 XLASER(10), YLASER(10), ZLASER(10), NCTARS, XPOSI(40), YPOSI(40),
     3 ZPOSI(40), XVELI(40), YVELI(40), ZVELI(40), NCMANS, TMAN(100),
     4 NTMAN(100), XVMAN(100), YVMAN(100), ZVMAN(100)
C
      DIMENSION STAT(40, 6)
C
      EQUIVALENCE (STAT, XPOSI)
C
C***** SIMINTR - SIMULATION INTERFACE COMMON BLOCK
C
      COMMON/SIMINTR/INDTAK, TIMNOW, LSTDET, LSTUDET, INDFIRE, INPHDR(3)
     1, SLEWTIM, AZNOW
C
      DIMENSION XPOS(1), YPOS(1), ZPOS(1), XVEL(1), YVEL(1), ZVEL(1),
     1 STATE(1, 6), NUMTAR(1), TIME(1), TIMKTP(1), TLSTKTP(1), IMPACT(1)
     2, NTRKP(1), TYPTRAK(1), RNGLASR(1), TIMOFIR(1), TIMOKIL(1),
     3 PREDPOS(3), ALIVE(1), NHSTP(1), ENERGY(1)
C
      LOGICAL ALIVE
C
      INTEGER TYPTRAK
C
      EQUIVALENCE (Q(1), NUMTAR), (Q(2), TIME), (Q(3), XPOS, STATE),
     1 (Q(4), YPOS), (Q(5), ZPOS), (Q(6), XVEL), (Q(7), YVEL), (Q(8),
     2 ZVEL), (Q(9), NTRKP), (Q(10), TYPTRAK), (Q(11), IMPACT), (Q(12),
     3 TLSTKTP), (Q(13), TIMKTP), (Q(14), RNGLASR), (Q(15), TIMOFIR),
     4 (Q(16), TIMOKIL), (Q(17), PREDPOS), (Q(20), ALIVE), (Q(21), NHSTP
```

SEQ NEST SOURCE   SUBROUTINE PRIOR(IPRNT, IST, NWORDS)

E), (Q(22), ENERGY)

```
      C
            DIMENSION ISAV(3), XYZ(3)
      C
            DO(I = 1, 3)
   1        .  ISAV(I) = INDHOR(I)
   1        .  INDHOR(I) = 0
            END DO
            ISAVE = INDFIRE
      C
      C     MAKE UP A LIST OF ACCEPTABLE TARGETS
      C
            LIST = 0
            LINK = LSTOPT
            LOOP
   1        .  CALL NEXT(LINK, IND)
            EXIT IF(IND .EQ. 0)
   1        .  IF(ALIVE(IND) .AND.IMPACT(IND) .GT. 0 .AND. TIMKTP(IND) .GT. 0.)
   2        .  .  IF(TIMKTP(IND) .LE. 5. .AND. IND .NE. INDFIRE)
   3        .  .  .  CALL PUTTOP(LIST, IND)
   2        .  .  END IF
   1        .  END IF
            END LOOP
      C
      C     SEE IF WE HAVE ANYTHING
      C
      C
      C     SEE IF WE HAVE TO PUT A FORMER NO. 1 BACK ON THE LIST
      C
            IF(INDFIRE .NE. 0)
   1        .  IF(TYPTRAK(INDFIRE) .LE. 2)
   2        .  .  CALL PUTORA(LIST, INDFIRE, 14)
   2        .  .  INDFIRE = 0
   1        .  END IF
            END IF
            IF(LIST .NE. 0)
   1        .  CALL LSTSRT(LIST, 0, 14)
   1  C     *
   1  C     *  PUT THE TOP 3 INTO HIGH DATA RATE TRACK
   1  C     *
   1        .  N = 0
   1        .  LCUT = 0
   1        .  LINK = LIST
   1        .  LOOP
   2        .  .  N = N + 1
   1        .  EXIT IF(N .GT. 3)
```

```
SEQ NEST SOURCE                SUBROUTINE PRIOR(IPRNT, IST, NWORDS)

     2            .   .   CALL NEXT(LINK, IND)
     1            .   EXIT IF(IND .EQ. 0)

     2    C       *   *   COMPUTE TIME TO OPEN FIRE
     2    C       *   *

     2            .   .   CALL ORJALT(NUMTAR(IND), 0., XYZ)
     2            .   .   TIMOFIR(IND) = XYZ(1) - 2. - TIMKIP(IND)
     2            .   .   IF(TIMOFIR(IND) .LE. TIMEFCN(INDEX) + 1.)
     3            .   .   .   N = N - 1
     2            .   .   ELSE
     3            .   .   .   INDHOP(N) = IND
     3            .   .   .   CALL FUTPOT(LOUT, IND)

     3    C       *   *   *

     3            .   .   .   IF(INDFIRE .EQ. 0 .AND. N .EQ. 1)
     4            .   .   .   .   TIMOKIL(IND) = TIMOFIR(IND) + TIMKIP(IND)
     4            .   .   .   .   INDFIRE = IND
     4            .   .   .   .   IF(IND .NE. ISAVE)
     5            .   .   .   .   .   SLEWTIM = 1.E20
     5            .   .   .   .   .   CALL SCHEDUL(6, 8, TIMEFCN(INDEX), IND)
     4            .   .   .   .   END IF
     3            .   .   .   END IF
     3            .   .   .   IF(IPRNT .GE. 2)
     4            .   .   .   .   WRITE(6, 10)NUMTAR(IND), N
     4    10      .   .   .   .   FORMAT(/5X,'TARGET NO.',I3,' HAS BEEN CHOSEN AS PRIORITY NO.',I2)
     3            .   .   .   END IF
     2            .   .   END IF
     1            .   END LOOP

     1    C       *
     1    C       *   SEE IF A HIGH DATA RATE PULSE PREVIOUSLY WAS SCHEDULED
     1    C       *

     1            .   LNK = IOD(7, SQUEUE)
     1            .   CALL NEXT(LNK, ID)
     1            .   IF(ID .NE. 0)
     2            .   .   L = LISTOUT(ID)
     2            .   .   TSKED = TIMEFCN(ID)
     2            .   .   CALL CANCEL(7, L)
     1            .   ELSE
     2            .   .   TSKED = TIMEFCN(INDEX)
     1            .   END IF

     1    C       *

     1            .   IF(LOUT .NE. 0)
     2            .   .   CALL SCHEDUL(6, 7, TSKED, LOUT)
     1            .   END IF
                  END IF

          C
          C       SEE IF WE HAVE TO RE-SET TRACK-WHILE-SCAN ON ANY OBJECTS
          C
```

```
SEQ NEST SOURCE                 SUBROUTINE PRIOR(IPRNT, IST, NWORDS)

             DO (I = 1, 3)
       1     .    IF(ISAV(I) .NE. 0)
       2     .    .    FOR (J = 1 TO 3 UNTIL ISAV(I) .EQ. INDHDR(J))
       2     .    .    END FOR
       2     .    .    IF(J .GT. 3)
       3     .    .    .    IND = ISAV(I)
       3     .    .    .    TYPTRAX(IND) = 1
       3     .    .    .    NTRKE(IND) = 1
       3     .    .    .    CALL SCHEDUL(6, 4, TIMEFCN(INDEX) + DT1, IND)
       2     .    .    END IF
       1     .    END IF
             END DO
        C
             CALL DSTROY(INDEX)
             CALL BKEEP(NWORDS, IST)
        C
             RETURN
             END
```

THIS PAGE LEFT BLANK INTENTIONALLY

APPENDIX C

(U) SELECTED PRINTOUTS FROM HELWS TEST CASE

```
SEARCH CONTROL      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     2.00000000

                CURRENT CP TIME IS     14.576 WHICH IS      .065 SECONDS INTO THE RUN
                    OVERLAY  CALLED WITH CONTYPE =    1

FUNCTIONAL MODULE  SEARCH CONTROL       ENDED AT CP TIME     14.575
     EXECUTION TOOK        .001 CP SECONDS AND     1 WORDS WERE INPUT TO IT.

*********************************************************************************************

SEARCH CONTROL      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     5.00000000

                CURRENT CP TIME IS     14.577 WHICH IS      .067 SECONDS INTO THE RUN
                    OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 AT AN ALTITUDE OF      4700. AND RANGE FROM RADAR OF      9889. HAS BEEN DETECTED.

     STATE =      7520.00     4703.00     4700.00   -160.0000    -99.0000   -100.0000

TARGET NO.  3 AT AN ALTITUDE OF      4697. AND RANGE FROM RADAR OF      9899. HAS BEEN DETECTED.

     STATE =      7526.00    -4716.00     4697.00   -158.0000     94.0000   -101.0000

     FUNCTIONAL MODULE  SEARCH CONTROL       ENDED AT CP TIME     14.579
          EXECUTION TOOK        .002 CP SECONDS AND     1 WORDS WERE INPUT TO IT.

*********************************************************************************************

TRACK INITIATE      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     3.00000000

                CURRENT CP TIME IS     14.581 WHICH IS      .072 SECONDS INTO THE RUN
                    OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 NOW BEING PROCESSED.

TARGET NO.  3 NOW BEING PROCESSED.

     FUNCTIONAL MODULE  TRACK INITIATE       ENDED AT CP TIME     14.582
          EXECUTION TOOK        .001 CP SECONDS AND    48 WORDS WERE INPUT TO IT.

*********************************************************************************************

SEARCH CONTROL      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00000000

                CURRENT CP TIME IS     14.585 WHICH IS      .075 SECONDS INTO THE RUN
```

```
                    OVERLAY CALLED WITH CONTYPE =    1

TARGET NO.  4 AT AN ALTITUDE OF      4634. AND RANGE FROM RADAR OF      9949. HAS BEEN DETECTED.

     STATE =      -7340.00    -4576.00     4634.00    165.0000    106.0000    -99.0000

     FUNCTIONAL MODULE  SEARCH CONTROL       ENDED AT CP TIME      14.586
          EXECUTION TOOK        .001 CP SECONDS AND      1 WORDS WERE INPUT TO IT.

**************************************************************************************

       TRACK WHILE SCAN    SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00000000

                    CURRENT CP TIME IS      14.589 WHICH IS      .079 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 IS AT AN ALTITUDE OF       4630. AND RANGE FROM RADAR OF      9676.
    NO. OF T-W-S PULSES =    1

     FUNCTIONAL MODULE  TRACK WHILE SCAN     ENDED AT CP TIME      14.589
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

**************************************************************************************

       PRED IMPACT POINT   SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00000000

                    CURRENT CP TIME IS      14.592 WHICH IS      .082 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 IS ATTACKING THIS LASER.  RANGE =       50.

     FUNCTIONAL MODULE  PRED IMPACT POINT   ENDED AT CP TIME      14.592
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

**************************************************************************************

       KILL TIME PREDICT   SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00000000

                    CURRENT CP TIME IS      14.595 WHICH IS      .085 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 AT A RANGE TO THE ASSET OF        50. HAS A SHOT TIME OF    1.850

     FUNCTIONAL MODULE  KILL TIME PREDICT   ENDED AT CP TIME      14.595
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.
```

```
**********************************************************************************

     TRACK WHILE SCAN    SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00000000

                      CURRENT CP TIME IS      14.598 WHICH IS      .088 SECONDS INTO THE RUN
                              OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  3 IS AT AN ALTITUDE OF       4566. AND RANGE FROM RADAR OF       3689.
        NO. OF T-W-S PULSES =    1

     FUNCTIONAL MODULE  TRACK WHILE SCAN    ENDED AT CP TIME      14.599
             EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**********************************************************************************

     PRED IMPACT POINT   SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00000000

                      CURRENT CP TIME IS      14.601 WHICH IS      .092 SECONDS INTO THE RUN
                              OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  3 IS A TARGET OF OPPORTUNITY.

     FUNCTIONAL MODULE  PRED IMPACT POINT   ENDED AT CP TIME      14.602
             EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**********************************************************************************

     KILL TIME PREDICT   SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00000000

                      CURRENT CP TIME IS      14.604 WHICH IS      .095 SECONDS INTO THE RUN
                              OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  3 AT A RANGE TO THE ASSET OF        330. HAS A SHOT TIME OF      .931

     FUNCTIONAL MODULE  KILL TIME PREDICT  ENDED AT CP TIME      14.605
             EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**********************************************************************************

     TRACK INITIATE      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00000000

                      CURRENT CP TIME IS      14.609 WHICH IS      .09? SECONDS INTO THE RUN
                              OVERLAY  CALLED WITH CONTYPE =     1
```

```
TARGET NO.  4 NOW BEING PROCESSED.

     FUNCTIONAL MODULE  TRACK INITIATE        ENDED AT CP TIME      14.609
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

*****************************************************************************************

       PRIORITIZATION      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00001000

                     CURRENT CP TIME IS      14.611 WHICH IS       .101 SECONDS INTO THE RUN
                          OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  1 HAS BEEN CHOSEN AS PRIORITY NO. 1

TARGET NO.  3 HAS BEEN CHOSEN AS PRIORITY NO. 2

     FUNCTIONAL MODULE  PRIORITIZATION        ENDED AT CP TIME      14.612
          EXECUTION TOOK        .001 CP SECONDS AND      4 WORDS WERE INPUT TO IT.

*****************************************************************************************

       SLEW CONTROL        SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00001000

                     CURRENT CP TIME IS      14.616 WHICH IS       .105 SECONDS INTO THE RUN
                          OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  1 HAS A SLEW TIME OF      2.6418

     FUNCTIONAL MODULE  SLEW CONTROL          ENDED AT CP TIME      14.615
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

*****************************************************************************************

       HIGH DATA RATE      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     4.00001000

                     CURRENT CP TIME IS      14.618 WHICH IS       .108 SECONDS INTO THE RUN
                          OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  1 IS AT AN ALTITUDE OF       4600. AND A RANGE OF       9575.
               NO. OF HIGH DATA RATE PULSES =     1
TARGET NO.  3 IS AT AN ALTITUDE OF       4596. AND A RANGE OF       9693.
               NO. OF HIGH DATA RATE PULSES =     1

     FUNCTIONAL MODULE  HIGH DATA RATE        ENDED AT CP TIME      14.619
          EXECUTION TOOK        .001 CP SECONDS AND     48 WORDS WERE INPUT TO IT.
```

```
        FUNCTIONAL MODULE  SEARCH CONTROL        ENDED AT CP TIME      19.179
            EXECUTION TOOK        .000 CP SECONDS AND      1 WORDS WERE INPUT TO IT.

*****************************************************************************

        HIGH DATA RATE      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    46.00001000

                    CURRENT CP TIME IS      19.179 WHICH IS     4.668 SECONDS INTO THE RUN
                        OVERLAY  CALLED WITH CONTYPE =     1

    TARGET NO.  2 IS AT AN ALTITUDE OF        492. AND A RANGE OF       1066.
                NO. OF HIGH DATA RATE PULSES =  800
    TARGET NO.  3 IS AT AN ALTITUDE OF        354. AND A RANGE OF        330.
                NO. OF HIGH DATA RATE PULSES =  861
    TARGET NO.  4 IS AT AN ALTITUDE OF        446. AND A RANGE OF        765.
                NO. OF HIGH DATA RATE PULSES =  747

        FUNCTIONAL MODULE  HIGH DATA RATE        ENDED AT CP TIME      19.179
            EXECUTION TOOK        .001 CP SECONDS AND     70 WORDS WERE INPUT TO IT.

*****************************************************************************

        HIGH DATA RATE      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    46.05001000

                    CURRENT CP TIME IS      19.181 WHICH IS     4.672 SECONDS INTO THE RUN
                        OVERLAY  CALLED WITH CONTYPE =     1

    TARGET NO.  2 IS AT AN ALTITUDE OF        487. AND A RANGE OF       1053.
                NO. OF HIGH DATA RATE PULSES =  801
    TARGET NO.  3 IS AT AN ALTITUDE OF        349. AND A RANGE OF        320.
                NO. OF HIGH DATA RATE PULSES =  862
    TARGET NO.  4 IS AT AN ALTITUDE OF        441. AND A RANGE OF        755.
                NO. OF HIGH DATA RATE PULSES =  748

        FUNCTIONAL MODULE  HIGH DATA RATE        ENDED AT CP TIME      19.183
            EXECUTION TOOK        .001 CP SECONDS AND     70 WORDS WERE INPUT TO IT.

*****************************************************************************

        CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    46.05001000

                    CURRENT CP TIME IS      19.185 WHICH IS     4.676 SECONDS INTO THE RUN
                        OVERLAY  CALLED WITH CONTYPE =     1

    TARGET NO.  1 HAS HAD 365 HITS.  IT IS AT A RANGE OF   733. AND AZIMUTH OF 34.91

        FUNCTIONAL MODULE  CORRELATION TRACK  ENDED AT CP TIME      19.186
            EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.
```

```
*******************************************************************************************

   HIGH DATA RATE      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    46.10000000

                  CURRENT CP TIME IS      19.189 WHICH IS      4.679 SECONDS INTO THE RUN
                        OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  2 IS AT AN ALTITUDE OF       482. AND A RANGE OF       1043.
        NO. OF HIGH DATA RATE PULSES =  802
TARGET NO.  3 IS AT AN ALTITUDE OF       344. AND A RANGE OF        911.
        NO. OF HIGH DATA RATE PULSES =  843
TARGET NO.  4 IS AT AN ALTITUDE OF       436. AND A RANGE OF        745.
        NO. OF HIGH DATA RATE PULSES =  749

      FUNCTIONAL MODULE  HIGH DATA RATE      ENDED AT CP TIME     19.190
            EXECUTION TOOK        .001 CP SECONDS AND    70 WORDS WERE INPUT TO IT.


*******************************************************************************************

   OPEN FIRE           SIMULATION MODULE TO BEGIN EXECUTION AT TIME    46.15000000

                  CURRENT CP TIME IS      19.193 WHICH IS      4.683 SECONDS INTO THE RUN
                        OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  1 IS AT A RANGE OF    845. FROM THE LASER.
      ESTIMATED TIME-TO-KILL IS    1.950000 SECONDS

      FUNCTIONAL MODULE  OPEN FIRE           ENDED AT CP TIME     19.193
            EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.


*******************************************************************************************

   HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    46.15000000

                  CURRENT CP TIME IS      19.195 WHICH IS      4.686 SECONDS INTO THE RUN
                        OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  1 HAS HAD   1 HITS.  IT IS AT A RANGE OF   712. AND AZIMUTH OF  34.99

      FUNCTIONAL MODULE  HOT SPOT TRACK      ENDED AT CP TIME     19.197
            EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.


*******************************************************************************************

   HIGH DATA RATE      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    46.15000000
```

```
HIGH DATA RATE       SIMULATION MODULE TO BEGIN EXECUTION AT TIME    47.95001003

                    CURRENT CP TIME IS      19.461 WHICH IS      4.951 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  2 IS AT AN ALTITUDE OF        301. AND A RANGE OF        561.
              NO. OF HIGH DATA RATE PULSES =  613
TARGET NO.  3 IS AT AN ALTITUDE OF        157. AND A RANGE OF        564.
              NO. OF HIGH DATA RATE PULSES =  630
TARGET NO.  4 IS AT AN ALTITUDE OF        253. AND A RANGE OF        592.
              NO. OF HIGH DATA RATE PULSES =  716

     FUNCTIONAL MODULE  HIGH DATA RATE        ENDED AT CP TIME     19.462
          EXECUTION TOOK        .001 CP SECONDS AND    70 WORDS WERE INPUT TO IT.

*******************************************************************************

CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    47.95001003

                    CURRENT CP TIME IS      19.465 WHICH IS      4.955 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD 384 HITS.  IT IS AT A RANGE OF   350. AND AZIMUTH OF 37.64

     FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME     19.466
          EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

*******************************************************************************

CEASE FIRE          SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.00000000

                    CURRENT CP TIME IS      19.468 WHICH IS      4.959 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

     FUNCTIONAL MODULE  CEASE FIRE            ENDED AT CP TIME     19.469
          EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

*******************************************************************************

DAMAGE ASSESSMENT   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.00000000

                    CURRENT CP TIME IS      19.471 WHICH IS      4.962 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS BEEN DECLARED KILLED.
```

```
FUNCTIONAL MODULE  DAMAGE ASSESSMENT   ENDED AT CP TIME     19.472
     EXECUTION TOOK       .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.


**************************************************************************************


  PRIORITIZATION      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     48.00000000

                CURRENT CP TIME IS      19.474 WHICH IS      4.965 SECONDS INTO THE RUN
                     OVERLAY  CALLED WITH CONTYPE =     1

FUNCTIONAL MODULE  PRIORITIZATION       ENDED AT CP TIME     19.475
     EXECUTION TOOK       .001 CP SECONDS AND      4 WORDS WERE INPUT TO IT.


**************************************************************************************


  SEARCH CONTROL      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     48.00000000

                CURRENT CP TIME IS      19.473 WHICH IS      4.968 SECONDS INTO THE RUN
                     OVERLAY  CALLED WITH CONTYPE =     1

FUNCTIONAL MODULE  SEARCH CONTROL       ENDED AT CP TIME     19.478
     EXECUTION TOOK       .001 CP SECONDS AND      1 WORDS WERE INPUT TO IT.


**************************************************************************************


  HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     48.05000000

                CURRENT CP TIME IS      19.481 WHICH IS      4.971 SECONDS INTO THE RUN
                     OVERLAY  CALLED WITH CONTYPE =     1

 TARGET NO.  1 HAS HAD  20 HITS.  IT IS AT A RANGE OF    331. AND AZIMUTH OF  37.92

FUNCTIONAL MODULE  HOT SPOT TRACK       ENDED AT CP TIME     19.481
     EXECUTION TOOK       .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.


**************************************************************************************


  CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME     48.05001000

                CURRENT CP TIME IS      19.484 WHICH IS      4.974 SECONDS INTO THE RUN
                     OVERLAY  CALLED WITH CONTYPE =     1
```

```
TARGET NO.  1 HAS HAD 385 HITS.  IT IS AT A RANGE OF    331. AND AZIMUTH OF 37.92

     FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME      19.495
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

*******************************************************************************************

       HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     48.15000000

                    CURRENT CP TIME IS      19.487 WHICH IS      4.978 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH COMTYPE =     1

TARGET NO.  1 HAS HAD  21 HITS.  IT IS AT A RANGE OF    313. AND AZIMUTH OF  38.23

     FUNCTIONAL MODULE  HOT SPOT TRACK      ENDED AT CP TIME      19.488
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

*******************************************************************************************

       CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME     48.15001000

                    CURRENT CP TIME IS      19.491 WHICH IS      4.981 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH COMTYPE =     1

TARGET NO.  1 HAS HAD 386 HITS.  IT IS AT A RANGE OF    313. AND AZIMUTH OF 38.23

     FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME      19.491
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

*******************************************************************************************

       HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     48.25000000

                    CURRENT CP TIME IS      19.494 WHICH IS      4.984 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH COMTYPE =     1

TARGET NO.  1 HAS HAD  22 HITS.  IT IS AT A RANGE OF    295. AND AZIMUTH OF  38.57

     FUNCTIONAL MODULE  HOT SPOT TRACK      ENDED AT CP TIME      19.494
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

*******************************************************************************************
```

```
CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.25001000
              CURRENT CP TIME IS     19.497 WHICH IS      4.987 SECONDS INTO THE RUN
                  OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD 387 HITS.  IT IS AT A RANGE OF   295. AND AZIMUTH OF 38.57

    FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME      19.498
        EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**************************************************************************************

HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.35000000
              CURRENT CP TIME IS     19.500 WHICH IS      4.991 SECONDS INTO THE RUN
                  OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD  23 HITS.  IT IS AT A RANGE OF   277. AND AZIMUTH OF  38.94

    FUNCTIONAL MODULE  HOT SPOT TRACK      ENDED AT CP TIME      19.501
        EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**************************************************************************************

CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.35001000
              CURRENT CP TIME IS     19.504 WHICH IS      4.994 SECONDS INTO THE RUN
                  OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD 388 HITS.  IT IS AT A RANGE OF   277. AND AZIMUTH OF 38.94

    FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME      19.504
        EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**************************************************************************************

HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.45000000
              CURRENT CP TIME IS     19.507 WHICH IS      4.997 SECONDS INTO THE RUN
                  OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD  24 HITS.  IT IS AT A RANGE OF   260. AND AZIMUTH OF  39.36

    FUNCTIONAL MODULE  HOT SPOT TRACK      ENDED AT CP TIME      19.507
        EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.
```

```
*****************************************************************************************

        CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.45001000

                         CURRENT CP TIME IS      19.510 WHICH IS      5.000 SECONDS INTO THE RUN
                              OVERLAY  CALLED WITH CONTYPE =    1

  TARGET NO.  1 HAS HAD 389 HITS.  IT IS AT A RANGE OF    260. AND AZIMUTH OF 39.36

      FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME     19.511
           EXECUTION TOOK       .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.


*****************************************************************************************

        HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.55000000

                         CURRENT CP TIME IS      19.513 WHICH IS      5.004 SECONDS INTO THE RUN
                              OVERLAY  CALLED WITH CONTYPE =    1

  TARGET NO.  1 HAS HAD  25 HITS.  IT IS AT A RANGE OF    244. AND AZIMUTH OF  39.84

      FUNCTIONAL MODULE  HOT SPOT TRACK      ENDED AT CP TIME     19.514
           EXECUTION TOOK       .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.


*****************************************************************************************

        CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.55001000

                         CURRENT CP TIME IS      19.517 WHICH IS      5.007 SECONDS INTO THE RUN
                              OVERLAY  CALLED WITH CONTYPE =    1

  TARGET NO.  1 HAS HAD 390 HITS.  IT IS AT A RANGE OF    244. AND AZIMUTH OF 39.84

      FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME     19.517
           EXECUTION TOOK       .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.


*****************************************************************************************

        HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    49.65000000

                         CURRENT CP TIME IS      19.520 WHICH IS      5.010 SECONDS INTO THE RUN
                              OVERLAY  CALLED WITH CONTYPE =    1
```

```
TARGET NO.  1 HAS HAD  26 HITS.  IT IS AT A RANGE OF   228. AND AZIMUTH OF  40.37

     FUNCTIONAL MODULE  HOT SPOT TRACK      ENDED AT CP TIME     19.521
          EXECUTION TOOK       .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**********************************************************************************************

          CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.65001007

                    - CURRENT CP TIME IS      19.523 WHICH IS      5.014 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD 391 HITS.  IT IS AT A RANGE OF   228. AND AZIMUTH OF 40.37

     FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME     19.524
          EXECUTION TOOK       .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**********************************************************************************************

          HOT SPOT TRACK      SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.75000000

                      CURRENT CP TIME IS      19.525 WHICH IS      5.017 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD  27 HITS.  IT IS AT A RANGE OF   214. AND AZIMUTH OF  40.98

     FUNCTIONAL MODULE  HOT SPOT TRACK      ENDED AT CP TIME     19.527
          EXECUTION TOOK       .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**********************************************************************************************

          CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME    48.75001011

                      CURRENT CP TIME IS      19.531 WHICH IS      5.020 SECONDS INTO THE RUN
                         OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD 392 HITS.  IT IS AT A RANGE OF   214. AND AZIMUTH OF 40.98

     FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME     19.530
          EXECUTION TOOK       .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.

**********************************************************************************************
```

```
     HOT SPOT TRACK       SIMULATION MODULE TO BEGIN EXECUTION AT TIME      48.85000000

                           CURRENT CP TIME IS       13.535 WHICH IS       5.023 SECONDS INTO THE RUN
                                OVERLAY  CALLED WITH CONTYPE =      1

TARGET NO.  1 HAS HAD  28 HITS.  IT IS AT A RANGE OF    201. AND AZIMUTH OF  41.68

      FUNCTIONAL MODULE  HOT SPOT TRACK       ENDED AT CP TIME      19.534
           EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.


**********************************************************************************************


     CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME      48.85001000

                           CURRENT CP TIME IS       13.536 WHICH IS       5.027 SECONDS INTO THE RUN
                                OVERLAY  CALLED WITH CONTYPE =      1

TARGET NO.  1 HAS HAD 393 HITS.  IT IS AT A RANGE OF    201. AND AZIMUTH OF 41.68

      FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME      19.537
           EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.


**********************************************************************************************


     HOT SPOT TRACK       SIMULATION MODULE TO BEGIN EXECUTION AT TIME      48.95000000

                           CURRENT CP TIME IS       13.539 WHICH IS       5.030 SECONDS INTO THE RUN
                                OVERLAY  CALLED WITH CONTYPE =      1

TARGET NO.  1 HAS HAD  29 HITS.  IT IS AT A RANGE OF    189. AND AZIMUTH OF  42.50

      FUNCTIONAL MODULE  HOT SPOT TRACK       ENDED AT CP TIME      19.540
           EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.


**********************************************************************************************


     CORRELATION TRACK   SIMULATION MODULE TO BEGIN EXECUTION AT TIME      48.95001000

                           CURRENT CP TIME IS       13.543 WHICH IS       5.033 SECONDS INTO THE RUN
                                OVERLAY  CALLED WITH CONTYPE =      1

TARGET NO.  1 HAS HAD 394 HITS.  IT IS AT A RANGE OF    189. AND AZIMUTH OF 42.50

      FUNCTIONAL MODULE  CORRELATION TRACK   ENDED AT CP TIME      19.543
           EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.
```

```
*********************************************************************************

     DROP TRACK          SIMULATION MODULE TO BEGIN EXECUTION AT TIME    49.00000000

                      CURRENT CP TIME IS      19.546 WHICH IS      5.036 SECONDS INTO THE RUN
                           OVERLAY  CALLED WITH CONTYPE =     1

   FUNCTIONAL MODULE  DROP TRACK            ENDED AT CP TIME      19.546
        EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.


*********************************************************************************

     TRACK WHILE SCAN    SIMULATION MODULE TO BEGIN EXECUTION AT TIME    49.00000000

                      CURRENT CP TIME IS      19.549 WHICH IS      5.039 SECONDS INTO THE RUN
                           OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  2 IS AT AN ALTITUDE OF        198. AND RANGE FROM RADAR OF        451.
     NO. OF T-W-S PULSES =    2

   FUNCTIONAL MODULE  TRACK WHILE SCAN    ENDED AT CP TIME      19.550
        EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.


*********************************************************************************

     TRACK WHILE SCAN    SIMULATION MODULE TO BEGIN EXECUTION AT TIME    49.00000000

                      CURRENT CP TIME IS      19.552 WHICH IS      5.042 SECONDS INTO THE RUN
                           OVERLAY  CALLED WITH CONTYPE =     1

TARGET NO.  3 IS AT AN ALTITUDE OF         51. AND RANGE FROM RADAR OF        401.
     NO. OF T-W-S PULSES =    2

   FUNCTIONAL MODULE  TRACK WHILE SCAN    ENDED AT CP TIME      19.553
        EXECUTION TOOK        .001 CP SECONDS AND    26 WORDS WERE INPUT TO IT.


*********************************************************************************

     TRACK WHILE SCAN    SIMULATION MODULE TO BEGIN EXECUTION AT TIME    49.00000000

                      CURRENT CP TIME IS      19.555 WHICH IS      5.046 SECONDS INTO THE RUN
                           OVERLAY  CALLED WITH CONTYPE =     1
```

```
TARGET NO. 4 IS AT AN ALTITUDE OF        189. AND RANGE FROM RADAR OF        270.
     NO. OF T-W-S PULSES =   2

     FUNCTIONAL MODULE  TRACK WHILE SCAN     ENDED AT CP TIME     19.556
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

****************************************************************************************************

       TRAJECTORY COMP     SIMULATION MODULE TO BEGIN EXECUTION AT TIME     49.00000000

                  CURRENT CP TIME IS      19.559 WHICH IS      5.049 SECONDS INTO THE RUN
                       OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  1 HAS HAD A TRAJECTORY COMPARISON.  THERE APPEARS TO BE ENOUGH DEFLECTION FROM NOMINAL TO ASSUME KILLED.

     FUNCTIONAL MODULE  TRAJECTORY COMP      ENDED AT CP TIME     19.559
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

****************************************************************************************************

       SEARCH CONTROL      SIMULATION MODULE TO BEGIN EXECUTION AT TIME     49.00000000

                  CURRENT CP TIME IS      19.561 WHICH IS      5.052 SECONDS INTO THE RUN
                       OVERLAY  CALLED WITH CONTYPE =    1

     FUNCTIONAL MODULE  SEARCH CONTROL      ENDED AT CP TIME     19.562
          EXECUTION TOOK        .000 CP SECONDS AND      1 WORDS WERE INPUT TO IT.

****************************************************************************************************

       TRACK WHILE SCAN    SIMULATION MODULE TO BEGIN EXECUTION AT TIME     50.00000000

                  CURRENT CP TIME IS      19.564 WHICH IS      5.055 SECONDS INTO THE RUN
                       OVERLAY  CALLED WITH CONTYPE =    1

TARGET NO.  2 IS AT AN ALTITUDE OF        100. AND RANGE FROM RADAR OF        274.
     NO. OF T-W-S PULSES =    3

     FUNCTIONAL MODULE  TRACK WHILE SCAN     ENDED AT CP TIME     19.565
          EXECUTION TOOK        .001 CP SECONDS AND     26 WORDS WERE INPUT TO IT.

****************************************************************************************************
```