RL-TR-91-343
Final Technical Report
December 1991

INTEGRATED MULTI-DOMAIN RADAR DEMONSTRATION

Digicomp Research Corporation and Sensis Corporation

Carol C. Shilepsky, Mary Bucknell, Rick Taylor

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-91-343 has been reviewed and is approved for publication.

APPROVED:

VINCENT C. VANNICOLA
Project Engineer

FOR THE COMMANDER:

JAMES W. YOUNGBERG, Lt Col, USAF
Deputy Director of Surveillance

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL(OCTS) Griffiss AFB, NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Final Apr 89 - Apr 91

4. TITLE AND SUBTITLE
INTEGRATED MULTI-DOMAIN RADAR DEMONSTRATION

5. FUNDING NUMBERS
C - F30602-89-C-0045
PE - 62702F
PR - 4506
TA - 11
WU - OC

6. AUTHOR(S)
Carol C. Shilepsky, Mary Bucknell, Rick Taylor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Digicomp Research Corp
Terrace Hill
Ithaca NY 14850

Sensis Corp
5793 Widewaters Parkway
DeWitt NY 13214

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (OCTS)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-91-343

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Vincent C. Vannicola/OCTS/(315) 330-4437

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
The objective of the IMRD program is to apply artificial intelligence techniques to the adaptive control of a state-of-the-art radar environment. The radar operates in the C-Band and is located within the Rome Laboratory Surveillance Facility (RLSF), Bldg 106, Griffiss AFB. The artificial intelligence is embedded in an adaptive control expert system which is written in Prolog.

This system identifies sources of interference in each antenna beam position of the surveillance region and responds with the appropriate adaptive controls to maximize the probability of target detection consistent with operator-specified tactical objectives. In addition, the system has the following features:

1. Radar inputs provided by a real, as opposed to a simulated, radar.
2. Real-time operation with one scan response time of ten seconds or less.
3. Modular design for rulebase and system evolution.
4. Extensive parameterization for different radar configurations and operational specifications.
5. Control of a large number of radar parameters.

The report includes IMRD organization, parameterization options for configuring it to different environments, the expert system software development, and results.

14. SUBJECT TERMS
Radar, Signal Processing, Artificial Intelligence Control

15. NUMBER OF PAGES
172

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Contents

List of Figures

List of Tables

1 Introduction

For the past several years, radar technology development has focused on phased array radars of many types. These radars lend themselves to multi-purpose, multi-function uses and to automatic control of their modes of operation and parameters. Traditionally, these mechanisms have been implemented through pre-programmed or template control.

In parallel with phased array radar developments, work has been done in the field of artificial intelligence (AI). AI is a branch of computer science concerned with modeling human intelligence and exploiting such models within a computer environment to obtain useful information. AI is particularly effective when knowledge is incomplete, uncertain, and/or based on heuristics. Expert systems is a subfield of AI in which knowledge or expertise is captured in the form of facts and rules, and an inference component deduces conclusions from these facts and rules.

Lately, there is growing interest in applying AI technology to adaptive radar control. Using information assessed from a radar environment to control radar parameters and mode of operation requires judgement and heuristics that can be more easily embodied in rules than in algorithms. Moreover, the dynamic environment and the need for rapid control decisions require automated computer support that can best be provided by an expert system.

An earlier Rome Laboratory project, Adaptive Control for Multi-Domain Sensor Processor (ACMDSP, Contract F30602-86-C-0204), applied AI and expert system technology to automating adaptive radar control within a simulated surveillance environment. The system was intended to be used in a laboratory environment as a demonstration and development tool, and is documented in *Adaptive Controller for Multi-Domain Sensor Processors*, dated September 1988.

This contract, Integrated Multi-Domain Radar Demonstration (IMRD), combines AI and state-of-the-art experimental radar technology within a real, as opposed to a hypothetical, radar environment. Its major objectives include real-time performance, a framework for rulebase evolution, end-product flexibility for migration to other hardware environments, and effective use of existing resources.

In the resulting testbed, artificial intelligence is embedded in an adaptive control expert system which identifies interference in radar returns and recommends the appropriate responses to be implemented during the next scan. Categories of identified interference include ground clutter, weather clutter, and jammers. Jammers are discriminated by wideband or narrowband, by pulse or continuous, and by mainlobe or sidelobe. Countermeasures include frequency changes, sidelobe blanking, sidelobe cancelling, polarization changes, and adaptive moving target indicator (MTI). Additionally, the surveillance region can be partitioned into threat zones prioritized by the importance of maintaining detection within them. Antenna beam dwell time is allocated to the different regions so as to achieve the highest probabilities of target detection in the highest threat regions.

Chapter 2 of this report is an executive summary. The remaining chapters present the system design and development (Chapter 3), the resulting demonstration (Chapter 4), and our conclusions and recommendations (Chapter 5). Appendices document specific system features.

Specific instructions for system configuration and use are contained in the *User Manual for Integrated Multi-Domain Radar Demonstration* [UM] dated May 30, 1991. The *Test Plan and Procedures for Integrated Multi-Domain Radar Demonstration* [TP], April 17, 1991, documents the tests performed during the final system demonstration.

2 Executive Summary

The objective of the IMRD program is to apply artificial intelligence techniques to the adaptive control of a state-of-the-art radar environment. The radar operates in the C-Band and is located within the Rome Laboratory Surveillance Facility (RLSF), Building 106, Griffiss Air Force Base (GAFB). The artificial intelligence is embedded in an adaptive control expert system which is written in Prolog.

This system identifies sources of interference in each antenna beam position of the surveillance region and responds with the appropriate adaptive controls to maximize the probability of target detection consistent with operator-specified tactical objectives. In addition, the system has the following features:

1. Radar inputs provided by a real, as opposed to a simulated, radar.
2. Real-time operation with one scan response time of ten seconds or less.
3. Modular design for rulebase and system evolution.
4. Extensive parameterization for different radar configurations and operational specifications.
5. Control of a large number of radar parameters.

The following sections review the top-level IMRD organization, parameterization options for configuring it to different environments, the expert system software development, and our results.

2.1 IMRD Organization

The IMRD block diagram is shown in Figure 2-1. It can be conceptually divided into five major subsystems, all operating in real-time:

1. C-Band radar and related equipment.
2. Environmental signal processing (ESP), which resides on a Star Technologies ST-100 array processor.
3. Detection signal processing (DSP) including moving target indicator (MTI), and continuous false alarm rate (CFAR) processing, which provides video to a UPA-62 PPI display.

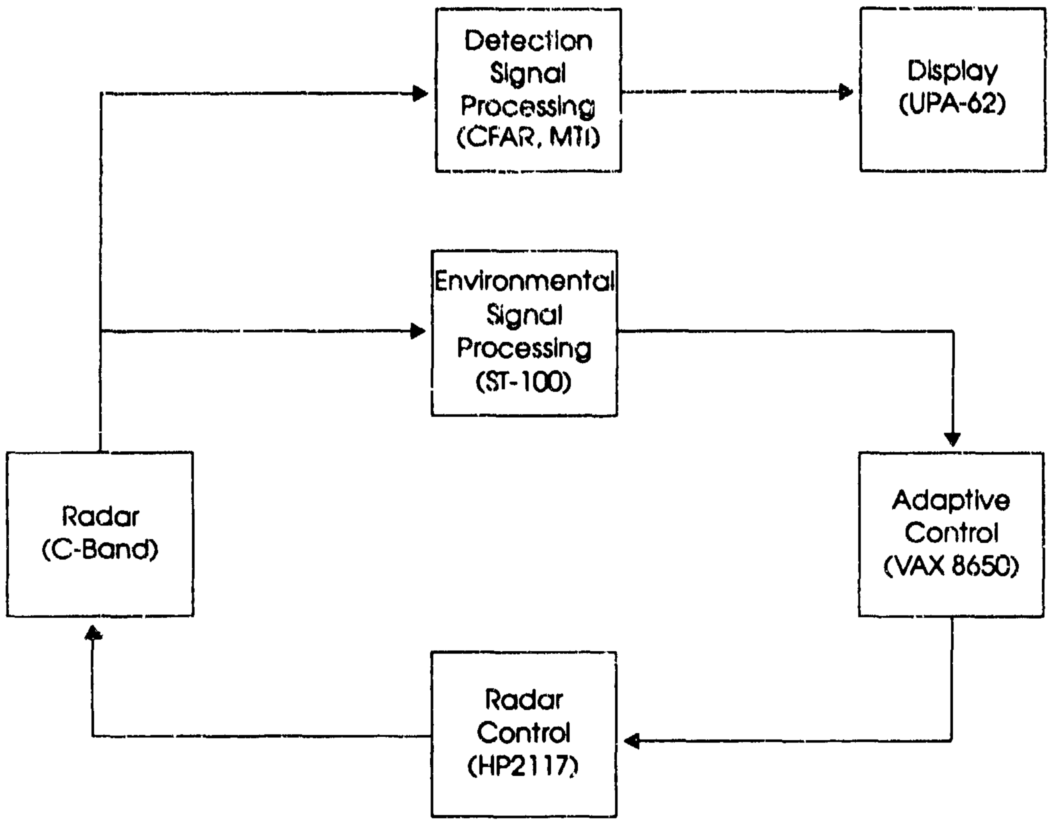

Figure 2-1: IMRD Block Diagram

4. An Adaptive Controller, which resides on a Digital Equipment Corporation VAX 8650.

5. A Radar Controller, which resides on a Hewlett Packard HP2117.

The C-Band radar processes consecutive beams in a scan region of up to 90° in azimuth. Azimuth range and number of beams are among the user-supplied parameters. The top part of Figure 2-2 illustrates a typical surveillance region containing 40 beams over 45° in azimuth.

The radar is step-scanned over the azimuth range of 90° with variable dwell time per beam (apportioned by the Adaptive Controller so that the total scan time is within a user-specified constraint, nominally set to ten seconds). As indicated in the bottom of Figure 2-2, the waveform for each beam dwell is divided into three processing intervals:

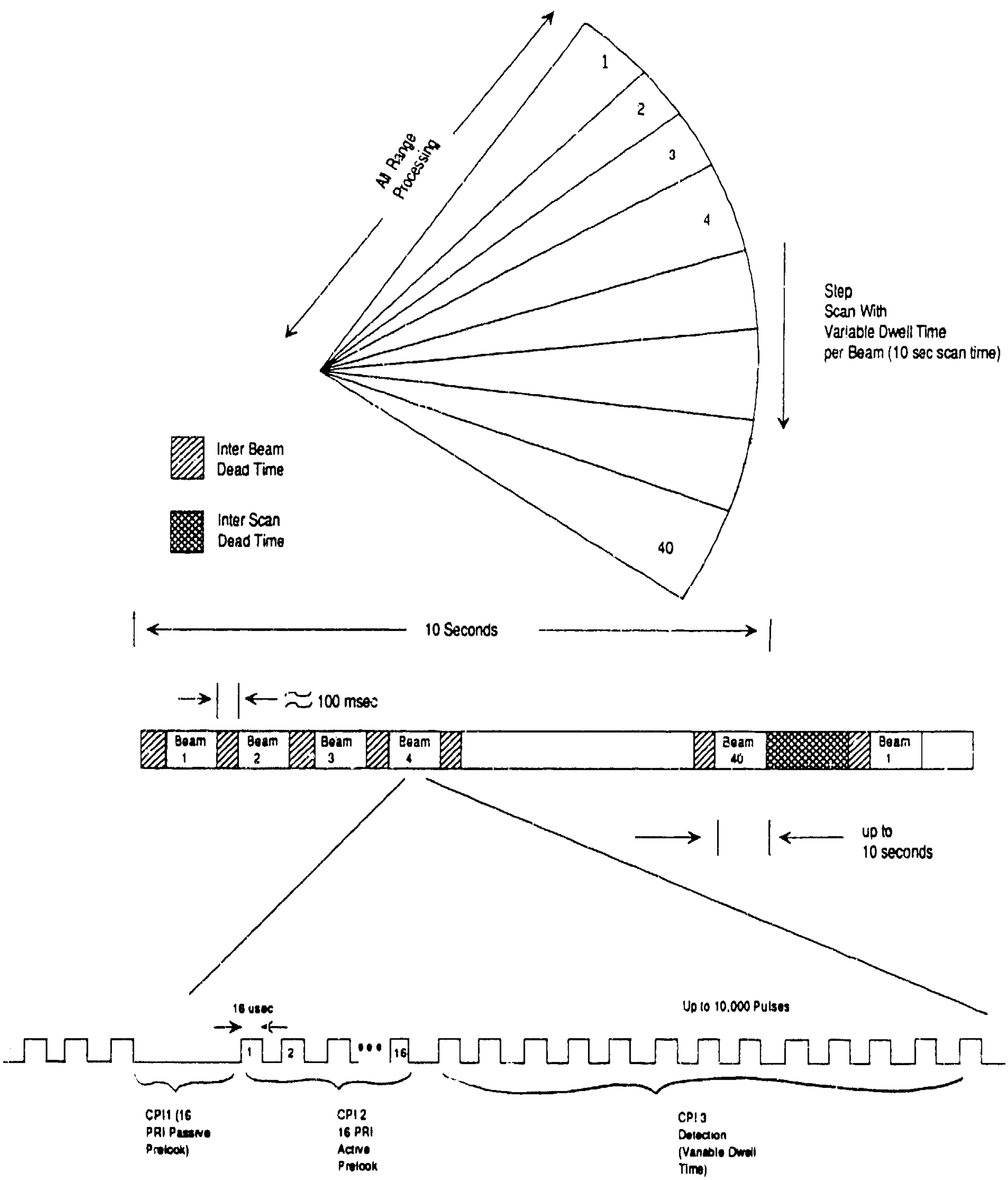

Figure 2-2: Real-Time Scan Timeline

1. A passive listening interval, designated Coherent Processing Interval 1 (CPI1), for detecting jammers and radio frequency interference (RFI) sources.
2. An active environmental assessment interval, designated Coherent Processing Interval 2 (CPI2), for detecting ground clutter and weather.
3. An adaptive target detection waveform, designated Coherent Processing Interval 3 (CPI3), with appropriate electronic counter-counter measures (ECCM) enabled, as determined by the expert system.

CPI1 and CPI2 are passed to the ST-100 array processor which averages the passive dwell data to enable jammer detection and calculates the jammer frequency spectrum to enable bandwidth assessment. For the active dwell, the ST-100 finds the range/doppler characteristics via a doppler filter bank to detect ground and weather clutter. The range-doppler surface has 900 range cells and 16 doppler bins for a total of 14,400 measurements per beam position. CPI3 is sent through the real-time detection signal hardware to a UPA-62 PPI for display.

The results of ST-100 processing are passed as Beam Scan Data inputs to the Adaptive Controller expert system which is written in Prolog and resides on the VAX 8650. The Adaptive Controller analyzes these data for sources of interference, determines what control changes should be made during the next scan to maximize probability of target detection in conjunction with prioritized regions of coverage, and sends the resulting Radar Control Parameters to the radar via the HP2117 Radar Controller.

The Adaptive Controller can also interface with a Flexible Radar Data Executive (FRDE) for stand-alone testing and demonstration. The FRDE software resides on the VAX and is spawned as a process by the Adaptive Controller when the user selects this option. In this case, the FRDE sends stored radar data to the Adaptive Controller and retrieves Adaptive Controller outputs, displaying them and/or saving them in a file.

Radar control software resides on an HP2117 computer. Expert system outputs are sent from the VAX to the HP2117. The latter converts them to hardware commands and controls the waveform timing and signal processing for the next scan.

Table 1 lists the the RLSF equipment used to implement the functions of Figure 2-1, the specific interference identified, and adaptive controls applied. Figure 2-3 shows the hardware/software configuration.

2.2 Parameterization

The adaptive control software can be configured for a wide variety of radar hardware environments and user tactical objectives. Table 2 summarizes the parameters which

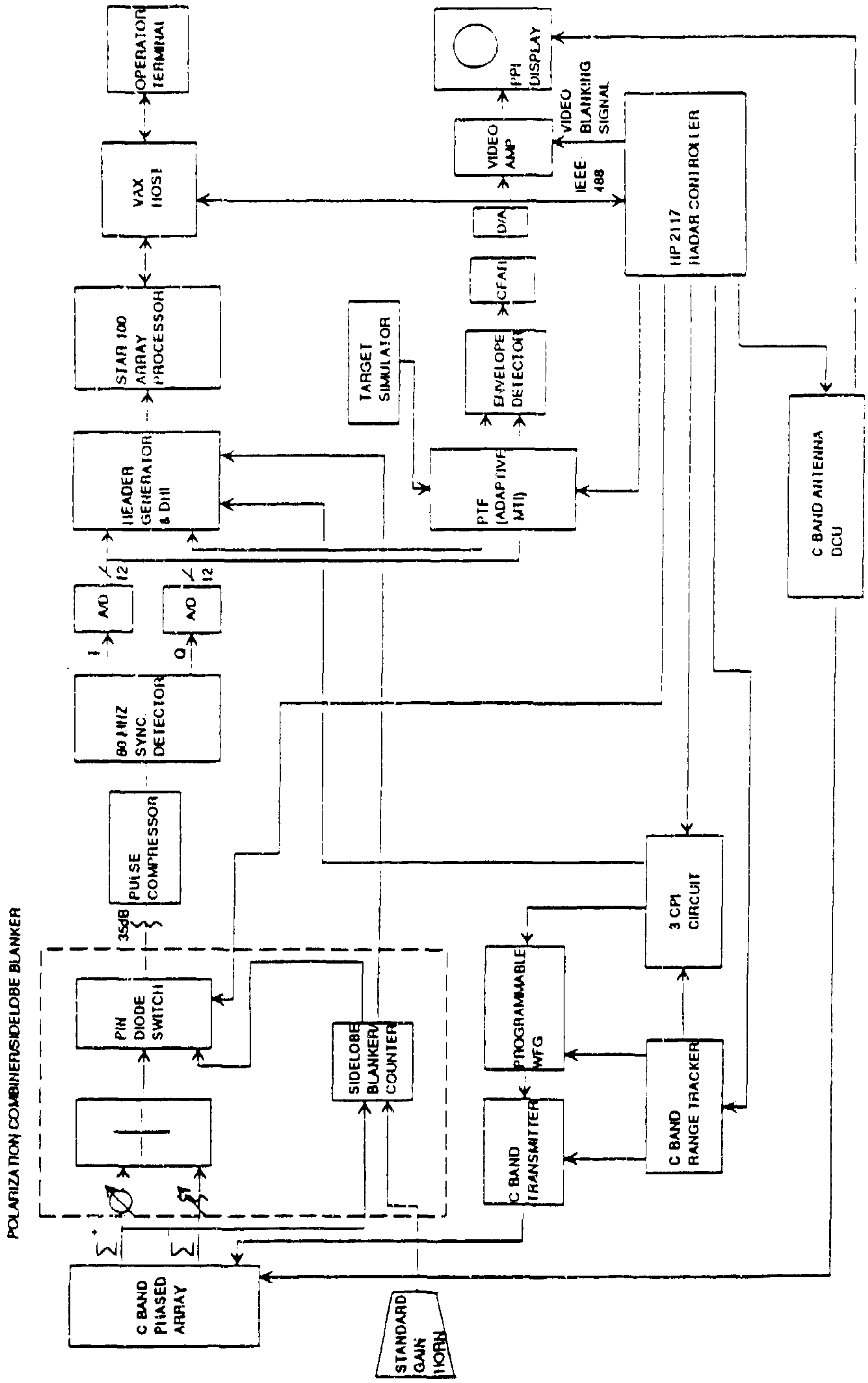

Figure 2-3: Hardware/Software Configuration

Subsystem	RLSF Implementation
Radar system	C-Band
Adaptive control host computer	VAX 8650
Adaptive control software	Prolog driver with Pascal procedure calls and spawned FORTRAN interface processes. Expert system rules in Prolog; computation and data structures in Pascal.
Environmental assessment signal processing host computer	ST-100
Radar control host computer	HP2117
Detection display	UPA-62 PPI
AI display	VT100 type terminal
Waveform parameters • PRI • Bandwidth • Pulse duration • Pulse coding • Doppler resolution • Spatial resolution • Polarization	 3.0 msecs 1.0 MHz 16 μsecs LFM 25 HZ 1° azimuth by 150m range Matched or orthogonal linear polarizations with provision for circular
Interference identified	• Jammers: – Narrowband/wideband – Mainlobe/sidelobe – CW/asynchronous pulsed • Ground clutter • Weather clutter
Radar controls	• Sidelobe blanking on/off • Transmit polarization • Frequency hopping • Variable MTI weights • Variable beam dwell time • Sidelobe cancelling on/off

Table 1: RLSF Implementation

are selected by the user through a file or interactive entry. The Radar Design Parameters and Radar Environment Parameters are specific to RLSF C-Band testbed and would be changed only to accommodate a different hardware environment. The remaining parameters allow the user to set adaptive control objectives and output.

2.3 Expert System Software Development

The Adaptive Controller expert system uses Beam Scan Data inputs from one scan to select Radar Control Parameters for the next scan. Figure 2-4 illustrates the software modules that support this process:

1. A **User Interface** allows the user to specify default input parameters or to enter the parameters interactively.
2. An **AI Engine** assesses the current environment and selects radar control options for the next scan.
3. An **AI Display** displays user-selected parameters on the screen and/or saves them in a file.

Figure 2-5 shows a breakdown of the AI Engine subproblems: ID Passive Interference, ID Active Interference, Determine Applicable ECCM, and Allocate Radar Resources.

ID Passive Interference uses CPI1 passive dwell data to detect CW and asynchronous pulsed (ASP) jammers. If a jammer is found, the peak jammer-to-noise ratio (JNR), jammer range, and mainlobe duty factor are reported.

ID Active Interference uses CPI2 active dwell data to detect weather and ground clutter. If active interference is identified, the peak clutter-to-noise ratio (CNR), the maximum range of the clutter, and the doppler extent of the clutter are reported.

Determine Applicable ECCM analyzes the interference identifications and selects appropriate ECCM, ranking plans that counteract the interference by increasing probabilities of target detection.

Allocate Radar Resources uses the plans developed by Determine Applicable ECCM along with user-assigned Priority Zones and Quality Options Tables to select the best operating plan for the next scan that meets a user-specified scan time.

The Adaptive Controller contains approximately 10,000 lines of code written in Prolog, with Pascal procedure calls and spawned FORTRAN processes. Prolog is used for the main driver, rulebases, and expert system inferencing. Pascal is used for computation and data structures. VAX, HP2117, and the stand-alone FRDE processes are written in FORTRAN.

Type	Description
Radar Design Parameters	Defines physical characteristics c the radar, desired detection range, desired target cross section
Radar Environment Parameters	Thermal noise, ground and weather clutter maps* used for interference identification
Initial Radar Control Parameters	Radar Control Parameter settings for the first scan
Priority Zones	Divides scan region into zones, based on the importance of maintaining detection within them
Quality Option Plans	Operating plans used to allocate beam dwell time by Priority Zones
AI Control Parameters	
• Desired Scan Time	Maximum time allowed per scan
• AI On	Selects full adaptive control or interference identification only
• ECCM Usage	Selects which ECCMs are available in a given radar system
• Quality Option Selection	Allows automatic override of AI Quality Option Plan selection, regardless of scan time constraints
• Configuration Selection	Selects radar interface (live data) or stand-alone (taped data) software configuration
AI Output Parameters	Selects parameters displayed on the VT100 or stored in a file

* Boolean maps that specify regions in range/doppler where clutter is expected.

Table 2: User-Selectable Parameters

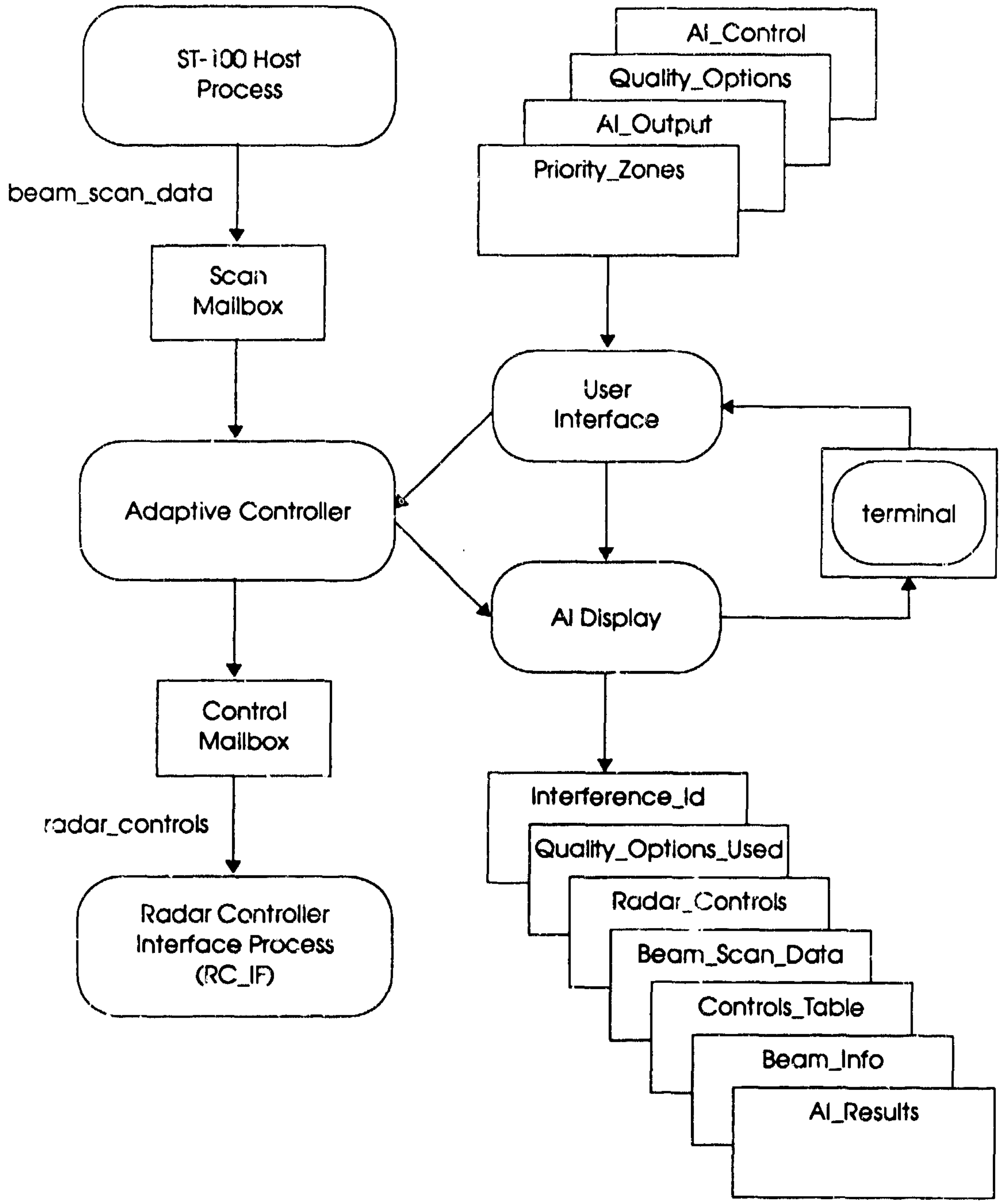

Figure 2-4: Adaptive Controller Configuration

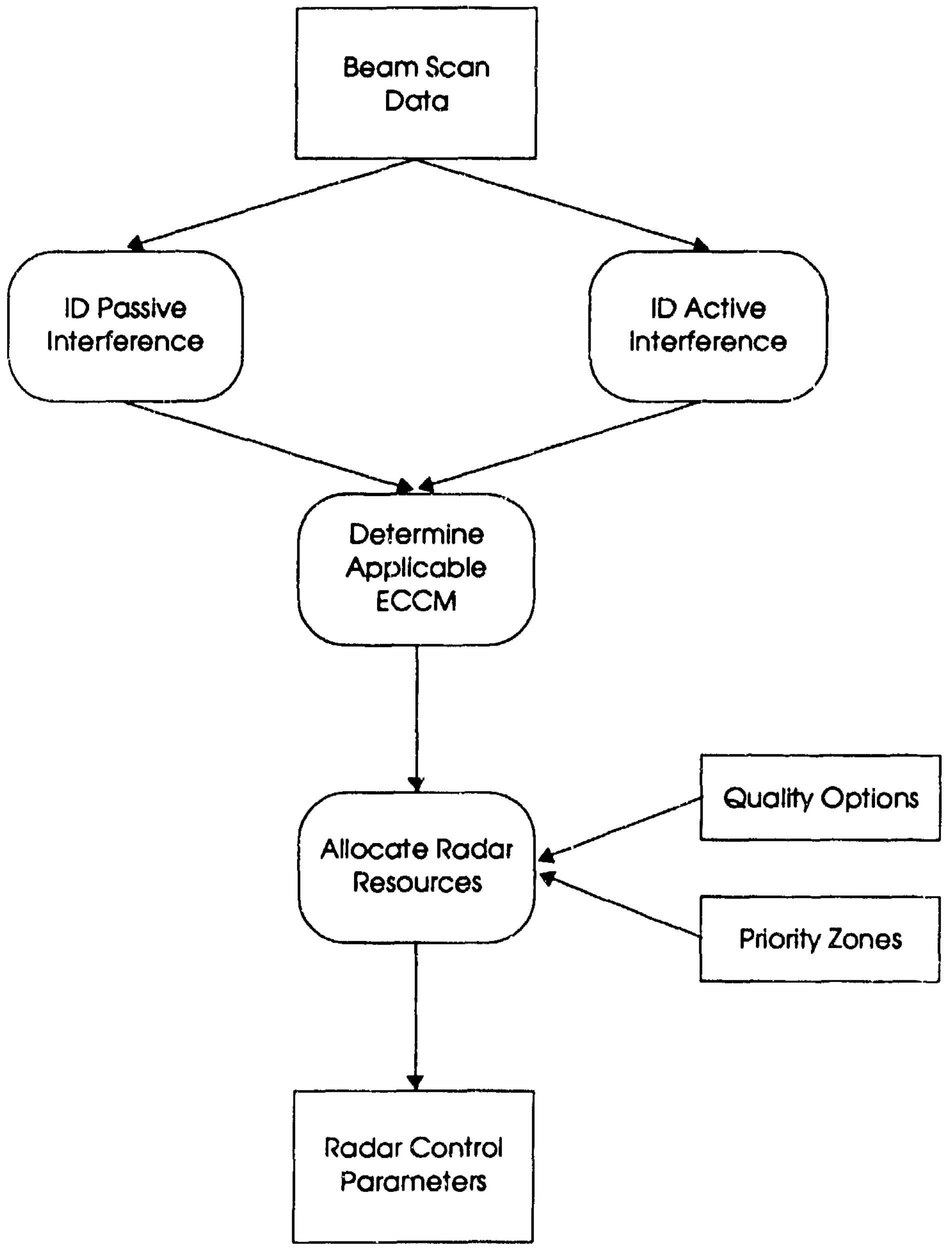

Figure 2-5: AI Subproblem Decomposition

Prolog has been especially convenient for system development: the similarities between Prolog predicates and English rules make the expert system visible for rulebase formulation and evolution; its built-in logic handles rulebase inferencing transparently; it supports foreign language interfaces well for incorporating code in other languages.

2.4 Demonstration and Training

Formal demonstration and training were held on April 17 and 18, 1991. The following capabilities were demonstrated:

1. Identification of single and multiple sources of interference in the radar data.
2. Control of radar parameters including frequency, sidelobe blanking, sidelobe cancelling,[1] polarization, and MTI weights.
3. Differential allocations of dwell time based on user-specified priority zones and scan times.
4. Observation of targets of opportunity, as available, and simulated targets injected into the display.
5. Performance within the government-specified ten second scan time.

Training consisted of demonstrating how to set-up the radar, how to to initialize the various computer systems, and how to run the software. In addition, users had the opportunity to input and modify various policy control parameters while the radar was running. The convenient menus enabled new ers to modify parameters with little assistance or training.

2.5 Summary

This contract has successfully applied artificial intelligence to adaptive radar control. The resulting IMRD hardware/software configuration operates in real-t in the presence of multiple sources of interference, resulting in observable improvements in target detection.

We have provided the government with an AI testbed and performance benchmark for continued adaptive control development. IMRD modular design, combined with

[1]Although SLC equipment was unavailable for the tests, the Adaptive Controller selected this option when appropriate and printed a message to this effect.

extensive parameterization, supports rulebase evolution and tailoring the adaptive control software to other hardware environments. Comprehensive hardware and software documentation will facilitate this continued development.

3 System Design and Development

Figure 2-1 shows the high-level functionality of an adaptive control radar system and the resulting RLSF implementation. The major components are the radar, signal processing, adaptive control, display, and radar control. The radar transmits a waveform and receives returns in which targets may be obscured by different sources of interference. Signal processing includes real-time processing necessary for display of the radar returns (DSP) as well as spatial and spectral analysis of data sent to the Adaptive Controller for ascertaining interference (ESP). The Adaptive Controller identifies sources of interference in the processed data and selects control changes for the next scan. The Radar Controller implements the recommended changes.

We conducted a series of tradeoff studies to assess available hardware and software, and to propose an implementation for this basic configuration. Alternate RLSF architectures were evaluated with respect to their capabilities for satisfying the following demonstration goals:

1. Use available equipment for all functions except possibly for minor modifications to equipment interfaces.
2. Select an approach with low enough risk to ensure a high probability of success, but that would still allow a wide range of experiments.
3. Maximize processing speed within the constraints of available equipment to effect a real-time demonstration.
4. Control a sufficient number of radar parameters with the Adaptive Controller for a credible demonstration.
5. Establish performance and evaluation criteria.

Specific concerns in implementing the system architecture are discussed in the following paragraphs. They include:

1. How best to incorporate the AI adaptive control simulation performed on a previous Rome Laboratory contract.
2. The choice of L- versus C-Band radar.
3. Signal processing features, platform, and implementation.
4. Adaptive control features, platform, and implementation.
5. Radar control platform and implementation.
6. Display features and implementation.

3.1 Survey of Existing Resources

3.1.1 ACMDSP Simulation

One of the first tasks of the contract was to evaluate the applicability of the ACMDSP simulation (Contract F30602-86-C-0204) for incorporation into the AI testbed. This section gives the conclusions of this evaluation at a summary level. Additional details regarding ACMDSP applicability are in the *Interim Report for Integrated Multi-Domain Radar Demonstration* [IR], August 31, 1989, pages 5-16.

The ACMDSP, developed under a prior Rome Labs contract, simulates a tactical radar environment and performs adaptive control on the resulting data. The radar is specified by operator inputs such as coverage, maximum instrumented range, azimuth beam width, elevation beam width, scan time, frequency, target cross section, and available ECCM. Operator-selected target and interference characteristics are used to generate the radar data that would be produced by the hypothetical environment. These data are analyzed for sources of interference, and the simulated radar operation is modified in the next scan to maximize detection range in the presence of that interference.

The ACMDSP main software components are a Radar Model that simulates the radar environment and aircraft targets, and an Adaptive Controller that processes the radar data for a scan and determines the operating plan for the next scan. Figure 3-1, taken from the ACMDSP Final Report [ACMDSP], shows this organization and the major program inputs. Communication between the Radar Model and the Adaptive Controller is through the VAX/VMS mailboxes. The Adaptive Controller is written in Prolog, and interfaces to routines written in Pascal which is used for data structure implementation, mailbox utilities, and computation.

At the start of this contract, Digicomp/Sensis had hoped to transfer the ACMDSP Adaptive Controller to the RLSF with minor changes to the content of the scan data inputs and the control parameter outputs. The Radar Model would be replaced by RLSF equipment, and the VMS mailboxes would be maintained for communication between Adaptive Controller and RLSF hardware. The Radar Model would be modified as necessary to reflect discrepancies with the real radar, and a Radar Parameters File would be generated with the RLSF C-Band characteristics. The Radar Model could then simulate the real environment for testing the IMRD Adaptive Controller or for evaluating new adaptive control policies, and could be used as a demonstration when operating the real radar was infeasible.

However, we found that the ACMDSP was not as straightforward to transfer as we had hoped. The Radar Model does not provide a true simulation since it produces the measured values of interference parameters as opposed to the signals themselves.

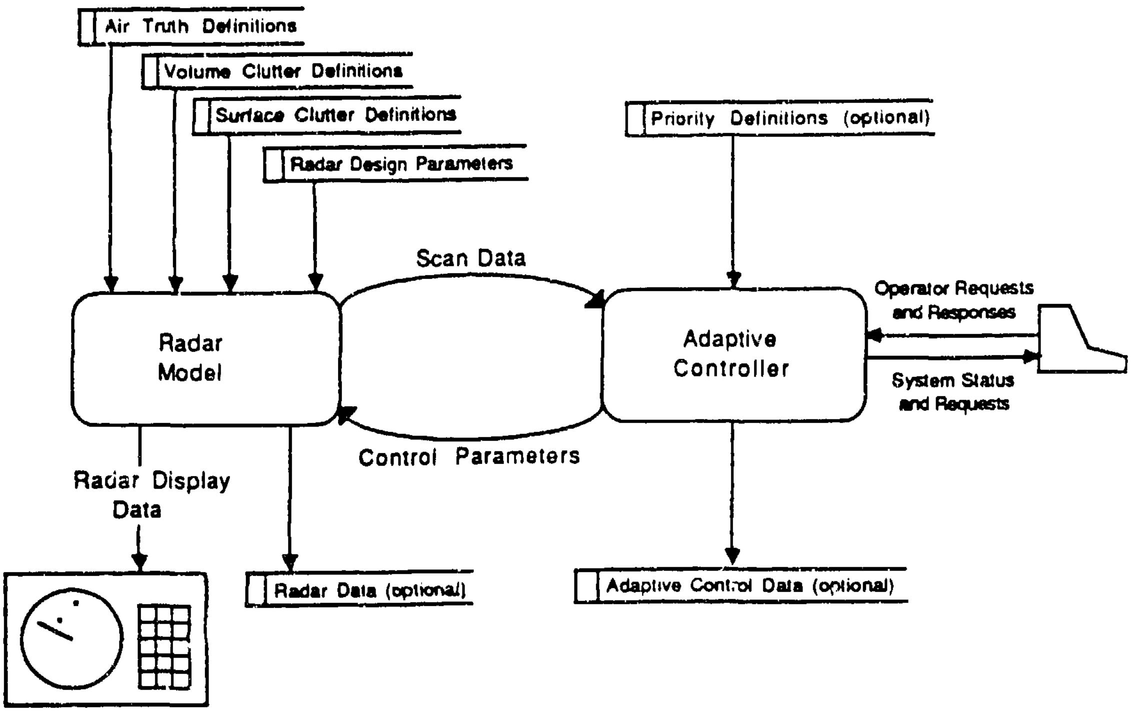

Figure 3-1: ACMDSP System Block Diagram

Moreover, substantial changes were needed in its adaptive control expertise (as discussed in [IR]). For example, all non-zero doppler data in the ACMDSP is aggregated into a single number with jammer and clutter determinations based on comparison between zero and non-zero components; we found it necessary to use unaggregated data from the entire spectrum to identify interference. Moreover, neither signal processing assumptions nor inputs to and outputs from the simulated radar are compatible with what would be generated in the RLSF radar environment. For example, we determined that a single waveform would be insufficient for identifying interference such as jammers and clutter *and* for detecting targets with an adapted waveform.

Despite these findings, the ACMDSP was a good starting point for the IMRD effort. Much of the design and some of the data structures were directly reusable. Those that were not provided a good first approximation. Many of the code modules—in particular, mailbox utilities—were reuseable. The Prolog/Pascal coordination was adopted, combining the best features of both languages: Prolog for AI inferencing and Pascal for data structures, mailbox utilities, and computation.

3.1.2 L- Versus C-Band Radar

The contract statement of work (SOW) allowed for the use of either the L- or C-Band system for IMRD demo. A review of RLSF resources was conducted and various equipment configurations were evaluated to determine which L- or C-Band radar parameters were controllable as a result of adaptive control decisions, and, on this basis, to select one for IMRD development.

Since the L- and C-Band RLSF radars share the same processing equipment, both systems have essentially the same capability for signal processing, environmental assessment, and AI decisions. The major differences are in the degree to which each radar can be controlled. Table 3 assesses the control capabilities of each system as was projected for late 1989, with respect to key experimental objectives. For each comparison, the system we considered most capable of meeting the objective is highlighted in bold text.

The table shows that, for this application, the C-Band radar offers substantial advantages over the L-Band system, in many areas, primarily due to its phase steering capability, provision for adaptive mainlobe/sidelobe cancellation, adaptive polarization capability, and the existence of GFE (government furnished equipment) jammer sources. One projected deficiency of the C-Band system, limited range processing, was later eliminated by using a direct interval frequency all-range interface to the signal processing equipment see Section 3.2). The limited elevation coverage of the C-Band antenna was initially a concern, but later experiments verified that a reasonable number of targets of opportunity could be observed near the radar horizon.

The projected advantage—indicated in the table—of the C-Band radar with respect to sidelobe blanking proved to be less than expected. Initially, we projected that the weather blanker could be converted to a sidelobe blanker (SLB) with only minor modifications. However, as SLB development progressed, the design evolved into a completely new circuit independent from the weather blanker, and the weather blanker was returned to its original configuration.

Therefore, the results of the tradeoff indicated that a meaningful demonstration could be conducted with C-Band radar by configuring the experiment as follows:

1. Configure the C-Band system for adaptive control of the sidelobe blanker, sidelobe canceller, RF frequency, polarization, MTI (using the programmable transverse filter), and energy per beam (dwell time).

2. Operate the radar over an azimuth surveillance volume commensurate with full beam adaptability with electronic azimuth steering. The azimuth coverage will be up to ±45 degrees, the electronic scan limit of the C-Band antenna.

3. Allow some dead time (with the radar transmitter disabled) between beam positions (i.e., beam dwells) to enable the synthesizer, waveform generator, and other RLSF equipment to receive the programmable parameters for the next beam dwell and to enable enough time to process the data received during each beam dwell.

Objective	L-Band	C-Band
Support demonstrations using far-field jammer equipment	RADCSL has no operating far-field L-Band jammer sources that can be remotely controlled	**Has two remotely-controlled far-field jammer sources (controllable from HP-2117)**
Controllable transmit and receive polarization	No	**Required a circuit to form receive polarization**
Beam to beam dwell time variability	Not at present (1989)	**Yes, via DCU interface and electronic steering**
Capable of skipping beam positions if no target detected on last scan	Not at present (1989)	**Yes, via DCU interface and electronic steering**
Comp-controllable scan time	Not at present (1989)	**Yes**
Variable transmit peak power	No (TWT)	No (TWT)
Variable transmit average power	**Yes, (limited by tolerance of TX to varying load)**	**Yes, (limited by tolerance of TX to varying load)**
All range processing	**Yes**	Has buffer/mux that limits range extent. Can be achieved by stepping range window on consecutive scans.
360 degree azimuth coverage	**Yes**	**Yes, although up to 90 degrees electronic steering recommended for demonstration**
Elevation coverage	**Fixed cosecant squared**	2 degree beam with limited elevation mech scan capability limits chances of seeing targets of opportunity
Capability for frequency variability	Limited to about 30 MHz by PTS sythesizer. Changeable from beam dwell to beam dwell with sufficient dead time for synthesizer lockup	**Approx 500 MHz changeable from beam dwell to beam dwell with less than 15 μsec synthesizer lockup time (antenna limits # freqs to 8)**

Table 3: Tradeoffs of Existing RLSF L-Band and C-Band Equipment

Objective	L-Band	C-Band
Range resolution	Up to 20 MHz instantaneous bandwidth	Up to 10 MHz instantaneous bandwidth
Capable of mainlobe antenna nulling	No	**Yes, using subarray outputs with SPARC/DBF equipment (* equipment not then operational)**
Capable of sidelobe antenna nulling	**Yes, (Hazeltine adaptive loops)**	**Yes, via SPARC/DBF equipment using subarray outputs (jammers must be within beamwidth of subarrays)***
Changeable PRF, number of pulses, and pulse duration from burst to burst	**Yes**	**Yes****
Changeable doppler filtering	**Yes, (Signal processing in PTF)**	**Yes, (Sig proc in PTF)**
Track capability for up to 10 targets per scan	Only if new track-while-scan mods are implemented	Only if new track-while-scan mods are implemented
Sidelobe blanker	No	**Yes, (weather radar blanker in receiver string)*****
Real-time operation	Dead time must be allowed between beam positions to update equipment parameters and perform processing.	Same as L-Band
Noncoherent integration	**Yes, via PPI persistence**	**Yes, via PPI persistence**
Variable bandwidth	**Yes, (dwell to dwell)**	**Yes, (dwell to dwell)**

* Some hardware modifications required.

** It was subsequently determined that PRF and pulse duration changes precluded real-time operation with available equipment and risked transmitter damage.

*** Subsequently, a new blanking circuit was developed and the weather blanker was returned to its original state.

Table 3: Tradeoffs of Existing RLSF L-Band and C-Band Equipment (Cont'd)

3.2 Signal Processing

3.2.1 Requirements

The AI testbed requires two types of real-time signal processing

1. ESP of special predetermined waveforms to assess the jamming and clutter interference background.
2. DSP for extracting and displaying target reports in the raw data.

This functional distinction does not necessarily imply that separate equipment is required for each; indeed, one of the major tradeoffs was to decide whether available throughput capabilities and interfaces were sufficient for real-time operation using a common processor (e.g., the ST-100) for both. A single processor could exploit commonality of requirements via sharing functions, resulting in an overall simplification of the processing. However, this advantage is outweighed if that processor has insufficient throughput capacity for the required processing in real-time.

Therefore, tradeoffs were necessary to balance functional capabilities with implementation complexity and risk. Table 4 summarizes the functional requirements for each processor, as well as the final configuration which evolved from these tradeoffs. Descriptions of the rationale for each table parameter, as well as the individual tradeoffs, are given subsequently.

A synopsis of the resulting design, illustrated in Figure 2-3, is as follows:

1. ESP is performed by the ST-100, using its capability for spectral analysis of radar data. A pre-transmit passive dwell is used for sensing non-responsive jammers. The environmental assessment measurements for clutter are based on calculating the doppler filter bank output for the coherent active dwell that precedes the detection waveform.
2. DSP is performed primarily by existing real-time hardware to offload the ST-100 of these functions. Pulse compression is implemented by existing RLSF SAW compressors. Clutter filtering is accomplished by programming the programmable transversal filter (PTF) processor as an N-pulse adaptive MTI with weights adaptively determined for cancelling the clutter spectrum. We initially planned to use the SPARC-DBF equipment as a real-time sidelobe canceller. More detail pertaining to this design and its evolution follows.

3.2.2 Environmental Signal Processing

3.2.2.1 Design

	ESP	DSP
Waveform Type	Consistent waveform tailored to clutter and jammer measurement	Variable waveform tailored adaptively for target detection
PRF:	>1 KHz preferred*	> 1 KHz preferred*
Pulse Duration:	Nominally 16 μsec	Nominally 16 μsec
Coding:	LFM	LFM
Bandwidth:	$\geq$1 MHz	> 1 MHz
Dwell Time:	CPI1: 16 PRIs CPI2: 16 PRIs	CPI3: variable dwell
Processing Required	Pulse compression that matches DSP Doppler filtering to measure range-doppler characteristics of clutter Envelope and square law detection for magnitude and power estimation of interference Threshold detection to ascertain presence of interference Spectral analysis of single PRIs for ASP and CW jammer bandwidth determination Noncoherent integration of magnitudes of consecutive range cells to smooth measurements Estimation of whether jammer is in antenna sidelobes or mainlobe	Pulse compression for increased radar range and clutter performance Adaptive MTI or doppler filter bank for weather and groundclutter visibility Envelope detector Polarization adaptivity CFAR and threshold PPI display SLC† Sidelobe blankers Noncoherent integration
Implemented on	ST-100 array processor, SAW pulse compressor, dedicated SLB	Dedicated digital and analog GFE hardware

* 333 HZ implemented in testbed.

† Adaptive control implemented, but equipment not available.

Table 4: Signal Processors Functional Requirements and Design

Environmental assessment requires temporal and spectral measurements of the signal in a passive listening interval to identify jamming. It also requires measurements of the range-doppler characteristics of the radar return to identify ground and weather clutter. A pre-transmit passive dwell, CPI1, is used for sensing non-responsive jammers; an active environmental assessment dwell, CPI2, is used for measuring clutter characteristics.

Table 5 gives an overview of the waveform for each dwell. ESP is performed during CPIs 1 and 2, DSP during CPI3. Figures 3-2 and 3-3 show the processing for CPIs 1 and 2 in greater detail.

<——One Beam Dwell——>		
Environmental Assessment Interval		**Detection Interval**
CPI1	**CPI2**	**CPI3**
Passive (Rx only)	Active (Rx and Tx)	Detection Waveform
Assess ECM without clutter or targets	Assess clutter with consistent waveform	Use adapted waveform determined from previous scan
SLC (SPARC) calculates adaptive weights*	SLC turned so as not to corrupt clutter measurement*	All necessary ECCM enabled
Assess SLB % blanking	PPI is blanked	Routed to signal processor and PPI display
SLC doesn't apply weights to data*		

* Growth capability not implemented due to SLC unavailability

Table 5: Waveform Timeline

For CPI1, a power computation is made of the received signal and sent to the Adaptive Controller, where a decision is made as to the presence of a jammer. The computation is made by segmenting the receive signal over eight pulse repetition intervals (PRIs) into block integrations and reporting the output of each block integrator. The Adaptive Controller then thresholds these voltages to ascertain the presence or absence of a jammer signal. Further identification of the jammer is done by calculating the peak-to-average ratio of this vector, as is discussed in Section 3.3.2.3. The bandwidth of the jammer is estimated via a fast Fourier transform (FFT) of the contiguous range cells of the PRI with the highest duty factor discriminating narrowband from wideband jammers. For this application, the former type of jammer occupies a small percentage of the receive bandwidth; the latter occupies at least the full receive bandwidth.[2]

For CPI2, measurements of the range-doppler response are made from an active environmental assessment waveform which is constant from scan to scan. Block integrators for each of 16 doppler filters average the received magnitude over 32 contiguous range cells within 30 multiple contiguous range segments. These measurements are reported to the Adaptive Controller, where they are thresholded to detect and to determine range and doppler boundaries for ground and weather clutter.

Second order statistics are computed for estimating interference power levels and as an indicator of the degree of variation (variability ratio) of the measured values over the block intervals. The variability ratio, computed as the ratio of the mean squared to mean value squared, provides a discriminant which can be used by the AI rulebase to detect interference with Rayleigh-like magnitude statistics (such as weather, noise jamming, and Rayleigh ground clutter). Even though the variability ratio is not presently used, it is calculated and can be incorporated into future rules.

3.2.2.2 Design Evolution

The choice of waveform and environmental assessment processing was influenced by the following:

Real-time operation was one of the greatest challenges of this program. This was due to the volume of processing required and the fact that not all existing equipment (e.g., the VAX) is especially suited for real-time operation. The real-time requirement had a major impact on the design decisions, driving the design toward a multiple-processor architecture.

Maximum isolation of measurements was necessary between jammer and clutter sources to prevent complications in discriminating overlapping interference. Jammers are assessed with the transmitter off during CPI1 to prevent clutter and targets from corrupting the measurements. However, the clutter assessment interval (CPI2) contains overlapping jammer interference, hampering clutter identification unless the jammers are cancelled. As a result, clutter is identified after ECCMs have been applied to a jammer.

A consistent waveform with parameters tailored to clutter measurements is preferable to using the variable-parameter detection waveform (CPI3) for clutter measurement.

[2]Ideally, this measurement is made over as large a receive bandwidth as possible, and preferably over the entire radar agile bandwidth for a true wideband jammer assessment. For the IMRD demo, the measurement bandwidth was constrained to the receiver instantaneous bandwidth so as to remain within RLSF capabilities and to preclude excessive cost for implementing an additional wideband processing channel. Nonetheless, the bandwidth is sufficient for proof of concept.

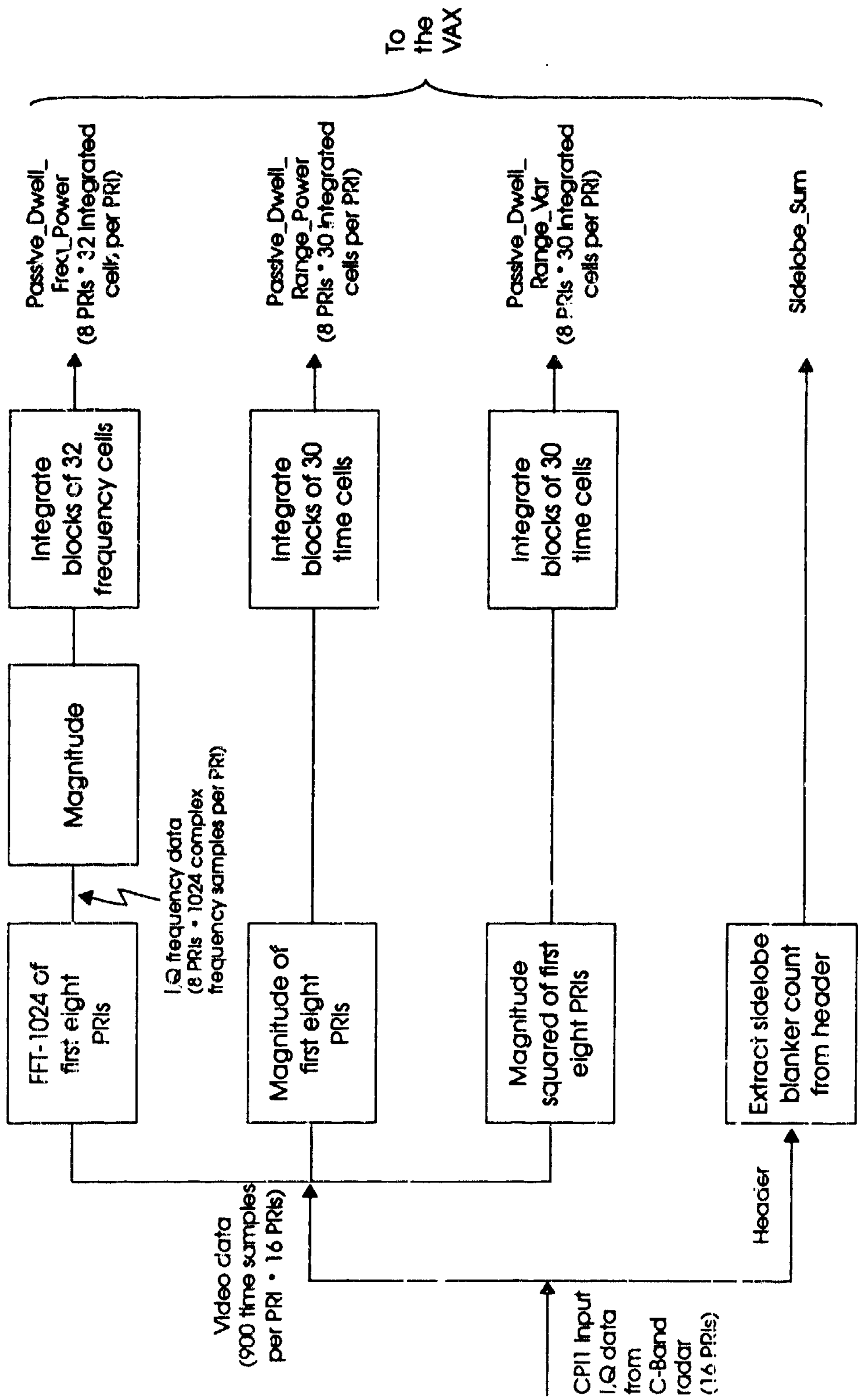

Figure 3-2: CPI1 Processing

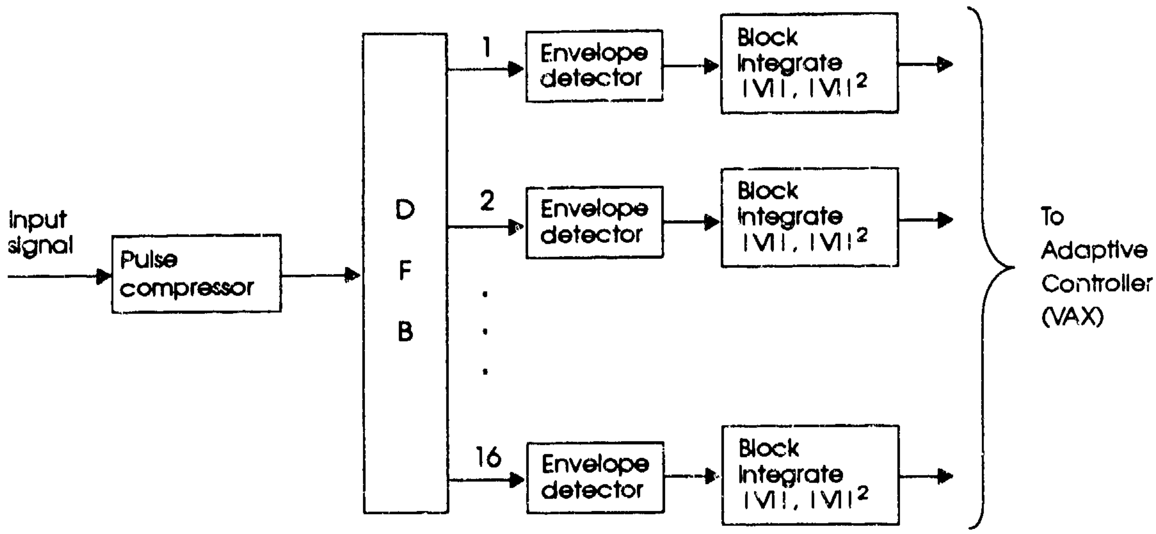

Figure 3-3: CPI2 Processing

Depending on the Adaptive Controller decisions, the latter will not necessarily be proper for measuring clutter. For example, in an extreme case, if the AI controller decides to skip a beam, the detection waveform may not be transmitted at all, even though a clutter measurement in that beam may still be desired. CPI2 waveforms with PRFs greater than 1 KHz are necessary to prevent spectral foldover of measurements of the weather clutter spectral spread.[3] Also, to enable sufficient accuracy for clutter bandwidth estimations, the coherent processing interval for the doppler filter bank was chosen to be at least 16 μsec (about 2 m/sec doppler resolution at C-Band). This is not necessarily guaranteed for CPI3.

The jammer detection interval must be long enough to detect low PRF asynchronous pulsed (ASP) jammers. For this reason, eight radar PRIs were dedicated for detecting the jammer pulses, with a timeline which supports growth to 16 PRIs. In retrospect, 16 PRIs was probably overkill since radar timeline is impacted by being unable to transmit in this interval. Finer tuning of the CPI1 dwell time and incorporating scan-to-scan history are possible improvements.

Pulse compression is necessary for clutter measurements to provide similar sensitivity to the detection waveform (CPI3) which likewise uses pulse compression. It is also necessary for achieving sufficient clutter-to-noise (CNR) ratio for clutter measurement.

Averaging measurements over contiguous time samples reduces the data that must be passed to the Adaptive Controller. This reduces I/O time, and enables a more

[3]However, a subtlety in the PTF MTI equipment—which was discovered during the integration and test phase—constrained the operating PRF for the demo to be 333 Hz maximum.

accurate determination of power level for each range-doppler sample.

Processing tradeoffs for detecting weather were a major consideration. The chosen approach involves doppler filter bank processing to derive range-doppler characteristics of the clutter and discriminants based on expected locations of the weather in the range-doppler plane. As an added discriminant–measurement of the clutter power difference between two polarization channels, one matched to the transmit polarization, the other orthogonal to it–would offer improved performance. However, equipment limitations made the polarization ratio measurement impractical.

GFE capabilities were evaluated for all design candidates. These tended to constrain variables such as the maximum bandwidth, maximum PRF, pulse coding for efficient real-time hardware pulse compression, and the amount of processing that could be implemented in real-time.

3.2.2.3 Equipment Tradeoffs

Tradeoff studies were conducted to determine the best equipment for meeting the environmental assessment requirements discussed in Section 3.2.2.2. We considered using existing digital processors within the RLSF, analog pulse compression, a hardware sidelobe jammer duty factor estimator, and programmable signal processing using either the AP-120, ST-100, HP2117, or VAX computers.

Doppler Filter Bank: The requirement for a complete doppler filter bank implied a programmable signal processor since none of the existing real-time signal processing equipment could form 16 simultaneous doppler filters in real-time. The ST-100 was the clear choice of the programmable signal processor candidates because of its superior throughput rate relative to the others and an available doppler filter applications program. Moreover, the other functions (shown in Figures 3-2 and 3-3) were well within the ST-100 capabilities and many had applications programs in development at the time of this decision.

Pulse Compression: The two candidate solutions for pulse compression were to implement a two-transform pulse compressor with the ST-100 software, and to use the existing SAW compressor.

To maximize the testbed's capability for real-time operation, we offloaded this function from the ST-100 to dedicated hardware. This enabled more parallel processing, albeit with some sacrifice of waveform flexibility.[4]

Bandwidth and PRF: As a result of the decision to use the ST-100, our next con-

[4]The RLSF SAW pulse compressor is optimized for LFM waveforms with 2.5 MHz bandwidth and 40 μsec duration, although it can compress other LFM waveforms with the same frequency/time slope.

sideration pertained to the maximum sample rate (i.e., waveform bandwidth). Due to I/O limitations of the digital hardware interface (DHI), the maximum complex word transfer rate into the ST-100 is 1 MHz; to process higher bandwidth signals, an existing FIFO (first in, first out) can be used to range window the receive data, slowing it down for transfer into the processor. The FIFO approach precludes "all range" processing and incurs additional complexity. Therefore, we decided that 1 MHz bandwidth waveforms were sufficient for demonstrating proof of concept, and that the FIFO was not necessary.

The product of the sample rate and PRI gives the number of samples that needs to be processed in the unambiguous range interval. Available ST-100 application programs had been successfully demonstrated within the RLSF with up to 1K sample PRIs; from a risk management standpoint, this factor favored the choice of a PRI of about 1 msec (i.e., 1 KHz PRF). The weather measurement function favors a higher PRF, although a 1 KHz PRF was deemed marginally sufficient for preventing spectral foldover for a worst-case weather spectral spread at a 5.7 m/sec-km wind shear.

Sidelobe Jammer Determination: One of the Adaptive Controller functions is to determine whether a jammer is in the radar mainlobe or in the sidelobes. For mainlobe jammers, traditional sidelobe cancellers (SLC) and sidelobe blankers (SLBs) offer no improvement. Therefore, when confronted with mainlobe interference, and in the absence of a mainlobe canceller, the AI rulebase puts the radar in a burnthrough mode of operation. If the jammer is declared to be in the sidelobes, a multiple sidelobe canceller (MSLC) or SLB may be enabled for high or low jammer duty factors, respectively.

Three approaches were considered for supporting the mainlobe versus the sidelobe assessment:

1. Compare the jammer powers measured over all beam positions. Mainlobe jamming is declared for those beams that have substantially higher (at least the mainbeam to peak sidelobe ratio) jammer power than the other beam positions.
2. Compare, in the ST-100, the jammer power in the sum beam channel with that in an auxiliary antenna channel with gain slightly above the sum beam's sidelobe level. Sidelobe jamming is declared in those time samples in which the auxiliary channel power is greater than the sum beam (similar in concept to a conventional sidelobe blanker).
3. Implement a dedicated hardware circuit that performs the comparison described in option (2), counts the number of sidelobe jammer declarations per PRI, and outputs this count to the ST-100 for determination of the sidelobe jammer duty factor.

Option (1) is the least expensive of the three options; however, it is vulnerable to false declarations in the case of more than one ASP jammer with diverse power levels. The other two approaches perform better in this situation because assessments can be made instantaneously on each time sample.

Option (2) has the disadvantage of requiring two radar channels to be interfaced and processed as opposed to *one* for the other approaches. Due to the 1 MHz maximum data transfer rate into the ST-100, a two-channel interface would require either halving the radar bandwidth to .5 MHz and time-multiplexing the ST-100 input port, or using the FIFO of C-Band buffer/mux to limit range processing to half the PRI extent. Also, new ST-100 software would be needed to demultiplex the input data.

Option (3) was preferred because of synergism with the DSP. Given that sidelobe blanker hardware was already required for detection processing, only minor additional circuitry (counter and interface to the header generator) was necessary to effect a capability of both sidelobe blanking and sidelobe jammer duty factor estimation. Figure 3-4 illustrates the new circuitry that was implemented for these combined functions.

Isolation of Measurements: In cases where multiple interference sources overlap in the measurement time interval, identification of the individual component interference sources proved to be extremely difficult, if not impossible in some cases. For example, a strong continuous wideband mainlobe jammer will mask the entire range-doppler surface computed during CPI2, precluding correct clutter declarations.

Eliminating targets and clutter from the jammer measurements is much easier than eliminating jamming from the clutter measurements. Clutter and targets are eliminated from the jammer measurement by keeping the transmitter off during the CPI1 measurement interval. Unfortunately, jammer signals are also present during CPI2 and tend to corrupt the clutter measurements.

Sidelobe jammer levels during CPI2 can be reduced—if not eliminated—by using a sidelobe canceller for continuous jammers, or sidelobe blanking for pulsed jamming. In the case of mainlobe jamming, a mainlobe canceller would reduce the jamming interference and mainbeam gain on the target would be maintained, provided that there is no more than one jammer in the null-to-null mainbeam width. For the demo, neither SLCs nor MLCs were available; we were, therefore, faced with the problem of NB or WB CW jamming overlapped with ground and weather clutter returns. Some headway was made on a rulebase for resolving the individual interference sources for this situation, but it was not implemented. We suggest additional analysis of this problem.

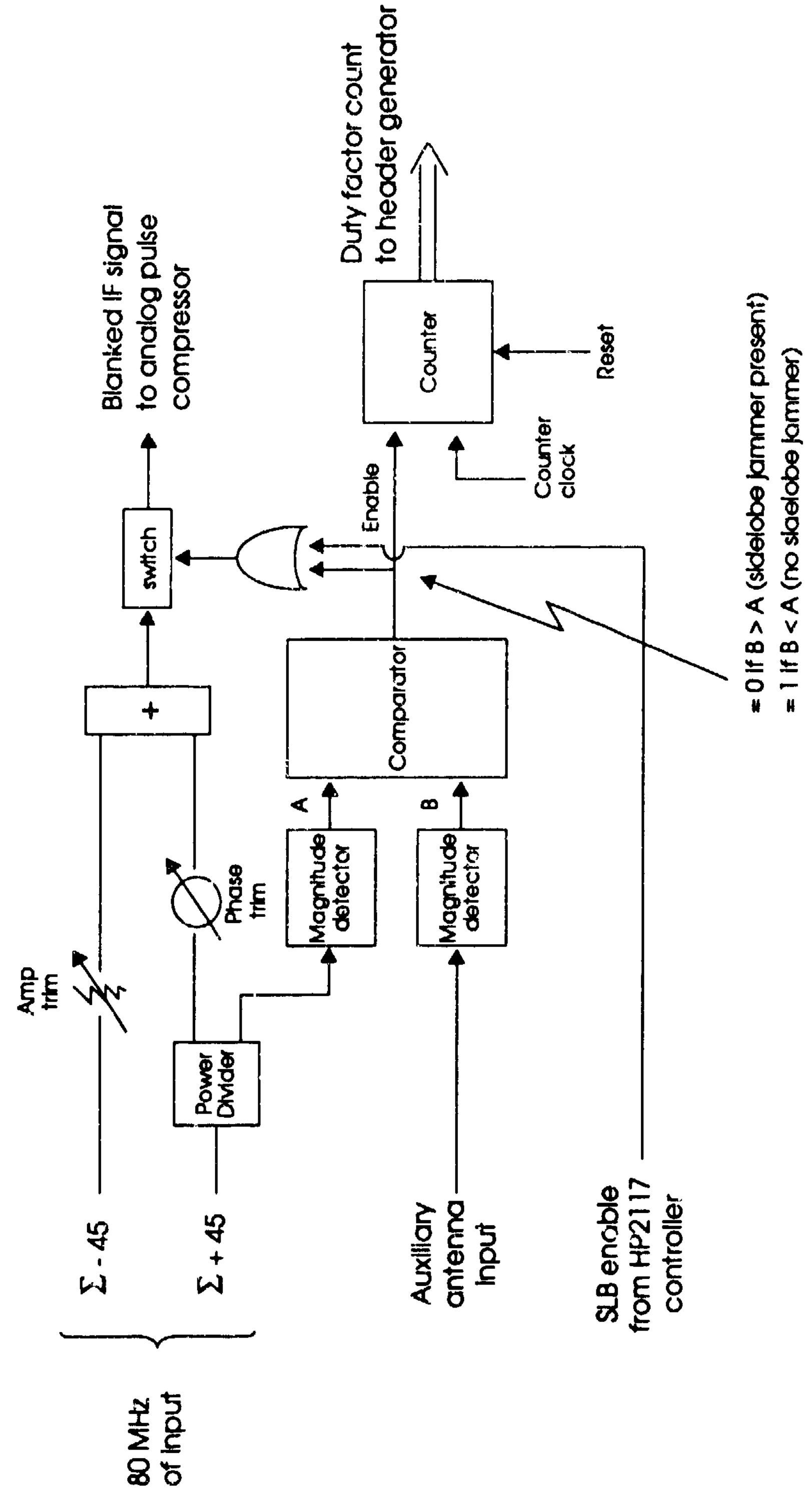

Figure 3-4: Sidelobe Blanker/Sidelobe Jammer Detection Circuit

3.2.3 Detection Signal Processing

The function of DSP is to extract and display targets from the radar return during CPI3 (see Section 3.2.2 and the functional requirements in Table 4). We evaluated alternate RLSF architectures with respect to their capabilities for satisfying goals 1–3 (listed at the beginning of Section 3).

The DSP is implemented with real-time dedicated RLSF hardware. Figure 2-3 illustrates the DSP equipment. Some of the major issues and tradeoffs in the DSP design were

1. Should the real-time signal processing functions (i.e., pulse compression, MTI, CFAR, and envelope detection) be performed using the ST-100, or could they be performed via a less-risky approach using the existing RLSF signal processing hardware?
2. Although the transmit polarization can be controlled by the DCU to be vertical, horizontal, left-circular, or right-circular, the receive polarization antenna outputs are only ±45°, and thus don't match any of the transmit polarizations. Therefore, these receive polarizations must be formed by the appropriate weighted addition of the existing +45 and -45° output channels. Should dedicated analog hardware, the ST-100, or the Mini-DBF equipment be used to apply the appropriate phase weights and to add these channels.
3. What is the best way to perform the adaptive antenna sidelobe cancellation?
4. Which interface (digital muxed data versus 80 MHz IF cabling) should be used to transfer received data from the antenna to the processing equipment?

Each of these issues will be discussed in greater detail.

3.2.3.1 ST-100 Versus Dedicated Hardware Tradeoffs

Table 4 suggests that the ESP and DSP functional requirements are sufficiently different to have a significant impact on loading if DSP were also to be implemented on the ST-100. We estimated that the ST-100 is at least 50 percent loaded and possibly more, depending on the efficiency of the applications code. Therefore, we believed that performing DSP on the ST-100 would substantially exceed the ST-100's capabilities for meeting the real-time requirement. Dedicated hardware was, therefore, used for the DSP.

Pulse Compressor and Data Interface: This tradeoff involved using one of two candidate architectures: 1) one or more existing SAW pulse compressors, or 2) the ST-100 with an existing applications program. The latter approach had the advantage of increased flexibility (it could accommodate more variations of waveform coding), but had the disadvantage of increased processing time.

Based on all considerations, we deemed the SAW compressor preferable. It is not as flexible for compressing arbitrary waveforms as the ST-100, but performs reasonably well with any LFM waveform that has a frequency slope of .0625 MHz/μsec. For the IMRD demo, the pulse duration and PRF were fixed for risk management, making operation at lower than maximum bandwidth unimportant. Therefore, the waveform was fixed at 1 MHz bandwidth and 16 μsec duration to give the requisite frequency slope.

The following three additional sub-options were considered for the SAW pulse compressor implementation:

1. Use two PCs (pulse compressors): one for the sum+45, and the other for the switched output of either the sum-45 or subarray auxiliary channel. The PCs would be installed ahead of the existing synchronous detectors in the C-Band receive string, the synchronous detector digital outputs would be cabled to the SPARC SLC, and the SLC digital output would be cabled to the RLSF equipment room over a 300 ft. digital interface.
2. Use one PC in series with the ROM Waveform Generator (WFG) connected to the output of the SPARC DBF. The PC output would then be transferred (at IF) over the 300 ft. coaxial cable to the RLSF, connected to the existing 80 MHz synchronous detector, and digitized by A/D converters. The digital data would then be fed into the A/D converter data input of the mux board.
3. Use one PC at the receiving end of a 300 ft. coaxial cable (within the RLSF) that carries an 80 MHz IF signal from the C-Band room.

Of the three options, the first two could be used in conjunction with the SPARC SLC; the third was applicable only if the SPARC equipment was not used. Between (1) and (2), the former was preferred; the latter would only be used in the event that the requisite 300 ft. digital interface was not implemented in time for the IMRD demo. Option (2) had the disadvantage of introducing two additional error sources—namely, the ROM Waveform Generator and a second synchronous detector—into the data string.

Initially in the contract, we estimated that the SPARC SLC would be operational in time for the IMRD demo, and equipment design for option (1) proceeded under that assumption. However, we later determined that technical problems with the SPARC SLC would preclude its use. At that point, option (3) became viable since the architecture no longer required that digital data be available within the C-Band room for input to the SLC. As a result, the final AI testbed incorporated option (3).

For all options, the pulse compressor is installed ahead of a synchronous detector which has about 63 dB of dynamic range (12 bits I and Q). Considering the dynamic range increase through the PC of 12 dB (10log(bandwidth * time)), this restricted the dynamic range input to the PC to about 51dB. Throughout the equipment tests and final demo, adjustments were made via variable attenuators to prevent signals from exceeding this dynamic range window.

Doppler Processing: Tradeoffs between adaptive MTI and the "doppler filter bank+greatest of" implementation favored the former option because the effective spectral notch is equivalent with much less complexity. The loss of integration gain with MTI relative to a full doppler filter bank had minimal impact on demonstration credibility since the C-Band system has the sensitivity to detect targets beyond 100 nmi, provided that pulse compression is used.

The adaptive MTI was implemented using the PTF lab equipment, with the weight sets calculated by the VAX computer using an existing algorithm developed by Sensis. The PTF accepts input data at up to a 2 MHz rate; the number of pulses is up to 16.

The weights are precomputed for different spectral regions and stored in a table. A weight set is selected, based on environmental assessment outputs that specify the center frequency and bandwidth of ground and weather clutter. The wider the clutter bandwidth relative to the radar PRF, the more degrees of freedom (i.e., processed pulses) are used to implement a spectral notch encompassing the clutter. Narrow spectral widths, such as those for discrete stationary scatterers, require only a two-pulse MTI to cancel to desired levels.

A requirement, stated in the manufacturer's users manual, for 2048 range samples per PRI was misinterpreted to mean "up to 2048 range cells" instead of the manufacturer's intention that "greater than 2048 range cells" be processed per PRI. During equipment integration and test, we discovered that the baseline design—which had a PRI of 1 μsec and a sample rate of 1 MHz—had an insufficient number of samples per PRI for the PTF to operate. An increase in the PRI (i.e., decrease in the PRF) was therefore necessary, making the system less than optimal for weather spectral measurements as well as for weather cancellation.

Envelope Detector: This function is implemented using the existing RLSF equipment in cabinet E-2. The sample rate through the envelope detector is limited to 2 MHz and the dynamic range is 10 bits, both more than sufficient for demonstrating feasibility.

CFAR/Log Normalizer: The CFAR/Log Normalizer implements the function illustrated in Figure 3-5. The RLSF normalizer uses the geometric mean of the background to normalize the test cell. The number of cells comprising the background estimate

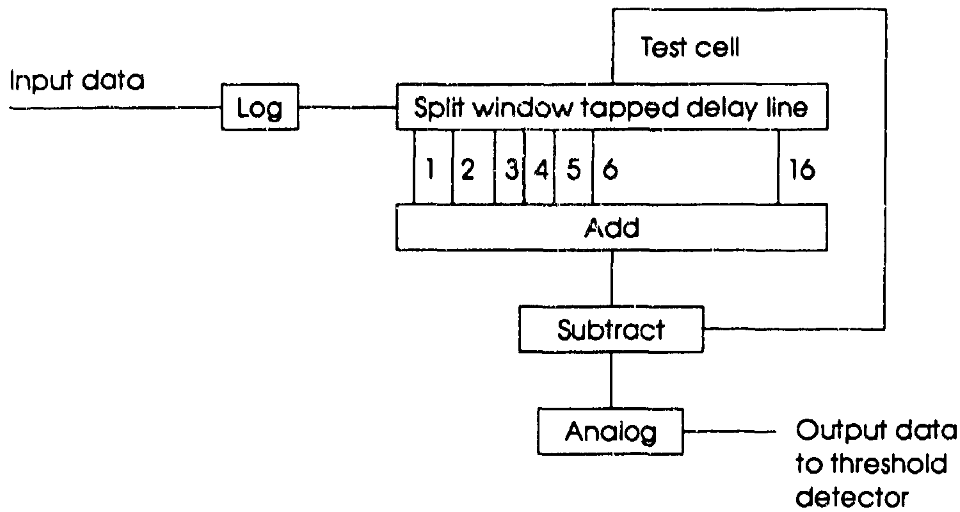

Figure 3-5: CFAR Equipment Algorithm

can be either 32, 64, or 128; of these options, the 128 was used since the other two exhibited excessive digital glitch outputs.

CFAR was not used for the final demo to enable a better PPI display indication of false alarms with and without AI enabled.

Threshold Detector and Noncoherent Integration: Threshold detection and noncoherent integration are built into the UPA-62 PPI display, so no additional circuitry was necessary for implementing these functions. Signals below approximately .5 V input are not displayed, depending on the sensitivity adjustment. Persistence of the PPI phosphors improves target detectability with increased dwell time (therefore, more target hits) and approximates a noncoherent integration function. The advantages of reduced complexity using this technique far outweigh the slight loss as compared with an ideal noncoherent integrator.

3.2.3.2 Receive Polarization

The options for implementing a receive polarization that is either matched or orthogonal to the transmitter involved the following weighted combinations of the +45 and -45° sum channel outputs:

1. Add the +45 and -45° channels to yield horizontal polarization.
2. Subtract the -45° polarization channel from the +45° channel to give vertical polarization.

3. Apply 90° phase shift to +45 channel and add to -45 channel to give left circular polarization.
4. Apply 90° phase shift to -45 channel and add to +45 channel to give right circular receive polarization.

These calculations could be performed in one of three ways: 1) in analog hardware at the 80 MHz IF (via a power splitter, phase shifter, power combiner, and switch), 2) in ST-100 software, or 3) in the SPARC-SLC digital equipment.

The SPARC-SLC option would have involved a minor equipment modification to enable applying weights to and summing two receive channels (+45 and -45°) instead of the eight channels for which it is designed. This was the preferred approach for synthesizing the appropriate polarizations, but the equipment was not operational at the time of the demonstration.

The ST-100 option would have required interfacing the two polarization channels to the ST-100 input as well as writing additional applications code. For these reasons, as well as to offload the ST-100 for maximizing real-time operation, this approach was eliminated.

Even though the analog network requires precise line length, amplitude, and phase matching, it was chosen for the final equipment configuration. We experienced some difficulty in adjusting the phase and time delay matching of the input channels, but eventually realized over 35 dB cancellation of an orthogonally-polarized wideband test signal as well as approximately 20 dB cancellation of real weather returns with an orthogonal transmit polarization.

3.2.3.3 Sidelobe Cancellation

We considered two options for sidelobe cancellation: 1) use the SPARC-SLC Mini-DBF, or 2) implement this function in the ST-100. We chose the former option to reduce ST-100 functional throughput requirements and to enhance the capability for real-time operation. Other reasons which favored the SPARC-SLC were that the ST-100 would have required a new software input interface for demuxing two input channels and that substantial additional applications code would have been necessary to implement the SLC.

The SPARC processor in the Mini-DBF would have been used in an operating mode which implements a single degree of freedom sidelobe canceller. This would have required that the SPARC-SLC Mini-DBF equipment receive digital inputs for the sum and a single auxiliary beam. The auxiliary would have been one of the subarray channels of the array.

3.2.3.4 Data Interface

Paragraph 2 in Section 3.2.3.1 discusses the three candidate data interface approaches. Each is related to different implementations of the pulse compressor and depends on operational SPARC-SLC equipment.

The initial design involved using the SPARC-DBF with a digital output interface to the RLSF, 300 ft. away. The output interface was to consist of a Hot Rod interface card at each end of the cable. The card at the SPARC-DBF output end of the cable would have taken parallel word inputs, converted them to a serial bit stream, then sent the serial data over a 300 ft. coaxial cable interface to the PLSF. On the receiving end of the cable, another Hot Rod card would convert the serial data stream back to parallel for input to the PTF and ST-100.

The need for a 300 ft. *digital* interface was driven by the SPARC-DBF, which has a digital input/output and is installed in the C-Band room. An alternate interface (see Section 3.2.3.1, paragraph 2) was a fallback in the event that the Hot Rod was unavailable. This option would have used the ROM Waveform Generator at the SPARC-DBF output instead of the Hot Rod transmitter to modulate the signal on an 80 MHz IF for transfer to the RLSF where the signal would have been converted back to digital via the existing 80 MHz synchronous detector.

Since the SPARC-DBF was unavailable, neither of the interface options discussed above were used. Instead, the IF signal from the array was transferred directly to the RLSF over a 300 ft. coaxial cable to the RLSF synchronous detector, thus bypassing the synchronous detectors in the C-Band room. Figure 2-3 illustrates this interface.

3.3 Adaptive Control

The IMRD objective is to identify targets in the presence of clutter and jammer interference. Adaptive control requirements in support of this objective are:

1. Receive raw radar data on a beam-by-beam basis.
2. Identify factors that impact system performance.
3. Determine what, if any, changes can be made in operating conditions so as to improve performance.
4. Select an operating plan for the next scan that optimizes system performance.
5. Send control parameters for the next scan to the radar.

Section 3.3.1 discusses the initial tradeoffs made in the Adaptive Controller design. Sections 3.3.2 through 3.3.4 detail the resulting implementation.

3.3.1 Tradeoffs

The Adaptive Controller design required a series of studies to determine the following:

1. A means of assessing the effectiveness of an adaptive control policy (Section 3.3.1.1).
2. Categories of interference that impact target detection (Section 3.3.1.2). Particular emphasis was placed on those that could be identified in and compensated for in the RLSF environment, and on techniques that could be applied to mitigate the identified interference.
3. Policy control decisions that should be made by the user; for example, the ability to prioritize regions of coverage and to constrain system resources such as scan time (Section 3.3.1.3).
4. Which programming languages to use (Section 3.3.1.4).

3.3.1.1 Evaluation Criteria

Evaluating the impact of an adaptive control policy on subsequent radar performance is necessary for two reasons. First, the AI component of the Adaptive Controller must be able to select among alternate operating plans *before* one is implemented. Second, a measure of adaptive control effectiveness *after* a plan is implemented is needed to demonstrate the accomplishments of this contract.

A priori evaluation is required by the AI software when generating a plan for the next scan. Alternate operating plans will assign different amounts of dwell time to the individual beams, with better target detection associated with higher levels of dwell time per beam. For purposes of policy selection within the adaptive control software, we used a computed probability of detection estimate as a criterion for selecting between two choices and as a measure of control effectiveness.

The probability of detection calculation assumes a conventional Bayesian likelihood ratio test comparing the received signal with a threshold normalized to the interference background. The target is assumed to be slowly fluctuating with Rayleigh statistics (i.e. Swerling I) and the false alarm probability is maintained at 10^{-6} per range cell via a CFAR threshold. Other false alarm rates or target models could be accommodated with minor changes to the software.

In addition to probability of false alarm and target statistics, the detection probability is a function of the net signal-to-interference ratio (SIR), the interference statistics, and the number of looks that are noncoherently integrated by the signal processor. To simplify the AI testbed, the spatial distribution of clutter is assumed to be Rayleigh, and the other variables are accounted for via a table lookup.

The SIR estimate is based on actual measurements of the interference-to-noise ratio and the calculated SNR for a user-specified cross section and detection range. The SNR calculation includes other parameters of the radar range equation–such as average power, antenna gain, and losses–which are stored in the Radar Design Parameters file for the specific radar.

A posteriori evaluation of adaptive control effectiveness is more difficult. Given knowledge of the true radar environment, one could compare the number of detected targets and false alarms with AI enabled, and evaluate the same scene without AI enabled. On a single scan basis, it is unlikely one will know what the real environment is and whether an identified target is real or is a false alarm; hence, this approach is not practical in a real-time demonstration of target detection. However, it was effectively used to verify identification and countering interference such as jammers that could be controlled from the RLSF.

3.3.1.2 Interference

Section 3.1.2 summarizes the L-Band/C-Band tradeoffs and establishes what adaptive controls can be achieved with the C-Band radar. Given the choice of C-Band radar, we next considered what types of interference can be identified that impact system performance, which adaptive controls can be applied to them, and what can be demonstrated within the RLSF environment.

Types of Interference: Two types of interference appear in the radar signal: that which can be detected by listening (which shall be denoted passive interference) and that which can be detected in the returns of a transmitted waveform (which shall be denoted active interference). The first category includes various types of nonresponsive jammers and radio frequency interference (RFI); the second, ground and weather clutter, chaff, and responsive jammers. This distinction led to including a passive listening interval and an active assessment interval in each beam dwell prior to the detection waveform. See Section 3.2.2 for a full description of the beam dwell partition.

Nonresponsive jammers are sensed during the passive dwell. They will show up as threshold crossings in the power level for consecutive PRIs. They can be discriminated by narrowband versus wideband, CW versus asynchronous pulsed, and mainlobe versus sidelobe.

Measures of ground and weather clutter are made during the active dwell via a Doppler filter bank. Ground clutter shows up in radar data as threshold crossings near zero doppler and range out to the radar horizon (except in anomolous propagation conditions). Weather clutter shows up as threshold crossings at ranges corresponding to its location and at doppler corresponding to weather radial velocity. If the weather is near the transmitter with a velocity vector tangential to the beam, it is expected to lie in the same range-doppler cells as ground clutter. If jammers are present, they will mask both ground and weather clutter; hence, ground and weather clutter will not be visible unless appropriate adaptive controls have been applied during CPI2 to detected jammers.

We have maintained a growth path to other categories of interference such as chaff and responsive jamming.

Adaptive Controls: The following adaptive controls can be applied within the RLSF environment:

1. Narrowband jammers: frequency change, growth to sidelobe canceller.
2. Pulsed sidelobe jammers: sidelobe blanking.
3. CW sidelobe jammers: sidelobe cancelling.
4. Ground clutter: adjustable MTI weights.
5. Weather clutter: receive and transmit polarization, adjustable MTI weights.
6. All: dwell time per beam.

Development and Demonstration: Existing RLSF equipment includes two fixed C-Band jammer sources, which can simulate pulsed and CW narrowband and wideband jammers. The jammers can be used to investigate system time and performance response to jammer interference. They are remotely controlled from the HP2117 terminal in the RLSF equipment.

Ground and weather clutter provide interference to be sensed in the active dwell. Within the RLSF environment, ground clutter extends out to about twenty miles from the transmitter. Its intensity can be increased by aiming the antenna down or decreased by aiming it up. Weather clutter control is limited for testing and demonstration.

Targets of opportunity and simulated targets were illuminated in the clear for system verification and in competition with the various clutter and jamming sources for demonstration. Targets on final approach to GAFB, Syracuse Hancock, and Oneida County Airports were of interest since they appear in the ground clutter.

We concluded that these categories of interference were sufficiently representative and testable to exercise our rulebase development methodology and to permit a meaningful demonstration. Our procedure for developing and testing the rulebase was:

1. Develop and test rules for identifying individual sources of interference and applying appropriate ECCM.
2. Repeat step (1) with multiple sources of interference.
3. Develop rules for allocating dwell time and evaluate them with simulated targets.

Recorded radar returns allowed rulebase development for interference identification without repeatedly needing to turn on the radar and transmitter. This played an important role in step (1) which required successive iterations to fine-tune the rules. With respect to live testing, weather needed to be tested as opportunities arose, but the other sources of interference could be tested with live data at any time.

Thus, we obtained representative classes of interference for which adaptive controls existed within the RLSF and which were available for development and demonstration.

3.3.1.3 User Policy Control

The IMRD allows user inputs consistent with tactical objectives. For example, a piece of equipment such as a sidelobe blanker may be unavailable, track-while-scan data rate requirements may dictate a scan time requirement, or the operator may wish to factor in the relative importance of regions of coverage. These capabilities are built-in through user-selectable AI Control Parameters, Quality Options, and Priority Zones.

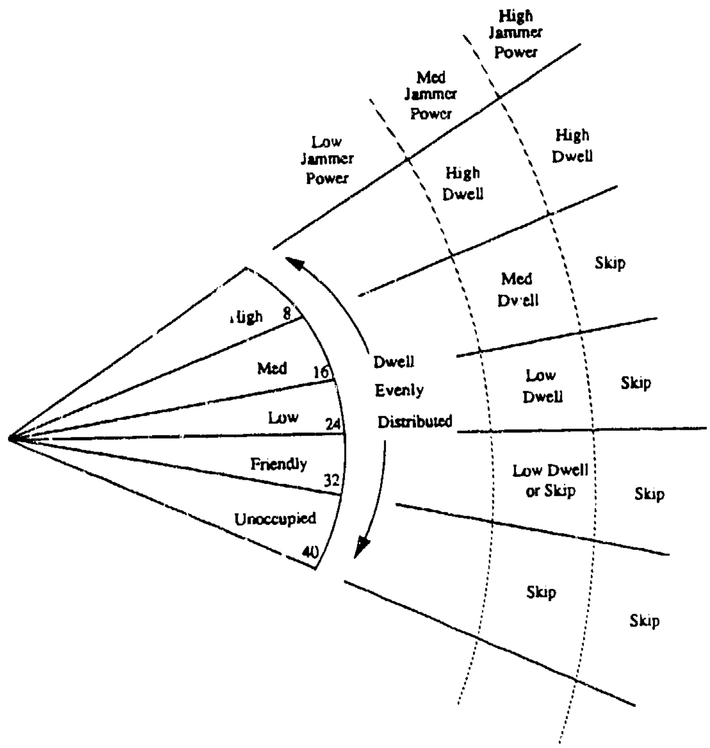

Figure 3-6: Priority Zone Example

Additionally, the user can control the AI Display through AI Output Parameters. The rulebase will then tailor its decisions to accommodate these inputs.

AI Control Parameters: The user can assign values (such as desired scan time, whether AI is enabled, and which adaptive controls can be applied) to AI Control Parameters. They are either read from a file or entered interactively on-line at system initialization, and may be modified during operation either on-line or by reading a new file. This data structure is described in Appendix E.2.

Priority Zones: Regions under radar surveillance can be partitioned into Priority Zones based on their expected threat level and the importance of maintaining detection within them. Five types of threat regions (high, medium, low, friendly, and unoccupied) are specified by near and far range and by start and stop azimuth.

Figure 3-6 illustrates a Priority Zone configuration that was used to demonstrate differential dwell time allocations under various types of interference. In this example, the first beams of the scan region are of highest priority, with regions of decreasing importance in a clockwise rotation.

A beam may contain more than one priority level with, for example, a higher priority region near the transmitter and a lower one further out. At present, for purposes of adaptive control, the entire beam is considered to be at the highest level it contains.

Priority Zones may be entered from a file or edited on-line. A default level of unoccupied is associated with unassigned zones. This data structure is described in Appendix C.2.

Quality Options: Beam dwell time is adaptively apportioned to different beams by the Adaptive Controller based on a set of alternate plans stored in a Quality Options Table. A quality plan assigns a probability of detection to each priority level. The Quality Options Table contains up to eight plans, ordered by decreasing probabilities of detection in each of the priority zones. The Adaptive Controller selects the first plan that can execute within the desired scan time.

A typical set is given in Table 6. In plan (1), all regions are assigned the highest Pd. If the scan time required to meet the Pd of .9 in each region exceeds the user-specified upper limit, a plan will be selected in which less time is used for lower-level threat regions relative to the higher threat regions. In plan (8), for example, dwell time is assigned to the high and medium threat regions and others are skipped.

If there is insufficient time to satisfy any of the first eight plans, the Adaptive Controller chooses a default plan (9). This default may be required when there is strong interference background and/or when the system is stressed by a low desired radar cross section (RCS) or a large desired detection range. In this case, dwell time is assigned to the high threat zones so as to obtain the highest probability of detection possible within the total scan time constraint.

Note that even if two beams are at the same threat level, they may not be assigned the same dwell time under a plan. This can happen in the case that there is interference in one beam, requiring more time to reach the same Pd as another beam in which there is no interference.

Quality Options also include passive and active update rates (not shown in Table 6) to indicate how frequently the passive and active dwells for each beam should be repeated. During IMRD development and testing, we assumed that update occurs on each scan to meet the SOW requirement for a one scan response time, but it might make sense to update beams at lower priority levels or regions in which no interference is identified less frequently.

Also, it should be noted that a variable other than Pd might be appropriate for the table entries in some other applications. For example, the entries could be detection range for a fixed Pd (say .9) and a fixed target RCS, or could be RCS for fixed Pd and detection range.

The Quality Options Table is read from a file or entered interactively at system initialization. It may be modified during operation either on-line or by reading in a

Priority Level	Overall Quality Option								
	1	2	3	4	5	6	7	8	9
High threat	.9	.9	.9	.9	.9	.7	.7	.5	*
Medium threat	.9	.9	.9	.7	.5	.5	.4	.3	0
Low threat	.9	.9	.7	.5	.5	.3	0	0	0
Friendly	.9	.7	.5	.5	.3	0	0	0	0
Unoccupied	.9	.6	.5	.3	0	0	0	0	0

* As high as possible within scan time constraint.

Table 6: Sample Quality Options Table

new file. Full descriptions of Quality Options Tables and sample input are given in Appendix D.3.

AI Output Parameters: These variables define the types of information and destinations of outputs the user may display or record during a session. AI Output Parameters are read from a file or entered interactively at system initialization and may be modified during operation either on-line or by reading a new file. This data structure is described in Appendix F.2.

3.3.1.4 Choice of Languages

The IMRD SOW stated that, unless it could be justified that some other choice was more appropriate, Prolog was to be used for AI (the expert system) and either Pascal or FORTRAN could be used for other components. With this in mind, we selected the language that was most appropriate for each component.

We used Prolog as a starting point for language decisions because it was the basis for the prior AI simulation and because of its reputation as an AI language. The advantages of Prolog are:

1. We could use code developed under the previous contract.
2. Prolog predicates are a convenient way to express expert system rules.
3. Prolog's built-in inferencing capabilities eliminate the need to code an inference engine.
4. Prolog is considerably more readable than LISP (the other logical choice for AI language), making it especially appropriate for a project in which the rulebase

was expected to evolve.

5. The foreign language interface capability in Quintus Prolog (the available compiler) supports integration of routines written in other languages.
6. Quintus Prolog has acceptable run-time performance.

The disadvantages of Prolog are:

1. Prolog has poor facilities for data structuring. Values for a single variable and for simple lists can be asserted into the Prolog database, but the ability to declare variables and to organize information into a record structure is missing. Since one IMRD goal was to support transfer to other hardware environments through parametrization, this was a serious shortcoming.
2. Prolog has poor facilities for procedural abstraction. There is no distinction between local and global variables. This makes information hiding difficult, and means that more care must be taken to modularize a program.
3. Prolog cannot be used for direct interaction with the VMS operating system and, hence, is inappropriate for mailbox utilities, process spawning, the ST-100 Host process, and the VAX/HP2117 interface.

The final solution was a combination of Prolog, Pascal, and FORTRAN. Table 7 summarizes the uses for each. We found the Prolog-Pascal-FORTRAN combination effective, using the capabilities of each language well.

3.3.2 Knowledge Engineering

Knowledge engineering consisted of a precise definition of inputs derived from the radar signals and outputs to the Radar Controller, subproblem decomposition of the Adaptive Controller (expert system), and rulebase evolution.

3.3.2.1 Expert System I/O

Defining expert system data inputs and control parameter outputs was an iterative process with the corresponding parameters evolving as our understanding of hardware and software capabilities broadened. The inputs are organized in a Pascal data structure called Beam Scan Data. The outputs are organized in a Pascal data structure called Radar Control Parameters.

Beam Scan Data Inputs: We defined Beam Scan Data to correspond to signal processing capabilities and adaptive control objectives. One Beam Scan Data record for each beam is derived from ST-100 inputs, 16 PRIs of passive dwell listening data, 16 PRIs of active dwell data, and a measurement of the sidelobe jammer duty factor. It contains:

1. **Passive_Dwell_Range_Power:** An array whose components contain the power level returned from the passive listening dwell for consecutive averaged range blocks in each of eight PRIs. Only the first eight of the 16 CPI1 PRIs are currently used. See Figure 3-2.
2. **Passive_Dwell_Freq_Power:** An array whose components contain the FFT of the complex voltages of all range cells for each PRI. See Figure 3-2.
3. **Active_Dwell_Power:** An array whose components represent the doppler filter bank calculated over the 16 PRIs of the active listening dwell. See Figure 3-3.
4. **Sidelobe_Sum:** An array representing the number of time samples in which sidelobe jamming was sensed in the passive listening dwell for each PRI.

The following are computed and allow for future growth, but are not used in the present rulebase. The variability can be used as a further discriminant among multiple sources of interference.

1. **Passive_Dwell_Range_Var:** An array whose components contain the variabilty of the power level returned from the passive listening dwell for consecutive range cells in each of eight PRIs.
2. **Active_Dwell_Var:** An array whose components contain the square of the doppler filter bank calculated over the 16 PRIs of the active listening dwell.

The scan information is received from the ST-100, then forwarded to the Adaptive Controller from a VMS mailbox. This data structure is described more fully in Appendix G.2.

Radar Control Parameter Outputs: We defined Radar Control Parameters based on our determination of potentially-controllable parameters during the tradeoff studies and on their expected impact on subsequent radar performance. This data is sent by the Adaptive Controller to the VMS mailbox from which it is forwarded by the interface to the HP2117 computer. It contains fields for the following information:

1. **Trans_Center_Freq:** The frequency to use for the beam position. The software can control up to 16 unique frequencies; however, the IMRD system uses only eight. This index is translated by the HP2117 to a real frequency value.
2. **ECCMs_On:** Whether a given ECCM should be on. The components are Sidelobe_Blanking, Sidelobe_Cancelling, and CFAR.

3. **PRI:** The pulse repetition interval to use for the beam position. It can assume values between 1 and 4096 microseconds. Within the IMRD, the value is set to 3000.
4. **Pulse_Duration:** The pulse duration for the beam position. It can assume values between 0 and 40 microseconds. Within the IMRD, the value is set to 16.
5. **Pulse_Bandwidth:** The pulse bandwidth for the beam position. It can assume values between 0 and 10 MHz. Within the IMRD, the value is set to 1.
6. **TX_Polarization:** The antenna polarization to use for transmitting in the beam position. Possible values include Horizontal, Vertical, Left_Circular, and Right_Circular. Within the IMRD, the value is Horizontal or Vertical.
7. **RX_Polarization:** The antenna polarization to use for receiving in the beam position. Possible values include Horizontal, Vertical, Left_Circular, and Right_Circular. Within the IMRD, the value is set to Horizontal.
8. **Adaptive_MTI_Weights:** One of 64 possible MTI weights sets. In the IMRD, 37 unique sets are used; the number 64 represents the default weight set.
9. **Dwell_Time_per_Look:** The PRI for CPI3. The values range from 1 to 1000 msec.
10. **Number_of_Looks:** The number of pulses to be used in CPI3. The values range from 0 (indicating that CPI3 should be skipped) to 10000.
11. **Passive_Dwell_On:** Whether CPI1 should be included. Within the IMRD, the value is set to true.
12. **Active_Dwell_On:** Whether CPI2 should be included. Within the IMRD, the value is set to true.

As indicated in their descriptions, not all fields are controlled under this hardware implementation—they are included to allow different radar control configurations. This data structure is described more fully in Appendix H.2.

3.3.2.2 AI Subproblem Decomposition

Figure 2-5 shows a breakdown of the subproblems that contribute to assigning Radar Control Parameters for each scan: ID Passive Interference, ID Active Interference, Determine Applicable ECCM, and Allocate Radar Resources. The following sections summarize these rulebases. Their complete specification is given in Appendix 1.6.3.

3.3.2.2.1 ID Passive Interference

This subproblem identifies jammers in a beam from the passive dwell, CPI1 (using Beam Scan Data variables Passive_Dwell_Range_Power, Passive_Dwell_Freq_Power, and Sidelobe_Sum described in Section 3.3.2.1). Figure 3-7 summarizes the steps and discriminants used.

1. The first step is to determine whether any jammers are present by examining the Passive_Dwell_Range_Power (the power level in consecutive range cells for each PRI). If any of its elements exceeds a threshold, a jammer of some type is assumed to exist.

2. The rulebase then determines whether a CW-type jammer is present by calculating the peak-to-average power ratio for the Passive_Dwell_Range_Power. This is calculated for each PRI; the largest value (Range_Var) is used to determine the jammer type. A small Range_Var indicates a CW jammer. A large Range_Var indicates that the jammer is pulsed or of some other unknown type.

3. If the jammer is CW, Freq_Var,the peak-to-average power ratio for the Passive_Dwell_Freq_Power, (the FFT of the chosen PRI) is calculated using PRI 1. A low Freq_Var indicates wideband jammer, a high value indicates narrowband.

   The Sidelobe_Duty_Factor is calculated by comparing the Sidelobe_Sum values (the number of time samples where the auxiliary voltage is greater than the sum voltage) from each PRI and selecting the one that occurs most frequently. If Sidelobe_Duty_Factor exceeds a threshold, a sidelobe jammer is declared; otherwise, a mainlobe jammer is declared.

4. Determining whether a jammer is pulsed or of some unknown type is more involved than the CW determination. A mainlobe duty factor estimate is calculated for each PRI in order to determine whether the jammer is pulsed (duty factor less than or equal to ten percent) or of some other unknown type. Experimentation with thresholding techniques indicated that an adaptive threshold gives the best results due to variability in the thermal noise level; we decided to use an adaptive threshold that was midway between the peak and average value of the passive range vector for a particular PRI. The PRI with the largest duty factor is chosen to determine the bandwidth of the interference. This requirement was placed on the software since the jammer returns may be eclipsed during some of the PRIs due to dead time in some interval. If more than one PRI qualifies, the PRI with the pulse closest to the center of the range vector is used. This last qualification was added because we found that the Freq_Var calculation was more accurate if the pulse was in the center of the processed data due to weighting employed in the time dimension. This was especially true of low-power jammers. At this point, the specific type of jammer is determined

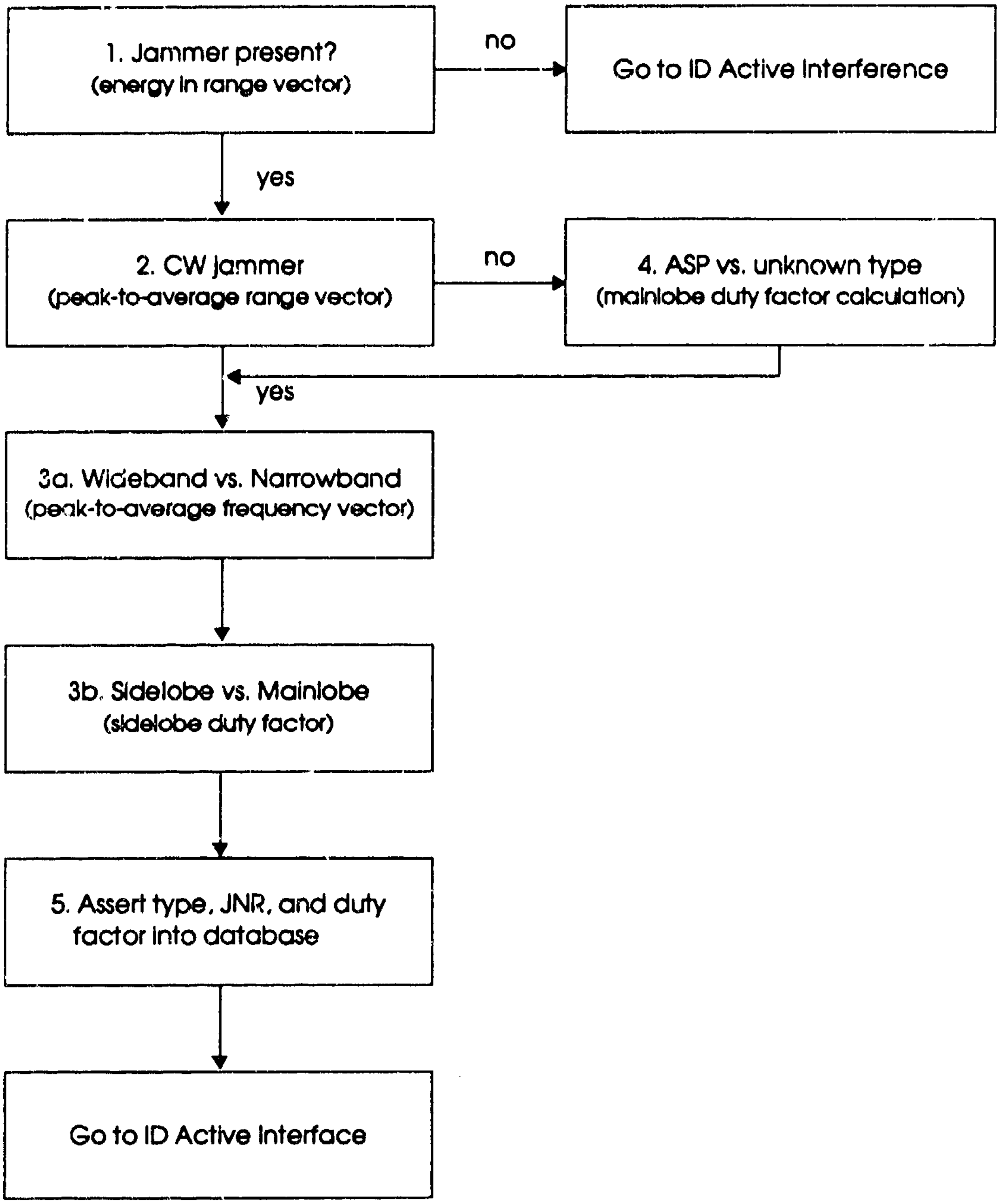

Figure 3-7: ID Passive Interference Flow

from the Mainlobe_Duty_Factor, the Freq_Var, and the Sidelobe_Duty_Factor, as described above.

5. Jammers are asserted into the Prolog database as jammer_found facts with variables describing the specific type of jammer found, the peak jammer-to-noise ratio, and the duty factor.

3.3.2.2.2 ID Active Interference

This subproblem determines which types of clutter are present from the Active_Dwell_Power. The input data on which decisions are based is the response of 16 doppler filters over a 900 microsecond range interval during CPI2. For the current rulebase discrimination of the each of the types of interference is based on recognizing patterns that emerge on this range-doppler surface. Examples of patterns are given in Figure 3-8 which indicates range-doppler regions in which interference would likely occur for the various environments.

The ID Active Interference rules then find the best match of the measures range-doppler response with prestored maps, each representing the expected range-doppler surface for a different clutter environment. The criteria for determining which map most closely resembles the measurement is based on the power calculation over the segmented zones and the pattern of threshold crossings as described below.

The various maps are boolean arrays with dimension equal to the input array (16 doppler rows x 30 range samples). These maps contain values of true or false for each range-doppler cell, depending on whether the particular type of clutter could reasonably occur.

Two types of interference maps are incorporated into the current rulebase: ground clutter and weather clutter. For this demonstration we used ground clutter maps with trues in the doppler filters that overlap the DC response out to the range horizon, but falses elsewhere. The weather map contains values of true or false for each range-doppler cell depending on whether weather clutter could reasonably occur within it.

This technique is a two dimensional extension of a concept developed under the previous Rome Laboratory AI work (ACMDSP, Contract F30602-86-C-0204), and represents a logical compromise between simplicity and performance for the initial software build. An early set of rules calculated the size and power of clusters of threshold crossings in the active dwell Beam Scan Data. Since these calculations required a lot of CPU time *and* were difficult to modify and expand, we used the predetermined boolean maps to indicate for which array elements within the active dwell power array we expect to see threshold crossings for a given type of interference. Now that real-time performance has been established, we recommend further rulebase

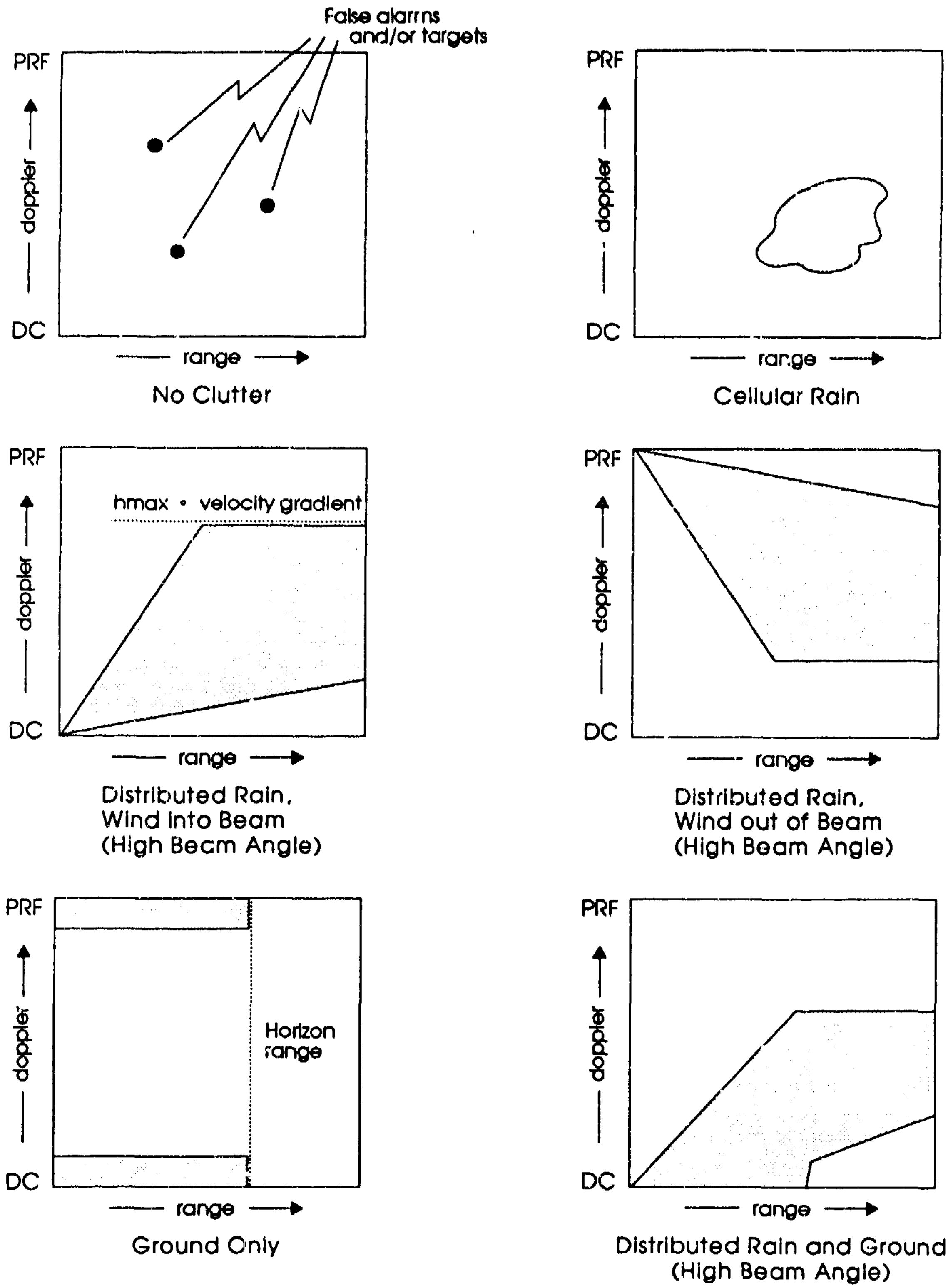

Figure 3-8: Interference Patterns in Range-Doppler Space

experimentation rulebase to enable more sophisticated identification.

Figure 3-9 summarizes the steps and discriminants used.

1. Several discriminants are calculated over these maps including the sum of the power over the map, the percentage of matching threshold crossings, and the percentage of non-matching threshold crossings. The active dwell thermal noise array multiplied by a constant is used to threshold the active dwell array.
2. The power calculation and the percentage of matching threshold crossings over the ground clutter map are made. Ground clutter is declared if the calculations over the ground clutter map are above specified thresholds.
3. The power calculation and the percentage of matching threshold crossings over the weather clutter map are made. Weather clutter is declared if the calculations over the weather map are above specified thresholds.
4. If ground clutter is found, a ground_clutter fact is asserted into the Prolog database with variables describing the peak clutter-to-noise ratio, the maximum range of the ground clutter, and the clutter-to-noise ratio at the maximum range of the clutter. If weather is detected, a weather fact is asserted into the database with variables describing the peak clutter-to-noise ratio, the maximum range of the weather, the clutter-to-noise ratio at the maximum range, and the doppler extent of the weather.

3.3.2.2.3 Determine Applicable ECCM

This subproblem determines radar control changes and their expected impact on detection probability for each jammer and clutter identification. The inputs are the outputs of the ID Passive Interference and ID Active Interference subproblems, as well as the list of available ECCM selected by radar and user constraints.

Jammer ECCM, including frequency hopping, sidelobe cancelling, and sidelobe blanking for narrowband, CW sidelobe, and asynchronous pulsed jammers, respectively, are enabled where appropriate. Jammer-to-noise ratios are adjusted, corresponding to the impact of the adaptive controls.

Clutter ECCM, including MTI weights for ground and weather clutter, and transmit polarization for weather clutter, are enabled as appropriate. Clutter-to-noise ratios are adjusted, corresponding to the impact of the adaptive controls.

Determine Applicable ECCM selects the controls in the order indicated in Table 8. Those controls that have the least impact on scan time and the least SNR loss are applied first. For each control applied, a new estimate of the signal-to-interference ratio is based on user-supplied nominal cancellation values for that control, parame-

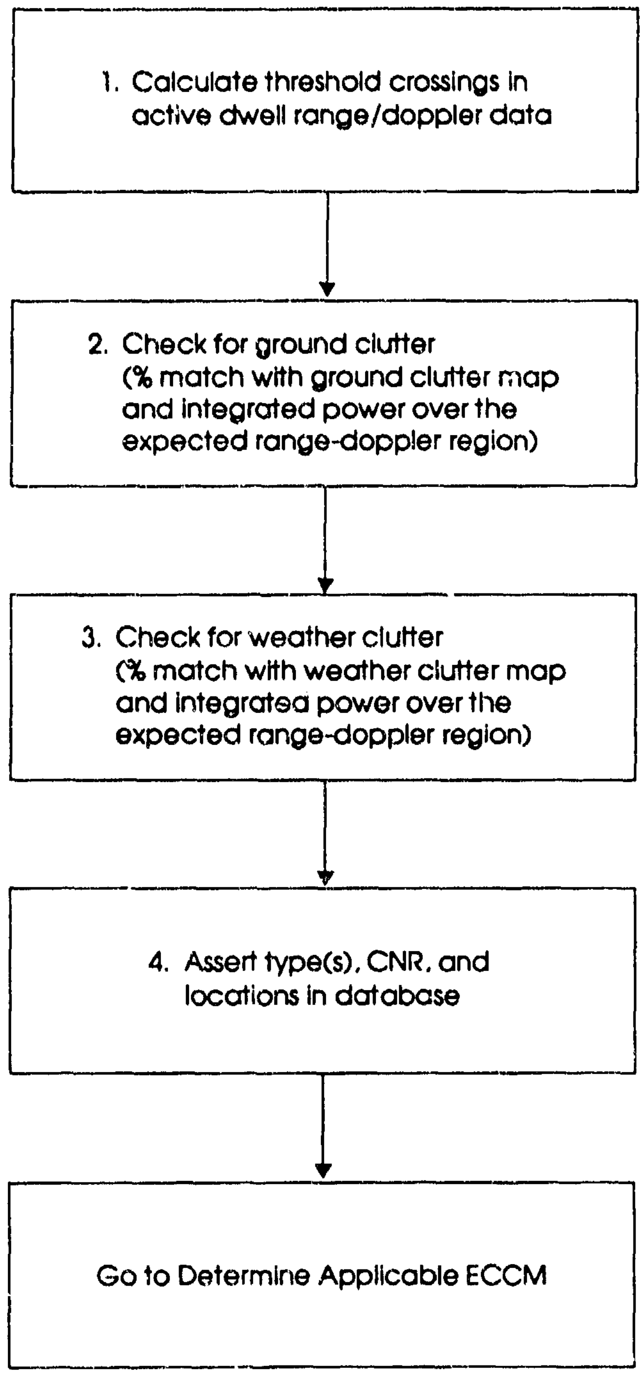

Figure 3-9: ID Active Interference Flow

ters of the radar range equation (to estimate SNR for the desired target cross section and range), and on JNR and CNR measurements made during CPI1 and CPI2.

The numbers of looks required to reach probabilities of detection .1 through .9 are calculated for each beam. If the signal-to-interference ratio in a beam after higher priority controls are applied is still insufficient for achieving the desired Pd (probability of detection), the last step is to increase the energy in the beam via a noncoherent integration over a longer dwell time, with a direct impact on scan time.

The benefit gained by dwell-time increase with noncoherent integration depends on the target fluctuation characteristics, the SIR per pulse, and the degree of decorrelation of interference (relative to spectral bandwidth). These effects are accounted for in the AI rulebase via a table look-up of Pd versus number of looks and SIR for a Swerling I target. The spectral width of the interference is estimated to determine the effective decorrelation time. For a given estimated SIR, the product of the required number of independent looks to achieve a given Pd with the decorrelation time gives the total CPI3 dwell time.

The outputs for each beam are a table of adaptive controls and the numbers of looks.

3.3.2.2.4 Allocate Radar Resources

This final step produces the plan for the next scan. The inputs to this subproblem are the outputs from Determine Applicable ECCM for each beam, the Priority Zones, and the Quality Options Tables. The outputs are Radar Control Parameters.

The total scan time required to achieve the probabilities of detection assigned to the different priority zones under each quality plan is computed, starting with the first plan and continuing until a plan is reached that does not exceed the desired scan time. This is done by adding (for each beam) interbeam dead time, total dwell time for CPI3 as calculated using the appropriate probability of detection, CPI1 time if it is to be performed, and CPI2 time if it is to be performed.

Once a plan has been selected, any remaining dwell time is allocated uniformly across all beams in order to use the remaining available time.

3.3.2.3 Rulebase Evolution

Rulebase development is an iterative process of formulating rules and evaluating their impact. The well-designed experiment will, therefore, provide a framework within which rules can evolve.

The IMRD differs from other expert systems work in that its expertise is a combination of existing human knowledge (available at the beginning of the contract) and detailed knowledge of how the system behaves (available as the contract progressed).

Thus, rulebase flexibility was critical to successful development.

Both the initial hardware/software design and the adaptive control software supported iterative development well, and enabled us to model adaptive control expertise successfully. Section I.7 summarizes our observations and findings during this process.

3.3.3 Software Development

3.3.3.1 Goals

IMRD software development goals included:

1. Reuse of ACMDSP design and code where practical.
2. A system that supports rulebase evolution.
3. Sufficient IMRD generality to support migration to other radar environments.
4. Software portability to platforms other than the VAX.
5. Stand-alone operation for demonstration and testing.

Reuse of existing work: The ACMDSP made good use of modular design principles and was well documented. As a result, its code was accessible for reuse. To capitalize on this, our initial design was based on the ACMDSP model in which an Adaptive Controller spawns the Radar Model as a subprocess and communicates with it via well-defined interfaces. This same approach was taken with the Adaptive Controller as main driver, spawning processes to handle VAX/ST-100 and VAX/HP2117 interfaces. The main control pattern—read and process data from each of the beams in a scan, assign control parameters for the next scan, and send control parameters for the next scan—was also maintained. As a result, many of the predicates within each of the major code blocks could be reused with little or no modification.

Although the major premises of the ACMDSP were revised, approximately ten percent of the final 10,000 lines of IMRD code was derived from existing software. Mailbox and process spawning code were directly reusable. Other portions served as a starting point and saved significant development time on our part.

Rulebase evolution: The IMRD schedule mandated that hardware and software be developed in parallel. However, characteristics of the interferences identified, equipment tolerances, and the impact of adaptive control applied were understood only after closed loop operation was achieved. Hence, much of the rulebase development was done after major portions of the software were written.

This type of development was supported by isolating expert system subproblems within separate predicates. The similarity between Prolog predicates and English-language rules further supports modification and expansion.

Other radar environments: Ideally, the IMRD framework and perhaps even the specific expertise are applicable to other hardware environments. In particular, the concept of a 3-CPI beam dwell and the Adaptive Control AI subproblem decomposition should transfer.

Additionally, Adaptive Controller data structures include radar-specific data (Radar Design Parameters), environment-specific data (Radar Environment Parameters), input from the radar to the Adaptive Controller (Beam Scan Data), and data output from the Adaptive Controller to the radar (Radar Control Parameters). These can be tailored to other software and radar environments. They also proved useful during system development since values for a number of the radar-specific constants were not known until closed loop operation had been attained.

Software portability: Software portability is critical to avoid restricting this work to a VAX host. VAX system services are used to spawn subprocesses, for interprocess communication, and for interfacing to the radar control software, but calls to system services are isolated within the spawned subprocesses and within specific Pascal procedures. The Prolog code could be used on another host operating system without change. The system service calls and spawned process code would need to be modified.

Stand-alone operation: Stand-alone operation of the IMRD software was necessary for both system development and rulebase evolution. The interface between the Adaptive Controller and radar is through VMS scan and control mailboxes. The FRDF can send recorded scan data to the scan mailbox and retrieve control parameters from a control mailbox was built for stand-alone operation. As a result, we could test the Adaptive Controller before hardware/software integration permitted closed loop operation of the entire system and rulebase development using taped data could proceed in parallel with hardware/software integration.

3.3.3.2 Implementation

Adaptive control software consists of a Main Adaptive Control process which performs system initialization and spawns the appropriate subprocesses to perform communication tasks. At initialization, the Main Adaptive Control process reads in configuration options, one of which tells the process whether the system should run in Radar Interface or in Stand-Alone configuration.

Radar Interface configuration: Under this option, inputs are received from the radar via the ST-100, and Radar Control Parameters are distributed to the radar via the HP2117. The three processes used under Radar Interface configuration are the Main process, the ST-100 Host process, and the Radar Controller Interface (RC_IF) process, as shown in Figure 3-10. Communication among the processes is through VMS mailboxes.

Stand-Alone configuration: The Adaptive Controller can be used in Stand-Alone configuration to test the system without requiring live radar inputs. The FRDE process is spawned to handle communications tasks with the Main process, as shown in Figure 3-11. Beam Scan Data sent to Main is retrieved from an ASCII file rather than from the ST-100. Radar Control Parameters are sent by the Main process to the FRDE rather than to the RC_IF process which would normally forward them to the HP2117. Both of these functions are accomplished via the same VMS mailboxes used in the Radar Interface configuration.

3.3.3.2.1 Inputs

Inputs to the Adaptive Controller are Beam Scan Data, which is received from the ST-100 signal processing software via the VAX/ST-100 interface. These inputs consist of one Pascal record for each beam in the surveillance region, and contain results after signal processing of passive and active radar dwells in the most recent scan.

Beam Scan Data is used by the subproblems, ID Passive Interference and ID Active Interference, to determine what, if any, interferences are present.

3.3.3.2.2 Process Descriptions

Main Adaptive Control: The Adaptive Controller consists of the following software modules, with interrelations as shown in Figure 2-4:

1. A **User Interface** allows the user to specify default input parameters or to enter the parameters interactively.
2. An **AI Engine** assesses the current environment and selects radar control options for the next scan.
3. An **AI Display** displays user-selected parameters on the screen and/or saves them in a file.

The AI Engine is further divided into the four subproblems discussed in Section 3.3.2.2: ID Passive Interference, ID Active Interference, Determine Applicable ECCM, and Allocate Radar Resources.

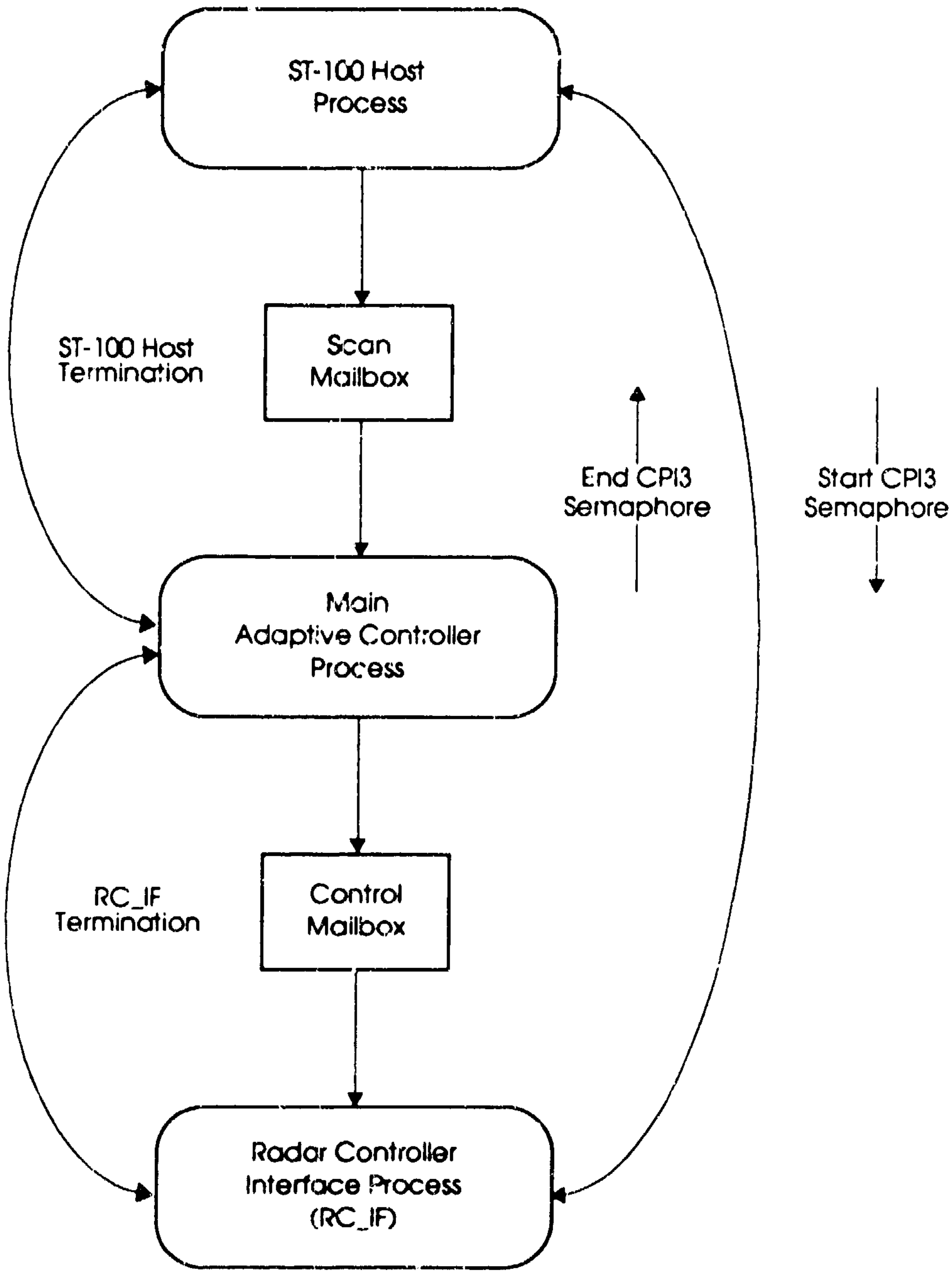

Figure 3-10: Radar Interface Software Organization

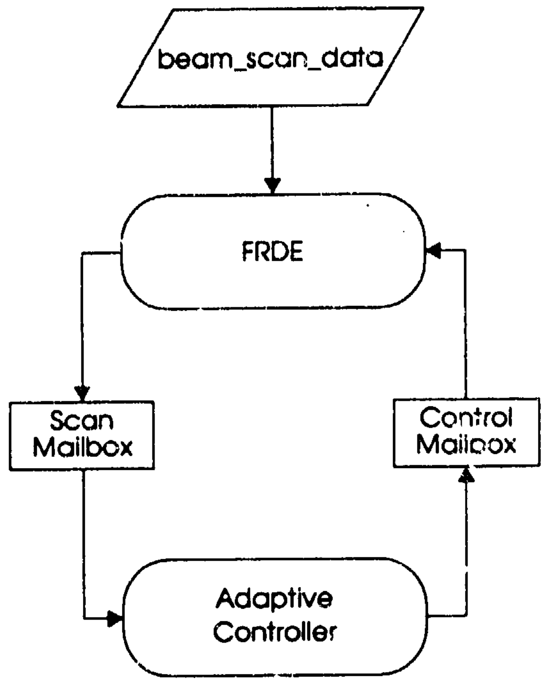

Figure 3-11: Stand-Alone Software Configuration

The Main process spawns the two subprocesses, as shown in Figure 3-10. The ST-100 Host process handles communications between the signal processing software residing on the ST-100 and the other VAX processes. The RC_IF process handles communcations between the HP2117 and the other VAX processes.

VMS mailboxes and common event flags are used for VAX interprocess communication. Common event flags are used for interprocess synchronization. A scan mailbox is used for communication between the Main process and the ST-100 Host process. A control mailbox is used for communication between the Main process and the RC_IF process. This allows the Main process to remain independent of the processes that interface to external equipment.

ST-100 Host process: The primary responsibility of the ST-100 Host process is to communicate with the ST-100 signal processing software. This includes downloading Array Processor Code Language (APCL) process code, passing synchronization semaphores from the adaptive control and radar control software, and uploading the calculated Beam Scan Data from the ST-100 to the VAX.

The ST-100 Host process downloads signal processing software to the ST-100 and performs ST-100 initialization routines, as described in the *Array Processor Executive User's Guide*. VMS common event flags are then used to synchronize the radar control and the signal processing software. RC_IF sets an event flag to signal completion of

CPI3. This information is forwarded to the signal processing software via ST-100 semaphores. Another ST-100 semaphore signals that the ST-100 is ready to receive CPI1 data. The ST-100 Host process then uses an event flag to tell the RC_IF process that the ST-100 is ready to start CPI1. In this way, each beam is synchronized.

Other ST-100 semaphores synchronize uploading the Beam Scan Data from the ST-100 as the data is calculated. Once a complete Beam Scan Data record is assembled, the process uploads the record to the Main process via the scan mailbox. This mailbox is also used at the end of each scan to inform the ST-100 Host process whether to terminate or to initiate another scan. Another event flag is set by the ST-100 Host process to tell the Main process that its termination procedures are complete.

Radar Controller Interface process: The RC_IF process controls communications between the HP2117 and the VAX. This consists of passing Radar Control Parameters and synchronization semaphores to the HP2117.

The primary RC_IF process responsibility is to receive Radar Control Parameters from the Main process via the control mailbox and forward them to the HP2117 Radar Controller. This mailbox is also used at the end of each scan to inform RC_IF whether to terminate or to initiate the next scan. Two event flags synchronize beam timing between the ST-100 and the Radar Controller. Two additional event flags are sent by RC_IF to Main to signal process initiation and process termination.

Flexible Radar Data Executive: The FRDE process takes the place of the ST-100 Host and RC_IF processes when the system executes in Stand-Alone configuration. The FRDE opens the file, beam_scan_data, and reads Beam Scan Data for one scan. This file has the same format as that produced when using the AI Output Parameter, diagnostic_output (beam_scan_data), to collect Beam Scan Data. The file can contain data for any number of beams. The Beam Scan Data is then sent on a beam-by-beam basis to the Main process via the scan mailbox. The FRDE also receives the Radar Control Parameters from the Main process via the control mailbox. Both of these functions are accomplished via the same VMS scan and control mailboxes used in the Radar Interface configuration. Both the ST-100 Host and the RC_IF termination messages are sent to the FRDE. The FRDE then sets both termination complete event flags to signal process termination.

3.3.3.2.3 Outputs

Outputs from the Adaptive Controller are sent to the HP2117 Radar Controller in the form of Radar Control Parameters. These consist of one Pascal record for each beam in the surveillance region, specifying radar operating characteristics for the next scan.

The Allocate Radar Resources subproblem produces the Radar Control Parameter outputs to be used for the next scan. Radar Control Parameters for the first scan are read from a file at system startup. Subsequent Radar Control Parameters are generated by the Adaptive Controller.

User-requested output information is sent to the user's terminal or to files, as specified by the AI Output Parameters.

3.3.4 Parameterization

The AI testbed can be configured to accommodate a variety of radar environments through Radar Design Parameters and Radar Environment Parameters. The first category contains parameters related to the specific radar, the second to the radar environment. This data is read from files during system initialization. The files can be modified by using any standard text editor.

Radar Design Parameters define the operating characteristics of the physical radar with which the Adaptive Controller interfaces. The parameters are read from files at system initialization and may be modified off-line by using a text editor. They include information such as azimuth limits and the pulse repetition interval.

Collecting these parameters in one data structure makes any assumptions about the radar explicit and visible. We have had occasion to change these parameters during operation and appreciated not being required to locate embedded constants or to recompile the system. For instance, the PRI was modified several times in order to experiment with performance improvement. This was easily accomplished by editing the edm_radar_design.dat file and setting the new PRI value. Appendix A.3 describes this data structure.

Radar Environment Parameters describe the thermal noise received by the radar as well as the ground and weather clutter maps to be used in the ID Active Interference subproblem. The parameters are read from files at system initialization and may be modified off-line by using a text editor. Appendix B.3 describes this data structure.

Language	Usage
Prolog	• The main driver, calling routines in Pascal as necessary. The foreign language interface assigns a link between a Prolog predicate and a Pascal procedure or function. When the predicate is invoked, control is transfered to the assigned routine • Rulebases. These are accessable for modification and evolution • Inferencing. Prolog backtracking made this function transparent • Most of the on-line user interface. Prolog was selected here so as to use portions of the Motorola code • Priority zone data structure. Prolog was selected here so as to use portions of the Motorola code
Pascal	• Global data structures which organize the types of information needed • Computation • Mailbox utilities through which the Adaptive Controller interfaces with the remainder of the IMRD system • Subprocess spawning and control
FORTRAN	• The RC.IF process which supports communication between the VAX and the HP2117F • The ST-100 host process which supports communication between the VAX and the ST-100

Table 7: Language Selection

Interference	MLC (Growth)	SLB	SLC (Growth)	Adaptive Polariz	MTI	Frequency Change	Adjust Dwell Time
NB CW mainlobe	2					1	3
NB CW sidelobe			2			1	3
WB CW mainlobe	1						2
WB CW sidelobe		1					2
NB ASP mainlobe						1	2
NB ASP sidelobe		2				1	2
WB ASP mainlobe							1
WB ASP sidelobe		1					2
NB unknown						1	2
WB unknown							1
Ground clutter					1		
Rain clutter				1	2		3

Table 8: Applicable Adaptive Controls and Order of Application

3.4 Radar Control

Radar control and timing synchronization is performed using a combination of HP2117 software and special-purpose hardware. The following sections summarize the function and implementation of IMRD control and timing functions.

3.4.1 Requirements

The Radar Controller performs the following functions:

1. Initialize all the radar hardware (i.e., antenna controller, Tx/Rx frequency, MTI weights, video blanker, sidelobe blanker, and 3-CPI circuit.
2. Establish a communications channel to the VAX via the IEEE-488 bus port.
3. Prior to each scan, receive inputs from the VAX AI processor specifying the parameters for that scan.
4. Verify that the desired Radar Control Parameters are consistent with RLSF equipment capabilities. If not, alert the operator of the inconsistency. Abort the process for that beam dwell if the inconsistency is fatal.
5. Encode the parameters from the AI processor into hardware commands that are recognizable by the RLSF hardware.
6. Send the encoded hardware commands to the RLSF equipment during the dead time interval prior to the beam dwell. During this time, the radar will be turned off.
7. Synchronize the radar operation by properly timing control signals sent to the real-time processing and control equipment.

3.4.2 Variables and Interfaces

Table 9 lists the inputs from the VAX to the Radar Controller via the IEEE-488 bus. One set of these controls is sent for each beam in the scan. The index is the order in which the components are sent over the interface. Several of the parameters passed for each beam are not currently set by the adaptive control software. Their function is to allow future growth of the control scheme.

Table 10 lists the semaphores that are sent from the HP2117 to control CPI1 and CPI2 processing in the ST-100. Table 11 lists the mechanisms which control the physical hardware.

Index	Parameter	Description
1	last_beam	Flag indicating last beam of scan (1 = last)
2	face_indx	Pedestal azimuth
3	beam_az_in	Beam azimuth offset in millidegrees relative to -45 degrees from pedestal azimuth
4	el_indx	Pedestal elevation index (not used)
5	freq	Frequency index (0–7)
6	s_blank	Sidelobe blanker flag (1 = blanking on)
7	s_cancel	Sidelobe canceller flag (1 = canceller on)
8	cfar	CFAR value (not used)
9	pri	PRI value (not used)
10	pulse_dur	Transmit pulse duration (not used)
11	pulse_bw	Transmit pulse bandwidth (not used)
12	tx_polar	Transmit polarization index
13	rx_polar	Receive polarization index (not used)
14	weights	MTI weight set index (1–64)
15	dwell_tim	CPI3 dwell period (in μsec)
16	num_looks	Number of CPI3 dwells
17	passive_on	Disables CPI1 (not used)
18	active_on	Disables CPI2 (not used)

Table 9: Radar Control Inputs

Value	Description
2	End of CPI2 processing
4	Start CPI1
998	Terminate processing
999	End of scan

Table 10: Control Semaphores

Description	Destination	Interface
Pedestal azimuth	C-Band DCU	DCU
Beam azimuth	C-Band DCU	DCU
Transmit polarization	C-Band DCU	DCU
Antenna frequency	C-Band DCU	DCU
C-Band LO frequency	C-Band frequency synthesizer	IEEE-488
MTI weights	PTF	Existing parallel
Sidelobe blanker enable	Sidelobe blanker	Static bit
Video blanker enable	Video blanker	Static bit
3-CPI enable	3-CPI circuit	Static bit

Table 11: Radar Control Outputs

3.4.3 Controller Design

The Radar Controller consists of a control function which is performed by software resident on the HP2117, and radar timing and synchronization which is performed in hardware by the 3-CPI circuit and the C-Band Range Tracker. The system block diagram (Figure 2-3) illustrates the control and timing functions, and the relationship to the radar and processing equipments. A description of the evolution of the design of the radar control functions and the ultimate implementation follows.

3.4.3.1 Software Design Evolution

The following considerations influenced the radar control software design.

Synchronization and communication of the processing elements and the radar system was perhaps the biggest consideration in designing the Radar Controller. Raw radar data must be provided to the array processor for the three different dwell periods (CPIs), Radar Control Parameters must be received from the VAX, and control must be provided for the radar by the controller.

Controlling the radar timing by generating strobes for the transmitter, receiver, data transfer, and display control functions was achievable by several methods. The evaluation factors in the selection of timing control were the ease of interface to the Radar Controller, the precision which was required for the timing signals, and the requirement of maximum scan time.

3.4.3.2 Radar Controller Platform Selection

Potential hosts for the radar control function included the VAX, a special-purpose hardware controller, an IBM-compatible personal computer, and the HP2117.

Real-time processing limitations and lack of existing interfaces to the radar system eliminated the VAX. A dedicated special-purpose hardware controller had the advantage of the most precise control timing, but was eliminated due to the lack of existing interfaces to any of the RLSF equipment and to the high cost involved in its design and construction. The personal computer approach was also discarded because there were no existing interfaces between it and the RLSF equipment. The HP2117 remained the logical candidate to host the radar control function.

The HP2117 has interfaces with much of the required RLSF equipment (e.g., C-Band antenna controller, C-Band LO, VAX, and PTF), yet was not well suited to timing synchronization control at the level required by the radar system. For the precisely-timed radar operation, the existing C-Band range tracker was used in conjunction with the 3-CPI circuit which was designed and built under this contract. Interfaces to the sidelobe blanker, video blanker, and the 3-CPI were added using static register bits from the HP2117. These bits are toggled during execution to enable or disable the blankers or the 3-CPI circuit.

3.4.3.3 VAX/HP2117 Communications and Synchronization

The major driver for selecting an interface between the VAX and the HP2117 controller was the quantity of data that must be transferred for each scan. The existing IEEE-488 bus was selected to transfer the Radar Control Parameters to the Radar Controller because it has a significant bandwidth advantage over a serial link. Synchronization messages ware also passed over the IEEE-488 bus to minimize the amount of interfaces present between subsystems and the amount of software necessary to maintain each interface.

Four messages (semaphores) are passed from the VAX to the HP2117 during the operation of the system, as described in Table 10. A graphic view of process control timeline is shown in Figure 3-12. The VAX sends a message to the HP2117 indicating that it is ready for an event (such as end-of-scan) to occur. The VAX will not continue processing until that message has been physically read by the HP. Using this synchronization technique, the VAX will never get ahead of the HP2117 controller. The process is terminated by passing a message "998" to the HP2117 at the beginning of a scan. This signals the HP2117 to terminate the link to the VAX and to end its execution.

3.4.3.4 Software

A high-level flowchart of the radar control software resident on the HP2117 is shown

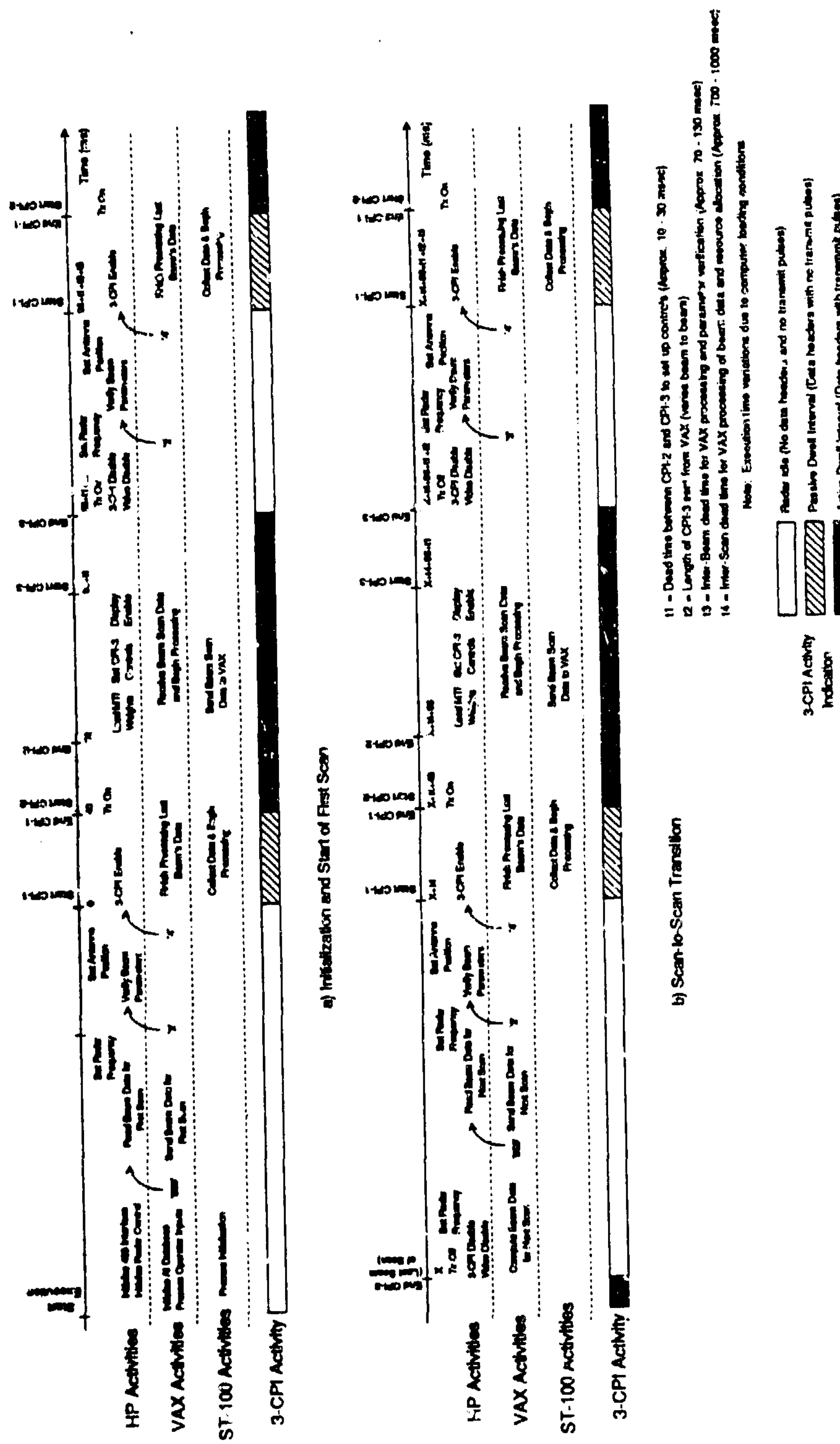

Figure 3-12: Process Control Timeline

in Figure 3-13.

The radar control software is organized in a modular structure with three separate levels of control. The top level (RUNAI) establishes communication with the VAX and begins operating the control software. The second level contains the main processing and synchronization functions (CTLAI, CPI1_2, LASTCPI) which perform the radar control and timing function. The third level of software contains the individual device control routines (DCU_CONTROL, C_FREQ, RECVI_SUB, PTFSUB) which send the actual controls to perform the radar control. A call diagram is shown in Figure 3-14.

Functionally, the process is initiated by RUNAI which initializes the HP2117 to VAX IEEE-488 bus interface and calls CTLAI. The CTLAI is the main control routine for the process. The CTLAI routine first initializes the antenna position for all beams, the receiver frequency, and reads in the PTF weight table for all weight indices. After receiving the first end-of-scan semaphore (semaphore = 999), the processing begins. For each scan, 720 words of data (40 beams/scan * 18 parameters/beam) are received from the VAX via the IEEE-488 bus. (The parameters received for each beam are given in Table 9.) Those not implemented by the radar control software are included for future growth of the control scheme.

After all 720 parameters are received, CTLAI enters a beam loop which is executed 40 times (or until a last_beam flag is encountered). The first operation in the beam loop is to check the frequency index for this beam. If it is different than the last beam, the receive frequency is changed. The process then waits for an "end of CPI3" semaphore (semaphore = 2) and verifies all received parameters for this beam. If all are within limits, a "start CPI1" semaphore is expected (semaphore = 4) which initiates the antenna to be instructed to the proper position and for CPI1 to begin. Following the completion of CPI 1 and 2, the PTF is loaded with the selected weights and CPI3 is executed. This is the end of the beam loop.

The user can request that control parameters received from the VAX and implemented on the radar be saved in a file. This option is intended to assist the operator in post experiment analysis of controller actions taken in response to the environment and to aid in debugging the communications link, if necessary. Writing the control parameters to a file slows the control software and is not normally used. The software also allows the user to specify the mechanical pedestal position which is the center of the scan volume. The VAX software commands the antenna position relative to the user-selected scan center angle.

3.4.3.5 Radar Control Timing Signals

We used as many of the existing radar timing signals as possible in order to preserve

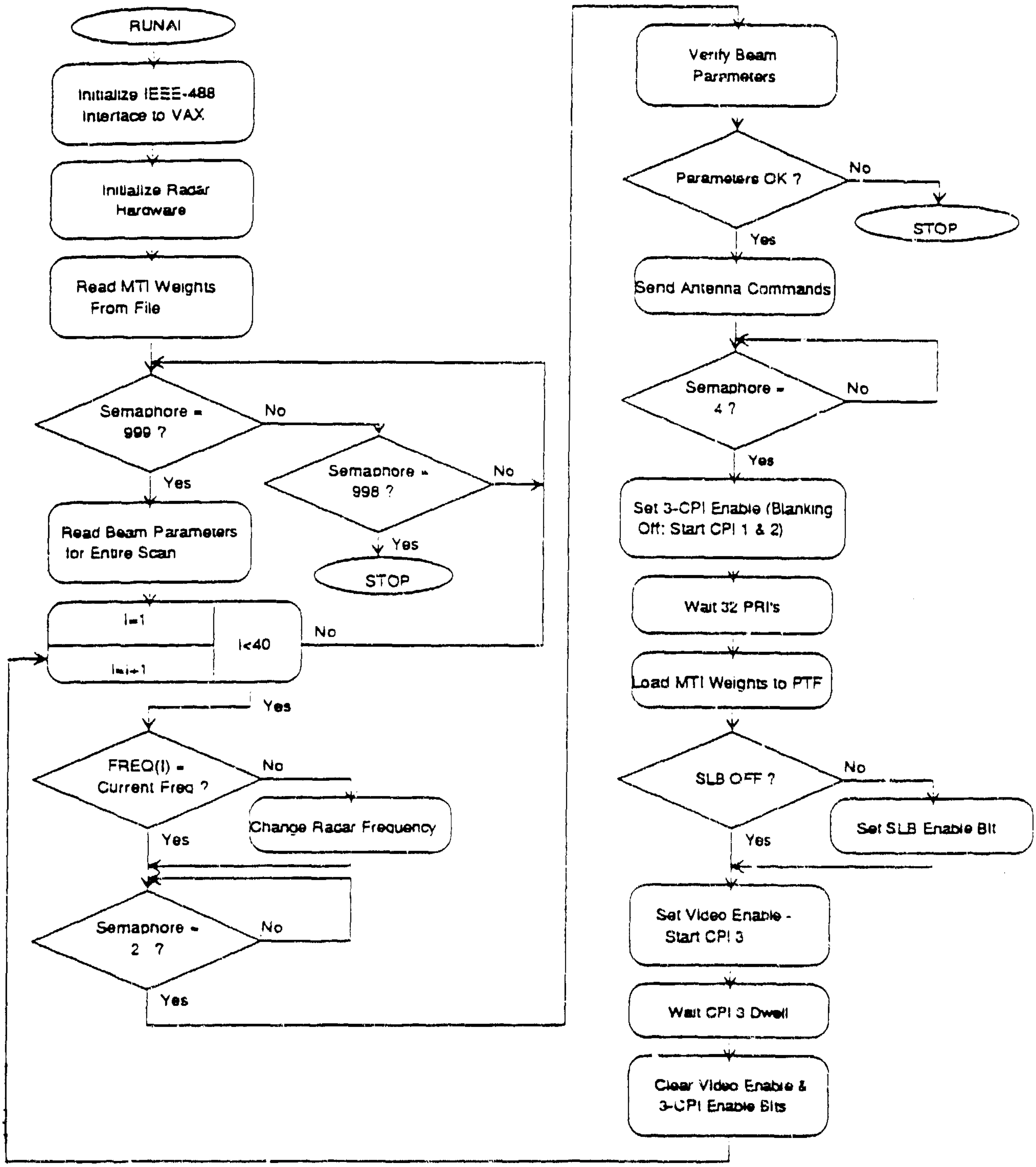

Figure 3-13: Radar Control Software Flowchart

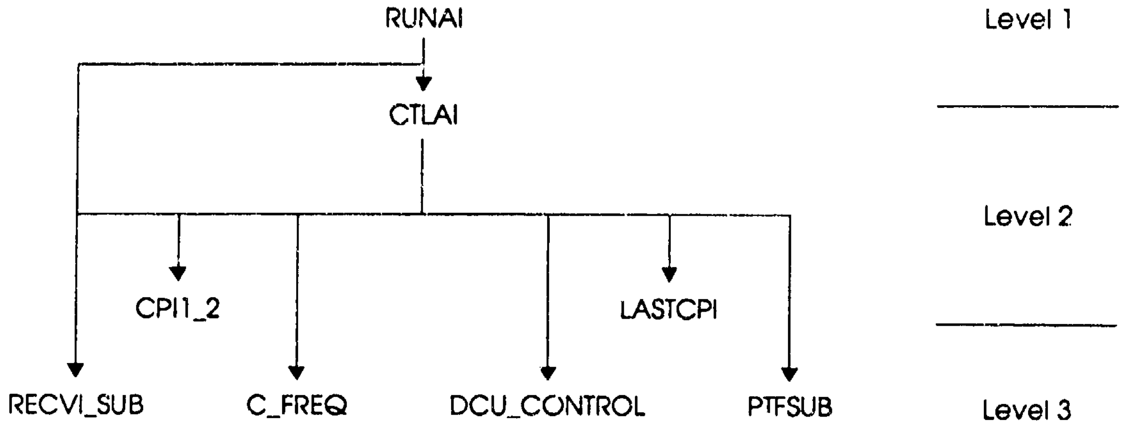

Figure 3-14: Radar Control Software Call Diagram

the relative timing and to minimize the amount of additional hardware and software required to generate the signals. The approach taken was to use the free-running PRF strobe from the existing radar control and timing unit (RTCU) equipment to trigger all events in the radar. The C-Band range tracker was employed to generate the transmitter timing signals, the exciter strobe (WFG trigger), the antenna phase shifter timing gate (PSTG) signal, and the range zero trigger for the display. The timing of the signals generated in the range tracker relative to the PRF trigger is downloaded during the set-up of the experiment and does not change during the experiment. The existing AIGATES software downloads a set of prestored parameters to the range tracker. Additional hardware gates some of the timing signals for proper operation of the AI software. The ESP, therefore, requires that the transmitter be off during CPI1 and on during CPI2 and CPI3. The 3-CPI circuit performs the function of gating the transmitter and WFG trigger during CPI1. Additionally, the 3-CPI synchronizes radar data transfer to the ST-100 by gating the PRF strobe to the header generator which is used to initiate transfers to the ST-100. A macro-level timing diagram is given in Figure 3-15.

When the ENABLE signal from the HP2117 static register control bit is inactive (low), no strobes are generated to the DHI or to the WFG, and no RF gates are generated to the transmitter. When ENABLE transitions to the active state (high, controlled by the radar control software), 16 replicas of the PRF strobe are sent to the DHI, allowing 16 PRIs of passive dwell (no transmit signal) data to be sent to the ST-100. Following these 16 passive dwells, timing signals are sent to the DHI, WFG, and RF gate to allow transmission of waveforms and transfer of data to the ST-100 until the ENABLE bit is again placed in the inactive state. This provides an active dwell (transmitter on) equal to the duration the ENABLE bit is high minus the 16

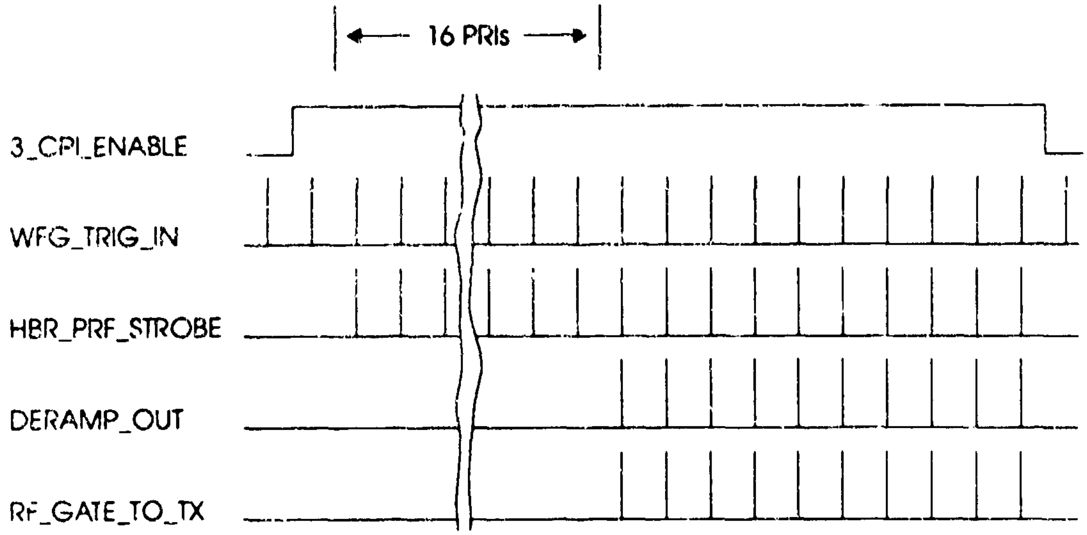

Figure 3-15: Macro-Level 3-CPI Circuit Timing

PRI passive dwell. The length of CPI3 is controlled by the radar control software by maintaining the ENABLE bit in the active state for the amount of time passed from the VAX.

3.4.3.6 Radar Configuration

Certain radar parameters such as the waveform characteristics, PRF, and A/D clock rate are not presently controllable by the radar control software. The primary reason is that the amount of effort required to make these parameters adaptive would have detracted from the efforts necessary to achieve the program goals. Control of these parameters was, therefore, not made a part of the experimental set-up and was not integrated into the run-time radar control software. The existing SPLEX software is used to configure the radar PRF, the WFG parameters, the radar data rate, and the receiver data path selection prior to the experiment. These settings remain constant during the experiment.

3.4.3.7 Video Blanking

Due to the sequential processing of data from three CPIs, and the fact that the desired countermeasures were implemented only during CPI3 (the detection waveform), the display (UPA-62) needed to be blanked during all but CPI3. Initial attempts were made at gating the range zero pulse to the PPI as a means of blanking the display. The result was a loss of the range reference during the first several dwells of the detection waveform, and a blurry display of uncancelled target and clutter returns. The final implementation, which resulted in a very clear display and timely gating of the video signal, was to use the bias control input on the final video amp to gate the signal. By placing the bias control in an inactive state except during the detection waveform, the PPI is blank during dead time and environmental assessment periods. Only the processed result of the AI controlled radar, with the appropriate countermeasures implemented, is displayed.

3.5 Display

3.5.1 Requirements

The IMRD top-level display requirements were:

1. Display all processed real targets and environmental interference over the selected scan volume in real-time at a bandwidth commensurate with that for the radar video.
2. Provide a believable visual indication of target detectability with specific cross sections, as required by the SOW.
3. Maximize credibility by basing target detection indicator on real-world (as opposed to computed or simulated) outputs.
4. Use existing equipment to the maximum extent to reduce costs.
5. Display current AI status for each beam.

The first four are detection display requirements; the fifth is an AI display requirement.

Several options were evaluated, including:

1. Adapt the ACMDSP simulation to display detections, detection range contour, and AI status on a computer monitor.
2. Develop a new VAX-driven display.
3. Use a UPA-62 PPI for real-time status display and a separate VAX-driven Tektronics display to show detection performance and AI status. UPA-62 display input could be driven by either the dedicated hardware output data or the ST-100 output data via the ST-100 to PPI (SPI) interface.
4. Use UPA-62 PPI to display detections and detection performance, and a separate VAX-driven display to show AI status.

Option (1) would have required extensive rework of the existing ACMDSP code to enable displaying real-world data– especially for clutter, which would consist of several individual reports (one for each cell) instead of variables defining boundaries of the clutter as in the ACMDSP. Also, the detection contour performance indicator is highly suspect since it is based on a top-level software simulation as well as on knowledge of the scenario being simulated and, hence, was impractical for a credible real-world display. It is unlikely that it could have performed in real-time.

Option (2) would have required extensive new software coding and consumed a disproportionate amount of contract resources. Moreover, it is doubtful whether it would have been able to keep up with all detection reports in real time. Therefore, requirements (1) and (4) would probably not have been met.

The UPA-62 options were preferred from both a cost and a performance standpoint: interfaces already existed from either the dedicated digital processing chain or the SPI interface; furthermore, they supported the bandwidths of the video data (nominally 1 MHz). The AI status display could then easily be handled by a VAX-driven graphics terminal since the bandwidth requirements are much lower.

The performance indicator display distinguishes options (3) from (4). Within options (3) and (4) two sub-options for graphical indication of performance were considered:

a. a detection range performance indicator, and

b. a minimum detectable signal performance indicator.

Generating the detection range contour option requires a software simulation of at least portions of the entire signal processing string to generate, and hence lacks credibility. Also, if detection is clutter-limited, more than one detection range value may be necessary if there is significant variance in interference characteristics in range, e.g. a transition from severe ground clutter at close ranges (with possibly a low detection range) to *in the clear* at further ranges (with possibly a large detection range). In lieu of the detection range display, the PPI display shown in Figure 3-16 (Option 4b) was selected.

We determined that the most credible demonstration would use the UPA-62 to display returns from the adapted waveform and a separate VAX display to track AI performance status.

3.5.2 Detection Display

The advantages of the UPA-62 detection display are

1. It does not rely on simulations of the signal processor to display true performance.
2. It enables the display of target detectability in selected range regions for each beam and thus better accounts for variations in detectability versus interference background variations within a beam.
3. It provides relative performance advantages in dB of AI adaptivity, providing a bottom-line assessment of AI for the demonstration.
4. We believed it unlikely that the VAX could display real-world data in real-time.

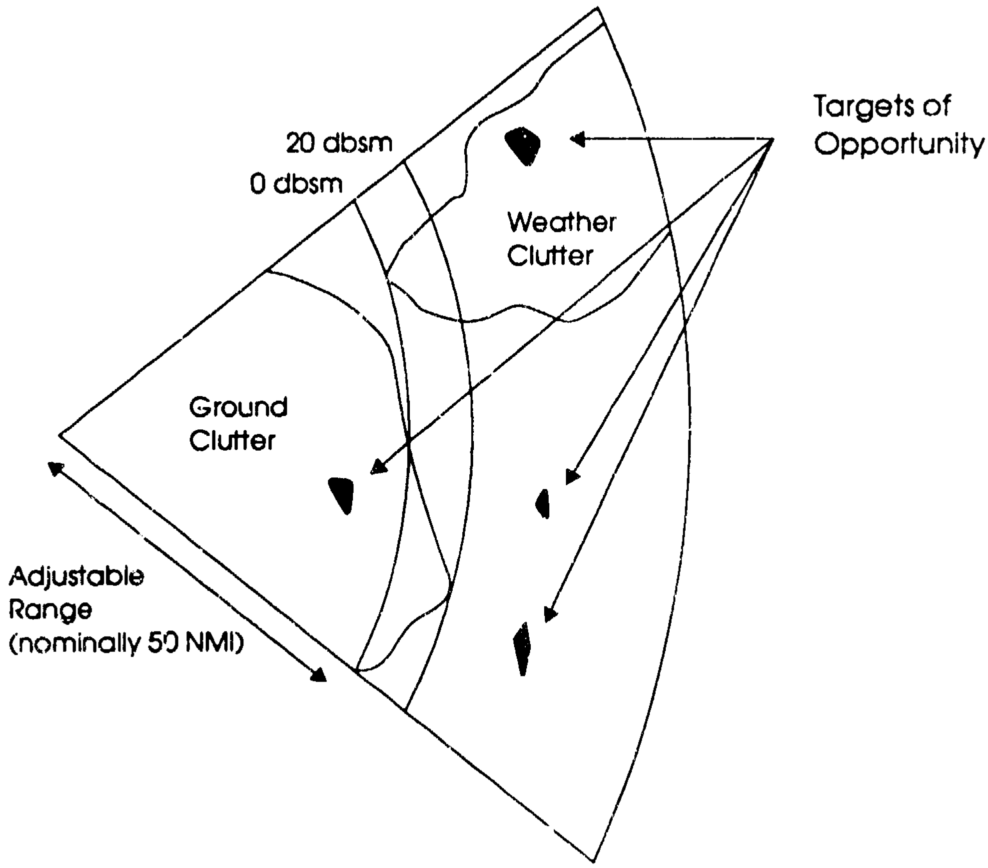

Figure 3-16: Real-Time Detection and Performance Display

As shown in Figure 3-16, the display has two concentric rings indicating two constant-range targets in each beam with varying cross sections controlled from the SPLEX T menu. The range of the rings is selected by the operator for positioning, as desired, within different interference zones. These simulated target rings are injected digitally ahead of the detection signal processor (see Figure 2-3).

The procedure for gauging target detectability is to determine the minimum discernible signal (MDS) ring at each azimuth cell; this roughly indicates the minimum target RCS that can be detected at the ring's range. Comparison of the MDS before and after AI adaptivity (with constant scan time) thus provides an approximate visual indication of the advantage in dB afforded by AI as function of azimuth (and also as a function of range by moving the range position of the rings).

We considered two options for injecting the simulated target rings: the existing target simulator (digital injection) or the HP8770 Arbitrary Waveform Synthesizer (AWS) and existing up-converter circuitry to inject the signal at the C-Band antenna. Both implementations are straightforward. The former technique has disadvantages in that no more than two target rings can be generated by the existing equipment and the injection point bypasses the sidelobe blanker, sidelobe canceller, and environmental signal processing. The latter technique requires some software to calculate the waveform code for two superimposed targets at the positions of the rings and control software for reprogramming the AWS in concert with waveform changes. The former was chosen due to its relative simplicity.

3.5.3 AI Display

The AI-specific display requirements are 1) to provide variable levels of information about the adaptive control status for diagnostics during rulebase development (diagnostic output), and 2) to inform the user of detected environmental conditions and resulting radar parameter changes during operation (runtime output). Information to be displayed is selected through default parameters at system startup or through the User Interface during operation.

Diagnostic output for use during system development can be sent to a file. It includes

1. Beam Scan Data.
2. The radar parameters array, which is the output of Determine Applicable ECCM.
3. Beam Information, a data structure which merges Priority Zone and Quality Option information.

Runtime output during system operation can be sent to the screen and/or to a file. It includes

1. Sources of interference identified for each beam, including associated parameters.
2. The quality option selected for the next scan.
3. Radar Control Parameters as sent to the HP2117. This output can only be sent to a file due to the quantity of information.
4. Summary of the interference identified, controls applied, and dwell time allocated for each beam.

Typical output for interference identification would appear as follows:

```
A nb cw sl jammer was found in beam 1, frame 1
   Peak_JNR = 1.3339e+03, Jammer_Range = 1, and
   Mainlobe_Duty_Factor = 1.000
```

or

```
Ground clutter was found in beam 3, frame 1
   Peak_CNR = 5.1289e+03, CNR_Maxrange = 1.1714e+03, and
   Maxrange = 1.5750e+04
```

Typical output indicating the plan for the next scan:

```
AI results for Frame 1

Beam    Dwell Time     Interferences           ECCMs Changed
         per Beam          Found

 1        0.057       nb_cw_ml_jammer        frequency changed
 2        0.057       nb_cw_ml_jammer        frequency changed
 3        0.057       nb_cw_ml_jammer        frequency changed
 .
 .
 .
```

The AI Display is menu-driven and completely self-explanatory. Sample dialogues are included in the User Manual [UM].

4 Demonstration and Training

4.1 Overview

The formal Integrated Multi-Domain Radar Demonstration occurred on April 17 and 18, 1991. This consisted of a briefing, followed by a demonstration of the IMRD configuration, and training in its operation.

At the briefing, we reviewed IMRD capabilities and plans for demonstration. It was also an opportunity to discuss contract findings and suggestions for further work.

The demo followed the *Test Plan and Procedures for Integrated Multi-domain Radar Demonstration*, delivered on April 17, 1991. This document describes equipment set-up procedures, the demonstration plan, and the specific test cases with their expected results.

During training, RLSF personnel learned to set-up, operate, and modify the IMRD experiment. Training material included the *User Manual for Integrated Multi-Domain Radar* (CDRL A006), and *Training Notes for Integrated Multi-Domain Radar* (CDRL A004), both delivered on April 17, 1991.

The remainder of this section describes the test objectives and specific categories of test conducted.

4.2 Test Objectives and Categories

Demonstration and training focused on three test objectives:

1. Identify and counter single sources of interference in radar data.
2. Identify and counter multiple sources of interference in radar data.
3. Allocate dwell time based on user-specified Priority Zones and scan times.

Targets of opportunity, as available, and simulated targets injected into the display were observed during tests. Three categories corresponding to these objectives are described in the remainder of this section. Table 12 lists the specific tests that were conducted and Appendix J.9.2 contains their complete descriptions.

Category 1: Identify Single Sources of Interference and Adapt Radar Parameters. Single sources of interference were identified by the ID Passive Interference and ID Active Interference rulebases. The first phase of each of these tests was conducted with the AI adaptive control turned off. The AI Display listed interference identifications, but with no subsequent radar parameter changes were made. The following types of interference were identified:

1. Jammers discriminated by narrowband or wideband, CW or pulsed, and mainlobe or sidelobe.
2. Ground clutter.
3. Weather clutter.

The next phase of each of these tests was conducted with the AI adaptive control turned on. The AI Display listed detections and subsequent radar parameter changes as selected by Determine Applicable ECCM.

The following control changes were made:

Interference	Radar Control Output
1. NB CW and pulsed jammer	Frequency change
2. Pulsed widelobe sidelobe jammer	Sidelobe blanking
3. CW wideband sidelobe jammer	Sidelobe cancelling*
4. Ground clutter	MTI
5. Weather clutter	MTI, transmit polarization

* Sidelobe cancelling was selected by the Adaptive Controller in software but was not implemented in hardware.

Beam dwell time was increased, as necessary, to increase the transmit energy per beam for achieving the probabilities of detection assigned to the different quality option plans.

This category is implemented by test cases 1 through 3 as well as by the first steps in tests 4, 6, and 7 listed in Table 12. Test 4 was also conducted with the ECCM frequency change disabled to demonstrate that the AI rulebase selects alternate controls when the first choice is not available.

Category 2: Identify Multiple Sources of Interference and Adapt Radar Parameters. Multiple sources of interference were identified by the ID Passive Interference and ID Active Interference rulebases. This was conducted with AI on. The AI display listed interference detections and subsequent radar parameter changes as selected by Determine Applicable ECCM subproblem.

The following combinations of interference were identified:

1. NB jammers and ground clutter
2. Two NB CW jammers
3. Two WB CW jammers
4. NB Jammers and ground and weather clutter (weather permitting)

This category is implemented by test cases 4 through 7, listed in Table 12.

Category 3: Allocate Dwell Time Subject to Priority Zones and Quality Options. We demonstrated the impact of user-defined Priority Zones on scan time allocation as well as software facilities for changing Radar Control Parameters. This was conducted with AI on and interference detections were listed on the AI Display. Subsequent control changes, as selected by the Determine Applicable ECCM subproblem, and beam dwell times, as calculated by the Allocate Radar Resources subproblem, were also displayed. The user had the option to make changes to the Priority Zones and scan time and to observe the impact on resulting dwell time and target detectability in the presence of various types of interference.

This category is implemented by test case 8, listed in Table 12. See Figure 3-6 for the Priority Zones and types of interference used for this test.

Test	Description	Declaration	Adaptive Control
0	Benign Environment		
1	Ground Clutter	ground	mti_weight changed
	(poss. Weather)	weather	tx_polarization changed
			mti_weight changed
2	WB ASP Jammer	wb_asp_sl_jammer	sidelobe_blanking on
		(or _ml_jammer)	
3	WB CW Jammer	wb_cw_sl_jammer	sidelobe_cancelling on
		(or _ml_jammer)	
4	NB CW Jammer	nb_cw_sl_jammer	frequency changed
		(or _ml_jammer)	frequency changed
	(plus Ground)	ground	mti_weight changed
5	Two NB CW Jammers	nb_cw_sl_jammer	frequency changed
		(or_ml_jammer)	frequency changed
	(plus Ground)	ground	mti_weight changed
6	NB ASP Jammer	nb_asp_sl_jammer	
		(or _ml_jammer)	frequency changed
	(plus Ground)	ground	mti_weight changed
7	NB ASP, no freq. change	nb_asp_sl_jammer	sidelobe_blanking on
		nb_asp_ml_jammer	
	(plus Ground)	ground	mti_weight changed
8	Quality Options	wb_cw	dwell time adjustments

Table 12: IMRD Test Cases

5 Conclusions and Recommendations

The IMRD program has successfully demonstrated the feasibility of an AI-based adaptive radar controller implemented on a real, as opposed to a hypothetical, radar system. In doing so, a major step has been taken toward the longer-term goal of employing knowledge-based radar control on operational advanced tactical surveillance radar systems. We believe that this program has established a solid foundation for capability enhancements and additional investigations using the testbed equipment.

5.1 Major Conclusions and Lessons Learned

Our program achievements include:

1. Successful demonstration of all components of the knowledge-based controller in a real-world scenario, including the ability to assess and discriminate among multiple interference sources, to formulate candidate plans of action to improve target detectability, to choose the plan that best meets user-defined scan time and priority zone constraints, and to carry out the plan via changing control parameters for a real radar system.
2. Extensive signal and data processing capabilities implementation in real-time using available RLSF equipment.
3. A testbed and performance benchmark to enable the government continued rulebase evolution; we have trained RLSF personnel to use it.
4. Hardware and software documentation to facilitate development.

Accomplishing the above was a challenge due to the leading-edge, exploratory nature of the effort, and to our decision to use the RLSF equipment at near maximum capabilities. As a result, many issues were flushed out that would not have emerged from a paper study. We therefore consider this report's documentation of lessons learned and of the design evolution to be as important as the successful demonstration performed at the conclusion of the contract.

Lessons learned are discussed throughout this report. Some of the more substantial observations are summarized below.

Rulebase Development: Of all challenges we faced during this contract, the most difficult and time-consuming was defining a robust AI rulebase for identifying interference, especially when the observed data contains multiple, overlapping interference sources. It was difficult to develop a set of rules that matches the effectiveness of an experienced human operator in recognizing and discriminating interference. This problem was exacerbated by real-world factors such as equipment errors and variability in the appearance of the interference. For example, the range and doppler characteristics of weather are dependent on many factors including rainfall rate, beam angle with respect to the wind, wind velocity, and location of the rain.

Real-Time Operation: The second most difficult challenge was paring down system functionality to enable real-time operation within equipment and contract resources. Indeed, we evaluated a number of more sophisticated control features, processing techniques, and rulebase capabilities that may be applicable to an eventual operational radar, but these were not implemented because the performance improvements (i.e., credibility enhancements for the demonstration) were insignificant compared with the additional program risk. Our philosophy was that the successful demonstration of modest capabilities was preferable to an unsuccessful demonstration of a more elaborate system. The result of this philosophy is a firm foundation on which additional capability can be built.

Provisions for Growth: While focussing on achieving a *working* demonstration, we needed to build in flexibility for *eventual* system enhancement and evolution. We have accomplished this through overall design, software modularity, and parameterization. As a result, this testbed is a foundation for rulebase evolution within RLSF C-Band radar *and* can be configured for or evolve to different equipment environments.

AI versus Human Intelligence: The bottom-line is inevitably how does a human operator compare to the AI controller. One may argue that an experienced human will be at least as effective as the AI controller in recognizing interference and making simple changes to radar control such as turning on a sidelobe blanker or changing the radar frequency. However, two major advantages of the AI controller over a human operator are its automation relative to a human operator (with capability for unattended radar operation) and its ability to make complicated *real-time* control decisions that may be beyond human capability. Examples of such decision policies are the Priority Zone and Quality Option plans demonstrated under this contract.

5.2 Suggestions for Future Work

5.2.1 Rulebase Development

The following sections suggest improvements that might be made in identifying the different sources of interference.

5.2.1.1 Neural Net Implementation

Because of their ability to find patterns in noisy data, neural nets are applicable to interference identification. The expert system submodules which identify jammers and clutter could be replaced by neural nets whose input neurons correspond to elements in the Beam Scan Data and whose outputs are interference identifications. Taped radar data from the RLSF C-Band radar would be used to train the neural net and to compare neural net and expert system performance.

The major outcomes of this would be enhanced performance of the AI testbed and an opportunity for technology assessment through the neural net/expert system comparision.

IMRD Enhancement: The IMRD rules for identifying interference are relatively straightforward and are based on the reasoning of radar experts. Neural net supervised learning should allow the system to reflect the real environment more accurately, and to better accommodate noise data. As the system evolves, changing environments can be modeled by adjusting neuron weights through new training cases more efficiently than if expert system rules needed to be redefined.

Neural Net Technology Assessment: Comparative performance measurement is one component of the DARPA program in Artificial Neural Networks. Its objective is to determine the advantages and disadvantages of neural nets with respect to conventional technologies. Looking for interference patterns in noisy data appears to be a natural application for neural nets, and IMRD presents a convenient opportunity to make such a comparison with expert system technology.

5.2.1.2 Uncertain Knowledge

A measure of certainty could be included in each interference identification to use in the plan selection for the next scan. Uncertainty was to some extent incorporated into JNR and CNR measurements that accompanied detections, but no explicit probabilities were used. This higher level of sophistication would have been overkill at this stage will be appropriate as the rulebase evolves. Among possible models are Bayesian probability, certainty factors, Dempster-Shaffer logic, and fuzzy sets. These should be reviewed in the context of contract achievements.

Bayesian probability is most applicable when conditional probabilities among events are understood. For example, it would be useful to correlate detections over multiple scans or in adjacent beams.

Certainty factors are a popular way to build uncertainty in a rulebase, but fairly ad hoc and hard to validate. Implementing them in the current Prolog rules would be straightforward.

Demster Shaffer logic entails a set of beliefs about the truth of an event. It is more qualitative than this application, but could be appropriate in a system that supports a greater level of real-time tactical input by the operator. There is some disagreement about the best way to incorporate this logic into inferencing, but several different options could be explored.

Fuzzy sets are particularly good model for areas such as sensor fusion where information from independent sources is available. They could be applied to merging independent determinations of interferences such as from separate rulebases on physically separated systems.

5.2.1.3 Jammers

Wideband jammer identification could be improved by maintaining a scan-to-scan history of jammer measurements, with each measurement at a different frequency, or preferably by a wide-open instantaneous bandwidth receiver that covers the entire radar agile bandwidth. The former would require an increase in data processing software complexity, the latter, additional equipment with very high-speed A/D converters.

For a low-power pulsed jammer with low duty factor (i.e., very low average power jamming signals), improvements in sensitivity and accuracy of the jammer bandwidth measurement could be afforded by confining the spectral analysis (i.e., FFT) to those samples within the detected jammer pulse rather than over the entire PRI interval.

Improvements of the adaptive control response for narrowband jammers could be afforded by choosing the least-jammed frequency based on a scan-to-scan history or possibly a pseudorandom selection. Presently, frequency changes are made sequentially over the channel spacing of the C-Band radar.

5.2.1.4 Ground Clutter

A constant clutter detection threshold was implemented for all range and doppler. This threshold must be set low enough so that returns near the maximum clutter range are detected. However, low threshold settings cause false detections in the doppler sidelobes of near-range ground clutter. A suggested improvement is a range varying threshold, or preferably STC preprocessing to better model the approximately $1/R^3$ level variation of the ground return.

To estimate clutter power versus range and doppler, the range-doppler plane was segmented into block integrations for smoothing the variations. Additional smoothing and suppression of moving targets and nonstationary interference could be afforded by calculating a scan-to scan clutter map.

Ground clutter assessments are made at every beam position for every scan to satisfy the SOW requirement of a one-scan reaction time to changes in the environment.

This one scan response is perhaps overkill in a groundbased radar since ground conditions will not change rapidly. Therefore, to free up the radar timeline and reduce processing, it makes sense to perform these measurements less frequently. This can be accomplished by using the software capability to change CPI2 update rate that is built into the Quality Options Tables.

5.2.1.5 Weather Clutter

The weather ID function is performed on every beam and every scan using the CPI2 waveform. Identifications are based on integrating the clutter power in the wedgelike range-doppler region where the clutter is expected to appear, as well on as counting range-doppler cell threshold crossings in that region. Even though we successfully identified weather interference based on range-doppler characteristics, the rulebase could be made more robust by using additional discriminants or by more complex estimation of interference statistics.

As with the ground clutter declaration, a range-varying threshold (on the order of $1/R^2$) should improve the sensitivity and accuracy as well as multiple scan integration for additional smoothing. To save radar timeline and reduce the processing load, the weather may not need to be measured on every scan since it will not change rapidly.

Additional measurement discriminants would also offer improvement. Two examples are incorporating the power ratio of same-sense versus opposite sense circular polarization, and including the variability ratio discriminant that is already calculated by the ST-100 but not used in the rulebase. The former technique was not implemented since we believed that the equipment interface modifications required to do so would have exceeded contract resources.

5.2.1.6 Overlapping Interference

Suggestions to alleviate problems with overlapping interference include:

1. Incorporate jammer cancellation during CPI2 (via SLB, SLC, and MLC) to improve isolation of measurements by reducing or eliminating jamming signals that tend to interfere with the ID clutter process.
2. Incorporate additional discriminants such as variability ratio, polarization ratio, or determination of clusters.
3. Use range-varying thresholds to improve sensitivity of clutter detection.

5.2.1.7 Multiple Scan Processing

In addition to the single scan growth items discussed previously, track-while-scan processing of multiple targets, of point clutter sources, and of jammer sources could be considered for a follow-on effort. Rulebase enhancements could be formulated to examine track features and insure radar survivability. A multi-scan history of past ECCM could be maintained so that the AI rulebase can deal with past failures, or maintain past success in optimizing the radar response to the target, clutter, and jamming environment.

Additional measures that could be made include

1. Passive dwell measures (for each jammer): angle, angle rates and polarization, polarization rate, % jamming duty versus time, inband frequency modulation, and mean and variance of amplitude versus time and doppler.
2. Active dwell point clutter measures: radar-based position and velocity state versus time of point clutter, erratic motion targets, and track quality indicators (missed reports, SNR)
3. Active dwell target measures (for each target): principle and opposite polarization amplitude statistics versus time, inertial position and velocity state of credible targets versus time, inertial position and velocity covariance of credible targets versus time, and track quality indicators (missed reports, residuals, SNR).

Potential additional control responses not implemented in the present AI testbed which could be simulated or executed in the real test environment are summarized in Table 13.

Table 14 summarizes potential scan to scan environmental measures. A brief discussion for two of them follows.

Point Clutter Map (PCM): A PCM could be implemented as an m-scan history track-while-scan (TWS) file maintained in radar coordinates. The PCM inhibits reporting point clutter or target returns which do not satisfy operator-designated threat characteristics in terms of speed, heading, and maneuverability. The PCM also inhibits reporting point clutter returns which do not associate from scan to scan. This allows identification of interference as uncorrelated clutter due, perhaps, to anomalous propagation, or to correlated returns such as bird flocks. Such information is important in subsequent scan waveform selection. As part of an adaptive control, the target screening characteristics may be changed, according to operator designated priority, to increase or decrease the point clutter screening performed by the PCM.

Target Track-While-Scan File: The target TWS file could be implemented as a multi-scan history of track states established on credible targets which survive the

Applicable ECCM	Rationale
Intradwell frequency agility	Estimate target parameters
Bandwidth increase	Resolve targets versus point clutter
Beampacking adjustment	Increase target detectability
Beamspoiling adjustment	Increase azimuth coverage
Adaptive AGC/STC	Respond to saturating targets/clutter
Adaptive track filter gain, correlation gates, update rates	Detect and respond to target maneuvers and jammer angle maneuvers
Adaptive point clutter, filter gain, gates, update rates	Maintain false alarm control on point clutter, erratic motion targets
Designation to alternate RLSF radar (S- or L-Band)	Avoid spot or barrage jamming
Emission control (EMCON)	Survival during antiradiation missile (ARM) launch event
Decoy activation	Survival during ARM launch event

Table 13: Additional Active Target Measures and ECCM

Interference Estimate	Purpose
Point clutter map	Birds, discrete clutter, anomolous propagation
Passive dwell jammer map	Non-responsive jammers
Active dwell jammer map	Responsive jammers
Active dwell clutter map	Area and volume scatters
Target TWS file	Credible targets

Table 14: Potential Scan-to-Scan Environment Sensors

PCM discriminants. Track states are maintained in an inertial coordinate set (such as local topocentric) to allow maximum observability of target motion components, ease of track maintenance and propagation, and fusion with other sensor data. Although the track states could be updated primarily by successive surveillance scans of the C-Band radar, track states from the S-Band dish tracker could also be considered. The target TWS file would provide the AI decision-making process with an accurate, unambiguous history of target motion, particularly acceleration and split track events (i.e., missile launch) and allow assessment of threatening events. Similarly, the jammer TWS file would provide an accurate, unambiguous history of jammer angular motion. Tracking software would most likely be hosted on the VAX.

A digital report extractor would be required in the PTF signal processor string to provide target digital range, angle, SNR reports to the TWS host computer. The Sensis MSC 68000 platform is a potentially cost-effective implementation technique. As a less costly alternative, the digital buffer which is presently at the output of the PTF string may be sufficient. This 10K buffer records limited data, however, and requires significant overhead time to read the data into the HP2117 and VAX. The VAX must then extract digital range and angle reports.

5.2.2 DC Offset

Small amounts of DC offset in the receive video amplifiers or A/D converters lie in the zero doppler bin and can mask ground and weather returns in the doppler mainlobe and sidelobes, respectively. The DC offset was manually adjusted to below an LSB prior to each demo of the AI testbed, but it tended to drift with time and varied by about an LSB depending on whether the transmitter was on or off (perhaps this was due to slight changes in ground voltage potential).

One solution is to build a software DC canceller in the ST-100 with the requisite time constant. In moderate wind, ground clutter has a velocity standard deviation of about .1 m/sec, or 3.33 Hz at C-Band. To remove the DC offset without cancelling significant portions of the ground clutter requires a canceller bandwidth that is substantially less; a time constant of greater than three seconds would be sufficient.

5.2.3 Signal Processing

The signal processing in the current IMRD configuration is limited by the capabilities of the real-time signal processing hardware and by the throughput and I/O capacity of the ST-100. Additional signal processing hardware would allow for system enhancement such as target detection processing, simultaneous transmission and processing of the active environmental assessment dwell (CPI2) and the detection dwell

(CPI3), and finer granularity of the range and frequency vectors to allow more precise characterization of the environment.

The current system and rulebase estimate signal-to-noise improvements based on assumed ECCM effectiveness against a postulated target. There is no feedback in the rulebase to evaluate the effectiveness of the system to improve target detectability and reduce false alarms. With increased signal processing capability, quantifiable performance improvement could be made using target data (both simulated and real) obtained during CPI3.

Parallel channels into the signal processor would provide the capability to process two or more of the transmitted waveforms simultaneously. For example, if the raw radar from a single active dwell were available to the signal processor, it could derive both the environmental assessment data and the detection data from the raw input. The main advantage to this enhancement is a reduction in total scan time.

Additional signal processing horsepower would allow for larger numbers of range and frequency samples to be evaluated during environmental assessment. In the current system, several radar range cells are averaged to form a composite processing range cell due to the limited processing throughput of the ST-100 and scan time requirements. Finer range resolution of targets and clutter would be gained by utilizing smaller range cells. Larger FFT sizes could also be used to give better estimates of the passive interference frequency characteristics.

5.2.4 Radar Control

Initially, we intended to implement the Radar Controller exclusively by the HP2117 software, augmented with the existing RTCU and remote module capabilities. During the implementation and test phase of the contract, however, the design evolved toward a hardware controller. For example, the 3-CPI circuit performs one of the functions originally intended for the HP2117. In retrospect, a special-purpose hardware controller implementation might have been more suitable for the remaining functions, although success was still achieved using the HP2117 for this function.

This evolution should be continued in the context of projected RLSF radar control needs. The alternate Adaptive Controller platform (Section 5.2.6) could provide an interface to this control function.

5.2.5 Enhanced Graphics Display

The present system has a somewhat limited display capability since the detection waveform is displayed on a PPI. A graphics display would allow the AI information now displayed on a separate terminal to be merged with the detection information into a single, integrated display. Additional CPI3 processing would be necessary to format the data for an enhanced graphics display showing the areas of interference (jamming and clutter) as well as target returns.

5.2.6 Alternate Adaptive Controller Host

The VAX has allowed a real-time demonstration of adaptive control and has been a convenient development vehicle. However, it is not specifically appropriate for real-time processing and the current AI testbed is operating at the edge of its processing capabilities. Other platforms (e.g., one or several Suns) would allow more flexibility for an expanded rulebase, additional signal processing as suggested in Section 5.2.3, for other hardware interfaces for radar control (Section 5.2.4), and for the graphics display Section 5.2.5). Including higher speed processors to perform the interscan computations such as allocating radar resources (currently approximately 2-3 seconds per scan) would also decrease the scan rate. The portability of the Prolog code, as discussed in Section 3.3.3.1, would support this evolution.

6 Acronyms and Abbreviations

Acronym	Definition
ACMDSP	Adaptive Control for Multi-Domain Signal Processing
AI	Artificial intelligence
ASP	Asynchronous pulsed
CFAR	Continuous false alarm rate
CNR	Clutter-to-noise ratio
CPI	Coherent processing interval
CPU	Central processing unit
CW	Continuous wave
DARPA	Defense Advanced Research Projects Agency
dB	Decibels
DC	Direct current
DCU	Digital control unit
DHI	Digital hardware interface
DSP	Detection signal processing
ECCM	Electronic counter-counter measure
EMCON	Emission control
ESP	Environmental signal processing
FFT	Fast Fourier transform
FRDE	Flexible Radar Data Executive
GAFB	Griffiss Air Force Base
GFE	Government furnished equipment
IF	Intermediate frequency
IMRD	Integrated Multi-Domain Radar Demonstration
JNR	Jammer-to-noise ratio
LFM	Linear frequency modulation
LSB	Least significant bit
MLC	Mainlobe canceller
MSLC	Multiple sidelobe canceller
MTI	Moving target indicator

Acronym	Definition
NB	Narrowband
PC	Pulse compressor
PCM	Point clutter map
Pd	Probability of detection
PPI	Plan position indicator
PRI	Pulse repetition interval
PSTG	Phase shifter timing gate
PTF	Programmable transversal filter
RCS	Radar cross section
RF	Radio frequency
RFI	Radio frequency interference
RLSF	Rome Laboratory Surveillance Facility
RTCU	Radar timing and control unit
SIR	Signal-to-interference ratio
SLB	Sidelobe blanker
SLC	Sidelobe canceller
SNR	Signal-to-noise ratio
SOW	Statement of work
SPI	ST-100 to PPI Interface
STC	Sensitivity time control
TWS	Track while scan
WB	Wideband
WFG	Waveform generator

A Radar Design Parameters

Radar Design Parameters define the radar that is being adaptively controlled. They are read from three files at system initialization into a Pascal variable of design_data_type. The files are stored in ASCII format. Note that all values read in as dB are converted internally to ratio values prior to insertion in the design_data_type record.

A.1 Data Description

Radar Design Parameters contain the following components:

Number_Faces is an integer between 1 and 4 representing the number of active faces in the radar. This is set to 1 in the IMRD but the software will have the flexibility to model a radar with up to 4 faces.

Number_El_Angles is an integer between 1 and 40 representing the number of elevation angles in a scan. This is set to 1 in the IMRD but the software will have the flexibility to model a radar with up to 40 elevation angles.

Azimuth_Limits contains the following real data for each radar face:

- **Start_Az** is the first angle of the array face to be scanned.
- **End_Az** is the last angle of the array face to be scanned.

Set to -22.5 degrees, +22.5 degrees respectively in the IMRD (angle is referenced to array normal.)

Max_Range_Cells is an integer between 1 and 32 representing the maximum number of integrated range cells for active and passive range measurements. In the IMRD the width of these range cells (set to 4.5 km) is 30 times the width of the A/D output range cells due to integration. Set to 30 in the IMRD.

Max_Doppler_Cells is an integer between 1 and 16 representing the maximum number of doppler cells for active doppler measurements. Set to 16 in the IMRD. Note that this variable applies only to the active dwell (CPI2) doppler measurement.

Max_Pass_Dop_Cells is an integer between 1 and 32 representing the maximum number of integrated frequency cells for passive frequency measurements. In the IMRD the width of these frequency bins (nominally 31.25 KHz) is 1/32 of the A/D sampling rate due to integration of the raw data FFT output. Set to 32 in the IMRD. Note that this variable applies only to the passive dwell (CPI1) doppler measurement.

Peak_Power is a real number between 50,000 and 250,000 W representing the peak power output of the transmitter. Set to 100,000 W in IMRD.

Boresight_Gain_dB is a real number between 0 and 35 dBI representing the maximum antenna gain at the beam peak. Set to 31 dBI in IMRD.

Number_Of_Dwells is an integer between 1 and 100 representing the number of beam positions per scan. Set to 40 in the IMRD.

Coher_Bursts_Per_Beam_Dwell is an integer between 1 and 16 representing the number of coherent bursts in each beam dwell transmitted during CPI3. This is not used in the IMRD.

Samples_per_PRI is an integer between 900 and 1100 representing the number of data samples per PRI processed by the ST-100 for each assessment dwell. Set to 900 in the IMRD.

Number_of_PRI is an integer between 4 and 64 representing the number of PRIs in the active and passive listening dwells. Set to 16 in the IMRD.

PRI is a real number between 1e-6 and 4096e-6 representing the pulse repetition interval of the current beam in seconds. Set to 3000e-6 in the IMRD.

Sigma_Q is a real number representing the ratio of thermal noise rms to the A/D quantization level. Set to 1.0 in the IMRD.

Max_Radar_Freq_Index is an integer between 1 and 16 representing the number of unique radar frequency settings. Set to 8 in the IMRD.

Nominal_Radar_Freq is a real number representing the center frequency in Hz at which the radar is transmitting. Set to 5.775e9 Hz in the IMRD.

Radar_Instantaneous_BW is a real number between 0 and 10e6 Hz representing the instantaneous waveform bandwidth. Set to 1e6 Hz in the IMRD.

ECCM_Available is a record variable representing the ECCM that are enabled within the radar. Each boolean record indicates whether that ECCM is enabled or disabled.

- **SL_Blanking_On** represents whether the sidelobe blanker is available.

- **SL_Cancelling_On** represents whether the sidelobe canceller is available.
- **Freq_Change** represents whether the transmit frequency can be changed.
- **Circular_Polar** represents whether receive polarization can be changed.

Sig_To_Jam_Improve is a real number between 0 and 30 dB representing the improvement in signal to noise ratio achieved by employing sidelobe cancellation. Set to 0 dB in the IMRD.

Polar_Rain_Gain is a real number between 0 and 20 dB representing the increase in SCR achieved by applying polarization techniques to improve radar detection in rain clutter. Set to 10 dB in the IMRD.

Polar_Target_Loss is a real number between 0 and 10 dB representing the loss in SNR caused by employing polarization. Set to 7 dB in the IMRD.

MW_Loss_Trans_To_Antenna is a real number between 0 and 5 dB representing the loss in the signal path from the transmitter to the antenna. Set to 1.5 dB in the IMRD.

MW_Loss_Antenna_To_Preamp is a real number between 0 and 5 dB representing the loss in signal path from antenna to the preamp. Set to 1.5 dB in the IMRD.

Signal_Processing_Loss is a real number between 0 and 10 dB representing the loss suffered by a signal due to signal processing techniques employed. Set to 5 dB in the IMRD.

Freq_Sidelobe_Level is a real number between 0 and 100 dB representing the JNR improvement achieved by employing frequency hopping. Set to 10e5 in the IMRD.

Antenna_Sidelobe is a real number between 0 and 50 dB representing the antenna sidelobe level in dB. Set to 25 dB in the IMRD.

Sampling_Rate is a real number between 500e3 and 10e6 Hz specifying the sample rate of the video A/D converters. Set to 1e6 Hz in the IMRD.

Nom_Target_RCS is a real number between -30 and +30 dBsm representing the required minimum cross section of a target to be detected. Set to 10 dBsm for IMRD.

Nom_Target_Range is a real number between 0 and 200e3 meters representing the desired detection range to the target. Set to 40.0e3 m in the IMRD.

Range_Sidelobe_Level is a real number between 0 and 50 dB representing the range sidelobe levels achieved as a result of the pulse compression process. Set to 25 dB in IMRD.

Noise_Figure is a real number between 0 and 10 dB representing the noise figure at the receiver input. represented in db. Set to 4 dB in the IMRD.

Min_Instrumented_Range is a real number between 0 and 100,000 m representing the minimum range of the radar commensurate with receiver blanking during transmission. Set to 2400 m in the IMRD.

Beam_Spacing is a real number between 0 and 3 representing the minimum spacing between two adjacent beams in $\frac{\sin(\text{spacing})}{\sin(\text{beamwidth})}$ space. Set to 1 for IMRD.

MTI_Loss is a real number representing the loss in the target signal-to-noise ratio when MTI weighting is applied. Set to 0dB in the IMRD.

Active_Dwell_Time is a real number representing the data collection time for the active listening dwell (CPI2). Set to 48e-3 sec in the IMRD.

Passive_Dwell_Time is a real number representing the data collection time for the passive listening dwell (CPI1). Set to 48e-3 sec in the IMRD.

MTI_Weights is an array indexed by the Start Clutter Doppler and the End Clutter Doppler of the weather and/or ground interference found which contains three records: number of pulses; clutter-to-noise improvement and an index into an MTI weight table.

Signal-to-interference ratio (SIR) is an array indexed by probability of detection (Pd) and number of looks whose elements contain signal-to-interference ratios. The Pd is rounded up to the nearest .05 and the table and ranges from 0.1 Pd to 0.95. The number of looks ranges from 1 to 100 and then two additional rows represent 300 and 1000 looks respectively.

A.2 Pascal Declarations

Radar Design Parameters are stored in a Pascal structure of type design_data_type which is declared as follows.

```
const
  MAX_MAX_RANGE_CELLS    = 32;
```

```
  MAX_MAX_DOPPLER_CELLS = 16;

type

sir_pd_index_type = 1..18;
sir_no_looks_index = 1..102;

sir_array_type = array [sir_pd_index_type, sir_no_looks_index] of real;

start_dop_index_type = -5..4;
end_dop_index_type   = -4..5;

mti_weight_type =
  record
    No_Pulses : non_negative;
    CNI       : real;
    Index     : 1..64;
  end;

azimuth_limit_type =
  record
    Start_Az : real;
    End_Az   : real;
  end;

eccm_type =
  record
    SL_Blanking_On    : boolean;
    SL_Cancelling_ON  : boolean;
    Freq_Change       : boolean;
    Circular_Polar    : boolean;
  end;

design_data_type =
  record
    Number_Faces              : 1..4;
    Number_El_Angles          : 1..40;
    Azimuth_Limits            : azimuth_limit_type;
    Max_Range_Cells           : 1..MAX_MAX_RANGE_CELLS;
    Max_Doppler_Cells         : 1..MAX_MAX_DOPPLER_CELLS;
    Max_Pass_Dop_Cells        : 1..32;
```

```
    Peak_Power                 : real; {W}
    Boresight_Gain             : real;
    Number_of_Dwells           : 1..100;
    Coher_Bursts_Per_Dwell     : 1..16;
    Samples_Per_PRI            : non_negative;
    Number_of_PRI              : non_negative;
    PRI                        : real; { seconds }
    Sigma_Q                    : real;
    Max_Radar_Freq_Index       : non_negative;
    Nominal_Radar_Freq         : real; { HZ }
    Radar_Instant_BW           : real; { HZ }
    ECCM_Available             : eccm_type;
    Sig_To_Jam_Improve         : real;
    Polar_Rain_Gain            : real;
    Polar_Target_Loss          : real;
    MW_Loss_Trans_To_Antenna   : real;
    MW_Loss_Antenna_To_Preamp  : real;
    Signal_Processing_Loss     : real;
    Freq_Sidelobe_Level        : real;
    Antenna_Sidelobe           : real;
    Sampling_Rate              : real; {HZ}
    Nom_Target_RCS             : real;
    Nom_Target_Range           : real; {m}
    Range_Sidelobe_Level       : real;
    Noise_Figure               : real;
    Min_Instrumented_Range     : real; {m}
    Beam_Spacing               : real; {(sin(spacing))/(sin(beamwidth))}
    MTI_Loss                   : real;
    Active_Dwell_Time          : real; {sec}
    Passive_Dwell_Time         : real; {sec}
    MTI_Weights                : array[start_dop_index_type,end_dop_index_type]
                                 of mti_weight_type;
    SIR                        : sir_array_type;
end;
```

A.3 Sample Input

The file, user3:[digicomp.imrd.data_files]edm_radar_design.dat contains all the Radar Design Parameters except the records MTI_Weights and SIR. Shown below is example of the file format.

```
1       { Number_Faces }
1       { Number_El_Angles }
-22.5   { Azimuth_Limits.Start_Az }
22.5    { Azimuth_limits.End_Az }
30      { Max_Range_Cells }
16      { Max_Doppler_Cells }
32      { Max_Pass_Dop_Cells }
100e3   { Peak_Power }
31      { Boresight_Gain_dB }
40      { Number_of_Dwells }
2       { Coher_Bursts_Per_Dwell }
900     { Samples_PRI }
16      { Number_of_PRI }
3.0e-3  { PRI }
1.0     { Sigma_Q }
7       { Max_Radar_Freq_Index }
5.775e9 { Nominal_Radar_Freq }
1e6     { Radar_Instant_BW }
true    { ECCM_Available.SL_Blanking_On }
true    { ECCM_Available.SL_Cancelling_On }
true    { ECCM_Available.Freq_Change }
true    { ECCM_Available.Circular_Polar }
0.0     { Sig_To_Jam_Improve }
10      { Polar_Rain_Gain }
7       { Polar_Target_Loss }
1.5     { MW_Loss_Trans_To_Antenna }
1.5     { MW_Loss_Antenna_To_Preamp }
5       { Signal_Processing_Loss }
50      { Freq_Sidelobe_Level }
25      { Antenna_Sidelobe }
1e6     { Sampling_Rate }
10      { Nom_Target_RCS }
40e3  { Nom_Target_Range }
25      { Range_Sidelobe_Level }
4       { Noise_Figure }
2400    { Min_Instrumented_Range }
1       { Beam_Spacing }
0       { MTI_Loss }
48e-3   { Active_Dwell_Time }
48e-3   { Passive_Dwell_Time }
```

The MTI_Weights record is contained in the file, user3:[digicomp.imrd.data_files]mti_weights.dat. The table is shown below.

Start Clut Dop	End_Clut_Dop	Number Pulses	CNI	Index
0.3 PRF	0.4 PRF	4	1000	1
0.2 PRF	0.3 PRF	4	1000	2
0.2 PRF	0.4 PRF	5	1000	3
0.1 PRF	0.2 PRF	4	1000	4
0.1 PRF	0.3 PRF	5	1000	5
0.1 PRF	0.4 PRF	6	1000	6
0.0 PRF	0.0 PRF	2	1000	7
0.0 PRF	0.1 PRF	4	1000	8
0.0 PRF	0.2 PRF	5	1000	9
0.0 PRF	0.3 PRF	6	1000	10
0.0 PRF	0.4 PRF	7	1000	11
-0.1 PRF	0.0 PRF	4	1000	12
-0.1 PRF	0.1 PRF	4	1000	13
-0.1 PRF	0.2 PRF	5	1000	14
-0.1 PRF	0.3 PRF	6	1000	15
-0.1 PRF	0.4 PRF	10	1000	16
-0.2 PRF	-0.1 PRF	4	1000	17
-0.2 PRF	0.0 PRF	5	1000	18
-0.2 PRF	0.1 PRF	6	1000	19
-0.2 PRF	0.2 PRF	7	1000	20
-0.2 PRF	0.3 PRF	8	1000	21
-0.2 PRF	0.4 PRF	12	1000	22
-0.3 PRF	-0.2 PRF	4	1000	23
-0.3 PRF	-0.1 PRF	5	1000	24
-0.3 PRF	0.0 PRF	6	1000	25
-0.3 PRF	0.1 PRF	7	1000	26
-0.3 PRF	0.2 PRF	8	1000	27
-0.3 PRF	0.3 PRF	9	1000	28
-0.3 PRF	0.4 PRF	10	1000	29
-0.4 PRF	-0.3 PRF	4	1000	30
-0.4 PRF	-0.2 PRF	5	1000	31
-0.4 PRF	-0.1 PRF	6	1000	32
-0.4 PRF	0.0 PRF	7	1000	33
-0.4 PRF	0.1 PRF	8	1000	34
-0.4 PRF	0.2 PRF	12	1000	35

-0.4 PRF	0.3 PRF	16	1000	36
-0.4 PRF	0.4 PRF	16	1000	37

The SIR record is contained in the file,

`user3:[digicomp.imrd.data_files]sir_array.dat.`

An example table entry is shown below.

No Look	PD .10	.15	.20	.25	.30	.35	.40	.45	.50	.55
1	5.0	6.3	7.6	9.0	10.5	12.2	14.1	16.3	18.9	22.1
2	2.9	3.6	4.4	5.1	6.0	7.0	8.0	9.3	10.8	12.6

The table actually extends to Pds of 0.95 in increments of 0.05. There are 100 rows corresonding to 1 through 100 number of looks. Two additional row represent 300 and 1000 looks respectively. The software translates the indices accordingly.

B Radar Environment Parameters

Radar Environment Parameters describe the environment surrounding the radar. This includes thermal noise characteristics and ground and weather clutter maps.

B.1 Data Description

Radar Environment Parameters contain the following components.

Clutter_Map is an array of boolean variables which indicates where in the doppler/range Active_Dwell_Power array threshold crossings are expected if ground clutter is present. There is one array for each beam.

The columns in the array represent range samples in which 900 samples are integrated, in groups of 30, down to 30 columns. The rows represent 16 doppler bins, each separated by PRF/16 (21 HZ in the IMRD).

Weather_Map is an array of boolean variables which indicates where in the doppler/range Active_Dwell_Power array threshold crossings can be expected if weather is present.

The rows and columns have the same interpretation as for the Clutter_Map.

Pass_Freq_Thermal_Noise is an array of real values representing the root mean square (rms) thermal noise in each component of the Passive_Dwell_Freq_Power vectors.

Pass_Range_Thermal_Noise is an array of real values representing the rms thermal noise in each component of the Passive_Dwell_Range_Power vectors.

Active_Thermal_Noise is an array of real values representing the rms thermal noise in each component of the Active_Dwell_Power vector.

B.2 Pascal Declarations

The Radar Environment Parameters are stored in a Pascal structure of type radar_environment_type which is declared as follows:

```
type

  beam_type            = 1..MAX_BEAMS;
  doppler_cell_type    = 1..MAX_MAX_DOPPLER_CELLS;
  pass_dopp_cell_type  = 1..32;
  range_cell_type      = 1..MAX_MAX_RANGE_CELLS;
  thresh_crossing_type = array [doppler_cell_type,range_cell_type] of boolean;

  spectrum_type = array[doppler_cell_type, range_cell_type] of real;

  radar_environment_type =
  record
    Clutter_Map                : array [beam_type] of thresh_crossing_type;
    Weather_Map                : thresh_crossing_type;
    Pass_Freq_Thermal_Noise    : array [pass_dopp_cell_type] of real;
    Pass_Range_Thermal_Noise   : array [range_cell_type] of real;
    Active_Thermal_Noise       : spectrum_type;
  end;
```

B.3 Sample Input

Three files in the directory user3:[digicomp.imrd.baseline_3.data_files] contain all the Radar Environment Parameters. The Clutter_Map array is contained in the file edm_ground_clutter_map.dat and contains a boolean array for each beam as shown below. The map is organized as rows representing decreasing doppler indices (1 to 16 bins) and columns representing increasing range indices (1 to 30 range values).

Note that in the sample map, 'trues' occur only in the first and last rows (ground clutter is assumed to be zero mean) and only out to the 7th range cell (ground clutter is normally visible to the radar horizon). Moreover, the first range cell has been excised because of transmitter blanking at this cell location.

```
Beam Number = 1
f t t t t t t f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
```

```
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f t t t t t t f f f f f f f f f f f f f f f f f f f f f f f
```

The Weather_Map array is filled from the file mdr_data:edm_weather_map.dat and contains the boolean map as shown below. The map is organized as rows representing decreasing doppler indices (1 to 16 bins) and columns representing increasing range indices (1 to 30 range values).

Note that in this map 'trues' occur only after the 7th range cell (it was necessary to excise cells 1 through 7 to remove doppler sidelobes of close-in ground clutter). The ground clutter also made it necessary to excise three rows straddling the DC bin because of ground clutter leakage.

```
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f t t t t t t t t t t t t t t t t t t t t t t t
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
```

The thermal noise record, Pass_Freq_Thermal_Noise, Pass_Range_Thermal_Noise and Active_Dwell_Thermal_Noise records are filled from the file edm_thermal_noise.dat. Shown below are representative portions of the file showing the format used for each record. This file can be automatically generated from Beam Scan Data saved by the Adaptive Control software. The logical beam_scan_data must be assigned to the file you wish to convert. Then switch to directory, mdr_data, and run the program, convert_thermal_noise.

The Pass_Freq_Thermal_Noise data is the filled first and has the following format:

```
PASS DOPP                     PFV
INDEX
   1     3093.907227
   2      617.143372
   3      515.613281
   4      361.361664
   . . .
```

The Pass_Range_Thermal_Noise data is separated from the Pass_Freq_Thermal_Noise data by a blank line and has the following format:

```
RANGE       PRV                PRSV
INDEX
   1      112.330116
   2      112.330116
   3      112.330116
   4      112.330116
   . . .
```

The Active_Dwell_Thermal_Noise data is separated from the Pass_Range_Thermal_Noise data by a blank line and has the following format:

```
DOPPLER  RANGE      ASA        ASSA
INDEX    INDEX
 1         1     673.622131
 1         2     673.622131
 1         3     673.622131
 1         4     673.622131
 . . .
```

For more information including complete data formats see the appropriate files.

C Priority Zone Parameters

Regions under radar surveillance can be partitioned into Priority zones based on the expected importance of maintaining detection within them. The IMRD allows for five different levels of Priority Zone and is able to allocate scan time by level. The Priority Levels in decreasing order of importance are: high threat; medium threat; low threat; friendly; and unoccupied.

C.1 Data Description

Priority Zones are delimited by start azimuth, stop azimuth, near range, and far range:

Az_Start: Azimuth start angle in degrees.

Az_End: Azimuth end angle in degrees.

Near_Range: Start range zone in meters.

Far_Range: End range zone in meters.

Threat: The zone priority.

These are asserted into the Prolog database as fact priority_zone. Consistency checks are made to assure that no region is assigned more than one priority. A default level of "unoccupied" is assigned to any region that is unassigned.

C.2 Sample Input

Priority Zone data can be entered interactively online or read from a file. The file contains a single line for each zone. The default priority zone file name is user3:[digicomp.imrd.data_files]priority_zone.dat and an example is shown below.

```
priority_zone(-22.5, 22.5, 0, 5000, high_threat).
```

D Quality Options

Quality Options define a set of operating plans to be used to allocate scan time to beams in the surveillance region. There can be at most eight assigned plans with the first plan allowing the best performance and the last plan, the worst. Each plan states the desired probability of detection, the passive dwell update rate and the active dwell update rate for each of the five types of threat zone defined by the Priority Zone facts.

The desired probability of detection is stated for each Priority Zone within each plan. Normally, it will be higher for the higher Priority Zones, and will decrease for the higher plan numbers.

The passive and active update rates which specify how frequently passive and active dwells for each beam should be repeated in each plan. In the IMRD configuration the maximum passive dwell update rate and the maximum active dwell update rate is one (meaning every beam should have an passive and active detection dwell). This is because the Radar Controller code is set up to always include passive and active dwells, but a future version of the IMRD may allow these rates to vary.

The Adaptive Controller will select the first plan that can execute within the desired scan time. If no one of the plans can execute within the desired scan time, a default plan (nine) assigns all dwell time to the highest priority zone so as to achieve the highest Pd within it.

D.1 Data Description

The Quality Options Tables are constructed from the following component:

quality_option_entry: assigns the probability of detection and update rates for CPIs 1 and 2 to a priority level.

- **Pd:** the probability of detection.
- **cpi_update_rate:** a record with CPI1 and CPI2 update rates.

  There is one quality_option_entry for each Priority Zone and for each of the eight plans.

D.2 Pascal Declarations

The Quality Option Tables are stored in Pascal data structures which are declared as follows:

```
type

no_of_options_type = 1..(MAX_NO_OPTIONS + 1);

cpi_update_rate =
  record
    Passive : 0..MAX_PASSIVE_UPDATE_RATE;
    Active  : 0..MAX_ACTIVE_UPDATE_RATE;
  end;

quality_option_entry =
  record
    Pd   : real;
    Rate : cpi_update_rate;
  end;

quality_options_table_line = array [no_of_options_type] of
                             quality_option_entry;

quality_options_table =
  record
    High       : quality_options_table_line;
    Med        : quality_options_table_line;
    Low        : quality_options_table_line;
    Friendly   : quality_options_table_line;
    Unoccupied : quality_options_table_line;
  end;
```

D.3 Sample Input

The file user3:[digicomp.imrd.data_files]quality_options.dat contains the default quality option table shown below.

```
                         Probability Detection Table

                              Quality_Options
                              ---------------

Priority_Level     1     2     3     4     5     6     7     8
--------------

High threat       0.9   0.9   0.9   0.9   0.9   0.9   0.5   0.5
```

Medium threat	0.9	0.9	0.9	0.7	0.5	[illegible]	0.3	0.0
Low threat	0.9	0.9	0.7	0.5	0.5	0.3	0.0	0.0
Friendly	0.9	0.7	0.5	0.5	0.5	0.0	0.0	0.0
Unoccupied	0.9	0.5	0.5	0.3	0.0	0.0	0.0	0.0

Passive Update Rate Table

	Quality_Options							
Priority_Level	1	2	3	4	5	6	7	8
High threat	1	1	1	1	1	1	1	1
Medium threat	1	1	1	1	1	1	1	1
Low threat	1	1	1	1	1	1	1	1
Friendly	1	1	1	1	1	1	1	1
Unoccupied	1	1	1	1	1	1	1	1

Active Update Rate Table

	Quality_Options							
Priority_Level	1	2	3	4	5	6	7	8
High threat	1	1	1	1	1	1	1	1
Medium threat	1	1	1	1	1	1	1	1
Low threat	1	1	1	1	1	1	1	1
Friendly	1	1	1	1	1	1	1	1
Unoccupied	1	1	1	1	1	1	1	1

E AI Control Parameters

AI Control Parameters are a set of Prolog facts through which the user can customize the Adaptive Control software capabilities.

E.1 Data Description

AI Control Parameters contain the following components:

desired_scan_time is the maximum time desired for a single scan of the surveillance region. This is then used by the Allocate Radar Resources subproblem. The argument of this fact is a floating number.

scan_input_mode indicates whether the Adaptive Controller receives data from and sends data to the radar or the stand-alone FRDE. The allowed value for the argument is the atom "frde" or "radar."

ai_on indicates whether the Adaptive Controller will change Radar Control Parameters from scan to scan or will use the same default parameters each time. In this later case the primary purpose of the Adaptive Controller is to identify any interferences present in the scan. The allowed value for the argument is the atom "yes" or "no."

operating_mode indicates whether the program is suspended at the end of a scan automatically for user input (manual) or whether execution continues until the user requests suspension (automatic).

use_highest_quality_option indicates whether the highest quality option should always be used regardless of scan time. The allowed value for this argument is the atom "yes" or "no."

eccm_on indicates which ECCM the user desires to use. Note that only those ECCM for which the corresponding Radar Design Parameter record indicates are available will be enabled. The allowed values for this argument are the atoms frequency_hopping, sidelobe_blanking, sidelobe_cancelling and circular_polarization.

E.2 Sample Input

These Prolog facts can be entered online or from a file. The file should contain one fact per line. The default file is user3:[digicomp.imrd.data_files]ai_control.dat and is shown below.

```
operating_mode( automatic ).
scan_input_mode( radar ).
desired_scan_time( 10 ).
ai_on( y ).
use_highest_quality_option( no ).
eccm_on( frequency_hopping ).
eccm_on( sidelobe_canceller ).
eccm_on( sidelobe_blanker ).
eccm_on( circular_polarization ).
```

F AI Output Parameters

AI Output Parameters are a set of Prolog facts which allow the user to customize the type and destination of output displays.

F.1 Data Description

AI Output parameters contain the following components:

user_display is used to display data which is a final output product of the Adaptive_Controller software. The fact has a two part argument with the first atomic argument indicating the type of data and the second atomic argument indicating the data destination. The possible values of the first argument are: "interference_id", which identifies which interferences, along with its parameters, were found for every beam; "quality_option_used", which shows which quality option plan was chosen for the next scan; "ai_results", which displays the dwell time, interferences identified and eccms applied for each beam in the scan; and "radar_controls", which displays the radar controls sent to the HP2117F. The second argument can have two values, "screen" and/or "file". Note that radar_controls can only be sent to a file due to the verbose nature of this output. If file is chosen as destination, the data will go to a file whose name is the first argument followed by the file extension ".out".

diagnostic_output is used to display intermediate data and is normally used for debugging or data analysis. The atomic argument can have three values representing three data types: "beam_scan_data", the data being sent by the ST-100; "radar_parameters_array", the output of the Determine_Applicable_ECCM subproblem; and "beam_info", the data structure which combines the priority_zone and quality_options data structure. These data structures can only be sent to a file. The files name is the argument followed by the file extension ".out".

F.2 Sample Input

These facts can be entered online or using a file. Each line in the file should contain one prolog fact. The default file is user3:[digicomp.imrd.data_files]ai_output.dat and is shown below.

```
user_display( quality_option_used, screen ).
user_display( ai_results, screen ).
```

G Beam Scan Data

Beam Scan Data contains the beam by beam data used by the Adaptive Controller to perform the adaptive control process. This data is transferred for each search beam in a scan from the radar environment through the ST-100 to VAX/VMS mailboxes.

G.1 Data Description

Inputs from the ST-100 to the VAX Adaptive Controller will be in the form of Beam Scan Data, a record which consists of the following:

Frame_Count is an integer representing the frame count of the current scan. The frame count starts at 1 for the first frame and is incremented by 1 for each successive scan.

Beam_Id is an integer representing the beam within the current frame (scan). It starts at 1 for the first beam in the scan and is incremented by 1 for each successive beam.

Passive_Dwell_Range_Power is an array of real numbers whose components contain the average power level returned from the passive listening dwell for consecutive integrated range cells in each of 8 PRIs.

Passive_Dwell_Range_Var is an array of real numbers whose components contain the variabililty of the power level returned from the passive listening dwell for consecutive integrated range cells in each of 8 PRIs.

Passive_Dwell_Freq_Power is an array of real numbers whose components contain the FFT of each Passive_Dwell_Range_Power vector for each PRI, with each sample representing a block integration of 32 magnitude-detected FFT output samples.

Active_Dwell_Power is an array of real numbers whose components represent the doppler filter bank calculated over the 16 PRIs of the active listening dwell, and integrated in blocks of 30 range cells.

Active_Dwell_Var is an array of real numbers whose components contain the square of the Active_Dwell_Power array.

Sidelobe_Sum is an integer array representing the number of samples where sidelobe jamming was sensed by the sidelobe blanking circuit in the passive listening dwell for each PRI.

G.2 Pascal Declarations

The Beam Scan Data are stored in a Pascal structure of type beam_scan_data_type which is declared as follows:

```
type

  frame_type          : 1..MAX_INTEGER;
  beam_type           : 1..MAX_NO_BEAMS;
  range_cell_type     : 1.. MAX_MAX_RANGE_CELLS;
  pass_dopp_cell_type : 1..32;
  doppler_cell_type   : 1..MAX_MAX_DOPPLER_CELLS;
  no_pri              : 1..MAX_PRIS;

  range_type    : array [no_pri,range_cell_type] of real;
  freq_type     : array [no_pri,pass_dopp_cell_type] of real;
  spectrum_type : array [doppler_cell_type,range_cell_type] of real;

  beam_scan_data_type =
    record
       Frame_Count                : frame_type;  { Scan number }
       Beam_Id                    : beam_type;   { Which beam within the scan }
       Passive_Dwell_Range_Power  : range_type;
       Passive_Dwell_Range_Var    : range_type;
       Passive_Dwell_Freq_Power   : freq_type;
       Active_Dwell_Power         : spectrum_type;
       Active_Dwell_Var           : spectrum_type;
       Sidelobe_Sum               : array [no_pri] of integer;
     end;
```

H Radar Control Parameters

Radar Control Parameters contain the beam by beam data used to control the radar for the next scan. They are transfered from the VAX to the HP2117F Radar Controller via the IEEE-488 bus interface. All records are two byte words.

H.1 Data Description

Radar Control Parameters contain the following components:

Last_Beam is a boolean variable indicating whether the current beam is the last beam control record for the current scan.

Face_Index is an integer representing the radar face index for the current beam position. Within the IMRD the value is always 1, but the variable is included to allow the possibility of generalizing to other radar environments. It can assume values between 1 and 4.

Azimuth_Index is an integer indicating the azimuth angle for the current beam position. This value is represented to the HP by an integer between 0 and 9000 with 0 corresponding to -45 degrees and 9000 corresponding to +45 degrees.

Elevation_Index is an integer indicating the elevation angle for the current beam position. It can assume values between 0 and 40 degrees. Within the IMRD the value is always 0, but the variable is included to allow the possibility of controlling this variable from the rulebase.

Trans_Center_Freq is an integer variable indicating which frequency to use. The software can model up to 16 unique frequencies, however the IMRD system will only use eight. This index is then translated by the HP2117F to a real frequency value.

ECCMs_On is a record variable where each record is a boolean variable which indicates whether the given ECCM should be on. The components are:

- Sidelobe_Blanking
- Sidelobe_Cancelling
- CFAR

PRI is an integer representing the pulse repetition interval in microseconds to use for the beam position. It can assume values between 1 and 4096. Within the IMRD the value is always 3000, but the variable is included to allow the possibility of controlling this variable from the rulebase.

Pulse_Duration is an integer representing the pulse duration for the beam position. It can assume values between 0 and 40 microseconds. Within the IMRD the value is always 16, but the variable is included to allow the possibility of controlling this variable from the rulebase.

Pulse_Bandwidth is an integer representing the pulse bandwidth for the beam position. It can assume values between 0 and 10 MHz. Within the IMRD the value is always 1, but the variable is included to allow the possibility of controlling this variable from the rulebase.

TX_Polarization is an enumerated variable representing the antenna polarization to use for transmitting in the beam position. Possible values include Horizontal, Vertical, Left_Circular, and Right_Circular.

RX_Polarization is an enumerated variable representing the antenna polarization to use for receiving in the beam position. Possible values include Horizontal, Vertical, Left_Circular, and Right_Circular. Within the IMRD the value is always Horizontal, but the variable is included to allow the possibility of controlling this variable from the rulebase.

Adaptive_MTI_Weights is an integer representing which of 64 possible MTI weights sets to be used in a particular beam position. In the IMRD, 37 unique sets are used with the number 64 representing the default "all pass" weight set.

Dwell_Time_per_Look is an integer representing the PRI of CPI3. The values can range from 1 to 1000 msec.

Number_of_Looks is an integer representing the number of PRIs to be used in CPI3. The values can range from 0 (indicating the beam should be skipped) to 10000;

Passive_Dwell_On is a boolean variable indicating whether CPI 1 should be skipped. Within the IMRD the value is always true, but the variable is included to allow the possibility of controlling this variable from the rulebase.

Active_Dwell_On is a boolean variable indicating whether CPI 2 should be skipped. Within the IMRD the value is always true, but the variable is included to allow the possibility of controlling this variable from the rulebase.

H.2 Pascal Declarations

The Radar Control Parameters are stored in a Pascal data structure which is declared as follows:

```
type

face_index_type     = [WORD] 1..4;
polarization_type   = [WORD]
                      (Right_Circular, Vertical, Left_Circular, Horizontal);
azimuth_angle_type  = [WORD] 0..9000; { range from -45.0 to +45.0 degrees }
center_freq_type    = [WORD] 1..15; { One of 15 center frequencies }
pri_type            = [WORD] 1..4096; { Microsec }
pulse_duration_type = [WORD] 0..40; { Microsec }
pulse_bw_type       = [WORD] 0..10; { MHz }
adapt_mti_type      = [WORD] 1..64; { one of 64 weight sets }
dwell_time_type     = [WORD] 0..1000; { Dwell time will range from
                                         1 msec to 1 sec }
no_looks_type       = [WORD] 0..[illegible]000;

eccm_controls_type =
  record
    Sidelobe_Blanking   : [WORD] boolean;
    Sidelobe_Cancelling : [WORD] boolean;
    CFAR                : [WORD] boolean;
  end;
control_data_type =
  record
    Last_Beam             : [WORD] boolean;
    Face_Index            : face_index_type; { index of active face}
    Azimuth_Angle         : azimuth_angle_type; { index of azimuth angle}
    Elev_Index            : [WORD] non_negative; { index of elevation angle}
    Trans_Center_Freq     : center_freq_type; { center frequency in Mhz }
    ECCMs_On              : eccm_controls_type;
    PRI                   : pri_type; { Microsec }
    Pulse_Duration        : pulse_duration_type; { Microsec }
    Pulse_Bandwidth       : pulse_bw_type; { MHz }
    Tx_Polarization       : polarization_type;
    Rx_Polarization       : polarization_type;
    Adaptive_MTI_Weights  : adapt_mti_type;
    Dwell_Time_per_Look   : dwell_time_type;
```

```
    Number_of_Looks        : no_looks_type;
    Passive_Dwell_On       : [WORD] boolean;
    Active_Dwell_On        : [WORD] boolean;
  end; {control_data_type}
```

I AI Rulebases

I.1 Introduction

This appendix documents the expertise which is incorporated into the Adaptive Control software developed under the Integrated Multi-Domain Radar Demonstration, Contract #F30602-89-C-0045. This set of rules evolved during the contract and perform well in the RLSF environment, but can be expected to evolve with further system experimentation and if transported to other radar environments. This evolution is facilitated by software organization.

The Adaptive Control software resides on a VAX 8650 and performs the IMRD Adaptive Control function. It accepts the Beam Scan Data for each beam in a surveillance region and determines what control changes should be made for each beam during the next scan. At the end of each scan, Radar Control Parameters for each beam in the next scan are sent to the Radar Controller (HP2117).

Inputs to the Adaptive Controller are organized into the following data structures:

1. **Radar Design Parameters** which describe the radar configuration including such parameters as the number of beam dwell positions per scan and the available ECCM. See Appendix A.3 for a description of this data.

2. **Radar Environment Parameters** which describe the radar environment including such parameters as thermal noise characteristics and ground and weather clutter maps used to identify candidate positions for interference in the active dwell. See Appendix B.3 for a description of this data.

3. **Priority Zone Parameters** which map threat levels to specific scan regions. See Appendix C.2 for a description of this data.

4. **Quality Options** which describe quality plans to be used to allocate dwell time to beams in the next scan so as to maximize probability of target detection subject to Priority Zone constraints. See Appendix D.3 for a description of this data.

5. **AI Control Parameters** which control various modes of program operation. See Appendix E.2 for a description of this data.

6. **AI Output Parameters** which describe the types and destinations of output produced during Adaptive Controller execution. See Appendix F.2 for a description of this data.

7. **Beam Scan Data** which contains data from passive and active listening dwells. This is read and processed on a beam-by-beam basis for each beam in a surveillance region. See Appendix G.2 for a description of this data.

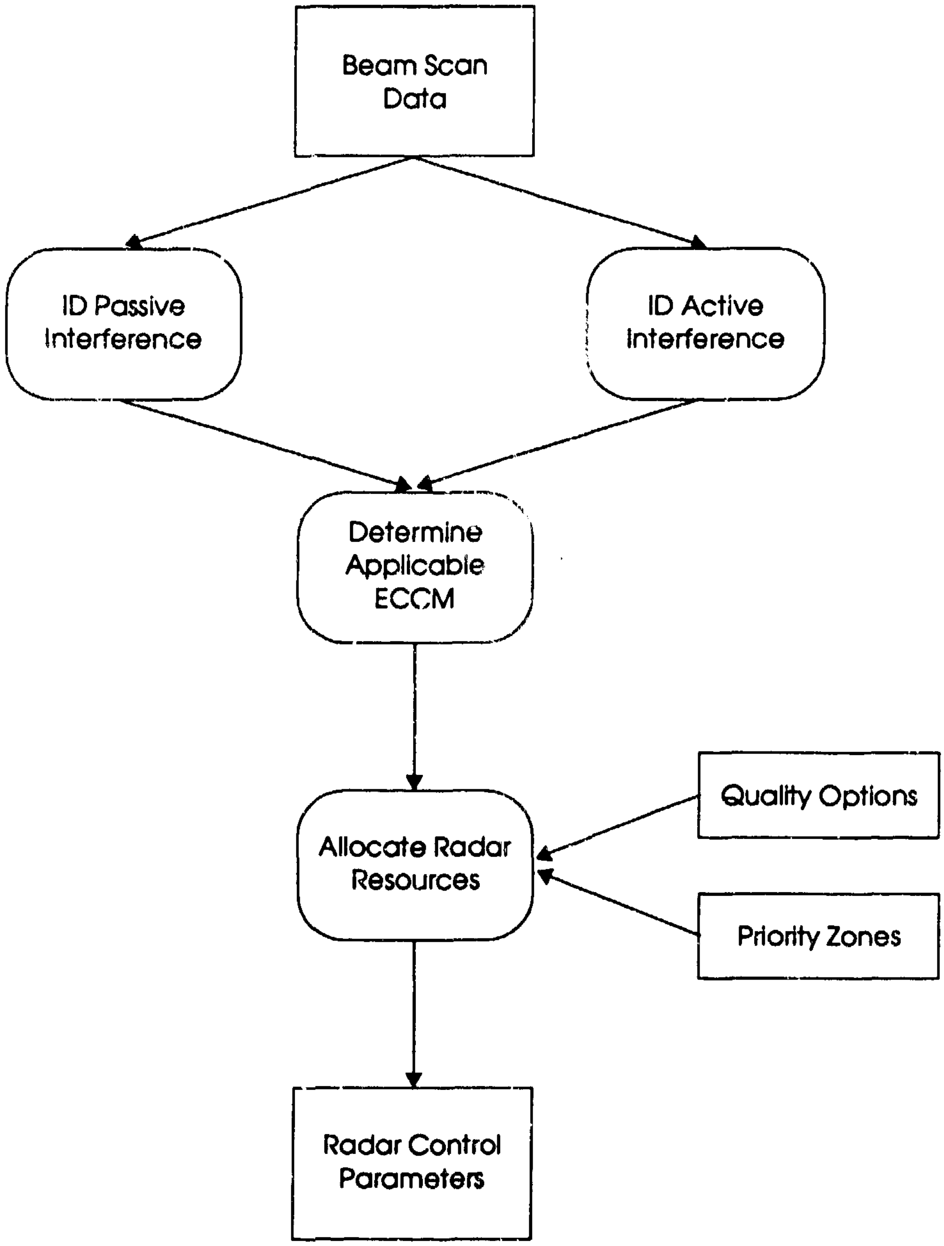

Figure I-1: AI Subsystem Decomposition

The artificial intelligence is embodied in an Adaptive Control AI Engine which is organized into the following rulebases:

1. ID Passive Interference;
2. ID Active Interference;
3. Determine Applicable ECCM; and
4. Allocate Radar Resources

as shown in Figure I-1.

ID Passive Interference computes discriminant values from the passive listening vectors of the Beam Scan Data. These are used to determine the presence of CW and pulsed jammers. If a jammer is found, the peak jammer-to-noise ratio (JNR), the range cell of the jammer, and the mainlobe duty factor are reported.

ID Active Interference computes discriminant values from the active listening arrays of the Beam Scan Data. These are used to detect weather and ground clutter. If active interference is identified, the peak clutter-to-noise ratio (CNR), the peak clutter to noise ratio at the maximum range of the clutter, the maximum range of the clutter, and the doppler extent of the clutter are reported.

Determine Applicable ECCM analyzes the interference identifications and ranks plans that counteract the interference by increasing probabilities of target detection.

Allocate Radar Resources uses the plans developed by Determine Applicable ECCM along with user-assigned Priority Zones and Quality Options to select the best operating plan that meets the user-desired scan time.

I.2 Initialization Procedures

Several initialization procedures translate system configuration inputs into values that are used directly by the AI Engine software. By calculating these values at system initialization and then saving them in global Pascal variables computation, time is saved during real-time processing of incoming Beam Scan Data.

I.2.1 Inputs

The inputs to initialization procedures include the Radar Design Parameters and Radar Environment Parameters, mentioned in Section I.1 and described in Appendices A.3 and B.3.

I.2.2 Processing

The following are calculated during system initialization.

Normalization Constants: Normalization constants are used to calculate variability of the Active_Dwell_Power array over a given map. The constants are calculated as follows:

for Range_Index = 1 to Max_Range_Cells do

$$
\begin{aligned}
i &= \text{Range\_Index} - 1 \\
R_0 &= 16\mu\text{sec} * 150\frac{\text{m}}{\mu\text{sec}} = 2400 \\
R_i &= R_0 + \frac{((\text{Range\_Index} - 1) * C * 30e - 6)}{2}
\end{aligned}
$$

The **Passive_Range_Threshold** vector is calculated by multiplying the passive range thermal noise vector by 2.

The **Active_Dwell_Threshold** array is calculated by multiplying the active dwell thermal noise array by 3.

The **Per_Pulse_SNR** is calculated from Radar Design Parameters for a user-specified target RCS and detection range.

Beam Scan Data is indexed from 1 to Number_of_Dwells as opposed to by actual azimuth angle. The **azimuth angle** corresponding to each beam is calculated and inserted into the Scan_Radar_Control data structure.

I.2.3 Outputs

The constants calculated at initialization are saved in the following global Pascal data structures:

- Norm_Const[0..Max_Range_Cells - 1]
- Passive_Range_Threshold[1..Max_Range_Cells]
- Active_Dwell_Threshold[1..Max_Doppler_Cells,1..Max_Range_Cells]
- Per_Pulse_SNR
- Scan_Radar_Controls[1..Number_of_Dwells].Azimuth_Angles

I.3 ID Passive Interference Rulebase

ID Passive Interference identifies the following categories of jammer:

- Wideband CW Mainlobe
- Wideband CW Sidelobe
- Narrowband CW Mainlobe
- Narrowband CW Sidelobe
- Wideband Asynchronous Pulsed Mainlobe
- Wideband Asynchronous Pulsed Sidelobe
- Narrowband Asynchronous Pulsed Mainlobe
- Narrowband Asynchronous Pulsed Sidelobe
- Wideband Unknown
- Narrowband Unknown

The ID Passive Interference rulebase takes the passive listening component of Beam Scan Data as input. The passive rules determine if a jammer is present and then determines its type. If a match is found, the jammer type as well as the peak JNR, the jammer range, and the mainlobe duty factor are asserted with the beam identification number as facts into the Prolog database.

I.3.1 Inputs

The specific components of Beam Scan Data that are used in ID Passive Interference are:

- **Passive_Dwell_Range_Power:** an array whose components contain the average voltage magnitude returned from the passive listening dwell for consecutive range cells in each of 8 PRIs.
- **Passive_Dwell_Range_Var:** an array whose components contain the power level returned from the passive listening dwell for consecutive range cells in each of 8 PRIs.
- **Passive_Frequency_Power:** an array whose components contain the FFT of the range samples returned from the passive listening dwell for consecutive frequency bins in each of 8 PRIs.

- **Sidelobe_Sum:** an integer array whose components represent the number of samples in the passive listening dwell where sidelobe jamming is sensed for each of 8 PRIs.

Due to transmitter leakage, the first range cell in Passive_Dwell_Range_Power and Passive_Dwell_Range_Var is excised from any determinate or parameter calculations.

The variable Pass_Range_Thermal_Noise calculated during initialization is also used.

I.3.2 Processing

I.3.2.1 Determine Jammers Present The Threshold Crossings Ratio is used to determine if a jammer is present in the current beam. It is calculated by counting the number of elements in the Passive_Dwell_Range_Power array which exceed the corresponding element in Passive_Range_Threshold vector and dividing by the total number of array elements (8 * Max_Range_Cells). It is denoted as Thresh_Cross_Ratio.

The rule to determine the presence of jammers is:

```
IF
    Thresh_Cross_Ratio > 0.0

THEN
    Jammers are present.
    Go to Identify Jammer Type Rules.

ELSE
    No jammers are present.
    Exit Id Passive Interference subproblem.
```

I.3.2.2 Identify Jammer Type The peak and average value of the Passive_Dwell_Range_Power vector are calculated for each PRI and used as inputs to a CFAR detection technique with the threshold normalized ot the average value. The maximum peak-to-average ratio, denoted as Range_Var, is used to discriminate between CW and other types of jammer.

The rule to determine the jammer type is:

```
IF
   1.0 <= Range_Var <= 2.0
```

```
THEN
   Jammer is CW type.
   Go to Identify CW Jammer.

ELSE
   Jammer is non-cw type.
   Go to Identify Non-CW Jammer.
```

Note: limit values were determined by experimentation.

I.3.2.3 Identify CW Jammer The discriminant Freq_Var, calculated for the first PRI of the passive dwell, is used to distinguish wideband and narrowband jammers. The discriminant Sidelobe_Duty_Factor is used to distinguish mainlobe and sidelobe jammers. Passive_Var is also calculated for the first PRI and can be used in future upgrades to distinguish multiple jammers in a beam.

Freq_Var is computed by finding the peak and the average value of the Passive_Dwell_Freq_Power vector for the PRI chosen and dividing the peak by the average to get the Freq_Var.

Passive_Var is computed as follows. Using the PRI chosen, compute the mean of the elements of the Passive_Dwell_Range_Var (the sum of the Passive_Dwell_Range_V... vided by Samples_Per_PRI). Divide it by the mean of the Passive_Dwell_Range... (the sum of Passive_Dwell_Range_Power divided by Samples_Per_PRI) squa...

$$\text{Passive\_Var} = \frac{\dfrac{\sum_{i=1}^{\text{Max\_Range\_Cells}} \text{Passive\_Dwell\_Range\_Var}[i]}{\text{Samples\_per\_PRI}}}{\left(\dfrac{\sum_{i=1}^{\text{Max\_Range\_Cells}} \text{Passive\_Dwell\_Range\_Power}[i]}{\text{Samples\_per\_PRI}}\right)^2}$$

Sidelobe_Duty_Factor is computed by comparing all eight Sidelobe_Sum and finding the one that occurs most frequently, within + or - 5 counts. This count is then divided by the total samples per PRI (which equals 3000 for this implementation).

Using the results of the preceding calculations, the rules to identify CW jammers are:

```
IF
   1.0 <= Freq_Var <= 3.0 AND
```

```
   0.0 <= Sidelobe_Duty_Factor <= 0.10

THEN
   Declare wideband cw mainlobe jammer.


IF
   1.0 <= Freq_Var <= 3.0 AND
   0.10 <= Sidelobe_Duty_Factor <= 1.00

THEN
   Declare wideband cw sidelobe jammer.


IF
   Freq_Var > 3.0 AND
   0.0 <= Sidelobe_Duty_Factor <= 0.10

THEN
   Declare narrowband cw mainlobe jammer.


IF
   Freq_Var > 3.0 AND
   0.10 <= Sidelobe_Duty_Factor <= 1.00

THEN
   Declare narrowband cw sidelobe jammer.
```

I.3.2.4 Identify Non-CW Jammers At this point, a jammer has been detected which is not CW. Since a given PRI may not contain the jammer pulse due to the asynchronous nature of the jammer, the software needs to determine which PRI data to use. To do this, the Mainlobe_Duty_Factor is calculated for each PRI of the Passive_Dwell_Range_Power vector.

For each PRI where the peak-to-average ratio exceeds 2.0, set the range threshold half way between the peak and average value of the Passive_Dwell_Range_Power vector. Using this threshold, count the number of threshold crossings in the Passive_Dwell_Range_Power vector for that PRI. Divide this count by Max_Range_Cells to obtain the duty factor. The PRI with the largest ratio is the PRI used for other determinate calculations. The ratio is denoted as the Mainlobe_Duty_Factor. If multiple PRIs have the same duty factor, the PRI with the pulse closest to the center of the range vector is chosen. This PRI is then used to calculate the Freq_Var and the Passive_Var discriminants as above. The Sidelobe_Duty_Factor will also be used and is calculated as in Section I.3.2.3.

The rules for identifying non-CW jammers are as follows:

```
IF
   0.01 <= Mainlobe_Duty_Factor <= 0.10 AND
   1.0 <= Freq_Var <= 3.0 AND
   0.0 <= Sidelobe_Duty_Factor <= 0.01

THEN
   Declare wideband asynchronous pulse mainlobe jammer.


IF
   0.01 <= Mainlobe_Duty_Factor <= 0.10 AND
   1.0 <= Freq_Var <= 3.0 AND
   0.01 < Sidelobe_Duty_Factor <= 0.15

THEN
   Declare wideband asynchronous pulse sidelobe jammer.


IF
   0.01 <= Mainlobe_Duty_Factor <= 0.10 AND
   Freq_Var > 3.0 AND
   0.0 <= Sidelobe_Duty_Factor <= 0.01

THEN
   Declare narrowband asynchronous pulse mainlobe jammer.


IF
   0.01 <= Mainlobe_Duty_Factor <= 0.10 AND
   Freq_Var > 3.0 AND
   0.01 < Sidelobe_Duty_Factor <= 0.15

THEN
   Declare narrowband asynchronous pulse sidelobe jammer.
```

```
IF
  1.0 <= Freq_Var <= 3.0

THEN
  Declare wideband unknown jammer.


IF
  Freq_Var > 3.0

THEN
  Declare narrowband unknown jammer.
```

Notice the redundancy in several of the rules. In particular, the last two rules ensure that even if a jammer is unable to be discriminated as CW or asynchronous pulsed, it will still be identified as wide or narrowband and the appropriate ECCM may be applied. Because of the Prolog inferencing order, the most informative identification possible will be made.

I.3.3 Outputs

When a match with one of the jammer rules is found, the declared jammer is asserted into the Prolog database along with the applicable parameters: Peak_JNR, Jammer_Range, and Mainlobe_Duty_Factor. These parameters are calculated over the same PRI used to make the discriminant calculations. The Peak_JNR is computed by finding the largest Passive_Dwell_Range_Power element and then divided by the product of the number of samples per range cell and the square of Sigma_Q. The Jammer_Range is the range bin where the Peak_JNR is found. The Mainlobe_Duty_Factor has already been calculated for pulsed jammers. For CW jammers it is set to 1.0. The facts take the following form:

```
jammer_found( Beam_Cnt, Type, Lobe, BW, Peak_JNR,
     Jammer_Range, Mainlobe_Duty_Factor ).
```

where

- **Beam_Cnt** is the beam number in which the jammer is present.

- **Type** is cw (noise), asp (async_pulsed), or unknown.
- **Lobe** is ml (mainlobe), sl (sidelobe), or unknown.
- **BW** is nb (narrowband) or wb (wideband).
- **Peak_Range_JNR** is the peak jammer-to-noise ratio.
- **Jammer_Range** is the range bin where the Peak_Range_JNR was found.
- **Mainlobe_Duty_Factor** is the ratio of the number of threshold crossings in the Passive_Range vector to the number of range bins. For CW jammers this value is set to 1.0.

I.3.4 Upgrade Options

Among the parameters that are computed, but not yet used in the rules is a Passive_Variability. This, in conjunction with the other parameters, will be used to detect multiple jammers in a single beam.

The current rules assume that a jammer is identified or not, with no provision for uncertainty. A next iteration of rules should associate a probability or certainty factor with each identification.

I.4 ID Active Interference Rulebase

ID Active Interference takes as input a set of discriminant values, which are computed for every beam, and a clutter identification rulebase. The function uses the current beam discriminants and searches the rulebase for a match. If a match is found, it declares the appropriate clutter, and computes the appropriate parameters. If no match is found then it declares that no clutter has been found.

I.4.1 Inputs

The components of Beam Scan Data that are used in ID Active Interference are:

- **Active_Dwell_Power:** an array whose components contain the results of a doppler filter bank computed over the 16 PRIs of the active listening dwell.
- **Active_Dwell_Var:** an array whose componenents contain the square of the doppler filter bank computed over the 16 PRIs of the active listening dwell.

The Clutter_Map and Weather_Map records from the Radar Environment parameters are used.

The following parameters calculated at system initialization are used:

- Norm_Const
- Active_Dwell_Threshold

I.4.2 ID Active Interference Preprocessing

The following are calculated to be used as discriminants in the ID Active Interference rulebase.

Active_Variability: Compute the sum of the elements of the Active_Dwell_Var and divide it by the sum of the elements of the Active_Dwell_Power squared, as follows, to form the Active_Variability:

$$Active\_Variability = \frac{\frac{\sum_{j=1}^{\text{Max\_Range\_Cells}} \sum_{i=1}^{\text{Max\_Doppler\_Cells}} \text{Active\_Dwell\_Var}[i,j]}{(\text{Max\_Doppler\_Cells} * \text{Max\_Range\_Cells})}}{\left(\frac{\sum_{j=1}^{\text{Max\_Range\_Cells}} \sum_{i=1}^{\text{Max\_Doppler\_Cells}} \text{Active\_Dwell\_Power}[i,j]}{(\text{Max\_Doppler\_Cells} * \text{Max\_Range\_Cells})}\right)^2}$$

Clutter Map Parameters: Four discriminants calculated using the Radar Environment Parameter, Clutter_Map. They are Ground_Power, Ground_Var, In_Ground_Match, and Out_Ground_Match.

Ground_Power: Sum the Active_Dwell_Power array of the true locations in the Clutter_Map.

Ground_Var: To compute this relative to the Clutter_Map, do the following:

1. For each element in Active_Dwell_Power which exceeds the threshold, Active_Dwell_Threshold, normalize by

   $$\frac{1}{R_i^2}$$

   where i is the range index of the element minus one.

2. Normalize the corresponding elements in Active_Dwell_Var by

   $$\frac{1}{R_i^4}$$

3. Compute Ground_Var as is done to calculate the Active_Variability above, using the normalized elements from above and dividing by the number of threshold crossings.

In_Ground_Match: Threshold the Active_Dwell_Power array and finding the number of trues that correspond to the trues of the Clutter_Map and then divide by the total number of trues in the Clutter_Map.

Out_Ground_Match: Using the thresholded Active_Dwell_power array from above, and find the number of falses that match the falses of the Clutter_Map and divide by the total number of falses in the Clutter_Map.

Weather Map Parameters Four discriminants are calculated using the Radar Environment parameter, Weather_Map. They are Weather_Power, Weather_Var, In_Weather_Match and Out_Weather_Match. They are calculated similarly to the corresponding Clutter Map parameters by substituting the Weather_Map for the Clutter_Map.

I.4.3 Rules

The following rules are used to detect ground and/or weather clutter:

```
IF
   Ground_Power is low AND
   Weather_Power is low

THEN
   Exit ID Active Interference subproblem


IF
   Ground_Power is high AND
   In_Ground_Match is high AND
   Weather_Power is high AND
   In_Weather_Match is high

THEN
   declare Weather Clutter AND
   declare Ground Clutter
```

```
IF
   Ground_Power is high AND
   In_Ground_Match is high

THEN
   declare Ground_Clutter


IF
   Weather_Power is high AND
   In_Weather_Match is high

THEN
   declare Weather_Clutter
```

I.4.4 Output

If Ground Clutter is present, compute the Peak CNR, Maxrange_Ground and the Ground_CNR_Maxrange of the cells over which the ground clutter map falls.

If Weather clutter is present, compute the Peak CNR, Min_Doppler_Index, Max_Doppler_Index, Maxrange_Weather, and Weather_CNR_Maxrange of the cells over which the weather clutter map falls.

Peak CNR is computed by finding the maximum Active_Dwell_Power entry which falls within the clutter map of interest. The corresponding Active_Dwell_Var element is then divided by the product of the number of samples per cell, the square of Sigma_Q, and Max_Doppler_Cells.

Min_Doppler_Index_WX and Max_Doppler_Index_WX are computed as follows:

1. Translate the doppler indices in the following manner:
   if Doppler_Cnt $\geq$ 9 then Doppler_Index = Doppler_Cnt - 17
   else Doppler_Index = Doppler_Cnt - 1
2. Find the minimum Doppler_Index where a threshold crossing exists in the Active_Dwell_Power and denote as Min_Doppler_Index_WX.
3. Find the maximum Doppler_Index where a threshold crossing exists in the Active_Dwell_Power and denote as Max_Doppler_Index_WX.

To calculate Maxrange and CNR_Maxrange, find the maximum range at which a threshold crossing occurs with respect to the appropriate map and designate it as the

Maxrange. Then find the largest element in this range bin in the Active_Dwell_Power array, with respect to the appropriate map. Take the corresponding element of the Active_Dwell_Var array and divide it by the product of the number of samples per range cell, square of Sigma_Q, and Max_Doppler_Cells and denote as CNR_Maxrange.

When a match is found, report the existence of the declared interference by asserting the appropriate fact into the Prolog data base as shown below:

```
ground_clutter( Beam_Cnt, Ground_Peak_CNR, Ground_CNR_Maxrange,
                Maxrange_Ground ).
```

where

- Beam_Cnt is the beam number in which the ground clutter is found.
- Ground_Peak_CNR is the peak clutter-to-noise ratio.
- Ground_CNR_Maxrange is the peak CNR at the maximum range of the ground.
- Maxrange_Ground is the largest range index at which ground interference is found.

and

```
weather( Beam_Cnt, Weather_Peak_CNR, Min_Doppler_Wx, Max_Doppler_Wx,
         Weather_CNR_Maxrange, Maxrange_Weather )
```

where

- Beam_Cnt is the beam number in which weather is found.
- Weather_Peak_CNR is the peak clutter-to-noise ratio of the weather.
- Min_Doppler_Wx is the normalized minimum doppler extent of the weather.
- Max_Doppler_Wx is the normalized maximum doppler extent of the weather.
- Weather_CNR_Maxrange is the peak CNR at the maximum range of the weather interference.
- Maxrange_Weather is the largest range index at which weather interference is found.

I.4.5 Upgrade Options

The current rules have no provision for uncertainty. A next iteration of rules should associate a probability or certainty factor with each identification.

Presently, no responsive jammers are detected. Rules could be added to enable detection of these types of jammers.

I.5 Determine Applicable ECCM

Determine Applicable ECCM is a procedural subproblem whose inputs are the detections and parameters asserted into the Prolog database by the ID Passive Interference and ID Active Interference subproblems and whose outputs are a schedule for ECCM allocated to the beams in the next scan.

I.5.1 Inputs

Inputs include:

- Jammer_found facts from the ID Passive Interference subproblem.
- Ground clutter and weather facts from ID Active Interference subproblem.
- Various Radar Design Parameters used to modify signal-to-noise ratios and target losses.

I.5.2 Processing

Processing is on a beam-by-beam basis. The following Radar Control Parameters are initialized to default parameters as specified at system initialization:

- ECCMS on
- Pulse Duration
- Pulse Bandwidth
- Transmit Polarization
- Receive Polarization
- Adaptive MTI Weights
- Dwell Time Per Look
- Number of Looks

Transmit_Center_Freq is initialized to the last value used rather than the value at system initialization.

I.5.2.1 Jammer ECCM The following jammer ECCM will be enabled as allowed by Radar Design Parameters and AI Control Parameters.

Frequency hopping is enabled if the beam contains a narrowband jammer. When frequency hopping is performed, the peak jammer-to-noise ratio of the jammer is decreased by the Freq_Sidelobe_Level Radar Design Parameter.

The **sidelobe canceller** is enabled if the beam contains a CW sidelobe jammer with a corresponding decrease in the jammer-to-noise ratio by the signal-to-jammer improvement.

The **sidelobe blanker** is turned on if the beam contains an asynchronous pulsed sidelobe jammer. The peak jammer-to-noise ratio of each such jammer is set to zero.

The composite wideband jammer-to-noise ratio is calculated by adding the square of the peak JNRs for each wideband jammer in the beam and then taking the square root of the sum.

The composite narrowband JNR is calculated by adding the square of the peak JNRs for each narrowband jammer in the beam and then taking the square root of the sum.

I.5.2.2 Clutter ECCM If polarization is allowed and weather is present, the **transmit polarization** is changed from Horizontal to Vertical. The Peak_CNR and the CNR_Maxrange are decreased by the Polarization_Rain_Gain. The Target_Loss (which is initially one) is increased by the Polarization_Target_Loss.

An **MTI weight set** is chosen so as to notch any weather and/or ground clutter spread. The Peak_CNR and CNR_Maxrange of the weather and/or ground are then decreased by the Clutter-to-Noise Improvement. The Target_Loss is increased by the MTI_Loss.

I.5.2.3 Dwell Time Calculations The Per_Pulse_SNR is divided by the target loss and then multiplied by the Pulse_Duration and the Radar_Instant_BW. Using this adjusted Per_Pulse_SNR, the signal-to-clutter ratios are calculated for any weather or ground clutter still present. Signal-to-jammer ratios are calculated using the composite JNR_NB and JNR_WB.

Dwell time per look is calculated by multiplying the number of MTI pulses by the PRI used.

The number of looks required to maintain probabilities of detection .1 through .9 is calculated for each beam. This is done by finding the number of looks to compensate for each interference type found as well as the number of looks to compensate for noise. The maximum number of looks is then used.

I.5.3 Outputs

A table of radar controls is output with each column representing increasing probability of detection. This Pascal array consists of the following records:

- Trans_Center_Freq
- Sidelobe_Blanking_On
- Sidelobe_Cancelling_On
- CFAR
- PRI
- Pulse_Duration
- Pulse_Bandwidth
- Tx_Polarization
- Rx_Polarization
- Adaptive_MTI_Weights
- Dwell_Time_per_Look
- Number_of_Looks

I.5.4 Upgrade Options

Presently several of the control parameters are not used to counteract interference. In the future new rules could be added utilizing these ECCMs and radar controls.

I.6 Allocate Radar Resources

Allocate Radar Resources is a procedural subproblem whose inputs are the outputs from the Determine_Applicable_ECCM subproblem as well as the Priority Zones, Quality Option plans and desired scan time, and whose outputs are the Radar Control parameters to be used for the next scan.

I.6.1 Inputs

The inputs to this subproblem include the table of Radar Control Parameters produced by the Determine Applicable ECCM subproblem, the Priority Zone Parameters and the Quality Option Plan. These data structures are discussed in detail in previous sections and the appropriate appendices.

I.6.2 Processing

As part of AI Control Parameters and Priority Zone Parameters, the user has input a desired scan time and has assigned priorities to the different regions of scan coverage. The Quality Option Plan assigns a desired probabilty of detection to each Priority Zone in a set of decreasingly costly plans.

The total scan time required to achieve the probabilities of detection assigned to the different priority zones under each Quality Option is computed, starting with the first quality level until a plan is reached that does not exceed the desired scan time. This is done by adding together for each beam interbeam dead time, total dwell time for CPI3 as calculated using the appropriate probability of detection, CPI1 dwell time if it is to be performed and CPI2 dwell time if it is to be performed.

Once a quality option plan has been found, any remaining dwell time is allocated across all beams in order to maximize use of the desired scan time.

If the desired scan time is exceded for each Quality Option in the schedule, a default option assigns the total available scan time to the zones with highest priority so as to achieve equal probability of detection in each.

I.6.3 Outputs

The output will be the Scan_Radar_Control array which contains Radar Control Parameters for each beam in the next scan.

I.7 Rulebase Evolution

Rulebase development is an iterative process of formulating rules and evaluating their impact. The well-designed experiment will, therefore, provide a framework within which rules can evolve.

The IMRD differs from other expert systems work in that its expertise is a combination of existing human knowledge (available at the beginning of the contract) and detailed knowledge of how the system behaves (available as the contract progressed). Thus, rulebase flexibility was critical to successful development.

Both the initial hardware/software design and the adaptive control software supported iterative development well and enabled us to model adaptive control expertise successfully. The following sections summarize our observations and findings during this process.

I.7.1 ID Passive Interference

A number of findings during ID Passive Interference development influenced the final rulebase. Among the areas that required modification were the following:

Changes in Beam Scan Data inputs: Originally, we calculated the outputs of doppler filter banks over 16 PRIs of passive dwell and 16 PRIs of active dwell. This is appropriate for the active dwell, but was modified for the passive dwell to better reflect how interference appears in the passive dwell data. We had assumed that pulsed jammers would show up as threshold crossings at specific time bins, whereas CW noise jammers would be spread across all time bins. However, pulse jammers are asynchronous to the received PRI and, therefore, are spread across the doppler filter map. This was accounted for by performing an FFT on each of the first eight PRIs and producing eight separate frequency and range vectors. The number of PRIs used in the ID Passive Interference subproblem was varied during testing. We determined that using eight gave good performance against asynchronous jammers with a very small probability of eclipsing in all PRIs.

Missing ASP jammers: Even with the enhancements described above, ASP jammers were not always detected. This was ascribed to several factors: 1) the jammer might not be transmitting during the single PRI from which data was collected, 2) a small region of the range vector is gated out by the equipment, and 3) we process only the first 900 samples of a given PRI.

To lessen the chance of missing an ASP jammer because it does not appear in the single PRI, the duty factor was calculated based on the threshold crossings for each PRI. The PRI containing the largest duty factor was then used to calculate the frequency variability, as previously described. When we tested this new discriminant calculation, we started to get false CW declarations when expecting pulsed. This occurred more frequently when we were looking at a mainlobe jammer. After analyzing the data, we concluded that the pulse jammer had a CW noise component which was raising the base thermal noise level. Therefore, a duty factor based on a fixed thresholded thermal noise would incorrectly produce a duty factor of close to one, unless the threshold is significantly raised with a corresponding decrease in sensitivity.

Wideband/narrowband discrimination: The first wideband/narrowband discriminant tested was Percent_Passive_Frequency_Crossings (the percent of cells in the passive frequency vector which exceed the passive frequency threshold) in a single PRI. The threshold was calculated by multiplying the passive frequency thermal noise vector by a constant and was set high by necessity to discriminate narrowband and wideband jammers .

Narrowband and wideband jammers were not always correctly discriminated by the Percent_Passive_Frequency_Crossings when the jammers were in the antenna sidelobes because of the high threshold setting and the low received jammer power: when the passive frequency threshold was set low, narrowband jammers were erroneously identified as wideband.

A more useful observation was that narrowband jammers had large differences in value over the different frequencies, whereas the wideband jammers had more nearly consistent values over all frequencies.

Frequency Variability–the peak frequency value divided by the average frequency value–is now used to discriminate between wideband and narrowband jamming, with wideband having the lower values.Thus, large relative differences in amplitude which may occur over only a small percentage of the frequency—rather than the percentage of the frequency spectrum in which the power level exceeds an absolute threshold—indicate narrowband jamming.

ASP/CW discrimination: An early version of the rules thresholded the Passive_Dwell_Freq_Power for all eight PRIs, calculated the duty factor for each, and chose the PRI with the highest duty factor for further calculations. We found that pulsed mainlobe jammers leaked continuous noise energy that exceeded the fixed threshold and caused ASP jammers to be identified as CW jammers. Analysis of the data indicated that a peak-to-average ratio, denoted Range_Var, of the range vector could be used to distinguish between a CW and a pulsed jammer. However, we now needed a new discriminant to determine if any jammers were present before preceding with any further jammer identification. A Range_Crossings_Ratio was calculated by taking the number of passive range vector threshold crossings in eight PRIs, using the passive range thermal noise vector multiplied by a constant as the threshold, and dividing by the total number of threshold cells in all eight PRIs. If this value is larger than zero, some kind of jammer is present and jammer identification can continue.

The Mainlobe Duty Factor was designed to discriminate between CW and ASP jammers, with CW jammers having a high value and ASP a low value. The rationale was that a jammer with more range crossings during the passive listening interval operated over a larger time frame and would be CW, whereas a pulsed jammer would appear in only a few cells.

Modifications to accommodate errors in sidelobe count: The sidelobe count is occasionally placed incorrectly in the header, causing incorrect identification of mainlobe and sidelobe jammers. To compensate for this, we compare sidelobe counts for all eight PRIs and choose the count which occurs most frequently within ± 5 counts. This count allows correct discrimination between mainlobe and sidelobe jammers.

I.7.2 ID Active Interference

Clutter maps versus clustering algorithms: Original rules used discriminants based on calculating the size and power of clusters of threshold crossings in the active dwell Beam Scan Data. Since these calculations required a lot of CPU time as well as being difficult to modify and expand, we used the different approach of predetermined boolean maps to indicate elements in the active dwell power array at which threshold crossings for a given type of interference are expected.

Ground clutter and DC offset in weather data: Initially, we used only the power determinate calculated over both the ground and the weather maps. The ground clutter map had trues in the zero doppler bin out to the seventh integrated range cell. The weather map was the inverse of the ground map. A high ground power indicated ground clutter and a high weather power indicated weather.

One observation was that ground clutter tended to spread into all doppler at close in ranges due to higher-than-expected equipment doppler sidelobe levels. This caused these returns to overwhelm weather returns. The first two range cells of the predefined weather map were blanked to eliminate ground returns; the zero and two adjoining doppler bins were notched out to excise DC offset; and additional determinate tests were added.

A test for In_Ground_Match and In_Weather_Match were added to eliminate false alarms. In order to declare ground clutter, the In_Ground_Match had to be above a certain percentage, as did weather clutter.

Low fall-rate versus distributed weather: The rulebase did not initially detect cellular weather that appears in only a few range-doppler cells. This was due in part to a collapsing loss effect of integrating over many more cells than that which encompasses the weather return. This was alleviated by lowering the thresholds and excising some of the range doppler cells as discussed in the previous paragraph.

J IMRD Test Cases

This appendix describes the tests that were performed during the IMRD final presentation. The test cases and expected results are summarized in Table 12 contained in Section 4.2.

J.1 Test 0: Benign Environment

The purpose of this test is to initialize the IMRD system and witness baseline performance under conditions of no interference. The remaining tests assume the same conditions unless noted.

J.1.1 Procedure

1. Perform software initialization procedures as specified in the User Manual, Section 3.1.
2. Set antenna elevation to +6 degrees.
3. Start the Adaptive Controller on the VAX terminal and specify the default inputs.
4. Start the software on the HP terminal. Set the antenna azimuth to 95 degrees.
5. To terminate this test type any key. This will return control to the User Interface at the end of the current scan.
6. Note the following:
   (a) the contents of the PPI display
   (b) sources of interference indicated on the AI display.

J.1.2 Expected Results

No interference should be identified for any of the beams. A sample AI display line will be similar to the following:

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.054		

The simulated uncompressed target will be present at range 5 miles.

When the user types a key at the VAX terminal, the Adaptive Controller Options Menu, as described in User Manual, Section 3.2.1, will appear at the end of the next scan.

```
                    -- OPTIONS MENU --

Select an option by entering one of the numbers:
 0 - Continue                : Continue program execution.
 1 - AI Control Parameters : Change AI control facts.
 2 - Priority Zone           : Change Priority Zone Definitions.
 3 - Quality Options         : Change Quality Options Table.
 4 - AI Outputs              : Change amount of interscan output.
 5 - Quit                    : Quit the IMRD program.
Please enter number of menu selection. ->
```

At this time, the user can change items in any of the listed categories.

J.2 Test 1: Ground Clutter

The purpose of this test is to verify that the IMRD software is capable of identifying ground clutter and applying appropriate adaptive control, MTI weights.

J.2.1 Procedure

1. Set antenna elevation to +.3 degrees.
2. Use the Adaptive Controller Options Menu to set the ai_on parameter to "no". display
3. Continue the Adaptive Controller, observing the PPI and AI displays.
4. Use the Adaptive Controller Options Menu to set the ai_on parameter to "yes".
5. Terminate the test by striking any key.
6. Note the following:

   (a) the contents of the PPI display

   (b) sources of interference indicated on the AI display

   (c) AI decisions indicated on the AI display.

J.2.2 Expected Results

During the scans with ai_on set to "no", ground clutter will appear in selected beams on the UPA-62 PPI. Targets (simulated and targets of opportunity) will be masked by ground clutter. The AI display will indicate that ground clutter has been detected in some or all of the beams, but that no adaptive control actions will be taken. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.054	ground	

During the first scan with ai_on set to "yes", ground clutter will appear on the UPA-62 PPI. The AI display will indicate that ground clutter has been detected, and that MTI will be used to counteract the ground in those beams where it is detected. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.060	ground	mti_weight changed

During subsequent scans the MTI weights will be applied to CPI3, ground clutter will be less visible on the PPI. The simulated target will be present at a range of 5 miles.

J.2.3 Expected Results if Weather is Present

If there is weather present in the scan region during this or any subsequent test, the AI display will indicate that weather, as well as ground clutter, has been detected. Sample display lines with ai_on set to "no" are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.054	weather ground	

When ai_on is set to "yes", the AI display will indicate that polarization has changed and the MTI is applied to those beams containing weather interference. The MTI weights will be turned on to null the doppler spectrum of the weather and ground sensed during the ID Active Interference process. If the Adaptive Controller deems that polarization is by itself sufficient to reduce the weather, the MTI will not be enabled. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.060	weather ground	tx_polarization changed mti_weight changed

During subsequent scans, transmit polarization will be set to vertical and the MTI weights will be applied to CPI3. The weather clutter will be less visible on the PPI and targets should become more discernible.

J.3 Test 2: WB ASP Jammer

The purpose of this test is to verify that the IMRD software is capable of identifying a wideband asynchronous pulsed jammer and applying the appropriate adaptive control, sidelobe blanking.

J.3.1 Procedure

1. Turn on the wideband pulsed jammer via the HP2117.
2. Use the Adaptive Controller Options Menu to set the ai_on parameter to "no".
3. Continue the Adaptive Controller, observing the PPI and AI displays.
4. Use the Adaptive Controller Options Menu to set the ai_on parameter to "yes".
5. Terminate the test by striking any key.
6. Note the following:
   (a) the contents of the PPI display
   (b) sources of interference indicated on the AI display
   (c) AI decisions indicated on the AI display.

J.3.2 Expected Results

During the scans with ai_on set to "no", the jammer will appear in the mainlobe and sidelobes on the UPA-62 PPI and targets (simulated and targets of opportunity) will be masked by jammer interference.

The AI display will indicate that a wideband pulsed jammer has been detected. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.054	wb_asp_sl_jammer	
.			
.			
20	0.054	wb_asp_ml_jammer	
.			

During the first scan with ai_on set to "yes", the jammer will appear in the mainlobe and sidelobes on the UPA-62 PPI.

The AI display will indicate that a wideband pulsed jammer has been detected and that sidelobe blanking is enabled for those beams containing a pulsed sidelobe jammer.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.054	wb_asp_sl_jammer	sidelobe_blanking on
.			
.			
20	0.054	wb_asp_ml_jammer	
.			

During subsequent scans after ai_on has been set to "yes", there will be no evidence of the jammer in the sidelobes on the PPI although the mainlobe jammer will still be visible. Targets will appear on the PPI.

J.4 Test 3: WB CW Jammer

The purpose of this test is to verify that the IMRD software is capable of identifying a wideband CW jammer and applying appropriate adaptive control, sidelobe cancelling. It should be noted that presently no sidelobe canceller is available. The Radar Design

Parameters are initialized with the cancellation factor for the sidelobe canceller se to 0 dB. Therefore, although the display indicates sidelobe cancelling will be enabled, no sidelobe cancelling is actually performed.

J.4.1 Procedure

1. Turn on the wideband CW jammer via the HP2117.
2. Continue the Adaptive Controller, observing the PPI and AI displays.
3. Terminate the test by striking any key.
4. Note the following:
   (a) the contents of the PPI display
   (b) sources of interference indicated on the AI display
   (c) AI decisions indicated on the AI display.

J.4.2 Expected Results

During each scan the jammer will appear in the mainlobe and sidelobes on the UPA-62 PPI. Targets will be masked by the jammer interferences.

The AI display will indicate that a wideband CW jammer has been detected and that the sidelobe canceller is enabled for all beams containing a CW sidelobe jammer. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.054	wb_cw_sl_jammer	sidelobe_cancelling on
.			
.			
20	0.054	wb_cw_ml_jammer	
.			

Since there is no sidelobe canceller available and the jammer-to-noise ratio will not be decreased. No improvement will occur on the PPI other than due to dwell time changes.

J.5 Test 4: NB CW Jammer and Ground Clutter

The purpose of this test is to verify that the IMRD software is capable of identifying a narrowband CW jammer and applying appropriate adaptive control which is to change frequency. Ground clutter will appear after the jammer is removed. It will be identified by the Adaptive Controller and removed by applying MTI weights.

J.5.1 Procedure

1. Turn on the narrowband CW jammer via the HP2117.
2. Use the Adaptive Controller Options Menu to set the ai_on ai_control parameter to "no".
3. Continue the Adaptive Controller, observing the PPI and AI displays.
4. Use the Adaptive Controller Options Menu to set the ai_on ai_control parameter to "yes".
5. Instruct the IMRD to continue.
6. Terminate the test by striking any key.
7. Note the following:
   (a) the contents of the PPI display
   (b) sources of interference indicated on the AI display
   (c) AI decisions indicated on the AI display.

J.5.2 Expected Results

During the scans with ai_on set to "no", the jammer will appear in the mainlobe and sidelobes on the UPA-62 PPI. Targets will be masked by the jammer interferences.

The AI display will indicate that a narrowband CW jammer has been detected. Sample display lines are shown below.

```
Beam    Dwell Time    Interferences         ECCMs Changed
         per Beam         Found

 1        0.054       nb_cw_sl_jammer
 .
 .
20        0.054       nb_cw_ml_jammer
 .
```

During the the first scan with ai_on set to "yes", the jammer will appear in the mainlobe and sidelobes on the UPA-62 PPI over all ranges. Targets will remain masked by the jammer interferences.

The AI display will indicate that a narrowband CW jammer has been detected and that a frequency change is enabled for every beam containing a narrowband jammer declaration. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.054	nb_cw_sl_jammer	frequency changed
.			
.			
20	0.054	nb_cw_ml_jammer	frequency changed
.			

During the next scan, there will be no evidence of the jammer on the PPI and ground clutter will appear in some, or all, of the beams. MTI will be selected to counter that ground clutter where it appears. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.060	ground	mti_weight chang
.			

During subsequent scans the MTI weights will be applied to CPI3 and ground clutter will be less visible on the PPI. The simulated target and targets of opportunity will appear on the PPI during this and any subsequent scans.

J.6 Test 5: Two NB CW Jammers and Ground Clutter

The purpose of this test is to verify that the IMRD software is capable of identifying two narrowband CW jammers and applying appropriate adaptive control which is to change frequency.

J.6.1 Procedure

1. Turn on two narrowband CW jammers via the HP2117.
2. Repeat steps 2 through 8 of Test 4.

J.6.2 Expected Results

The results should be the same as for Test 4 with the exception that a mainlobe declaration may not appear. A frequency change should remove both jammers.

J.7 Test 6: NB ASP Jammer and Ground Clutter

The purpose of this test is to verify that the IMRD software is capable of identifying a narrowband pulsed jammer and applying appropriate adaptive control, which is to change frequency. Ground clutter will appear after the jammer is removed. It will be identified by the Adaptive Controller and removed by applying MTI weights.

J.7.1 Procedure

1. Turn on the narrowband pulsed jammer via the HP2117.
2. Use the Adaptive Controller Options Menu to set the ai_on parameter to "no". display
3. Continue the Adaptive Controller, observing the PPI and AI displays.
4. Use the Adaptive Controller Options Menu to set the ai_on parameter to "yes".
5. Terminate the test by striking any key.
6. Note the following:
   (a) the contents of the PPI display
   (b) sources of interference indicated on the AI display
   (c) AI decisions indicated on the AI display.

J.7.2 Expected Results

During the scans with ai_on set to "no", the jammer will appear in the mainlobe and sidelobes on the UPA-62 PPI. Targets will be masked by jammer interferences.

The AI display will indicate that a narrowband pulsed jammer has been detected. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed

```
1        0.054        nb_asp_sl_jammer
.
.
20       0.054        nb_asp_ml_jammer
.
```

During the first scan with ai_on set to "yes", the jammer will appear in the mainlobe and sidelobes on the UFA-62 PPI. Targets will remain masked by the jammer interferences.

The AI display will indicate that a narrowband pulsed jammer has been detected and that frequency hopping is enabled for those beams where narrowband jamming occurs. Sample display lines are shown below.

```
Beam    Dwell Time      Interferences         ECCMs Changed
         per Beam          Found

1        0.054        nb_asp_sl_jammer      frequency changed
.
.
20       0.054        nb_asp_ml_jammer      frequency changed
.
```

During the next scan ground clutter will appear on the PPI.

The AI display will indicate that ground clutter has been detected and that MTI weights will be enabled for those beams containing ground clutter during the next scan. Sample display lines are shown below.

```
Beam    Dwell Time      Interferences         ECCMs Changed
         per Beam          Found

1        0.060            ground            mti_weight changed
.
```

During subsequent scans the MTI weights will be applied to CPI3 and ground will be less visible on the PPI.

Targets will appear on the PPI during this and subsequent scans.

J.8 Test 7: NB ASP Jammer, Frequency Change Disabled

The purpose of this test is to verify that the IMRD software is capable of identifying a narrowband pulsed jammer and applying the appripriate ECCM. In this case, however, the most appropriate adaptive control, frequency hopping, is unavailable and the next best choice, sidelobe cancelling will be selected. The test will be conducted with ai_on set to "yes".

J.8.1 Procedure

1. Turn on the narrowband pulsed jammer via the HP2117.
2. Use the Adaptive Controller Options Menu to remove frequency_hopping as an available ECCM. This is an AI Control parameters.
3. Continue the Adaptive Controller, observing the PPI and AI displays.
4. Terminate the test by striking any key.
5. Note the following:
   (a) the contents of the PPI display
   (b) sources of interference indicated on the AI display
   (c) AI decisions indicated on the AI display.

J.8.2 Expected Results

During the first scan the jammer will appear in the mainlobe and sidelobes on the UPA-62 PPI. Targets will be masked by jammer interferences.

The AI display will indicate that a narrowband pulsed jammer has been detected and that sidelobe blanking is enabled for those beams in which pulsed sidelobe jamming occurs. Sample display lines are shown below.

Beam	Dwell Time per Beam	Interferences Found	ECCMs Changed
1	0.400	nb_asp_sl_jammer	sidelobe_blanking on
.			
.			
20	0.400	nb_asp_ml_jammer	
.			

During subequent scans ground clutter will appear on the PPI and will be countered as in Test 1.

J.9 Test 8: Quality Options

The purpose of this test is to verify that the IMRD software is capable of incorporating user-specified Priority Zones and Quality Options into decisions for allocating dwell time.

J.9.1 Procedure

1. Use the Adaptive Controller Options Menu to edit the Priority Zones table, producing the following:
   (a) a high threat region in beams 1-8 (azimuth angles -22.5 to -13.5)
   (b) a medium threat region in beams 9-16 (azimuth angles -13.5 to -4.5)
   (c) a low threat region in beams 17-24 (azimuth angles -4.5 to 4.5)
   (d) a friendly region in beams 25-32 (azimuth angles 4.5 to 13.5)
   (e) an unoccupied region in beams 33-40 (azimuth angles 13.5 to 22.5)

   See Figure J-1
2. Use the Adaptive Controller Options Menu to examine the contents of the Quality Options table.
3. Continue the Adaptive Controller, observing the PPI and AI displays.
4. Turn on a wideband CW jammer at 20 dB attenuation via the HP2117.
5. Change the jammer to 0 dB attenuation and continue.
6. Terminate the test by striking any key.
7. Note the following:
   (a) the contents of the PPI display
   (b) sources of interference indicated on the AI display
   (c) the quality option selected for each scan

J.9.2 Expected Results

With the jammer off, the signal-to-interference ratio is sufficient that all regions can reach the highest probability of detection. The best Quality Option (1) is selected and dwell time is allocated equally over all beams.

While the first jammer (medium jammer power) is operating there will be insufficent time to acheive the highest probability of detection in all zones. Thus the higher threat zones will receive more energy than the lower ones.

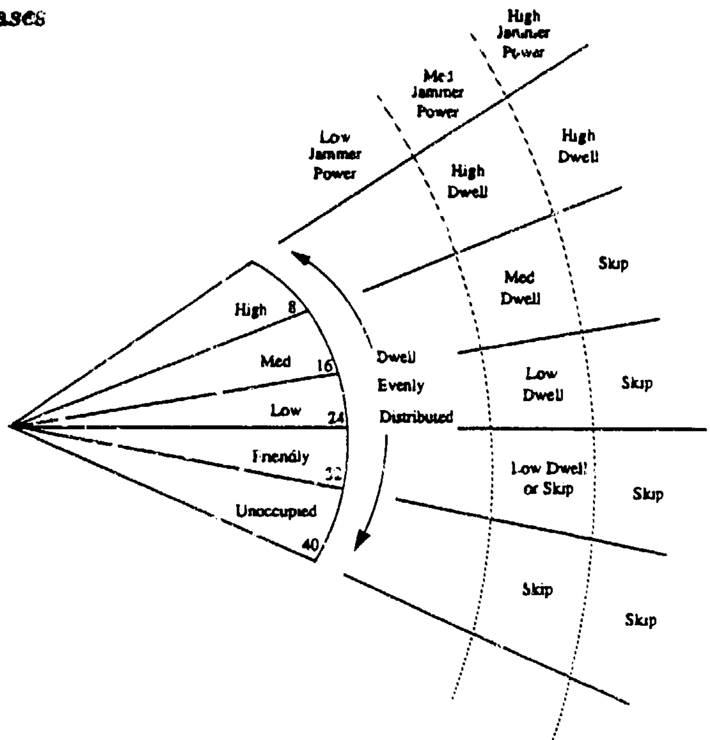

Figure J-1: Priority Zones

While the second jammer (high power) is operating, the lowest Quality Option (9) is most likely to be selected, allocating all dwell time to the highest priority region, beams 1 through 8.

References

[ACM] Kennedy, Peter, and Blasi, Wayne, *Adaptive Controller for Multi-Domain Sensor Processors (ACMDSP)*, RADC Contract No. F30602-85-C-0204, September, 1988.

[IR] *Interim Report for Integrated Multi-Domain Radar Demonstration*, RADC Contract No. F30602-89-C-0045, August 31, 1989.

[AP] *Array Processor Control Language (APCL) Programmer's Guide*, Star Technologies, Inc.

[APX] *Array Processor Executive (APX) User's Guide*, Star Technologies, Inc.

[MAX] *MAX Software Modules Working Paper*, Technology Services Corporation

[TP] *Test Plan for Integrated Multi-Domain Radar Demonstration*, Contract No. F30602-89-0045, December 28, 1989.

[UM] *User Manual for Integrated Multi-Domain Radar Demonstration*, Contract Number F30602-89-C-0045, April 30, 1991.

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.