Signal Processing Evaluation, Analysis, and Research (SPEAR) Testbed at AFRL/RRS

Jeffrey E. Tyler, William Baldygo, Robert Bozek, Todd Cushman, Mark Novak, Walter Szczepanski

Abstract—Over the past year much interest has been given to developing new algorithms and strategies to approach radar signal processing using a priori or acquired knowledge. To meet the demands for the ongoing interest in evaluation of these new techniques, AFRL/DARPA has developed a scalable Signal Processing Evaluation, Analysis & Research (SPEAR) testbed to effectively analyze any new algorithms. SPEAR is an end-to-end analysis facility with functionality ranging from target and data generation to the tracking performance evaluation. This provides a controlled environment for both the sensor signal processing and information exploitation communities to simulate effects of knowledge-aided technologies on both the signal processing and tracking performance. In order to facilitate accurate and efficient algorithm transition to this evaluation testbed, an Interface Control Document (ICD), allowing algorithm developers to tailor their algorithms to this resource, has been developed.

***Index Terms*—KASSPER, Knowledge-Aided Signal Processing, SPEAR, Algorithm evaluation, KASSPER**

I. Introduction

The Signal Processing Evaluation, Analysis & Research Facility at AFRL, Rome Research Site has been established to provide a stable, unbiased testbed upon which algorithm developers can evaluate algorithm performance. The performance of signal processing algorithms has a direct impact on the ability of the tracker to initiate and maintain tracks. Therefore, a primary goal of SPEAR is to provide all the necessary tools, algorithms, and data sets to evaluate radar signal processing architectures spanning data generation to tracking output.

Manuscript received May 9, 2003. This work was supported by USAF/AFRL Rome Research Site and DARPA under contract F30602-01-C-0146.

Jeffrey Tyler is a Software Engineer with Black River Systems Co., Utica, NY 13502 USA (email: tyler@brsc.com).

William. Badldygo, Jr., is the Technical Advisor for AFRL/SNRT at Rome Research Site, Rome, NY 13440 (e-mail: William.Baldygo@rl.af.mil).

Robert E. Bozek is an Electrical Engineer with Black River Systems Co., Utica, NY 13502 USA (email: bozek@brsc.com).

Todd Cushman. is an Electrical Engineer with AFRL/SNRT at Rome Research Site, Rome, NY 13440 (e-mail: Todd.Cushman@rl.af.mil).

Mark Novak is an Electrical Engineer with AFRL/SNRT at Rome Research Site, Rome, NY 13440 USA (email: Mark.Novak@rl.af.mil).

Walter Szczepanski is the Chief Scientist with Black River Systems Co., Utica, NY 13502 (email: szczepanski@brsc.com).

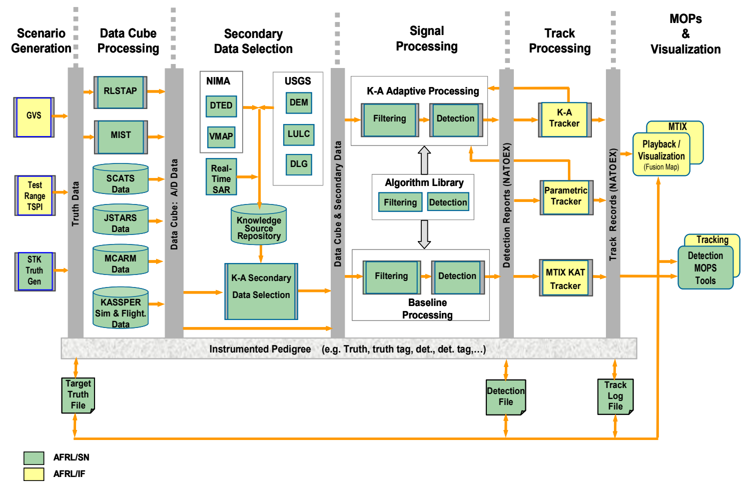

Figure 1. SPEAR Processing Architecture

The SPEAR processing architecture, shown in Figure 1, consists of six components. The first component is the set of scenario generation tools that provide the ability to create target lay downs using Ground Vehicle Simulator (GVS) and platform trajectories using the Satellite Toolkit (STK). Also, available in this component is measured Time-Space-Position Information (TSPI) data. This data is the input to the next component, data cube processing. Both simulated and measured data cubes are available. Currently data cubes can be generated in SPEAR using Multi-INT Simulation Technology (MIST) or Research Laboratory Space-Time Adaptive Processing (RL-STAP). Simulated datasets, including those provided as part of Knowledge-Aided Sensor Signal Processing with Expert Reasoning (KASSPER), are cataloged for use on the SPEAR testbed. In addition, Joint STARS data is available on the classified testbed. Other datasets will be made available as they are incorporated.

The secondary data selection component includes algorithms for intelligent data selection, as well as the databases and imagery to accompany it. This is the input into the filtering and detection component which draws from a library of algorithms, both knowledge-aided and conventional.

After the signal processing, the tracking components consists currently of the Parametric Tracker (AFRL developed) and the Kinematic Auto Tracker (Northrop-Grumman developed). The outputs of these trackers provide the viewable impact of the signal processing algorithms for the War Fighter. This information is then passed to the visualization tools and Measures of Performance (MOPS). A large set of detection and tracking MOPS are currently available for use in these evaluations.

This paper will discuss the general software design and operation of SPEAR, as well as explain how to write and evaluate algorithms using the SPEAR testbed. The emphasis in Section II will be to introduce the software functionality of the SPEAR facility.

Section III will discuss the operation of the SPEAR evaluation tools. This includes the user interface and configurations of the server software.

In Section IV, the programming interface will be described. The emphasis will be on making it easy to integrate algorithms into the existing testbed.

Section V describes the philosophy behind the end-to-end evaluations and the importance of tracking outputs to the War Fighter.

II. Software Architecture

A. Design and Functionality

SPEAR is a complex testbed and, therefore, requires control software that can provide the functionality that is needed to optimize the evaluation process. The software allows users, on their PC's, to connect to the SPEAR testbed and set up and evaluate processing experiments utilizing a user-friendly interface. The intention is to verify signal processing algorithm performance individually and to evaluate several algorithms in parallel.

The functionality of the SPEAR software (Figure 2) is designed using the Client-Server paradigm. This means that there are two parts to this software: the client, where the user interfaces with the program, and the server, where the actual processing takes place. For SPEAR, these pieces are on separate machines, with the server residing on remote hosts.

The SPEAR Graphical User Interface (GUI) provides a convenient method of generating configurations of test runs. This GUI accepts parameters and passes them to the eXtensible Markup Language (XML) generation process which is implemented as a MATLAB mex DLL. This process generates an XML configuration file and sends it to the node that was chosen by the user.

The SPEAR server software authenticates the incoming connection and passes the data through to the receiver process. Once the configuration file is received, the user and experiment are registered in the session database. This database manages all user sessions, thereby, making it easy for users to maintain a session based on a session ID. The user is able to bring up all sessions from their display and view progress on any of them. This data will be stored in a database and updated continually.

An active messaging service handles all communications between the server and the client software and makes it possible to query the server for information at any time during processing.

Once the session has begun and the configuration file is placed on the queue, the file is parsed and the MATLAB processing begins. Once the processing completes, all outputs are sent to the client via the message service. The report

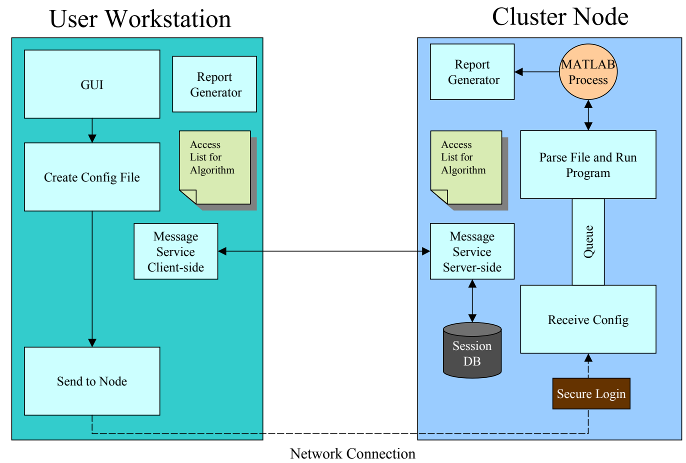

Figure 2. SPEAR Software Architecture

generator creates XML reports containing both numerical and graphical results. These reports are easily transformed into PDF using XSLT. This report generator resides on both the server and client nodes.

The SPEAR software provides an interactive environment for algorithm developers who desire meaningful feedback on the performance of their algorithms. This allows for manual intervention during experiments since the feedback can provide the developer with the ability to use the data derived knowledge to impact future processing methods and parameters.

Another feature of this processing environment is the multi-node operation of the cluster. This computation model allows SPEAR facility users to evaluate several processing lineups in parallel to increase evaluation turnaround time. For example, this permits evaluations of different secondary data selection techniques with different processing chains.

B. Security

Security is an important factor when dealing with proprietary intellectual property, as well as dealing with classified material. Therefore, there are multiple security layers built into SPEAR to protect both classified data and intellectual property rights.

File-level security has been implemented in SPEAR in order to protect source code and executables. This is the standard and straight-forward method of securing the files. Access to source code may be granted on a case by case basis.

Password authentication will be required both to put configurations onto the server queue and to retrieve results from the server. This provides checkpoints where only authorized users can access appropriate material. Standard encryption methods will be used.

A third layer of security is at the algorithm access level where protection of intellectual property is the primary concern. Only authorized users are permitted access. This protection will be at the request of the contractor pending approval from the KASSPER program manager.

In addition to the software level security, the DoD classification guidelines will be strictly enforced in the SPEAR classified laboratory. All data and results that fall under the classification rules will be treated in the appropriate manner. There will be use of classified data sets and databases during the evaluation of algorithms, however, access will be strictly limited on a need-to-know basis.

C. *Performance Measures*

The SPEAR facility provides a non-biased platform to compare algorithm performance. The basis of these comparisons is a set of performance metrics available as a component of the testbed itself. These metrics include detection MOPS (Measures Of Performance) developed by Black River Systems Co. / AFRL-SNRT and tracking MOPS provided by the AFRL-IFEA Fusion Laboratory.

A graphical tool called "SPEAR Performance Analysis Toolkit," shown in Figure 3, is accessible through the SPEAR Graphical User Interface. The GUI has options to input the detection and tracking log files, as well as the name for the NATOEX file to be used. NATOEX is a GMTI data format widely used in trackers and visualization tools. It organizes the detection or track data in an easily useable format.

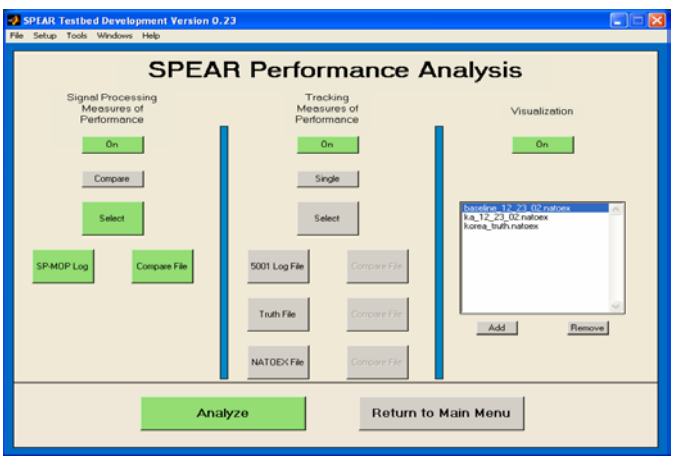

Figure 3. SPEAR Performance Analysis GUI

Analysis is reasonably quick and outputs are available primarily in MATLAB figures. Numerical results are logged. These results are processed through the report generator into the XML files that will be archived. eXtensible Style-Sheet Language Transforms (XSLT) is used to convert the reports to Hyper Text Markup Language (HTML) or Portable Document Format (PDF) depending on the need.

Detection and signal processing MOPS include means for measuring clutter suppression and target detection. Also, metrics specific to knowledge-aided algorithms, such as knowledge value, will be added as they are developed. An example of a signal processing MOP is shown in Figure 4.

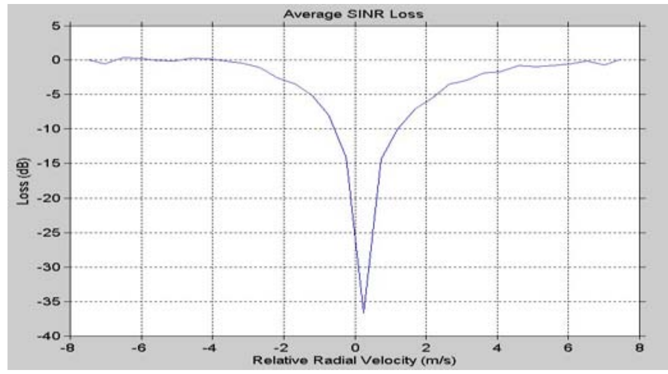

Figure 4. Average SINR Signal Processing MOPS Plot

Tracking MOPS measure the performance of the radar tracker. MOPS such as track continuity, track identity lifetime, target identity lifetime, and many more are provided. This ability to analyze performance through the tracker provides a thorough understanding of the true radar performance.

III. OPERATIONS

SPEAR operation is a simplified process which is controlled by the SPEAR GUI. All information about the processing node choices is included on the GUI setup screen. Specific menu access is on a need-to-know basis. Some algorithms will be grayed out if the proper access rights do not exist for the current user. For example, in the Figure 5, "Range-Variable CSI" is not available to the current user or is not usable with the data set chosen. This option could not be selected.

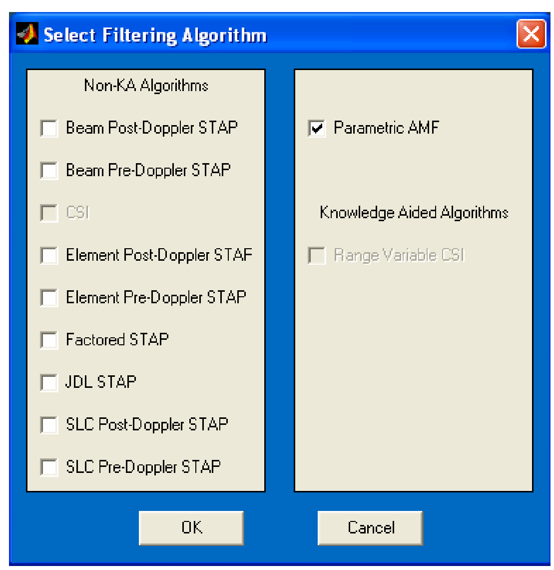

Figure 5. Filtering Algorithm Selection

The GUI (Figure 6) provides an overall summary of the configuration parameters so that all parameters can be viewed before the file is sent to the processing node. The user selects the number of CPI's to be processed, which allows for the interactive environment previously mentioned. The user can process one CPI at a time, manually alter parameters, or act as a man in the loop.

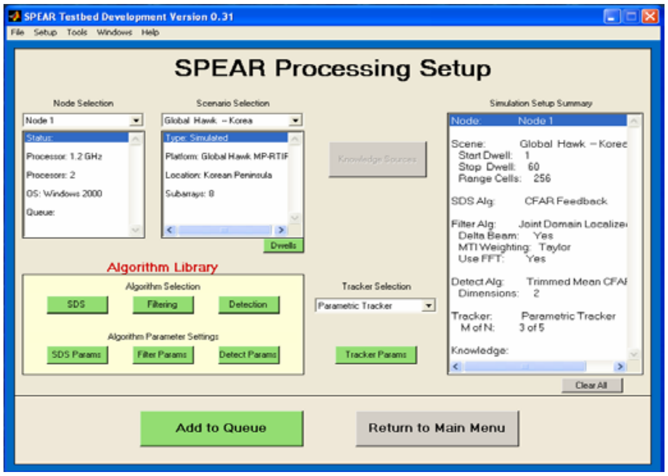

Figure 6. SPEAR Processing GUI

The GUI creates XML configuration files that are readable by any node on any system that has access to the schema. This allows for maximal interoperability since XML support exists on both the Windows and Linux/Unix platforms. In fact, the use of a standard technology avoids the common pitfall of proprietary formats and complex parsing structures that become a hassle for third party developers.

The STAP library consists of the interference suppression algorithms currently available at AFRL. There are approximately 9 algorithms currently available for use on the SPEAR testbed. There are also a limited number of intelligent secondary data selection techniques available in STAP.

Currently, only one tracker is available on SPEAR when using the graphical configuration but there are plans to add support for several additional trackers. The initial step includes streaming output to the Northrop-Grumman KAT tracker as part of the MTIX workstation.

IV. Interface for Developers

Algorithm developers often have very specific parameter sets that define their algorithms' interfaces. A SPEAR Interface Control Document (ICD) that defines how to map these parameters onto the existing SPEAR structures is being developed. Algorithms can be delivered to SPEAR written in either C (or derivative), Java, or MATLAB.

Databases containing both knowledge sources and data sets will be available. The query interface is accessible through MATLAB and is currently being defined. Land areas are implemented in a common coordinate system and some basic transforms will be made available in order to ease integration with different data sets. DTED, Land Use Land Cover (LULC), Digital Elevation Model (DEM), and others are already being incorporated into the SPEAR facility with the plans to add more databases as they are needed and as they become available. The classified processing environment will support finer resolution data of a classified nature.

In addition to the geographical and cultural databases, there is a common access point for data sets. The data is stored in its respective format and may not be transformed because much of the measured data would be more difficult to use. There are several data sets available that were generated using GVS, STK, and MIST. These simulated data sets reflect regions of current interest and the conditions are configurable to specific terrain and clutter types to examine for a particular experiment. One such foreign scenario used in our Global Hawk experiments was generated using the tools available in the SPEAR facility.

The preferable delivery from algorithm developers is a package containing not only the code, but also the data set and databases used. It is also requested that they provide test input data and test results and graphics to allow for easier validation at the SPEAR facility.

Currently for the Secondary Data Selection and STAP algorithms, the input is the IQ data in Range x Pulse x Channel format with a parameters structure containing all the parameters for processing. Also, if a knowledge-aided algorithm uses feedback, feedback requirements should be indicated. This will be more clearly defined in the ICD that will be released shortly.

V. Evaluations

A. Hardware

The SPEAR facility consists of two separate processing environments: classified and unclassified. Both environments have similar configurations but different data sets and databases. The unclassified environment, depicted in Figure 7, is accessible from client machines outside the SPEAR facility. Access is restricted to only those that have the need to access it. The classified environment is restricted to particular

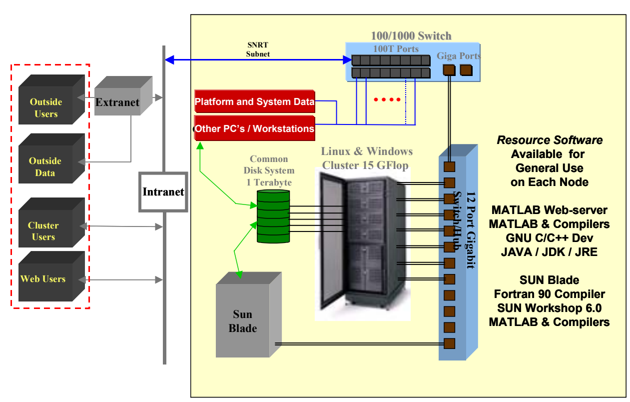

Figure 7. SPEAR Facility Hardware Configuration

clients located within the SPEAR laboratory. The classified nodes also have access to SIPRNet which allows for the transfer of data to other classified hosts.

Both environments have SUN and PC hardware with 100MB switches for network connectivity. The PC's utilize both Windows and Linux environments which have individual advantages that are beneficial to the SPEAR testbed.

B. Effect on tracking performance

Signal processing performance is not only a factor of how well it performs its particular task but of how its output impacts results downstream. This is a common misconception that leads to trouble downstream in the processing chain. Poor signal processing typically leads to inadequate tracking. This means radar signal processor performance should be evaluated, not just by signal detection metrics, but also should be judged based on the tracking results it achieves. The operator and/or War Fighter do not see the STAP algorithm output, however they see the tracking output. This means that the detection reports that are sent to the tracker directly affect the action taken by the operators.

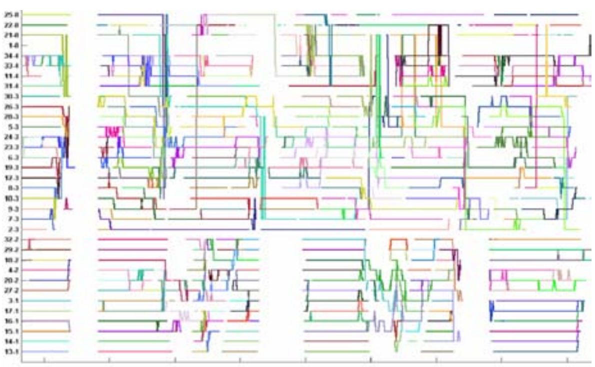

Figure 8. Tracking Waterfall Plot

A waterfall plot, like the one in Figure 8, can be very enlightening, highlighting poor tracking performance resulting when detection consistency is poor. The numbers on the vertical axis are target numbers and the horizontal axis is time. Each colored line represents a track. When a signal processing system misses detections the tracker may lose or switch tracks as can be seen above.

C. Impact on War Fighter

The KASSPER program is primarily concerned with the effect of knowledge-aided radar processing and its direct impact on the mission-level effectiveness of the War Fighter. The detection and tracking performance results should be analyzed from this perspective since the goal of surveillance radar is to increase situational awareness in a region of interest.

End-to-end evaluations are not common since many algorithms are developed with little concern for total system performance. The algorithms in SPEAR will be evaluated, not just on their signal processing performance, but also on system and mission level effectiveness.

VI. CONCLUSIONS

The SPEAR facility is nearing completion and much of the functionality is currently in place. The laboratory is prepared to begin receiving algorithms as the infrastructure exists and will provide support for the addition of new algorithms. This facility provides an end-to-end evaluation environment that will objectively evaluate knowledge-aided signal processing algorithms. This effort will generate results on the effectiveness of knowledge-aided radar architectures and their effect on the War Fighter.

REFERENCES

[1] Bozek, R. et al., "Using SPEAR to Evaluate Knowledge-Aided Algorithms," Proceedings of the 2nd Annual KASSPER Workshop 14-16 April 2003.