Autonomous Littoral Warfare Systems Evaluator – Monte Carlo (ALWSE-MC)

J. Marc Eadie
NAVSEA – Coastal Systems Station
Panama City, Florida (www.ncsc.navy.mil)
eadiejm@ncsc.navy.mil

ABSTRACT

Performing surveillance, reconnaissance, mapping, reacquisition, identification, and neutralization of mines in the littoral environments is a very complex and difficult problem. The natural environment of this area is extremely dynamic and energetic, creating a formidable challenge for any appointed system.

Evaluating system performance and interoperability are also very difficult in this environment. System evaluation is further complicated by the availability and flexibility of autonomous platforms, resulting in thousands of available concept permutations addressing various missions in the littorals. Evaluating the effectiveness of these concepts in an orderly, structured, and consistent way requires the use of a simulation-based analysis approach.

A fast, easy-to-learn, and capable toolset has been developed to provide an operationally meaningful evaluation of the performance of these concepts. The Autonomous Littoral Warfare Systems Evaluator – Monte Carlo (ALWSE-MC, pronounced "always") is a system level simulation that provides analysis support for autonomous systems used in the littoral regions. ALWSE allows the characterization of many different classes of autonomous vehicles, including underwater (UUV), ground (UGV), surface (USV), and airborne (UAV) systems. The tool provides user-friendly facilities related to establishing sensor-to-target relationships, defining vehicle mobility, and enabling search/avoidance algorithms, as well as many pre/post-processing features. In addition, non-tactical components of the system may be evaluated using ALWSE's navigational and communication evaluation facilities.

ALWSE allows the developer or analyst to easily play "what-if" games, or perform relative comparisons related to autonomous system configuration and performance. The analyst can focus on a true definition of "better" in terms of mission performance, thereby more effectively directing the technology or system development. All tradeoffs are done strictly in the virtual conceptual domain, requiring very little effort to modify and re-evaluate, a contrast to costly bench/field/sea tests.

INTRODUCTION

This paper will serve as an introductory overview for the Autonomous Littoral Warfare Systems Evaluator – Monte Carlo (ALWSE-MC) simulation tool. It will present background information in the problem area, the design considerations of the simulation tool, a brief description of features, and usage examples.

ALWSE-MC is being developed in support of the Very Shallow Water / Surf Zone Mine Countermeasures (VSW/SZ MCM) Reconnaissance Program, and is located at NAVSEA/Coastal Systems Station. The program sponsor is the Office of Naval Research (ONR) Code 321OE, Dr. Tom Swean.

PROBLEM STATEMENT

Ensuring the timely and reliable delivery of resources in support of new operational concepts such as Operational Maneuver From the Sea (OMFTS) and Ship to Objective Maneuver (STOM) while reducing the risk to personnel and materiel is a primary goal of the current VSW/SZ technology development programs. These technologies will carry the responsibility of providing an operationally safe path from seaward areas to the on-shore objective(s). See Figures 1 and 2. Maneuvering and staging areas/lanes of passage must be verifiably cleared of threats that may impede the flow of personnel and resources. Current methods and technologies employed are quite time consuming and impose significant risk levels. New and innovative technologies are under development and will be employed to both increase operational capability and reduce operational risk.

Performing the required missions of surveillance, reconnaissance, mapping, reacquisition, identification, and neutralization of threats in the littoral environments is a very complex and difficult problem. The natural environment of this area is extremely dynamic and energetic, creating a formidable challenge for any appointed system.

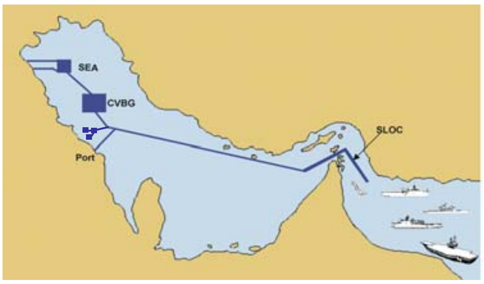

Figure 1. Reference Mission Example

A more formidable problem is that of *assessing* the ongoing technology development in such an environment. Determining and establishing limits on operational requirements and system performance parameters require an in-depth knowledge of both the performance of developmental technology and the characteristics of the environment. Along with the ever-increasing sub-system interoperability requirements and system complexities, system evaluation is further complicated by the availability and flexibility of autonomous platforms, giving thousands of available concept permutations addressing various missions in the littorals.

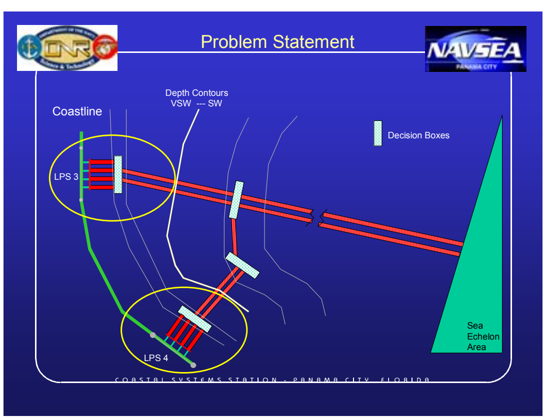

Figure 2. Mission Geometry Example

Technology solutions involved may be notional, without available hardware solutions or detailed engineering-level models. In this case, the engineer or analyst must have the ability to assess solutions of this type as well via a generalized system or component characterization.

Due to the complexity and variability of the problem domain, evaluating the effectiveness of these concepts in an orderly, structured, and consistent way requires the use of a simulation-based analysis approach.

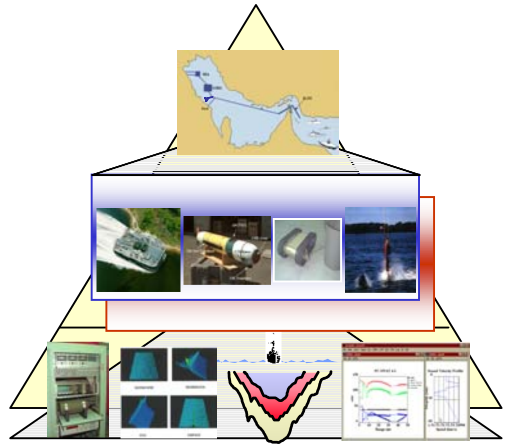

Figure 3. Modeling & Simulation (M&S) Hierarchy

In order to effectively and consistently evaluate enabling technologies, a tightly integrated set of modeling and simulation (M&S) tools is required. Tools, ranging from the highly abstract "war game"-type to the physics-based, high-resolution models, must be viewed as part of the overall toolset, as defined by the pyramid-type architecture of Figure 3. These tools must work together to support all aspects of system and technology development. M&S tools at all levels must have the appropriate mix of programmatic and interface integration, depending on the user requirement. The goal is a seamless characterization of system(s) and technologies at all levels of analysis.

TOOL BACKGROUND

As such, a fast, easy-to-learn, and capable evaluation tool has been developed to provide an operationally meaningful evaluation of the performance of these concepts. The Autonomous Littoral Warfare Systems Evaluator – Monte Carlo (ALWSE-MC, pronounced "*always*") is a system level simulation that provides analysis support for autonomous systems used in the littoral regions. ALWSE-MC was developed to provide simulation support to the "mid-pyramid" level of analysis by focusing on generalized performance of VSW/SZ recon systems.

ALWSE-MC allows the developer or analyst to easily play "what-if" games, or perform relative comparisons related to autonomous system configuration and performance. The analyst can focus on a true definition of "better" in terms of mission performance, thereby more effectively directing the technology or system development. All tradeoffs are done strictly in the virtual conceptual domain, requiring very little effort to modify and re-evaluate, a contrast to costly bench/field/sea tests.

ARCHITECTURE

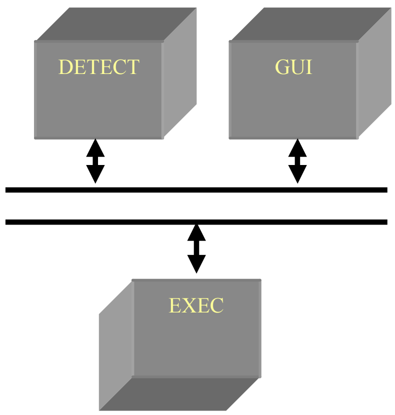

Figure 4. ALWSE-MC Architecture

ALWSE-MC is designed as a multi-threaded application capable of segmenting runtime tasks over multiple processes. The functionality of the tool is broken into several segments, each providing a specific service, as indicated in Figure 4. Each module is then dynamically allocated, as hardware resources are available, providing increased execution speed. Object motion, navigation, terrain, and simulation timing are handled by the <EXEC.C> component. Dynamic interactions of all participants, through sensor interactions or collisions, are included in the <DETECT.C> module. The <GUI> module handles the graphical user interface (GUI) services including scenario setup, threat and system configuration, and data collection. Data post-processing services are provided by well-defined interfaces to the external commercial off-the-shelf (COTS) applications, Microsoft Excel® and Mathworks MATLAB®. All GUI-related services are written in Visual Basic. The remainder of the tool's services are written in C.

ALWSE-MC uses shared memory regions to enable select process communications. Critical simulation data is stored in a well-defined shared memory structure and is used for simulation control, data passing, etc. In addition to the popular locally shared memory, ALWSE-MC has the capability to allocate select facilities over multiple computer platforms using Systran's SCRAMNet® shared memory. SCRAMNet® is a hardware-based shared memory system linking compute platforms via high-speed, fiber optic links.

ALWSE-MC was designed to be used by the analyst, at the desktop, and will run on any of the Windows® operating systems (95/98/Me/NT/2000). Extensive installation services, modeled after commercial software practices, provide automated installation. Hardware requirements are minimal and include: Pentium-class processor, 32MB RAM, and approximately 5MB of hard disk space for program installation. The software is available via an installation CD that guides the user through the installation process. A user's guide is also available, although many help functions are embedded in the program.

SCENARIO GENERATION

ALWSE-MC was designed to be an analyst's tool, not requiring frequent visits or contact with tool developers, allowing simulation scenarios and studies to be created easily. The tool provides user-friendly facilities related to establishing sensor-to-target relationships, defining vehicle mobility, and enabling search/avoidance algorithms, as well as many pre/post-processing features. All of the scenario definition information is stored in well-defined files. These files may be recalled at any time to re-produce simulation runs. There are four basic definition files:

1. Sensor-Target Database (*.db)
2. Threat Laydown (*.lay)
3. Vehicle Laydown (*.veh)
4. Scenario File (*.set)

The overarching definition file is the scenario file (*.set). This files contains all information related to GUI field data and links to the associated database and laydown files. The role and content of these files will be better understood after the specific editors are described.

The basic process by which a scenario is created is simple. First, the analyst storyboards (or outlines) both the threat elements and the countering systems that will be involved. Relationships between each threat and system are then characterized. The systems characteristics are entered, along with definitions of program search patterns, and the data collection options are set. The runs are executed and the data is examined via the post-processor.

Sensor-Target Database

This editor handles the definition of several key parameters (Figure 5 depicts example database elements):

1. Available type and size of threat
2. Available type of sensor systems
3. Available types of counter-countermeasures (CCMs)
4. Sensor-to-Target relationships

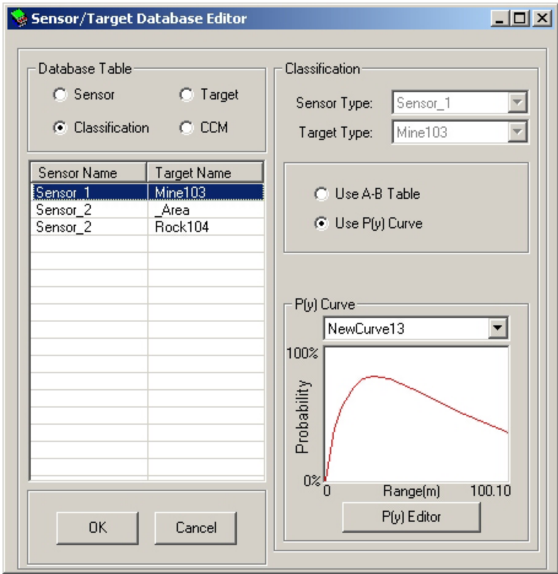

Figure 5. ALWSE-MC Database Editor

The sensor-target database editor is used to manipulate the data associated with all threat, counter-countermeasures (CCM), and sensor components of the simulation. Threats are characterized by their name, physical size, and representative icon. Sensor systems are defined by name and localization error. CCMs are defined by their ability to either prevent or delay the systems from performing their tasks.

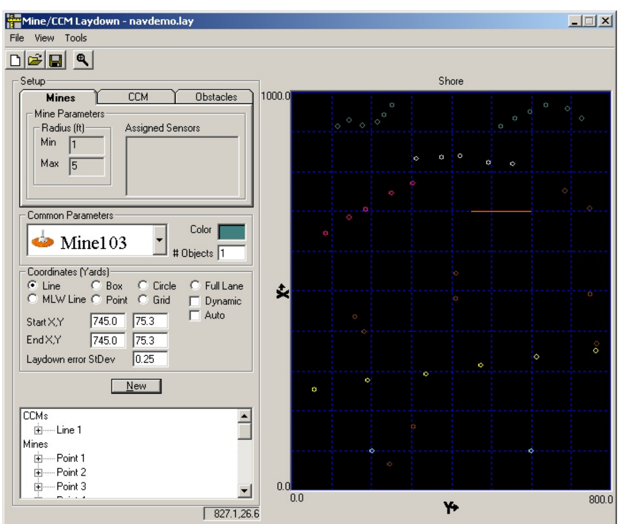

Figure 6. ALWSE-MC Threat Editor

Finally, the sensor-to-target database defines the sensor's ability to detect and/or classify various types of targets. ALWSE-MC is parametric in nature, and thus does not require detailed, physics-based characterizations of the natural environment or systems under test. Effects on sensor performance are modeled in the tool as "A-B" tables or P(y) ("P-of-Y") curves, one for each sensor-target pair.

Threat Laydown Editor

Once the types of sensors and threats are defined, the user may then proceed to define the study-specific threat configuration using the threat laydown editor. Figure 6 shows an example threat laydown session. This editor allows the user to easily define the types and locations of the threats to play in the scenario, including threat objects, CCMs, and obstacles. Threat lines, regions, and points are quickly defined. Threat configurations are stored in the `*.lay` file format and may be stored and used again as required. In addition, ALWSE-MC allows the creation of laydown files independently of the scenario generation. This enables the user(s) to "play" red against blue teams, restricting the flow of information (or intelligence) related to threat location and makeup.

Vehicle Laydown Editor

ALWSE-MC's vehicle laydown editor allows the user to characterize the systems that will counter the threat previously defined in the threat laydown editor (Figure 7).

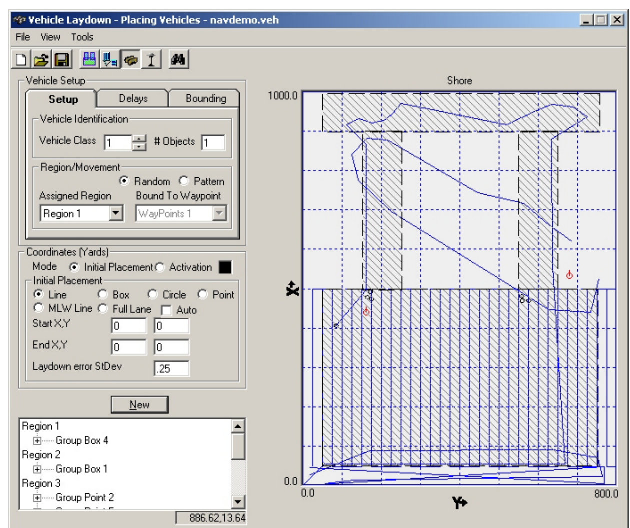

Figure 7. ALWSE-MC Vehicle Editor

ALWSE-MC allows the characterization of many (currently set to 10) different classes of autonomous vehicles, including underwater (UUV), ground (UGV), surface (USV), and airborne (UAV) systems. ALWSE-MC is not limited to a single instance of each class but can track any number of each class type. In addition, non-tactical components of the system may be evaluated using ALWSE-MC's navigational and communication evaluation facilities.

System mobility is entered as speeds, turn rates/radii, endurance, search patterns, sensor processing delays, navigation, and communication capabilities.

System Console

The ALWSE-MC console is treated as the main menu of the application and is the first menu presented to the user after ALWLSE-MC is started (Figure 8). The console allows the user to specify the target and vehicle laydown filenames and locations, the end run conditions, as well as numerous data collection and batch mode operations.

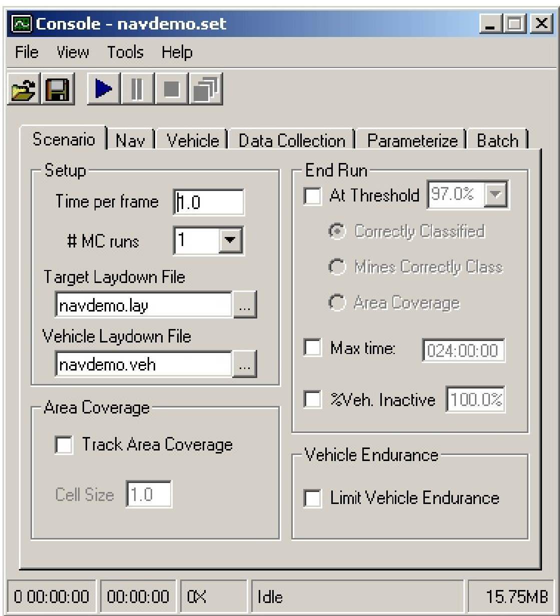

Figure 8. ALWSE-MC Console

The scenario file (`*.set`) referenced earlier saves all the information entered into the console and serves as a single point of definition for the scenario.

The console includes parameterization facilities that allow the automated, programmatic control of simulation parameters. Using the *tags* property of Visual Basic fields, the user can vary the range and step size of appropriate simulation parameters.

Options for default values, simulation execution speed, and display options are also available via the console menu.

Post-Processor Facilities

Depending on the scenario setup and the types of data collected, ALWSE-MC provides a convenient facility for handling the potentially large amounts of data produced during simulation execution. Figure shows the data analysis user interface. This data collection format is specifically designed to work seamlessly with the MATLAB® analysis software package. ALWSE-MC will assemble and format the data automatically and will provide the user with several default options for viewing. MATLAB® is executed automatically, by the ALWSE-MC application, and produces the plots selected in the Data Analysis post-processor.

Types of MATLAB® data displays include area coverage, target localization, participant tracks, ground truth information, et al.

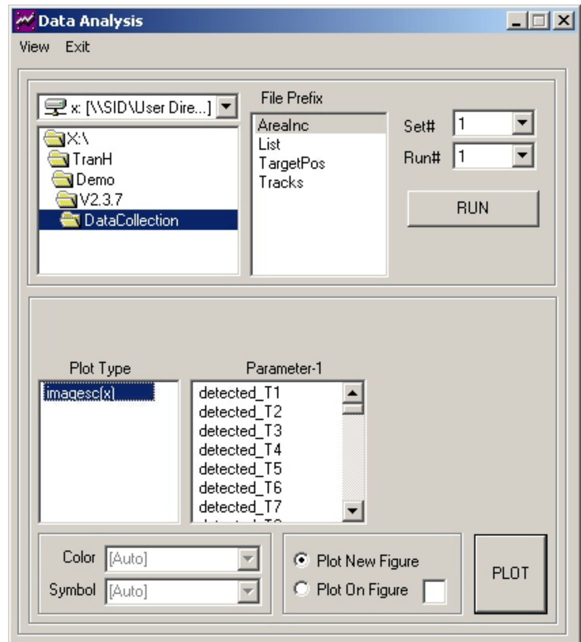

Figure 9. ALWSE-MC Post-Processor

In addition, ALWSE-MC is capable of writing directly to Excel®-format files for further analysis.

A native simulation display included with ALWSE allows the user to track the progress of a given run. A 3-dimensional display capability (the Real-time ALWSE Visualization Environment, or RAVE) is also available, but requires the user to obtain additional toolsets and licenses from third party sources.

TOOL USAGE

ALWSE-MC is used in a wide variety of simulation-based analyses. To date, its primary use is to examine the effect of various system performance options on the paramount Measure Of Effectiveness (MOE), *Time*. It has been used in the VSW/SZ MCM Reconnaissance program to aid in the determination of which system parameters will provide the maximum operational benefit (What configuration is “better”?). This would include setting the envelope for sensor capabilities and resolution, determining the requirements for system navigational accuracy, identifying the by-product implications of logistical constraints, and comparing various established system alternatives.

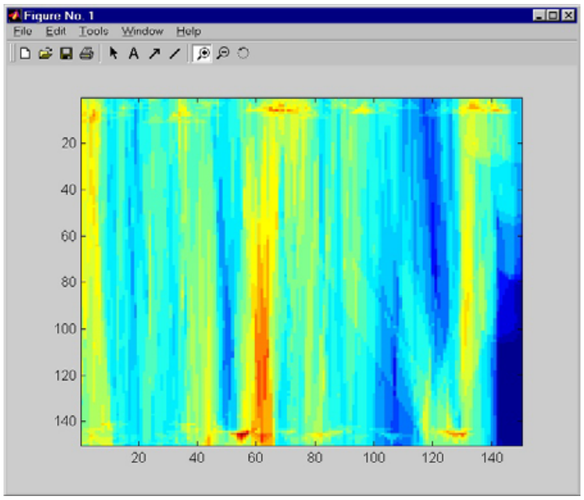

Figure 10. ALWSE-MC Area Coverage

ALWSE-MC is also used in parameter / error sensitivity analysis, “twisting the knobs” on system performance parameters to gauge the effect on overall effectiveness. Performance characteristics are then grouped into levels of influence on system Measures of Performance (MOPs) and MOEs, which then allow the mission and systems analysts to make better choices as to the application of developmental resources. Figure 10 shows the area coverage runs of a vehicle exercising a pattern. The red areas indicate increased coverage instances; blue areas indicate potential coverage holidays. Figure 11 shows a trade study between sensor performance and time to classify area threats.

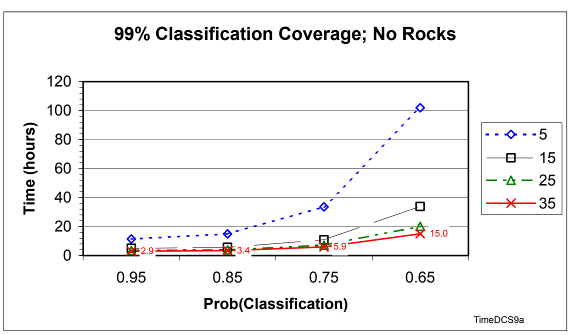

Figure 11. ALWSE-MC Parameter Trade Study

In support of the SZ technology programs, ALWSE-MC was employed to focus test requirements and plans for the CRAWLEX series of land tests. Using estimates of vehicle performance and initial test objectives, ALWSE-MC was able to predict time and resource usage and enabled the systems engineers to refine the scheduled testing accordingly.

SUMMARY

The necessity of an organized, complete simulation-based analysis approach in the assessment of new and novel technologies in the VSW/SZ regions is critical. Energetic, dynamic environmental conditions coupled with the increasing complexities of these small unmanned systems require that engineers and analysts have access to verified, validated, repeatable, and flexible simulation analysis tools in order to provide the warfighter with the appropriate mix of enabling technologies.