Intelligent sensor networks differ from traditional sensor networks in several important ways:

1. Intelligent sensor networks use advanced processing, including machine learning, to develop an understanding of the scene they are observing and the actors within the scene.
2. Intelligent sensor networks are adaptive, capable of using their scene understanding to change their focus, following and monitoring information deemed more important, including real-time information on target movement since the time of initial observation.
3. The network’s information distribution strategy, modifying the dissemination structure, frequency, and routing strategy to assure that information is timely, relevant, and does not overload users.

Intelligent sensor networks are adaptive, capable of using their scene understanding to change the network’s information-processing methods, adapting the frequency, node responsibility, and methods used to extract useful knowledge from data, and selecting from half a dozen vignette production algorithms.

Cost-effective hardware-in-the-loop testing of intelligent systems requires a live-virtual-constructive test infrastructure that models the information gathering, flow, and use for friendly and adversarial forces.

In this partial data set, we explore the use of alternate methods to define tests: Monte Carlo methods produce random samples that can be used for statistical analyses; T-wise testing maximizes coverage of the state space; genetic algorithms are effective at identifying boundary conditions and allow us to characterize the conditions in which critical properties do, or do not, hold; and criticality testing produces test sets that allow us to prove key capabilities will always hold.

Criticality-based testing produces an ordered list of tests starting with the most "critical." By testing in critically ordered, we can define more precise areas of ignorance and make the most efficient use of available time and resources to improve the state of the art.

In complex, constantly changing scenes there exists an optimal organizational structure and strategy for exchanging and processing information used to understand the operating environment.

Testing of Autonomy in Complex Environments (TACE) is a distributed live-virtual-constructive test infrastructure capable of testing complex interactive engagements in which both friendly and adversarial forces observe, communicate, and react realistically in complex, interactive engagements.

TACE’s unique live-virtual-constructive infrastructure models complex cognitive tasks and live-virtual-constructive interactions on the ground over a Test and Training Enabling Architecture (TENA) bus. Abstract interactions are sent to TACE hardware co-located with the system under test (SUT) over a real-time network where high-fidelity synthetic images are produced from the abstract models and injected into the sensor's data stream.

This work was funded by TRMC under contract W900KK-13-C-0036-P00006 and ONR under contract N00013-01-0155.


This work was funded by TRMC under contract W900KK-13-C-0036-P00006 and ONR under contract N00013-01-0155.

Live-Virtual-Constructive Test and Evaluation of Intelligent Sensor Networks

David Scheidt

When testing an intelligent sensor network, it is inadequate to simply measure whether the sensor network did or did not detect or track the target. When measuring an intelligent sensor network, we need to understand, in a battlefield sense, who knew what when, what they did with the information, and how that decision impacted the engagement.

Accurate testing of intelligent sensor networks requires a test infrastructure that accurately models the use of the information provided by the sensor network. To do this, the test infrastructure must incorporate C2 and decision processes of sensor network users and adversaries alike.

In complex, constantly changing scenes there exists an optimal organizational structure and strategy for exchanging and processing information used to understand the operating environment.

In this partial data set, we explore the use of alternate methods to define tests: Monte Carlo methods produce random samples that can be used for statistical analyses; T-wise testing maximizes coverage of the state space; genetic algorithms are effective at identifying boundary conditions and allow us to characterize the conditions in which critical properties do, or do not, hold; and criticality testing produces test sets that allow us to prove key capabilities will always hold.

Criticality-based testing produces an ordered list of tests starting with the most "critical." By testing in critically ordered, we can define more precise areas of ignorance and make the most efficient use of available time and resources to improve the state of the art.

In complex, constantly changing scenes there exists an optimal organizational structure and strategy for exchanging and processing information used to understand the operating environment.

In this partial data set, we explore the use of alternate methods to define tests: Monte Carlo methods produce random samples that can be used for statistical analyses; T-wise testing maximizes coverage of the state space; genetic algorithms are effective at identifying boundary conditions and allow us to characterize the conditions in which critical properties do, or do not, hold; and criticality testing produces test sets that allow us to prove key capabilities will always hold.