Closed-Loop Collaborative Intelligence, Surveillance, and Reconnaissance Resource Management

Andrew J. Newman and Jonathan T. DeSena

Intelligence, surveillance, and reconnaissance (ISR) encompasses activities related to planning and operating sensors and systems that collect, process, exploit, and disseminate data in support of military operations. As the number and diversity of sensing assets continues to expand, human operators are less able to effectively manage, control, and exploit the ISR ensemble. Automated support for processing sensor data and controlling sensor assets can relieve the burden on human operators, particularly in dynamic environments, where it is essential to react quickly to information. Our approach is to apply principles of feedback control to ISR operations, "closing the loop" from sensor collections through automated processing to ISR asset control. Closed-loop collaborative ISR (CLCISR) is a feedback process that continually reallocates ISR resources to respond to changing conditions, maximize the relevance of data collected, and reduce errors and uncertainty about a tactical commander's situation of interest. APL has developed a CLCISR prototype that dynamically tasks a diverse ensemble of ISR platforms and sensors in a closed feedback loop with an upstream data fusion process that combines information to produce an accurate and current tactical picture. This article introduces the CLCISR concept and details the primary technical elements, applications, and APL's current research directions.

INTRODUCTION

The term "intelligence, surveillance, and reconnaissance" (ISR) encompasses a variety of activities related to planning and operating sensors and related systems that collect, process, exploit, and disseminate data in support of military operations. ISR data can take many forms, including electronic signals, message traffic from various modes of communication, a wide variety of still and motion imagery [e.g., panchromatic, multispectral, hyperspectral, infrared, wide-area motion imagery, and full-motion video (FMV)], and a wide variety of additional measurements and signals. These data can come from a variety of sources including satellites, manned and unmanned aircraft, aerostats, ground-based and sea-based collection systems, and human intelligence sources. Some unmanned aircraft are as large as jet fighters; others are as small as radio-controlled model airplanes. ISR activities provide critical support across the full range of military operations and provide information to battlefield commanders to understand and make decisions related to enemy activity and threats.

Many operational sensor systems and much of the processing of their data were designed to support decision makers at the strategic and operational levels. This approach was effective during the Cold War years when the Soviet Union was the principal threat to the United States and its deployed forces. But it has been less effective during recent conflicts such as those in Kosovo, Afghanistan, and Iraq, where fleeting and disguised targets require significantly greater speed and agility in operating and managing ISR assets.

In recent conflicts, warfighters have increasingly used information from ISR systems to support tactical operations by providing an accurate and up-to-date picture of a complex and rapidly changing battlefield environment (Fig. 1). The quality and timeliness of this tactical picture has become a critical factor in the effectiveness of the tactical commander's decision-making process. The trend toward a more time-critical role for ISR in tactical operations, and the demand for greater accuracy, precision, and continuous surveillance coverage with faster delivery of reconnaissance information, is rapidly increasing.

Attempting to meet this demand, the U.S. military has acquired a diverse ensemble of spaceborne, airborne (manned and unmanned), and surface-based ISR assets that it currently operates in theaters such as Afghanistan. Moreover, it envisions a future layered heterogeneous sensing environment including dedicated traditional ISR assets such as U-2 Dragon Lady, RC-135 Rivet Joint, EP-3 Orion, Guardrail Common Sensor, E-8 Joint Surveillance Target Attack Radar System (JSTARS), Littoral Surveillance Radar System (LSRS), RQ-4 Global Hawk, MQ-1 Predator, and MQ-9 Reaper; small unmanned systems organic to tactical units such as ScanEagle, Shadow, and Raven; and attack systems such as F-35 Lightning II, F-22 Raptor, and B-2 Spirit equipped with a variety of highly capable imaging and electronic support sensors (Fig. 2). Each platform and sensor type has unique capabilities, strengths, and weaknesses.

The U.S. military and Congress recognize the need for agile and synergistic employment of the full spectrum of assets in such an ISR environment.[1–5] Currently, however, ISR assets are tasked and exploited in self-contained enterprises, often referred to as "stovepipes," each of which is highly specialized to a particular ISR asset,

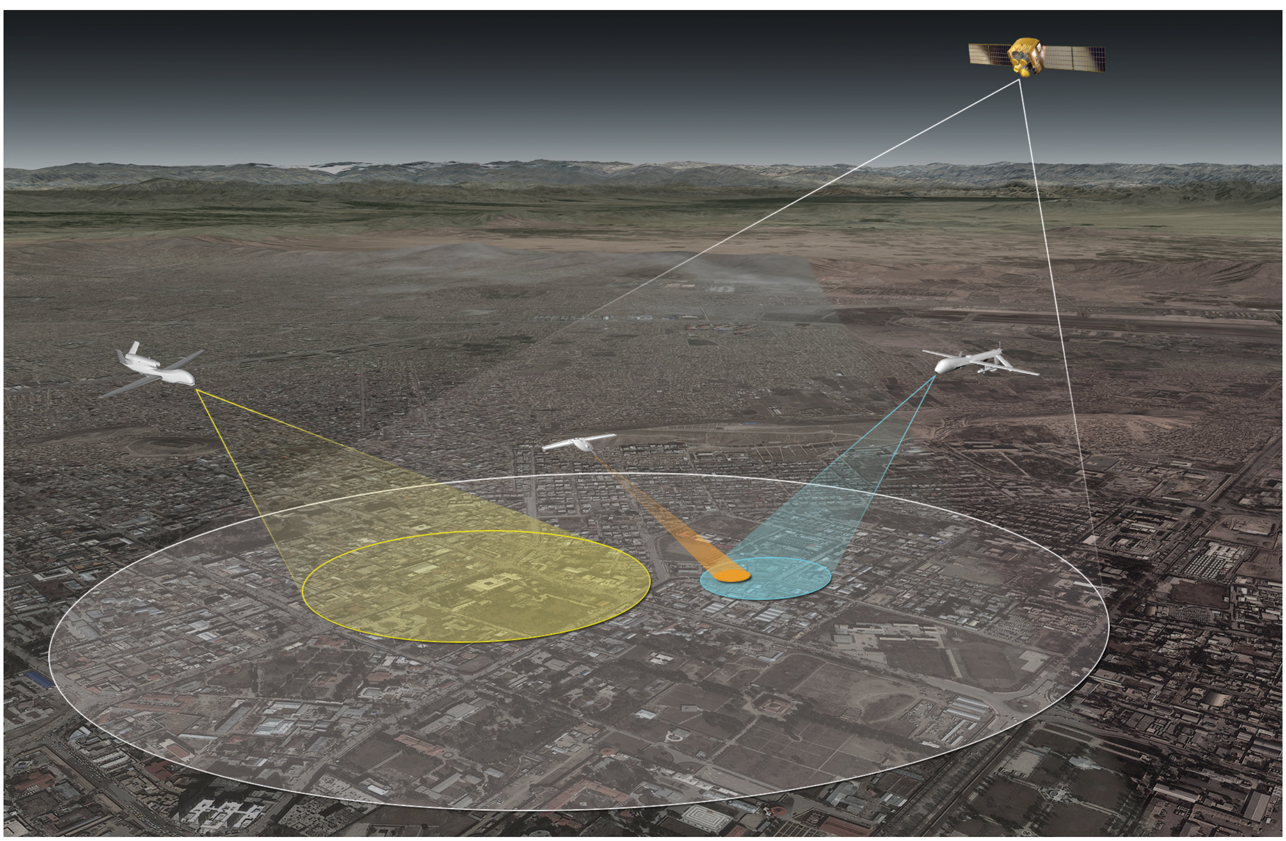

Figure 1. ISR support of tactical operations.

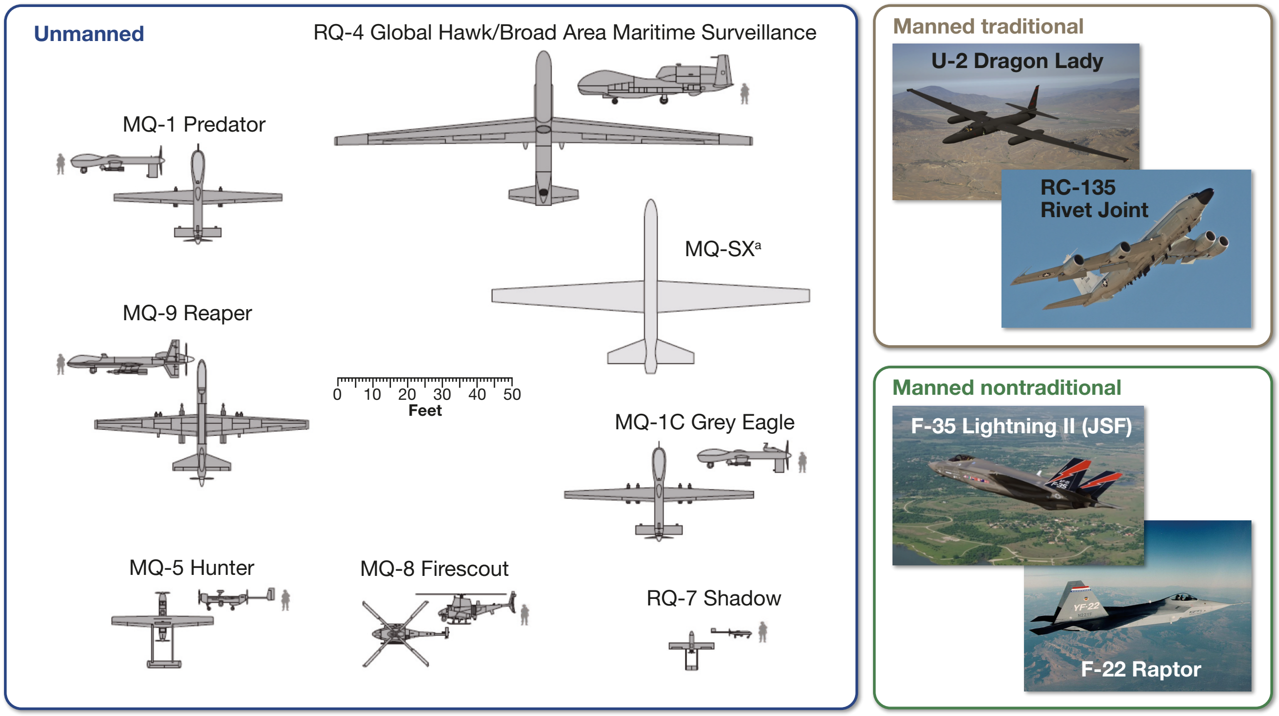

Figure 2. Examples of airborne ISR platforms: unmanned (left) and manned traditional and nontraditional (right).

data type, application, or military domain and does not systematically interact with other such stovepipes. This approach fails to exploit complementary capabilities and opportunities for collaboration. Moreover, the ISR tasking, processing, exploitation, and dissemination process is generally divorced from the rapidly evolving tactical picture and end-user need (e.g., theater commander or exploitation system operator). Consequently, the U.S. military's combatant commands consistently report ISR shortfalls, coverage gaps, and failure to find or track high-value targets, despite the rapidly increasing investment in and deployment of a wide variety of new ISR assets.

The term "collection management" refers to the planning and orchestration of the ensemble of available ISR assets to best satisfy the intelligence requirements of the military operation. Traditionally, ISR collection management operates on a 24-hour air tasking order cycle.[6] Ad hoc tasking procedures are applied to prosecute time-sensitive targets or respond to events that occur during the plan execution cycle. This is primarily a manual process requiring frequent or continual input from experienced operators. It is typical for a single ISR asset to require multiple operators and analysts to employ and exploit it effectively. However, in realistic operational scenarios, the capacity of human operators and analysts to make asset management decisions and interpret the available information can be severely challenged by the fast pace of operations, the enormous volume of incoming sensor data, and the sheer number of decision variables involved to effectively task and control a large number of diverse platforms and sensors.

The ISR enterprise must function efficiently and effectively to provide the tactical commander with timely, accurate, and actionable information in the face of adversary countermeasures and stressing environmental conditions. Improved and automated ISR asset management and exploitation capabilities are needed to satisfy warfighter accuracy, persistence, and timeliness requirements; reduce operator workload; support tactical operations; satisfy mission objectives; and realize the full potential of the envisioned integrated network. These new capabilities will correlate diverse data streams, dynamically allocate resources among competing priorities, achieve the desired synergistic employment, and marshal the full spectrum of assets against a variety of missions.[7]

Closed-Loop Collaborative ISR Resource Management Concept

Achieving the envisioned ISR capability involves the interplay of several key concepts:

- **Fusion of sensor data.** Exploiting the synergies of complementary sensors demands that sensor data be shared and combined (fused) in a manner that extracts useful information from the synergistic data collections.
- **Coordinated control.** Without coordinated control of sensor actions, exploitation of synergy in sensor data is merely opportunistic. Achieving the most out of any sensor data fusion process requires coordi-

nated control with the deliberate intent of improving the output of the data fusion.

- **Dynamic tasking.** Relying on preplanned tasking (such as the 24-hour tasking order) limits the ability to respond to events that occur during the plan execution cycle and to prosecute time-sensitive targets. In addition, it limits tasks to those known at the time of planning, rather than considering all possible current tasks, and thus inhibits efficient use of sensor timeline. Dynamic tasking is essential to react quickly to current information and to avoid stale, suboptimal plans.

- **Feedback control.** It is well known that feedback is a key engineering principle for achieving robustness in the face of uncertainty. ISR missions are characterized by many sources of uncertainty related to targets, sensors, and environment. For this reason, ISR collection plans and the tactical ground picture go stale quickly; there is a rapidly diminishing value of information collected in support of tactical ground operations. Feedback of the tactical picture information state is essential to guide ISR reconfiguration and retasking decisions.

A general closed-loop collaborative ISR (CLCISR) resource management and data exploitation concept, depicted in Fig. 3, addresses these key concepts in an integrated solution. The ISR resources element represents a heterogeneous ensemble of platform and sensor assets that execute motion and sensing commands and produce data of various types. They operate in an environment that includes the entities under surveillance as well as factors that influence their behavior, performance, and output. The processing element represents any and all manual and automated processes that consume data and produce information or situational awareness. This includes data exploitation and fusion processes at several levels of awareness [e.g., Joint Directors of Laboratories (JDL) level 1 entity assessment, level 2 situation assessment, and level 3 impact assessment].[8] These processes use appropriate physical and statistical models of the platforms, sensors, and environment to derive estimates. The control element represents any and all manual and automated processes that generate commands for ISR assets to execute based on current and predicted information. This includes centralized, hierarchical, and distributed control processes. In addition to using models of the ISR resources and environment, the control processes also use models of the exploitation and fusion processes to inform their decisions.

Note that the control loop for ISR resource management is different than the classical plant-controller feedback loop.[9] In the former, actions for sensing assets are selected to improve information about the states of a process that the controller is not influencing. In the latter, actions that directly affect the evolution of the states of a process or system are selected to stabilize, regulate, or otherwise control its behavior. The *Feedback Loop Realization* section presents additional details on the feedback loop realization for CLCISR.

Application of this concept to tactical ISR operations leads to development of a feedback process that continually reallocates ISR resources to respond to changing conditions, maximize the relevance of data collected, and reduce errors and uncertainty about a tactical commander's situation of interest. This process continuously retasks platforms and sensors to maximize the value of information collected and direct ISR resources to resolve evolving ambiguities, appropriately weighted according to mission priorities. The relevant control problem is to dynamically regulate an adaptive data acquisition process to optimize performance criteria related to the quality of the tactical picture in the face of deception and changing conditions.

CLCISR represents a major shift in the way ISR assets are used and managed. ISR assets have traditionally been managed within independent command hierarchies that were not designed to support dynamic and shortened operational planning and execution cycles required for tactical operations. ISR platforms and sensors have traditionally been deployed so as to enhance intrinsic sensor performance but not for contribution to the combined information needs of command. By contrast, in the CLCISR concept, assets are employed according to the tactical value of the information to be provided to the appropriate operational echelon within

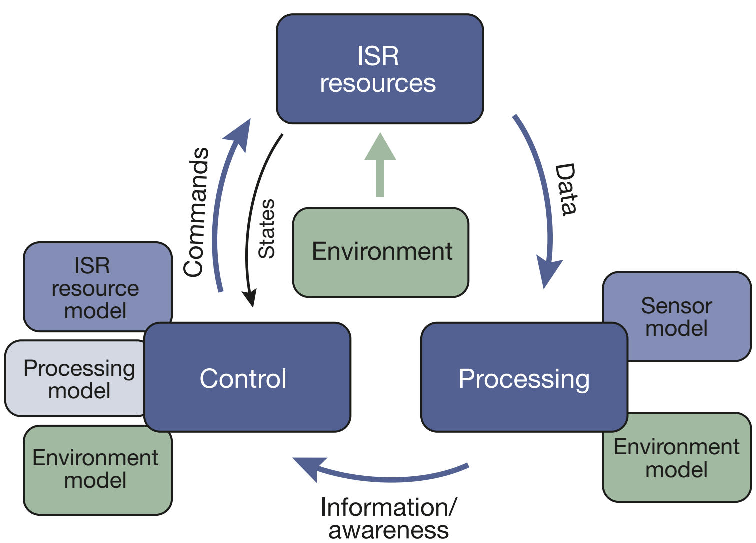

Figure 3. CLCISR resource management and data exploitation concept. (Adapted from Ref. 23.)

timelines consistent with the operational command tempo. Moreover, the diverse sensor ensemble is managed as an integrated and coordinated system to enable gains in effectiveness, productivity, and timeliness that cannot be achieved using legacy practices. By treating diverse platforms and sensors as an integrated ISR enterprise, CLCISR opportunistically exploits the strengths and mitigates the weaknesses of the different assets and provides an effective counter to adversary denial and deception techniques.

The CLCISR approach involves the processing of very high volumes of sensor data and the continuous evaluation of a very large number of decision variables that grows quickly with the number of sensors, the number of targets, and desired precision. In most situations, this will severely stress human operators relying on manually intensive tools and techniques. A semiautomated or fully automated closed-loop process that allows the operator to manage the activity is required to satisfy the rapidly changing information needs of tactical operations.

Moreover, the need is magnified when the ISR ensemble is coupled with an automated upstream data fusion (UDF) process, such as one of those under development by APL (see the article by Newman and Mitzel, in this issue), where sensor tasking can have a strong effect on the performance of the data fusion. In such a case, closing the loop between the data fusion process and the sensors that feed it via automated dynamic resource management can help exploit the full capabilities of the data fusion. CLCISR has the potential to be an important enabler for APL's UDF capability and to dramatically enhance the U.S. military's employment of its rapidly increasing and diverse inventory of ISR platforms and sensors.

CLCISR Scope and Assumptions

APL's work in CLCISR resource management addresses the critical challenge of effectively employing and exploiting the U.S. military's rapidly growing and diversifying ISR capabilities in support of tactical operations under challenging and uncertain battlefield conditions. The goal can be stated in general terms as one of employing the available ISR resources to maximize their efficiency and productivity to satisfy commanders' information needs. Translating these general performance criteria to operationally relevant and quantifiable optimization figures of merit is one of the key problems to realizing a useful CLCISR capability.

This article defines the value of information with respect to its contribution to a commander's estimate of the current picture of the battlespace but does not go further in addressing the problem of determining utility of information with respect to overall mission effectiveness. Moreover, the article assumes a framework in which such tactical picture estimates are derived through data fusion techniques that combine data from multiple sources to produce new information and inferences and achieve more complete, clear, precise, accurate, and timely estimates of the unknown quantities than could be achieved by the use of a single source alone (see Newman and Mitzel, in this issue, and Ref. 10).

Research and development in the field of ISR resource management has been increasing in recent years. Comprehensive reviews of the various technical problems and issues can be found in Refs. 11–14. Effective and efficient employment of ISR assets involves a spectrum of relevant functions at different scales, such as mission planning (e.g., developing target decks and assigning priorities), resource deployment (e.g., assigning assets to different theaters or areas of responsibility), platform motion planning (e.g., orbit definition or dynamic routing), sensor scheduling (e.g., assigning sensor observations to targets), and low-level sensor waveform control (e.g., for multimode radars). The work described in this article focuses on tactical-level asset employment (platform routing and sensor scheduling). It assumes that the available ensemble of ISR assets has been prescribed and that the platforms and sensors are capable of executing the required motion, pointing, and mode-switching commands.

History and Context of CLCISR Development

APL has been active in CLCISR research and development since the early 2000s. The current work traces back to the Effects-Driven Knowledge Management System for Command, Control, Communications, Computers, ISR, and Targeting internal research and development (IR&D) project led by Dr. David W. Porter during 2001–2003. In that project, APL developed the initial concept for an ISR retasking controller inspired by the anticipated dynamic sensor tasking needs of APL's Global Net-Centric Surveillance and Targeting (GNCST) program. The Tactically Responsive ISR Management (TRIM) IR&D project led by Dr. Andrew J. Newman during 2003–2006 built on the initial concept and developed a software prototype ISR retasking controller[15] that was applied to the problem of detecting, tracking, and identifying mobile ground threats using a mix of theater and national ISR assets. The TRIM prototype was demonstrated using simulations based on scenarios motivated by APL's GNCST and Dynamic Time Critical Warfighting Capability programs.[16] However, it did not fully realize the closed-loop architecture because it used a static prediction model of the fusion process in the feedback loop rather than an operational multisensor fusion process. The TRIM IR&D project won the APL Hart Prize for IR&D in the Development category in 2006.

The Precision Engagement of Moving Ground Targets (PEMT) IR&D project led by Dr. Andrew J. Newman during 2006–2009 developed a complete software prototype realization of the CLCISR architecture,[17,18] including an automated resource manager for multiple airborne platforms and an automated multiple-target tracker capable of supporting real-time closed-loop operation for scenarios with a modest number of sensor platforms and targets. The PEMT prototype was applied mainly to scenarios focused on controlling and coordinating teams of unmanned aerial vehicles (UAVs) to track and engage ground targets performing unpredictable maneuvers and operating in cluttered environments, such as urban areas. The project developed a closed-loop simulation test bed integrated with APL's Augmented Reality Environment at APL (ARENA) physics-based simulation environment[19] and demonstrated the closed-loop surveillance and tracking capability against a realistic simulated irregular warfare scenario in an urban area of Baghdad.[20] The project also performed experiments that integrated the APL mission-level autonomy capability into the closed-loop implementation.

The Closed-Loop Layered C4ISR (CLLC4ISR) IR&D project led by Jonathan T. DeSena during 2009–2012 demonstrated a fully realized distributed, asynchronous, real-time CLCISR processing and control system. The project dramatically extended the algorithms, software prototype, and simulation test bed capabilities to validate technical feasibility of the concept in more complex irregular warfare scenarios featuring a wider variety of ISR assets operating in a future layered ISR environment[21] and more realistic operating constraints and battlespace conditions. The project demonstrations more fully take advantage of a variety of APL simulation capabilities and present compelling illustrations of CLCISR benefits in realistic irregular warfare scenarios in the Kabul region of Afghanistan.[22,23] In addition, the project implemented a decentralized control approach that gracefully degrades ISR performance as communications links degrade, yet still achieves performance equivalent to that achieved by the centralized approach when the network allows full communication connectivity.

In recent years, APL has transitioned capabilities developed under IR&D to government-sponsored programs. The Dynamic ISR Management Service (DIMS) project led by Ms. Teresa Fitzpatrick during 2009–2010 and sponsored by Air Force Electronic Systems Center developed, deployed, demonstrated, and evaluated a dynamic ISR management decision aid provided as an application in a service-oriented architecture (SOA).[24] The Dynamic Space Surveillance Network Sensor Tasking for Space Situational Awareness project led by Dr. Eric Klatt during 2010–2012 and sponsored by the Air Force Research Laboratory Information Directorate developed a closed-loop dynamic sensor tasking prototype and demonstrated proof of concept for a capability that would support rapid decision making in space surveillance scenarios where the current deliberative, manually intensive process for tasking the U.S. Space Surveillance Network is insufficiently responsive, such as in response to spacecraft maneuvers, new launches, and lost objects.[25,26]

DESIGN OVERVIEW AND PRIMARY TECHNICAL ELEMENTS

CLCISR Design Overview

Technical Foundations

The technical foundations of the CLCISR approach are sequential decision theory, stochastic control theory, and information theory. The fundamental ISR resource management problem is a sequential decision problem in the presence of uncertainty, where the sequence of decisions influences the information that is acquired about the underlying process or evolving situation. Sequential decision theory and stochastic control theory provide a framework for optimal decision making concerning how to steer, point, and otherwise use the available ISR platforms and sensors to acquire information where the outcome of any decision or control action is uncertain.[27] Information theory provides a framework for quantifying the information that is gained through sensor observations and lost over time as the situation evolves without being observed.[28]

The core elements of an applicable formal probabilistic framework are states, measurements, actions, and reward. The states represent the unknown and possibly evolving quantities of interest that describe the tactical situation (e.g., target presence, position, class, or intent). The measurements represent observed quantities of interest (e.g., azimuth angle, slant range, brightness, signal frequency, or signal time of arrival). They are state dependent and modeled statistically. In certain cases measurements can correspond directly to states, but in general, states can be inferred from measurements only via some estimation process. Sources of uncertainty include sensor noise and bias error, platform dynamics, target motion, variations in clutter and background, occlusions, weather, illumination, and other environmental factors. To differentiate between the actual and estimated or inferred states, the term "information state" is used to refer to the ISR system's current knowledge or representation of the actual states.

The actions correspond to platform and sensor commands that influence actual or potential measurements. For example, an action to slew a radar in the direction of a particular target can result in a measurement of range to target (with some error), whereas an action to turn an airborne platform to a new trajectory can result in the potential to observe a target not otherwise in view (with some probability of success). Each action causes an information state transition with some error or uncertainty in the actual outcome. Actions are constrained by platform maneuverability, airspace restrictions, occlusions, time windows of opportunity, sensor precision, communications bandwidth, power and fuel consumption, environmental phenomena, and other physics of the situation (e.g., slew limits and signal loss). The reward quantifies the expected improvement in information known about the states (i.e., information state) derived from an action or sequence of actions and the consequent state transitions and trajectories.

The notions of control policy and optimality are also needed to complete the framework. A control policy is a function that specifies the action given the information state. For example, one possible control policy would be to assign each sensor to its closest target (i.e., by geographic distance); another would be to assign sensor aimpoints to maximize the expected number of targets detected; another would be to assign each sensor to observe the fastest moving target within a certain distance; another would be to assign an imaging sensor if the target is stopped, or a radar sensor if the target is moving; another would be to minimize target track estimate covariance.

An objective function (or value function) is a function of the reward accumulated over stages and possibly modulated by factors of importance to the particular mission. For example, reward accrued on state information for certain targets could be scaled higher or lower depending on their respective importance to the mission. As another example, it is common to apply a discount factor that diminishes expected reward according to the time delay between the decision and actual execution of the action. An optimal control policy is one that maximizes the chosen objective function (or minimizes the corresponding cost function).

A policy for sensor control is a mapping from the current information states to a feasible sensor tasking. Intuitively, a sensor control policy can be viewed as a table or a function that specifies which sensor action to take under different conditions. For example, a possible surveillance-based policy is to activate the sensor that makes maximum detection improvement; similarly, a tracking-based policy can be designed by choosing the sensor that minimizes the expected uncertainty for all ongoing tracks. Hence, the concept of a sensor control policy is very general and can be used to encode the desired mission objectives.

The fundamental engineering problems for CLCISR, therefore, are to

- formulate a suitable information-theoretic objective function that incorporates relevant mission-related factors; and
- design tractable algorithms that compute an approximately optimal control policy for a discrete time sequential decision process.

The objective functions used for APL CLCISR prototype implementations are based on measures of net information gain and expected mission risk reduction accumulated over a finite planning time horizon and scaled by time discount factors that reflect the diminished predictive value of state estimates over time.

The relative importance to the mission of each target, spatial region, and piece of information at any given time is determined by a risk model (described in the *Sensor Resource Manager* section). The control policy then specifies platform and sensor retasking actions at each time step to maximize the aggregate value of information collected (or, equivalently, minimize aggregate uncertainty or entropy) and maximize the aggregate expected mission risk reduction over the planning horizon.

The sequential decision process models the influence of the ISR platform and sensor information gathering actions on the unknown or uncertain quantities of interest to the mission (e.g., target attributes). In many cases of interest, the state transitions of a decision process possess the Markov property, i.e., the new state depends on only the current state and the action and is conditionally independent of all previous states and actions. In these cases, the sequential decision process is called a Markov decision process (MDP).[29] Moreover, in ISR applications, it is often the case that sensor observations do not exactly determine the current system state, i.e., there is residual uncertainty due to the various sources of error. This class of problem is called a partially observable MDP (POMDP). There is a vast literature concerning MDP, POMDP, and dynamic programming and approximate dynamic programming solution approaches (e.g., Ref. 30). APL CLCISR prototype development efforts apply some of these techniques as described in the *Policy Optimization* section.

Design Principles

The key design principles and technical challenges for implementing an effective and practical CLCISR capability that applies to the scenarios of interest to APL sponsors are

- applying the principles of feedback control to ISR operations by regulating aggregate uncertainty via feedback of a suitable information state;
- orchestrating and coordinating heterogeneous ensembles of ISR platforms and sensors to best exploit the strengths and mitigate the relative weaknesses of different ISR systems and modalities;
- concurrently optimizing and balancing attention among area coverage, surveillance persistence, and tracking, classification, and identification of multiple targets;
- trading current benefit against future cost and vice versa through nonmyopic (far-sighted) dynamic replanning; and
- enabling scalable real-time operation by applying principled approximations to maintain computational tractability while meeting performance objectives.

In realistic scenarios where demand from decision-makers for information exceeds the capacity of ISR resources to produce it, there will be inherent trade-offs among persistence (temporal coverage), area or spatial coverage, and interrogation of individual targets. The ability to adjudicate among competing priorities requires a methodology that evaluates and ranks decisions based on aggregate benefits and costs. This can be accomplished by centralized global optimization, emergent cooperation among autonomous sensor nodes, or hybrid combinations.

Moreover, the ability to optimize exploitation and employment of heterogeneous ISR ensembles to search, track, classify, and identify all areas and targets of interest requires the fusion and control processes to treat diverse data types and information attributes using a common mathematical and engineering framework. This is enabled by the use of multiple modality UDF techniques (see article by Newman and Mitzel, in this issue, and the *Upstream Data Fusion* section) and information-theoretic representations of sensing, dynamics of the environment, and reward.[28] The *CLCISR Prototype Realization* and *CLCISR Design Elements* sections present additional details on the application of these techniques to CLCISR.

The ability to optimize platform and sensor planning over a time horizon beyond the current time step is important in situations where the information-based reward depends on time and the sequence of control actions executed. This is typically the case in the ISR scenarios of interest. Specifically, the reward evolves over time due primarily to sensor and target motion (relative and absolute), which continually changes the sensor–target access (e.g., line-of-sight intervisibility and illumination) and view geometry (range and view angles) as the targets and sensors move with respect to each other, the earth, terrain features, other occlusions, the sun, and other changing phenomena that affect data collection. The dynamics are particularly exacerbated in the case of an aware adversary applying denial and deception techniques.

This situation suggests that a nonmyopic, or far-sighted, approach that plans over an appropriate time horizon (e.g., a sequence of actions) will be superior to a greedy, myopic, or near-sighted approach that plans only for the current time step (e.g., a single action). This approach is analogous to a chess player planning several moves ahead, accounting for the possible moves of his opponent at each stage, to gain an advantage. The planning horizon is selected (or dynamically adjusted) to account for current and future cost and reward while still supporting a tractable computation. The controller trades current cost for future reward and vice versa over the planning horizon and then slides the planning horizon forward in time as it executes the plan. The *Receding Time Horizon Control of MDP* section presents additional details on the application of the receding time horizon control method to CLCISR.

The ISR resource management problem suffers from a combinatorial explosion of decision variables. Specifically, computing the optimal control policy for a sequential decision process suffers from Bellman's "curse of dimensionality" and is intractable for all but trivial problem sizes. However, the control algorithm must generate platform and sensor commands in a timely manner, which means that the method must be computationally tractable and able to terminate if necessary after achieving a good but suboptimal solution to keep latency below a suitable threshold. There are several classes of techniques used to achieve a computationally tractable method that scales to realistic scenarios of interest:

- **Judiciously decompose the problem into subproblems for which the couplings are relatively weak or can be mitigated.** This can significantly reduce the dimensionality of the state space or action space. For example, despite the interdependencies of platform motion and sensor pointing, they can be controlled sequentially rather than jointly without a catastrophic sacrifice of performance. By using this approach, an algorithm would generate a platform route segment and then the sensor pointing commands along that segment rather than generating the combined plan together. Note that fully decentralized cooperative autonomy approaches fall into this category. In these approaches, system-level optimization is achieved through the emergent cooperative behavior of autonomous agents making decisions according to local beliefs and optimization criteria.[31] Thus, the decomposition into subproblems is highly granular. This approach provides effective performance in many applications.

- **Restrict or dynamically adjust the planning horizon.** For example, this can be accomplished by a greedy, short time-horizon or receding time-horizon algorithm.

- **Optimize the policy for the MDP with respect to an approximation of the objective function.** The idea is to obtain a policy that is nearly optimal with respect to the true objective function. For example, this can be accomplished by approximating the objective value for future time steps (referred to as the "value-to-go") using policy rollout, off-line learning, or other techniques that substantially shrink the space of action sequences.

- **Approximately optimize the policy for the MDP with respect to the true objective function.** These techniques produce solutions that are suboptimal but still good enough and provide the flexibility to trade performance against latency by terminating the process when thresholds or deadlines are met. For example, stochastic optimization algorithms such as particle swarm, evolutionary, genetic, and simulated annealing quickly converge to near-optimal solutions, at which point they can be terminated and reset for the next planning time step.

- **Apply heuristics and simplifications that are specific or tailored to the particular application.** These techniques are typically designed to reduce the dimensionality of the state space or action space by taking advantage of an inherent characteristic of the problem. For example, the action space for dynamic sensor tasking may be limited to small perturbations of a nominal tasking plan approved by decision makers.

APL CLCISR prototype implementations use various combinations of these techniques as suited to the tempo and complexity of the particular application, e.g., counter enemy air defense, irregular warfare, or space surveillance. The *Policy Optimization* section presents additional details on some of these techniques as applied to CLCISR.

Operational Practices and Prior Work

In practice, ISR planners and operators apply static optimization techniques to generate daily plans and rely on ad hoc retasking procedures for opportunistic or otherwise time-sensitive asset management. Current automated platform and sensor management tools predominantly apply to a single platform or sensor system and handle search, track, and classification functions separately. A number of researchers have recently presented approaches for controlling ISR assets or ensembles by using feedback of uncertainty or information derived from sensor data and dealing with the inherent intractability of the stochastic dynamic resource management problem. Some of these are listed and described below.

Sinha et al.[32,33] presented algorithms using a Fisher information-based reward function for UAV path planning. Mahler and co-workers[34,35] have presented techniques and principled approximations based on point process and random set theory, most prominently including the posterior expected number of targets method and its subsequent derivatives. Kreucher et al.[36] have presented techniques based on a particle filter representation of the joint multitarget probability density (JMPD) and reward from information-theoretic divergence metrics between prior and posterior JMPD. Stein et al.[37] apply similar techniques using a Gaussian mixture approximation of the multitarget probability density and a mutual information-based reward. Kalandros et al.[38] have presented a covariance control technique for minimizing multiple-target tracking uncertainty. Yang et al.[39] have presented a thorough analysis of sensor management performance measures derived from covariance and information matrices. Scheidt and Schultz[40] have presented techniques for representing and using situational complexity, situational entropic drag, and the processing characteristics of the system to optimize information processing and control of sensing assets.

Prior work has focused primarily on myopic sensor management, i.e., optimizing over a single time step or very short planning horizon. Some results on nonmyopic sensor management for simplified scenarios have appeared recently.[41] The various ISR system optimization techniques have primarily been applied to ISR problems in the ground surveillance domain. However, several researchers have recently applied variations of these techniques to sensor management for space surveillance.[42–46]

CLCISR Prototype Realization

Feedback Loop Realization

APL projects have realized the general CLCISR feedback loop depicted in Fig. 3 in prototype form for specific applications by developing automated data processing and resource management algorithms and scenario simulations. APL CLCISR prototype implementations feature a number of common elements that are tailored for each particular application. The ISR resources and environment elements are realized by a set of scenario simulation components. The processing element and supporting models are realized by a set of UDF components and a pseudo-track manager component. The control element and supporting models are realized by a set of dynamic resource manager components. The feedback loop realization and its core components are illustrated in Fig. 4 and described in the following subsections.

Scenario Simulation

The feedback loop is integrated and exercised within a simulation test bed as described in the *Simulation Test Bed* section. The scenario simulation consists of an integrated set of software components providing a representation of scenario truth and generation of synthetic sensor data. These components simulate essential elements of ISR scenarios including platform aerodynamics, satellite orbit propagation, sensor phenomenology, target dynamics, target features, natural and man-made terrain, background traffic, weather, sun illumination, and other factors influencing the time evolution of the scenario. A sensor simulator for each sensor modality generates synthetic measurement data (including applicable random and bias error) for processing by fusion components.

The current simulation capability includes generation of synthetic optical and infrared imagery and idealized radar contacts to support experiments and demonstrations based on typical ground and space surveillance scenarios. The simulation test bed components and underlying algorithms are configurable, accepting parameter adjustments to support variations and experiments. Additional details are contained in the *Simulation Test Bed* section.

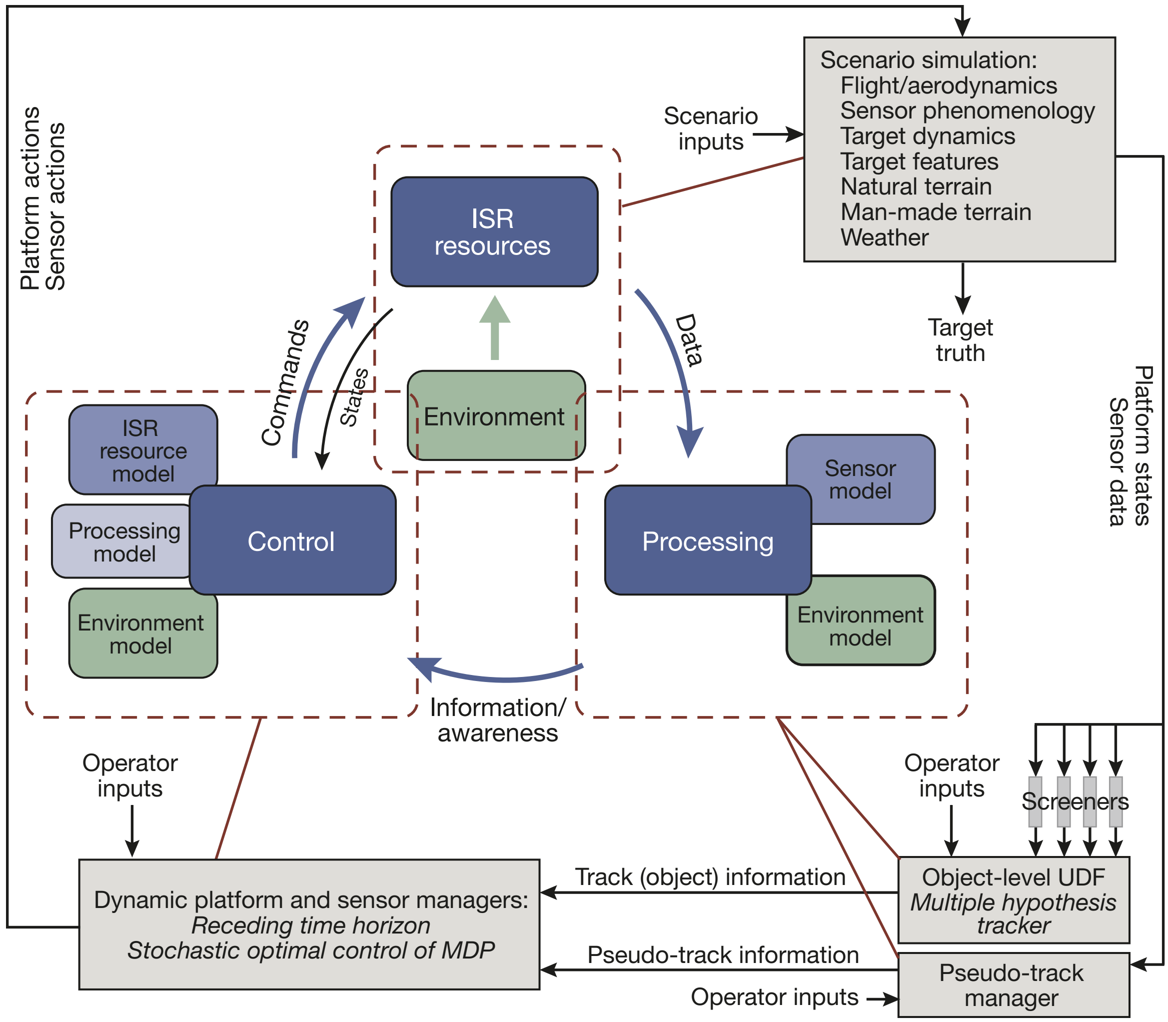

Figure 4. Realization of CLCISR feedback loop. (Adapted from Ref. 23.)

Upstream Data Fusion

UDF (see Newman and Mitzel, in this issue, for a detailed exposition) refers to the processing, exploitation, and fusion of sensor data as closely to the raw sensor data feed as possible. The UDF functionality consists of a distributed set of automated screener and fusion software components performing object-level upstream data exploitation to detect, locate, track, and classify objects observed by the sensors. The screener components are sensor specific; typically there is one for each physical sensor or data type. They process data captured as closely to the sensor source as possible and generate detection data including false alarms. The object-level fusion component uses Bayesian evidence accrual to combine data from multiple sources. It is realized by a multiple hypothesis tracker (MHT)[47,48] specialized for the particular data types to be processed that performs state estimation (filtering) and data association to generate track information including state and classification estimates with quantified uncertainties.

Pseudo-Track Manager

The pseudo-track manager (PTM) is an automated software component that creates, destroys, and updates potential but unobserved tracks (i.e., pseudo-tracks) that represent the information gain available by discovering previously undetected or lost targets and information loss induced over time when not observing.[18,32,33] The PTM is considered a processing component (versus control) because it updates pseudo-tracks on the basis of data inputs and provides a partial representation of the system information state that complements the representation provided by the UDF components.

The PTM and the UDF components are complementary. Whereas UDF tracks encourage the controller to direct sensor attention to currently tracked objects, the PTM pseudo-tracks encourage the resource manager components to direct sensor attention to regions where new or lost objects may be discovered or reacquired. The controller objective function treats both real tracks and pseudo-tracks in the same way. This provides unified treatment of search and track and a mechanism to score and balance sensor actions on an equivalent basis.

The PTM injects pseudo-tracks judiciously to model possible locations of lost or undiscovered objects. The concept is illustrated through examples in Fig. 5. The example on the upper left shows pseudo-tracks arranged in a fixed search grid. The example on the upper right shows pseudo-tracks placed at fixed surveillance points corresponding to major intersections in a city on transit routes of interest. The example on the lower left shows pseudo-track dispersal over a search region as the predicted trajectory of a lost target moves from the point at which the target was lost and becomes more uncertain over time. The example on the lower right shows pseudo-tracks along the expected trajectory of a satellite target immediately after it went undetected by multiple sensor observations and was deemed lost.

For a track representing a real object, the track covariance and probability are given directly by the MHT. By contrast, a pseudo-track covariance is initialized such that its three-sigma error ellipse roughly fits within the search cell of interest. The pseudo-track probability represents the certainty of information about the number of targets in a search cell. It is updated by sensor observations and over time to model information gain and loss. The probability of each pseudo-track j is updated after each sensor scan by

$$\pi_j(k) = \pi_j(k-1)\prod_s (1 - \pi_D(k, s, j)), \qquad (1)$$

where $\pi_D(k, s, j)$ is the probability of sensor s detecting pseudo-track j at step k. Equation 1 represents the information gain from observing a pseudo-track (directing a sensor to search a cell). The probability of each pseudo-track j is updated as time passes by

$$\pi_j(k) = \pi_{\max} - (\pi_{\max} - \pi_j(k-1))e^{-(t_k - t_j(k))/\lambda}, \qquad (2)$$

where π_{max} specifies the *a priori* probability of an undetected target in a cell, t_k is the time at step k, $t_j(k)$ is the time of the last collection on pseudo-track j up to step k and λ specifies the probability growth rate. Equation 2 is simply the solution to the time evolution equation for pseudo-track probability given by

$$\frac{d\pi_j}{dt} = \frac{1}{\lambda}(\pi_{\max} - \pi_j(t)), \qquad (3)$$

and represents the information loss or decay over time.

Figure 6 shows an example where the PTM injects pseudo-tracks after a satellite target is lost after maneuvering from low Earth orbit (LEO) to geosynchronous orbit (GEO). The controller is encouraged to direct attention of sensors to the pseudo-tracks. Consequently, the satellite and its track are reacquired, at which point the PTM destroys the pseudo-tracks and the controller directs attention of the sensors back to tracking known targets.

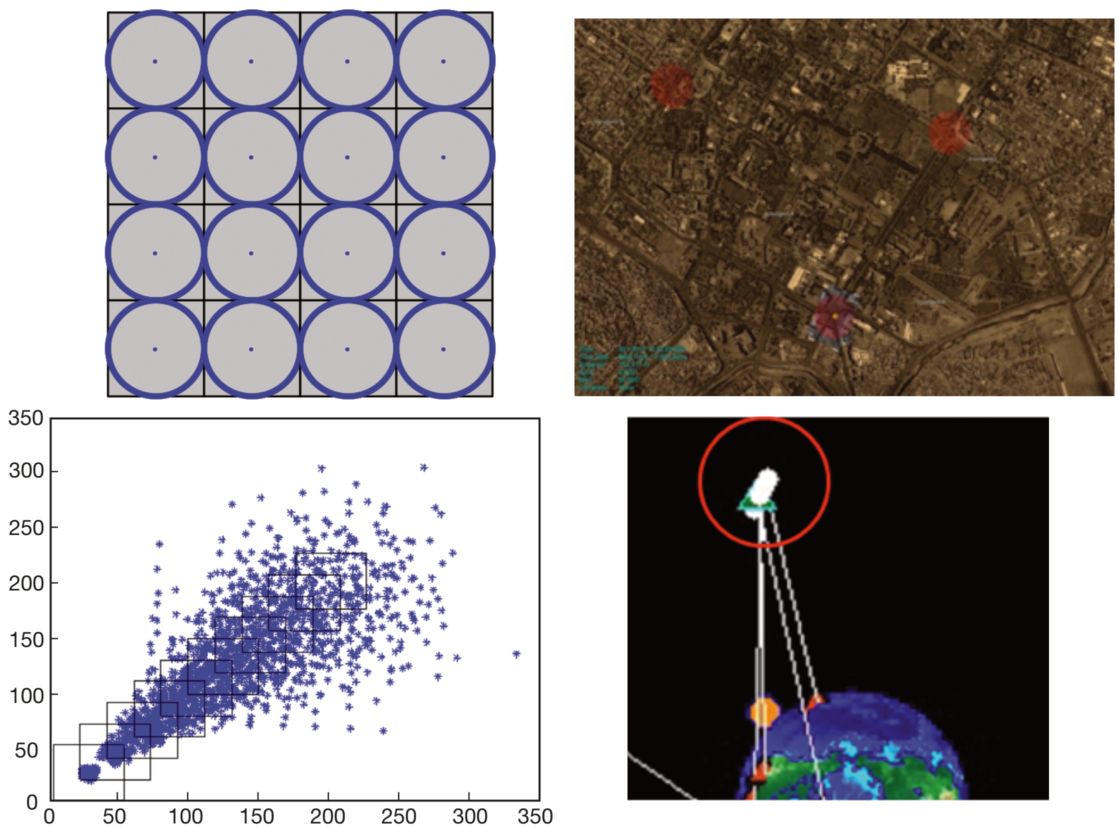

Figure 5. Example pseudo-track distributions over search regions. (Upper left) Uniform grid. (Upper right) Major road intersections. (Lower left) Predicted path of lost maneuvering ground target. (Lower right) Predicted path of lost satellite. (Reprinted from Ref. 23.)

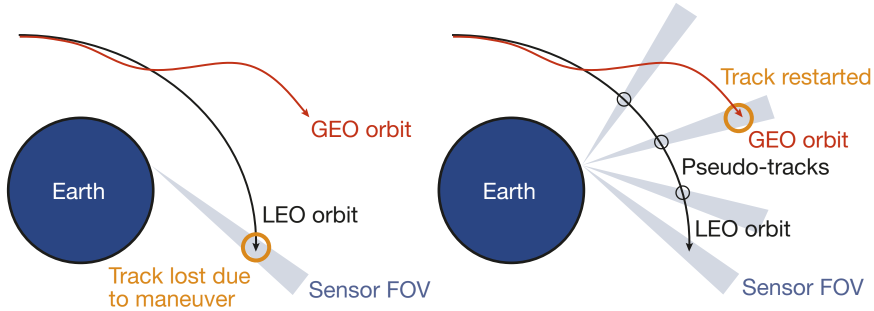

Figure 6. Pseudo-tracks dispersed over a region where a satellite was lost due to a LEO-to-GEO maneuver. (Illustration by S. Martin.)

Dynamic Resource Management Realization

The dynamic resource management functionality is implemented as a set of automated software components that continually optimize ISR resources to collect the most informative and relevant data while respecting physical and operational constraints. It is realized by an interacting pair of stochastic optimal controllers: one functioning as a platform resource manager (PRM) that generates platform navigation commands (e.g., waypoints or motion primitives) and the other functioning as a sensor resource manager (SRM) that generates sensor commands (e.g., aimpoints and modes). Each component optimizes an information-based objective function within constraints and given current and predicted information states provided by the MHT and PTM components, models internal to the controller, information regarding the current states of the platforms and sensors, and (optionally) plans generated by other resource manager components.

The POMDP representing the sequential decision process is typically computationally intractable without forms of approximation.[49] By breaking the joint routing and scheduling problem into a two-step process where the results from the PRM are used as inputs to the SRM, several practical advantages are gained:

- The problem is more computationally tractable.
- Planning timescales and sensor model abstraction can be more appropriately tuned to either the routing or scheduling problems (e.g., longer planning timescales and higher sensor model abstraction for routing than scheduling).
- Modularity is increased. The SRM can operate with sensors on platforms being dynamically routed or not. The PRM can control platforms without concern for the specific underlying sensor scheduling algorithm.

In this scheme, the SRM generally operates with shorter planning cycles than the time required for platforms to reach their next planned waypoint.

Platform Resource Manager

The PRM is a nonmyopic algorithm in the sense that it considers future platform states in addition to the next states being planned during the process of assigning routes to platforms. The PRM component utilizes a receding horizon controller coupled with a particle swarm optimizer as detailed in Refs. 17 and 18. The use of maneuver automatons and non-Markovian reward decision processes were employed as an alternative model instead of POMDPs.[50,51] The PRM was modified to convert the classification estimate information from the MHT into a priority value that scales the reward for collecting on a track in order to encourage platforms to keep high-priority tracks inside the field of view.

The controller uses an objective function based on the Fisher information expected to be gained by a sequence of platform actions.[18,32,33,39] The Fisher information can be interpreted as inverse covariance for purposes of this discussion. The gain in Fisher information is summed over tracks generated by the MHT to encourage improving and maintaining existing tracks, and over pseudo-tracks that encourage searching for lost or previously undetected targets. The reward function treats both real tracks and pseudo-tracks in the same manner.

The objective function scales the information gain by subjective priority factors corresponding to each target and a time discounting factor to represent decay of prediction accuracy. Specifically, the reward function $J(k)$ at time step k to be maximized is defined recursively over the N future time steps of the finite planning horizon, for all sensors and all tracks j (real and pseudo-), as

$$J(k) = \sum_{l=k}^{k+N} \sum_{j} \gamma^{l-k} \alpha_j \left(\log\left|\hat{I}_j(l|l)\right| - \log\left|\hat{I}_j(l|l-1)\right| \right), \quad (4)$$

where γ is the reward discount factor, α_j is the target priority, $\hat{I}_j(l|l-1)$ is the expected Fisher information matrix after predicting the track state estimate to time step l, and $\hat{I}_j(l|l)$ is the expected updated Fisher information matrix for the track after all sensor suite collections up to time step l.

For $l = k$ (current time step), the predicted Fisher information matrix is initialized using the track covariance $P_j(k|k-1)$ and the track probability $\pi_j(k)$ according to $\hat{I}_j(k|k-1) = \pi_j(k)P_j(k|k-1)^{-1}$. The track covariances and probabilities come from the MHT (for real tracks) and PTM (for pseudo-tracks). For $l > k$ (future time steps), the predicted information matrix for each track is computed recursively using a motion model with transition matrix F and process noise covariance Q according to

$$\hat{I}_j(l|l-1) = \left(F(l)\hat{I}_j(l-1|l-1)^{-1}F(l)^T + Q(l)\right)^{-1}. \quad (5)$$

The expected Fisher information after an update at time step l by all sensors s in the ensemble is given by

$$\hat{I}_j(l|l) = \hat{I}_j(l|l-1) + \sum_s \pi_j(k)\pi_D(l,s,j)H(l,s,j)^T R^{-1}(l,s,j)H(l,s,j), \quad (6)$$

where the probability of detection π_D, measurement matrix H and measurement noise covariance matrix R depend on the sensor-to-track geometries.

Sensor Resource Manager

After the next waypoint has been planned nonmyopically in the PRM for each platform via maneuver automatons, a separate myopic algorithm is used to schedule sensors using the risk-based objective function detailed in the next section of this article. This objective function combines both classification and kinematic information. The myopic algorithm uses Monte Carlo simulation[52] over random assignments of tracks to sensors with the assignments of lowest risk being used as the final sensor schedule. This approach does not consider future track states like a POMDP model does. However, because the PRM and SRM are separate optimizations, the SRM is kept at a much smaller time step, on the order of seconds, such that schedules are frequently replanned. This allows the closed-loop system to respond quickly to changes in track state and/or classification estimates that would necessitate replanning.

The SRM uses a risk-based reward approach that attempts to minimize and balance the risks of misclassifying and losing track on an object. It supports the requirement to generate tasking for metric and feature data concurrently and synergistically and to account for both tracking accuracy and object characterization, jointly, in computing reward and cost for optimizing tasking decisions. The algorithm used is based on work by Papageorgiou and Raykin on risk-based sensor management.[53] It was first presented by the authors in Ref. 22 in simplified form for the case of discriminating objects into one of two classes (e.g., nominal versus anomalous). A more general treatment is presented by the authors in Ref. 23 and is summarized below.

The system estimates target state composed of a continuous kinematic state vector $x \in \mathcal{R}^n$ and a discrete object class $y \in \{1 \cdots K\}$. Together these form the mixed continuous-discrete state (x,y). A decision process seeks to choose a specific (x′,y′) to declare as the solution but incurs a cost for that decision that depends on the actual state as defined by the cost function c(x,y;x′,y′). Given that the actual state is unknown and can be characterized by the joint probability density function (pdf) $p_{XY}(x,y)$, the decision process then seeks to declare a state with minimal expected cost, or risk. The risk for a particular declaration (x′,y′) is the expected value of the cost, given that declaration:

$$R(x',y') = \sum_y \int_x c(x,y;x',y')p_{XY}(x,y)dx. \quad (7)$$

It is presumed that the decision process always chooses the minimum risk declaration according to $R = \min_{x',y'} R(x',y')$. In this context, the job of the sensor manager is to pick a sensor parameter vector θ for the next sensing action that affects the probability density of the updated estimate $p_{XY}(x,y)$ in such a way as to minimize the expected future risk for the decision process after the sensing action is completed.

The sensors provide metric and feature data with each observation. Define the metric portion of the measurement as the vector $z \in \mathcal{R}^m$. The feature data are processed to generate a discrete class output $w \in \{1 \cdots L\}$. The risk after a measurement update with a particular measurement (z,w) is determined by predicting the state pdf forward to the measurement time, applying the Bayes update using the measurement likelihood function $p_{ZW}(z,w;\theta)$, and substituting the updated pdf into Eq. 7. The result is

$$R^+(z,w;\theta) = \min_{x',y'}\Sigma_y \int_x c(x,y;x',y')\frac{p_{ZW}(z,w \mid x,y;\theta)p_{XY}{}^+(x,y)}{p_{ZW}(z,w;\theta)}dx, \quad (8)$$

where $p_{XY}{}^+(x,y)$ is the pdf of the state predicted forward to the measurement time. Because the actual measurement outcome is unknown, the sensor manager must rely on the expected value of the risk, given by

$$\langle R^+(\theta)\rangle = \Sigma_w \int_z R^+(z,w;\theta)p_{ZW}(z,w;\theta)dz. \quad (9)$$

After substituting Eq. 8 into Eq. 9, the $p_{ZW}(z,w;\theta)$ term can be canceled, yielding

$$\langle R^+(\theta)\rangle = \sum_w \int_z \left(\min_{x',y'} \sum_y \int_x c(x,y;x',y')p_{ZW}(z,w \mid x,y;\theta)p_{XY}{}^+(x,y)dx \right) dz. \quad (10)$$

Although it is reasonable to use Eq. 10 directly as a sensor manager objective function, the sensor manager described in this article instead uses the expected risk reduction achieved through a sensor action:

$$J(\theta) = R^+ - \langle R^+(\theta)\rangle, \quad (11)$$

where R^+ is the risk achieved in the absence of a measurement update, given by

$$R^+ = \min_{x',y'}\Sigma_y \int_x c(x,y;x',y')p_{XY}{}^+(x,y)dx. \quad (12)$$

The SRM then chooses sensor actions to maximize expected total risk reduction over all targets at each time step.

For a particular application, the sensor manager designer must choose an appropriate cost function suitable for the ISR mission. One such useful cost function is the so-called "notch" cost function, defined as:

$$c_{notch}(x,y;x',y') = \begin{cases} 0, & \text{if } 2\|x-x'\| < \varepsilon(y') \text{ and } y = y' \\ c_{yy}, & \text{if } 2\|x-x'\| \geq \varepsilon(y') \text{ and } y = y' \\ c_{y'y}, & \text{if } y \neq y' \end{cases}. \quad (13)$$

This function provides a class-specific cost $c_{y'y}$, independent of kinematic state, that is incurred for any misclassification of a target truly of class y as class y′. For a correct classification (y = y′), it yields zero cost if the true target kinematic state x is within a class-dependent distance from the declared kinematic state x′; otherwise the cost c_{yy} is incurred. The distance parameter $\varepsilon(y')$ is a mission-specific parameter that acts as a form of resolution requirement. It is allowed to depend on the target class to potentially allow different such requirements for each class. Both it, as well as the c_{yy} term, could also be allowed to depend on the declared target state x′ to allow interesting formulations such as proximity-based cost, but this is deferred for future work. Specifying a particular notch cost function is achieved by defining a square, but not necessarily symmetric, cost matrix

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1K} \\ \vdots & \ddots & \vdots \\ c_{K1} & \cdots & c_{KK} \end{bmatrix}$$

and an epsilon vector $E = (\varepsilon(1)\cdots\varepsilon(K))$. The elements of the cost matrix are subjective parameters based on the operator's judgment of the relative impact of an incorrect classification for the particular application or situation of interest. For example, incorrectly classifying a threat as a safe object may result in damage caused by the threat, whereas incorrectly classifying a safe object as a threat may result in expending resources on responding to a nonexistent threat. The elements of the epsilon vector are subjective parameters based on the operator's judgment of the relative impact of an incorrect kinematic state estimate for each class.

Using the notch cost function of Eq. 13 with $c_{yy} = 1$ in combination with the simplifying independence assumptions $p_{XY}(x,y) = p_X(x)p_Y(y)$ and $p_{ZW}(z,w|x,y) = p_z(z|x)\,p_w(w|y)$, as well as with the standard Gaussian assumptions that allow use of Kalman filter variants to handle the kinematic state estimation process, leads to the following simplified version of the expected risk:

$$\langle R^+(\theta)\rangle = \sum\nolimits_w \min_{y'} \sum\nolimits_{y \neq y'} c_{y'y}\, p_w(w|y)p_Y(y) + p_w(w|y')p_Y(y')q(y',P^+). \qquad (14)$$

Similarly, under these conditions, the risk prior to any sensor action becomes

$$R^+(\theta) = \min_{y'} \sum\nolimits_{y \neq y'} c_{y'y}\, p_Y(y) + p_Y(y')q(y',P). \qquad (15)$$

In Eqs. 14 and 15, P and P^+ are kinematic state estimate covariance matrices predicted forward to the measurement time, but the latter also accounts for the result of applying the measurement update. The function $q(y',P)$, also called the track risk, represents the risk due to uncertainty in the kinematic state, which is a function of the kinematic state estimate covariance matrix and the declared class y' [acting through the resolution parameter $\varepsilon(y')$]. The dependence on x' has been removed because, for Gaussian pdf, the risk is always minimized by setting x' equal to the kinematic state estimate mean $\hat{x}$, for any choice of y'. The track risk function $q(y',P)$ is equal to the probability mass of the kinematic state falling more than $\varepsilon(y')$ away from the kinematic state estimate mean. The discrete likelihood function $p_w(w|y)$ is characterized by a confusion matrix that need not be square (because the number of state classes K need not equal the number of measurement classes L). The risk formulation of Eqs. 14 and 15 are substituted into Eq. 11 to form the risk-based SRM objective function.

CLCISR Design Elements

Information-Theoretic Representations

Information-theoretic representations of sensing, dynamics of the environment, and reward allow the fusion and control processes to treat diverse data types and information attributes using a common mathematical and engineering framework. The approach requires:

- a representation of the state of knowledge about a collection of areas and targets under surveillance;
- a representation of the evolution of the state of information due to motion, arrival, departure, and other processes;
- a representation of the effect of combining the new information contained in a sensor measurement with the current state of knowledge;
- a measure of the change in information content caused by time evolution or updating with new data;
- an information-based reward metric relating information gain (uncertainty reduction) to mission value.

Mathematically rigorous and highly general treatments often use the concept of a belief density (or multitarget belief density), Markov time-prediction integrals (e.g., Chapman–Kolmogorov), Bayesian update via likelihood functions or appropriate analogs, Shannon entropy, and mutual information or Kullback–Leibler divergence (see Refs. 16, 46, and 54 for detailed expositions and applications to ISR scenarios with multiple dissimilar sensor types).

To develop a practical CLCISR capability, APL has adopted simpler representations and measures that are based on principled approximations and provide engineering design flexibility and scalability to scenarios and applications of interest. The state of knowledge, or information state, is derived from track and pseudo-track information. The UDF components create, destroy, and update tracks that represent the information known about discovered objects. The PTM creates, destroys, and updates pseudo-tracks that represent the information known about undiscovered or lost objects. The components are complementary from the standpoint of providing a practical representation of the information state. The evolution of the information state due to scenario dynamics and the incorporation of new information from sensors are modeled by track and pseudo-track data and time update functions. The information-based reward is computed as a function of the updated track and pseudo-track uncertainties via suitable computationally tractable means. Practical alternatives to approaches using the Shannon entropy or Kullback–Leibler divergence include the method using the determinant of the Fisher information matrix given in the *Platform Resource Manager* section and the risk-based reward formulation as given in the *Sensor Resource Manager* section.

The spatial distribution of information at any given point in time can be thought of as a landscape of attractive value as seen by the resource management algorithms. Figure 7 shows a simulated scenario in which the surveillance region has been discretized into area sectors where dark sectors indicate high information availability (lower current knowledge or certainty) and light sectors indicate low information availability. It is clear from the figure that recently searched area sectors have lower current information availability as expected. The spatial distribution of available information will evolve over time as governed by the dynamics models. In particular, information decay will cause light sectors to darken and become more attractive over time. The image on the right shows an example of how terrain and other factors can affect the information landscape as the scenario evolves. In particular, the central obscuration results in more localized information gain as the platforms move along their trajectories and, hence, a wider distribution of available information.

The controller requires a representation of information-based reward that can be achieved through sensor actions. As stated earlier, the PTM and the UDF components are complementary in this regard. The UDF process produces track information representing real objects observed by the sensors, whereas the PTM provides the controller with a representation of information-based reward that can be achieved through sensor actions observing areas where the number of targets present is unknown, new targets may be discovered, and lost targets may be recovered. The tracks and pseudo-tracks, along with sensor models and target or environment dynamics models, provide a means of computing the information-based reward available through observations in the sense of a Bayesian evidence accrual process.

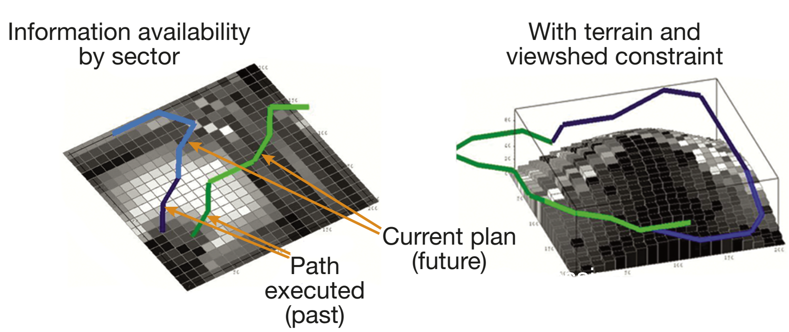

Figure 7. "Information landscape" snapshots—spatial distribution of information at fixed times in different scenarios with platform trajectories resulting from the sequence of control decisions. (Simulations by S. Martin.)

For example, the action of pointing a sensor at a particular track or pseudo-track will improve its covariance and probability according to a sensor observation model, which produces a corresponding positive reward. Conversely, when a particular track is not observed by any sensor, its covariance and probability degrade according to some dynamics model, which incurs a corresponding negative reward, or loss. The reward gained (or lost) for observing (or not observing) a track or pseudo-track is cumulative over all sensors that may observe it and over the planning time horizon. The system reward considered by the optimization is accumulated over all tracks and pseudo-tracks, over all sensors, and over the planning time horizon.

MDP State and Action Space Representation

A MDP[29, 30] is a discrete-time stochastic system with state transitions that are influenced by partially or entirely random dynamics and the actions of a controller or decision maker. The random dynamics create uncertainty in the system state that will result from applying any given control action. At each time step, the process is in some state s, with a set of actions a available from that state. The controller or decision maker chooses one of the available actions, which causes the process to transition into a new state s'. The state transition is governed by the state transition function $P_a(s, s')$, which represents and incorporates all random, uncontrolled deterministic, and control action influences on the system to give the probability of transitioning from state s each possible new state s' after taking action a. The transition from state s to state s' after taking action a gives reward $R_a(s, s')$.

The MDP state transitions possess the Markov property; i.e., the new state s' depends only on the current state s and the action a and is conditionally independent of all previous states and actions. A control policy for the MDP is a function π that specifies the action $\pi(s)$ chosen when the MDP is in each state s. The goal of the controller is to choose a policy that maximizes a function of the rewards accumulated over some period of time. The finite horizon MDP can be represented as a graph where vertices represent states, and edges represent available actions. Figure 8 shows an example with a two-time-step look-ahead horizon and three available actions per state.

For CLCISR, the state space (or information state space) represents the information known about the environment under surveillance. This is modeled by pseudo-track probabilities and track kinematic state estimates, classification state estimates, and statistical

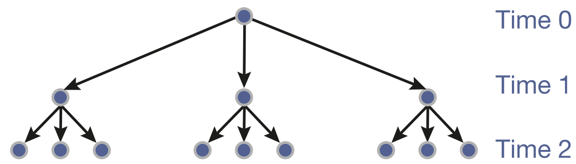

Figure 8. Example graph representation of finite-horizon MDP; states (blue circles) may represent target tracks and pseudo-tracks; actions (arrows) may represent the choice of which track or pseudo-track to observe; tree structure is a special case often applicable to CLCISR. (Illustration by S. Martin.)

uncertainties. The system's information state is given by the entire collection of tracks and pseudo-tracks at a given time step. State transitions result from sensor actions, sensor data processing, and system dynamics. For example, sensor actions generate detections (or lack thereof) in observed regions of space and corresponding measurement data on detected objects. The MHT and PTM process these sensor data and apply system dynamics models to update or propagate the system information state. In general, state transitions are stochastic; they are influenced by unpredictable factors such that the resulting state can only be predicted statistically. The random factors influencing the state transitions include sensor noise, platform deviations from expected trajectory (e.g., due to wind), unpredictable target dynamics (e.g., maneuvers, drag fluctuations, etc.), unpredictable obscurations (e.g., clouds) and other sources of missed detections, false detections, mis-associations, prior track uncertainties, and processing order dependencies in the exploitation algorithms.

For CLCISR, the action space consists of platform motion primitives (e.g., forward, backward, left, right, ascend, descend) or waypoints, sensor aimpoint assignments (e.g., to tracks and pseudo-tracks), and sensor mode assignments. The action space is constrained by platform maneuverability, time windows of opportunity (deadlines), launch and recovery points, no-fly zones, sensor timeline availability, power and fuel consumption, and communications bandwidth. The policy optimization algorithm must enforce these constraints.

For moving or maneuvering sensor platforms, the action space can be modeled using a set of maneuver automatons (one for each platform) to provide a simple abstraction of the air vehicle dynamics suitable for generating action sequences that correspond to feasible trajectories accounting for motion constraints. The maneuver automaton framework[50] provides a modeling language based on motion primitives that correspond to feasible vehicle maneuvers. This formulation assumes that each platform maintains a constant velocity (i.e., trim trajectory) for a period of time between maneuvers. Hybrid maneuver automatons have continuous time variables and discrete maneuvers that can be represented as a graph, where the nodes represent trim trajectories and the edges represent maneuvers. For example, a simple maneuver automaton for an airborne platform might have three trim trajectory states: straight forward, climb, and descend; five transitions for each state: no change, turn right, turn left, tilt up, and tilt down; and a constant discrete time step.

Receding Time Horizon Control of MDP

CLCISR requires a feedback control policy for which approximately optimal platform actions can be computed in a tractable manner, i.e., without excessive exploration of future actions and system states. The receding time-horizon control approach [also referred to simply as receding horizon control (RHC) and as model predictive control] has been shown to produce good results for a variety of applications where a time-varying environment (plant model) results in a time-varying reward function.[55] The RHC generates an approximately optimal sequence of control actions for a finite horizon MDP. Essentially, the RHC approximates the infinite horizon optimal control policy with an iterative series of finite horizon optimizations. The basic RHC algorithm[56] determines the optimal open-loop control $u^*(t,\hat{x}(t'))$ over a finite length interval $t \in [t', t' + T]$ starting at the current time t' and given a current state estimate $\hat{x}(t')$; then sets the control at the current time equal to $u^*(t')$; and then repeats this calculation continuously to yield a feedback control $u^*(t,\hat{x}(t))$ for all t.

The process of planning platform and sensor actions over a finite horizon (N steps), executing only the actions for the next step, sliding the planning window forward in time, and repeating the procedure is illustrated in Fig. 9. The planning horizon and time step size must be chosen to meet the performance and computation latency needs of the given application. Experimentation with APL CLCISR realizations against ground and space surveillance scenarios of interest has determined

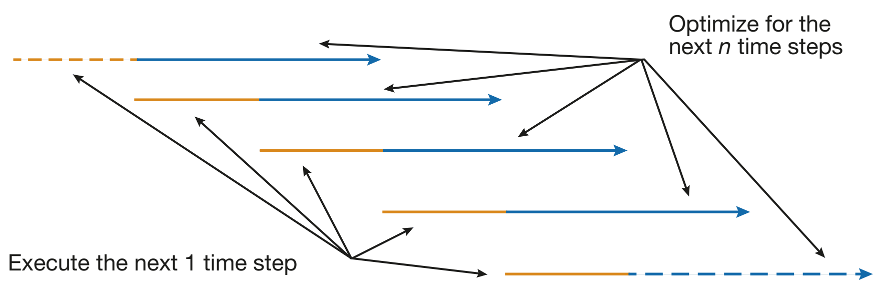

Figure 9. RHC approach to generating an approximately optimal feedback control.

that a dynamic planning horizon of three or fewer steps (current time plus two-step look-ahead) is often sufficient to achieve good performance.

Policy Optimization

The controller computes the optimal policy at each time step, where the policy is optimal with respect to the objective function and subject to the various constraints on platform motion, sensor–target intervisibility, resource consumption, communications bandwidth, etc. The objective function is time varying, nonlinear, multimodal, and discontinuous. The dimensionality of the state space depends on the scope of the surveillance problem, i.e., the number of targets and surveillance points or areas of interest and the number of kinematic and other attributes of the targets that are of interest. The dimensionality of the action space depends on the number of degrees of freedom available for controlling the different ISR platforms and sensors, the granularity with which each actuator can be controlled, and the constraints. In realistic scenarios, the state and action spaces are complex and can be enormous. Given the complexity of the objective function and the combinatorial growth of the state and action spaces, standard optimization methods will not generally apply or be effective for this problem.

In light of this, controller design has predominantly been devoted to simplifying the problem to fit the profile of a computationally tractable algorithm or to designing heuristics that exploit characteristics of the specific problem. There is a vast number of such application-specific solution approaches, which often do not translate from one problem setting to another.

In certain cases it is possible to decompose the problem into subproblems for which the couplings are relatively weak or can be mitigated. In the extreme, the problem can be decomposed into highly localized subproblems, each assigned to an autonomous platform or sensor agent making decisions according to local beliefs and optimization criteria. In such fully decentralized approaches, system-level optimization is achieved through the emergent cooperative behavior of the autonomous agents. This approach provides effective performance in many applications but is beyond the scope of this article. See the article by Scheidt[57] for a description of recent APL work applying decentralized control of autonomous sensing agents to ISR problems.

Dynamic programming[58] is a recursive method for solving MDPs (and sequential decision problems in general) and is based on Bellman's principle of optimality. The optimal policy is computed recursively by iterating on subproblems that optimize over a subset of the planning horizon. The algorithm begins by determining the optimal value (solving Bellman's equation) for the final decision stage and then iterates backwards in time until the optimal value and policy is determined over the entire planning horizon. For this reason, dynamic programming is also referred to as backward induction.

The value iteration and policy iteration algorithms are recursive algorithms that converge to the optimal reward (solution of Bellman's equation). The value iteration algorithm computes a sequence of value functions that is guaranteed to converge under conditions that will generally be satisfied by the MDPs of interest. The optimal policy is then derived from the optimal value function. The policy iteration algorithm computes a sequence of policies directly. The policy iteration algorithm has a much faster convergence rate than value iteration but at the cost of a much more complex computation at each iteration.

These algorithms suffer from Bellman's "curse of dimensionality." They become intractable as the number of states and actions and the length of the planning horizon grow large. For realistic ISR scenarios, these algorithms are not practical because of the computation latencies incurred. Therefore, approximate dynamic programming techniques are applied to achieve a computationally tractable method that scales to realistic scenarios of interest.

One approach is to optimize the policy for the MDP with respect to an approximation of the objective function. This approach is often referred to as approximate dynamic programming.[59] The idea is to obtain a policy that is nearly optimal with respect to the true objective function. Techniques in this category include policy rollout, reinforcement learning, and multiarmed bandits. These methods substantially shrink the space of action sequences under consideration at each iteration of the optimization algorithm.

The policy rollout method[60] approximates the objective value for future time steps (referred to as the "value-to-go") with the objective value that would be achieved using a base policy. For example, a base policy could be a greedy policy or other low-complexity policy. In theory, the schedule derived via policy rollout is guaranteed to outperform the schedule derived by following the base policy. Hence, it is expected that its performance will depend on the choice of base policy in rollout. In practice, it has been demonstrated that the policy rollout can effectively trade-off performance over time in applications including waveform optimization and scheduling in radar and sonar systems, as well as various problems in dynamic resource allocation in communication networks. By contrast, the reinforcement learning method[61] approximates the objective value for future time steps with a function derived from offline trial and error learning. Finally, the multiarmed bandit method efficiently computes approximately optimal solutions to dynamic programming problems through forward induction and priority index rules that are linear, rather than exponential, in the number of targets.[62,63]

Another approach is to approximately optimize the policy for the MDP with respect to the true objective function. These techniques produce solutions that are suboptimal but still good enough and provide the flexibility to trade performance against latency by terminating the process when thresholds or deadlines are met. The family of techniques known as stochastic optimization algorithms[64] can be particularly effective in this regard. They fall into a number of categories such as evolutionary computation, simulated annealing, and tabu search. These algorithms have attracted considerable attention for application to resource management problems because they often perform well for problems where there is a positive correlation between the form that candidate solutions take and their resultant objective values, i.e., for problems where similar solutions yield similar objective values.[65, 66] They are particularly useful for CLCISR because they can be terminated and reset for the next planning time step after quickly converging to a near-optimal solution at the current time step.

Stochastic optimization techniques are advantageous for dealing with the time-varying, nonlinear, multimodal, and discontinuous objective function and large, complex state and action spaces of the CLCISR policy optimization problem. They can optimize over a discrete or hybrid continuous-discrete action space where derivative information is not available and hence gradient-based search methods are not feasible. They perform a parallel search, which can prevent the algorithm from becoming stuck in one particular local optimum and can efficiently sample a very large decision space. However, stochastic optimization techniques generally do not offer guarantees of convergence or bounds on convergence rate. Moreover, they often require complex data encoding and design of heuristics for generating new solutions from a prior set of solutions. The design trade space is large and complex. Design and implementation is often more art than science.

Evolutionary algorithms employ a heuristic approach based on genetic operators such as gene crossover and mutation. These operators are applied to a collection of parents chosen through a selection algorithm. Genetic algorithms are one such class of evolutionary computation methods. They are inspired by the biological process of natural selection. The search space is encoded by a genome representation, and the population is initialized by a set of randomly generated individuals. The algorithm evolves new solutions and searches the decision space by testing individuals for objective fitness, mating the better ones and continuing the process until an acceptable solution is found or until time or other resources are exhausted.

Particle swarm algorithms[67] are another class of stochastic optimization techniques that use a swarm model to simulate a type of social interaction among candidate solutions that often results in convergence toward a global optimum. Individual candidate solutions in the swarm are referred to as particles and are represented as coordinates in the domain space of the function to be optimized, along with a velocity along each dimension of the domain. APL projects have recently applied particle swarm techniques to CLCISR for ground surveillance scenarios.[17, 18]

In addition, APL CLCISR prototypes have recently used Monte Carlo tree search algorithms,[68] which have been shown to be highly effective in planning problems over large domains consisting of an exponential search space. Monte Carlo tree search algorithms attempt to intelligently sample paths through the MDP tree by determining which areas of the tree have been least explored. They have been shown empirically to approach the performance of an optimal value iteration algorithm under certain conditions.

APL CLCISR prototype implementations use various combinations of the above techniques as suited to the characteristics of the particular application. For example, see Ref. 69 for a comparison of different stochastic optimization techniques applied to scheduling the tasks (observations) of an ensemble of space-observing kinematic sensors.

CLCISR Design Summary

In the CLCISR approach, automated PRM and SRM components coordinate and synchronize the ISR ensemble as an integrated unit to continuously maximize aggregate net fused information gain (accounting for search, track, and characterization-based information) and adjudicate tasking among competing priorities across the entire search volume and all targets, as well as over a configurable finite planning time horizon. The objective function and optimization constraints provide a natural adjudication of competing priorities in a resource-limited setting. The far-sighted, or nonmyopic, planning approach requires the resource managers to make decisions based on predictions using uncertain current estimates.

A receding horizon optimal control algorithm determines a set of current and future sensing actions that approximately maximizes net accrued priority-weighted fused information gain over the look-ahead horizon. The net reward accrued from combined actions by multiple platforms over several time steps incorporates relative geometry, time discounting to reflect prediction error, information decay, predicted occlusion, and nonlinear effects resulting from combining multiple dissimilar sensing modalities. Real-time adjudication among competing targets and search versus track priorities is incorporated naturally in the optimization constraints and cost function. The multiasset tasking plan is continually optimized by selecting the best sensor or combination of sensors to reduce the current aggregate statistical uncertainty state.

The information state is derived from feedback of track uncertainties and probabilities computed by the MHT and pseudo-track uncertainties and probabilities computed by the PTM. The information-based reward function accommodates a heterogeneous ensemble of sensors, distributed geographically (and possibly in space) and including different sensor modalities, by scoring sensor actions according to their predicted improvement of track (and pseudo-track) precision and probability. The information gains and losses are scaled by mission-driven priority factors that quantify the relative importance of different targets and search regions. This results in the lowest achievable aggregate priority-scaled uncertainty in number of targets present, target kinematic state, and target class or identity. The algorithm also accounts for urgency (reward for earlier information) and predictive value of estimates (discounting gains achieved further out in time) via scale factors in the reward function.

Computing the optimal control policy requires solution of a large-scale constrained optimization problem at each time step, which in general is not tractable. Computational tractability is achieved by decomposing the problem into manageable weakly coupled subproblems, optimizing subproblems over a finite receding look-ahead horizon, approximating the objective function value-to-go, applying stochastic optimization techniques, and applying application-specific heuristics where appropriate. These techniques produce an approximately optimal control policy while allowing the optimization to be terminated at each step to control latency according to the current needs of the application.

APPLICATIONS AND SIMULATION-BASED EXPERIMENTS

Simulation Test Bed

The CLCISR system prototype design has been implemented and is fully realized as distributed, asynchronous components that communicate via message passing over an Internet Protocol (IP)-based network. Test, development, experimentation, and demonstration of the closed-loop sensor exploitation and tasking system are facilitated by integrating with a simulation test bed. The architecture is modular; individual control and processing components can operate within or outside the simulation test bed environment. One current realization of the simulation test bed for ground surveillance scenarios emphasizing the use of multiple FMV sensors is shown in Fig. 10.

The simulation test bed provides a representation of scenario truth and sensor and tasking operation to support experimentation without incorporating excessive or irrelevant detail. It integrates a set of software components for propagating target motion, simulating airborne platform flight, generating synthetic sensor observation data with specified error statistics, tasking the sensors, displaying the scenario as it evolves in real time, and compiling performance metrics. The simulation test

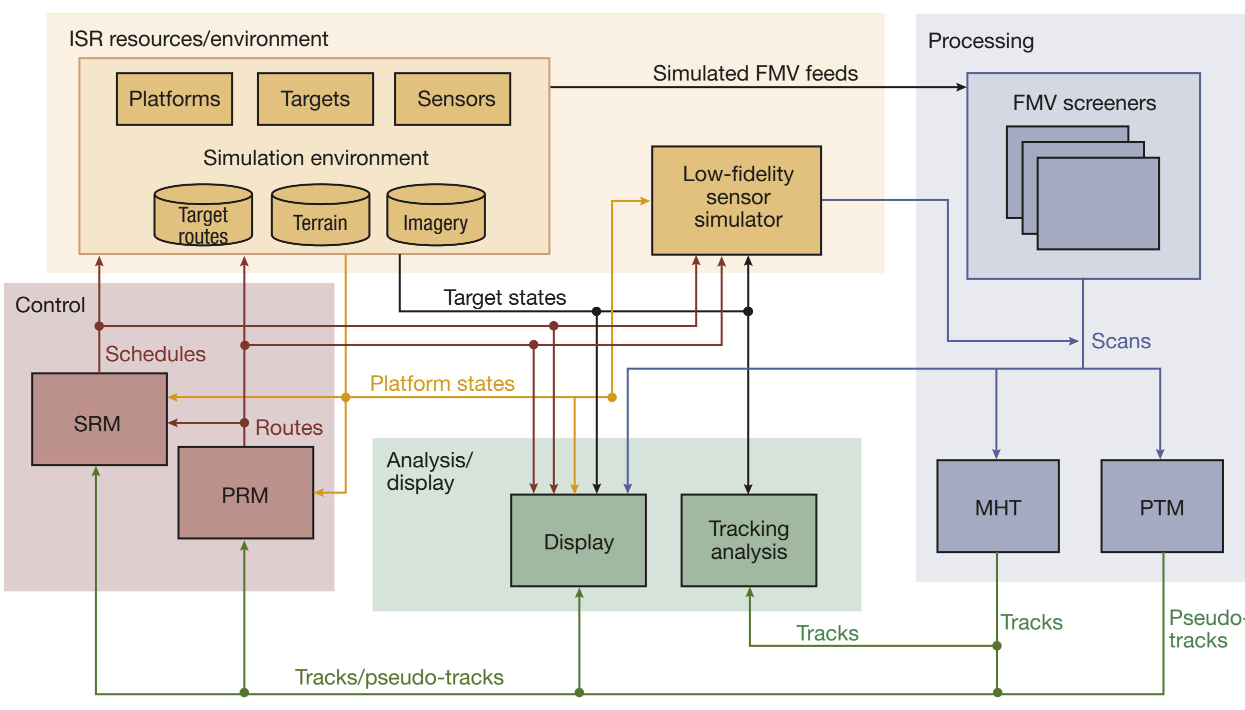

Figure 10. A realization of the CLCISR simulation test bed for ground surveillance scenarios emphasizing the use of multiple FMV sensors. (Adapted from Ref. 23.)

bed components are configurable, accepting parameter adjustments to support variations and experiments.

Support for data exchange among simulation test bed components is provided by the Apache ActiveMQ (http://activemq.apache.org) open-source Java Message Service broker. ActiveMQ provides a platform-independent interface for passing messages between components. Messages between components use the JSON (JavaScript Object Notation; http://www.json.org) format. JSON is a lightweight, text-based, human-readable open standard for data interchange that uses unordered string-value pairs.

APL's ARENA laboratory[19] provides a real-time high-fidelity physics-based simulation environment including urban terrain generation, platform aerodynamics (i.e., six-degree-of-freedom flight models and auto-pilots), urban aerodynamic environment airflows and vehicle interactions, sensor operation and phenomenology (e.g., synthetic optical and infrared imagery), target features, target motion, and obscurations (terrain, buildings, and clouds).

Simulations are populated with high-resolution imagery and terrain for the region of interest. However, detailed models of man-made urban structures were not yet included in the simulation capability. This limitation means that occlusions due to such structures are not accounted for in the current simulation realization described in this article.

Simulation-Based Performance Assessment Experiments

In 2009 the PEMT IR&D project conducted a set of simulation-based experiments to provide a quantitative performance assessment of CLCISR algorithms executing multiple-platform trajectory control in a multiple-target, search-and-track setting. The simulation test bed was used to simulate relatively simple ground surveillance scenarios featuring representative sensors with varying fields of view and typical target densities and motion profiles. The experiments used the centralized RHC for motion planning of a heterogeneous ensemble of airborne sensor platforms operating in a closed feedback loop with the centralized MHT that fused the disparate sensor data to produce target declarations and state estimates. The RHC action space for each air vehicle was represented via maneuver automaton with simple motion primitives. The reward function was based on expected Fisher information gain and priority scaling of target tracks and ground regions. A customized particle swarm optimization algorithm was used to handle the resulting non-Markovian, time-varying, multimodal, and discontinuous reward function. Simulation results showed improved aggregate target detection, track accuracy, and track maintenance for closed-loop operation as compared with typical open-loop surveillance plans. The results of this study were originally documented in Ref. 18.

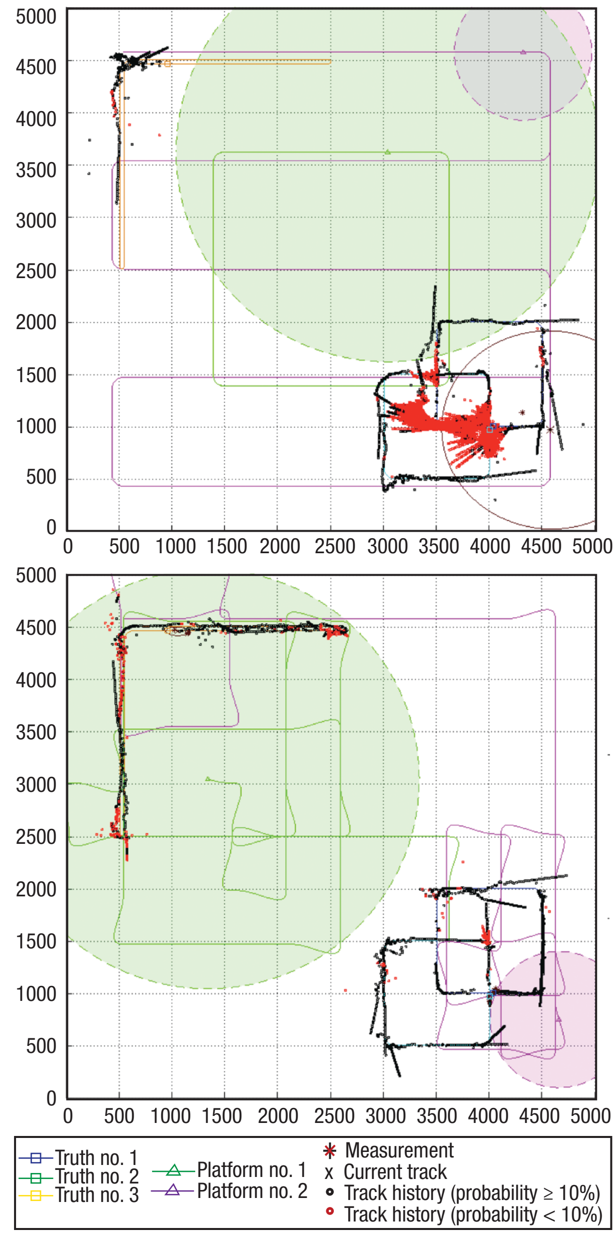

Figure 11. Snapshots of simulation at 800 s with two heterogeneous sensors flying fixed loiter patterns (left) and running closed-loop (right). The closed-loop performance exhibits significantly less confusion (red tracks) and more continuous and accurate tracks.

For example, Monte Carlo simulation trials were run using two sensors with different fields of view and measurement error covariances. Figure 11 shows snapshots at 800 s from one selected trial of the open-loop (left) and closed-loop (right) cases, respectively. Figure 12 provides a comparison of the open-loop and closed-loop root mean square (RMS) error performance. The closed-loop platform control results in significantly reduced tracking error compared to when using the open-loop loiter patterns. Tracker confusion is kept to a minimum at target crossing points; track accuracy and probability are recovered quickly. Intervals where open-loop RMS error is better than closed-loop RMS error are short and infrequent and can be attributed to inherent trade-offs among targets and search sectors in the global optimization.

The open-loop case shows the bimodal behavior caused by the predetermined routes (causing tracking lapses seen in Fig. 12), whereas the closed-loop case shows more consistent results. Closed-loop provides improved performance over open-loop for fraction of time tracked (target 1, 0.50 versus 0.46; target 2, 0.45 versus 0.42; target 3, 0.42 versus 0.24). The significant improvement is for target 3; its remote location and motion pattern created difficulty for the open-loop surveillance that was overcome to a large degree by the closed-loop controller.

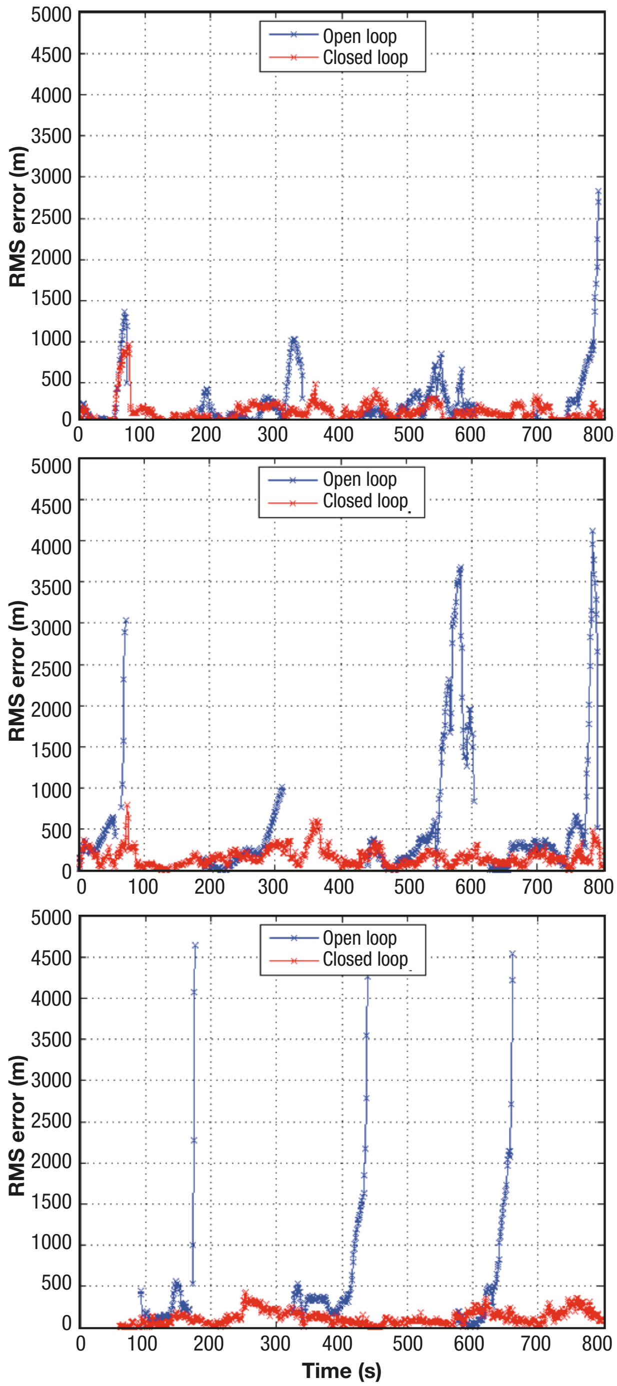

Figure 12. Comparison of average RMS tracking error (blue: open-loop; red: closed-loop) for Target 1 (top), Target 2 (middle), and Target 3 (bottom) in simulation with two heterogeneous sensors.

Irregular Warfare Applications

The CLCISR prototype system has been exercised using the networked simulation test bed against irregular warfare scenarios in Baghdad, Iraq, and Kabul, Afghanistan. The simulation scenarios were designed to exercise, demonstrate, and assess the following capabilities of the full closed-loop system:

- Joint search, track, and classification of multiple targets of interest among background traffic
- Routing of multiple platforms
- Multisensor pointing
- Collaborative ISR control
- Multisensor data fusion

Baghdad Scenario

In 2009, APL conducted a simulation-based experiment and demonstration of CLCISR applied to an irregular warfare scenario set in Baghdad, Iraq (Fig. 13). The purpose was to assess and demonstrate CLCISR capabilities in persistent surveillance, multiple-target tracking, and cooperative, dynamic routing of UAVs. The ISR mission was to maintain persistent surveillance on multiple known safe houses and maintain accurate track on any vehicles leaving safe houses. In the scenario, three high-value targets (HVTs) flee a safe house and scatter over a broad area in the midst of background traffic as the scenario progresses. The available ISR assets consist of a small UAV (SUAV) that has an infrared sensor with a narrow field of view and a radar sensor with a wider field of view (modeled as an idealized radar detector). Both sensors were restricted to look straight down from the platform. The automated exploitation processes in the feedback loop consisted of a video moving object detection component and a ground target tracker component realized by MHT.

The scenario presented several challenges, including multiple targets scattering over a broad area, targets employing evasion maneuvers, the possibility of undiscovered targets, background traffic (clutter), and sensor false alarms. The number of decision variables and the tempo required to actively manage the two platforms would severely challenge a human operator. The automated CLCISR capability successfully managed the two platforms to maintain persistent surveillance on the

Figure 13. Simulated irregular warfare scenario set in Baghdad.

maximum possible number of targets at any given time and to minimize the target tracking errors.

Kabul Scenario

In 2010 and 2011, APL conducted simulation-based experiments and demonstrations of CLCISR applied to an irregular warfare scenario set in the region of Kabul, Afghanistan (Fig. 14). The Kabul region under surveillance in the simulated scenario was significantly larger than the Baghdad region that was used in the prior experiment. The size of the area and the number of targets under surveillance were sufficient to create significant resource contention.

The ISR mission was to find and track a small number of hostile HVTs expected to be operating in the Kabul area of approximately 15 sq. km. In the simulation, the exact number of HVTs is unknown, but more than one are expected. Two previously identified safe houses act as potential centers of activity; and therefore these should be monitored closely. The two safe houses are at cross-town locations, so ISR resources must divide their attention. Targets may be found by monitoring the safe houses as well as by maintaining surveillance on potential target travel routes between safe houses and around the area. Nonhostile (neutral) elements are not of interest, so ISR resources should not be wasted tracking them. Consequently, as entities are detected and tracked it is vital to determine whether the entity can be classified as hostile or neutral.

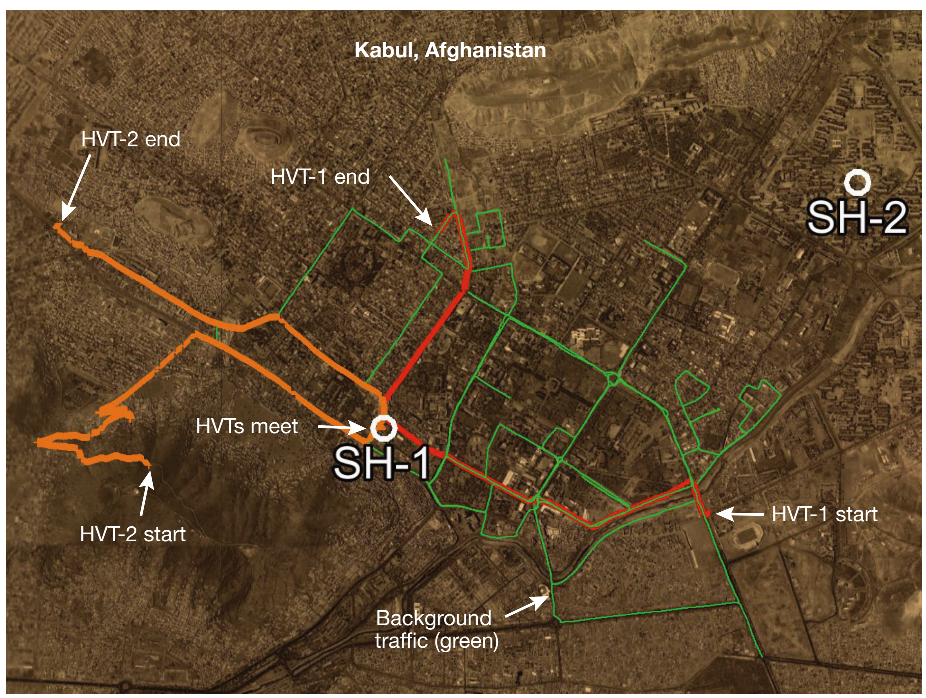

Figure 14. Simulated Kabul scenario target truth including HVT (hostile) routes, safe house locations, and background traffic (neutral) routes. (Adapted from Ref. 23.)

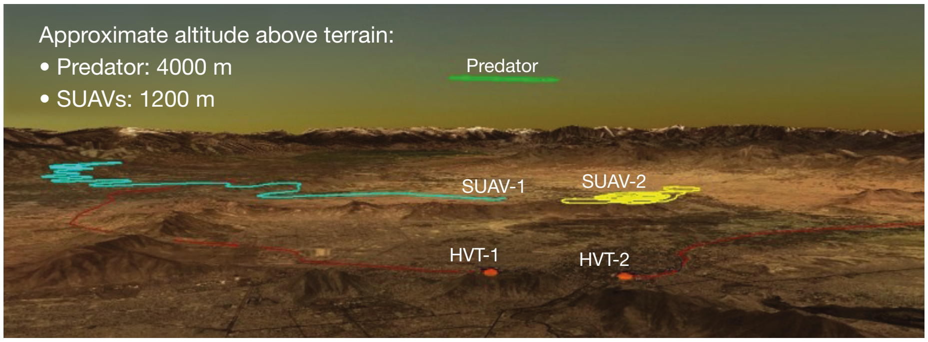

Figure 15. Simulated Kabul scenario ISR platforms and their relative altitudes.

The simulation scenario was populated with eight ground vehicles. Of these, two are designated as hostile and, therefore, are HVTs for the ISR mission. These are designated HVT-1 and HVT-2. The remaining six vehicles are neutral and provide background traffic to the scenario. All vehicles follow prescribed routes. Figure 14 shows these routes along with the locations of the safe houses, which are identified as SH-1 and SH-2. In the figure, HVT-1 follows the red path, HVT-2 the orange, and neutrals travel roads marked in green. HVT-1 starts on the southeast side of town, while HVT-2 starts on the southwest side. The two travel to meet at SH-1 at about 4 min into the scenario. They remain stopped there together for a little over 1 min before HVT-2 leaves and travels to a location in the northwest. About 2 min later, HVT-1 leaves SH-1 and travels to a location in the north of the city. By approximately 10 min into the scenario the HVTs have reached their final locations. The simulation is allowed to continue to run beyond this for up to 25 min as neutrals continue to drive around the city. No activity actually occurs at or near SH-2. It serves as a distraction away from the real action to the ISR system.

The ISR ensemble consists of two SUAVs and one Predator drone (Fig. 15). The two SUAVs can be dynamically routed and operate at an altitude of approximately 1200 m above the terrain. The Predator drone flies a fixed orbit over the center of Kabul at an altitude of approximately 4000 m above the terrain and cannot be dynamically routed. Each SUAV is outfitted with an image sensor capable of outputting object-level detections twice per second. These detections contain only location information (derived from the pixel position); they provide no information distinguishing between hostile and neutral classes. The Predator is outfitted with an image sensor capable of outputting object-level detections once per second. The detections from Predator's image sensor contain both location and classification information. The classification capability is presumed to result from a process (automated or manual) that can be characterized in terms of a square confusion matrix. For this simulation experiment, the 2 × 2 confusion matrix is set to be diagonal such that the sensor's classification output of hostile or neutral is correct 95% of the time. All imaging sensors (those on the Predator and both SUAVs) can be dynamically pointed. Additionally, all image sensors exhibit noisy pixel measurements, a near-unity probability of detection, and a low but non-zero false alarm rate.

The CLCISR system was configured for the scenario as follows. To bootstrap the system (i.e., start from a state of zero information about the number, location, or class of targets) and to stimulate the UAVs to search for targets throughout the scenario timeline, the PTM is initialized with 12 pseudo-tracks (Fig. 16). A high-probability, fast-growth-rate pseudo-track is positioned at each safe house location. In the figure these are shown as bright red opaque spheres. These serve to encourage the system to continually monitor these locations for possible activity during the whole mission. In addition, 10 other low-probability, slow-growth-rate pseudo-tracks are positioned at

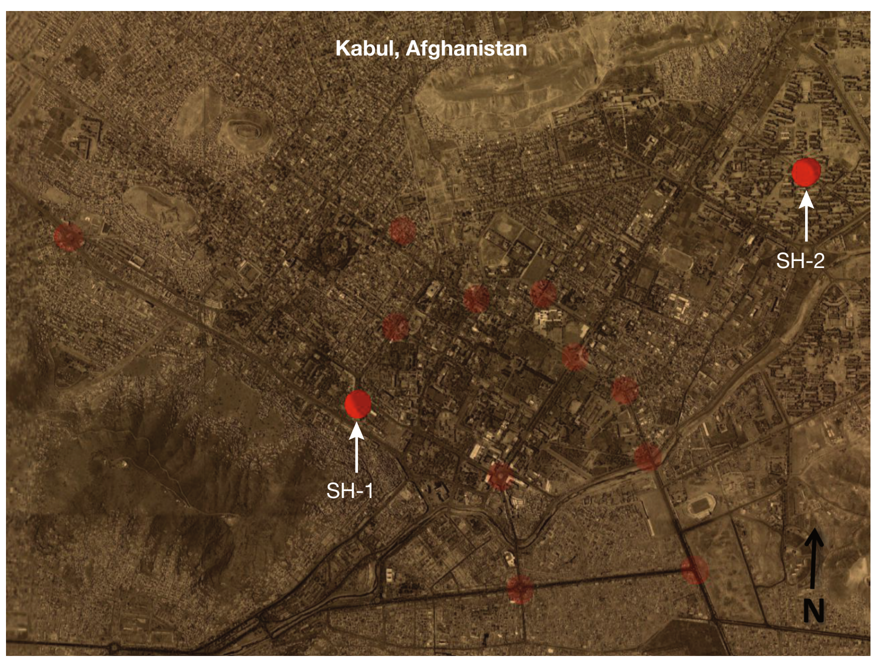

Figure 16. Simulated Kabul scenario pseudo-track locations (red spheres). More opaquely colored spheres indicate higher-probability pseudo-tracks. (Adapted from Ref. 23.)

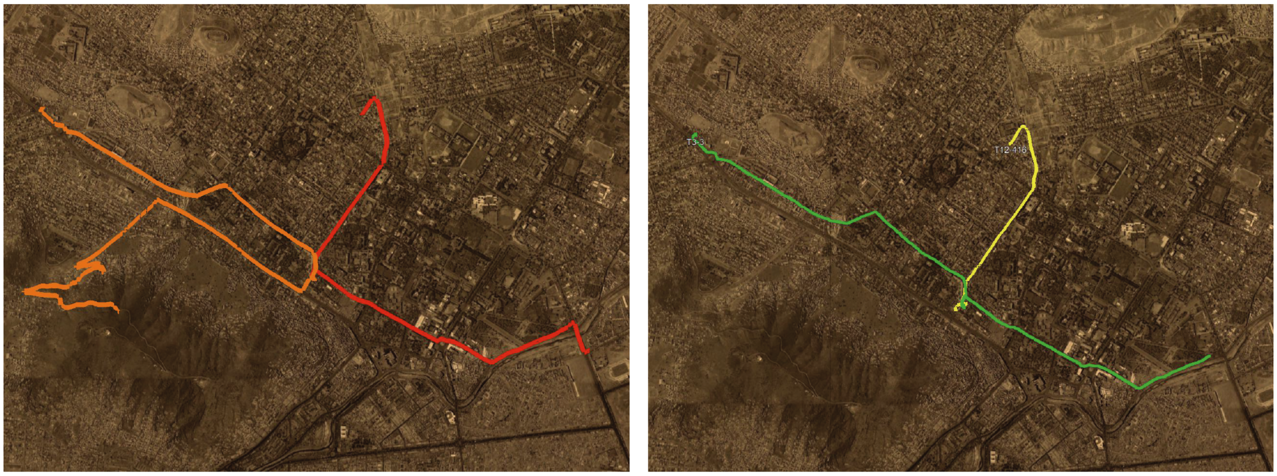

Figure 17. Simulated Kabul scenario with truth and track history. (Left) HVT-1 (red) and HVT-2 (orange) history trails. (Right) Track T3-3 (green) and track T12-416 (yellow) history trails. (Reprinted from Ref. 23.)

various intersections to allow the system to potentially find hostiles en route. Pseudo-tracks represent only the potential existence of targets of interest, which in this case are hostiles. The PTM dynamically updates the pseudo-track probabilities using scan data from each of the three image sensors and sends the pseudo-tracks to the PRM and SRM components.

The MHT processes scans of object-level detections from the all three image sensors to form the fused tactical picture. MHT tracks, each including a classification estimate distinguishing between hostile and neutral, are forwarded to the PRM and SRM components. The SRM jointly plans for all three sensors every second. The PRM jointly plans for the two SUAV platforms. Waypoints for the platforms are generated by the PRM at 15-s intervals. Its receding horizon control look-ahead considers three such intervals into the future for a total planning horizon of 45 s. The PRM does not consider the Predator in its planning process. The effect of this limitation is mitigated by the fact that the Predator's field of regard covers the entire area of interest for the full length of time of the scenario; thus, the best coverage given by the two SUAV platforms is independent of the Predator position.

The system is configured to avoid wasting effort collecting on background traffic (neutrals) via the PRM and SRM objective functions. The PRM scales the information gain for collecting on a track by the probability that that track represents a hostile. The SRM sets the track risk cost (the term depending on the track uncertainty) for the neutral class near zero such that there is little risk associated with not collecting on tracks that are classified with high probability as neutral.

Kabul Scenario Simulation Experiment Results

One simulation realization of the Kabul scenario was run at real-time speed. Data from this run were captured to allow post-run visualization and analysis. In the following subsections, the tracking performance, PRM behavior, and SRM behavior for this simulation run are presented.

Tracking Performance

The system produced only two tracks persisting for any substantial amount of time. Both of these are quickly determined with high probability to be hostile. These two hostile tracks directly correspond to the two HVT vehicles, as can be seen in Fig. 17. HVT-1 is detected early in the scenario as it travels through an intersection in the southeast that is covered by a pseudo-track. This gives rise to track T3-3, shown by the green line in the figure. This track follows HVT-1, unbroken, to the safe house where HVT-1 and HVT-2 meet. Meanwhile, just before HVT-2 arrives at the meeting location, the ISR system detects it and produces track T12-416, shown as the yellow line in Fig. 17. This track then follows HVT-2 to the meeting place. While the HVT vehicles remain stopped at the safe house together, the two tracks formed on them persist, but because of the close proximity of the targets, their covariance ellipsoids overlap and the tracking system cannot distinguish sufficiently between them. Thus, when HVT-2 leaves the safe house, it is understandable that track T3-3 follows it, although it previously followed HVT-1. When HVT-1 subsequently leaves and travels north, it is track T12-416 that follows it. Both tracks follow their targets, unbroken, to their final destinations and maintain track on the targets on the stopped HVT vehicles for many minutes until the scenario concludes.

Qualitative analysis using a 3-D visualization tool reveals that during the times that HVT-1 and HVT-2 are being tracked, their positions are generally contained within the track's 95% confidence ellipsoid. An example of this track containment for each HVT-1 and HVT-2 at an instant in time can be seen in Fig. 18.

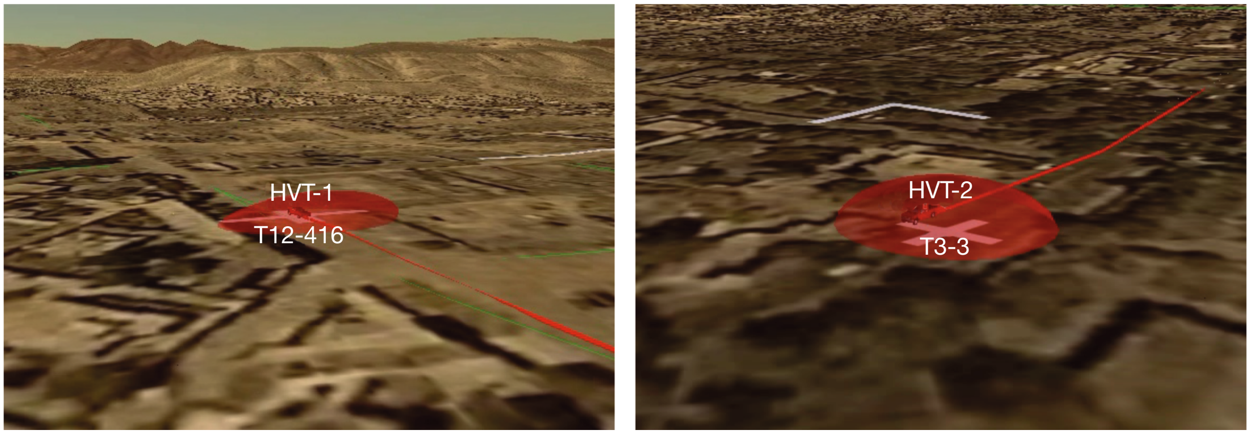

Figure 18. Simulated Kabul scenario example of track containment. The transparent red ellipsoids correspond to track 95% containment ellipsoids. The current true location of target is indicated by the small red trucks. The red trail indicates true prior locations of the target (history). (Adapted from Ref. 23.)

RMS error statistics as a function of time for HVT-1 and HVT-2 are given in Fig. 19. In the figure, RMS error is not plotted at times when the target is not being tracked. While under track, RMS error is held mostly below 5 m with a few spikes above 10 m. RMS error is generally lower while the vehicles are stopped, which occurs while the vehicles are at the safe house (at approximate scenario times 250–430 s for HVT-1 and 200–320 s for HVT-2) and again from time 10 min to the end.

With respect to the goals of the mission, the ISR system was largely successful. Both HVTs were found, correctly identified as hostile, and maintained under constant track thereafter. In addition, although it was by no means guaranteed, one of the HVTs (HVT-1) was discovered while en route and was able to be tracked well before their arrival at the safe house. Furthermore, the system was not confused by the background traffic. Neutrals were tracked mostly just as long as it took to classify them with high probability.

PRM Behavior

The behavior exhibited by the SUAV platforms under control of the PRM is sensible and intuitive. Snapshots of the platform routes at different times in the scenario can be seen in Fig. 20. The two platforms start the scenario across town from each other, with SUAV-1 in the southwest and SUAV-2 in the northeast. Early in the scenario, once a track has been formed on HVT-1 but still before HVT-2 has been detected, SUAV-1 approaches and attempts to follow HVT-1. Meanwhile, SUAV-2 positions itself away from SUAV-1 near SH-2. During the meeting of the HVTs at SH-1, SUAV-1 enters a circular loiter pattern over the safe house while SUAV-2 continues to cover SH-2. As HVT-2 leaves SH-1, the PRM must decide whether to follow it or leave SUAV-1 over the safe house where the other target resides. Given that the positioning of SUAV-2 allows it to shift westward to help out at SH-1, it is sensible that the PRM sends SUAV-1 after HVT-2 at this point. Subsequently, as HVT-1 continues to egress from the city and moves

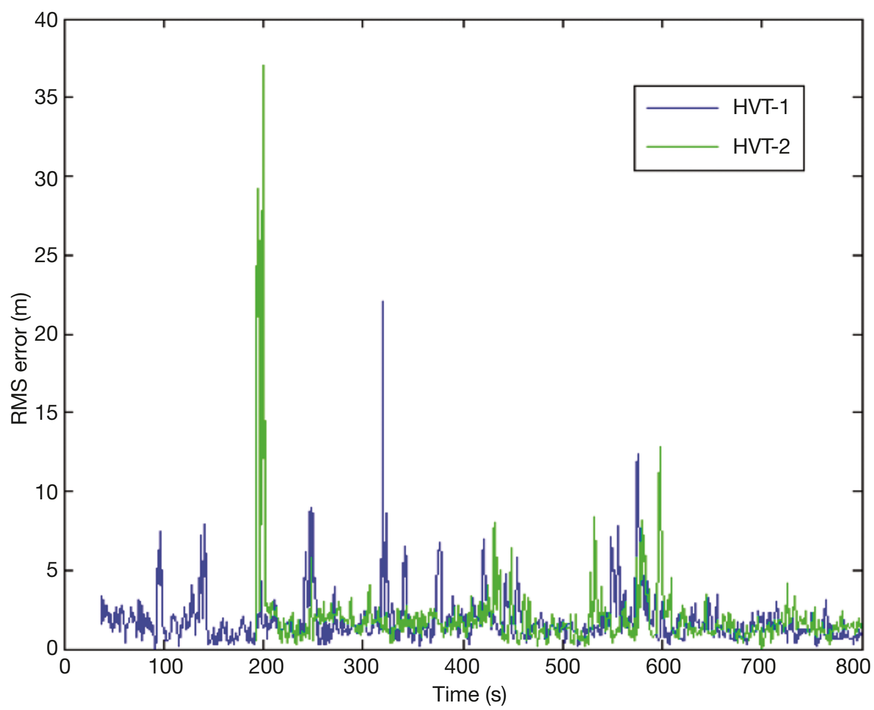

Figure 19. Simulated Kabul scenario RMS tracking error for HVT-1 and HVT-2. (Adapted from Ref. 23.)

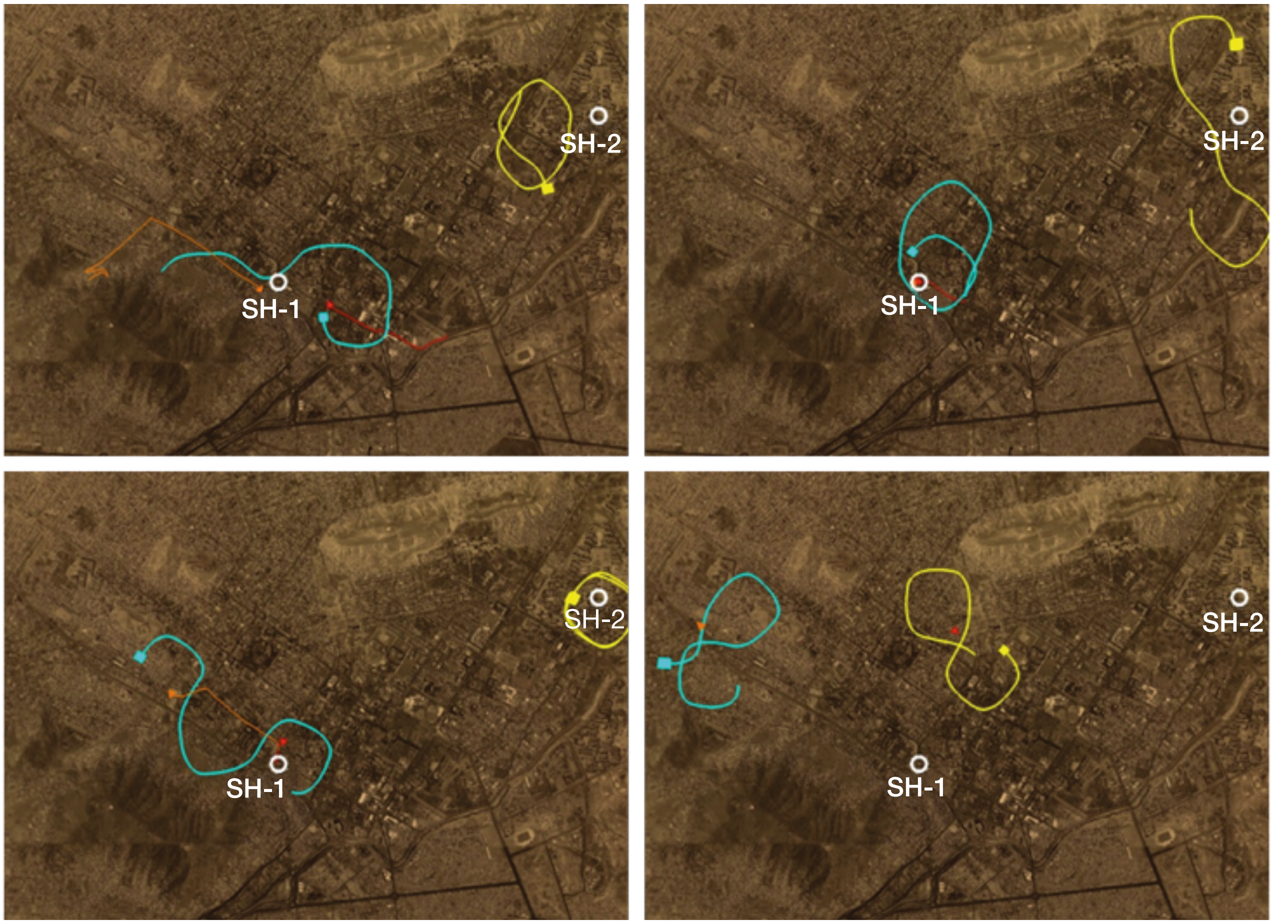

Figure 20. Simulated Kabul scenario PRM routing behavior snapshots. (Upper left) Before HVTs meet at safe house. (Upper right) During HVT meeting. (Lower left) Just after HVTs meet. (Lower right) Scenario end. Trails indicate current positions and 2 min of history for SUAV-1 (cyan), SUAV-2 (yellow), HVT-1 (red), and HVT-2 (orange). (Adapted from Ref. 23.)

northward, SUAV-2 does indeed move over from its position near SH-2 to help track HVT-1 while SUAV-1 can continue to concentrate on HVT-2.

SRM Behavior

It is much harder to intuitively understand the behavior of the SRM than the PRM. This is in part due to the larger action space and faster pace of operation. The behavior is perhaps best understood with respect to the search–track–classify performance. Nevertheless, a sense of the collaborative behavior can be seen in the snapshots of the joint sensor pointing given in Fig. 21.

Space Surveillance Applications

During 2010–2012, APL completed the first two phases of a project sponsored by the Air Force Research Laboratory Information Directorate to demonstrate proof of concept for a semiautomated dynamic sensor-tasking capability that supports rapid decision making in space surveillance scenarios where the current deliberative, manually intensive process for tasking the U.S. Space Surveillance Network (SSN) is insufficiently responsive. The technical and operational feasibility of the concept was validated by developing a dynamic SSN sensor-tasking prototype and simulation test bed, conducting simulation-based performance assessment experiments, and quantifying performance in terms of track maintenance (target custody), search efficiency, and responsiveness to emergent information needs and changing priorities.[25]

The prototype interfaces with and complements the special perturbations (SP) Tasker system[70] that is currently in operational use. It accepts the SP Tasker daily SSN tasking plan as input and dynamically retasks the SSN sensors in a continuous feedback loop with a data exploitation (fusion) process for opportunistic and synergistic data collection and exploitation. It optimizes and coordinates the tasking of multiple geographically dispersed space-observing sensors for timely response to emergent targets while maintaining the integrity of the daily plan as much as possible. It retasks sensors in real time when cued by emergent tasking events for stressing time-sensitive needs such as new launch, lost object (Fig. 22), unknown object, maneuver, transfer orbit, and a variety

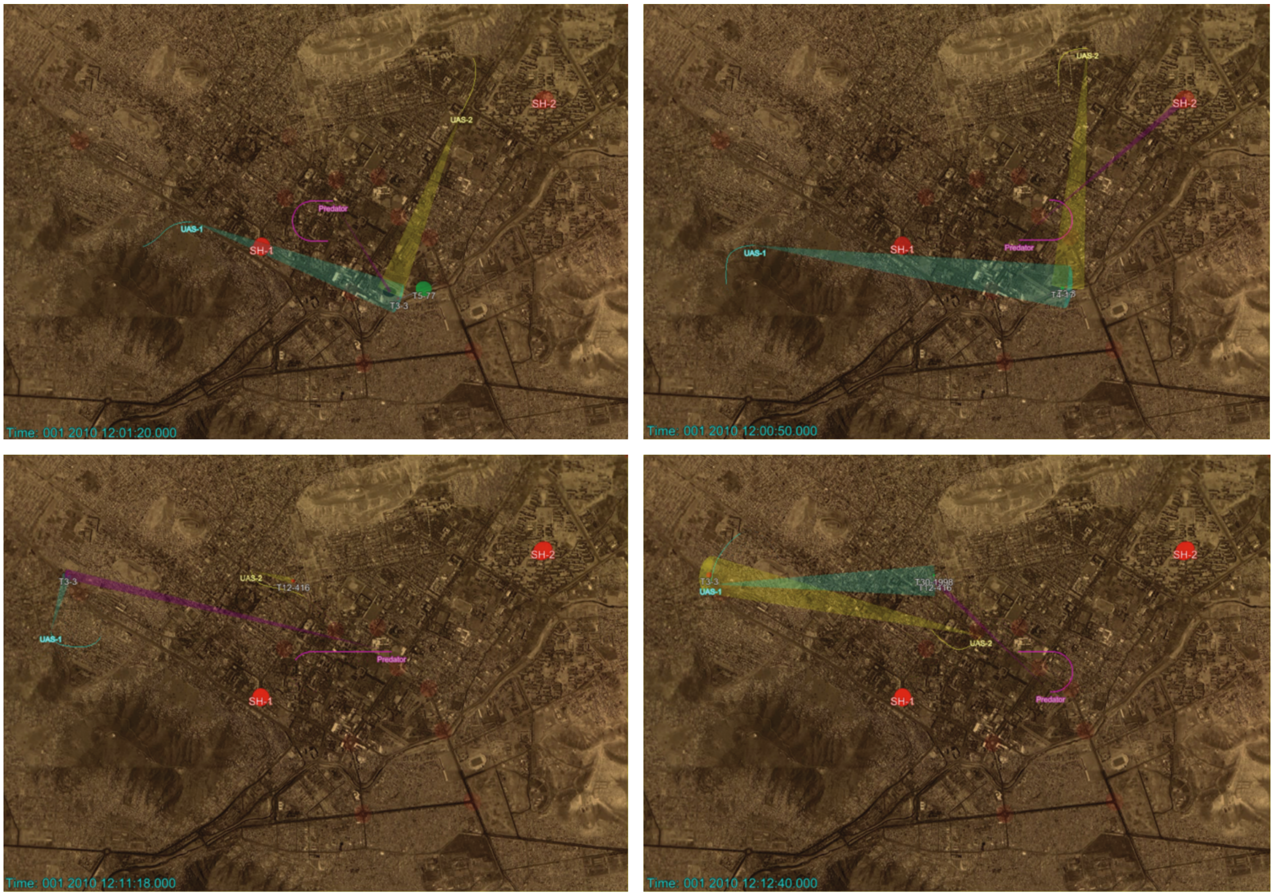

Figure 21. Simulated Kabul scenario SRM sensor pointing behavior snapshots. Scheduled aimpoints are shown by colored cones from sensor to aimpoint for SUAV-1 (cyan), SUAV-2 (yellow), and Predator (magenta). (Reprinted from Ref. 23.)

of potential threats. The core algorithm showed improved performance over several alternative tasking policies in acquiring track, maintaining track, and recovering after maneuvers. It effectively balances attention among many targets judiciously and gives sufficient attention to high-priority emergent targets while sacrificing only modest tracking performance on catalog maintenance tasking.

Dynamic ISR Management in a Service-Oriented Architecture

In 2010, APL completed a project sponsored by the Air Force Electronic Systems Center to develop, deploy, demonstrate, and evaluate a dynamic ISR management decision aid provided as an application in a SOA. The project implemented the prototype decision aid as a Web service called Dynamic ISR Management Service (DIMS).[24] The DIMS is exposed to users by and integrated within the iC2ISR (improved command and control of ISR) simulation environment (Fig. 23) developed for the Air Force Electronic Systems Center by MITRE Corporation (see, e.g., Ref. 71).

The DIMS application provides a semiautomated dynamic tasking optimization capability supporting operator decision making (Fig. 24) for managing a fleet of ISR assets performing surveillance and reconnaissance tasks in an irregular warfare scenario. The DIMS was designed to conform to and take advantage of the general characteristics of a SOA, which include services mapping to business processes, services exchanging data via messages, distributed architecture, discoverable and loosely coupled components, granular service definition, and implementation neutrality.

The DIMS application was exercised in a simulation-based experiment using an irregular warfare scenario set in Afghanistan (scenario constructed by Tim Kehoe of the MITRE Corporation). The ISR mission was to observe activities around 25 known safe houses, located throughout a valley running from the southwest to the northeast in northeastern Afghanistan, that have been determined to be of interest (persistent surveillance) and to support engagement of positively identified enemy operatives (track moving HVTs). The available ISR assets consisted of dedicated Predator and ScanEagle UAVs with FMV sensors and a Global Hawk with an electro-optical camera that was also tasked with a large number of strategic targets throughout the theater. The DIMS periodically generated retasking recommendations for the ScanEagle, Predator, and Global Hawk assets. These

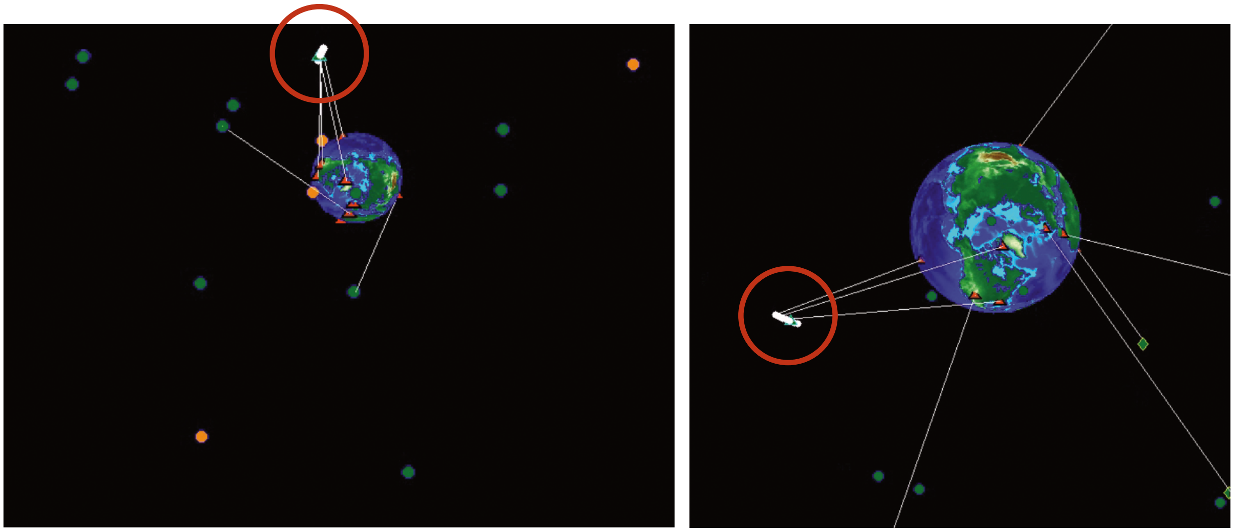

Figure 22. Simulation of closed-loop sensor tasking enabling recovery of satellite track lost because of satellite maneuver. (Simulations by J. Clarke and S. Martin.)

recommendations were presented to an operator for further action, ignored, or automatically accepted.

CURRENT RESEARCH DIRECTIONS

APL is currently pursuing research and development to enhance the CLCISR capability in several directions with the goal of supporting experiments and demonstrations featuring an increasingly broad spectrum of ISR assets, data exploitation techniques, control architectures and techniques, and battlespace operating conditions. The goal is to ultimately demonstrate scenarios featuring the full spectrum of ISR platforms including satellites, aerostats [e.g., Persistent Ground Surveillance System (PGSS)], traditional airborne platforms (e.g., E-8 JSTARS, U-2 Dragon Lady, and MQ-9 Reaper), nontraditional airborne platforms (e.g., F-35 Lightning II and F-22 Raptor), quick-reaction capabilities [e.g., Vehicle and Dismount Exploitation Radar (VADER), MC-12W Liberty, and Blue Devil], swarms of autonomous SUAVs, fixed-location camera networks, and unattended ground sensors equipped with a variety of sensor payloads including electro-optical, infrared, multi-spectral, and hyper-spectral imaging, FMV, wide-

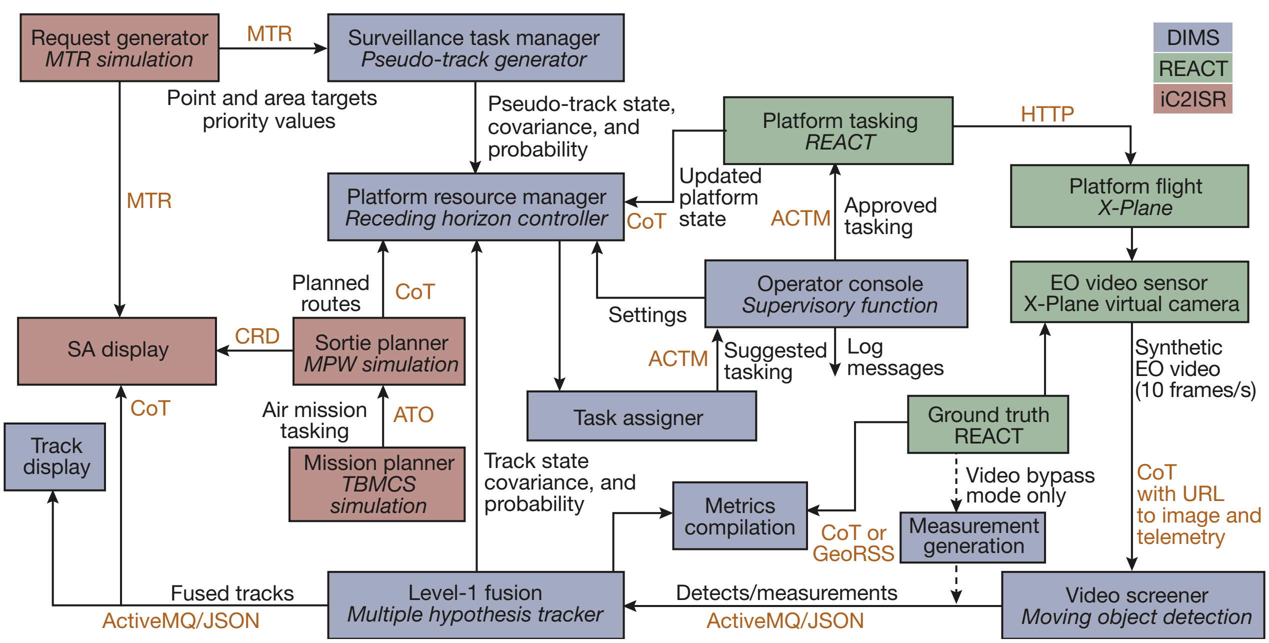

Figure 23. DIMS-iC2ISR simulation functional architecture.

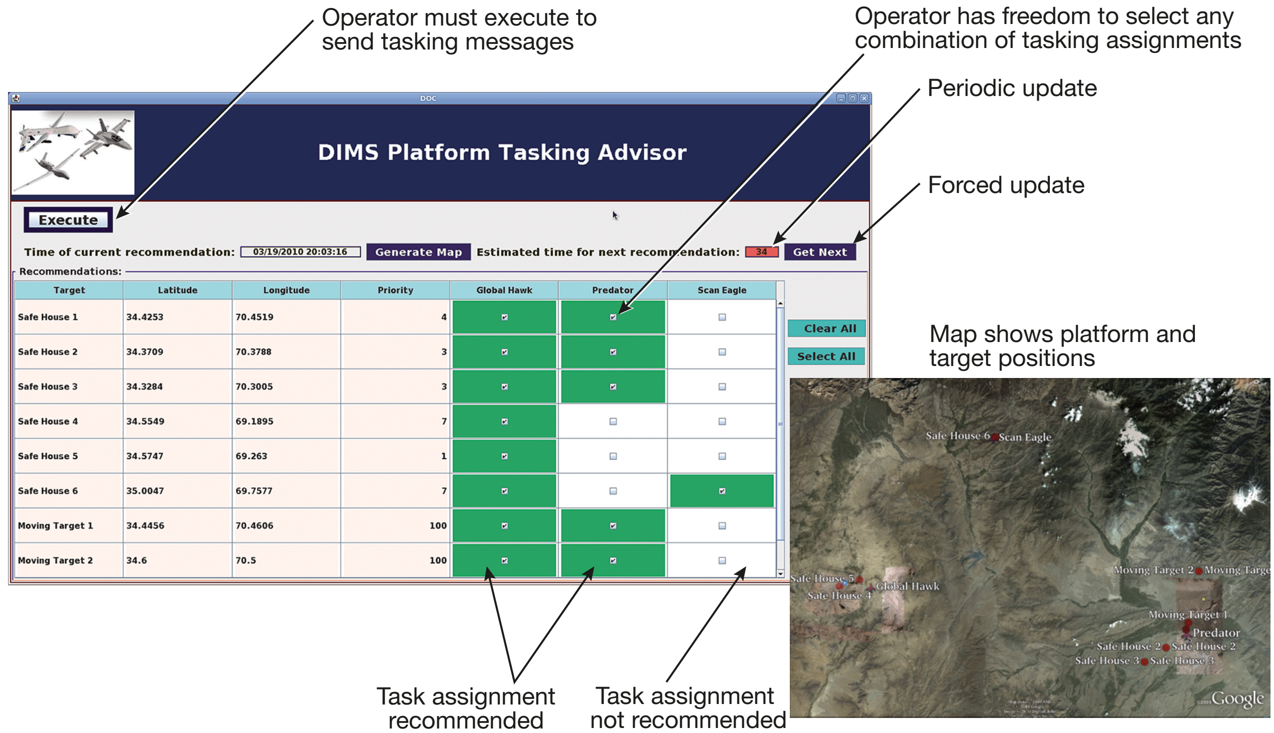

Figure 24. DIMS operator console. (Web interface by S. Subbarao.)

area motion imagery, multi-camera systems [e.g., Gorgon Stare and Autonomous Real-time Ground Ubiquitous Surveillance Imaging System (ARGUS-IS)], synthetic aperture radar, and signals intelligence. Another goal is to enhance performance in the face of challenging factors such as varying terrain, daylight, weather, and contested airspace conditions. Improvements in computational efficiency to enable architectures and algorithms that scale to highly complex problems will be needed. It will also be important to continue extending the CLCISR capability within the ground and space domains as well as to apply it to new domains such as maritime and cyber.

It is reasonable to expect that the ensemble of ISR assets at the disposal of a commander, and the mission that those ISR assets are supporting, will vary over time, perhaps drastically. Different control architectures and techniques will be required to manage the assets depending on the current situation. Research to understand the performance of hierarchical and fully or partially decentralized CLCISR management approaches and development of techniques to integrate short-term and longer-term resource planning are required. To deal with an aware adversary, game-theoretic ISR resource management will be an increasingly important area of research. Algorithms will be needed to estimate the adversary's intent and to adapt to the adversary's strategy.

Additional important topics for future research in CLCISR management include integration of national, theater, and tactical ISR; representation and feedback of different levels of awareness (object, situation, impact, process); incorporation of dissimilar and unstructured information sources (e.g., text and human intelligence reports); the use of different performance figures of merit; net-centric services; the interface to the human operator; and, more generally, enabling humans and automated systems to act in concert in collection, processing, and control.

CONCLUSIONS

CLCISR applies the principles of feedback control to ISR operations. It coordinates a diverse ISR ensemble consisting of traditional and nontraditional air, space, and surface platforms and sensors to operate and produce information as an integrated enterprise. APL has made significant progress in developing prototype CLCISR capabilities and applying them to critical challenges in the ground and space domains. CLCISR has the potential to be an important enabler for APL's UDF capability and to dramatically enhance the U.S. military's employment of its rapidly increasing and diverse inventory of ISR platforms and sensors.

ACKNOWLEDGMENTS: The authors recognize Sean Martin for his immense contribution to development of the algorithms and software prototypes described in this article. The authors also recognize the vast and varied contributions of the following individuals (listed in alphabetical order) to the development of CLCISR applied to critical challenges in the ground and space domains: Jesse Clarke, Michael Cramer, Daniel Dutrow, Tim Frey, Jeff Garretson, Gregg Harrison, Kevin Huber, Eric Klatt, Ed Kreinar, Andrew Lee, Steve Marshall, Jeromy McDerment, Nishant Mehta, Cammy Peterson, David W. Porter, Walter Preissler, Ben Rodriguez, John Samsundar, Suma Subbarao, Adam Watkins, and Danny Williams. Finally, the authors thank the following individuals (listed in alphabetical order) for their long-term financial, managerial, and administrative support and advocacy: Chris Baumgart, Bob Behler, Ray Briscoe, Tim Collins, Bob Finlayson, Teresa Fitzpatrick, Mike Foust, Glenn Gealy, Jim Happel, Jack Keane, Suzy Kennedy, Mark LoPresto, Brooke Mays, Glenn Mitzel, Joe Suter, and Dave Watson.

REFERENCES

[1] U.S. Government Accountability Office, *Intelligence, Surveillance, and Reconnaissance: Preliminary Observations on DOD's Approach to Managing Requirements for New Systems, Existing Assets, and Systems Development*, Report GAO-07-596T, GAO, Washington, DC (Apr 2007).

[2] U.S. Government Accountability Office, *Defense Acquisitions: Greater Synergies Possible for DOD's Intelligence, Surveillance, and Reconnaissance Systems*, Report GAO-07-578, GAO, Washington, DC (May 2007).

[3] U.S. Government Accountability Office, *Unmanned Aircraft Systems: Advanced Coordination and Increased Visibility Needed to Optimize Capabilities*, Report GAO-07-836, GAO, Washington, DC (Jul 2007).

[4] U.S. Government Accountability Office, *Intelligence, Surveillance, and Reconnaissance: DOD Can Better Assess and Integrate ISR Capabilities and Oversee Development of Future ISR Requirements*, Report GAO-08-374, GAO, Washington, DC (Mar 2008).

[5] U.S. Government Accountability Office, *Intelligence, Surveillance, and Reconnaissance: Actions Are Needed to Increase Integration and Efficiencies of DOD's ISR Enterprise*, Report GAO-11-465, GAO, Washington, DC (Jun 2011).

[6] U.S. Joint Chiefs of Staff, *Command and Control for Joint Air Operations*, Joint Pub 3-56.1 (14 Nov 1994).

[7] U.S. Air Force, *Lead Turning the Future: The 2008 Strategy for United States Air Force Intelligence, Surveillance, and Reconnaissance*, Headquarters, Washington, DC (4 Jul 2008).

[8] Steinberg, A. N., and Bowman, C. L., "Revisions to the JDL Fusion Model," Chap. 2, *Handbook of Multisensor Data Fusion: Theory and Practice*, 2nd Ed., M. E. Liggins, D. L. Hall and J. Llinas (eds.), CRC Press (2009).

[9] Levine, W. S. (ed.), *The Control Handbook*, CRC Press, Boca Raton, FL (1996).

[10] Hall, D. L., and Llinas, J., "Multisensor Data Fusion," Chap. 1, *Handbook of Multisensor Data Fusion: Theory and Practice*, 2nd Ed., M. E. Liggins, D. L. Hall, and J. Llinas (eds.), CRC Press, Boca Raton, FL (2009).

[11] Ng, G. W., and Ng, K. H., "Sensor Management – What, Why, and How," *Information Fusion*, **1**(2), 67–75 (2000).

[12] Xiong, N., and Svensson, P., "Multi-Sensor Management for Information Fusion: Issue and Approaches," *Info. Fusion* **3**(2), 163–186 (2002).

[13] Castañón, D. A., Mahler, R., Hintz, K. J., Reich, J., Kadar, I., et al., "Issues in Resource Management with Applications to Real-World Problems," *Signal Processing, Sensor Fusion, and Target Recognition XV*, I. Kadar (ed.), Proc. SPIE, Vol. 6235, SPIE, Bellingham, WA, pp. 62351O-1–62351O-55 (2006).

[14] Hero, A. O., Castañón, D., Cochran, D., and Kastella, K. (eds.), *Foundations and Applications of Sensor Management*, Springer, New York, 2008.

[15] Newman, A., DeSena, J., Samsundar, J., and Porter, D., "A Hybrid-Genetic Algorithm for Fusion-Optimized Dynamic Sensor Retasking," in *Proc. MSS National Symp. on Sensor and Data Fusion*, Laurel, MD, paper SG05 (2004).

[16] Newman, A., *Tactically Responsive ISR Management*, JHU/APL IR&D FY06 Completed Project DTIC Report (30 Sept 2006).

[17] Martin, S. R., and Newman, A. J., "The Application of Particle Swarm Optimization and Maneuver Automatons During Non-Markovian Motion Planning for Air Vehicles Performing Ground Target Search," in *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS) 2008*, Nice, France, pp. 2605–2610 (2008).

[18] Newman, A., Martin, S., DeSena, J., Clarke, J., McDerment, J., et al., "Receding Horizon Controller Using Particle Swarm Optimization for Closed-Loop Ground Target Surveillance and Tracking," in *Signal Processing, Sensor Fusion, and Target Recognition XVIII*, I. Kadar (ed.), Proc. SPIE, Vol. 7336, SPIE, Bellingham, WA, pp. 73360M-1–73360M-12 (2009).

[19] Frey, T. M., and Valencia, M. C., "Enhancing Systems Engineering Agility with Modeling and Simulation," *Johns Hopkins APL Tech. Dig.* **29**(2), 186–196 (2010).

[20] Newman, A., *Precision Engagement of Moving Ground Targets*, JHU/APL IR&D FY09 Completed Project DTIC Report (30 Sept 2009).

[21] Bryant, M., Johnson, P., Kent, B. M., Nowak, M., and Rogers, S., *Layered Sensing: Its Definition, Attributes, and Guiding Principles for AFRL Strategic Technology Development*, AFRL Sensors Directorate Report WPAFB-08-1245 (May 2008).

[22] DeSena, J. T., Martin, S. R., Clarke, J. C., Dutrow, D. A., and Newman, A. J., "Closed-Loop ISR Applied to Irregular Warfare Scenarios," in *Proc. 2011 MSS National Symp. on Sensor and Data Fusion*, Washington, DC, paper NA03 (2011).

[23] DeSena, J. T., Martin, S. R., Clarke, J. C., Dutrow, D. A., and Newman, A. J., "Dynamic Optimization of ISR Sensors Using A Risk-Based Reward Function Applied to Ground and Space Surveillance Scenarios," in *Signal Processing, Sensor Fusion, and Target Recognition XXI*, I. Kadar (ed.), Proc. SPIE Vol. 8392, SPIE, Bellingham, WA, pp. 83920B-1– 83920B-17 (2012).

[24] Newman, A., Cramer, M., and Fitzpatrick, T., *Dynamic ISR Management Service: Final Technical Report*, JHU/APL Technical Report GED-R-10-0053 (30 Apr 2010).

[25] Newman, A. J., Klatt, E. M., Martin, S. R., Clarke, J. C., Rodriguez, B. M., and Mehta, N. L., "Dynamic Space Surveillance Network Sensor Tasking for Space Situation Awareness," in *Proc. 2011 MSS National Symp. on Sensor and Data Fusion*, Washington, DC, paper NA04 (2011).

[26] Newman, A. J., Klatt, E. M., Martin, S. R., Clarke, J. C., Rodriguez, B. M., and Mehta, N. L., *Dynamic Space Surveillance Network Sensor Tasking for Space Situational Awareness: Final Technical Report*, JHU/APL Technical Memorandum SNS-11-04 (20 Dec 2011).

[27] Castanon, D. A., and Carin, L., "Stochastic Control Theory for Sensor Management," Chap. 2, *Foundations and Applications of Sensor Management,* A. O Hero, D. Castañón, D. Cochran, and K. Katella (eds.), Springer, New York (2008).

[28] Hero, A. O., Kreucher, C. M., and Blatt, D., "Information Theoretic Approaches to Sensor Management," Chap. 3, *Foundations and Applications of Sensor Management,* A. O Hero, D. Castañón, D. Cochran, and K. Katella (eds.), Springer, New York (2008).

[29] Bellman, R., "A Markovian Decision Process," *J. Math. Mech.* **6**(4), 679–684 (1957).

[30] Puterman, M. L., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, New York (2005).

[31] Gazi, V., and Passino, K. M., *Swarm Stability and Optimization*, Springer (2011).

[32] Sinha, A., Kirubarajan, T., and Bar-Shalom, Y., "Autonomous Surveillance by Multiple Cooperative UAVs," *Signal and Data Processing of Small Targets 2005*, O. E. Drummond (ed.), Proc. SPIE Vol. 5913, SPIE, Bellingham, WA, pp. 59131V-1–59131V-12 (2005).

[33] Sinha, A., Kirubarajan, T., and Bar-Shalom, Y., "Autonomous Search, Tracking, and Classification by Multiple Cooperative UAVs," *Signal Processing, Sensor Fusion, and Target Recognition XV*, I. Kadar (ed.), Proc. SPIE Vol. 6235, SPIE, Bellingham, WA, pp. 623508-1–623508-12 (2006).

[34] Mahler, R., "Multitarget Sensor Management of Dispersed Mobile Sensors," Chap. 12, *Theory and Algorithms for Cooperative Systems*, D. Grundel, R. Murphey, and R. Pardalos (eds.), World Scientific Publishing Co. Pte. Ltd., Singapore (2004).

[35] El-Fallah, A., Zatezalo, A., Mahler, R., Mehra, R. K., and Alford, M., "Unified Bayesian Situation Assessment Sensor Management," *Signal Processing, Sensor Fusion, and Target Recognition XIV*, I. Kadar (ed.), Proc. SPIE Vol. 5809, SPIE, Bellingham, WA, pp. 253–264 (2005).

[36] Kreucher, C., Hero, A., Kastella, K., and Morelande, M., "An Information-Based Approach to Sensor Management in Large Dynamic Networks," *Proc. IEEE* **95**(5), 978–999 (2007).

[37]Stein, D., Theophanis, S., Kuklinski, W., Witkoskie, J., and Otero, M., "Mutual Information Based Resource Management Applied to Road Constrained Target Tracking," in *Proc. MSS National Symp. on Sensor and Data Fusion*, McLean, VA, paper SH02 (2006).

[38]Kalandros, M., and Pao, L. Y., "Covariance Control for Multisensor Systems," *IEEE Trans. Aerospace Electron. Sys.* **38**(4), 1138–1157 (2002).

[39]Yang, C., Kaplan, L., and Blasch, E., "Performance Measures of Covariance and Information Matrices in Resource Management for Target State Estimation," *IEEE Trans. Aerospace Electron. Sys.* **48**(3), 2594–2613 (2012).

[40]Scheidt, D., and Schultz, K., "On Optimizing Command and Control Structures," in *Proc. 16th International Command and Control Research and Technology Symp. (ICCRTS)*, Quebec City, Quebec, Canada, pp. 1–12 (2011).

[41]Pitre, R. R., Li, X. R., and Delbalzo, R., "UAV Route Planning for Joint Search and Track Missions—An Information-Value Approach," *IEEE Trans. Aerospace Electron. Sys.* **48**(3), 2551–2565 (2012).

[42]El-Fallah, A., Zatezaloa, A., Mahler, R., Mehra, R. K., and Donatelli, D., "Dynamic Sensor Management of Dispersed and Disparate Sensors for Tracking Resident Space Objects," *Signal Processing, Sensor Fusion, and Target Recognition XVII*, I. Kadar (ed.), Proc. SPIE Vol. 6968 SPIE, Bellingham, WA, pp. 69680P-1–69680P-11 (2008).

[43]Chen, H., Chen, G., Shen, D., Blasch, E., and Pham, K., "Orbital Evasive Target Tracking and Sensor Management," Chap. 12, *Dynamics of Information Systems, Theory and Applications Series: Springer Optimization and Its Applications*, Vol. 40, M. J. Hirsch, P. M. Pardalos, and R. Murphey (eds.), Springer, New York, pp. 233–255 (2010).

[44]Hill, K., Sydney, P., Cortez, R., Nishimoto, D., and Luu, K., "Covariance-Driven Sensor Network Tasking for Diverse Space Objects," in *Proc. AAS 11-576, AAS/AIAA Astrodynamics Specialist Conf.*, Girdwood, AK (2011).

[45]Williams, P. S., Spencer, D. B., and Erwin, R. S., "Coupling of Nonlinear Estimation and Dynamic Sensor Tasking Applied to Space Situational Awareness," in *Proc. AAS/AIAA Astrodynamics Specialist Conf.*, Girdwood, AK, paper AAS 11-575 (2011).

[46]DeMars, K. J., and Jah, M. K., "Evaluation of the Information Content of Observations With Application to Sensor Management for Orbit Determination," in *Proc. AAS/AIAA Astrodynamics Specialist Conf.*, Girdwood, AK, paper 11-606 (2011).

[47]Reid, D., "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automatic Control* **24**(6), 843–854 (1979).

[48]Blackman, S., "Multiple Hypothesis Tracking for Multiple Target Tracking," *IEEE Aerospace Electron. Syst. Magazine* **19**(1), 5–18 (2004).

[49]He, Y., and Chong, E., "Sensor Scheduling for Target Tracking: A Monte Carlo Sampling Approach," *Digital Signal Processing* **16**(5), 533–545 (2006).

[50]Frazzoli, E., Dahleh, M., and Feron, E., "Robust Hybrid Control for Autonomous Vehicle Motion Planning," in *Proc. 39th IEEE Conf. on Decision and Control*, Sydney, Australia, pp. 821–826 (2000).

[51]van Kampen, N. G., "Remarks on Non-Markov Processes," *Brazilian J. Phys.* **28**(2), 90–96 (1998).

[52]Metropolis, N., and Ulam, S., "The Monte Carlo Method," *J. Am. Stat. Assoc.* **44**(247), 335–341 (1949).

[53]Papageorgiou, D., and Raykin, M., "A Risk-Based Approach to Sensor Resource Management," in M. J. Hirsch, P. M. Pardalos, R. Murphey, and D. Grundel (eds.), *Advances in Cooperative Control and Optimization: Proceedings of the 7th International Conference on Cooperative Control and Optimization* (Lecture Notes in Control and Information Sciences), Springer, New York, pp. 129–144 (2007).

[54]DeSena, J. T., *FISST Models and Information Theoretic Measures for Sensor Fusion and Resource Management*, JHU/APL Memorandum GVE-06-0043 (17 Aug 2006).

[55]Mayne, D., Rawlings, J., Rao, C., and Scokaert, P., "Constrained Model Predictive Control: Stability and Optimality," *Automatica* **36**(2000), 789–814 (2000).

[56]Mayne, D., and Michalska, H., "Receding Horizon Control of Nonlinear Systems," *IEEE Trans. Automatic Control* **35**(7), 814–824 (1990).

[57]Scheidt, D. H., "Organic Persistent Intelligence, Surveillance, and Reconnaissance," *Johns Hopkins APL Tech. Dig.* **31**(2), 167–174 (2012).

[58]Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Vols. I and II, Athena Scientific, Belmont, MA (2005).

[59]Barton, A. G., Powell, W., and Si, J. (eds.), *Learning and Approximate Dynamic Programming*, IEEE Press, New York (2004).

[60]Bertsekas, D., Tsitsiklis, J., and Wu., C., "Rollout Algorithms for Combinatorial Optimization," *J. Heuristics* **3**(3), 245–262 (1997).

[61]Sutton, R., and Barto, A., *Reinforcement Learning*, MIT Press, Cambridge, MA (1998).

[62]Gittins, J. C., *Multi-Armed Bandit Allocation Indices*, John Wiley and Sons, New York (1989).

[63]Washburn, R. B., "Application of Mulit-Armed Bandits to Sensor Management," Chap. 7, *Foundations and Applications of Sensor Management*, A. O Hero, D. Castañón, D. Cochran, and K. Katella (eds.), Springer, New York (2008).

[64]Spall, J., *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Wiley, NJ (2003).

[65]Luke, S., *Issues in Scaling Genetic Programming: Breeding Strategies, Tree Generation, and Code Bloat*, Ph.D. dissertation, University of Maryland at College Park (2000).

[66]Wall, M., *A Genetic Algorithm for Resource-Constrained Scheduling*, Ph.D. dissertation, Massachusetts Institute of Technology (Jun 1996).

[67]Eberhart, R., and Kennedy, J., "Particle Swarm Optimization," in *Proc. IEEE International Conf. on Neural Networks*, Perth, Western Australia, pp. 1942–1948 (1995).

[68]Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P. I., et al., "A Survey of Monte Carlo Tree Search Methods," *IEEE Trans. Comput. Intell. AI Games* **4**(1), 1–49 (2012).

[69]Newman, A. J., Martin, S. R., Rodriguez, B. M., Mehta, N. L., and Klatt, E. M., "A Comparison of Stochastic Optimizers Applied to Dynamic Sensor Scheduling for Satellite Tracking," *Signal Processing, Sensor Fusion, and Target Recognition XIX*, I. Kadar (ed.), Proc. SPIE Vol. 7697, pp. 76970M-1–76970M-12 (2010).

[70]Miller, J. G., "A New Sensor Allocation Algorithm for the Space Surveillance Network," *Military Operations Res.* **12**(1), 57–70 (2007).

[71]Van Cleave, D. A., "Faster Information Sharing Gives Edge to Warfighters," *MITRE Dig.*, http://mitre.org/news/digest/defense_intelligence/07_09/warfighters.html (Jul 2009).

The Authors

Andrew J. Newman is a member of the APL Principal Professional Staff and Section Supervisor of the Data Fusion Analysis Section of the Weapon and Targeting Systems Group. **Jonathan T. DeSena** is a member of the APL Senior Professional Staff in the Electronic Warfare Systems Group. Since 2003, Dr. Newman and Mr. DeSena have collaborated on projects in sensor and data fusion, target tracking, and dynamic platform and sensor resource management applied to ground, space, and maritime surveillance. They have led a number of IR&D and sponsored tasks and projects in those areas. They shared with co-workers the APL Hart Prize in 2006 for excellence in IR&D in the development category for the TRIM IR&D project. For further information on the work reported here, contact Andrew Newman. His e-mail address is andrew.newman@jhuapl.edu.

The *Johns Hopkins APL Technical Digest* can be accessed electronically at **www.jhuapl.edu/techdigest**.