AAS 13-359

AUTONOMY ARCHITECTURE FOR A RAVEN-CLASS TELESCOPE WITH SPACE SITUATIONAL AWARENESS APPLICATIONS

Ryan D. Coder* and Marcus J. Holzinger†

This paper investigates possible autonomy architecture designs of a Raven-class telescope as applied to the tracking and high level characterization problem in Space Situational Awareness (SSA). Various levels of autonomy are defined and existing systems and capabilities are discussed. Telescope interactions with distributed sensor networks such as the Space Surveillance Network (SSN) are reviewed, and several relationships between autonomy and scheduling of telescopes are addressed. An autonomy architecture design for a Raven-class telescope is presented and future extensions are proposed.

RESEARCH MOTIVATION

In 2001, the Rumsfeld Commission Report concluded that improvements in Space Situational Awareness (SSA) were needed to protect the US and its allies as well as maintain its economic and diplomatic objectives.[1] Joint Publication 3-14, Space Operations, defines the high level activities included in SSA as the characterization and analysis of space objects (SOs) and environmental conditions interacting with space based assets.[2] Space objects consist of active and inactive satellites, rocket bodies, and orbital debris.[3] A key element of SSA is determining whether the orbits of SOs might bring them into close proximity, an event known as a "conjunction," and the conditional probability of SO collision.[4] In order to establish a robust SSA capability, obtaining regular measurements is key.[5] The U.S. Stratetic Command (USSTRATCOM) Joint Space Operations Center (JSpOC) operates the Space Surveillance Network (SSN) and currently tracks 22,000 objects with diameters greater than 10 cm.[6] Additionally, NASA Johnson Space Center's (JSC) Orbital Debris Program Office (ODPO) has primary responsibility for SO population below the SSN detection limit.[7] The need for persistent SSA has been exacerbated by both the Chinese anti-satellite test in 2007[8] and the Iridium/Cosmos collision in 2009,[9] both of which have increased the low-earth orbit (LEO) SO population by more than 60%.[10]

The Space Surveillance Network (SSN) has historically been unable to collect enough raw measurements to fully characterize the SO population.[11] Hence, scheduling limited sensor resources to collect observations of the large SO population is a complex scheduling and resource allocation problem. While many efforts are currently under way to augment the current SSN with additional sensors,[7,12] a great number of additional sensors with improved capability will increase the complexity of this scheduling and planning problem. When the space fence radar becomes operational, the number of tracked objects is expected to exceed 100,000.[4] In addition, modifying established schedules under dynamically evolving scenarios, inclement weather conditions, and hardware faults

*Graduate Research Assistant, School of Aerospace Engineering, Georgia Institute of Technology
†Assistant Professor, School of Aerospace Engineering, Georgia Institute of Technology

is difficult.[11] The planning and scheduling is complicated by the fact that the current global network of SSN sensors are not exclusively controlled by the Air Force Space Command (AFSPC), but also by other entities that provide data to the command, such as foreign governments or other government agencies (OGA) such as the Missile Defense Agency.[4]

Not only has the number of SOs tracked by the SSN greatly increased, but the data products and services provided by JSpOC have a record number of customers as well. Currently more than 100 countries as well as commercial satellite operators regularly request conjunctions assessments and launch screenings.[4] This has greatly increased the workload of JSpOC, as human analysts are needed to catalog orbital debris, recover lost satellites from uncorrelated tracks, and ensure computer calculated answers are intuitively correct.[13] Compounding this problem, the Air Force has faced several challenges concerning staffing a sufficient number of qualified analysts.[4,13]

Outside of these current operational challenges, there is also room for improved utility of the sensors themselves. As previously discussed, as sensor assets with greater capabilities are brought online, a greater number of SOs will be able to be added to the current SSN catalog. Thus, It will become increasingly likely that in the process of completing an assigned observation, a sensor asset will detect additional SOs previously unknown, such as orbital debris, in the vicinity of space around the initial target. Due to the centralized planning yet distributed nature of current SSN sensors[14] and the fast time scales involved in making SO observations,[15,16] decisions concerning follow up observations are best made locally at the sensor location, and globally on time scales that are incompatible with human in-the-loop decision making.

The large number of SOs, complex scheduling constraints, high human workloads, and short time scales are persistent challenges that are well suited to using autonomy approaches.[17] The term 'autonomy' is often used to describe any system that can operate without human intervention.[18] For the purposes of this paper, autonomy will be discussed using the three levels of Intelligent Machine Design defined by NASA's Goddard Space Flight Center (GSFC) for use in spacecraft.[18] These three levels, Reaction, Routine, and Reflection, are defined by increasing levels of autonomy. The most basic of the three, the Reaction level, is primarily responsible for processing sensor information and commanding actuators over very short time scales. The Routine level is where routine evaluation and planning behaviors occur over medium time scales. Finally, the Reflective level is where high level planning, review, and learning occur over large time scales.

The core of an intelligent agent acting at the Reflective level is its ability to reason and learn about its environment. To discuss autonomous agents that learn, this paper will also utilize the three commonly accepted types of learning. Each of these are categorized by the type of feedback available to the agent for learning. In supervised learning, the agent observes input-output pairs and learns a function which maps between them through techniques such as linear regression. In reinforcement learning, an agent is presented with either a reward or punishment after taking some actions, and is tasked with learning which actions were most responsible for the presented reinforcement. In unsupervised learning, no explicit feedback is provided to the agent to facilitate learning.[19]

Various levels of autonomy aside, sensor networks like the SSN are also examples of distributed sensor networks (DSNs). As applied to the tracking of SOs, DSNs "...may be classified as a nonlinear consensus tracking problem on a time-varying graph with incomplete data and noisy communications links."[20] For DSNs, two primary goals are the scheduling and tasking of individual sensors, as well as the fusion of track information they generate. A major consideration when designing a DSN is whether the scheduling and planning problems are completed by a centralized agent, distributed among multiple functionally or spatially distributed agents, or some combination thereof.[14] While distributed agents may be desirable from a design for robustness perspective, the varying capabilities of distributed sensor assets may result in inconsistent SO state estimates.[20] The astronomical community has investigated the distributed networking and coordination of heterogeneous telescope networks, pioneered by astronomers looking to improve observation response time to Gamma Ray Burst (GRB) events.[21]

The proposed research presents an autonomy architecture for the Raven-class telescope system, and seeks to elevate the level of autonomy from "Routine" to "Reflection."[18] The Raven program began at the Air Force Research Laboratory (AFRL) Directed Energy Directorate's Air Force Maui Optical and Supercomputing (AMOS) site. A Raven-class telescope system is not defined by a specific combination of components. Rather, it is a design paradigm where commercial off-the-shelf (COTS) hardware and software are combined to fulfill the requirements set forth by a given mission, such as precision tracking of SOs. Physically, the Raven system is a combination of several physical components: the telescope and dome, the CCD, control computer, the weather station, and a GPS receiver and timing system. While the Raven program initially started as an R&D effort, in 2001 a Raven located at the Maui Space Surveillance Site became a contributing sensor to the SSN.[11] The low cost of COTS hardware coupled with the Raven's history in R&D make it an ideal research platform for an academic institution like the Georgia Institute of Technology.

This paper will present a loose history of autonomous telescopes and highlight current opportunities for beneficial application of autonomy to SO tracking and characterization. A proposed autonomy architecture for a Raven-class Telescope (RCT) is then discussed and a notional environment learning agent is described. The autonomous agents discussed will be implemented on a Raven-class telescope at the Georgia Institute of Technology to form a multi-agent system, which will be verified by performing SSA research. This paper will then discuss how this MAS could fit into a DSN like the SSN, along with directions for future work.

APPLICATION OF AUTONOMY

Past and Current Autonomous Telescopes

Telescopes have had various levels of autonomy over the last decades, starting with the first scheduled telescopes in the 1960s.[22] The Raven system at AMOS is capable of acting at the "Routine" level of autonomy. Using the weather station sensors, it can detect nautical twilight to begin automated boot sequences and detect inclement weather to suspend operation. More impressively, given a set of tasks, it can determine an appropriate observation schedule to satisfy the human-provided objectives.[11]

An existing autonomous telescope is the Meter-Class Autonomous Telescope (MCAT), deployed to Kwajalein Atoll in order to perform orbital debris observations for both NASA and the AFRL. MCATs mission is the detection of orbital debris in both low-inclination LEO orbits and at geosynchronous earth orbit (GEO), with detection limits of 2 and 10 cm diameters respectively. Also, MCAT is intended to observe targets of opportunity alongside Kwajalein range radar stations. MCAT was designed with autonomy in mind, and is planned to execute[7] "Tasking, data acquisition, data reduction, and even some analysis aspects..." with "...minimal or optional" user intervention.

An example DSN of telescopes was Los Alamos' Telescope ALert Operations Network (TALON). The TALON network server was comprised physically of RAPid Telescopes for Optical Response (RAPTOR) telescopes interconnected using standard TCP/IP protocols. Each RAPTOR had a "client" agent which was responsible for transmitting data to a centralized server. This centralized server synthesized these individual data logs to produce follow-up observation alerts which were pushed back to all telescopes connected to the TALON.[23]

Recently, detailed thought has been applied to moving telescope network autonomy to the Reflective level. The Thinking Telescope program at Los Alamos National Laboratories has taken lessons learned from TALON and has combined additional RAPTOR telescopes with unsupervised learning techniques. The architecture consists of a vast database of observational variations from persistent sources coupled with intelligent agents. These agents learn over time to distinguish between actual gamma ray burst (GRB) events and environment noise such as airplane lights and other non-celestial phenomena.[24]

Cognition Models for Autonomy

When examining past and present autonomous systems, it is apparent that despite their different applications many systems share similar autonomous architecture. This is a result of the common goals that an autonomous system must achieve, provided with similar means to achieve these goals. As autonomous systems become more intelligent, they ultimately strive to emulate the decision making processes of humans. Of the many decision making processes, or cognition models, that exist, perhaps one of the most widely known is the Observe, Orient, Decide, Act (OODA) loop proposed by John Boyd. Boyd is historically remembered for his part in the creation of Energy-Maneuverability theory and subsequent successes with the Air Force's Lightweight Fighter program. While his theory was initially applied to military conflicts, it was later abstracted to businesses, governments, and other societal structures.[25]

A basic diagram depicting Boyd's OODA loop can be seen in Fig. 1 below. The first step, Observe, involves collecting data from the environment. The second step, Orient, describes the analysis and synthesis of these observations in order to producing a current view of impression of the world. The next step, Decide, involves planning a course of action based on the current model of reality. The last step, Act, is simply the step where action is taken in the environment. This action imparts a last effect on the environment, the results of which are then observed, thus cycling the loop anew.[25] Part of the success of the OODA loop is that Boyd supports his ideas by deriving its basic tenets from physics and mathematical principles.[26]

Boyd first utilizes Gödel's Incompleteness Theorem to posit that any "...physical systems we evolve to represent or deal with large portions of reality will at best represent or dealt with that reality incompletely or imperfectly."[26] But, he then uses Heisenberg's Uncertainty Principle and the Second Law of Thermodynamics to postulate that "... any inward-oriented and continued effort to improve the match-up of concept with observed reality will only increase the degree of mismatch."[27] He concludes that the only way to fully understand and shape an evolving, uncertain reality is to adapt the abstract models of reality, on which decisions are made, by continually cycling through the OODA loop.[27] This principle seems to be of great relevance when designing an autonomous systems that operative at the Reflective level of autonomy. In order for such a system to successfully understand, synthesize, and interact with its environment, the underlying models of physics on which it makes decisions can not be static or unchanging with time. However, of the significant amount of research conducted in machine learning, most algorithms are implemented to solve simple games with static environments, so that criteria for stability of the learning process or local optimality can be established.[28,29] Thus, there is a need for a paradigm shift in the way current autonomous systems learn.

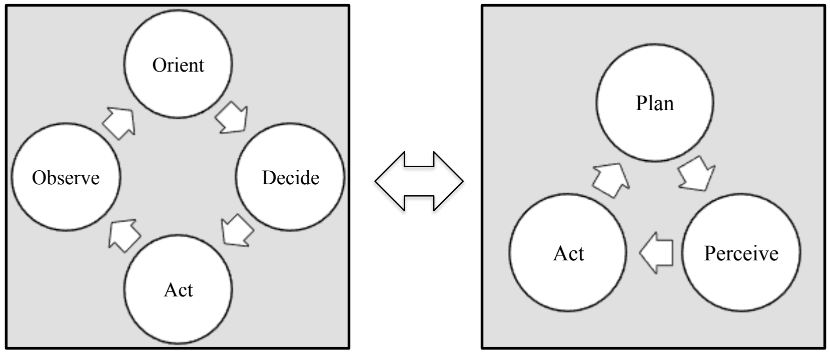

Figure 1. Cognition Models

The inherent structure of the OODA loop is given further credence by the similarity of cognition models developed by other entities. NASA Goddard describes a cognition model similar to the OODA loop, shown in Fig. 1. The only functional difference is that the Observe and Orient steps have been collapsed into a single step, resulting in an architecture containing three steps: Perceive, Plan and Act. The Perceive step involves sensing and interpreting the operating environment. The Plan step involves using information form the Perceive step to choose appropriate actions in pursuit of the goals and mission of the system. Finally, the Act step is simply the point in the autonomous cycle where action is taken in the environment.[18]

It is useful to conceptualize these cognition models in terms of a common classical control loop, as shown in Fig. 2. Here, the Observe steps can be thought of as the combination of system sensors and relevant filtering or estimating software. The Orient step involves comparing the estimate of the measured state to the reference or desired state. The Plan or Decide step uses the difference between the desired and measured state to define a course of action, and the Act step is the interaction of the actuator with the physical plant.

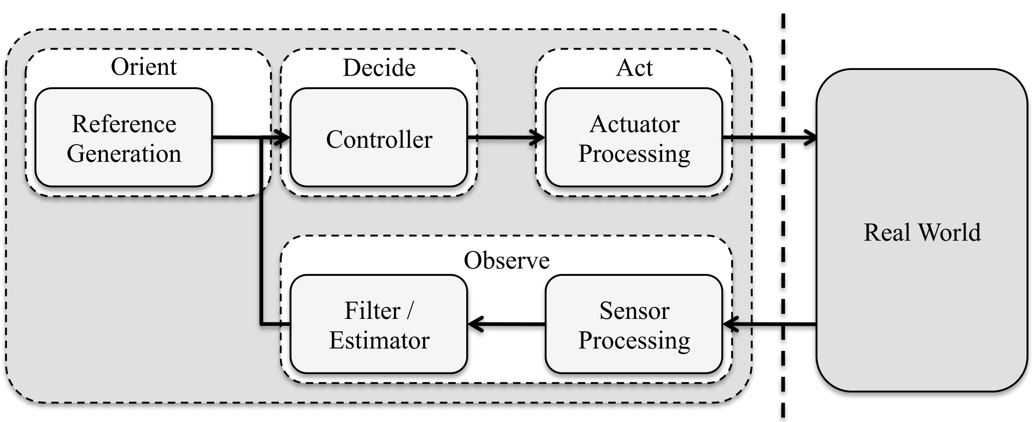

Figure 2. Cognition Model Visualized as Feedback Loop

Past and Current Autonomy Architectures

From these various cognition models spring the actual autonomy architectures which drive autonomous systems. The traditional architecture is comprise of three-tiers: the functional level, the

Table 1. **Relationship between IMD Levels, Cognition Models, and Autonomy Architectures**

IMD Levels	Cognition Model	Autonomy Architectures		
NASA IMD[18]	NASA[18]	3 Layer[30]	DARPA/ISO SARA[32]	JPL CLARAty[31]
Reflection	Plan	Planning	Mission Plane	Decision
Routine	Perceive	Executive	Hardware Plane	-
Reaction	Act	Functional	Cyber Plane	Functional

execution control level, and the decision level.[30] These tiers correlate closely to the intended functions of the previously discussion cognition models, as shown in Table 1. A variety of related three-tier architectures have been developed,[31] another of which is the Survivable Autonomic Response Architecture (SARA) sponsored by the DARPA/ISO Autonomic Information Assurance (AIA) program. The purpose of this program was to develop information assurance systems operating at the "Routine" level of autonomy to defend information systems from cyber-adversaries.[32]

A more recent example is the two layer architecture implemented in the Coupled Layered Architecture for Robotic Autonomy (CLARAty) software developed by the Jet Propulsion Laboratory for use on autonomous rovers. In their architecture the top layer is the "Decision" layer, which is responsible for breaking down high-level science goals into a plan of activities, subject to mission and resource constraints, that are executed by the "Functional" layer. The Functional layer is responsible for providing basic rover functionality which could include real-time control loops to control actuators or system level operations such as guiding a rover to a specified goal. This system breaks from the traditional three layer autonomy architecture in order to enable each layer to operate at varying levels of abstraction and blend declarative and procedural techniques for decision making.[31]

Potential DSN Architectures

Several DSN architectures for SSA have been developed by others as shown in Fig. 3, and those descriptions will be utilized in this paper as well.[14] One possible scenario is to utilize a single, centralized tasking agent and distributed scheduling agents. This configuration is analogous to the current SSN in that a centralized tasking agent has knowledge of SO covariance but only limited knowledge concerning individual sensor capability. A list of SO observations is created daily and sent to scheduling agents collocated with individual sensors. These scheduling agents have no knowledge of covariance information but do have detailed knowledge of sensor capability, and therefore use metrics such as probability of detection to allocate sensor resources. An alternative to this tasking and scheduling paradigm is one where the tasking and scheduling tasks are combined, which is sometimes referred to as "mission planning."

Thus, another scenario is to have mission planning agents spatially distributed with each sensor. This distributed mission planning scenario utilizes dynamic mission planning, which means that each mission planning agent uses the predicted outcome of an observation when scheduling the remainder of the observations to be made for the planning horizon. However, as developed by Hill et. al, the sensor does not have information on how the schedules of the other sensors will affect the catalog.

The last configuration discussed in the SSA literature is to utilize a single, centralized mission planner with knowledge of both orbit error covariance for known SOs and knowledge of sensor per-

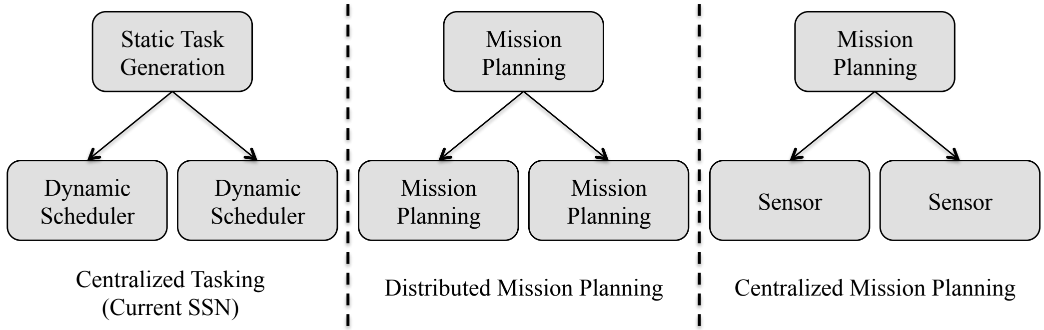

Figure 3. Possbile SSN DSN Architectures

formance. As in the distributed mission planning scenario, the mission planner is dynamic. However, unlike the distributed mission planning scenario, the centralized mission planner has knowledge on how a single scheduled observation will impact the observational utility of the remaining observations to be made for all sensors. Previous work studying which of these scenarios is optimal has yielded nuanced results, most likely as a result of the various assumptions made in preparing the simulation scenarios.

Hill et. al. found that distributed mission planning performed best at reducing track error of the most variant SOs in the catalog, but that centralized mission planning yielded a catalog with a lower median SO error. This study acknowledges that their use of homogenous sensors in the study was likely a key assumption, and that future work using heterogeneous sensors together would likely see increased benefit from using a DSN with centralized mission planning.[14] Hobson and Clarkson used only a single sensor to compare the three scenarios, and found that either mission planning scenario outperformed a centralized planning scenario, and that centralized mission planning was ultimately best in reducing catalog error.[33]

Finally, Jayaweera et. al. did not make the assumption that distributed mission planning meant that information about scheduled observations could not be shared. Rather, they proposed an algorithm where each agent fully processes its observations, then shares them with neighboring agents, making the mission planning process both distributed and collaborative. As a result, they found that a distributed architecture was most suitable from both an optimal network utilization perspective, and also from a robust standpoint.[20]

It is also helpful to consider how a DSN relates to a multi-agent systems (MAS). In the field of artificial intelligence, MAS are typically defined as a system in which the agents were not specifically designed to accomplish a common goal. As a result, a MAS is typically an environment in which multiple heterogeneous agents have separate goals, and may cooperate, compete, or coexist.[34] This also implies that each agent may not know the internal states of other agents within the MAS.[35] While a DSN may simply be thought of a specific subset of MAS where the agents are assumed to be cooperative, for a nationally sensitive task like SSA, it may be more appropriate to begin designing SSA systems with thought given to the amount of "trust" assumed between autonomous agents.

Challenges in Autonomy

As autonomous systems utilize increasingly intelligent behavior, a variety of challenges present themselves. NASA and the National Research Council (NRC) have identified several areas of autonomy research as key enablers to future success of U.S. space efforts. This includes research in the area of dynamic planning, sequencing tools, and multi-agent systems. Specific challenges with planning and scheduling tools are the ability to rapidly explore the action space, repair complex plans, and develop commands with traceability to the initial activity requirements. They also identify current challenges in multi-agent systems, including management of agent system group goal direction.[36,37]

The desire for more capable autonomous systems is also evident in recent Air Force Technology Roadmaps, which describes dozens of potential future capabilities and then selects four "Grand Challenges" to help guide Air Force Science and Technology efforts. Number 2 on the list is "Trusted Highly-Autonomous Decision-Making Systems," which is concerned with developing autonomous decision-making agents in parallel with generalized V&V methods that can be applied to a broad range of problems rather than application specific agents. There is a specific focus on systems which decrease human workload and are capable of operating at decision time scales beyond human capacity. Number 3 on the list "Fractionated, Composable, Survivable, Autonomous Systems" is concerned with designing robust, secure networks of autonomous systems almost synonymous with the definition of a DSN. The Air Force's top priority in this area is the development of advanced methods for collaborative control and adaptive autonomous mission planning. There is special verbiage to emphasis the importance of these systems in increasing mission robustness, not necessarily mission capability, through system redundancy.[12]

In order to mitigate these issues, this paper discusses a proposed multi-agent system that can reason about interdependent science and mission objectives and coordinate host sensor efforts. This paper is concerned mainly with high level autonomous activities, motivating the development of two distinct agent types: a dynamic scheduling and planning agent and a learning agent. As a space mission platform analog, a single instance of each agent type will be implemented on a Raven-class telescope for the purpose of performing SSA.

PROPOSED RAVEN-CLASS TELESCOPE AUTONOMY ARCHITECTURE

Proposed DSN and Autonomy Architecture

From the work conducted independently by Hill and Jayaweera, it was demonstrated that DSN are most efficient when spatially disparate sensors have knowledge about the mission planning activities of one another. In order to demonstrate how these gains can be realized in a system only incrementally different than the current SSN, the proposed autonomy architecture for the RCT will utilize centralized tasking, as shown in Fig. 4. Task scheduling will be completed locally at each Raven by a single, dynamic scheduling agent, referred to as the "CSP agent" as explained in the following section. The authors hypothesize that modern computers have enough computational capability to enable search algorithms to dynamically change observation schedules using only the resources of a single CSP agent. In future research efforts, the CSP agent could be made a mission planning agent. These agents could then be replicated on every sensor, creating a distributed mission planning capability that would be robust to communication link failures.

In order for the CSP agent to construct accurate schedules, knowledge about observational metrics like "probability of SO detection" and "mean time to detect SOs" are needed. This information

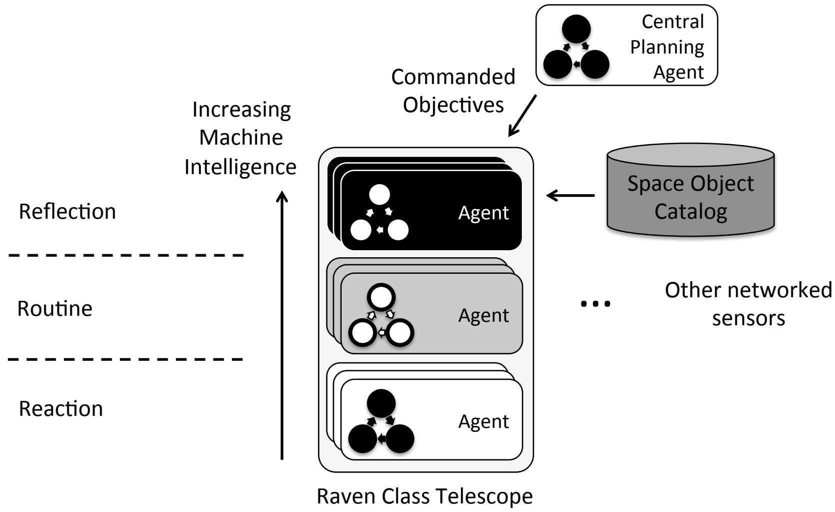

Figure 4. Proposed Georgia Tech Raven DSN

will be provided by a learning agent, referred to as an “RSM” agent as explained shortly. These agents will be developed as part of a traditional three layer architecture. The remainder of the paper is concerned mainly with the two highest levels of autonomy, and the two agents are segregated functionally as shown in Fig. 5. The RSM agent acts at the “Reflective” level of autonomy, while the CSP agent at the “Routine” level of autonomy.

Local Scheduling and Planning with Constraint Satisfaction

Within the task of SSA, the domain of possible actions and SO targets is vast, and the timescale for decision making between initial SO detection and follow up observations could be as low as fractions of seconds.[16,15] Thus, the scheduling agent must be able to rapidly explore the design space in order to locally repair complex schedules. One potential solution is to cast the planning and scheduling problem as a Constraint Satisfaction Problem (CSP), and thus this agent will be referred to as the “CSP agent.” CSPs have the advantage of using general purpose heuristics, and thus are applicable to a wide range of problems. Additionally, CSPs can identify which partial schedules are not solutions to the problem constraints and immediately remove schedules which are refinements of these infeasible solutions. This means CSPs are appropriate for problems where a solution via state-space search algorithms would be formidable.[19]

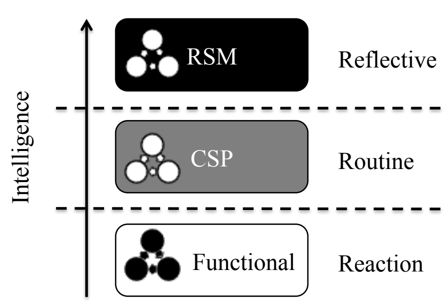

Figure 5. Intelligence hierarchy of local agents on the Georgia Tech RCT

Various CSP algorithms have been successfully used in the past to schedule astronomical observations for space telescopes such as the Hubble Space Telescope,[38] Chandra X-Ray Observatory,[39] and the Spitzer Space Telescope.[40] Also, CSPs have been utilized to identify opportunistic science targets,[41] prioritize science data downlinks,[41] as well as guide the actions of autonomous rovers.[31] A large number of these applications were built on top of the Continuous Activity Scheduling, Planning, Execution and Re-planning (CASPER) continuous planner. CASPER primarily utilizes an iterative repair search algorithm, which permits continuous modification and updating of a current working plan as well as incremental planning and scheduling.[42] Recent efforts have also demonstrated how CASPER could be modified to consider the interdependency of specified goals.[43]

The problem facing the CSP agent is to identify a sequence of activities that meet operational constraints and maximize utility of the schedule. In the Raven system, a span of time for each night, denoted $t \in [t_i, t_h]$, will be treated as a discrete planning horizon. CSPs are typically solved via the construction of action trees; a sequence of feasible decision points where each decision is termed a node. Thus, at each node there is a set of activities, $\mathbb{A}$, each associated with a required amount of time t_{Ai} and an expected utility.[19] The current notional concept of operations includes the following activities: K (known), observing known SOs, S (search), searching the regions of the sky with the highest probability of new SO detections, and L (learn), making observations in regions of the sky where little is currently known about the relationship between environmental states and telescope performance metrics. As currently envisioned, known SOs could be human provided or central planner generated objectives would be input as high-level goals and could include commands like "track catalog object number x" or "refine the orbital element estimates of SOs whose current orbital element covariance is in the highest 10% of the catalog population." With these activity definitions, the set of possible activities could be $\mathbb{A} = \{K, S, ..., L \cap S, ...\}$. The system constraints, $\mathbb{C}$, are imposed by mechanical limitations of the Raven hardware and environmental variables like weather, which could change the planning horizon length.

A multitude of algorithms exist that can be used to explore the activity tree, identify feasible schedules, and select a desirable activity schedule.[19] Using the notation developed here, a possible feasible schedule that satisfied all constraints is an ordered set $S(t_i, t_h) = \{L, H, ...L \cap S\}$. It is common for many such schedules to satisfy all the constraints, therefore utility functions will be used to determine which ordered set of events is preferable. While this utility based methodology has been extensively proven, each CSP is application specific, so one contribution of the proposed future research will be the development of novel utility functions that are both accessible to multiple agents and appropriate for the SSA problem. Initial utility functions could seek to maximize the total number of detections while minimizing false positive and false negative detections, or weight the importance of observing objects with the highest uncertainty in orbital ephemeris.

The CSP agent described implemented on a RCT would be an ideal testbed where various algorithms could be tested for timeliness in exploring action spaces and proficiency at repairing intricate plans. However, to trace telescope actions to initial requirements, it is necessary for the CSP agent to have knowledge about the current environment states and their relationship to each Raven performance parameter of interest. Examples include "probability of successfully detecting target A given current cloud cover" for action H or "mean time to detect new SOs in a specified region of the sky" for action S. One possible method to generate this information is constructing an elaborate, physics-based model, specific to the particular Raven's components and local environment. This is not always possible or desirable, as sophisticated models are often computationally expensive and may need to be calibrated for every Raven in the proposed multi-telescope system. Additionally, there are not always analytical relationships between dynamic environmental variables of interest and telescope performance. Thus, the next goal of the proposed approach seeks to rectify this with the development of a learning agent, which rigorously combines information theory and systems engineering principles to develop a physics-based model that is both flexible and computationally efficient.

Learning with Response Surface Methodology

Thus far, it has been shown that in order to form effective models of the environment in which autonomous systems act, the system must be capable of constantly adapting to a changing reality. Historically, this has meant that autonomous systems have been built using machine learning techniques with a variety of assumptions, such as static environments, so that criteria for stability of the learning process or local optimality can be established. However, this has limited their usefulness in real world applications.[28,29,34,35,44] Therefore, in order for the intelligent agents of an autonomous telescope to be capable of learning about the ever-changing world they operate in, new machine learning techniques are needed.

This paper proposes that the techniques developed in the fields of systems engineering can be leveraged to form a cross-cutting autonomous solution to serve as the basis for learning agent. The proposed learning agent utilizes Response Surface Methodology (RSM), which encompasses design of experiments (DoE) and response surface equations (RSE) techniques, and will be referred to as the "RSM agent." The area of artificial intelligence most closely associated with these techniques is called "active learning."[45]

The basic concept of active learning is that a machine learning algorithm can achieve greater predictive capability with fewer sample data points in its training set if it is allowed to select which data points it uses to learn. The common assumption often made as the motivation for the development of these techniques, is that an abundance of unlabeled data is available or easily obtained, but the correct labels for these data points is difficult, time consuming, or expensive to obtain. Thus, most techniques seek to solve an unsupervised learning problem where the machine makes queries by choosing which data points it wishes to learn with, and then the correct labels are provided by an "oracle," typically a human operator.[45]

Before discussion the application of RSM, it is helpful to define some standard terms and their synonyms for those unfamiliar with these techniques. A "design variable", also termed a "factor" or "parameter," is a quantity varied during the experiment. Design variables can be discrete or continuous and are typically represented as an $nx1$-dimensional vector **x** . The "design space" is the n-dimensional space defined by the upper and lower bound of each design variable. In many classical DoE texts the design space is scaled from -1 to 1 or from 0 to 1, which was a convenience for representing sample DoE tables, as well as a mathematical necessity for avoiding ill-conditioned matrices when generating DoE samples. Today, DoE generation is conducted almost exclusively via computer programs, so it is not a concern of the human analyst to normalize design variable ranges. A "sample," also synonymous with "design point" and "point," is a specific instance of each design variable **x** and is usually represented an an ordered n-tuple of the form $(x_1, x_2, ..., x_n)$. Finally, the "response" is the dependent quantity being measured by the experiment for a specific sample.[46]

Response Surface Methodology "...comprises a group of statistical techniques for empirical model building and model exploitation."[47] RSM approximates the relationship between the design variables and the response over the entire response space, not unlike a supervised learning process in the field of artificial intelligence.[19] This approximation, termed the "response surface" or "response surface equation" (RSE), usually takes the form of a low order Taylor Series Expansion.[48,49] While the RSE could take other forms, such as a Fourier Series, empirical evidence has shown that in most cases a 2nd order Taylor Series approximation is sufficient.[49] Synonyms for response surface equation include "model," "metamodel," or "surrogate model."[48] Finally, it should be noted that the statistically community uses the term response surface to denote the true unknown relationship between the design variables and the response. In this case, the term "response surface approximation" is used to describe the assumed form of the relationship between the design variables and the response.[46]

RSM has been used to characterize and study responses, determine design variables which result in an optimal response, select design variables which result in a robust response, and for the prediction of new responses.[49,50] RSM allows the modeling of responses to design variables when no empirical relationship is known, and the simplified equations of RSM allow for rapid evaluation and exploration of the design space. With an assumed polynomial form there is a known minimum number of data samples necessary to create an RSE. For a second-order polynomial, the number of samples is given by $samples = (n + 1)(n + 2)/2$. In RSM, the exact samples to be used are selected through a design of experiments.

Generally, a DoE is a process where purposeful changes are made to input variables such that resultant changes in output response may be observed and identified using a minimum number of experiments.[48] These methods were popularized in work determining optimum conditions for chemical processes,[51] but have since had an enormous impact on variety of scientific and engineering studies.[49] However, since these techniques have been used in such a variety of fields, some confusion has developed with regards to both terminology, as discussed above, and the appropriate use different DoE designs.[46]

Perhaps the most overlooked facet of DoE is the fact that classical DoE methods, which include Box-Behnken and Central Composite Designs, were developed for industrial processes which assumed the processes being analyzed were inherently non-repeatable. As a result, these designs typically select sample points at the extremes of the design variable ranges, resulting in more reliable trend extraction from non-repeatable experiments. However, modern DoE methods were developed with deterministic computer simulated experiments in mind. Hence, these designs, such as sphere-packing designs, assume there is no random error in the sample data and place sample points throughout the interior of the parameter space in an effort to minimize bias error.[46]

Other computer generated DoE include "optimal designs", which produce DoE that maximize the optimality criteria of the DoE. Optimality criteria are characterized by the letters of the alphabet, and hence are sometimes referred to as alphabetic optimality criteria.[49] Perhaps the most popular of these designs is the D-optimal design,[45,46,49] which maximizes the determinant of the Fischer information matrix, reducing the square of the volume of the confidence region on the regression coefficients.[49] Table 2 demonstrates the efficacy of two different DoE techniques compared to a full factorial exploration of the design space when a 2nd order Taylor Series approximation is used.[48]

In the Raven system each experiment can be an observation, and in order to determine the effect of an input variable, like sky brightness, on the probability of SO detection, a response, observations could be taken at different levels of sky background brightness. A response surface equation could then be fit to the collected sample points to form a physics-based model of the affect of sky brightness on SO detection, for the purpose of providing performance information to the CSP agent.

Table 2. Comparison of Required Experiments

DoE	Experiments for $n = 8$ Variables	Equation
Full Factorial	6561	3^n
Central Composite	273	$2^n + 2n + 1$
D-Optimal	45	$(n+1)(n+2)/2$

Data available on other Ravens suggests that the required amount of data could be collected in a few nights.[11]

Unfortunately, the traditional DoE techniques discussed above are not suited to the Raven system for two main reasons. The first is many of the variables which contribute to telescope performance are not controllable by the RSM agent, but are properties of the SO, such as angular velocity, or environmental properties, such as light pollution in the background sky.[16,15] The second is that these variables are not all independent of one another, a violation of classical designs. One solution is to formulate the task of collecting observations as an unsupervised learning process. Recent work has shown how non-orthogonal DoE designs, such as D-optimal designs, can be found from large databases, suggesting the DoE process can be autonomous.[52] Since no algorithm currently exists to explicitly find orthogonal plans in non-binary data, genetic algorithms (GA) and kernel methods have been utilized to find nearly orthogonal data points.[53] While promising, these results were obtained offline, with no consideration given to computational efficiency, motivating further investigation to prove utility for an operational Raven system. Interesting areas of further inquiry would be the implementation of alternative heuristic search patterns, such as particle swarm optimization (PSO).[54] The general claim that PSO has the same capability to find the global optimal solution as a GA but with significantly better computational efficiency has been demonstrated.[55] Additional items of interest include determining whether designs based on alternative alphabetic optimality criteria[49] are quicker to find, and what threshold of orthogonality is necessary to construct accurate RSE.

It should be noted that the RSE process described here has two limitations. The first is the requirement that the state spaces in question are homogenous, comprised of either all continuous or all discrete variables. The second is that the dimensionality of input variables allowed for model development be limited to approximately 10. However, a common method for circumventing this limitation is known as an "effects screening test", which, analogous to Pareto's Law, has qualitatively shown that approximately 20% of the input variables account for 80% of the variability of the response, ensuring its applicability to systems with a greater number of states.[48] Other challenges include automating the process of evaluating the "goodness of fit" metrics of the RSE, a task typically completed by human operators. Metrics for evaluating the accuracy of RSE include the coefficient of determination, actual by predicted and residual by predicted plots, as well as Model Fit Error and Model Representation Error. Common remedies for poor fit, including logarithmic transformations and the inclusion of higher order terms, are rigorously defined and thus well suited to automation.[48] While both fixes necessitate extra computational cost, the second calls for additional observations to be performed.

The proposed RSM agent combines information theory and systems engineering methodologies to autonomously select observations to approximate the relationship between various environmental parameters and telescope performance for which no analytical models exist. The RSE generated are physics-based and can change in a real world dynamic environment. This enables the CSP agent to predict future performance, justifying why certain actions were selected over others in order to satisfy initial requirements. Also, evaluation of the proposed RSEs are computationally inexpensive, ensuring their applicability to problems with fast timescales, like new SO detection.

RESPONSE SURFACE METHODOLOGY EXAMPLE

An example is now provided to further clarify the RSM agent concept, and demonstrate how the RSM agent could supply the CSP agent with the probability of successfully observing a known SO. The architecture for this example is shown in Fig. 6. Fig. 7 depicts a typical cloudy night in Atlanta, GA. These pictures, taken in the span of 4 hours on January 6th 2013, show how rapidly environmental factors like cloud cover and aircraft overflights can change SO viewing opportunities. These pictures were collecting using the AllSky340 manufactured by SBIG. The AllSky340 uses a 640x480 pixel Kodak KAI-340 CCD, and a F/1.4 Fujinon fisheye lens. The AllSky340 was connected to a laptop via RS-232 and was programmed to automatically take pictures in two minutes intervals.

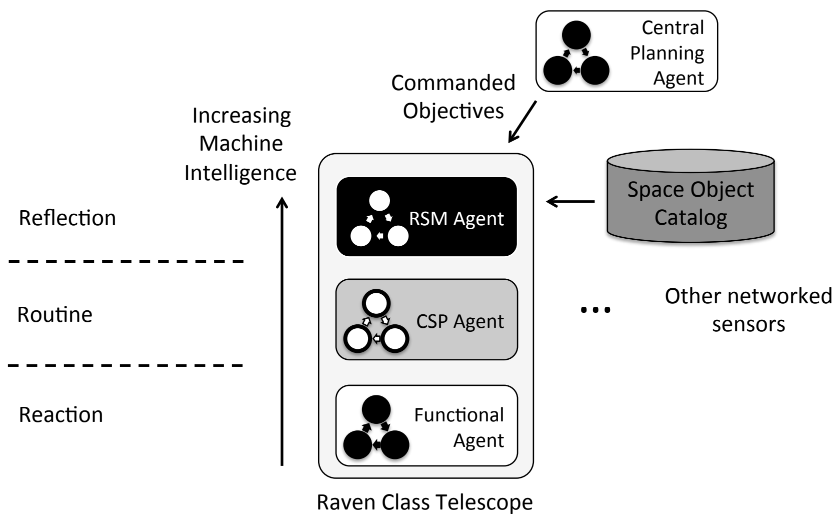

Figure 6. Example Raven DSN

To ensure the CSP can make the best decision possible in a highly dynamic environment, it is necessary for the RSM agent to calculate the probability of successful SO observation as a function of environmental variables not under control of the Raven. One way to quantify the probability of detection is to utilize the signal to noise ratio (SNR) of the optical observation of the SO. It should be stressed that SNR here refers to the ratio of the average signal value to the standard deviation of the background signal.[15] Eq. 1 below gives a simplified form of the SNR as developed by Shell.[16]

$$SNR = \frac{E_{SO} \cdot \tau_{atm} \cdot \tau_{opt} \cdot A \cdot QE \cdot t_{int}}{\sqrt{(L_b \cdot \tau_{opt} \cdot A \cdot QE \cdot t_{int} \cdot \mu^2) + e_n^2}} \tag{1}$$

In this equation the quantity E_{SO} is the irradiance of the space object, the transmittance of the atmosphere and optics are represented by τ, A is the aperture of the telescope, QE is the quantum efficient of the detector, and μ is the instantaneous field of view. The signal integration time is t_{int} and e_n is the read noise of the detector. Many of these parameters, including the quantum

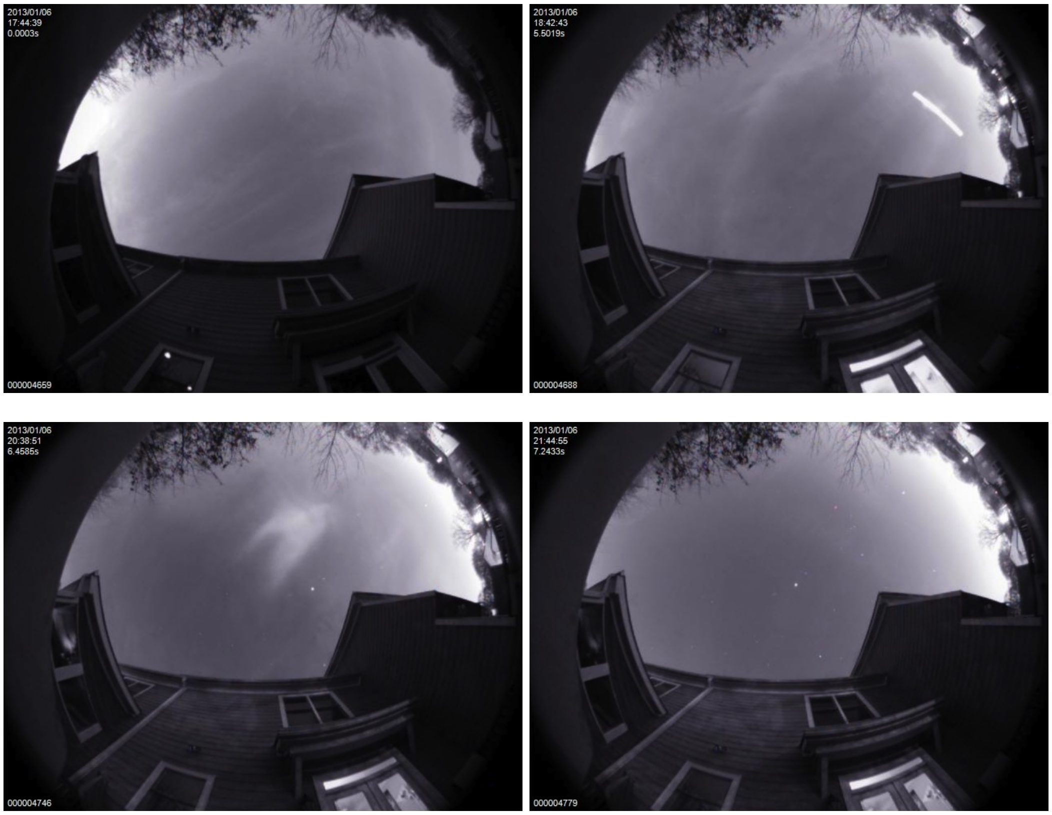

Figure 7. Changing conditions of the night sky in Atlanta, GA

efficiency of the detector and the transmission efficiency of the optics are fixed, known parameters. However, quantities like atmospheric transmittance are both time varying and spatially dependent. While sophisticated codes like MODTRAN are available for computing atmospheric transmittance, they are dependent on a large number of variables, many of which could not be measured without additional sensors.[56] These codes are also not capable of quantifying the effect of transient noise sources, such as clouds or airplanes, on atmospheric transmittance. Thus, the problem of quantifying the probability of successful SO detection as a result of the uncertainty in atmospheric transmittance due to these environmental parameters is an ideal demonstration of the utility of RSM.

In order to measure the atmospheric transmittance experimentally, the Raven would utilize a standard catalog of stars as a reference. The Raven could select a set of stars form this catalog, observe one star at a time, and then attribute the difference in catalog brightness from measured brightness to both optic efficiency, a known quantity, and atmospheric transmittance, the unknown quantity of interest. To efficiently select a subset from the standard catalog, the DoE methods discussed previously will be used. The first step is establish which factors most likely influence the atmospheric transmittance. In this example, it will be assumed that the azimuth and elevation angles, sky brightness, and cloud cover have the greatest effect on atmospheric transmittance. These independent variables are shown in Table 3 along with their anticipated range of values. The two independent variables under control of the Raven are the azimuth and elevation angles, and thus will be the factors in the DoE. In our example, it will be assumed that the star field is dense enough that for any combination of azimuth and elevation angles, a catalog star will be available in the Raven's

Table 3. Raven RSM Design Variables

Factor	Anticipated Range (Units)
Azimuth (α)	$0 - 360$ (deg)
Elevation (δ)	$0 - 90$ (deg)
Background Radiance (L_b)	$14 - 21$ $(mag/asec^2)$
Cloud Cover (C)	$(C(\alpha, \delta))$ $(\%)$

Table 4. Compiled DoE Table

Obs.	α $(degrees)$	δ $(degrees)$	L_b $\left(\frac{mag}{asec^2}\right)$	C (α, δ) $(\%)$	τ_{atm} $(\%)$
#1	45.5	36.9	17	.58	.6
#2	160	45.6	18	.2	.65
...					
#15	185	50	19	.43	.7

FOV. This scenario is known as "query synthesis" in the active learning literature.[45]

To minimize the number of observations necessary for building this model, thus freeing up time for operational observations, one of the optimal DoE designs, such as a D-optimal design, could be used. In the field of active learning, a query strategy that synthesizes new samples to reduce the variance of the model in exactly the same way as a D-optimal design is appropriately named a "variance reduction" strategy.[45] Because the initial assumption was that the relationship between these variables and the SNR of the observation is quadratic the minimum number of stars to observe is 15. For each observation, the value of each environmental variable will be recorded along with the atmospheric transmittance of the captured image containing the SO. Because the background sky brightness and percent cloud cover variables are not controllable, the value at the time of observation will simply be recorded along with the response, and the compiled results would appear as they do in Table 4. Please note that all data in this example is meant to be representative only and has not been taken from actual observations.

With the subset of observations selected, the linear least squares process can begin, creating the RSE. If every possible variable and resulting interaction term was included in the RSE, this example RSE would have 15 terms. This RSE can now be used for the purposes of predicting future performance of the Raven system when tasked with observation known SOs. Fig. 8 illustrates the completed RSM process, where the cartoon stars signify the catalog stars used in the DoE, and the brighter contours represent areas of greater atmospheric transmittance, which could be obscured by cloud cover.

In this case, the azimuth and elevation angles may be known precisely, but other variables like background radiance or cloud cover could be assigned probability distributions that, through Monte Carlo simulation, yield a cumulative distribution function of the atmospheric transmission during a future SO observation. This distribution can be used directly with the SNR given by Eq. 1 to yield a cumulative distribution function of the SNR of a future SO observation. This cumulative distribution function, combined with a user selected minimum required SNR, allows the probability of successfully conducting the observation of the known SO, as shown in Fig. 8. In this example, the minimum required SNR was specified as six, which by looking at the CDF, one can see that this implies about a 80% chance of successfully observing the SO.

CONCLUSION

A loose history of autonomous telescopes was presented along with two common cognition models that functionally decompose tasks common to autonomous systems. It was found that these two cognition models were very similar, despite development by two different organizations. Autonomy architectures that were developed based on these common tasks were discussed, along with possible SSN DSN architectures. These autonomous architectures were found to be very similar, empirically demonstrating that a traditional 3-layer autonomy architecture was suited to a diverse range of problems, including telescope autonomy. The brief survey of previous SSN DSNs work demonstrated that spatially disparate sensors are most efficient when knowledge about mission planning activities are shared between one another. Using these findings, the proposed DSN and autonomy architecture for a Raven-class Telescope (RCT) was then described. The proposed autonomy architecture seeks to utilize the traditional 3-layer architecture, wherein increasingly intelligent agents are built to handle increasingly complex tasks. Additionally, the DSN architecture proposed represents an incremental change from the current SSN, where a dynamic scheduling agent termed the “CSP” agent is used to perform local schedule repair. The functions of each intelligent agent were detailed along with requisite background information. Here, consideration was given to solving the dynamic mission planning problem as a constraint satisfaction problem. Additionally, cutting edge methodologies from both systems engineering, information theory, and machine learning were synthesized into a single intelligent agent, termed the “RSM” agent. The specific implementation of these agents was explained using a tractable SSA example, where the MAS of a single Raven is tasked with learning the dynamic, local atmospheric transmittance and the resultant impact on probability of SO detection.

FUTURE WORK

The implementation of the CSP and RSM agents on a single Raven will constitute a multi-agent system. This will serve as a platform to study the complex inter-agent dynamics typical of a MAS. For example, the RSM agent could learn that observations taken in the region of sky between 30° and 50° elevation have historically resulted in a greater number of previously unknown SO detections. Accordingly, the CSP agent repairs the current plan to include more observations in this region of the sky, resulting in greater system utility.

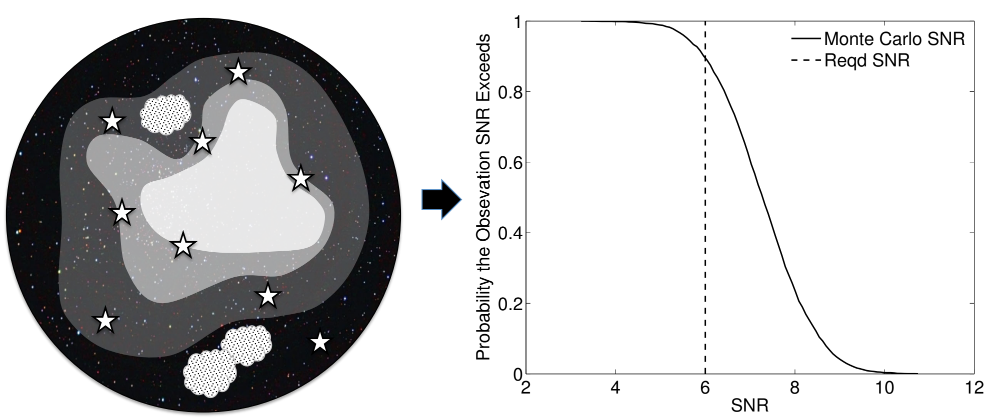

Figure 8. RSM Process and Resulting SNR CDF

Another useful investigation would be to use dynamic planning and scheduling to maximize the efficacy of the MAS when the observational objectives have interdependent utilities. An illustrative example is the case where a SO is predicted to flyby a high value space asset and the orbital elements of the SO have large variance. If an observation to refine these orbital elements is conducted before the time of the anticipated rendezvous, this action has a high utility value in the CSP. But, if scheduled to occur after after the expected flyby, this action has little utility.

The tight coupling between the CSP and the learning process suggests that another challenge is that of distributed learning. The Raven MAS is well suited to concurrent learning, as each Raven is independently learning and refining its performance predictions to improve overall team utility.[35] While previous work has shown how cooperative intelligent agents can share sensory feedback, resultant actions, and learned decision policies in order in improve MAS performance,[44] a central problem in concurrent learning is that each intelligent agent is adapting its behaviors using new, learned knowledge simultaneously.[35] Of particular interest is how the autonomous DoE approach discussed earlier could be broken down into autonomous subtasks for multiple agents.

REFERENCES

[1] D. H. Rumsfeld, "Commission to Assess United States National Security Space Management and Organization," january, Committee on Armed Services of the U.S. House of Representatives, 2001.

[2] J. C. o. Staff, "Space Operations," Tech. Rep. JP 3-14, January 2009.

[3] G. Stokes, C. Von Braun, R. Sridharan, D. Harrison, and J. Sharma, "The space-based visible program," *Lincoln Laboratory Journal*, Vol. 11, No. 2, 1998, pp. 205–238.

[4] P. D. Nielsen, K. T. Alfriend, M. J. Bloomfield, J. T. Emmert, Y. Guo, T. D. Maclay, J. G. Miller, R. F. Morris, A. B. Poore, R. P. Russell, D. G. Saari, D. J. Scheeres, W. P. Schonberg, and R. Sridharan, "Continuing Kepler's Quest: Assessing Air Force Space Command's Astrodynamics Standards," tech. rep., National Research Council, Washington, D.C., September 2012.

[5] P. Kervin, J. Africano, P. Sydney, and D. Hall, "Small satellite characterization technologies applied to orbital debris," *Advances in Space Research*, Vol. 35, No. 7, 2005, pp. 1214–1225.

[6] *The Space Report 2011*. Colorado Springs, CO: Space Foundation, 2011.

[7] M. Mulrooney, P. Hickson, E. G. Stansbery, and E. S. Barker, "Orbital Debris Detection and Tracking Strategies for the NASA/AFRL Meter Class Autonomous Telescope (MCAT)," *61st International Astronautical Congress*, No. IAC-10.A6.1.5, Prague, September 2010.

[8] S. Kan, "China's Anti-Satellite Weapon Test," crs report for congress, Congressional Research Service, The Library of Congress, April 2007.

[9] T. S. Kelso, "Analysis of the Iridium 33-Cosmos 2251 Collision," *Advanced Maui Optical and Space Surveillance Technologies Conference*, September 2009.

[10] J.-C. Liou, "Update on Three Major Debris Clouds," 2010.

[11] C. Sabol, K. K. Luu, P. Kervin, D. Nishimoto, K. Hamada, and P. Sydney, "Recent Developments of the Raven Small Telescope Program," *AAS/AIAA Space Flight Mechanics Meeting*, Vol. AAS 02-131, 2002, p. 397.

[12] "Report on Technology Horizons A Vision for Air Force Science and Technology During 2010-2030," tech. rep., Office of the Chief Scientist of the U.S. Air Force (AF/ST), 2010.

[13] B. Weeden, "Going Blind: Why America is on the Verge of Losing its Situational Awareness in Space and What Can be Done About it," tech. rep., Secure World Foundation, 2012.

[14] K. Hill, P. Sydney, K. Hamada, R. Cortez, K. Luu, M. K. Jah, P. W. J. Schumacher, M. Coulman, J. Houchard, and D. Naho'olewa, "Covariance-Based Network Tasking of Optical Sensors," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Wailea, HI, September 2010.

[15] T. Schildknecht, "Optical Astrometry of Fast Moving Objects Using CCD Detectors," *Geodätisch-geophysikalische Arbeiten in der Schweiz*, Vol. 49, 1994.

[16] J. R. Shell, "Optimizing Orbital Debris Monitoring with Optical Telescopes," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Space Innovation and Development Center, September 2010.

[17] D. S. Board, "The Role of Autonomy in DoD Systems," tech. rep., U.S. Dept. of Defense, 2012.

[18] W. Truszkowski, L. Hallock, C. Rouff, J. Karlin, J. Rash, M. G. Hinchey, and R. Sterritt, *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*. Springer, 2009.

[19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd ed., 2009.

[20] S. K. Jayaweera, R. S. Erwin, and J. Carty, "Distributed Space Situational Awareness (D-SSA) with a Satellite-assisted Collaborative Space Surveillance Network," *18th World Congress of the International Federation of Automatic Control*, Milano, August-September 2011.

[21] R. White and A. Allan, "An overview of the heterogeneous telescope network system: Concept, scalability and operation," *Astronomische Nachrichten*, Vol. 329, No. 3, 2008, pp. 232–236, 10.1002/asna.200710958.

[22] A. J. Castro-Tirado, "Robotic Autonomous Observatories: A Historical Perspective," *Advances in Astronomy*, Vol. 2010, No. 570489, 2010.

[23] R. R. White, J. Wren, H. R. Davis, M. Galassi, D. Starr, W. T. Vestrand, and P. Wozniak, "TALON: the telescope alert operation network system: intelligent linking of distributed autonomous robotic telescopes," 2004, pp. 302–312, 10.1117/12.549438.

[24] W. T. Verstrand, D. Heath, W. James, P. Wozniak, N. Ben, R. White, J. Bloch, E. Fenimore, B. Hogge, M. Jah, and R. Rast, "Autonomous Global Sky Surveillance with Real-Time Robotic Follow-up: Night Sky Awareness through Thinking Telescopes Technology," *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2008.

[25] F. Osinga, *Science, Strategy and War*. PhD thesis, Leiden University, 2005.

[26] J. Boyd, *A discourse on winning and losing*. 1987.

[27] J. Boyd, "Destruction and creation," *A Discourse on Winning and Losing*, 1976.

[28] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *arXiv preprint cs/9605103*, 1996.

[29] L. Busoniu, R. Babuska, and B. D. Schutter, "A Comprehensive Survey of Multiagent Reinforcement Learning," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Vol. 38, IEEE, 2008, pp. 156–172.

[30] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *The International Journal of Robotics Research*, Vol. 17, No. 4, 1998, pp. 315–337.

[31] T. Estlin, R. Volpe, I. Nesnas, D. Mutz, F. Fisher, B. Engelhardt, and S. Chien, "Decision-Making in a Robotic Architecture for Autonomy," *In Proceedings of the International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, 2001.

[32] S. Lewandowski, D. Van Hook, G. O'Leary, J. Haines, and L. Rossey, "SARA: Survivable autonomic response architecture," *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, Vol. 1, IEEE, 2001, pp. 77–88.

[33] T. Hobson and I. Clarkson, "Sensor-scheduling simulation of disparate sensors for Space Situational Awareness," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Vol. 1, 2011, p. 60.

[34] B. Chaib-Draa and F. Dignum, "Trends in Agent Communication Language," *Computational Intelligence*, Vol. 18, 2002.

[35] L. Panait and S. Luke, "Cooperative Multi-Agent Learning: The State of the Art," *Autonomous Agents and Multi-Agent Systems*, Vol. 11, Springer, 2005, pp. 387–434.

[36] S. C. f. NASA Technology Roadmaps, "NASA Space Technology Roadmaps and Priorities: Restoring NASA's Technological Edge and Paving the Way for a New Era in Space," tech. rep., National Research Council of the National Academies, 2012.

[37] R. Ambrose, B. Wilcox, B. Reed, L. Matthies, D. Lavery, and D. Korsmeyer, "Robotics, Tele-Robotics and Autonomous Systems Roadmap Technology Area 04," tech. rep., National Aeronautics and Space Administration, April 2012.

[38] M. D. Johnston, "Spike: AI Scheduling for NASA's Hubble Space Telescope," *Artificial Intelligence Applications, 1990., Sixth Conference on*, IEEE, 1990, pp. 184–190.

[39] R. J. Brissenden, "Chandra X-ray Observatory Operations," *Astronomical Data Analysis Software and Systems X, ASP Conference Series* (F. R. J. Harnden, F. A. Primini, and H. E. Payne, eds.), Vol. 238, 2001, pp. 22–31.

[40] S. Tyler, J. O'Linger, S. Comeau, L. Garcia, W. Mahoney, D. McElroy, and D. Mittman, "Rapid replacement of Spitzer Space Telescope sequences: targets of opportunity and anomalies," *Proceedings of SPIE, Observatory Operations: Strategies, Processings, and Systems II* (R. J. Brissenden and D. R. Silva, eds.), Vol. 7016, 2008.

[41] R. Sherwood, S. Chien, D. Tran, B. Cichy, R. Castano, A. Davies, and G. Rabideau, "The EO-1 Autonomous Sciencecraft," *Small Satellite Conference*, Logan, UT, August 2007.

[42] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Improve the Responsiveness of Planning and Scheduling," *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, 2000.

[43] T. Estlin, D. Gaines, F. Fisher, and R. Castano, "Coordinating Multiple Rovers with Interdependent Science Objectives," *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005.

[44] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," *Proceedings of the tenth international conference on machine learning*, Vol. 337, 1993.

[45] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, 2010.

[46] A. Giunta, S. Wojtkiewicz, and M. Eldred, "Overview of modern design of experiments methods for computational simulations," *Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2003-0649*, 2003.

[47] G. Box and N. Draper, *Empirical model-building and response surfaces.* John Wiley & Sons, 1987.

[48] M. R. Kirby, *A Methodology for Technology Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design*. PhD thesis, Georgia Institute of Technology, March 2001.

[49] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley and Sons, 3rd ed., January 2009.

[50] D. Mavris, "Design of Experiments and Introduction to Response Surface Methodology," Advanced Design Methods 1, Fall 2011 Lecture Notes.

[51] G. Box and K. Wilson, "On the experimental attainment of optimum conditions," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 13, No. 1, 1951, pp. 1–45.

[52] C. Pumplün, S. Rüping, K. Morik, and C. Weihs, "D-optimal plans in observational studies," *Komplexitätsreduktion in Multivariaten Datenstrukturen*, 2005.

[53] S. Rüping and C. Weihs, "Kernelized design of experiments," *Komplexitätsreduktion in Multivariaten Datenstrukturen*, 2009.

[54] G. N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design*. Vanderplaats Research and Development, Inc, 4th ed., 2005.

[55] R. Hassan, B. Cohanim, O. De Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," *Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference*, 2005.

[56] D. Robertson, A. Berk, and L. Bernstein, "MODTRAN: A Moderate Resolution Model for LOWTRAN 7," 1989.