SSC21-XII-01

Constellation Forecasting Tools for Autonomous Operations

Alexander Dunn, Joseph Miceli, Michael Reher
Redwire Space
Address: 9092 S. Ridgeline Boulevard, Littleton, CO 80129; Phone: +1 303-904-6060
Email: alex.dunn@redwirespace.com

ABSTRACT

Large constellations are quickly becoming the norm in small satellite missions. These constellations are being designed and developed faster than ever before through the utilization of smaller, heterogeneous spacecraft. Often, these constellations provide increased resiliency and capabilities over their heritage, highly tailored counterparts. The ability to replace on-orbit assets quickly and with lower costs is an advantageous feature of these large smallsat constellations. With the advent of these new architectures, though, come increased complexity in mission operations. The management and monitoring of potentially hundreds of heterogeneous space assets can be extremely challenging and negate much of the cost savings using current operational approaches. Additional complexity is added with the loss expectancy of some number of assets inherent to the design within these constellations. Rather than tasking individual assets to complete missions on behalf of the system, ideal operation would be conducted through tasking of the constellation as a whole. This approach requires tasking of the individual assets by the constellation using machine-to-machine (M2M) data sharing and on-orbit autonomous decision making. Recent advances in machine learning (ML) and artificial intelligence (AI) have now set the stage for the state of the possible in this regard.

The authors of this paper are part of a research and development team aiming to develop solutions and tools to support this operational approach. The ideas presented involve a procedural and technical implementation of using forecasted operational effects developed by a combination of state machines and ML tools. First, the system's state is gathered, time-synced, and produced into a "Dynamic Relative Telemetry Calculator." This is presented as an NxN matrix documenting each node's state relative to all other nodes in the system. Next, a desired operational command can be loaded into the system. Multiple possible operational scenarios and effects can be propagated. For each propagation, each asset must be capable of reporting the "cost" of performing a certain task within a certain operational scenario. By itself, this still requires a human in the loop to analyze the results and determine a command decision. However, the secondary and tertiary effects of these decisions are still unknown. To this front, the authors are developing a method of wrapping ML capability around the system's state machine and propagators to create a forecaster capable of autonomously determining optimal decisions within a system. The forecaster operates in real time, improving its predictions as more data is produced by each subsystem. Generated operational forecasts, and their effects, are validated with log data from a simulation. This data is being applied to proprietary mission scenarios, but could also be applied to historical open/mission data for validation or operational lessons learned. Over time, this forecasting tool could optimize large constellation management by reserving human in the loop for only the most severe/impactful decision thresholds. This paper will present current progress of the integrated solution, next steps in the research and development roadmap, and, most importantly, the current technical hurdles still to overcome to achieve true spaceflight autonomy.

INTRODUCTION

It is well documented and evident the current trajectory of the global spacecraft industry is investing in large operational constellations. These operational systems promise to enable game changing ideas such as Space Dial Tone and Internet of Things (IoT), and show real promise to further connect the globe and provide internet access to even the most remote locations on Earth. Historically, the market direction in satellite systems is driven by US Department of Defense (DoD) needs. However, the shift to large, robust constellations is now being driven by commercial entities. The commercial market has identified the business case for developing operational constellations. Most famously are two large commercial constellation endeavors intending to provide global internet access: Starlink and OneWeb.

Elon Musk and his SpaceX team are in full-deployment mode in building out the Starlink infrastructure in space. At the time of this paper, there are ~1,500 deployed Starlink satellites of a potential 12,000 (planned) to 42,000 (extended) satellites operating at once. Similarly, the OneWeb constellation (co-owned by OneWeb and Airbus) also promises "connection for people all over the globe."[1] At the writing of this paper, 146 of 650 satellites have been launched. OneWeb's constellation has already demonstrated significant progress in global engineering cooperation by transitioning the manufacturing from France to US locations.

With the commercial industry leading the way, the US DoD has recognized the opportunity to leverage this private investment to upgrade and expand their own capabilities. The Defense Advanced Research Projects Agency's (DARPA) Blackjack program "seeks to incorporate commercial sector advances in low Earth orbit (LEO), including design of LEO constellations intended for broadband internet service, of which the design and manufacturing could offer economies of scale previously unavailable."[2]

The Space Development Agency (SDA), only recently established in spring 2019, aims to develop the National Defense Space Architecture (NDSA). Their objective is to deliver minimum viable products to the warfighter on-time.[3] The SDA is committed to leveraging commercial capabilities in order to build out this proliferated space architecture. The final product will be a combination of procured commercial spacecraft and a Hybrid Architecture approach of tying in to existing space infrastructure for data acquisition. To reduce data latency from space to warfighter, the SDA is expected to invest in a Transport Layer consisting of 300-500 spacecraft from multiple commercial vendors.

By all accounts, competing commercial entities, such as SpaceX and OneWeb, will drive innovation for our industry. The addition of the US DoD's promise of support as anchor tenants will certainly accelerate innovation timelines, especially in manufacturing, interoperability, and efficient operations.

The team of authors here have previously presented their thoughts, expertise, and opinions on technical enablers for manufacturing and interoperability. This paper intends to focus on the third enabler: efficient operations. It is appropriate the theme of the 2021 SmallSat Conference is Autonomy and Mission Operations, as all evidence suggests efficient operations of large constellations must include some level of autonomy. For this paper and the conference at large, the authors intend to investigate the achievability of autonomous operations in large constellations from a number of vectors. The authors will first describe and define the problem statement regarding efficient operations and why autonomous operations are necessary. They will discuss the positives and negatives, the current limiting factors, and the current state of autonomous operations. Finally, the authors will discuss some of the technical roadmaps for autonomy of interest to their research.

PROBLEM STATEMENT

It is not difficult to perceive issues and limitations in mission operations as constellations begin to scale to the extremes as described in the previous section. With potentially thousands of nodes in any given system, traditional human in the loop operations must give way to more efficient autonomous operations. A number of factors in these large systems will drive the need for autonomy on orbit. The most commonly discussed need for autonomy is time related. Autonomous operations will be needed for most time-critical scenarios. This is evident in both the commercial world, such as data streaming for commercial products, but also highly emphasized in military applications. Reducing data latency is a key objective for multiple DoD organizations when supporting the warfighter. Hybrid Architectures and Mesh Networks promise to do this through proliferation of nodal access points for the warfighter, but autonomy will also be needed to further optimize the critical data path.

The next key argument for autonomous systems is massive reduction in operational costs. Autonomy can reduce costs through a number of ways, including by the obvious of eliminating the need for large teams of ground operators and support staff. Cost is also reduced through increased mission performance and efficiencies. Improved efficiency offered by autonomous systems means the cost per data product ratio can be minimized. For large constellations, the scale of data products created means massive profit and returns are achievable via autonomous operations.

The commercial industry has already recognized autonomous operations as a necessity for achieving both operational efficiency and profitability. In May of 2021, SpaceX announced the Starlink spacecraft will avoid collisions with other objects in space through autonomous operations.[4] The satellites will use their on-board propulsion systems to autonomously conduct avoidance-maneuvers for perceived collision threats based on objects tracked by the US military. The referenced article outlines, at a high level, the procedure for issuing avoidance maneuvers. Based on a ground based measured collision probability, the US Air Force will issue a conjunction alert. However, these alerts are then manually reviewed and assessed before a corrective action is deemed necessary. SpaceX's strategy is to remove the manual assessment and send the conjunction alert and necessary information directly to the affected spacecraft for an autonomous maneuver. This strategy will certainly save time and cost regarding the collision assessment, but may end up costing the spacecraft unnecessary propellant expulsion to satisfy a non-optimized threshold. Collision probabilities and resulting conjunction alerts will only increase as large constellations continue to populate LEO, meaning these maneuvers will cost the spacecraft significant margins in propellant, resulting in substantial cost increases for servicing and/or replacement. Optimizing the thresholds and rules driving autonomous operations is a critical element for future autonomy in space.

The US DoD also recognizes the need for autonomy, funding large space programs to address potential solutions. The first key program objective for DARPA's Blackjack program is to "develop payload and mission-level autonomy software and demonstrate autonomous orbital operations including on-orbit distributed decision processors."[2] These autonomous functions are strategically implemented to ultimately reduce the time between data collection to delivery to the warfighter. The autonomy operations in Blackjack and other similar DoD programs are attempting to process collected data, determine the authenticity and relative importance of data, build a useable data product in a communicable format, and deliver to a warfighter without human in the loop.

In the Civil Space sector, NASA has identified the need for autonomous capabilities in space. Specifically, NASA's upcoming deep space missions drive the urgency and reliability of autonomous operations. There is a significant time delay in spacecraft to ground communications when flying in deep space. For traditional spacecraft operations, which require ground controllers to send tasking based on spacecraft telemetry and status checks, these time delays can quickly exceed reliability thresholds. Further risk is added to these missions if a time delay significantly reduces the validity in spacecraft telemetry and state of health.

NASA established the Distributed Spacecraft Autonomy (DSA) project with the objective of advancing autonomous operations capabilities for NASA Missions. DSA efforts "will advance command and control methodologies for controlling a swarm of spacecraft as a single entity, demonstrate autonomous coordination between multiple spacecraft in the swarm, and demonstrate approaches for adaptive reconfiguration of the swarm's plan."[5] DSA is planning a small on-orbit demonstration of four CubeSats in 2021 and a ground test scaling up to 100 spacecraft. This approach shows NASA's perception of scalability as a critical requirement for autonomy to be successful in deep space missions, both from a cost and efficiency for operations.

CURRENT MISSION OPERATIONS PRACTICES

Mission operations encompasses both pre-launch and on-orbit functions necessary for mission success. Whether for single vehicle missions or large constellations, mission operations can be very complex with a broad range of requirements for success. Within the space industry, there are no standard approaches for mission operations. NASA and the DoD derive different requirements based on objectives for their diverse portfolio of missions resulting in very different approaches to mission operations. Similarly, within the commercial market, most entities develop custom mission operation approaches. Not only is the commercial industry limited by diverse mission requirements, but proprietary and intellectual property protection surrounding mission operation concepts and tools also contribute to a lack of publicly available information. However, when discussing these limitations with colleagues within the industry, whether or not isolated mission operation capabilities would be sustainable in support of Hybrid Architectures and interoperable constellations is often a point of discussion. While there will always be a need for and business case to maintain some mission operation concepts privately, for autonomous operations and autonomous constellations to integrate into Hybrid Architectures, the industry will undoubtedly be required to re-imagine how mission operations are conducted and the procedures potentially shared.

Though it can be reasonably predicted that future mission operations for autonomous constellations may be unrecognizable by today's implementations, it is still important to recognize the current state of constellation mission operations. As stated previously, there are many functions performed as part of mission operations. Figure 1, from *Space Mission Analysis and Design* (SMAD), identifies 13 standard functions in mission operations.[†6]

[†] SMAD is a useful and standard resource for Aerospace Engineers (both students and professionals). It follows the mission design of a single vehicle mission known as FireSat. Often, SMAD is used as a first reference when conducting research into a space system.

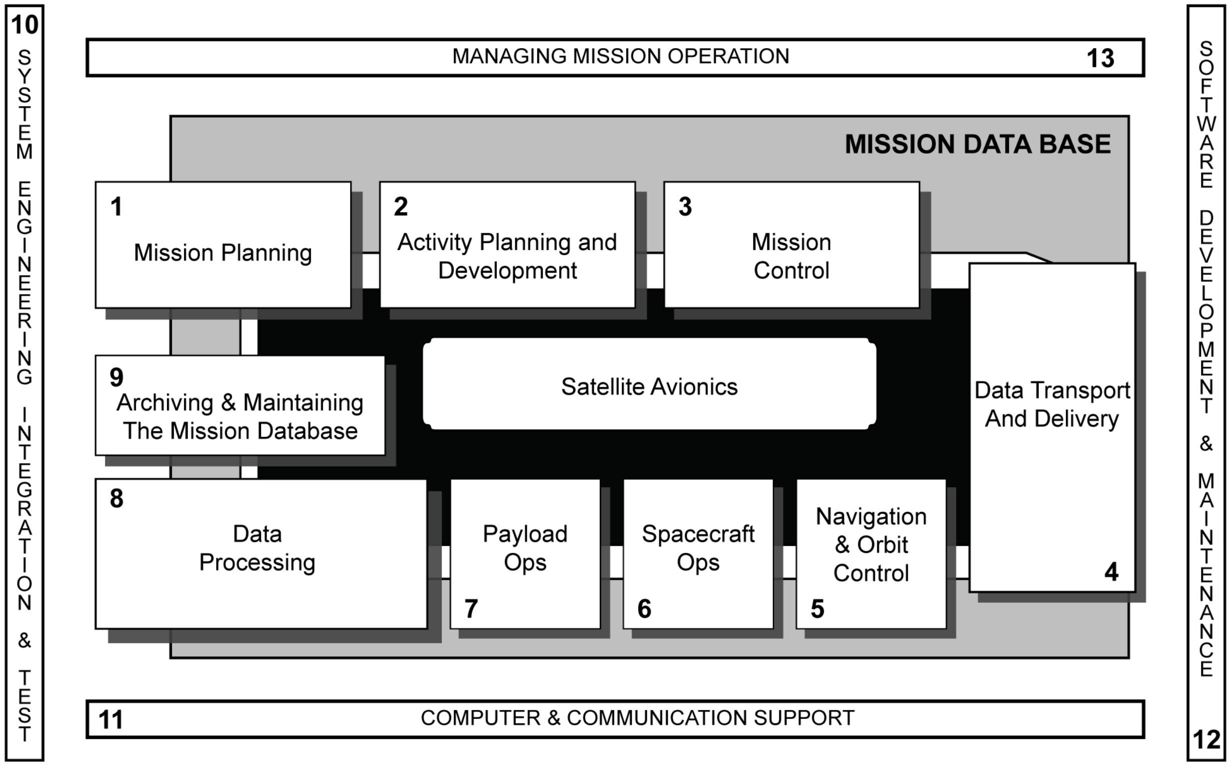

Figure 1. The 13 Functions of Mission Operations System (MOS) and How They Interact[6]

In this example, nine functions rely on the mission database and data sharing for appropriate actions to be considered. Further, Table 1 shows, in a traditional mission operations plan, whether these functions occur within the spacecraft avionics (autonomous) or within the ground operations (human in the loop).[6]

It is evident from the figure and table that humans in the loop are often required and relied upon in standard operations. Requiring human in the loop in this capacity has obvious negative effects on time-derived decision making by ultimately adding latency in delivering data to an end user. For time critical missions, this is not reliable. Similarly, the cost of maintaining humans in the loop for these operations is significant. When considering the scaling needs of large constellations, this is simply not sustainable.

Many of the mission operations functions rely on the host spacecraft avionics for the data, and all require some sort data processing. As is evident, some data management as well as guidance, navigation and control functions may well be automated in many systems today, but a vast majority of the mission operations functions in this table still rely on ground operators and humans in the loop. These functions make up a vast majority of the mission operations while in flight and are most often the functions looked at for automation. If these functions can be conducted on-board a spacecraft, significant time costs in communications windows and ground decisions can be eliminated.

Table 1. Identifying Where to Carry Out Functions[6]

MOS Function	Where Accomplished	
	Spacecraft Avionics	**Ground**
Mission Planning	-	Operator augmented with automated tools is primary
Activity Planning and Development	-	Operator augmented with automated tools is primary
Mission Control	-	Operator is primary
Data Transport and Delivery	Many LEO telecommunications spacecraft implement much of this function onboard	Software and hardware provide primary capabilities
Navigation and Orbit Control	Software and hardware on spacecraft is an option	Software and hardware is primary
Spacecraft Operations	-	Short- and long-term planning by operators, augmented with automated tools
Payload Operations	-	Short- and long-term planning by operators, augmented with automated tools

When shifting from single vehicle missions, such as the FireSat mission in SMAD, to much larger systems as discussed earlier, human in the loop operations represent drastic increases in costs both financially and temporally. Unsurprisingly, complexity and cost are exponentially coupled for space systems as documented by IEEE Aerospace Conference Paper "Cost and Risk Analysis of Small Satellite Constellations for Earth Observation."[7] Considering this, it should be explicitly evident that a method for decoupling mission cost from mission complexity is needed for the success of future Hybrid Architectures and Mesh Networks. Further analysis of cost estimating methodologies helped to determine a significant way of reducing mission costs would be achieved by operating the constellation as a single unit (which is not always practical in Hybrid Architectures) or increasing the use of automated operations on orbit.[8] Autonomous operations will reduce the number of personnel and ground support equipment needed to maintain traditional operations. Advances in autonomous operations are critical to future systems, and will drastically reduce mission cost by shifting many of the tasks in the "Ground" column of Table 1 into the "Spacecraft Avionics" column.

AUTONOMY LIMITATIONS

Autonomy has been a desired trait of satellite missions for many years, with multiple scientific discovery missions flying "autonomous operations" as payloads. However, it must be noted the significant difference between automation and autonomy. Automation is not "self-directed," but instead requires command and control.[9] Autonomy, in comparison, is the ability of a system to achieve goals while operating independently of external control.[10]

In many space applications, the term "autonomy" is used to reference a rules-based state machine either on board the spacecraft or at the mission operations center. The state machine is able to perform actions that have been encoded into the state machine so long as the transition criteria for those actions have been met. While this does represent a useful level of automation, a significant limit to these state machines is that only rules and outcomes that have been encoded can be acted upon. True autonomy, then, could only be accomplished by encoding every possible rule and outcome that the system could encounter. Preprogrammed state machines are obviously limited in fault mitigation use cases; preprogramming specific faults and mitigation actions for a constellation of spacecraft would quickly become too cumbersome.

A good example of this is NASA's Space Technology 5 (ST5) mission. A 3-ball constellation launched in 2006, ST5 was an early scientific mission seeking to test the ability of "smart" satellites by validating software tools developed for "autonomous ground operations," among other mission objectives.[11,12] Within its 100-day mission, ST5 operated with a "lights out" period during which the constellation executed pre-programmed commands without ground input. Though the mission validated the automated operations, which was certainly a major technological achievement, the ST5 mission encountered a number of unforeseen anomalies and ultimately required the activation of an "Anomaly Team" four hours after launch to mitigate anomalous sun sensor data. In NASA's ST5 mission, the limitations of automated operations using a rules based state machine method were immediately apparent. The mission was not able to adjust to unforeseen and un-programmed anomalies once on orbit. True autonomy will not be fully realized until the methods, procedures, and infrastructure for mission operations progress and evolve to a critical point necessary to recognize, identify, and react to anomalies without human in the loop. While this mission presented significant advances in automation, the goal of a truly autonomous system would be to recover from anomalous data without human intervention.

A major challenge in transitioning from automation to autonomous systems is the ability to enable the system to be self-aware and understand its current state and the state of its surroundings. The true state and awareness of a satellite's mission has most always been defined by a ground operator based on telemetry and satellite health data received while on mission. This status drives the next line of actions or tasking for the spacecraft. Self-awareness is also important for a spacecraft to adjust to unforeseen anomalies. The ability to identify and learn from new anomalies introduced into the environment will be necessary. For a satellite to accurately observe and define its current state, a few things must occur. Advance sensor technology, providing highly detailed, highly accurate, and highly discernible data, must be available to the satellite's on-board computer (OBC). Fortunately, significant advances in technology are now enabling highly accurate data products to be delivered directly to the OBC. Similarly, this raw data must also be processed and reduced into useable data products. This requires powerful on-board computing.

Scalability is also a major challenge in autonomous operations. To date, most on-orbit autonomous operations consist of a relatively low number of nodes within the system (see ST5 and DSA examples above). While verification and validation on smaller systems is necessary, the ability to scale to support the large systems must also be verified and validated. Data sharing and data routing between nodes, especially autonomously, is a very difficult challenge to solve. To solve this, significant ground testing and virtualization should be utilized.

AUTONOMY ENABLERS

Fortunately, interest in small spacecraft technology has been surging for years. This rapid growth has enabled countless advancements in space technology and space autonomy. Autonomy at the individual spacecraft level has become a popular research topic. Autonomous trajectory planning algorithms typically reserved for terrestrial robotics applications are finding their way to space. Spacecraft motion planning algorithms are being developed for optimal slews and even rendezvous and docking maneuvers.[13,14] Trajectory planning algorithms often operate in a similar manner; they attempt to connect the current state to the goal state. They typically do this by constructing a graph of states by sampling inputs and propagating the state forward. They then find the optimal path that connects the initial state to the goal state according to some criteria.

Though these algorithms are focused on solving autonomy challenges for individual spacecraft, their framework will be a key enabler for solving autonomy challenges for constellations. The major challenges to utilizing these methods will be defining cost functions for constellations and developing computationally efficient propagation algorithms to construct and search the graph.

Implementation of these techniques will undoubtedly be computationally intensive. Luckily, spacecraft on-board computing capabilities are advancing rapidly. Already, the industry is seeing small spacecraft carry more data-heavy payloads.[15] The industry will continue advancing the computational capabilities as more and more use cases are found for them. Effective integration of artificial intelligence (AI) with space data is key to realizing value within vast datasets collected. Furthermore, combining AI with cutting-edge machine learning (ML) capabilities will facilitate the advancement of on-orbit data processing, increasing the efficiency of smallsat constellations. Many in the industry are already championing AI/ML as the future driver for autonomous operations. In fact, it is widely accepted that AI/ML tools will play a major factor in future autonomy. However, there are still some limiting factors involved with AI/ML, many of which are being addressed and perhaps discussed at this year's SmallSat conference.

AUTONOMY RISKS

Autonomy is not a miracle technology. The mathematics and computer science supporting autonomous systems are often not novel, however decision-making architectures do present fault types that can be unfamiliar to software and computational systems. David Atkinson of the Institute for Human and Machine Cognition presents five processes from which fault modes may arise in autonomous systems:[16]

1. Goals and Goal Generation
2. Inference and Reasoning
3. Planning and Execution Control
4. Knowledge and Belief
5. Learning

Utilizing the configurable methods presented in this paper, the authors also believe that a sixth process exists that may lead to fault modes: *Problem Definition*.

Optimal trajectory planning algorithms are not aware of the application in which they are being applied. They are only successful when their models are implemented with a sufficient level of detail and their cost functions properly represent the goal that is actually trying to be achieved.

Defining a cost function for a given objective may be extremely difficult. If not defined properly, the task plan that successfully optimizes the cost function may not result in the desired behavior of the constellation. In this case, the task plan would still be "optimal." Engineering expertise is still required to configure the autonomous system to produce the desired outcomes.

The balance between over and under constraining the system is delicate. Over defining cost functions or adding unnecessary variables to the cost function imposes unnecessary constraints on the system's ability to identify tasks. Alternatively, if necessary state variables are left out of the cost function definition, the resulting task plan would likely negatively impact these parts of the state. Variables need to be intelligently considered when designing cost functions for each specific objective.

Therein lies the risk of autonomous systems. It is easy to place too much trust in autonomous systems. If their application and operational environment is not properly considered, mission operators may find that their constellation management system is performing outside the bounds of its expected behavior.

AUTONOMY ROADMAP

Current internal research and development (IR&D) efforts at Redwire Space are focused heavily on developing tools to enable the design and test of autonomous constellation operation systems, specifically in the realm of optimal orbital trajectories and attitude control. The goal of such a system would be to design tasks that meet mission objectives while maximizing the performance metrics related to these objectives. These metrics could be constellation health, data quality, information gain, etc.

To utilize the approach of modern trajectory planning algorithms, two key areas must be developed. First, a formal definition of cost functions must be defined. Both a qualitative and quantitative definition of cost is fundamental to optimizing a task plan for a constellation. Second, a method of state propagation must be developed that can utilize this definition of cost function. The propagator servers a major function in trajectory planning algorithms that is responsible for determining the state that would result from a given task.

Constellation Cost Functions

Describing the health of any complex system is complicated and often context dependent. Generally, health of a system can be defined as a qualitative measure of how well that system is meeting its objective. For example, a spacecraft may be capable of tracking a desired target on the surface of Earth but unable to power a sensor to collect images of that location. In the context of health, its attitude control system is qualitatively "healthy" but its electrical power system is "unhealthy." Moreover, because the spacecraft is incapable of meeting a specific mission objective (to image that target on the surface of Earth), the spacecraft is "unhealthy." This is one example of determining vehicle health; however, most examples are not this straight forward, and require knowledge of many independent variables across subsystems in order to estimate the vehicle health. To add, there may be several ways to meet a certain objective, and some objectives that are related to one another. It is possible that the constellation can meet an objective (or set of objectives) but requires a significant amount of resources to do so.

Redwire defines the health of a system as **the weighted deviation of the system's state from the "goal" state of a given objective while meeting conditions for that objective**. In the context of optimization, the term health is used interchangeably with cost. Cost functions can be defined to represent a quantitative status of a component, a complex subsystem on board the spacecraft, a cluster of spacecraft, or a constellation as a whole. To add, these cost functions must be defined in the context of each mission objective. Different objectives likely require different resources from the constellation. The systems engineering process can be leveraged to define mission objectives and connect them to cost functions. Metrics like Measures of Merit (MOMs), Measures of Performance (MOPs), and Measures of Effectiveness (MOEs) can quantify "goal" states for a given objective as well as the conditions that must be met for this objective to be relevant. That is, if a mission objective conditional is met, then its cost function can be evaluated. Formally, cost functions are defined as:

$$If\ C_n^*$$

$$Then: J = \left|\boldsymbol{W}\Delta\hat{X}\right|$$

Where

$$\Delta\hat{X} = \frac{abs\left(\vec{X^*} - \vec{X}\right)}{\left|\vec{X^*} - \vec{X}\right|}$$

$$\boldsymbol{W} = \begin{bmatrix} W_1 & 0 & \dots \\ 0 & W_2 & 0 \\ \dots & 0 & \dots \end{bmatrix}$$

And C_n^* is a data structure containing the conditionals of the nth objective. The cost is not evaluated if these conditionals are not met.

Example Cost Function and Conditional

An example cost function of a single spacecraft is now considered. Two simplified objectives are defined to illustrate the creation of conditionals and goal states. The first is to maintain its current orbit and power levels. The second is to image a certain area of the Earth. The conditional for objective 1 (C_1^*) is always true. The conditional for objective 2 (C_2^*) is true if the spacecraft is in a "surface imaging" mode. The variables and weights for each objective are defined as:

Table 2. Goal State for Objective 1

Variable Name	Goal Value, $\overrightarrow{X^*}$	Weight
Angular velocity	(0,0,0) rad/sec	(1,1,1)
Incoming Charge	100 W	0.01
Total Power Load	20 W	0.01
State of Charge	100 %	0.001

Table 3. Goal State for Objective 2

Variable Name	Goal Value, $\overrightarrow{X^*}$	Weight
Target Acquired	1 boolean	5
Attitude Error	(0,0,0) rad	(10,10,10)
Angle To Target	0 rad	2
State of Charge	100 %	0.5

First the cost is calculated assuming that the conditions for objective 1 are true but the conditions for objective 2 are false. To calculate the instantaneous cost at a given time, a state vector is obtained as:

$$\vec{X} = \begin{bmatrix} -0.0000622527 \\ -0.00140791 \\ 0.00104871 \\ 94.9206 \\ 21.67223 \\ 0.842347 \end{bmatrix}$$

The instantaneous cost is then determined to be:

$$J = \left|\boldsymbol{W}\widehat{\Delta X}\right| = 0.001093$$

Once the conditional for objective 2 is met at a later time, the cost function for objective 2 can be considered. To do this, the goal state of objective 2 (denoted with subscript "coll") and its corresponding weights are simply augmented with the goal state and weights of objective 1 (denoted with subscript "nom") to form a new goal state and weight matrix.

$$\overrightarrow{\Delta X} = \begin{bmatrix} \overrightarrow{\Delta X_{nom}} \\ \overrightarrow{\Delta X_{coll}} \end{bmatrix}, \boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_{nom} & [0] \\ [0] & \boldsymbol{W}_{coll} \end{bmatrix}$$

To evaluate the instantaneous total cost, a new state is obtained for objective 1 as:

$$\vec{X}_{nom} = \begin{bmatrix} -0.00076523 \\ 0.000426711 \\ -0.00015981 \\ 73.4568 \\ 36.4278 \\ 0.76 \end{bmatrix}$$

And objective 2 as:

$$\vec{X}_{coll} = \begin{bmatrix} 0.0 \\ 0.0515264 \\ 0.00383636 \\ 0.0363993 \\ 0.401426 \\ 0.76 \end{bmatrix}$$

The complete cost is then calculated as:

$$J = \left|\boldsymbol{W}\widehat{\Delta X}\right| = 0.16342$$

Knowing the largest element of $\boldsymbol{W}$, the cost can be contextualized. The closer the cost is to 0, the smaller the deviation between the goal and the current states. Similarly, the closer the cost is to the largest element of $\boldsymbol{W}$, the unhealthier the spacecraft is. Lower cost (J) implies a healthier space vehicle. In the previous example, given how close the cost is to 0 (and how far the cost is from the maximum value of 10), the spacecraft is fairly healthy. This assessment agrees with intuition given how close the spacecraft's state is to the goal state.

Redwire's Advanced Configurable Open-system Research Network (ACORN) is being leveraged as a high fidelity spacecraft simulation to support the development of this cost function tool. Cost functions will be implemented within ACORN's flight software as a module that ingests telemetry and uses it to evaluate user-defined conditionals and cost functions in real time. Monte Carlo simulations will be executed to evaluate the behavior of the cost function under different circumstances and in response to different commands.

The definition of cost functions is not trivial and is further complicated when considering an entire constellation. Telemetry from individual components or subsystems can vary wildly in values and ranges and have very different meanings. To add, individual telemetry points may not carry enough information alone to evaluate the health of anything. Telemetry points might need to be combined in order to establish meaningful evaluations of health. An additional dimension of telemetry is added at the constellation level as relative telemetry between spacecraft.

Redwire's Dynamic Relative Telemetry Calculator (DRTC) (Figure 2) tool is a constellation monitoring application that ingests raw telemetry from a constellation and aggregates the data to show how the spacecraft are behaving relative to one another. The tool calculates telemetry like range, range rate, and pointing angles. Redwire is actively developing this tool to add additional telemetry that can be used to describe the state of the constellation as a single entity. This relative telemetry data serves as additional dimensions in the constellation's state space. This allows cost functions to be described as combinations of individual spacecraft states and constellation states. Being able to describe the state of the constellation as single unit (rather than a collection of individual spacecraft states) is vital to constellation management. Task planning of a constellation requires definition of the constellations state and cost functions. If properly defined, trajectory planning algorithms can be used to identify tasks for the constellation as a function of tasks for individual spacecraft. The primary objective of the DRTC is to provide the mission operator with as much insight into the constellation's state as possible. An ability to use this information in the task planning loop ensures that the mission operator is unconstrained when configuring their autonomous constellation operation system.

Constellation State Prediction

The second major effort of development in support of designing autonomous constellation operation systems is developing an effective propagation method. Such a method needs to be both accurate to ensure the results are useful and computationally efficient to ensure it can operate in highly complex dynamic environments.

A key challenge in developing an efficient propagation method is in the size of the state space of a constellation. A cost function that utilizes 20 different elements of relative telemetry from a constellation requires the propagation of at least 20 dimensional states (some variables may require propagation of additional states to be determined). Sampling-based trajectory planning algorithms often utilize numerical integration methods such as Runge-Kutta to propagate states forward. This method works well for lower dimensional systems and can be used in online planning.[17] In designing a tool for constellation planning where an indefinite number of variables can be used to define a cost function, the dimensionality of the state space is essentially unbounded. Therefore, the propagation method must be efficient in potentially large problem areas. Traditional numerical integration methods will not be efficient enough to support this kind of complexity.

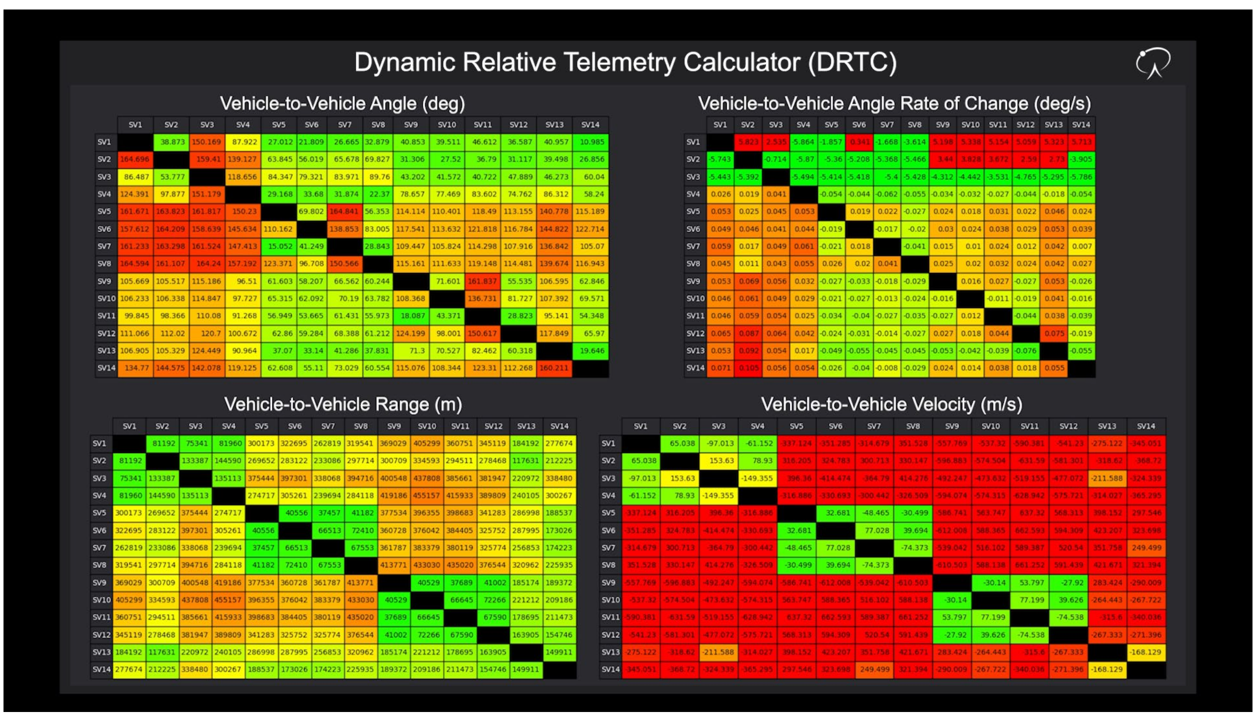

Figure 2. Dynamic Relative Telemetry Calculator (DRTC)

Redwire, in collaboration industry partners, is working to develop a prototype propagator that is capable of using current spacecraft subsystem states to predict future subsystem states with a high level of accuracy. The forecaster operates in real time, improving its prediction as more data is produced by each subsystem. The goal of this effort is multifaceted:

1. To understand what features are necessary to predict a desired forecast metric
2. Develop a machine learning model that can be used to predict a desired forecast metric
3. Develop a method for generating training data for a machine learning model

In this case, the metric of interest was "time until sunpoint." This metric was defined only for a single spacecraft as the time required to find the sun during a power generation maneuver.

ACORN was again used as the high fidelity spacecraft simulator. A Monte Carlo simulation loop was used to configure and execute the ACORN simulation 500 times. Log data from these simulations served as the input to the ML model. Each simulation varied the spacecraft's initial orbit conditions while keeping the physical configuration identical between runs. By changing the initial orbit conditions, each simulation resulted in a different amount of time required to find the sun. The physical configuration of the spacecraft was kept constant under the assumption that the resulting ML model would be spacecraft-specific. The goal of the model was to predict future states of a specific spacecraft using that spacecraft's telemetry.

Interestingly, some simulations produced results with infinite time until sunpoint. This meant that the simulation ended before the spacecraft actually found the sun. These data sets were not removed from the training set.

The model ingested the log data one element at a time (essentially mimicking what it would do if it were ingesting real time telemetry data on board the spacecraft) and produced an updated prediction after every ingestion. The complete list of spacecraft telemetry used to train the model is proprietary; however, it is representative of common telemetry sets. The results of the model's prediction for one of the simulations are shown in Figure 3.

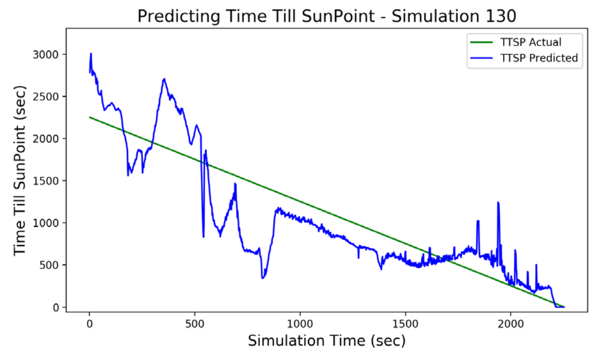

Figure 3. Example Result of Machine Learning Model to Predict Time Until Sunpoint

At t=0s for this simulation, the actual time until sunpoint was 2,320 seconds. The exact performance metrics of the model are proprietary but clearly the model's performance increased with time. Unsurprisingly, as simulation time converges to the time that sunpoint is achieved, the predicted value of time until sunpoint approaches the correct time. However, the value in a model like this is not in its accuracy at the end of the task but instead in its accuracy far from the point of interest. At t=1000s, the model predicted that time until sunpoint was 1,108 seconds; a difference of 212 seconds. A task planner could use this model to identify a task in the future under the assumption that the spacecraft is 1,108 seconds away from finding the sun. The task planner could then modify the plan using updated predictions until the task is actually executed. In this example, the planner might identify how long to operate a payload knowing how long the spacecraft's batteries must last before they can be recharged.

Spacecraft configuration data was intentionally NOT used to train the ML model. Doing so may have improved its prediction but would have made the model too dependent on the physical configuration of the spacecraft to be useful in a heterogeneous constellation. It is believed that the prediction results are representative the model's performance regardless of spacecraft configuration, however, more testing is planned to verify this. This effort successfully accomplished its three initial goals. A preliminary machine learning model and training pipeline was created that can predict a desired metric with an acceptable level of accuracy. Features needed to train this model were also identified, though more work is required to identify how models will be trained to predict other metrics.

Mission Scaling and Virtualization

Another critical component to achieving autonomy is the ability to thoroughly test and validate autonomous systems via ground testing platforms. Redwire has a strong heritage of building ground support test and modeling and simulation systems for space applications. However, the large scaling required to support mega constellations pushed the Redwire team to identify new solutions.

Redwire still relies on its ACORN platform as its base for representing space nodes in customizable levels of fidelity. ACORN can be implemented in many variations, each designed to enable scaling from mission concept design to full flatsat integration. These variations include an ACORN-S (Simulation), ACORN-R (Rack), ACORN-MT (Modular Testbed). For large constellation analysis, ACORN-VMs and ACORN-Cloud are offered in a virtual machine environment to quickly instantiate large quantities of simulated satellites.

Additionally, ACORNs are intended to be interconnected in a network to create a Distributed/Collaborative Architecture. This enables distributed engineering teams, whether geographically separated, organizationally separated, or both. ACORN has demonstrated this ability on multiple Redwire programs where the program team consists of engineers located throughout the United States at various Government Labs, Federally Funded Research and Development Centers (FFRDCs), or Government organizations. The networked ACORNs provide a secure and distributive environment for the program teams to actively collaborate on common Design Reference Missions (DRMs). This capability will be important for ground testing autonomous operations for Hybrid Architecture systems, such as SDA's NDSA or DARPA's Blackjack program.

Virtual machines on an ACORN network can be easily configured to create many spacecraft nodes within a constellation mission. This includes both homogenous and heterogeneous constellations. Establishing an ACORN network allows users to model and simulate multiple spacecraft buses within a single scenario. In this scenario, users can assess a potential autonomous flight software solutions on multiple satellites of varying classes in a single constellation.

However, the infrastructure required to host these VMs quickly became overwhelming. Through internal trade studies, Redwire has identified a commercially available server technology as the leading candidate for addressing this problem. The identified solution was required to facilitate a virtualization platform and has been architected utilizing native Kubernetes for application modernization. This innovative approach helps to decouple runtime (performance) and infrastructure (flexibility) services as key trading metrics in traditional server technology. Utilizing this service, Redwire has demonstrated the ability to virtualize complex laboratories, such as those modeling Hybrid Architectures, while optimizing both system agility and system efficiency.

CONCLUSIONS

The intent of this paper was to systematically assess and research the state and projections of the industry in terms of the future need and probability of autonomous operations. The large constellations and Hybrid Architectures will ultimately require some levels of autonomous operations in order to reduce data latency and improve operational efficiency to support the economies of scale desired. The commercial industry is driving this innovation, though the DoD and NASA are actively participating and supporting the development through various programs.

Undoubtedly, future autonomous mission operations will most likely be unrecognizable as compared to traditional mission operations. A more open sharing of operational procedures is predicted in order for Hybrid Architectures to operate autonomously. To achieve this, many novel tools and procedures must be developed.

For future work, the authors will seek to continue their research by combining their propagation tools with cost function development. This will predict a single telemetry value as well as cost for a collection of states. From there, the authors will seek to develop an optimizer to continuously produce a task list that minimizes the predicted collection of states.

Finally, the authors hope to open conversations with other industry partners seeking to establish autonomous operations. More can be achieved through open collaboration as opposed to historical closed door development. For Hybrid Architectures and Mesh Networks to truly work utilizing autonomous operations, engineers across the industry will need to communicate and share ideas and research to ensure the systems are interoperable.

REFERENCES

1. Airbus, OneWeb Satellites constellation. https://www.airbus.com/space/telecommunications-satellites/oneweb-satellites-connection-for-people-all-over-the-globe, (accessed 24 May 2021).

2. Forbes, S., Defense Advanced Research Projects Agency DARPA, Blackjack. https://www.darpa.mil/program/blackjack, (accessed 13 May 2021).

3. Space Development Agency SDA, https://www.sda.mil, (accessed 24 May 2021).

4. Harris, M., IEEE Spectrum, SpaceX Preps Self-Driving Starlink Satellites for Launch. https://spectrum.ieee.org/tech-talk/aerospace/satellites/spacex-preps-selfdriving-satellites-for-launch, 22 May 2019 (accessed 24 May 2021).

5. Smith, H., NASA, Distributed Spacecraft Autonomy (DSA). https://www.nasa.gov/directorates/spacetech/game_changing_development/projects/dsa, 23 October 2020 (accessed 13 May 2021).

6. Wertz, J.R. and W.J. Larson, Space Mission Analysis and Design, third ed., Microcosm Press, California and Springer, New York, 1999.

7. Nag, S., LeMoigne, J., and de Weck, O.L., Cost and Risk Analysis of Small Satellite Constellations for Earth Observation, IEEE Aerospace Conference, Big Sky, MT, USA, 2014.

8. Foreman, V.L., Le Moigne, J., and de Weck, O.L., A Survey of Cost Estimating Methodologies for Distributed Spacecraft Missions, AIAA SPACE, Long Beach, CA, USA, 2016.

9. Fong, T., Autonomous Systems NASA Capability Overview. https://www.nasa.gov/sites/default/files/atoms/files/nac_tie_aug2018_tfong_tagged.pdf, 24 August 2018 (accessed 13 May 2021).

10. NASA Office of the Chief Technologist, 2015 NASA Technology Roadmaps. https://www.nasa.gov/offices/oct/home/roadmaps/index.html, 2015 (accessed 13 May 2021).

11. O'Donnell Jr., J.R., Concha, M.A., Morrissey, J.R., Placanica, S.J., Russo, A.M., and Tsai, D.C., Space Technology 5 Launch and Operations, AAS Guidance and Control Conference, Breckenridge, CO, USA, 2007.

12. NASA, ST5 Spacecraft & Instruments. https://www.nasa.gov/mission_pages/st-5/spacecraft/index.html, (accessed 13 May 2021).

13. Frazzoli, E., Dahleh, A., Feron, E., and Kornfeld, R.P., A Randomized Attitude Slew Planning Algorithm for Autonomous Spacecraft, AIAA GNC Conference, Montreal, Quebec, CA, 2001.

14. Francis, G.D., Collins Jr., E.G., Chuy, O., and Sharma, A., Sampling-Based Trajectory Generation for Autonomous Spacecraft Rendezvous and Docking, AIAA GNC Conference, Boston, MA, USA, 2013.

15. NASA, State of the Art of Small Spacecraft Technology: 8.0 Command and Data Handling. https://www.nasa.gov/smallsat-institute/sst-soa-2020/command-and-data-handling, (accessed 13 May 2021).

16. Atkinson, D.J., Emerging Cyber-Security Issues of Autonomy and the Psychopathology of Intelligent Machines, AAAI Spring Symposium, Palo Alto, CA, USA, 2015.

17. LaValle, S.M., Sampling-Based Planning Under Differential Constraints in Planning Algorithms, Cambridge University Press, Cambridge, UK, 2006.