[SSC21-NST-04]

Design and Testing of Autonomous Distributed Space Systems

Nicholas Cramer, Daniel Cellucci, Caleb Adams, Adam Sweet, Mohammad Hejase, Jeremy Frank
NASA Ames Research Center
Moffett Field, CA
nicholas.b.cramer@nasa.gov

Richard Levinson, Sergei Gridnev, Lara Brown
KBR Wyle Services, LLC
Moffett Field, CA

ABSTRACT

Distributed Space Systems (DSS) are an emerging class of mission designs that enable new scientific and commercial opportunities. In order to enable those new opportunities, these systems will need to have significantly expanded autonomous capabilities compared to their single-spacecraft predecessors. In this paper, we present Distributed Spacecraft Autonomy (DSA) project, a payload on NASA's Starling spacecraft experiment. We first describe a step-by-step process for characterizing what features are needed in an autonomous DSS, and show how this process applied to DSA. We then describe the Starling mission, a four-spacecraft swarm hosting multiple DSS payloads. We then describe DSA, which will mature in-space networking and autonomous planning technologies to measure topside ionosophere features using data from the Starling spacecraft's GPS receivers. We describe how DSA will coordinate observations of GPS satellites using Starling's underlying communications infrastructure combined with novel DSS technology. The flight validation of DSS technology will provide mature technology to enable future DSS missions.

Introduction

Future visions of space missions are full of low-cost small satellite swarms operating in concert to achieve complex mission objectives. Real-time, on-board multi-spacecraft coordination, data analysis and prioritization will not only optimize science return from a mission by establishing observational parameters of interest or success, but will also enable outer solar system missions and missions in extreme environments (e.g., Io, Venus, sub-surface Europa) in which communication with ground operations and ground-based analysis times are limited. This capability will enable previously impossible classes of missions through unprecedented levels of autonomy.[1,2] Distributed Spacecraft Missions (DSMs) have been described as priorities in the Heliophysics Decadal Survey,[3] as well as several white papers submitted for the Astrobiology and Planetary Science Decadal Survey (APSDS).[4] Several white papers for the APSDS also demonstrate a strong need for real-time, on-board autonomous analyses. Realizing these envisioned space missions will require significant advancement of the capabilities of the Distributed Space Systems (DSS) architectures that implement them. DSS are composed of multiple spacecraft that function together to achieve a single mission objective.[5,6] In some cases, the DSS come together to form a sensory system that could not exist in a monolithic platform.[7,8] In other configurations, they take advantage of distributed measurement to extract information about the spatial and temporal effects of phenomena much large than what one spacecraft could image alone.[9,10]

Figure 1: A distributed space system of cubesats in orbit.

With missions requiring DSS becoming critical for many emerging applications, there has been par-

ticular interest in addressing the technologies needed to enable them. In this paper we focus on on-board processing, inter-satellite networks (also known as inter-satellite links or ISLs), and autonomous decision making. These three technologies are interdependent. For instance, in order to have effective autonomous decision making, the processors on board need to have the necessary compute power to both process the data needed to make decisions on and perform what could be computationally intensive or time-sensitive decision-making calculations. Additionally, the inter-satellite networks allow the spacecraft composing the DSS to coordinate amongst themselves without direct ground control. This capability is critical because the ground connection, especially in deep space applications, is constrained in bandwidth and latency. Further discussion of Low Earth Observing-based DSS technology maturation is described in.[11]

There are four key advantages that a DSS including these enabling technologies can provide:

- Distributed Coordination: can share data and change what they prioritize
- Autonomous Retasking: can respond to environmental stimuli autonomously, without requiring intervention from a ground operator
- Increased Availability: when only a single spacecraft can be reached, it can relay commands to the others
- Workload Balancing: can retask satellites based on available computation, power, and communications resources

In order to develop and demonstrate inter-satellite communication network and the autonomy technologies, NASA's Space Technology Mission Directorate has funded two projects. The first project is the Starling mission,[12] a four-satellite swarm launched into Low Earth Orbit that uses crosslink radios to support and demonstrate many DSS experiments.

The second is Distributed Spacecraft Autonomy (DSA),[13] a software payload on each of the spacecraft in the Starling mission that uses the crosslink radios as the backbone to build an autonomous DSS that responds to changing conditions in the topside ionosphere.

Each satellite senses its environment using the signals received from several GPS satellites using the on-board dual-band GPS radio. The signals from each GPS satellite are each assigned a channel. The DSA software payload assigns two reward values, an exploit and explore reward, to each channel. The explore reward encourages maximizing the number of GPS signals observed across the whole system, and the exploit rewards encourage maximizing the total TEC observed.

The system maintains the consistency of these rewards across all of the satellites using the crosslink network. Doing so sets up an opportunity for the DSS to respond to the environment by collectively determining what signals each spacecraft observes, based on the current phenomena being observed. The use of GPS signals generates a truth set to compare our system's efficacy.

While we use TEC as a reward generator, the autonomy algorithm can use any arbitrary value. The validation of the autonomous DSS performed by the DSA-Starling demonstration lays the groundwork for future self-organizing autonomous DSS.

Autonomous DSS Concept

While autonomous DSS have many great advantages, one of the first steps to formulating an autonomous DSS is to establish why and how autonomy is necessary and the critical aspects to the autonomous DSS design. The primary applications for DSS are spacecraft communication, precision navigation, and timing (PNT), remote sensing, meteorology, and on-orbit servicing.[14] These applications can then be grouped into two domains: service providing (communication, PNT, and on-orbit servicing) and data collection (remote sensing and meteorology). For this paper, we will focus on the data collection domain.

Starting from the premise that we are addressing the creation of an autonomous DSS with the focus on data collection, we have created an illustrative decision tree in Figure 2. This decision tree is not meant to be exhaustive, but we formulated each question to assess the need of the DSS and the type of autonomy that might be necessary. The DSA decision branches are highlighted in the figure.

For instance, the first question that needs to be asked is if the system being observed is dynamic in space and time. If the particular feature of interest is not dynamic in both space and time, then an autonomous DSS is probably unnecessary, barring a need for additional redundancy. If the feature being observed does not change with time, then a single autonomous spacecraft at many different locations could be used over time to collect the needed data. If the feature changes with time but not space, then a single stationary data collection device would be sufficient.

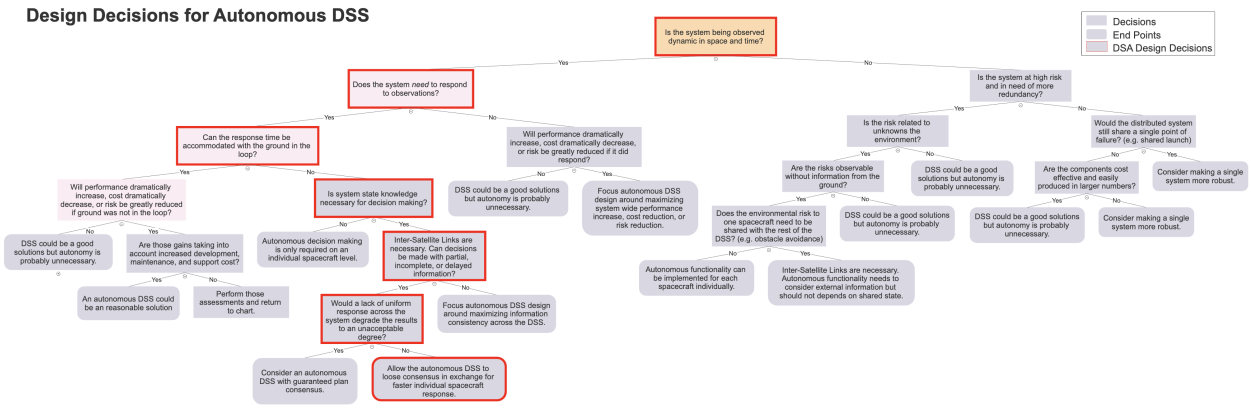


Figure 2: An illustrative decision tree to determine the need for an autonomous DDS and what characteristics are necessary.

The details of the DSA technology demonstration will be covered later in the paper.

The DSA decision tree from Figure 2 resulted in the features shown in Table 1. As stated in Figure 2, the need for system state knowledge in combination with operations autonomous of the ground creates a need for inter-satellite links so that the satellites can share their state.

Table 1: DSA Autonomy Characteristics

Characteristic	Needed
Observing temporal and spatial features	Yes
Ability to respond to observations	Yes
Operate autonomous of ground	Yes
System state knowledge needed	Yes
Decisions can be made on partial information	Yes
Consensus required	No

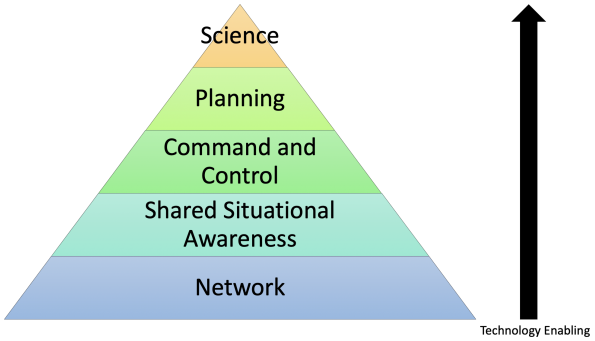


Figure 3: An autonomous DSS needs to have successive technologies enabling each other. The basis of the distributed autonomy is the ability to communicate through a network that enables a distributed state to be formed. That distributed state can be used to provide commands and control to the DSS, which are the mechanism for plans to be enacted. All of which enable the science to be performed.

Starting from the characteristics established for an autonomous DSS, there is a hierarchy of technologies that enable each other to form the foundation of the DSA autonomous DSS. These enabling technologies are displayed in a pyramid format in Figure 3. Each of the lower levels of the pyramid enable the levels above them. For example, the network enables the distributed state, and the distributed state enables command and control. These enabling technologies are as follows:

Definitions of Enabling Technologies

- **Network:** the ability of data to go from the application layer on an individual spacecraft to the application layer of all other or a subset of other spacecraft in the DSS.
- **Distribute State:** Cohesive state knowledge of the full DSS. This includes states that can only be estimated through the collection of individual spacecraft information.
- **Command and Control:** The ability to command and control the DSS as a single system with a shared objective.
- **Planning:** Utilizing system-level information to create a plan that provides DSS-level commands that can be executed to achieve the science objective.
- **Science:** A distributed spacecraft system with autonomous capabilities can rebalance task distribution without intervention from the ground.

Figure 4 shows the high-level architecture for DSA's autonomous DSS. There are two different architectures presented in Figure 4. The top architecture can control the spacecraft's trajectory and attitude. The lower one assumes these control vectors are unavailable. The lower configuration is a likely scenario for secondary payloads that will be operating alongside a primary payload. The architecture contains four primary components:

- **Intelligent sensor:** sensors system (hardware, firmware, and software) that can be commanded are becoming increasingly common in space applications. Sensors utilizing deformable mirrors (DMs) for high contrast imagery,[15,16] synthetic aperture radars,[17] adaptive or steerable LiDAR,[18,19] and on-board image analysis[20] are all examples of intelligent sensing systems that either currently use or could be, autonomously directed to maximize data collection about features of interest.

- **Autonomous Planner:** software that takes in the local and system-wide state information and then formulates a plan to achieve the science or data collection objectives.

- **Guidance, Navigation, and Control:** state estimation, trajectory management, and attitude control module. In some cases, such as formation flight, system-wide states can be required.

- **Communication Manager:** manages the inter-spacecraft and ground-based communications. This application is responsible for managing information consistency across the DSS.

Each of the identified components communicates in particular ways. The dashed squares in Figure 4 represent an individual spacecraft, and the red lines between them are the crosslinks. The only interaction that the spacecraft have with each other is via the communication manager and over the crosslinks. The communication manager will send the state of the objective and any plan information from one satellite to others. In the case where the intelligent sensor structure needs information from the other spacecraft (e.g. in the case of distributed radar sensors[21]), that information can also be shared over the network of crosslinks. The autonomous planner component then provides commands/feedback to the intelligent sensor, the GNC component, and the other versions of itself over the crosslinks.

The GNC component in this architecture will receive the plan from the planner and work with other instances of itself as necessary to execute the provided plan. It is worth noting that this navigation step could be performed separately and be included in the lower architecture. In that case, it is assumed that the navigation module has become part of the planner component as part of the overall state estimation.

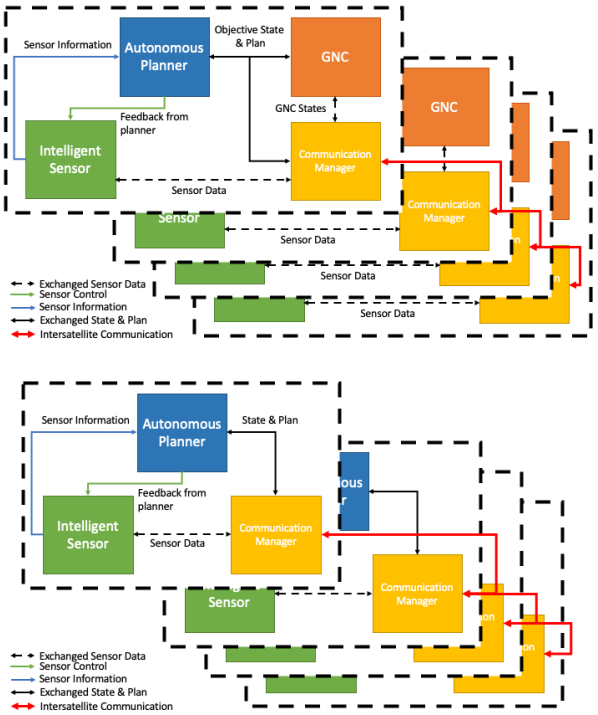


Figure 4: High-level flow chart of two different autonomous DSS architectures. The primary difference is that the top architecture can control the spacecraft position while the lower structure does not.

The DSA project includes an on-orbit technology demonstration operating alongside the main Starling payload. As a result, this project will use the architecture presented in the lower diagram of Figure 4.

Technology Demonstration Description

The DSA flight mission will demonstrate distributed coordination of observations of the topside ionosphere. The topside ionosphere is a transitional region between the ionosphere and the inner magnetosphere that displays many dynamic features like Equatorial Plasma Bubbles[22] and Polar Patches.[23] The satellites perform these observations using the onboard dual-band Novatel OEM719 GPS receiver

already integrated into the Starling payload that provides precise orbit determination to the satellites in the swarm.[24]

Previous missions like the European Space Agency's CHAMP[25] and Swarm[26] have used GPS receivers with similar capabilities to perform similar measurements. While the Swarm mission consists of multiple satellites, these satellites do not communicate with one another. However, the archived data from this mission forms a vital component of validating the DSA mission's expected behavior by providing high-quality measurements of the target environment from multiple vantage points. The critical validation data from the Swarm mission is the Total Electron Content (TEC) measured from the accumulated phase delay of the dual-band GPS signal as it passes through the plasma in the space between the transmitting GPS satellite and the receiver.

There are several forms of bias that can complicate the process of comparing TEC signals received both from different GPS satellites and by receivers on different satellites in the swarm.[27] The largest of which are the Differential Biases that are imparted during the processing of the two signals in the hardware of the receiver. Typically, estimates of these biases are made on the ground, where a large range of data are available. Creating rough estimates of the inter-spacecraft biases that allows comparison of signals between different receivers.

Figure 5 shows the same timeseries of received GPS data with three different bias estimates. Relative TEC is the slant TEC as measured, with no receiver bias correction. Absolute TEC is the Relative TEC with the estimated bias correction, a constant value calculated over one full GPS track. Initial Value Subtraction is the Relative TEC with the first value in the timeseries used as the estimated bias correction. While Initial Value Subtraction does not accurately capture the absolute value of the signal, its relative simplicity, the preservation of the relative magnitude of the signal, and the fact that it can be applied to data as it is received makes it an attractive replacement for the absolute bias correction. Furthermore, for the purposes of the demonstration where feature recognition is more important than scientific accuracy, these sorts of datastream-friendly corrections are sufficient.

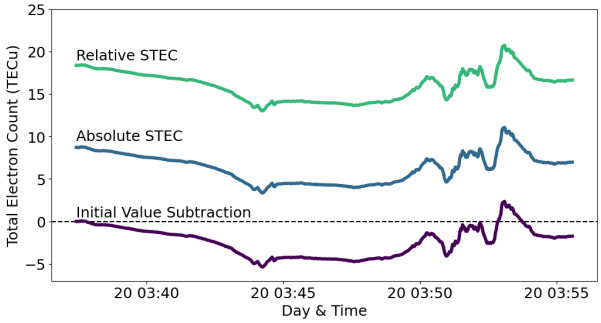


Figure 5: Comparison of absolute STEC, relative STEC, and a simplified offset, demonstrating that the relative magnitudes the TEC signal features of the simplified solutions match those of the corrected values making them reasonable to make decisions on.

The GPS receiver can receive signals from multiple GPS satellites simultaneously. The DSA mission flight demonstration involves each satellite in the swarm selecting a subset of these multiple GPS satellites in view to flag them for downlink, based on the competing objectives of coverage and science value. Since the full dataset for a day will be downloaded in addition to these flagged channels, the performance of these flagged measurements can be compared against optimal plans produced on the ground. This mission therefore provides a unique opportunity to validate mission performance of a distributed self-sampling swarm. The algorithms developed here can be extended to other, similar problems in distributed satellite operations, including coordinated pointing of cameras with a limited field of view for observation of a dynamic event,[28] communication with ground stations, and distributed radio tomography of the magnetosphere.[10]

Problem Formulation

With the technology demonstration formulated, the next task is to translate that demonstration into a problem formulation that can inform the design of an autonomous planner. To help with the explanation, we will first assume that the Starling spacecraft are in an in-train configuration. The problem formulation will be generalized, but this will help explaining assumptions.

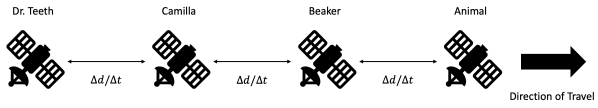


Figure 6: Visualization of the four spacecraft traveling in the same direction.

We will name the four satellites Animal, Beaker, Camilla, and Dr.Teeth, where they are alphabetically arranged, so Animal is the first, and Dr. Teeth is the last spacecraft to pass a specific point. We can furthermore assume the distances between each spacecraft are consistent and the same for all of the spacecraft. This means that Animal is separated from Beaker by Δd or the equivalent time displacement Δt. That same $\Delta d/\Delta t$ exists between Beaker and Camilla as well as Camilla and Dr. Teeth as shown in Figure 6.

The first assumption is that what the DSS senses changes over the sampling time of the DSS. This means that measurement Beaker collects when it reaches the location that Animal was at in Δt seconds will not be identical to those measured by Animal Δt seconds earlier. This is an instantiation the first decision in Figure 2.

We will also assume that there is a constraint on the number of GPS channels that can be processed on board, which we refer to as a capacity constraint. This constraint is similar to slewing or processing constraints that might exist for instruments with a limited field of view.

We can start the problem formulation by presenting the two relevant sets, the DSS set S, and the visible GPS set G.

$$\forall s_i \in S: \qquad 1 \leq i \leq N \tag{1}$$

where s_i is a spacecraft in the DSS and N are the number of spacecraft in the DSS.

$$\forall g_j \in G: \qquad 1 \leq j \leq |G| \tag{2}$$

and

$$G \cup s_i^v \tag{3}$$

where s_i^v is the visual set of the GPS that can be seen by spacecraft i. There are three types of rewards provided. For every visible GPS satellite j, there is an coverage reward r_c, and for each observed GPS channel there are explore $r_{i,j}^{explore}$ and exploit $r_{i,j}^{exploit}$ rewards, for a spacecraft i and GPS satellite j. The decision variables are if the GPS satellite is covered c_j and if the channel is observed $o_{i,j}$. A satellite is considered covered if there is a at least one channel from that satellite being observed. Coverage is included as a parameter to ensure that information from a particular GPS satellite is not being missed.

This concept is shown in Figure 7 where each channel that is observed receives two rewards, and all of the GPS satellites are covered. The blue bars in Figure 7 represent the exploit rewards which are intended to maximize the TEC features captured, while the explore reward is meant to force the solution to look at other channels. Not shown on the figure is the coverage reward, which encourages the observation of all channels for each of the GPS satellites.

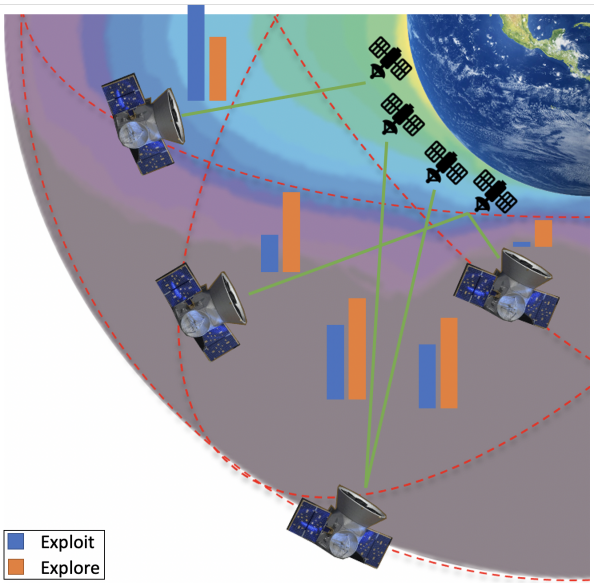


Figure 7: Notional example of the technology demonstration where the green lines are the selected GPS channels and the bar charts are the resulting exploit and explore rewards that get fed into the planner. Background image based on results presented[25]

Equation 4 describes the objective function that the channel selection algorithm maximizes, and is subject to the capacity constraint

$$\forall s_i \in S: \qquad \Sigma_j o_{i,j} \leq s_i^c \tag{5}$$

where s_i^c is the capacity constraint on the ith spacecraft. It is also subject to the coverage constraint

$$c_j = 1 \leftrightarrow \text{ GPS } g_j \text{ is covered by at least } b \text{ observers} \tag{6}$$

$$maximize \left(\overbrace{\alpha \Sigma_{i,j} \left(\beta r_{i,j}^{explore} o_{i,j} + (1-\beta) r_{i,j}^{exploit} o_{i,j} \right)}^{Scheduling} + \overbrace{(1-\alpha) \Sigma_j r_c c_j}^{Coverage} \right) \tag{4}$$

which means

$$\Sigma_i o_{i,j} \geq b_j : \\ 1 \leq b_j \leq \text{number of spacecraft that can see } g_j \tag{7}$$

where r_c is a constant to put the coverage on the same order as the normalized rewards. The parameter α and β are tuning parameters. α varies the weighting between the the of coverage and the scheduling. This trade determines the variation between breadth of observations (coverage) and prioritization of environmental characteristics (rewards). β varies the weighting between the exploit and explore rewards.

Software Solution

With the problem formulated and an objective defined, the approach needs to be translated into software. The software is developed in C++, as a user application in NASA's core flight software (cFS)[29] environment. Figure 8 shows a more detailed breakdown of the software components that implement the functionalities presented in Figure 4. It consists of three major modules: the communications manager, the intelligent sensor, and the autonomous planner.

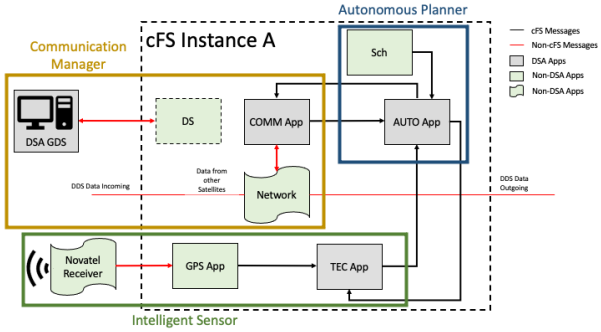


Figure 8: The DSA implementation of the software components identified in 4 where the primary apps are the TEC, AUTO, and COMM app. They are supported by the hardware, firmware, and existing core cFS applications

The communication manager includes the cFS Data Store (DS) application, which stores the telemetry data sent over the software bus for eventual download to the ground, and the communication application (COMM App), which manages the inter-spacecraft crosslinks messaging and implements the Distributed Data Service (DDS) that allows satellites to communicate using a publish-subscribe framework. The COMM App receives messages from other cFS apps, translates them from cFS bus messages to DDS messages, sends them over the crosslink network, and translates received DDS messages back to cFS messages. The crosslink network is represented by the Network block in Figure 8.

The network implementation and the DDS layer on top is a joint effort between the Starling and DSA missions. Figure 9 shows the components that go into the COMM App and Network. The sections labeled DSA are the application-level topics created by the COMM Apps on all of the spacecraft to which these spacecraft subscribe and over which these spacecraft publish. DSA uses RTI's Micro DDS,[30] which allows the management of certain quality of service (QoS) parameters. The Starling mission manages the transport protocol and communication network, which implements the Better Approach To Mobile Ad-hoc Networking (BATMAN)[31] protocol on top of the crosslink radio network.

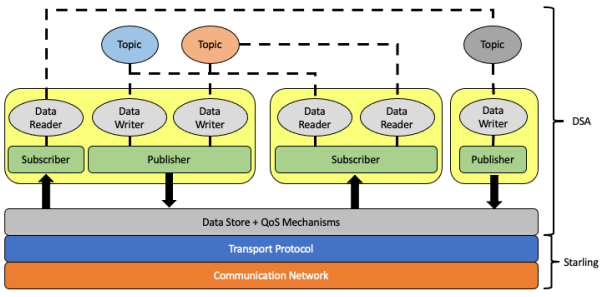


Figure 9: An notional example of the DDS implementation with the work split between the two projects. Figure based off the diagram from Wang et al.[32]

The Intelligent Sensor component consists of the

Novotel GPS receiver and its associated firmware, and GPS App that parses the messages, and the TEC app that turns those messages into rewards. The TEC app first performs a rough estimation of the TEC, then transforms the TEC and position data to generate the rewards. The current implementation of the reward generator, the "naive implementation", uses a simple estimation of the current TEC for the exploit rewards, with a compensation factor to ensure that the values are always positive, and the distance between the spacecraft and the GPS satellite for the explore rewards. The TEC app gets a list of satellites to observe from the AUTO app that it then uses to filter which channels for which it calculates rewards.

The AUTO app is the implementation of the optimization of the cost function from Equation 4. In combination with the cFS Scheduler application (Sch), which provides a consistent wake-up message, the AUTO app performs a calculation of the optimal channel allocation using current data once per tick. The AUTO App takes in the rewards from the TEC app, normalizes them, gathers the reward states from the other instances of the autonomy app running on the other satellites and communicated over the COMM app. Then each instance of the AUTO app uses the rewards it received through the COMM apto generate the observation plan for the next time tick, as determined by the scheduler application.

Testing Setup

In order to verify and validate the full multi-spacecraft mission scenario, the testing setup needed to implement a system that could easily instantiate a simulated DSS with a simulated network that also had minimal overhead. This low-profile testing framework allows multiple developers to work simultaneously. Figure 10 shows the basic software testing framework implemented. Each spacecraft runs in a separate container, with an instance of cFS loaded with the DSA software on it. The COMM app from Figure 8 then uses the network sim test module to communicate to the other instances on the COMM app in other containers. The host machine generates the data for the TEC app for each spacecraft as if there were separate unique GPS receivers in the container, and then feeds these data over the host machine connection. This host machine generation effectively replaces the GPS app from Figure 8.

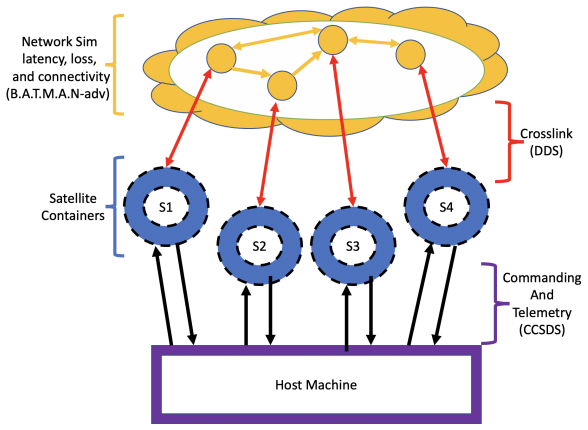


Figure 10: Software testing framework with the controllable network in yellow, containerized instances of cFS and the DSA software in blue, and the host machine acting as a ground station in purple.

The publicly-available TEC timeseries data from the ESA Swarm mission provides the source data for DSA scenario testing.[26] The Swarm mission provides the L1 data values, and the host machine then back-calculates what the expected input would be from these data. The ESA Swarm mission consists of three satellites, with only two of them, sats A and C, in a formation similar to the expected DSA mission. As a result, the host machine creates "virtual" satellites as part of its data generation process in order to generate estimates of what the data would look like for the four Starling spacecraft. These virtual satellites are the data from the ESA Swarm Sats A and C shifted Δt seconds forward and backward respectively in order to create four distinct timeseries for each of the DSA satellites. For the simulations in the Result section the capacity constraint, which is the number of channels that can be observed is set to three and does not change.

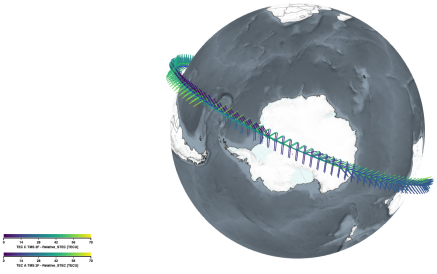
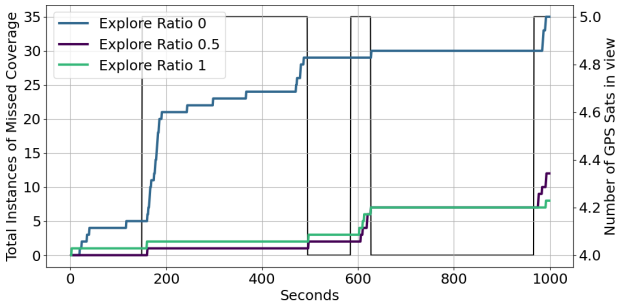
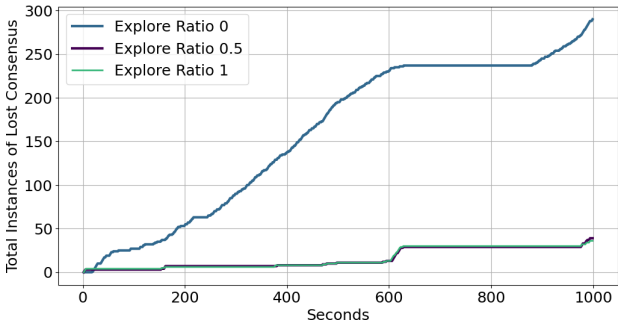


Figure 11: An graphical representation of the polar pass used to generate the simulation data for the results. Displayed using the ESA VIReS project[33]



(a) Cumulative missed coverage for each of the simulation runs, with the vertical black lines indicating the step changes in the total number of GPS satellites in view by the DSS during the scenario.



(b) Cumulative time spent with lost consensus for the scenario.

Figure 12: Cumulative behavior of the swarm coverage and consensus for different ratios of explore to exploit values.

The data used to generate the results for this paper are based on the ESA Swarm satellites A and C on the date 2016-01-01 from 08:00:08 to 08:16:46. Figure 11 shows a graphical representation of the relative slant TEC sensed by the satellites during the times from 08:00:08 to 08:47:51 on 2016-01-01.

Preliminary Analysis of DSA Performance

We have performed a preliminary analysis of DSA performance, as described further below. The results of this analysis consist of the simulation outputs from running the DSA software as described in Section with various reward ratios. The capacity constraint was kept constant at 3 channels for all of the simulations. Table 2 provides a summary of the simulation results.

We measured algorithm performance using two metrics: consensus and coverage. Consensus is the percentage of ticks where all of the spacecraft had the same plan. Coverage is the percentage of ticks where all of the GPS satellites in view were selected by at least one of the satellites in the DSS. Consensus is desirable because, all else equal, more simultaneous observations of interesting TEC data are of scientific value. Coverage is desirable to ensure no interesting TEC signals were missed.

Table 2: Simulation Summary

Explore Ratio	Consensus %	Coverage %
1.0	96.4	99.2
0.5	96.1	0.988
0.0	71	96.5

Figures 12b and 12a show the cumulative number of ticks where there was no consensus or where the DSS was unable to achieve full coverage. Figure 12a shows the number of GPS satellites in view. From these figures, it is apparent that, for explore ratios 1 and 0.5, most instances of loss of consensus occur when the number of satellites in view changes, most commonly when a GPS satellite enters or exits the field of view of the DSS. This is likely the result of minor mismatches in the shared state across the swarm. Typically, inconsistencies between communicated and actual satellite states over a single tick will not change the resulting plan for the swarm. However, the step change associated with the number of GPS satellites in view, coupled with the fact that this change in GPS satellites in view will not be observed by all satellites simultaneously, means that small discontinuities can result in significant changes to the plan.

Figure 12b also shows that the explore ratio one situation behaves uniquely to the other two explore ratios. Instead of discrete jumps at transitions, it is missing consensus at a consistent and nearly linear rate. We can see that it does stabilize when the number of satellites in view reduces to four between 600 and 800 ticks. This could be because GPS 26 was no longer in view, and looking at Figure 13 we can see that GPS 26 exploit reward is a particularly noisy signal. GPS 26 also only had spacecraft one monitoring it, so any disagreement due to the noise or delays left it uncovered, unlike GPS 15, which was noisy and covered by multiple spacecraft.

The coverage plot in Figure 12a shows that the explore ratio 0 run, which does not value the explore rewards at all, had a significantly higher amount of missed coverage instances than the other two. There are no indications in Figure 13 which shows the normalize exploit rewards for each spacecraft in the DSS

why this would be. The largest jump occurs just before 200 ticks, where a fifth GPS satellite came into view. When we look at Figure 13, we can see that there does look like a gap occurred during that transition where spacecraft three and four stopped observing, but spacecraft two had not started yet.

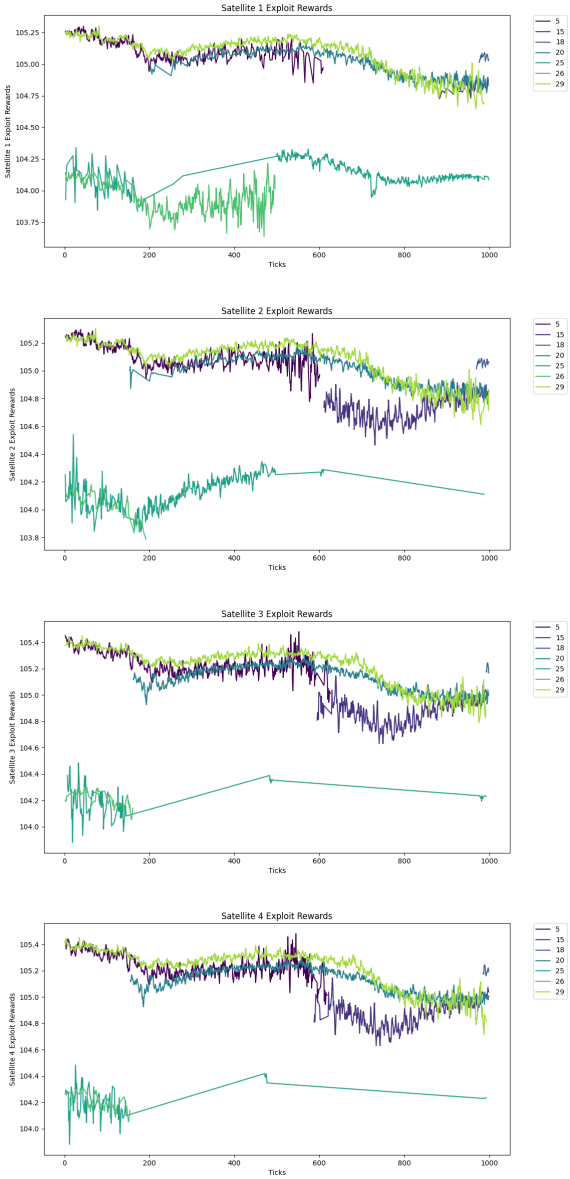


Figure 13: High level flow chart of two different autonomous DSS architectures. The primary difference is that the top architecture can control the spacecraft position while the lower structure does not.

Figure 14 shows the explore rewards for the explore ratio 0 run with shading on the background to show when one or more GPS satellites were in view but not being observed by spacecraft one. In this case, the white section round 600 ticks is the result of spacecraft one only having three GPS satellites in view, so it only observed those three. The darker sections indicate that multiple GPS satellites were not being observed.

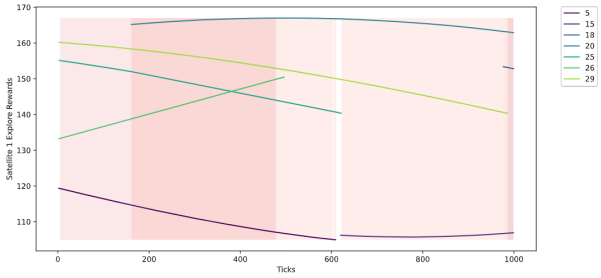


Figure 14: Explore rewards and the not watching segments highlighted in red. Darker red values indicate multiple GPS satellites not being observed, and the white section in the middle means that the number of visible satellites was less than or equal to the constraint.

Conclusions

This paper presents a formulation for determining if an autonomous DSS is the correct mission architecture and what characteristics that architecture might take. The DSA project was used as an example test case with the autonomous DSS enabling technologies being identified and a general software architecture proposed. The planned DSA tech demonstration was presented and translated in the planner formulation. The software representation of that formalization was discussed, and results from preliminary simulation were demonstrated.

We expect to continue to develop and mature the planner algorithm in the future and will demonstrate it on orbit as part of the Starling mission.

Acknowledgments

We thank the Game Changing Development Program in NASA's Space Technology Mission Directorate for funding this work.

References

[1] F. Tan and M. Seablom. 2018 Workshop on Autonomy for Future Nasa Science Missions.

In *https://science.nasa.gov/technology/2018-autonomy-workshop/output-results*.

[2] J. Lemoigne, M. M Little, M. Cole, and J. Ellis. New Observing Strategies (NOS) for future NASA Earth Science Missions. *Proceedings of the American Geophysical Union Fall Meeting*, pages 401–416, 2019.

[3] Heliophysics Decadal Survey Whitepaper Concepts. In *https://www.lpi.usra.edu/decadal_whitepaper_proposals/heliophysics/*.

[4] Planetary / Astrobiology Decadal Survey Whitepapers. In *https://baas.aas.org/vol-53-issue-4*.

[5] Space Studies Board, Engineering National Academies of Sciences, Medicine, et al. *Achieving Science with CubeSats: Thinking Inside the Box*. National Academies Press, 2016.

[6] Benjamin Andrew Corbin. *The Value Proposition of Distributed Satellite Systems for Space Science Missions*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and ..., 2015.

[7] M Grasso, A Renga, G Fasano, MD Graziano, M Grassi, and A Moccia. Design of an end-to-end demonstration mission of a formation-flying synthetic aperture radar (ff-sar) based on microsatellites. *Advances in Space Research*, 67(11):3909–3923, 2021.

[8] Eric Joffre, Dave Wealthy, Ignacio Fernandez, Christian Trenkel, Philipp Voigt, Tobias Ziegler, and Waldemar Martens. Lisa: Heliocentric formation design for the laser interferometer space antenna mission. *Advances in Space Research*, 67(11):3868–3879, 2021.

[9] Laura Plice, Andres Dono Perez, and Stephen West. Helioswarm: Swarm mission design in high altitude orbit for heliophysics. In *AAS/AIAA Astrodynamics Specialist Conference, Portland, ME, USA*, 2019.

[10] RE Ergun, DE Larson, T Phan, JP McFadden, CW Carlson, I Roth, GT Delory, S Bale, V Angelopoulos, RJ Strangeway, et al. Magnetospheric constellation and tomography mission concept. *Simulation*, 4:6, 1998.

[11] Daniel Selva, Alessandro Golkar, Olga Korobova, Ignasi Lluch i Cruz, Paul Collopy, and Olivier L de Weck. Distributed earth satellite systems: What is needed to move forward? *Journal of Aerospace Information Systems*, 14(8):412–438, 2017.

[12] Hugo Sanchez, Dawn McIntosh, Howard Cannon, Craig Pires, Josh Sullivan, Simone D'Amico, and Brendan O'Connor. Starling1: Swarm technology demonstration. 2018.

[13] Daniel Cellucci, Nick B Cramer, and Jeremy D Frank. Distributed spacecraft autonomy. In *AS-CEND 2020*, page 4232. 2020.

[14] Joseph N Pelton, Scott Madry, and Sergio Camacho-Lara. *Handbook of satellite applications*. Springer, 2017.

[15] Eduardo A Bendek, Garreth J Ruane, Camilo Mejia Prada, Christopher B Mendillo, AJ Eldorado Riggs, and Eugene Serabyn. Microelectromechanical deformable mirror development for high-contrast imaging, part 1: miniaturized, flight-capable control electronics. *Journal of Astronomical Telescopes, Instruments, and Systems*, 6(4):045001, 2020.

[16] Ewan S Douglas, Gregory Allan, Derek Barnes, Joseph S Figura, Christian A Haughwout, Jennifer N Gubner, Alex A Knoedler, Sarah LeClair, Thomas J Murphy, Nikolaos Skouloudis, et al. Design of the deformable mirror demonstration cubesat (demi). In *Techniques and Instrumentation for Detection of Exoplanets VIII*, volume 10400, page 1040013. International Society for Optics and Photonics, 2017.

[17] Alberto Moreira, Pau Prats-Iraola, Marwan Younis, Gerhard Krieger, Irena Hajnsek, and Konstantinos P Papathanassiou. A tutorial on synthetic aperture radar. *IEEE Geoscience and remote sensing magazine*, 1(1):6–43, 2013.

[18] Carl Weimer, Mike Lieber, Reuben Rohrschneider, and Lyle Ruppert. A spaceborne adaptive lidar for earth imaging. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4224–4227. IEEE, 2017.

[19] Hieu V Duong, Michael A Lefsky, Tanya Ramond, and Carl Weimer. The electronically steerable flash lidar: A full waveform scanning system for topographic and ecosystem structure applications. *IEEE transactions on geoscience and remote sensing*, 50(11):4809–4820, 2012.

[20] Steve Chien, Joshua Doubleday, David R Thompson, Kiri L Wagstaff, John Bellardo, Craig Francis, Eric Baumgarten, Austin Williams, Edmund Yee, Eric Stanton, et al. Onboard autonomy on the intelligent payload experiment cubesat mission. *Journal of Aerospace Information Systems*, 14(6):307–315, 2017.

[21] Marco D&'Errico. *Distributed space missions for earth system monitoring*, volume 31. Springer Science & Business Media, 2012.

[22] Nanan Balan, LiBo Liu, and HuiJun Le. A brief review of equatorial ionization anomaly and ionospheric irregularities. *Earth and Planetary Physics*, 2(4):257–275, 2018.

[23] Alex T Chartier, Cathryn N Mitchell, and Ethan S Miller. Annual occurrence rates of ionospheric polar cap patches observed using swarm. *Journal of Geophysical Research: Space Physics*, 123(3):2327–2335, 2018.

[24] NovAtel, Inc. *OEM719*, 11 2019.

[25] Stefan Heise, N Jakowski, A Wehrenpfennig, Ch Reigber, and Hermann Lühr. Sounding of the topside ionosphere/plasmasphere based on gps measurements from champ: Initial results. *Geophysical Research Letters*, 29(14):44–1, 2002.

[26] Eigil Friis-Christensen, Hermann Lühr, and Gauthier Hulot. Swarm: A constellation to study the earth's magnetic field. *Earth, planets and space*, 58(4):351–358, 2006.

[27] M Noja, Claudia Stolle, Jaeheung Park, and Hermann Lühr. Long-term analysis of ionospheric polar patches based on champ tec data. *Radio Science*, 48(3):289–301, 2013.

[28] Sreeja Nag, A Aguilar, R Akbar, A Azemati, Jeremy Frank, Rich Levinson, A Li, Mahda Moghaddam, V Ravindra, and Daniel Selva. D-shield: Distributed spacecraft with heuristic intelligence to enable logistical decisions. In *IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020.

[29] David McComas. Nasa/gsfc's flight software core flight system. In *Flight Software Workshop*, volume 11, 2012.

[30] Real-Time Innovations, Inc. *RTI Connext DDS Micro 3.0.0 documentation*, 5 2021.

[31] Daniel Seither, André König, and Matthias Hollick. Routing performance of wireless mesh networks: A practical evaluation of batman advanced. In *2011 IEEE 36th Conference on Local Computer Networks*, pages 897–904. IEEE, 2011.

[32] Nanbor Wang, Douglas C Schmidt, Hans van't Hag, and Angelo Corsaro. Toward an adaptive data distribution service for dynamic large-scale network-centric operation and warfare (ncow) systems. In *MILCOM 2008-2008 IEEE Military Communications Conference*, pages 1–7. IEEE, 2008.

[33] European Space Agency. Earth's magnetic field as observed by satellite. In *https://vires.services/*.