Advancing Autonomy in Distributed Space Systems: Results from the Distributed Spacecraft Autonomy Experiment on Starling 1.0

Caleb Adams, Michael Iatauro, Brian Kempa,
Richard Levinson, Jeremy Frank
Intelligent Systems Division
NASA Ames Research Center
Moffet Field, CA, USA
caleb.a.adams@nasa.gov michael.j.iatauro@nasa.gov

Sergei Gridnev, Caroline Lassiter
KBR Wyle Services, LLC,
NASA Ames Research Center
Moffett Field, CA, USA
sergei.gridnev@nasa.gov caroline.r.lassiter@nasa.gov

ABSTRACT

The Distributed Spacecraft Autonomy (DSA) team at NASA's Ames Research Center presents results of the DSA experiment onboard the Starling 1.0 mission. The DSA system showcases collaborative resource allocation for multi-point science data collection with four small spacecraft, highlighting autonomy in decision-making as a crucial factor for multi-spacecraft missions. Central to DSA's operations is an autonomous GPS channel selection process used to optimize channel selection across the spacecraft swarm; the goal is to demonstrate fast, autonomous reactions to relevant scientific data. This is showcased by demonstrating the ability to capture ionospheric phenomena such as the Equatorial Ionization Anomaly and Polar Patches by utilizing relative Total Electron Content (TEC) measurements as sensor inputs. The DSA flight software architecture consists of three apps written in NASA's Core Flight System framework: The Comm App, TEC App, and Autonomy (AUTO) App. The Comm App enables message routing over the Ad-Hoc Network of Starling 1.0. The TEC App processes GPS receiver data from the spacecraft bus and produces inputs to the Autonomy App based on the spacecraft geometry and ionospheric features extracted from the TEC signal. The Autonomy App fuses the rewards from the local TEC App with the rewards from the other spacecraft and uses a Mixed Integer Linear Program (MILP) solver to autonomously select the best set of observations to optimize performance while sending minimal information to other agents in the swarm. The DSA system's autonomous reconfiguration ability is demonstrated, emphasizing its adaptability to natural phenomena without significant or direct tasking. The experiment's success demonstrates the potential for autonomous systems in future space missions, enabling spacecraft to operate independently and efficiently.

INTRODUCTION

The technical developments of the Distributed Spacecraft Autonomy (DSA) group's experiment on the Starling 1.0 has been extensively described in prior research. The content of this publication is primarily focused on describing and quantifying initial results.[1–6] DSA recently published insights from on orbit testing, which is the predecessor to this content and contains the same conclusions from initial results.[7] In this work we expand upon those conclusions and offer a holistic view of the experiment's success. Overall, the results of DSA's experiment are relevant to five key technical areas: distributed resource and task management, reactive operations, system modeling and simulation, human-swarm interaction, and ad hoc network communications.

DSA Project Requirements

DSA is a collaboration between two programs under NASA's Space Technology Mission Directorate (STMD). DSA is primarily funded under the Game Changing Development (GCD) program and is operated on the Starling 1.0 4 spacecraft swarm, funded via the Small Satellite Technology Program (SSTP). As a result of this, DSA has expectations, managed through requirements, for both programs. There are 3 project level requirements that address these expectations:

1. Develop Distributed Autonomy Software: The DSA Project shall develop software to control a swarm of spacecraft.

2. Conduct a Technology Demonstration in Flight: The DSA Project shall perform a technology demonstration of swarm software in flight.

3. Conduct a Simulation Demonstrating the Swarm Software Scalability: The DSA Project shall demonstrate the scalability of the swarm software in a simulation environment.

The results discussed here address DSA's second top level requirement. A summary of results that correspond with this requirement are in the following tables 12.

SSTP Requirement			
Level 1 Req.	Minimum Success Criteria	Full Success Criteria	Status Summary
Starling 1.0 shall conduct autonomous swarm reconfiguration testing	3 spacecraft operate in reactive swarm	4 spacecraft operate in reactive swarm	minimum achieved with 3 spacecraft

Table 1: SSTP top level requirements and summary.

GCD Key Performance Parameters				
KPP	State-of-the-art	Threshold Value	Project Goal	In-Flight Result
data uplink reduction	50%	66%	75%	49.3 % - 61.3 %
Time to reconfiguration	NA	22 min	5 min	2.04 - 6 sec

Table 2: GCD Key Performance Parameters (KPPs) and summary.

DESIGN OVERVIEW

DSA is a software payload on the Starling 1.0 mission.[8] The DSA-Starling flight demonstration centers around a GPS Channel Selection Experiment, described in detail in prior papers.[6] This experiment utilizes a dual-band GPS receiver to measure the total electron content (TEC) of the plasma between the spacecraft and GPS satellites. By analyzing these measurements, the experiment aims to capture various phenomena in the ionosphere, such as the Equatorial Ionization Anomaly and the Polar Patches.[9,10] This experiment was selected as the primary demonstration due to its ability to showcase autonomous reconfiguration in response to natural phenomena without significant integration efforts or modifications to the spacecraft hardware. Additionally, distributed autonomy technologies such as these may inform future New Observation Strategies, as desired by NASA's Earth Science and Technology Office.[11,12]

GPS Channel Selection Experiment

The topside ionosphere is a transitional region between the ionosphere and the inner magnetosphere that displays many dynamic features. The GPS Channel Selection Experiment focuses on using a dual-band GPS receiver to estimate the plasma density in the ionosphere. By measuring the relative group delay between signals broadcast at different frequencies by GPS satellites, the receiver can capture a wide range of ionospheric phenomena. Two specific phenomena of interest, the Equatorial Plasma Bubbles[9] and Polar Patches,[13] exhibit distinct behavior in TEC, and thus act as the features to be observed during the experiment. The experiment employs explorative channel selections (observe as many channels as possible) when the phenomena being observed are large and homogeneous, and exploitative channel selections (focus observations on channels where the TEC count is highest) when the phenomena are spatially constrained and short-lived.

Figure 1 provides a simplified representation of a channel assignment scenario within the DSS, where multiple spacecraft receive signals from GPS satellites. The experiment constrains the number of channels each spacecraft can observe, requiring DSA to coordinate channel assignments across the DSS using shared sampling. In the case of spatially-constrained phenomena, simultaneous sampling allows multiple spacecraft to observe the features of interest from different vantage points. DSA performance will be evaluated based on their ability to match the optimal channel allocations and their responsiveness to changes in observed features and operating conditions.

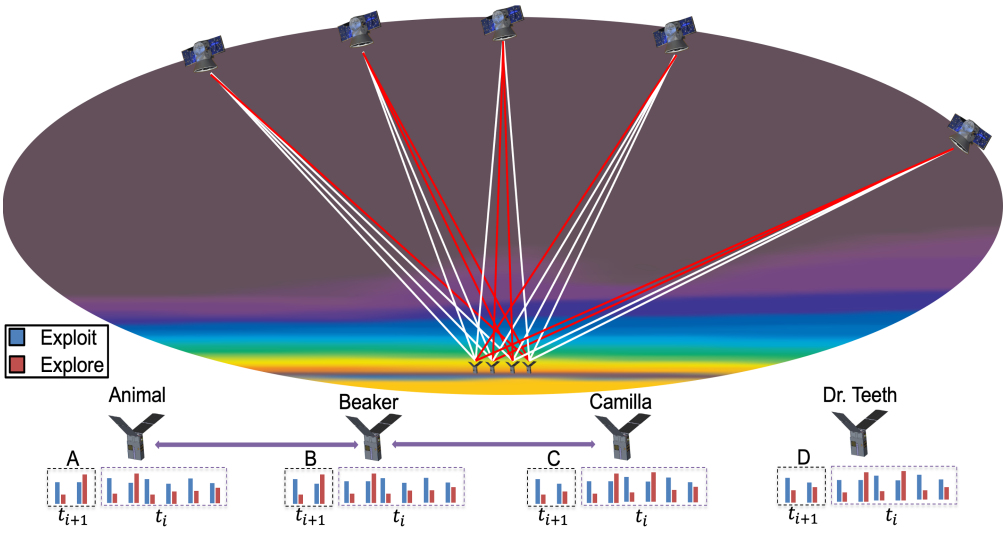

Figure 1: DSA Software performing autonomous GPS channel selection for TEC calculation. A time series below spacecraft A, B, C, and D represents the history of explore and exploit values used by the DSA software to autonomously select GPS channels; red and white lines of sight represented the variation in selected channels.

Flight Software

The DSA Flight Software utilizes the Core Flight System (cFS) as the framework for each satellite's flight software. This choice ensures compatibility with the Starling-1 flight mission software.[14] The DSA flight mission software consists of three apps within the (cFS) framework: the *COMM App, TEC App*, and *AUTO App*. The flight data-flow diagram (see figure 2) illustrates the flow of data from raw GPS instrument data to channel selections through the three cFS apps. The COMM App facilitates communication between the local autonomy software and other spacecraft, while the TEC App calculates relevant information from raw GPS range data.[15] The AUTO App utilizes an MILP solver to find optimal channel allocations by combining rewards from the TEC App and other spacecraft. Additional details regarding these application implementations can be found in prior papers[1,2,4,6,7] published by the DSA group.

Relative Total Electron Content Measurement

The TEC Application produces two reward values that it sends to the AUTO Application. Those values are known as Explore and Exploit values. This section captures the transformation from measured TEC to exploitative reward values, which are measurements of relative TEC.

The measured phaserange will be referred to as $\varphi 1_m(i,t)$ and $\varphi 2_m(i,t)$ for the L1 and L2 bands respectively, where i indicates the given signal and t is the time that the functional phaserange was taken in GPS reference time. The reference functional phaserange is defined as φ_R and exists to make TEC output positive. φ_R will be values uploaded via cFS tables for each test run (Experiment Period) of our system.

For short duration testing, it is appropriate to approximate φ_R as

$$\varphi_R \approx max(\Phi 2(\mathbb{S}, \{t \mid t_i < t < t_f\})) \\ -min(\Phi 1(\mathbb{S}, \{t \mid t_i < t < t_f\}))$$

where $\Phi 2$ is the test sets of $\varphi 2$ values for the set of all possible signals $\mathbb{S}$ over the sample time period starting at t_i and ending at t_f.

The reference phaserange is then used to calculate the relative phaserange, where and are the measured phaserange for signal at time t. The measured phaseranges and can be combined with the reference phase range to form the geometry free calculation that only includes the ionosphere and receiver dependents delays.

$$\phi(i,t) = \varphi 1_m(i,t) - \varphi 2_m(i,t) + \varphi_R$$

this is translated into estimated total electron count

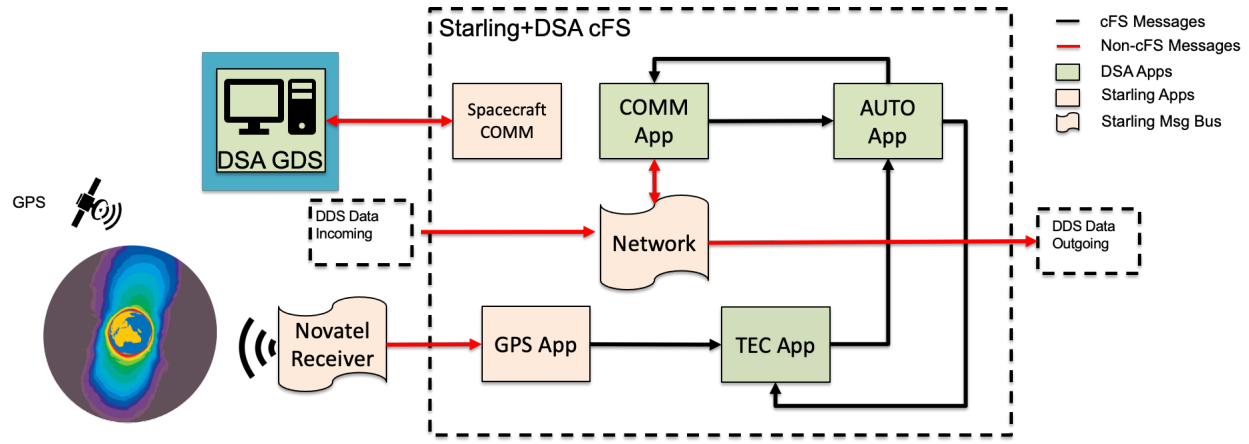

Figure 2: A simplified diagram of the 3 DSA applications operating within the Starling flight software environment, receiving GPS data, and communicating with the DSA Ground Data System (GDS), and utilizing the DDS network.

$$eSTEC(i,t) = \frac{f_1^2 f_2^2}{f_1^2 - f_2^2} \frac{\phi(i,t)}{\kappa}$$

which is converted into the common TECU

$$eSTECU = eSTEC * 10^{-16}$$

Simple Distance Exploration

The second reward value that the TEC Application produces is the exploration reward. This exploration reward is computed for each starling spacecraft and for each GPS spacecraft that an individual starling spacecraft can view. This is a simple euclidean distance where (p_x, p_y, p_z) represent the starling spacecraft location and (gp_x, gp_y, gp_z) represents the location of the GPS spacecraft; Thus, the exploration value for this Starling-GPS pair is simply:

$$explore = \sqrt{(gp_x - p_x)^2 + (gp_y - p_y)^2 + (gp_z - p_z)^2}$$

Autonomy

The purpose of the AUTO app is to decide what GPS channels each Starling spacecraft will monitor. As described in[1] we pose and solve the problem as a Mixed Integer Linear Program (MILP). Let S be the set of all Starling spacecraft s_i. Let s_i^c be each Starling spacecraft's channel capacity, the maximum number of GPS satellites it can observe. (Typically all s_i^c are identical; as discussed later, this is an artificial capacity constraint and can be configured operationally to explore different AUTO behavior.) Let s_i^v denote the set of GPS IDs visible to Starling spacecraft s_i. Let $g_j \in s_i^v$ denote GPS satellite with ID j. Let $r_{i,j}^e$ denote the explore reward for spacecraft s_i observing g_j; $r_{i,j}^e \propto \delta(s_i, g_j)$, the distance between the Starling satellite and the GPS satellite. Let $r_{i,j}^x$ denote the exploit reward for spacecraft s_i observing g_j. $r_{i,j}^x \propto TEC(s_i, g_j)$, the TEC reward for s_i monitoring g_j, obtained from the TEC app. Let r^c denote the coverage reward (a constant). Let $G = \cup_i s_i^v$, the set of all GPS satellites visible to at least one Starling spacecraft. Finally, $\alpha, \beta \in [0..1]$ are real-valued parameters controlling the blended combination of explore rewards, exploit rewards, and GPS coverage, corresponding to our requirements described in Sec 2.1.

The MILP has two types of binary $0, 1$ decision variables. Variable $o_{ij} = 1$ if and only if s_i is assigned to observe g_j. Assignments must obey the capacity constraints $\sum_j o_{i,j} \leq s_i^c$ for all $s_i \in S$. Variable $c_j = 1$ if and only if g_j is observed by at least b Starling spacecraft. At minimum, $b = 1$, meaning at least one Starling spacecraft observes g_j. This constraint translates to the following equivalence: $c_j = 1 \Leftrightarrow b \leq \sum_i o_{i,j}$ which can be modeled as linear constraints, leading to the following MILP:

The AUTO App takes in the rewards from the TEC app, normalizes them, and gathers the reward states from the other instances of AUTO running on the other satellites and communicated over the COMM app. This *distributed consensus* approach aims to ensure each Starling spacecraft has the same information, and can solve the same problem, and obtain the same answer. Each instance of the AUTO app uses the rewards it received through the COMM

$$\max \alpha \sum_{i,j} \beta r^e_{i,j} o_{i,j} + (1-\beta) r^x_{i,j} o_{i,j} + (1-\alpha) \sum_j r^c c_j \tag{1}$$

$$\text{s.t.} \sum_j o_{i,j} \leq s^c_i \qquad \forall s_i \in S \tag{2}$$

$$b \geq (\sum_i o_{i,j}) - M_1 c_j \qquad \forall g_j \in G \tag{3}$$

$$b \leq (\sum_i o_{i,j}) - 1 + M_2(1-c_j) \qquad \forall g_j \in G \tag{4}$$

Figure 3: The DSA-Starling Mixed Integer Linear Program (MILP) formulation.

app to generate the observation plan for the next tick.

Note that all of this information transmitted between Starling spacecraft *changes* the MILP from tick to tick; in particular, $r^e_{i,j}, r^x_{i,j}$ are recomputed from the rewards sent between Starling spacecraft. Furthermore, the sets s^v_i change as GPS satellites pass in and out of view. AUTO must therefore *regenerate* the MILP at each tick before it is solved. The fact that the MILPs are constantly changing requires more general purpose automated reasoning capability than a dedicated and highly tuned solver.

AUTO uses lp_Solve as the underlying MILP solver, using a C++ wrapper around the lp_Solve's pure C API to facilitate model construction and solution extraction using cFS. AUTO gathers TEC rewards constructs the MILP model, and extracts the resulting plans.

OPERATIONS

Experiment Periods

Operational products and planning consists of 24 or 48 hours segments known as Experiment Periods (EPs) and the DSA experiment consisted of 28 EP Opportunities. Early mission operations was difficult, and extensive work was required to build and maintain a stable 4-node platform for the Starling mission. Thus, early attempts to run the DSA experiment failed often. DSA ran on orbit during 16/28 (57.14 %) of EP Opportunities. DSA's success rate with orbital operations improved over time. When taking a rolling sum of the last 5 EP deliveries, DSA's success rate hits 80% (see figure 4).

We defined a successful run as the following:

- The DSA mission produced an outcome or delivered a product for execution on orbit.
- The DSA mission successfully ran on orbit and results were delivered.
- This increased the mission's likelihood of meeting full success.

We define a loosely successful run as the following:

- The DSA mission missed an opportunity but was not set back according to our operational schedule.
- This did not change the mission's likelihood of meeting full success.

Of the 16 EPs when DSA ran on orbit, 13 of those are considered successful or loosely successful runs. Thus, our total success rate after platform stability is approximately 86.7%.

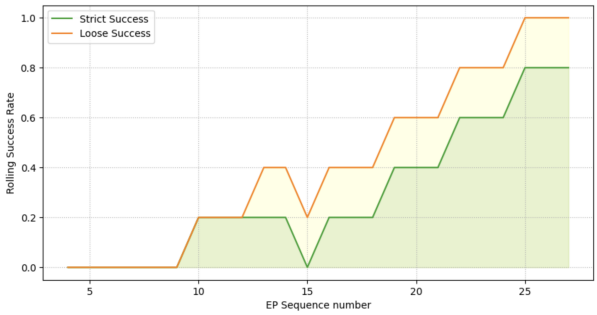

Figure 4: DSA's EP success rate over time, presented as a rolling average of the prior 5 EPs; This demonstrates improving operational capabilities.

Towards Distributed Autonomy

Over the course of 28 EP Opportunities the project moved towards demonstrating requirements, objectives, the goal of distributed autonomy. After each

EP, operations would verify the following criteria occurred before the team would perform a more complex data analysis:

1. **Cross-link Network Status:** Was a cross-link network established?
2. **Node-Node Network Visibility:** When did the first spacecraft view another spacecraft on the cross-link network?
3. **Multi-Node Network Visibility:** When could all of the spacecraft view all of the spacecraft on the cross-link network?
4. **Multi-Node Telemetry:** Did each spacecraft get telemetry from the other spacecraft?

The DSA experiment is entirely dependent on the successful demonstration of the Mobile Ad-hoc Network (MANET) experiment; The MANET experiment first builds a network topology for DSA's Comm application to utilize. If at any point one of the criteria listed above was verified to not have occurred, extensive effort was focused on debugging network stability for the relevant experiment.

DSA achieved initial verification of application requirements early in the mission. A 3-node network on EP Sequence Number 17, marked a positive transition into long streaks of successful runs (as seen in Figure 4). Generally, DSA continued to establish a 3-node network successfully for the remainder of the mission. The TEC Application and AUTO Applications functionality was demonstrated on EP Sequence Number 20. After small patches to our AUTO and TEC applications, reactive operations and autonomy was demonstrated on EP Sequence Number 23. The final EPs expanded on this success and further demonstrated autonomy under a variety of scenarios that the DSA group is still analyzing.

RESULTS

The following section is a more complete analysis of our preliminary results.[7] Here we expand our analysis to include data from our final experiment periods and assess our Key Performance Parameters. Additionally, we focus on singular events of interest that demonstrated distributed spacecraft autonomy.

Network Report

DSA and Starling 1.0 have consistently established a 3-spacecraft cross-link network via our COMM application, as described in detail within figure 5. Our DDS network shares spacecraft state information and group messages. In figure 5 the gray areas represent time periods with no available data; darker gray represents times were data is missing for more than one spacecraft. (Continuous data does exist onboard the spacecraft, but it is not downlinked sequentially; often we must wait long periods of time before full data is available from EPs.) Additionally, the green vertical striped lines represent times when crosslink is turned on and the red vertical striped lines represent times when crosslink is turned off.

Consensus Report

Consensus exists when all of the spacecraft have the same plan. Periods of non-consensus are expected and can occur due to changes in network topology and configuration, GPS satellite visibility set s_i^v changes, changes in explore/exploit rewards $r_{i,j}^e, r_{i,j}^x$, or changes in the AUTO app configuration. Changes often occur within a time step that will cause non-consensus, as simulated in prior work.[1] We have analyzed the interval [01:14:57,02:50:38] in figure 5 (left). Consensus is achieved frequently (893 of 2709 ticks in this period) but often holds for only a few seconds. Figure 6 shows an example of consensus holding for 4 ticks, followed by lack of consensus. Notably, one of the plans generated on the non-consensus tick is 'invalid' in that a GPS satellite that is not in view is assigned (an issue corrected on the next tick).

Coverage Report

Coverage requires every GPS satellites in view were selected by at least one Starling 1.0 spacecraft. Coverage is impossible when s_i^c (DSA channel capacity) is too small to cover s_i^v (GPS satellites visible to Starling spacecrafts s_i.). Coverage is achieved in 689 of the 893 ticks when consensus is achieved in the interval [01:14:57,02:50:38]. Figure 7 shows an instance of coverage.

Latency Report

One definition of *Latency* is the difference between the time an event occurs that requires reconfiguration, and the time that consensus is achieved. The same events that lead to lack of consensus may lead to the need to reconfigure, and thus allow measurement of latency. However, it is possible that numerous events occur in rapid succession, and reconfiguration may not always be needed even when events do occur. Figure 8 shows an example of reactive operations that illustrates latency. In this case, multiple GPS

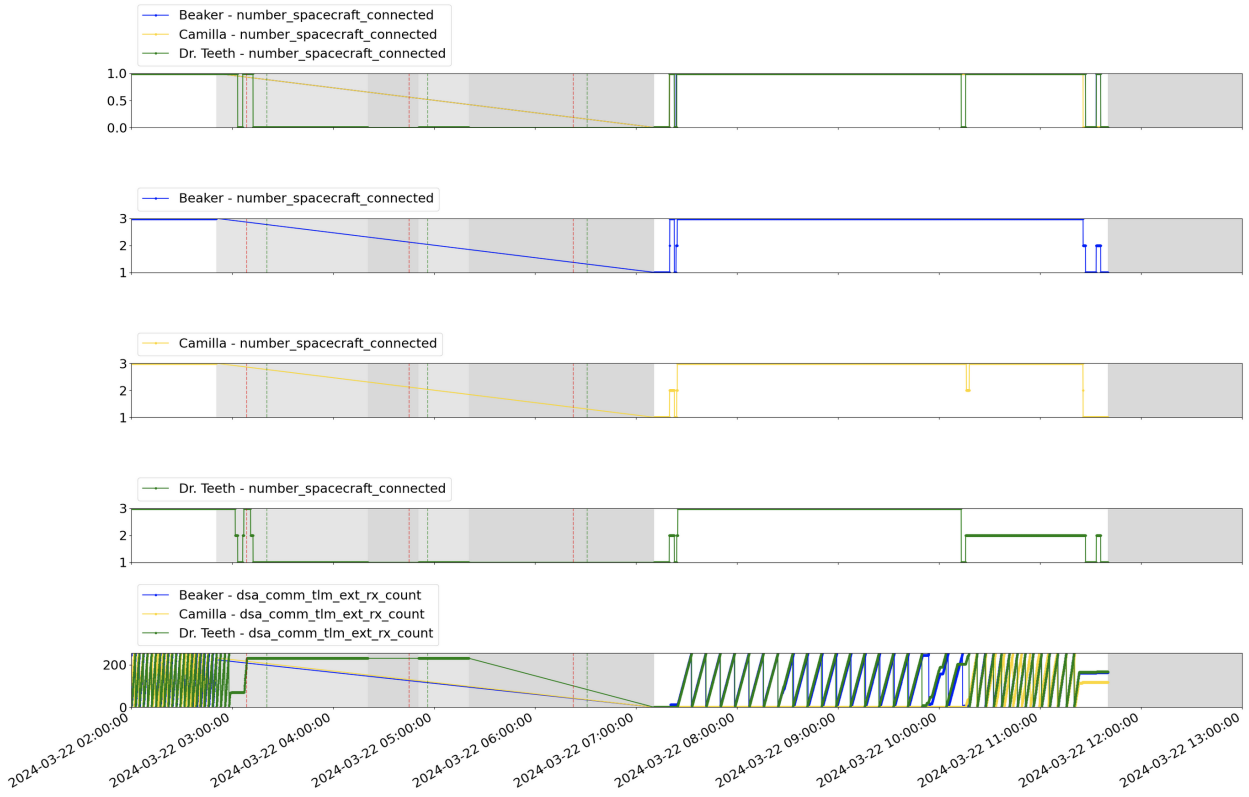

Figure 5: From top to bottom, over a select time period of EP Sequence Number 25, day 1: Times when all spacecraft are connected to the network (1.0) and when a full network is not achieved (0.0); Number of spacecraft connected to Beaker (SV2), Camilla (SV3) and Dr. Teeth (SV4); Accumulated telemetry packets from cross-link on Beaker, Camilla, and Dr. Teeth.

SV	Time	SV2 plan	SV3 plan	SV4 plan
SV2	01:19:03	(16, 17, 27, 30)	(8, 9, 14, 21)	(3, 4, 6, 7)
SV3	01:19:03	(16, 17, 27, 30)	(8, 9, 14, 21)	(3, 4, 6, 7)
SV4	01:19:03	(16, 17, 27, 30)	(8, 9, 14, 21)	(3, 4, 6, 7)
...	...	...	...	...
SV2	01:19:06	(16, 17, 27, 30)	(8, 9, 14, 21)	(3, 4, 6, 7)
SV3	01:19:06	(16, 17, 27, 30)	(8, 9, 14, 21)	(3, 4, 6, 7)
SV4	01:19:06	(16, 17, 27, 30)	(8, 9, 14, 21)	(3, 4, 6, 7)
SV2	01:19:07	(16, 17, 27, 30)	(8, 9, 14, 21)	(3, 4, 6, 7)
SV3	01:19:07	(16, 17, 27, 30)	(9, 14, 21, **22**)	(4, 6, 7, 8)
SV4	01:19:07	(16, 17, 27, 30)	(8, 9, 14, 21)	(3, 4, 6, 7)

Figure 6: A sequence of 4 consecutive periods of consensus at 2024-03-22 01:19:03 - 01:19:06 followed by no consensus at 01:19:07 (indicated by bold font).

satellites leave the visibility sets of DSA spacecraft at different times. However, after a 2-tick period of stability of GPS visibility sets, DSA is able to reach consensus.

SV	SV2 plan	SV3 plan	SV4 plan
SV2	(17, 21, 27, 30)	(7, 8, 9, 14)	(2, 3, 4, 6)
SV3	(17, 21, 27, 30)	(7, 8, 9, 14)	(2, 3, 4, 6)
SV4	(17, 21, 27, 30)	(7, 8, 9, 14)	(2, 3, 4, 6)
SV	Visible GPS Satellites		
SV2	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)		
SV3	(2, 3, 4, 7, 8, 9, 14, 17, 21, 27)		
SV4	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)		
All	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)		

Figure 7: Full coverage and consensus at 2024-03-22 01:16:08; the GPS visibility sets are shown in the bottom of the table.

Key Performance Parameters

Here we present an initial analysis of our Key Performance Parameters (see table 2).

KPP 164: Command Reduction

The products (such as command sequences, configuration tables, application updates, etc) delivered to

SV	Time	SV2 plan	SV3 plan	SV4 plan
SV2	01:14:58	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV3	01:14:58	(17, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV4	01:14:58	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV2	01:14:59	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV3	01:14:59	(**17**,21, 27, 30)	(7, **9**,14, **22**)	(3, 4, 6, 8)
SV4	01:14:59	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV2	01:15:00	(7, 21, 27, 30)	(8, 9, 14, 22)	(3, 4, 6, 7)
SV3	01:15:00	(7, 21, 27, 30)	(8, 9, 14, **17**)	(3, 4, 6, **22**)
SV4	01:15:00	(**17**,21, 27, 30)	(8, 9, 14, 22)	(3, 4, 6, 7)
SV2	01:15:01	(7, 21, 27, 30)	(8, 9, 14, **17**)	(3, 4, 6, 22)
SV3	01:15:01	(**17**,21, 27, 30)	(**7**,8, 9, 14)	(3, 4, 6, 22)
SV4	01:15:01	(7, 21, 27, 30)	(8, 9, 14, **17**)	(3, 4, 6, 22)
SV2	01:15:02	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV3	01:15:02	(17, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV4	01:15:02	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)

SV	Time	Visible GPS Satellites
SV2	01:14:58	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:14:58	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27)
SV4	01:14:58	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)
SV2	01:14:59	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:14:59	(2, 4, 8, 9, 14, 17, 21)(**3**, **6**, **7**, **27**)
SV4	01:14:59	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)
SV2	01:15:00	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:15:00	(2, **3**, 4, **7**, 8, 9, 14, 17, 21)
SV4	01:15:00	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)
SV2	01:15:01	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:15:01	(2, 3, 4, 7, 8, 9, 14, 17, 21, **27**)
SV4	01:15:01	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)
SV2	01:15:02	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:15:02	(2, 3, 4, 7, 8, 9, 14, 17, 21, 27)
SV4	01:15:02	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)

Figure 8: Reactive operations during the period 2024-03-22 01:14:58 - 01:15:02. Consensus is lost at 2024-03-22 01:14:59, then achieved at 2024-03-22 01:15:02, as shown in the top half the table. GPS visibility sets are shown in the bottom half of the table; bold font shows newly visible GPS, the second parenthetical set indicates GPS satellites are no longer visible at 01:14:59. GPS satellite visibility is consistent from 01:15:01 - 01:15:02.

Starling to run are how we measure the reduction of data uplink from the commanding articles. There are two primary assessment scenarios, the ground-based Swarm Comm Test and flight tests.

KPP-164 is measured by creating two sets of Absolute Time Sequences (ATSs) that accomplish identical tasks (an identical sequence of spacecraft states). The control ATSs, one per spacecraft, all issue local commands to activate/idle apps, start Relative Time Sequences (RTSs), and load/validate/activate application tables. The second set of ATSs are demonstration ATSs. All demonstration ATSs contain necessary local commands to do setup. We chose one spacecraft (SV2, also known as Beaker) arbitrarily to be the issuer of the swarm commands at the same times as the corresponding local commands in the control ATSs. Arbitrarily, we chose $2024 - 05 - 17$ (DOY 138) as the representative date for the measurement.

All ATSs:

- Have their first command no earlier than 00:45:00
- Have their last command no later than 23:15:00
- Include scheduled payload restarts as an exogenous event from 06:40:00-07:10, 11:40-12:10, 17:30-18:00, 22:40-23:10. Restarts put the spacecraft back into the same state as 00:45:00, modulo time.

The representative science timeline in the control ATSs and the swarm commander ATS divides a 90-minute orbit up into 15-minute sections of equatorial and polar regions and changes the autonomy priority at the time of entrance to each region. This is only

approximately representative of the Starling orbit, but better accuracy would only significantly affect the timing of the commands and not the number of them since the orbit is polar enough to have two polar (magnitude of latitude $> 70°$) and two equatorial (magnitude of latitude $< 15°$) regions in every orbit.

The method for calculating KPP-164 is as follows:

$$Reduction = 1 - \frac{\sum_{i=1}^{n=4} ATS_i}{4 \cdot ATS_B}$$

Where ATS_i and ATS_B are the compressed size of the demonstration and control ATS tables, respectively. We compress the tables because cFS SC ATS tables are padded to a fixed size and we use the compression to remove redundant padding. Given that, we obtain:

$$Reduction = 1 - \frac{3114}{6114} = 0.493$$

This reduction narrowly falls short of the state-of-the-art. This discrepancy is due to the approach that we chose to implement the swarm commanding. Instead of modifying the cFS commanding system or re-implementing each command structure we wished to issue over the swarm, we chose the more general, reusable approach of wrapping a command-like structure in the swarm command structure. This results in stored swarm commands being often more than twice as large as their local counterparts. This effect was not considered during the KPP formulation and should be considered when designing evaluations for future command reduction experiments.

KPP-164 was originally intended to measure a reduction in commanding from the ground by an operator, in terms of number of commands sent, and was reformulated when the Starling 1.0 mission eliminated the real-time operations component. It was formulated taking "number of bytes uploaded" as a closer analog to "number of commands sent", but if we consider "number of commands uploaded" the closer analog–that is, if we take ATS_i and ATS_B to be the number of commands in the ATS text file from which the ATS table is built, we obtain:

$$Reduction = 1 - \frac{308}{796} = 0.6130$$

The above measurement was made with only 3 of 4 Starling spacecraft. DSA would have likely received an in-flight reduction of $\approx 70\%$, which meets the threshold but does not achieve the project goal, if a command reduction was tested with all 4 nodes.

KPP 165: Reconfiguration Time

This KPP is assessed using the reward scores from the objective function as defined in Auto App Reference Implementation. This KPP is measured using at least three reward ratios tested 0 (pure exploit), 1 (pure explore), and 0.5 a balanced mix of the two. Over operations, this KPP was incrementally tested in more complex and realistic environments. Below are the initial result of operations.

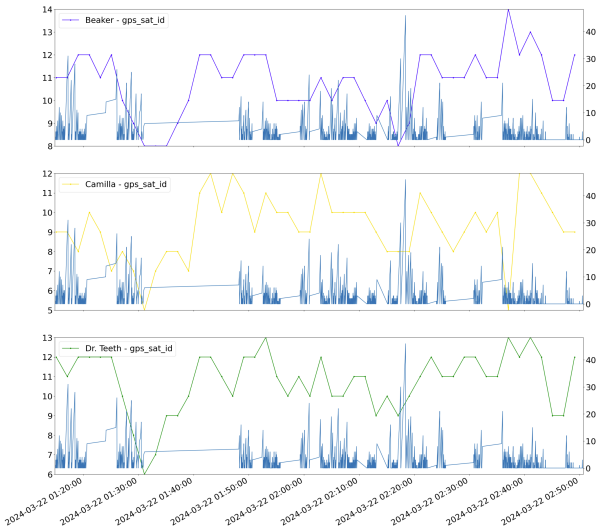

Figure 9: demonstrates time until re-consensus for all 3 operational SVs. Low frequency plots (blue, yellow, green) represent the total visible GPS satellites per spacecraft. The high-frequency, spiky blue line graph represents the total amount of connective time, measured in seconds, spent in non-consensus.

In this case, DSA's flight software is responding to variations in GPS visibility to optimally cover all GPS spacecraft visible to the swarm (optimal coverage is shown in table 7). As new GPS spacecraft become visible, the prior set of assigned swarm observations is no longer optimal. Thus, we expect a Δt in reconfirmation time driven by local and swarm GPS visibility. We expect moments of non-consensus (as defined in the prior section) to occur within our system design. Measuring consensus stability is another meaningful way to measure the swarm reconfiguration time. The following demonstrates the stability of our approach:

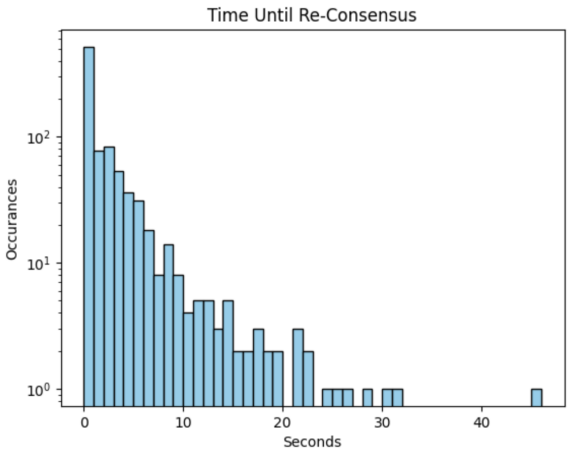

Figure 10: graph demonstrates that, over the selected period of 01:14:57 - 02:50:38 in EP Sequence Number 25, re-consensus happens quickly and often (log scale) with a worst case outlier of 46 seconds

The initial results in figure 11 display DSA reacting to observe relative TEC values in the final experiment period of the mission, EP Sequence Number 28. Unlike prior purely coverage-constrained cases, in this scenario the AUTO application does not choose to observe all visible GPS spacecraft. Notably, this occurs despite the fact that the AUTO application is set to run with a very large channel capacity of 8. This can be seen at 07:33:45, where GPS spacecraft 5 is not observed. During the following timestamp 07:33:46, consensus is lost. Note that the loss of consensus is not caused by a change in GPS visibility, as the visible spacecraft do not change over these selected 4 ticks. Instead, this loss of consensus is caused by a spike in a constituent relative TEC measurement. The reward value can be seen to differ from a measurement of 171.4×10^6 to 182.7×10^6 between SV2 and SV3. Later ticks of 07:33:47 and 07:33:48 stabilize their global reward values to 183.4×10^6 and then 185.4×10^6, thus reaching consensus again for 2 ticks.

The behavior exhibited during this EP is consistent with the design of the MILP, which seeks to maximize reward values. During this experiment period the rewardRatio was set to 0.5 and the exploreRatio was set to 1.0. This provided the AUTO application with a strong incentive to view all spacecraft (though not a requirement) and to prioritize those with large relative TEC measurements.

DSA achieved full success with measurements of KPP165. The Autonomy we've verified above is based on our prior analysis of multi-spacecraft consensus. Consensus exists when all of the spacecraft have the same plan. Periods of non-consensus are expected and can occur due to changes in network topology and configuration; Thus, measuring our responsiveness is what KPP165 was designed to measure. The threshold value for KPP165 was 22 minutes and the project goal was 5 minutes: depending on the EP, our mean responsiveness was between 2.04 - 6.44 seconds. In figure 10 the worst case found was 46 seconds; both of these measurements exceed our project goal of 5 minutes. In figure 9 we see a demonstration that DSA's autonomy system is "stable", as it tends to regain consensus often as it runs. Delays in network communication between nodes also contribute to reconfiguration time; it takes time for local state information to propagate through the network and allows the spacecraft's AUTO apps to solve the same globally optimal problem. Our results demonstrate that, despite network latency, fast 1Hz reactive operations are possible.

DSA Firsts

Previous missions have demonstrated many ingredients of fully autonomous distributed space systems, e.g. space-to-space communications and command relay between multiple spacecraft,[16] and onboard planning[17,18] and reactive operations for a single spacecraft.[19] However, DSA-Starling represents the *first demonstration of a fully autonomous distributed space mission on* 3 *Starling 1.0 spacecraft.* Specifically, DSA-Starling accomplished the following:

- First fully distributed autonomous operation of multiple spacecraft.
- First use of space-to-space communications to autonomously share state information between multiple spacecraft.
- First demonstration of fully distributed reactive operations onboard multiple spacecraft.
- First use of fully distributed automated planning onboard multiple spacecraft.
- First use of a general purpose automated reasoning system onboard a spacecraft.

Acknowledgments

The authors of this paper would like to acknowledge Dr. Nicholas Cramer, who was the Project Manager of DSA during many formative moments. We would also like to thank NASA's Game Changing Development (GCD) program for their support over

SV	Time	SV2 plan	SV3 plan	Total Reward Value
SV2	07:33:45	(4, 6, 9, 11, 16, 18, 25, 28)	(4, 9, 25, 26, 28, 29, 31)	170.8×10^6
SV3	07:33:45	(4, 6, 9, 11, 16, 18, 25, 28)	(4, 9, 25, 26, 28, 29, 31)	170.8×10^6
SV2	07:33:46	(4, 5, 6, 9, 11, 16, 18, 25)	(4, 9, 25, 26, 28, 29, 31)	171.4×10^6
SV3	07:33:46	(4, 6, 9, 11, 16, 18, 25, 28)	(4, 9, 11, 25, 26, 28, 29, 31)	182.7×10^6
SV2	07:33:47	(4, 5, 6, 9, 11, 16, 18, 25)	(4, 9, 11, 25, 26, 28, 29, 31)	183.4×10^6
SV3	07:33:47	(4, 5, 6, 9, 11, 16, 18, 25)	(4, 9, 11, 25, 26, 28, 29, 31)	183.4×10^6
SV2	07:33:48	(4, 5, 6, 9, 11, 16, 18, 25)	(4, 9, 11, 25, 26, 28, 29, 31)	185.4×10^6
SV3	07:33:48	(4, 5, 6, 9, 11, 16, 18, 25)	(4, 9, 11, 25, 26, 28, 29, 31)	185.4×10^6

SV	Time	Visible GPS Satellites
SV2	07:33:45	(4, 5, 6, 9, 11, 16, 18, 25, 26, 28, 29, 31)
SV3	07:33:45	(4, 9, 11, 25, 26, 28, 29, 31)
SV2	07:33:46	(4, 5, 6, 9, 11, 16, 18, 25, 26, 28, 29, 31)
SV3	07:33:46	(4, 9, 11, 25, 26, 28, 29, 31)
SV2	07:33:47	(4, 5, 6, 9, 11, 16, 18, 25, 26, 28, 29, 31)
SV3	07:33:47	(4, 9, 11, 25, 26, 28, 29, 31)
SV2	07:33:48	(4, 5, 6, 9, 11, 16, 18, 25, 26, 28, 29, 31)
SV3	07:33:48	(4, 9, 11, 25, 26, 28, 29, 31)

Figure 11: Reactive operations during the period 2024-05-18 07:33:45 - 07:33:48; Consensus exists at timestamp 07:33:45, is lost at time stamp 07:33:46, and is regained for 07:33:47 and 07:33:48; This demonstrates the minimum possible reconfiguration time (best case) to a spike in relative TEC.

the years. Finally, we would like to thank NASA's Small Spacecraft Technology (SST) program for their support.

REFERENCES

[1] Nicholas Cramer, Daniel Cellucci, Caleb Adams, Adam Sweet, Mohammad Hejase, Jeremy Frank, Richard Levinson, Sergei Gridnev, and Lara Brown. Design and testing of autonomous distributed space systems. In *Proceedings of the AIAA/USU Conference on Small Satellites*, 2021.

[2] Caleb Adams, Brian Kempa, Michael Iatauro, Jeremy Frank, and Walter Vaughan. An overview of distributed spacecraft autonomy at nasa ames. In *Proceedings of the AIAA/USU Conference on Small Satellites*, 2023.

[3] Caleb Adams, Brian Kempa, Walter Vaughan, and Nicholas Cramer. Development of a high-performance, heterogenous, scalable test-bed for distributed spacecraft. In *2023 IEEE Aerospace Conference*, pages 1–8, 2023.

[4] Walter Vaughan, Alan George, Brian Kempa, Daniel Cellucci, and Nicholas Cramer. Evaluating network performance of containerized test framework for distributed space systems. In *2022 Small Satellite Conference*. Utah State University, 2022.

[5] Jason Fugate. Distributed spacecraft autonomy (dsa): Development of swarm autonomy capability and scalability for spacecraft. In *Second AI and Data Science Workshop for Earth and Space Sciences*, 2021.

[6] Daniel Cellucci, Nicholas Cramer, and Jeremy Frank. Distributed spacecraft autonomy. In *2020 AIAA Ascend Conference*. AIAA, 2020.

[7] Caleb Adams, Jeremy Frank, Sergei Gridnev, Michael Iatauro, Brian Kempa, Caroline Lassiter, and Richard Levinson. Advancing autonomy in distributed space systems: Insights from on-orbit testing with the starling 1.0 mission. In *Proceedings of the 2024 Small Satellites Systems and Services Symposium*, 2024.

[8] Hugo Sanchez, Dawn M. McIntosh, Howard N. Cannon, Craig Pires, Joshua Sullivan, Simone D'Amico, and Brendan H. O'Connor. Starling1: Swarm Technology Demonstration. In *Proceedings of the AIAA/USU Conference on Small Satellites*, 2018.

[9] 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.

[10] 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.

[11] Jacqueline Le Moigne, Michael M Little, and Marjorie C Cole. New observing strategy (nos) for future earth science missions. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5285–5288. IEEE, 2019.

[12] Marco D'Errico and Et Alii. *Distributed Space Missions for Earth System Monitoring*. Springer Science and Business Media, 01 2013.

[13] 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.

[14] David McComas, Jonathan Wilmot, and Alan Cudmore. The core flight system (cfs) community: Providing low cost solutions for small spacecraft. In *Proceedings of the AIAA/USU Conference on Small Satellites*, 2016.

[15] Paolo Bellavista, Antonio Corradi, Luca Foschini, and Alessandro Pernafini. Data distribution service (dds): A performance comparison of opensplice and rti implementations. *2013 IEEE Symposium on Computers and Communications (ISCC)*, pages 000377–000383, 2013.

[16] J. Hanson, A. Guarneros Luna, R. DeRosee, K. Oyadomari, J. Wolfe, W. Attai, and C. Prical. Nodes: A Flight Demonstration of Networked Spacecraft Command and Control. In *Proceedings of the Small Satellite Conference*, 2016.

[17] A. Jónsson, P. Morris, N. Muscettola, K. Rajan, and B. Smith. Planning in Interplanetary Space: Theory and Practice. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, 2000.

[18] D. Tran, S. Chien, R. Sherwood, R. Castaño, B. Cichy, A. Davies, and G. Rabbideau. The Autonomous Sciencecraft Experiment Onboard the EO-1 Spacecraft. In *Proceedings of the* 19^{th} *National Conference on Artificial Intelligence*, pages 1040 – 1045, 2004.

[19] F. Ip, J.M. Dohm, V.R. Baker, T. Doggett, A.G. Davies, R. Castaño, S. Chien, B. Cichy, R. Greeley, R. Sherwood, D. Tran, and G. Rabideau. Flood Detection and Monitoring with the Autonomous Sciencecraft Experiment Onboard EO-1. *Remote Sensing of Environment*, 101(4):463–481, 2006.