SSC26-RAII-07

Results from NASA's Distributed Spacecraft Autonomy Project's Extended Mission on Starling

Caleb Adams, Michael Iatauro, Brian Kempa
NASA Ames Research Center
Mail Stop N269-3, Moffett Field, CA 94043; (650) 604-3353
caleb.a.adams@nasa.gov

K. Michael Dalal, Sergei Gridnev, Michael Ho, Caroline Lassiter,
KBR Wyle Services, LLC
Mail Stop N269-3, Moffett Field, CA 94043; (650) 604-0053
michael.dalal@nasa.gov

ABSTRACT

The Distributed Spacecraft Autonomy (DSA) experiments on Starling 1.0 marked a significant milestone as the first demonstration of fully autonomous distributed operations on multiple spacecraft. DSA demonstrated the ability to adapt to measurements of Total Electron Content (TEC) and visibility of GPS satellites in a fully distributed autonomous manner. In subsequent experiments, DSA 1.5 focused on improving its science proxy towards a higher-fidelity data product computed at the edge, as well as addressing emergent autonomy needs from Starling 1.0, particularly local autonomy and robust crosslink data transfer.

The purpose of improving the TEC measurement was to demonstrate that high-quality observation data could be generated onboard comparable to data typically produced through ground-based data processing systems. DSA 1.5 was able to incorporate corrections as well as a measure of data quality, both of which are available for the autonomy to incorporate in its decision-making. The autonomy focus of DSA 1.5 was to develop advanced capabilities for distributed space systems by increasing the robustness, resilience and self-sufficiency of spacecraft. While DSA successfully demonstrated autonomous distributed decision-making onboard Starling 1.0, this was accomplished only in software. To demonstrate the ability to command spacecraft hardware subsystems (e.g., crosslink radios) rather than just re-configuring software would greatly increase confidence in onboard autonomy. Similarly, the ability to safely and autonomously create, and power on and off, the cross-link network during operations will raise mission efficiency by decreasing power consumption, reducing bandwidth, and overall reducing spacecraft mission operation planning. DSA was able to autonomously create, break down, and re-create the crosslink network in response to changes in the radio frequency (RF) environment—TEC was again used as a proxy for demonstration purposes—as well as orbit position. These reactive autonomy demonstrations were driven by PLEXIL, a NASA-developed plan execution technology flown in space for the first time.

Finally, DSA 1.5 demonstrated the ability to send large files over the Ad-Hoc crosslink network by redelivering failed file segments. This redelivery was based on standard transfer measures in the COMM application and an additional layer of autonomy. DSA 1.5 file delivery (known as LFT) was event-driven instead of pre-scheduled based on absolute or relative time, thus more robust operations were enabled. LFT was successfully tested on the DSA 1.5 mission extension, during which the spacecraft successfully and repeatedly transferred 10MB files with guarantees on integrity and validity across multiple network interruptions lasting 60 minutes.

INTRODUCTION AND BACKGROUND

NASA's Starling[1] is a mission through the Small Spacecraft & Distributed Systems program designed to advance technologies for cooperative groups of spacecraft–"swarms". It launched four 6u CubeSats into low-Earth orbit in 2023. Each spacecraft is equipped with one ground-link S-band antenna, two crosslink S-band antennas, a GPS receiver, two star trackers, and an experimental propulsion system called HAMLET[2]. The initial mission consisted of four technology demonstrations:

1. Mobile Ad Hoc Networking (MANET)[1]
2. Reconfiguration and Orbit Maintenance Experiments Onboard (ROMEO)[1]
3. Starling Formation-flying Optical Experiment (StarFOX)[3]
4. Distributed Spacecraft Autonomy (DSA)[1]

These technologies represent key advancements necessary to enable a swarm to accomplish exploration objectives that require a large sensor network or multi-point science data collection. Starling completed its primary mission with full success in 2024.

During the Starling primary mission, DSA conducted demonstrations of a system to perform autonomous, distributed decision-making to gather proxy-science data in a highly dynamic environment where both the observation opportunities and the observable signals changed unpredictably. It used the GPS receiver to simplistically calculate Total Electron Content (TEC) on the path between the receiver and a given GPS satellite using the phase difference of the L1 and L2 GPS signals. TEC is a useful science proxy measurement in part because it can be calculated easily without additional hardware, but mostly because there are two phenomena that exhibit different behaviors and motivate different observation strategies: at the magnetic equator, there is a consistent and spatially extended region of elevated TEC called the "equatorial ionization anomaly"; at the poles, there are patches of ionization whose existence, location, size, and shape are highly variable.

The function of the autonomy was to assign GPS signal observations to each of the Starling spacecraft in a completely federated manner. At a 1Hz cadence, the autonomy on each spacecraft would take the most recent TEC measurements it had made, share that with the other spacecraft, combine it with the TEC measurements shared by the other spacecraft, generate a plan for the entire swarm for the next observation cycle, and then execute its own part of the plan. The autonomy had several parameters to adjust its decision making:

1. The number of GPS satellites each spacecraft could observe.
2. The minimum number of spacecraft needed to listen to a GPS satellite for it to be considered "observed".
3. How to balance the three observation priorities of:
    a. Total number of GPS satellites observed
    b. TEC measured
    c. Volume of space observed

The limitation on the number of observable satellites was necessary to make this representative of a situation where there are more targets of interest than opportunities to observe them. Without it, the specific case of TEC is trivial from an autonomy standpoint because GPS receivers can receive signals from more GPS satellites than can be visible to them at one time.

When all the spacecraft were able to successfully share their observations at each cycle, they would all have the same information and thus generate the same plan for the entire swarm. We called this "consensus". If any spacecraft had different information for any reason, the swarm would still plan and execute with the lack of consensus. This approach prioritizes agility of response over consistency of the swarm's behavior, which can be preferred in situations where an enforced consensus before execution could miss an observation opportunity or where the whole swarm doesn't need to respond to a signal.

The sharing of information was accomplished using the crosslink radios and the Better Approach To Mobile Ad-hoc Networking (BATMAN)[4] protocol to manage routing, over which DSA implemented a "COMM" application that used RTI Connext DDS Micro[5] to manage message transfer. With this, most losses of consensus were restored in under six seconds, much of the time in just two.

The Starling primary mission was one year long. When planning for an extended mission, DSA took its pre-launch wish-list as well as the lessons learned from a year of flight and selected three items to prioritize. First, we wanted to enhance the on-board TEC calculation to something scientifically useful. The DSA 1.0 calculation was sufficient to motivate the autonomy, but the quality of science autonomy improves with the quality of the science data available for the autonomy to reason with.

Second, we wanted to enhance the autonomy so that it could reconfigure itself and control spacecraft hardware–specifically, the crosslink radios–in response to changing events. During DSA 1.0, both tasks were performed by the DSA operations team and managed on-board using time-sequenced stored commands. The effort to coordinate these time sequences was appreciable and would sometimes result in what amounted to waste–command products uplinked, autonomy configured and crosslink network established, draining battery, but without polar ionization to observe.

Finally, we wanted to take advantage of the better availability and transfer speeds of the crosslink radios over the ground link to share files among the swarm. Uplinks from the ground were limited to 1 MB per day for most spacecraft, and one had experienced a radio failure that reduced that to 200 KB and made it unreliable. The crosslink radios, on the other hand, could be on and transmitting as much as 50% of the time without risking low battery state-of-charge, and protocols like BitTorrent had well demonstrated the advantages of decentralized file transfer.

TEC MEASUREMENT IMPROVEMENTS

Previously, DSA computed relative TEC using the geometry-free combination of carrier phase observations. However, because carrier phase observations are inherently ambiguous, they require leveling to pseudorange TEC to resolve phase ambiguity. DSA did not perform this leveling, which limited the scientific value of the resulting relative TEC measurements. To generate more scientifically meaningful data, we have transitioned to calculating absolute TEC using carrier-smoothed pseudorange observations. By leveraging the default 100-second pseudorange smoothing on Starling's NovAtel OEM7 receiver, we can natively mitigate high-frequency noise and filter out some multipath effects without having to implement a custom carrier-phase leveling algorithm.

While pseudorange smoothing partially mitigates multipath effects, enhancing the accuracy of TEC measurements requires the application of dedicated multipath error correction maps. For LEO antennae, multipath is generated due to signal reflection off of the spacecraft body6. The multipath error is thus largely dependent upon the spacecraft configuration, and can be large if reflective surfaces, such as solar panels, are within the antenna field of view[6,7]. Due to project schedule constraints, the generation of these multipath maps – originally planned in collaboration with the University Corporation for Atmospheric Research (UCAR) – has been deferred to future development.

In addition to using carrier-smoothed pseudoranges, we apply transmitter and receiver Differential Code Biases (DCBs) to our TEC measurements. For transmitter DCBs, monthly estimates provided by the Center for Orbit Determination in Europe (CODE) are uplinked to Starling before each experiment period. These estimates are directly applied to our TEC measurements on orbit. Following the method outlined by (8), we calculate receiver DCB using a least squares method. The basic assumption of this method is simultaneous observations of absolute TEC mapped to the vertical direction are equal. Mapping the slant TEC to vertical TEC is achieved by a mapping function depending on the elevation angle of the GPS ray[8]. All paired observations throughout the experiment period are used to create a system of linear equations, and the receiver DCB is derived by applying the method of least squares. This approach assumes a spherical ionosphere without significant horizontal gradient, which only holds when vertical TEC is low e.g., TEC $\leq$ 3 TECU[8]. We therefore limited the TEC data used for the receiver DCB estimation to measurements calculated from ray elevation angles $\geq 45°$, which follows what was done in (8).



Figure 1: TEC-Measured TEC

The next critical step is validating these improved, absolute TEC measurements. Typically, evaluation is done by comparing TEC measurements taken concurrently from a separate satellite in a similar orbit[9] . Thus, we plan to validate Starling's TEC data by comparing it against absolute TEC measurements taken by COSMIC-2 at the same location and time. With an orbital altitude of 550 km, COSMIC-2 provides an ideal comparative baseline for Starling's 575 km orbit.

REACTIVE AUTONOMY AND PLEXIL

Fully autonomous commanding of hardware and software systems in response to TEC measurements and latitude/longitude changes, with concurrent monitoring and enforcement of safely limits such as radio duty cycle, was provided by PLEXIL (PLan EXecution Interchange Language)[10], a NASA-developed, open-source software system[11], flown in space for the first time as a DSA 1.5 application on Starling. The PLEXIL system includes a language for encoding autonomy plans, a fast and lightweight executive to execute these plans, and an interfacing framework that supports a variety of protocols and messaging middleware for connecting plans and the executive to real or simulated systems.

PLEXIL Language

The PLEXIL language represents plans as a hierarchy of tasks, or *nodes*, where each node is entirely condition and event-driven. There are roughly a half dozen node types. Leaf nodes types include those that send commands, perform an assignment, and invoke an

external plan. The single internal and root *List Node* type gives the plan its tree structure and specifies aspects of its subtree's execution. A node can be annotated with any combination of *gate* conditions that specify when the node can start, end, repeat, or be skipped; and *check* conditions (*precondition*, *invariant*, *postcondition*) that determine the success or failure of the node. PLEXIL is *reflective* (within scoping rules) – nodes can query and act on the states and outcomes of neighboring nodes. Composition of these features enables a variety of control flows (sequential, concurrent, partially ordered) and support sophisticated autonomy for tasks such as health monitoring, fault handling, pause/resumption/abort of procedures, and mission phase transition. PLEXIL interacts with the external world primarily through its *lookup* and command constructs. Lookups read values from the environment and provide abstraction over telemetry; e.g. a lookup (which can have parameters) can compute a function over several telemetry points. Commands effect changes in the environment and provide an abstraction over spacecraft system commands. PLEXIL is a relative newcomer in the lineage of *synchronous programming languages*[12] which are well-tailored for reactive autonomy.

PLEXIL is rooted in its formal semantics with the aim of making plans robust, explainable, and amenable to formal verification and validation. The operational semantics of PLEXIL[13] is presented as a stacked layer of relations whose rules are specified in rewrite logic. The highest layer is the plan's start-to-finish execution as a response to a sequence of environment events. Lower layers specify the state transitions of individual nodes and the evaluation of expressions. Early in its design, PLEXIL's formal semantics was implemented in the Program Verification System (PVS)[14], a theorem prover that mechanically verified important properties in PLEXIL such as operational determinism and run-to-completion in all the semantic layers. Later on, a model checker called PLEXIL-V was developed by the NASA Langley Research Center and used in several NASA projects including DSA. The PLEXIL-V[15] toolset translates PLEXIL plans into its Maude[16] representation and checks properties specified by the user in Linear Temporal Logic[17] by executing the plan in an environment that can have non-deterministic inputs. If a property is violated, a counterexample environment is returned. Though no DSA 1.5 PLEXIL plans were pre-flight checked with PLEXIL-V, several of them were used to test PLEXIL-V, a process that led to many improvements in the model checker. Over the course of the mission, PLEXIL-V was matured to a level where it is now intended to be used in pre-flight verification of future DSA PLEXIL mission plans.

PLEXIL and DSA

For use as DSA flight software the PLEXIL executive was embedded in a Core Flight System (cFS)[18] application. The cFS layer of the PLEXIL application supports the Lookup construct by subscribing to relevant messages and defining message handlers that "wake up" the PLEXIL executive (i.e. induce an execution cycle) and also cache the telemetry value for subsequent Lookups. Similarly, PLEXIL commands are mapped to functions that publish the cFS messages needed to effect the command. The application has its own interface for operation by ground control. There are commands to enable/disable the application, load and run plans, and to load *library plans*, which are (often parameterized) PLEXIL plans strictly invoked from other plans. Library plans provide the means for a top-level plan P that uses library L to be loaded onto spacecrafts $S_1..S_n$ and have spacecraft-specific definitions or behaviors given by library variants $L_1..L_n$ that are loaded to the respective spacecraft.

We know of no other autonomy language that has all the aforementioned features. PLEXIL was chosen for DSA because of its success in prior NASA efforts and expertise within the DSA team – several of its members were among the early PLEXIL developers. Figure 1 contains a segment of a plan flown in DSA which illustrates several properties of PLEXIL including concurrent monitoring, event and condition driven node execution, iteration, commanding, use of library plans.

PLEXIL in Flight

Plexil's first flight was focused on demonstrating baseline capabilities. There were many successes during a baseline endurance test of the app: EVS messages from PLEXIL in all 4 categories: Info, Error, Critical, Debug; commanding another app by sending NOOPs to COMM; and verified that a hierarchy of plans could be loaded and run. Plexils second task was to control the crosslink radios. Powering the crosslink radios drains the spacecraft battery, which should be avoided when possible, but the radios still needed to be on so that the spacecraft could observe TEC features. Before interfacing with the TEC app and leveraging its feature detection capabilities, we decided to turn on the crosslink radios at latitudes at which high TEC is expected. Around the the equatorial bulge and polar patches, there is a significant rise in Total Electron Count (TEC) and these regions can be roughly defined by latitude. Plexil detects from GPS telemetry that the spacecraft are moving: into a polar region (i.e. crossing +80º going north or -80º going south), out of a polar region (i.e. crossing +80º going south or -80º going north), into an equatorial region (i.e. crossing +15º going south or -15º going north), or out of an equatorial region (i.e. crossing

+15º going north or -15º going south). Plexil commands the crosslink radios on when entering regions of high expectation and off when leaving them. This approach gives the system more time to observe features, in theory, as the radios are expected to be on in advance of feature detection, increasing the period of time during which a feature is detected and the crosslink radios are on.

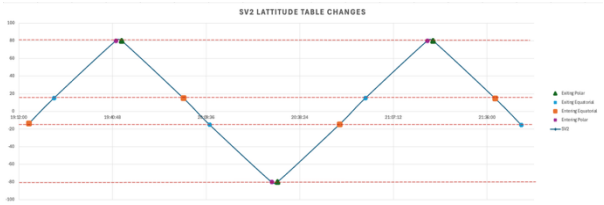


Figure 2: PLEXIL Table and Crosslink Changes vs. Latitude

A more specific way to address the problem of battery drain versus feature observation was to have a radio-controlling Plexil plan respond to output from the TEC app, which indicates whether or not there is a feature of interest that we would want to observe. When the TEC app reports that a TEC feature is detected, Plexil turns the radios on. This entails observing what the reward weights are set to, and once they cross a threshold which indicated that there is a feature to observe, the crosslink radios are commanded on. This was, in part, to minimize power drain. When a feature is detected, the crosslinks are powered on so that the swarm can share information about observing the feature. There are two versions of this plan, TecRadios and TecRadiosEnhanced that built on the former. The baseline plan turns the radios on when a feature has been detected, if the crosslinks are off and have been off for at least 30 minutes and turns the radios off if they have been on for at least 30 minutes. The enhanced version has this concurrent logic:

1. When crosslink radios are off, and they have been off for at least 30 minutes, and some TEC feature is detected, turn the radios on.
2. When crosslink radios have been on for at least 30 minutes, turn them off.
3. When crosslink radios are off and have been off for less than 30 minutes and some TEC feature is detected, log a missed opportunity.
4. When crosslink radios are on, and have been on for at least 10 minutes, and some TEC feature is no longer detected, turn off the crosslink radios.

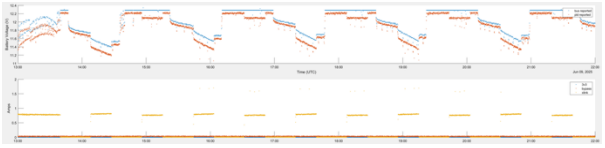


Figure 3: Battery voltage (top) and Crosslink Power Draw as controlled by PLEXIL

Building off of the previous PLEXIL plan, we have proven the ability to detect regions of interest, during which DSA would want to have the swarm communicate about features of interest. In addition to having the crosslinks on, the Autonomy app should be configured to maximize the swarms success during feature observation. Scenario2 demonstrated our ability to configure the Auto app by loading different configuration tables that specify different reward and coverage weights, inputs which determine how the swarm decides to observe a feature. The final Plexil plan that DSA ran in orbit was made to take over some of the work that the ATS was doing. Especially during LFT testing, it became important to be able to have the radios on for longer and scheduled periods of time. Until this point, the ATS scheduled these periods and commanded the crosslink radios. In order to preserve battery but also allow for files to be transferred over crosslink, a 1 hour duty cycle for the crosslink radios was chosen. This allows swarm activities to take place every other hour, while avoiding safe mode. Additionally, the plan does not rely on other app telemetry, providing a reliable schedule for when the crosslink radios would be on, which is required if using an ATS that is dependent on time, specifically.

Table 1 describes the primary flight scenarios for PLEXIL as well as their outcomes.

LARGE FILE TRANSFER

As previously published[19], our team developed a torrent-inspired Large File Transfer (LFT) system. The LFT system was specifically designed to minimize ground uplinks and enable opportunistic data transfer within small satellite swarms without the ground in the loop. Our specific implementation was influenced by our operational experience on the initial DSA-Starling

```
RadioLoop: {
   Repeat true;
   Monitors: {
     RadiosOn: {
       // When crosslink radios are off, and they have been off for at least
       // 30 minutes, and some TEC feature is detected, turn the radios on.
       String message = "TEC feature detected! Turning crosslink radios ON.";
       Start (!radiosOn &&
              Lookup(time) >= radiosOffTime + RadiosOffMin &&
              Lookup(TecFeatureDetected()));

       if (DebugEVS) dsa_debug(message);
       else dsa_info(message);
       endif;

       LibraryCall CrosslinkRadiosPowerOn();
       radiosOn = true;
       radiosOnTime = Lookup(time);
     }

     RadiosOff: {
       // When crosslink radios have been on for at least 30 minutes, turn
       // them off.

       String message = "Xlink radios are at their limit, turning them OFF.";

       Start radiosOn && Lookup(time) >= radiosOnTime + RadiosOnMax;

       if (DebugEVS) dsa_debug(message);
       else dsa_info(message);
       endif;

       LibraryCall CrosslinkRadiosPowerOff();
       radiosOn = false;
       radiosOffTime = Lookup(time);
     }
   }
 }
```

Figure 4: PLEXIL Plan Segment--Crosslink Control

mission while updating software experiments onboard the spacecraft. The technology requirements were motivated by the size limits of uplink data products per pass (~1MB), the failing receive radio on one of the spacecraft (making software updates nearly infeasible), and the healthy crosslink radios on the spacecraft. LFT relies on a torrent-inspired design that intelligently re transmits chucks of data from nodes with the available data to nodes that request the available data. We defined a "large file" to be a 10MB, which is on the order of a full flight software build for one spacecraft. Our results showed that the LFT system can efficiently facilitate strategic coordination and large file transfers among swarm members. The system demonstrated smart fail-over and re-transmit capabilities throughout several experiments in 2025.

FUTURE WORK

DSA is in a second extended mission with Starling, Swarm Tracking and Relative Dynamics for Optimized Maneuvering (STARDOM). STARDOM will continue to expand on DSA 1.0 and DSA 1.5's capabilities into planning observations of and with resident space objects (RSOs) of interest. One high-level objective is to collect data to validate the TEC calculation completely autonomously. STARDOM will select from an on-board catalog of RSOs high-value candidates that can measure TEC and that will see a similar region of the sky, plan

Table 1: PLEXIL Scenario Execution

Plan Name	Flight Date (2025)	Summary	Outcome
Hello	Mar. 22	PLEXIL Hello World	Success
LibraryTest	Apr. 06	Verifying PLEXIL language features	Success
CommandDemo LatLonDemo	May 04	Verifying PLEXIL's ability to receive latitude and longitude telemetry and to issue commands	Success
LatLonRadios	June 01	Commanding crosslink radios in response to latitude and longitude	Failure
LatLonRadios	June 09		Success
TecRadios	June 23	Commanding crosslink radios in response to TEC feature detection	Success
TecRadiosEnhanced	June 28	Adding reporting of TEC features detected while radios are off due to flight rule	Success
Scenario2	July 25	AUTO application configuration in response to latitude and longitude crossings	Failure (no GPS tlm)
Scenario2	Aug. 8		Success
BattVoltTest	Aug. 22	Commanding crosslink radios in response to battery voltage	Failure (unrelated crash)
BattVoltTest	Sept. 05		Failure (incorrect voltage tlm)
CycleRadiosDemo+Scenario2	Sept. 19	Manage radios and control AUTO configuration in parallel plans.	Success
CycleRadiosDemo+Scenario5_1	Oct. 31	PLEXIL controlling all other DSA applications for "set it and forget it" science.	Partial Success (no x-link)

TEC measurements to be taken while co-located with those RSOs, and execute the measurements. After the fact, we will request the corresponding data from our collaborators to validate our results.

We will continue to expand our use of PLEXIL to coordinate the growing number of components of the DSA flight software, including the observation planner, a maneuver planner, an orbit predictor, and we will integrate with StarFOX to enable star tracker observations of RSOs. PLEXIL will also be executing these plans--among the original design intentions for PLEXIL was that the language be a translation target for automated planners, and we will be exploring its use for that purpose as the plans we generate become more complicated.

Using PLEXIL-V for model checking PLEXIL plans has proved useful and DSA will explore its use in operations by checking human-authored plans as a pre-flight verification step alongside simulation.

LFT succeeded in its objectives, but much remains to be understood about the behavior of the protocol we designed. At its largest, Starling had only three spacecraft participating in the swarm, which is the smallest a swarm can be. Fortunately, DSA built the Distributed Intelligent Space System Test Rack (DISSTRack)[20] to test swarm software at scale, and we will explore using the DISSTRack to study how LFT performs with more spacecraft, more files to transfer, and a less reliable network with greater latencies.

CONCLUSION

DSA 1.5 built on DSA 1.0's successes and demonstrated enhanced TEC computation at the edge, making better data available to better autonomy.

We utilized a mature autonomy technology with a long history of successful use on Earth to demonstrate the value of enhanced local autonomy capabilities (including hardware control) to improve swarm autonomy. This allowed us to keep separate the concerns of what the swarm autonomy's priorities are from how the swarm autonomy applied those priorities while improving the robustness of the individual members of the swarm.

Finally, we showed that our COMM infrastructure was flexible enough to encompass robust, decentralized transfers of large files between swarm members. This brings us closer to the goal of treating the swarm as a holistic system rather than individuals, turning one

uplink per spacecraft per day into four uplinks to the swarm per day.

The DSA team accomplished this while the mission was flying, in response to emergent mission needs, essentially building the boat while sailing. The team was able to take features from requirements and design through to testing in flight in less than a month. DSA's flight cadence through the 1.5 mission was a biweekly run of 48 hours, and flight software updates with features and bug fixes were delivered alongside the command products for most of those runs.

These capabilities, both of software–autonomy at the individual and swarm level, high-quality edge computation--and of development–rapid implementation of reliable software--will be critical as large numbers of low-cost spacecraft become commonplace and even infrastructural around the Earth, in cislunar space, and out into the solar system.

ACKNOWLEDGEMENTS

DSA 1.5 was funded by the Small Spacecraft and Distributed Systems (SSDS) program within NASA's Space Technology Mission Directorate. DSA 1.0 was funded by the Game Changing Development (GCD) program within STMD.

The DSA team would also like to thank:

The Starling team, our partners in flight.

Micky Olynik, NovAtel Senior Applications Engineer for his contributions to the TEC computation.

John Braun, Jan Weiss, and William Gullotta of UCAR for their assistance with TEC measurements.

Chuck Fry for his tireless work on the PLEXIL executive implementation.

REFERENCES

List and number all bibliographical references at the end of the paper. When referring to references in the text, type the corresponding reference number in superscript form as shown at the end of this sentence.1 Use the **References** style for formatting citations, as shown in the following examples:

1. Miller, S. Adams, C., et al. “Starling CubeSat Swarm Technology Demonstration Flight Results,” Proceedings of the Small Satellite Conference, Logan, UT, 2024.
2. Stevenson, T. “Flight Results and Lessons Learned From the Starling Propulsion System,” Proceedings of the Small Satellite Conference, Logan, UT, 2024.
3. Kruger, J. and D'Amico, S., "On-Orbit Performance and Lessons Learned for Autonomous Angles-Only Navigation of a Satellite Swarm," *2025 IEEE Aerospace Conference*, Big Sky, MT, USA, 2025
4. Neumann A, Aichele C, Lindner M, Wunderlich S. Better approach to mobile ad-hoc networking (BATMAN). IETF draft. 2008 Apr 7:1-24.
5. https://www.rti.com/products/connext-micro, retrieved 2026-06-24.
6. Pedatella, N. M., Zakharenkova, I., Braun, J. J., Cherniak, I., Hunt, D., Schreiner, W. S., et al. (2021). Processing and validation of FORMOSAT-7/COSMIC-2 GPS total electron content observations. Radio Science, 56, e2021RS007267.
7. Yue, X., W. S. Schreiner, D. C. Hunt, C. Rocken, and Y.-H. Kuo (2011), Quantitative evaluation of the low Earth orbit satellite based slant total electron content determination, Space Weather, 9, S09001
8. Noja, M., C. Stolle, J. Park, and H. Lühr (2013), Long-term analysis of ionospheric polar patches based on CHAMP TEC data, Radio Sci., 48, 289–301, doi:10.1002/rds.20033.
9. Pedatella, N. M., Zakharenkova, I., Braun, J. J., Cherniak, I., Hunt, D., Schreiner, W. S., et al. (2021). Processing and validation of FORMOSAT-7/COSMIC-2 GPS total electron content observations. Radio Science, 56, e2021RS007267.
10. V. Baskaran, M. Dalal, T. Estlin, C. Fry, M. Iatauro, R. Harris, A. Jonsson, C. Pasareanu, R. Simmons, V. Verma, “Plan Execution Interchance Language (PLEXIL) Version 1.0”, NASA Technical Memorandum, Nov 2007.
11. https://github.com/plexil-group/plexil as accessed 2026-06-16.
12. N. Halbwachs. *Synchronous programming of reactive systems*. Kluwer Academic Pub., 1993.
13. Dowek, G., Muñoz, C., and Pasareanu, C., “A Formal Analysis Framework for PLEXIL”, 3rd Workshop on Planning and Plan Execution for Real-World Systems, 2007.
14. Owre, S., Rushby, J., and Shankar, N.. PVS: A prototype verification system. In Deepak Kapur, editor, 11th International Conference on Automated Deduction (CADE), volume 607 of Lecture Notes in Artificial Intelligence, pages

748–752, Saratoga, NY, June 1992. Springer-Verlag.

15. Rocha, C., Cadavid, H., Munoz, C., and Siminiceanu, R., "A Formal Interactive Verification Environment for the Plan Execution Interchange Language." In International Conference on Integrated Formal Methods, pp. 343-357. Springer, Berlin, Heidelberg, 2012.
16. https://maude.cs.illinois.edu, accessed 2026-06-19.
17. https://en.wikipedia.org/wiki/Linear_temporal_logic, accessed 2026-06-19.
18. https://etd.gsfc.nasa.gov/capabilities/core-flight-system/ as accessed 2026-06-18
19. Adams C, Iatauro M, Kempa B, Gridnev S, Vlastos P, Lassiter C. A Torrent-Inspired Large File Transfer System for Small Satellite Swarms. InSmall Satellite Conference 2025 Aug 13 (No. SSC25-XI-06). AIAA/USU.
20. Adams C, Kempa B, Vaughan W, Cramer N. Development of a high-performance, heterogenous, scalable test-bed for distributed spacecraft. In2023 IEEE Aerospace Conference 2023 Mar 4 (pp. 1-8). IEEE.
21. Wyatt, C.L., Radiometric Calibration: Theory and Methods, Academic Press, New York, 1978.