

Flight Test of the autonomous Navigation, Guidance, and Control (autoNGC) Software on the Starling Swarm

Robert Pritchett, Nathan Stacey, Sun Hur-Diaz, Michael Romeo, Nickalas Cason, Katherine Fowee Gasaway,
Nathan Esantsi, Behnam Azimi, Sean Semper, Hossein Mohammadkhan, Grant Hecht

NASA Goddard Space Flight Center
8800 Greenbelt Rd., Greenbelt, MD, 20771
robert.e.pritchett@nasa.gov

Joseph Patton, Liam Greenlee
Aurora Engineering
15 Main St., Unit B, Wilton, NH 03086
joseph.w.patton@nasa.gov

Justin Clavette
ADNET Systems LLC
6720B Rockledge Dr., Suite 504, Bethesda, MD, 20817
justin.d.clavette@nasa.gov

Alan Cudmore
Microtel LLC
11950 Democracy Drive, Suite 650, Reston, VA, 20190
alan.p.cudmore@nasa.gov

Ian Hammer
NASA Wallops Flight Facility
34200 Fulton St., Wallops Island, VA, 23337
ian.m.hammer@nasa.gov

Ted Hendriks
Metis Technology Solutions
2309 Renard Pl. SE Albuquerque, NM, 87106
theodore.g.hendriks@nasa.gov

Terry Stevenson
KBR Wyle Inc.
NASA Ames Research Center, Moffett Field, CA 95054
terry.h.stevenson@nasa.gov

Nahum Alem
NASA Ames Research Center
PO Box 1, Moffett Field, CA, 94035
nahum.alem@nasa.gov

ABSTRACT

Performing real-time autonomous guidance, navigation, and control (GN&C) functions onboard spacecraft is critical for enabling greater spacecraft autonomy. The autoNGC (autonomous Navigation, Guidance, and Control) software suite is a collection of applications, built on the NASA core Flight System (cFS), that provides these capabilities. This paper presents results from the first flight demonstration of autoNGC, conducted on the NASA Starling mission in 2025. The Starling spacecraft's use of cFS enabled autoNGC to be uplinked and run onboard despite the spacecraft already being in orbit. The flight test had three goals: to demonstrate onboard GPS navigation, onboard state and uncertainty prediction for space traffic management, and onboard maneuver planning. The autoNGC navigation filter,

GEONS (Goddard Enhanced Onboard Navigation System), was configured to generate state estimates from incoming GPS measurements. These estimates were then used to predict the spacecraft state and associated error covariance over the following three days and to compute station-keeping maneuvers. Flight test results demonstrated successful completion of all three goals while also highlighting opportunities for additional autoNGC enhancement. The Starling flight test was a critical milestone in advancing autoNGC's technology readiness level, achieving TRL 7 for low-Earth orbit by validating the ability to perform navigation, prediction, and maneuver planning onboard. This autonomy is especially valuable for constellations and other operationally demanding distributed systems missions. The wide adoption of cFS within the SmallSat community and the availability of key autoNGC apps on the NASA software catalog mean other missions could potentially leverage autoNGC to enhance their spacecraft autonomy.

INTRODUCTION

Ambitious space exploration goals and the constraints imposed by current technology are driving an effort to develop spacecraft with more autonomous guidance, navigation, and control (GN&C) capabilities.¹ NASA Moon to Mars objectives are dependent upon the maturation of technologies that enable humans and machines to operate farther from Earth, with greater autonomy.² Onboard GN&C capabilities support this vision by, for example, equipping spacecraft with onboard knowledge of their state and time, which can be used to independently recover from safe mode events or ensure crew safety during extended periods of loss of signal. Moreover, when provided by a network of spacecraft, this state and time knowledge could enable astronauts and other ground assets to determine their position on the surface of the Moon or Mars.

An increasing number of cutting-edge science missions require onboard GN&C. In some cases, this technology facilitates precision formation flying that enables scientific observations that are otherwise impossible to achieve. In addition, scientific missions to the outer solar system, particularly those requiring low-latency operations, e.g., proximity operations around a small body, benefit from onboard GN&C due to light-time delays that inhibit ground in-the-loop operations. Finally, onboard GN&C alleviates the rapidly growing demand placed on a limited number of ground assets, particularly the Deep Space Network (DSN), by enabling spacecraft to operate for longer periods without contact with Earth. While the accelerating pursuit of onboard GN&C capabilities is driven by current challenges and constraints, this development builds on decades of prior work.

The benefits of onboard GN&C capabilities, including enhanced spacecraft autonomy, have long been recognized. In the interest of brevity, this review focuses solely on robotic spacecraft, though the human spaceflight program has its own rich history of developing onboard GN&C. The Deep Space 1 mission was a milestone in demonstrating various autonomy and GN&C technologies.^{3,4} This mission demonstrated the first onboard autonomous navigation system, called AutoNav.^{5,6} The AutoNav system on Deep Space 1 used

celestial navigation to provide measurements to an onboard filter that produced state estimates that were employed to plan thrust vectors for the ion propulsion system. Subsequent deep space and Earth-orbiting missions offered additional opportunities to utilize and expand AutoNav capabilities.^{6,7,8} In particular, the Deep Impact mission relied on AutoNav to guide a spacecraft to a high-speed impact with a comet,⁹ a capability that can be expanded to enable autonomous landing systems.¹⁰ Similarly, the Double Asteroid Redirect Test (DART) used its own onboard navigation system, SMART Nav, to guide the DART spacecraft during its approach and impact with the asteroid Dimorphos.¹¹ For the Deep Impact and DART missions, onboard navigation was necessary to successfully impact the target bodies and fulfill mission objectives. Onboard navigation and control is critical for less destructive rendezvous problems as well. The low-latency operations required for rendezvous, proximity operations, and docking (RPOD), and for formation flying necessitate greater onboard autonomy. Autonomous formation flying, utilizing onboard autonomous maneuver control, was first demonstrated by NASA in 2001 when the Earth-Observing 1 (EO-1) spacecraft was repurposed to fly in formation with Landsat-7.¹² Since this demonstration, a number of other missions, including PRISMA,^{13,14} PROBA-3,^{15,16} CANX-4 and CANX-5,¹⁷ have demonstrated high-precision autonomous formation flying. The Starling mission, led by NASA Ames, continues this history of technology demonstration missions that test onboard GN&C technologies, including autoNGC.

The Starling flight test campaign marked the first flight test of the autoNGC software suite, a collection of cFS apps that provide onboard GN&C capabilities. This test demonstrated several core components of autoNGC: real-time navigation, orbit prediction, and maneuver planning. Together, these capabilities provide the tools that spacecraft require to autonomously determine their present state, predict their future state, and take action to achieve a desired state. Demonstrating autoNGC functionality in low Earth orbit (LEO) advances the technology readiness level of the software suite, thereby reducing the cost, schedule, and risk required to utilize autoNGC for future missions in LEO and beyond.

This document highlights the key results and lessons learned from a successful demonstration of autoNGC in LEO. First, an overview of the flight experiment is provided, beginning with background on the Starling mission and autoNGC. Next, the navigation software that is central to autoNGC's real-time state estimation capability is discussed along with the test campaign conducted prior to the flight experiment. The next major section details the onboard navigation experiment, first detailing the configuration of the filter, then the flight results. Successful onboard navigation results paved the way for the demonstration of onboard state and uncertainty prediction, which is discussed in the following section. This section discusses the prediction approach, particularly the choice of a technique for covariance propagation, before presenting flight results. Following this, the onboard maneuver planning section describes the construction of the targeting algorithm and the timeline for the flight test before offering a subset of flight results. Finally, some concluding remarks along are provided.

FLIGHT EXPERIMENT OVERVIEW

Starling Mission Background

The NASA Starling mission launched on July 17, 2023. The mission is a swarm of four 6U CubeSats flying in a circular low-Earth orbit with an approximate altitude of 550 km. Funded as a technology demonstration mission, the Starling spacecraft have served as a testbed for exhibiting several different cutting-edge technologies, particularly those related to distributed systems missions.¹⁸⁻²² The Starling mission successfully completed its primary mission objectives in 2024 but continues to serve as a platform for demonstrating novel spaceflight technologies. The spacecraft's utility as a testbed is enhanced by their use of the NASA core Flight System (cFS) to provide the flight software architecture.²³ The widespread use of cFS across NASA and the broader aerospace community, along with its modular architecture, has enabled new software packages, including autoNGC, to be uploaded onto the Starling spacecraft despite the spacecraft already being in orbit.

autoNGC Flight Software

autoNGC is a suite of cFS-based flight software applications and hardware configurations that provide autonomous GN&C capabilities. Although Starling is the first time the autoNGC flight software has been demonstrated on orbit, many of the individual components of autoNGC already have flight heritage. For example, autoNGC uses the Goddard Enhanced Onboard Navigation System (GEONS) flight software for estimation, which has many years of flight heritage.²⁴

Moreover, the library of optical navigation (OpNav) algorithms used by autoNGC derive from the Goddard Image Analysis and Navigation Tool (GIANT), a software package that was used for shadow operations during the OSIRIS-REx mission. These tested components, alongside newly developed applications for prediction, maneuver planning, and more equip autoNGC with the capabilities required to address a variety of spaceflight scenarios.

The modular nature of the cFS architecture allows autoNGC to be easily configured to meet the needs of a range of missions. For example, the Starling flight test utilized GPS navigation, but did not include OpNav, therefore apps that supported OpNav were excluded from the software build to reduce autoNGC's storage and memory footprint. Following the Starling flight test, in early 2026, a separate build of autoNGC, that included OpNav apps, was tested in cislunar space onboard the Cislunar Autonomous Positioning System Technology Operations and Navigation Experiment (CAPSTONE) spacecraft. Finally, while autoNGC flight tests have thus far focused on demonstrating software, autoNGC can be provided in a number of hardware configurations including a 1U box that includes the necessary processor, power, and input/output cards.

Overview of GEONS and GGMS

GEONS is a spacecraft navigation flight software library used by autoNGC. GEONS is a multi-mission library developed at NASA GSFC that is continuously updated to meet the needs of new missions. The core GEONS algorithms have over 20 years of flight heritage. Previous versions of GEONS successfully provided onboard real-time navigation for the following NASA missions: Magnetospheric Multiscale (MMS),²⁵ Global Precipitation Measurement (GPM), and Neutron Star Interior Composition Explorer (NICER). GEONS has been baselined to provide onboard navigation for additional space missions,²⁶ with and without autoNGC. The Starling mission is the first time GEONS has flown as a component of autoNGC. GEONS version 3.0 was used for the flight results presented in this paper.

GEONS is a UD-factorized extended Kalman filter (EKF) written in ANSI standard C. It includes high-fidelity models of spacecraft dynamics, navigation measurements, and coordinate and time transformations. GEONS can process a variety of spacecraft navigation measurement types to estimate the position and velocity of one or more spacecraft, onboard clock error states, force model parameters, and measurement biases. There are Python and MATLAB application programming interfaces (APIs) for GEONS to facilitate testing, simulation, and analysis on the ground with the GEONS flight software in the loop.

The GEONS Ground MATLAB Simulation (GGMS) is a collection of MATLAB files and data files that leverage the GEONS MATLAB API as illustrated in Figure 1. A GGMS simulation includes an instance of GEONS that is configured the same way it would be in flight. This flight instance of GEONS is fed either simulated or real measurement data and is commanded to process that data to provide estimates of the specified state vector. When simulated measurement data is used, a separate instance of GEONS is run to generate the measurement data. This truth instance of GEONS is typically configured to use higher fidelity dynamics modeling than the flight instance, and it includes realistic measurement error sources. When using GGMS for Starling, real measurement data, recorded on orbit, was fed to GEONS through GGMS as illustrated in Figure 1. This enabled rapid tuning of the filter on the ground to verify robust and accurate navigation performance before running GEONS onboard. Then the GEONS configuration in GGMS was saved and used to configure the onboard instance of GEONS. The GGMS results were used to verify the flight navigation performance was as expected.

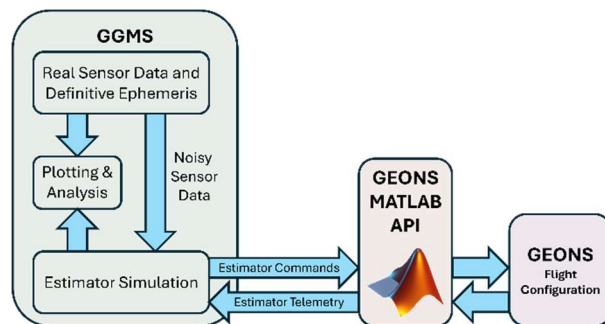


Figure 1 - Block diagram of how GGMS feeds real measurement data into a flight configuration of GEONS, using the MATLAB API, to characterize navigation performance on the ground. Image adapted from Ashman 2024 with permission.²⁷

Pre-Flight Testing

The autoNGC software was validated in a progression of increasingly realistic test environments before being uploaded and run on the Starling spacecraft. This progression, beginning with GGMS, is illustrated in Figure 2. GGMS simulations are run much faster than real-time, which allows rapid tuning of the GEONS filter configuration while processing pre-recorded flight GPS data. The GEONS configuration derived from GGMS testing was used for all the test phases in Figure 2. The GGMS results serve as a baseline to validate the more complex test environments. Slowly introducing complexity with each subsequent test phase made debugging more efficient since the root cause of any

encountered issues is isolated to the differences between the current and previous test phases.

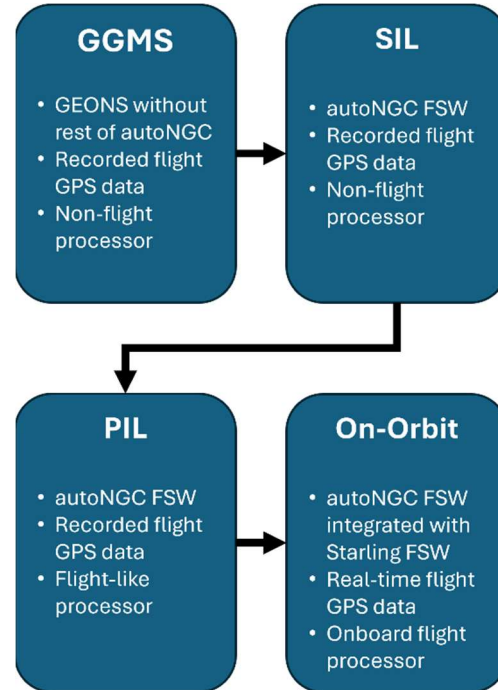


Figure 2 - Progression of the autoNGC test campaign leading up to on-orbit testing.

Software-in-the-loop (SIL) testing was the first test phase that included the full autoNGC flight software (FSW) build. The autoNGC SIL test setup utilizes a collection of Python code to provide the C/C++ autoNGC flight software with inputs, e.g., measurement data, and collect the expected outputs. In a SIL test, the autoNGC FSW is run locally on a laptop computer. The SIL testing of the Starling version of the autoNGC FSW used the same GPS data recorded on-orbit by the Starling team that was used with GGMS. Results achieved with GGMS were compared against those produced by SIL testing and the sources of discrepancies between these results were tracked down to resolve inconsistencies in the FSW. SIL testing enabled FSW issues to be identified and resolved before introducing the added complexity that comes with receiving measurements from actual hardware or running the software on a flight-like processor.

The next steps in the pre-flight test campaign of autoNGC involved testing the FSW using flight-like hardware. A processor in-the-loop (PIL) test setup was constructed in the autoNGC laboratory using ZedBoards, which use the AMD Zynq-7000 system on a chip (SoC). This is the same SoC as the processors used on the Starling spacecraft. Running the autoNGC FSW on the ZedBoards revealed new issues related to compiling the FSW for a different operating system and provided

insight into the expected runtimes of different autoNGC processes like onboard estimation or maneuver planning. The autoNGC PIL testing used either Starling recorded flight GPS measurements or simulated measurements from a SPIRENT GPS Constellation Simulator. When using recorded flight data, the results were compared to the GGMS, SIL, and PIL results to ensure the FSW was performing correctly. Testing with SPIRENT data enabled tests to be conducted for time periods where recorded flight data was not available.

The final test before flight was running the autoNGC FSW integrated with the rest of the Starling FSW on the Starling FlatSat located at NASA Ames. Pre-recorded flight GPS data was again used for this testing. The FlatSat testing verified that autoNGC could operate successfully with no negative impacts to other components of the software and hardware of the Starling spacecraft. Several updates were made to the autoNGC FSW over the course of the Starling flight test campaign, and each software update followed the same test progression outlined in

Figure 2. When FlatSat testing was first completed, the initial version of autoNGC was uploaded to the two spacecraft in the Starling formation available for testing at the time, SV2 and SV4, and on-orbit testing began.

ONBOARD NAVIGATION

Navigation Filter Configuration and Approach

The spacecraft designated SV2 and SV4, within the swarm of four Starling satellites are configured to run separate instances of autoNGC onboard, both of which include GEONS. The autoNGC app `ngc_geons` sends sensor data and commands to GEONS and publishes the GEONS state estimates and associated error covariances for the rest of autoNGC to use. The instance of GEONS onboard each satellite estimates the state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{b} \\ \Delta C_D \\ \Delta C_R \end{bmatrix} \quad (1)$$

of that satellite. Here, $\mathbf{r}$ and $\mathbf{v}$ are the Earth-Centered Inertial (ECI) position and velocity vectors respectively of the satellite. The state vector includes the bias, $\mathbf{b}$, of the spacecraft onboard clock multiplied by the speed of light such that $\mathbf{b}$ has units of meters. An additive correction, ΔC_D , to the modeled satellite drag coefficient is also estimated since drag is significant in LEO, and the drag coefficient is difficult to predict accurately. An additive correction, ΔC_R , to the nominal solar radiation pressure (SRP) coefficient was also estimated.

The spacecraft position and velocity are propagated using fourth-order Runge-Kutta numerical integration. The modeled forces are spherical harmonic gravity of the Earth through degree and order 10x10, atmospheric drag, solar radiation pressure, and point mass gravity of the Sun and Moon. The clock bias, drag coefficient correction, and SRP coefficient correction are each modeled as a Gaussian random walk. The nominal values of the drag and SRP coefficients are 2.37 and 1.37 respectively. The modeled drag uses an approximation of the Harris-Priester atmospheric density model that accounts for the diurnal bulge.

To estimate the state vector, GEONS only processes GPS pseudorange measurements for Starling, since these measurements alone provide sufficient navigation accuracy. The measurement noise standard deviation of the pseudorange measurements was modeled as 120 m. Higher navigation accuracy could be achieved by processing additional measurement types already supported by GEONS such as GPS time-differenced carrier phase measurements,²⁸ optical measurements, or ground station measurements. The filter was reinitialized before each test period onboard a satellite. None of the roughly nine-hour test periods included maneuvers. However, GEONS can provide accurate navigation through maneuvers if it has either knowledge of the planned maneuver or measurements of the maneuver acceleration from an accelerometer. The number of GPS satellites tracked over time is shown in Figure 3. GPS pseudorange measurements are rejected by the filter if the innovations (i.e., pre-fit residuals) are larger than six times their standard deviation computed by the filter. This protects the filter from ingesting bad measurements, providing robustness to GPS jamming and spoofing. GPS pseudorange measurements are also rejected if the signal path from GPS satellite to estimated receiver position passes too close to the Earth. This measurement rejection test is called a height of ray path (HORP) test, and it is the primary reason for the differences between the blue and red lines in Figure 3. The HORP test prevents the filter from ingesting measurements that are very noisy due to a long path through the atmosphere.

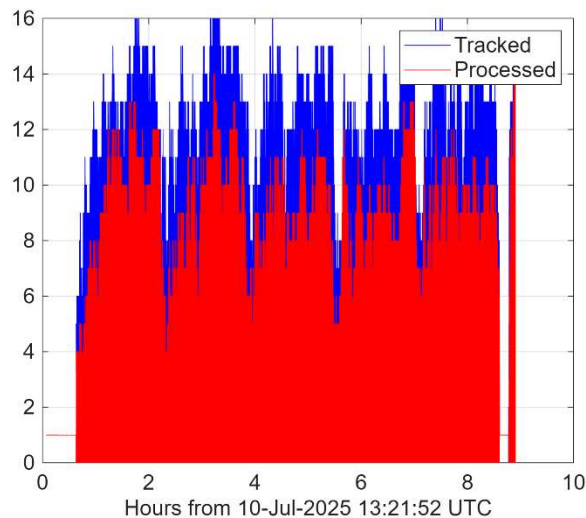


Figure 3 – Number of GPS satellites tracked by the Novatel GPS receiver on board Starling SV2 (blue) and the number of GPS satellites from which pseudorange measurements are accepted and processed in the filter (red).

It was challenging to tune the filter because the GPS receiver was configured to do clock steering. In other words, the GPS receiver altered the clock frequency over time, attempting to minimize the onboard clock offset. From the perspective of the GEONS filter, it appeared that the onboard clock was very noisy because the filter did not have any knowledge of how the onboard clock frequency was being steered. To account for these unknown clock dynamics, the clock bias process noise covariance had to be modeled relatively large for the filter to be consistent. The modeled clock bias process noise covariance was increased until the state estimation errors (as compared to the definitive ephemeris) and the measurement innovations were consistent with their theoretical uncertainties provided by the filter. The onboard clock drift is not included in the estimated state vector because it does not improve navigation accuracy. The reason is likely that the clock steering makes it difficult to accurately estimate the drift. It is expected that navigation performance would improve significantly if clock steering were disabled, clock drift was added to the state vector, and the filter were retuned.

Flight Results

Figure 4 through Figure 8 show the onboard autoNGC navigation performance for the Starling 2 satellite. These plots are representative of navigation performance for all the Starling satellites throughout the flight experiment.

Figure 4 is a plot of the measurement innovations from the GEONS EKF. The mean of the residuals is near zero, which is expected for an optimal filter.

Figure 5 and Figure 6 show the approximate estimation errors in spacecraft position and velocity with the associated theoretical $3\text{-}\sigma$ bounds, that is, the formal uncertainty from the filter. The estimation errors were computed by differencing the onboard state estimates provided by autoNGC and the definitive ephemeris interpolated to the state estimate times. The definitive ephemeris was constructed by the Starling flight operations team using the Orbit Determination Tool Kit (ODTK). The $3\text{-}\sigma$ bounds were computed by taking the square root of the diagonal of the filter computed error covariance and multiplying that by three. At steady state, the position and velocity $1\text{-}\sigma$ uncertainties are about 8-18 m and 8-20 mm/s respectively in each axis. The state estimation errors are always within their $3\text{-}\sigma$ bounds, indicating a robust, conservatively tuned filter. As stated earlier, navigation performance could be improved significantly by turning off the GPS receiver clock steering and retuning the filter.

Figure 7 shows the estimated onboard clock bias and the associated $3\text{-}\sigma$ uncertainty provided by the filter. Since the true clock bias is not known, it is not possible to quantify the clock bias estimation error. At steady state, the 1σ uncertainty in the clock bias is about 8.5 m, which is equivalent to approximately 28 ns. The estimate of the spacecraft drag coefficient correction is shown in Figure 8.

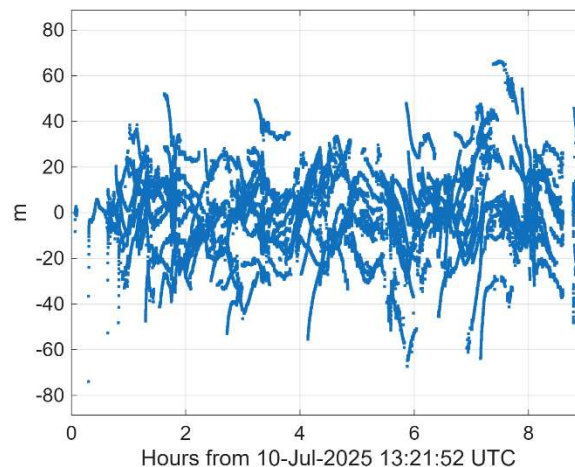


Figure 4 – GEONS EKF measurement innovations.

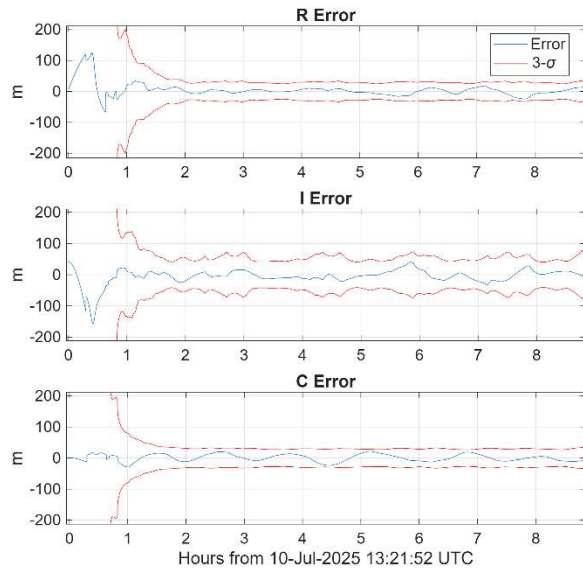


Figure 5 - Approximate position estimation errors and the associated 3- σ bounds.

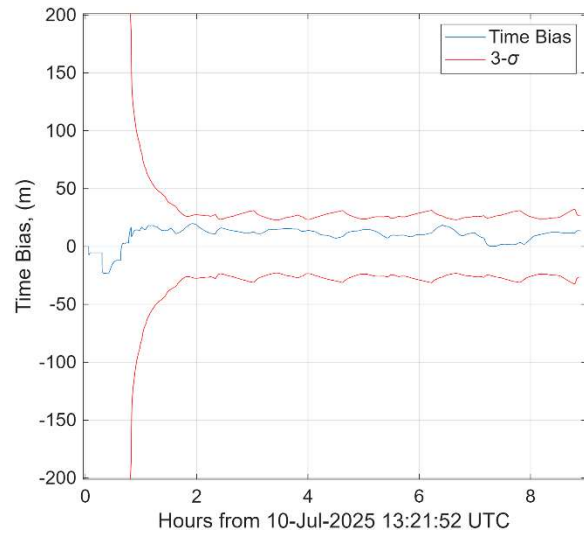


Figure 7 - Onboard clock bias state estimation errors (blue) and the associated theoretical 3- σ bounds (red).

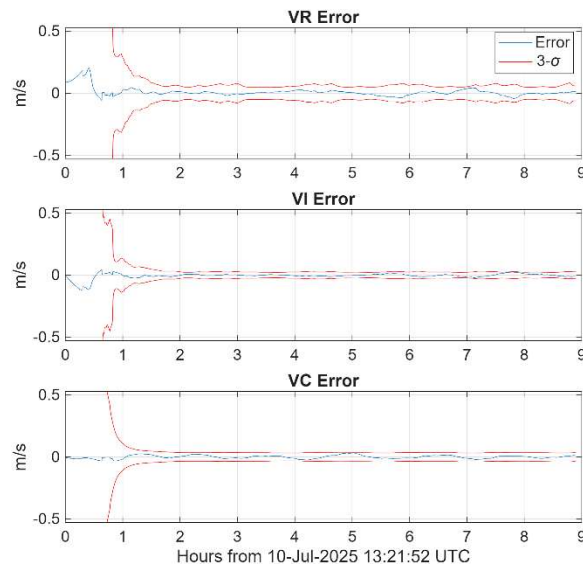


Figure 6 - Approximate velocity estimation errors and the associated 3- σ bounds.

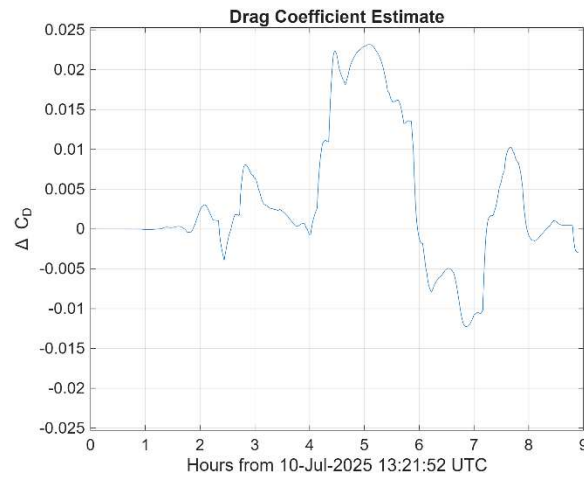


Figure 8 - Estimated drag coefficient additive correction.

ONBOARD STATE AND UNCERTAINTY PREDICTION

Prediction Approach and Rationale

The Starling SV2 and SV4 satellites ran the autoNGC state prediction cFS app, `ngc_predict`, which uses the unscented transform to propagate a state estimate and associated error covariance. The unscented transform is described in Stacey 2022.²⁹ Typically, the `ngc_predict` predicted state is for the spacecraft running that instance of `ngc_predict`, and the initial state and associated error covariance are provided by GEONS. However, `ngc_predict` provides a generic prediction capability that can propagate any provided state and error covariance. The propagated states and associated error covariances of multiple spacecraft can be used for assessing collision probability among them and informing collision mitigation actions. `ngc_predict` uses the same dynamics models as the GEONS filter to propagate the sigma points of the unscented transform. The unscented transform was used because it is very accurate over the time span required by Starling (3 days) while the linear covariance propagation employed in GEONS 3.0 provides poor accuracy over this time range. In general, the unscented transform more accurately propagates the mean state estimate and error covariance than the linearized propagation employed in an EKF because the unscented transform better captures nonlinearities.²⁹ However, the unscented transform is more computationally expensive than a linearized propagation.

GEONS propagates the mean state estimate using the nonlinear dynamics models and does a linear propagation of the state error covariance through

$$\mathbf{P}_{k|k-1} = \Phi_k \mathbf{P}_{k-1|k-1} \Phi_k^T + \mathbf{Q}_k \quad (2)$$

Here, $\mathbf{P}_{k|k-1}$ is the state error covariance at time step k given all the measurements through time step $k-1$, Φ_k is the state transition matrix, and $\mathbf{Q}_k$ is the process noise covariance. The state transition matrix is computed through a semi-analytic numerical integration approximation based on method H in Section 3.2.2 of Carpenter 2025.³⁰ However, the GEONS approximation is slightly different and less accurate than method H in Carpenter 2025.³⁰ For any numerical integration approximation of the state transition matrix, the propagated error covariance becomes less accurate as the numerical integration step size increases and as the length of the propagation interval increases. The computation of the state transition matrix in GEONS includes two-body and J_2 gravity of the central body as well as point-mass accelerations of all considered perturbing celestial bodies. This relatively low-fidelity

force model also introduces errors in the propagated error covariance. Improving the fidelity of the GEONS state transition matrix, both the numerical integration approximation and the considered forces, is planned future work.

Covariance Propagation Comparison

Figure 9 and Figure 10 compare multiple covariance propagation techniques using GGMS. All orbit propagations were completed using GEONS called from GGMS. The orbit propagations used the same dynamics models as the flight configuration of GEONS used for real-time navigation described earlier. The only exception is the central finite difference (CFD) approach discussed below. The cyan line is the propagated error standard deviation provided by the unscented transform multiplied by three. The unscented transform was computed through a MATLAB implementation of the unscented transform in `ngc_predict` using the same tuning parameters. The unscented transform very closely matches the Monte Carlo covariance propagation shown in red, which serves as the ground truth. The Monte Carlo covariance propagation used 200 samples, and the magnitude of the deviations of the individual Monte Carlo runs from the mean are shown in gray. That is why the gray lines are less than the red line about 99% of the time. The GEONS covariance propagation with an integration time step of one second is shown in blue. Decreasing the step size further did not improve accuracy. As compared to the Monte Carlo results, the GEONS covariance propagation is very accurate for about 7.5 hours (i.e., about 5 orbit periods) and then degrades significantly for longer time intervals.

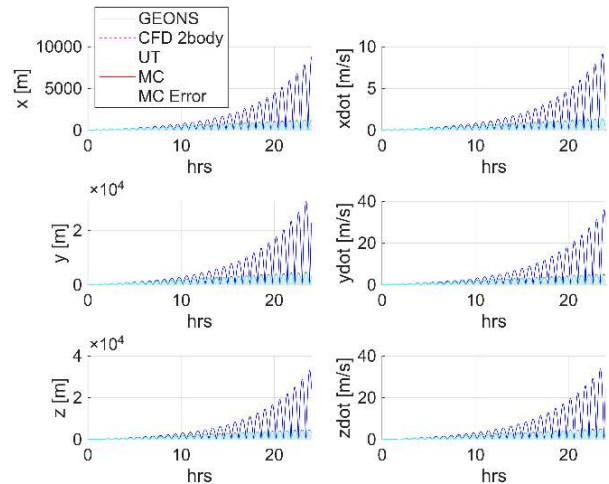


Figure 9 - Comparison of 3- σ uncertainties generated with multiple covariance propagation techniques for a single, 24-hour, representative Starling orbit scenario.

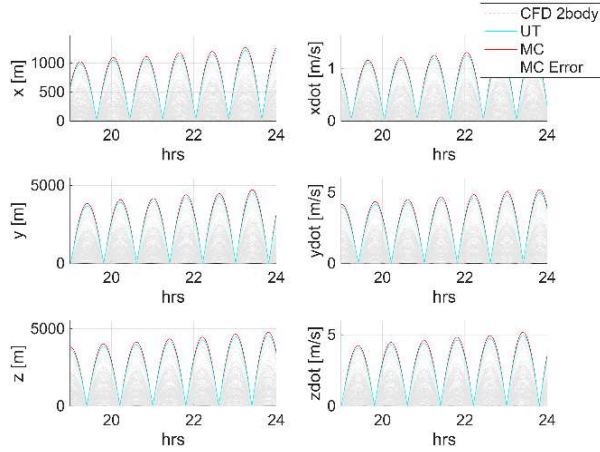


Figure 10 – Comparison of $3\text{-}\sigma$ uncertainties generated with multiple covariance propagation techniques for a single, 24-hour, representative Starling orbit scenario. Zoomed in view of final five hours of the propagation.

To determine whether the inaccuracy of the GEONS covariance propagation is due to the linearized propagation approach (i.e., Eq. (1)) or the accuracy of the state transition matrix, the covariance was also propagated using Eq. (1) but where the state transition matrix was computed through a CFD approximation. The CFD linear covariance propagation matches the unscented transform and Monte Carlo covariances, which indicates that a linear covariance propagation, as shown in Eq. (1), can provide excellent accuracy for the considered scenario. Thus, the reason the GEONS covariance propagation is poor over long time intervals is that the state transition matrix is not accurate enough. The CFD orbit propagations only model two-body gravity of the Earth, which means that the GEONS state transition matrix errors are driven by the numerical integration approximation employed and not by the considered force models. Updating GEONS to use a more accurate state transition matrix approximation (e.g., one of the methods in Section 3.2.2 of Carpenter 2025)³⁰ would enable accurate linear covariance propagation over the long time intervals shown in Figure 9 for a much lower computational cost than the unscented transform.

Prediction Flight Results

In total the `ngc_predict` app was run four times onboard the Starling SV2 and SV4 spacecraft. The results of the test run on July 10th, 2025, the same test period as the onboard navigation results shown in the previous section, are presented here. The data obtained during this test period is representative of the performance of `ngc_predict` during other flight tests. The `ngc_predict` app was not started until onboard navigation had been running for at least 4 hours. After this time, a relative

time sequence was used to send commands to the `ngc_predict` app that fetched the current estimated state from `ngc_geons` and began a prediction.

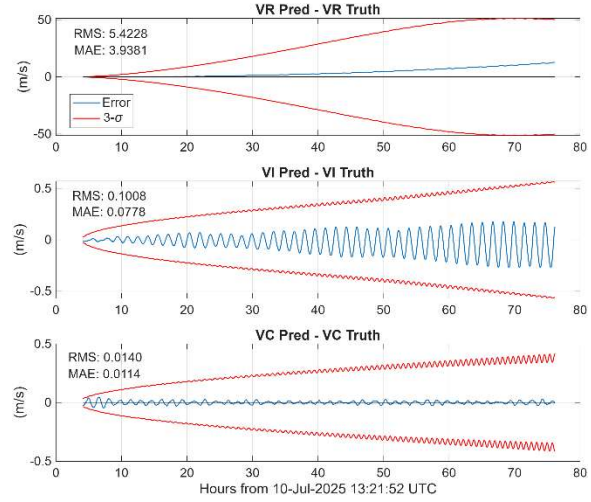


Figure 11 - The error of a propagated position state estimate and associated error covariance provided by `ngc_predict`.

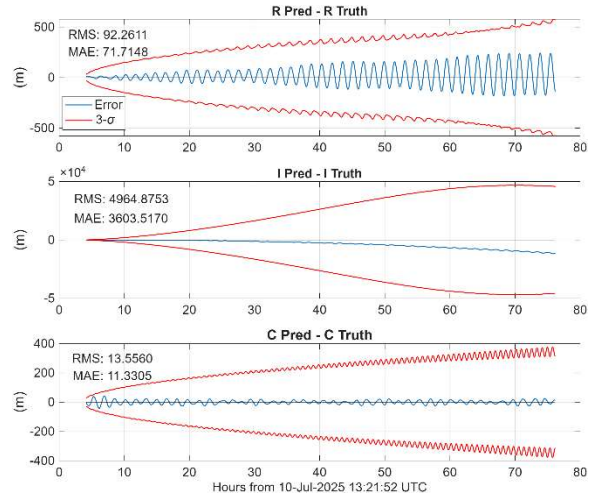


Figure 12 - The error of a propagated velocity state estimate and associated error covariance provided by `ngc_predict`.

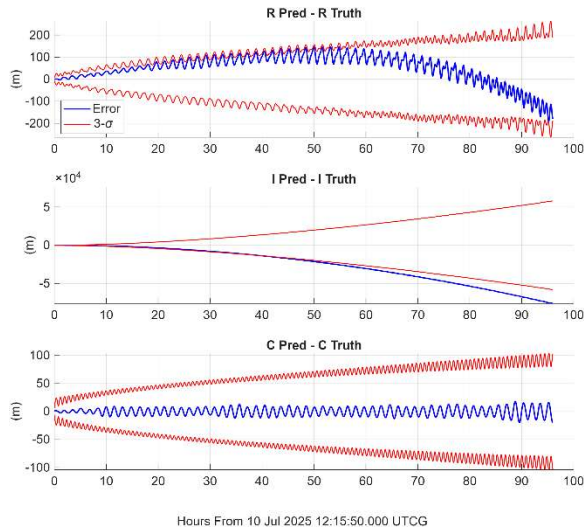


Figure 13 – The error of a propagated RIC position state estimate and associated error covariance generated using NASA Ames ground software.

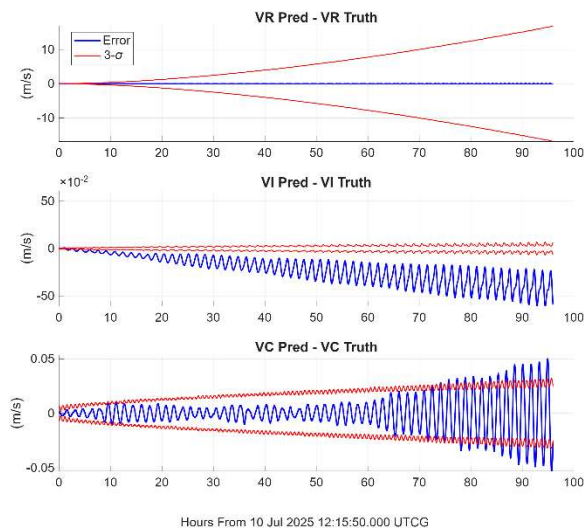


Figure 14 - The error of a propagated RIC velocity state estimate and associated error covariance generated using NASAS Ames ground software.

Figure 11 and Figure 12 show flight results where an onboard instance of `ngc_predict` propagated the state estimate and error covariance from an onboard instance of GEONS after the filter had converged. The GEONS filter was processing real, on orbit GPS measurement data captured by a Starling satellite. The state prediction errors shown in the figure are the difference between the predicted state and the definitive Starling ephemeris. The associated error $3\text{-}\sigma$ bounds are taken from the `ngc_predict` propagated error covariance. The state prediction errors are conservatively encapsulated by the $3\text{-}\sigma$ uncertainty bounds. The process noise covariance used in the covariance prediction could be tuned to make

the predicted error covariance more consistent with the state prediction errors. When running onboard, `ngc_predict` took about 17 min to complete a 72-hour prediction of the spacecraft state and associated error covariance. Future work will improve the GEONS linear covariance propagation so that it can be used in `ngc_predict`, which is expected to provide similar accuracy for a much lower computational cost. For example, Table 5.1 in Stacey 2022 shows that the computation time of the time update of an EKF is an order of magnitude less than that of the unscented transform for a considered scenario.²⁹

Independent from autoNGC, the Starling flight operations team also performs orbit prediction, but on the ground. STK and the associated ODTK, are used to propagate the spacecraft state and the associated error covariance. ODTK orbit prediction results are shown in Figure 13 and Figure 14 for a similar timespan as the `ngc_predict` results in Figure 11. The ODTK prediction error is computed by differencing the ODTK predicted state with the same definitive ephemeris as the `ngc_predict` results shown in Figure 11. Due to differences in the Starling team's process the predictions shown in Figure 13 and Figure 14 use a slightly earlier starting epoch, July 10th 12:15:50 UTC, and are propagated for a longer duration.

The curve evident in the plots for the R, I, and VI state components is explained by a slight under-prediction of drag for the first 48 hours, and then a higher drag predict for the next 48 hours due to off nominal pointing of SV2 to support another experiment which did not impart as much drag as predicted. This difference between predicted and actual drag force is also why the prediction error grows outside the uncertainty bounds for several of the plots in Figure 13 and Figure 14. The fact that the `ngc_predict` plots do not exhibit a similar trend in the same states indicates that the drag coefficient employed for the propagation was more accurate, perhaps because a correction to the drag coefficient was being estimated onboard.

The plots illustrate that the error of the ground prediction after about 72 hours is a similar order of magnitude as the flight prediction, but the associated error $3\text{-}\sigma$ bounds of the ground prediction are more consistent with the state prediction error, despite the fact that the error surpasses the $3\text{-}\sigma$ bounds in some cases. As stated previously, tuning the process noise covariance would make the `ngc_predict` covariance prediction more consistent with the state prediction and the results generated with ODTK. Nonetheless, the similarity of the state prediction error in the predictions generated on the ground and in flight indicates that `ngc_predict` provides accurate onboard prediction capabilities that could be

useful for collision risk assessment and onboard planning activities.

ONBOARD MANEUVER PLANNING

Maneuver Planning Approach

The goal of the autoNGC maneuver planner test was to demonstrate the ability of autoNGC to compute, onboard the Starling spacecraft, in-plane station keeping maneuvers that successfully maintain the desired formation of the Starling swarm. Demonstrating onboard maneuver planning shows that the autoNGC software enables a spacecraft to not only estimate where it is and where it will be, but to act on that information by planning and executing maneuvers.

The Starling team has successfully maintained various Starling swarm formations for nearly three years. When the autoNGC flight test began in the summer of 2025, SV3 had already suffered a radio failure and was unable to act on commands sent from the ground. Moreover, SV1 and SV2 no longer had functional propulsion systems, so their maneuvering was conducted by rotating the spacecraft into a high-drag orientation for extended periods of time. The higher drag force experienced during this orientation reduced the semi-major axis (SMA) of SV2 more rapidly. Despite this issue, flight tests of the autoNGC maneuver planner were run on SV2 and SV4, because these were the spacecraft that were running autoNGC to test onboard navigation and orbit prediction.

The primary metric used by the Starling team to plan station-keeping maneuvers is inter-satellite distance (ISD), that is, the straight-line distance between two spacecraft in the Starling formation. For SV1, SV2, and SV4, station keeping maneuvers were planned approximately once a week to keep the ISD of each spacecraft with respect to the others within a range of 75km to 250km. Maneuvers were planned such that the ISD trend over time would have a positive or negative slope that was gradual enough to achieve the desired ISD

within 7 days, thus preventing the need for another station-keeping maneuver during that time. As a result of this approach, the ISD trends of the Starling swarm “bounced” back and forth between the desired maximum and minimum bounds. Due to maneuver execution errors, the achieved ISD trend never exactly matched the trend predicted using the Systems Tool Kit (STK) software. The size of the maneuver execution errors was $\pm 22\%$ (1σ), due in part to the small size of the executed maneuvers and the difficulty in controlling plenum pressure in the propulsion system.³¹ Moreover, the ability to predict the Δv imparted by SV2 high-drag maneuvers was hindered by the uncertainty inherent in predicting atmospheric conditions that impact the magnitude of the drag force. Given these sources of error, the achieved ISD trend was expected to vary from the predicted trend. However, if the slope of the resulting ISD trend did not exceed ± 50 km/week the outcome was deemed acceptable. Note that the ISD with respect to SV3 was not factored into maneuver planning for the autoNGC flight test. Because this spacecraft could not be commanded, its SMA is decaying faster than the other Starling spacecraft and is not actively maintained.

The desired changes in ISD trends are achieved by executing maneuvers in the velocity or anti-velocity direction that respectively raise or lower the SMA of the orbit for a given spacecraft. Therefore, the targeting algorithm implemented in autoNGC was designed to target SMA and eccentricity values at a desired target time by allowing anywhere from 1 to N maneuvers. Typically, two maneuvers, $N = 2$, were used, as shown in Figure 15 and the maximum value of N permitted was $N = 5$. The targeting algorithm was posed as a set of design variables, $\mathbf{X}$, that are varied to satisfy a set of constraints, $\mathbf{F}$, in this case, a subset of the classic Keplerian orbital elements.

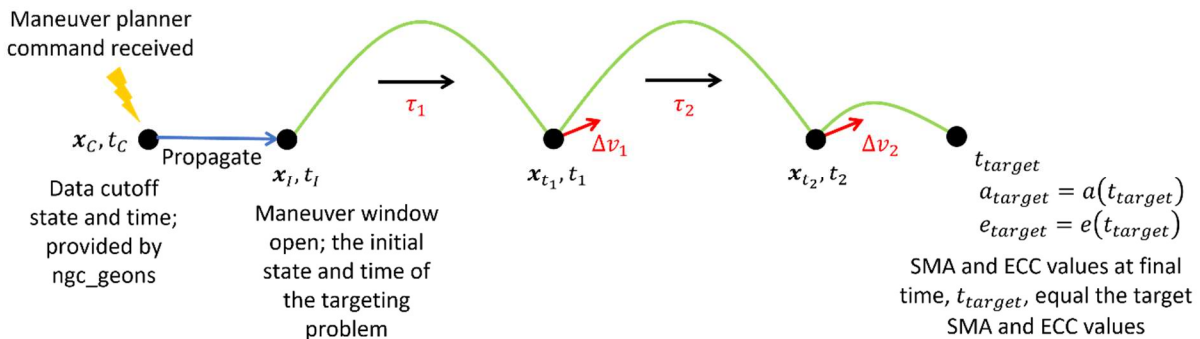


Figure 15 - Diagram of the Starling station-keeping targeting problem

$$\mathbf{X} = \begin{bmatrix} \Delta v_1 \\ \tau_1 \\ \vdots \\ \Delta v_N \\ \tau_N \\ \sigma_{\tau_1} \\ \sigma_{cd_1} \\ \vdots \\ \sigma_{cd_{N-1}} \\ \sigma_{\tau_N} \\ \sigma_{\Delta v_1} \\ \vdots \\ \sigma_{\Delta v_N} \end{bmatrix} \quad (3)$$

$$\mathbf{F} = \begin{bmatrix} a_{target} - a(t_{target}) \\ e_{target} - e(t_{target}) \\ \tau_1 - \sigma_{\tau_1}^2 \\ \tau_1 - (\tau_2 - \tau_{cd}) + \sigma_{cd_1}^2 \\ \vdots \\ \tau_{N-1} - (\tau_N - \tau_{cd}) + \sigma_{cd_{N-1}}^2 \\ \tau_N - t_{target} + \sigma_{\tau_N}^2 \\ \Delta v_1 - \Delta v_{max} + \sigma_{\Delta v_1}^2 \\ \vdots \\ \Delta v_N - \Delta v_{max} + \sigma_{\Delta v_N}^2 \end{bmatrix} = \vec{0} \quad (4)$$

The design variable vector $\mathbf{X}$, contains the Δv magnitudes for maneuvers $1, \dots, N$. The Δv is implicitly constrained to be applied in only the velocity or anti-velocity direction at the time of execution. Thus, only the Δv magnitude is varied in the targeting algorithm. The duration in seconds after the start of the targeting problem, $\tau_1, \dots, \tau_N$, is also permitted to vary. Finally, the design variable vector contains multiple slack variables: $\sigma_{\tau_1}, \sigma_{cd_1}, \dots, \sigma_{\Delta v_N}$ that transform inequality constraints into equality constraints that are included in the constraint vector $\mathbf{F}$. The first two components of the constraint vector constrain the semi-major-axis and eccentricity values at the target time, t_{target} to match the desired values, a_{target} and e_{target} , respectively. Since the Δv always occurs along the velocity vector, the other orbital elements will be maintained without explicit constraints. Next, the time of the first maneuver, τ_1 , is constrained to be greater than 0, to prevent maneuvers from being placed before the maneuver window opens. Following this, constraints are included to enforce a cooldown duration, τ_{cd} , which ensures that maneuvers are separated by at least a minimum amount of time. The last constraint on time constrains the time of the final maneuver τ_N to be less than or equal to the target time,

t_{target} . Finally, a series of constraints are included to ensure that no Δv exceeds a set maximum value, Δv_{max} .

A generalized Newton method is used to determine a set of design variables, $\mathbf{X}$, that satisfy the targeting problem constraints, $\mathbf{F}$, to within a specified tolerance. For $N > 1$ the targeting problem is not square, so a minimum-norm update is used to calculate each update to $\mathbf{X}$. The C++ library *Eigen* provides the linear algebra functionality needed to implement the targeting algorithm in flight software. The partial derivatives of the constraints with respect to the design variables needed for the update step are computed via a combination of forward finite differencing and analytical expressions.

Earlier implementations of the targeting algorithm developed for the Starling implementation of the autoNGC maneuver planner provided a more general targeting capability. A user could provide a six-state of three Cartesian position and velocity components as a target state, rather than targeting only SMA and eccentricity. Moreover, each Δv was a three-element vector in Cartesian coordinates that contained pointing and magnitude. However, the more general approach introduced unnecessary complexity for Starling. Furthermore, targeting a full six-state and time, obtained from a predictive ephemeris, led to large Δv maneuvers because it did not allow the flexibility necessary to accommodate errors in state estimation and prediction. The station-keeping maneuvers conducted by the Starling team were always along the velocity vector with the goal of either raising or lowering the SMA of the orbit of a given spacecraft. Therefore, the targeting algorithm used within the autoNGC maneuver planner was modified to replicate this approach. A lesson learned from this development is to pursue the simplest possible implementation of an algorithm necessary to meet mission requirements.

Since the principal metric for maintaining the Starling formation is ISD, it is preferable to design a targeting algorithm that explicitly constrains this value. However, using ISD in the targeting algorithm requires up to date knowledge of the trajectories of the other spacecraft, and the autoNGC experiment was not configured to utilize crosslink communication. Future work could include running autoNGC on multiple spacecraft and enabling spacecraft to share state estimates with one another via cross-link.

The autoNGC maneuver planner was always tested in “shadow mode”, that is the software computed a maneuver onboard, but it was not executed. However, these tests were structured to follow a timeline similar to what would be used if the maneuvers calculated by

autoNGC were going to be executed, this timeline is displayed in Figure 16.

The timeline is driven by the fact that contact periods with the Starling spacecraft occur roughly every 12 hours and that maneuver plans must be screened by the NASA Conjunction Assessment Risk Analysis (CARA) office prior to execution. The first ground contact period of the flight test, GC1, is used to uplink commands to *ngc_geons* that start the onboard estimation process. By the time the next ground contact period, GC2, occurs the onboard filter will have converged upon a state estimate, and the commands to plan a maneuver are sent to *ngc_mp*. The time at which the *ngc_mp* receives the current estimated state from *ngc_geons* is referred to as the data cutoff (DCO). The following ground contact period, GC3, is used to downlink the results of the maneuver planner. The maneuver plan computed onboard is checked, using STK and other ground software, to show that the maneuver will lead to acceptable ISD trends. If the maneuver plan is acceptable then the predicted 7-day ephemeris is sent to the Conjunction Assessment Risk Analysis (CARA) office for screening. If CARA approval is received, then GC5 is used to uplink command that publish the maneuver plan to the bus and arm the spacecraft's propulsion system to execute maneuvers. During "shadow mode" flight tests, the timeline in Figure 16 is followed up through step 5b, at which point the 7-day ephemeris would be sent to CARA. Multiple shadow mode tests allowed the Starling and autoNGC teams to practice steps 1-5b, refine the process for generating maneuver planner commands, and expedite the processing of

downlinked telemetry. Examples of these flight results are discussed next.

Flight Test Preparation

So far, the autoNGC maneuver planner has been run four times onboard SV2 and SV4. Due to spacecraft availability the initial shadow mode flight tests ran on SV2. However, because this spacecraft did not execute propulsive maneuvers it was more challenging to compare the trends that were predicted to result from an impulsive maneuver planned by autoNGC with the actual trends caused by implementing a high-drag maneuver. Therefore, the results presented here are primarily from a shadow mode flight test run on SV4 on April 29th, 2026.

The schedule of a shadow mode flight test is developed by working backward from the time of the planned maneuver. Following the schedule outlined in Figure 16, the autoNGC maneuver planner was typically run 24-36 hours prior to maneuver execution. The schedule for the April 29th, 2026, shadow mode test is shown in Table 1. Due to operational constraints, the duration between DCO and maneuver execution was only 20 hours. The autoNGC navigation app was started 6 hours prior to running the maneuver planner, and the trends shown in Figure 5 and Figure 6 indicate that this is a sufficient duration for the onboard filter to achieve a converged steady state. A combination of absolute and relative time sequences (ATS and RTS), organized in a Payload Activity List (PAL), was used to send commands to the maneuver planner at the desired time, at which point, the current estimated state is sent to the maneuver planner. The command that contained the DCO state also

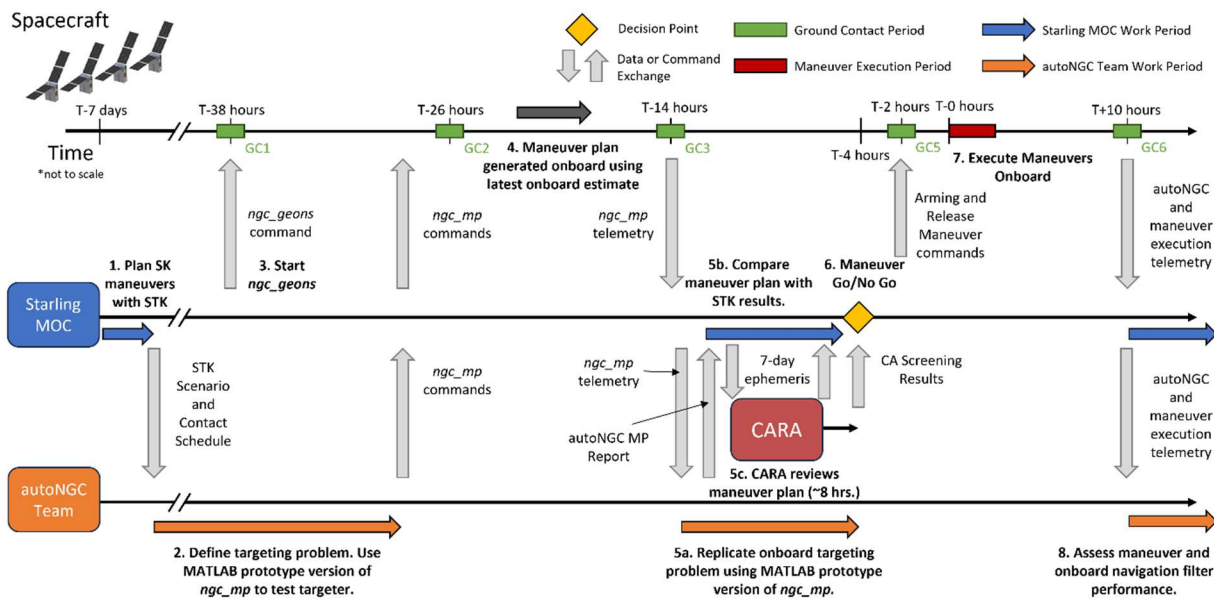


Figure 16 - Concept of operations for testing the autoNGC maneuver planner onboard Starling spacecraft.

included information about the targeting problem that will be run onboard. The onboard maneuver planner typically required less than one hour to run; however, a total of 2.5 hours is provided before the shadow mode test is concluded by shutting down autoNGC apps. Finally, 17 hours after the conclusion of the shadow mode test, the maneuver planned on the ground by the Starling team was executed by SV4.

Table 1 - Schedule for April 29th, 2026, autoNGC maneuver planner shadow mode test.

Epoch (UTC)	Event
2026 Apr 29 16:32:35	GEONS Start
2026 Apr 29 22:30:35	DCO; Maneuver planner starts
2026 Apr 30 01:00:00	End of shadow mode test
2026 Apr 30 18:43:31	Maneuver execution

Several days prior to running the shadow mode flight test, the autoNGC and Starling teams prepared the contents of the maneuver planner commands that were included in the ATS/RTS that was uplinked to the spacecraft one day prior to the shadow mode test. Formulation of these commands was conducted using predictive ephemeris and an STK scenario provided by the Starling team. The predictive ephemeris for all four spacecraft had a DCO on 4/27. The provided STK scenario included the maneuver that would be conducted by SV4 on April 30th. The trends in SMA and eccentricity exhibited by SV4 in the time period after the maneuver were compared against a version of the SV4 spacecraft, created in STK, that did not execute a maneuver. This comparison highlighted the expected change in SMA and eccentricity provided by the planned maneuver. The desired change in orbital elements at the target time, t_{target} in Equation (4), is part of the contents of the command sent to initiate the maneuver planner. The information in this command is used to construct a target state, i.e., a_{target} and e_{target} , when the maneuver planner begins onboard execution. Once a desired change in SMA and eccentricity was determined, the constructed maneuver planner commands needed to be tested prior to uplink.

A copy of the maneuver planner software running onboard SV4 was used to validate the commands that would be sent to the spacecraft. A prototype version of the autoNGC maneuver planner implemented in MATLAB, using GGMS, was used, along with the autoNGC flight software, to generate and test maneuver planner commands. The maneuvers planned by autoNGC when these commands were tested in MATLAB are shown in Table 2 alongside the maneuver planned on the ground by the Starling team. The

autoNGC maneuver planner was set up to compute two station-keeping maneuvers instead of one because this led to improved convergence of the targeting algorithm. The combined magnitude of the maneuvers computed by the autoNGC maneuver planner differs from the maneuver planned by the Starling team by about 10%. When expressed in an inertial reference frame, the Δv vectors computed by autoNGC exhibit slightly different pointing compared to the vector planned by the Starling team, and this difference is a result of the fact that the velocity vector used to define the different Δv vectors was determined using two different modeling and propagation tools, STK and GEONS.

Table 2 - Maneuver magnitude and timing results from April 29th, 2026, shadow mode flight test.

Maneuver	Starling Team Plan	autoNGC Ground Plan	autoNGC Onboard Plan
Man. 1 Epoch	2026 Apr 30 18:43:30.848	2026 Apr 30 18:43:27.000	2026 Apr 30 18:43:28.000
Man. 1 Δv	5.99 cm/s	3.2821 cm/s	3.2783 cm/s
Man. 1 Direction	Along Velocity Vec.	Along Velocity Vec.	Along Velocity Vec.
Man. 2 Epoch	---	2026 Apr 30 19:31:22.000	2026 Apr 30 19:31:21.000
Man. 2 Δv	---	2.0094 cm/s	2.0130 cm/s
Man. 2 Direction	---	Along Velocity Vec.	Along Velocity Vec.
Total Δv	5.99 cm/s	5.2915 cm/s	5.2913 cm/s

Validation of the maneuver planner commands includes checking that the maneuvers that the commands produce lead to the desired trends in the evolution of the Starling swarm. The maneuvers computed by autoNGC during the command validation step are applied to a new spacecraft in the STK scenario so that the ISD trends resulting from the two different maneuver plans can be compared. Comparison of the SV4 to SV1 and SV4 to SV2 ISD trends over one week are shown in Figure 17 and Figure 18, respectively. In both plots, small differences between the maneuvers planned for execution on April 30th result in noticeable differences in ISD trends after several days of ballistic propagation. Deciding whether a difference in these trends was small enough that the prepared maneuver planner commands could be used for a flight test involved a mix of quantitative and qualitative judgement. In Figure 17 and Figure 18 the average difference in SV4 to SV1 and SV4 to SV2 ISD trends after six days of propagation is approximately 10 km in both cases. This discrepancy was small relative to the absolute ISD between both spacecraft pairs; moreover, the difference in the slope of the ISD trends would not meaningfully change when the

next station-keeping maneuver needs to be performed. If instead the maneuver planned by autoNGC made the slope of the ISD trend line substantially more negative, then the next station-keeping maneuver would need to be performed much earlier than usual. The relatively minor differences in expected ISD trends that resulted from implementing the maneuver planned on the ground by autoNGC, shown in Table 2, provided confidence that the prepared commands could be uplinked to the spacecraft. The ATS/RTS including these commands was uplinked to the spacecraft on April 28th and the shadow mode flight test was conducted as planned.

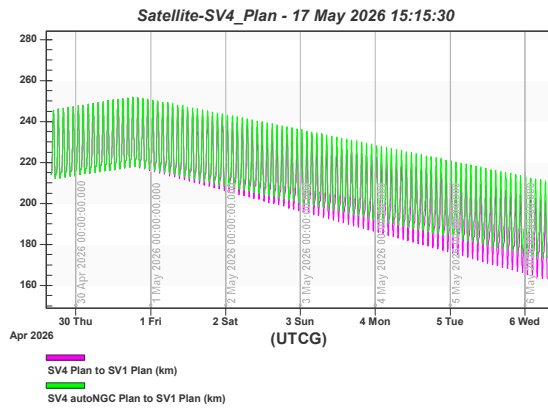


Figure 17 – Comparison of predicted SV4 to SV1 ISD trend if the maneuver planned by the Starling team is executed (magenta) to the predicted trend if the maneuver planned by autoNGC ground software is executed (green). Maneuver execution scheduled for April 30th, 2026.

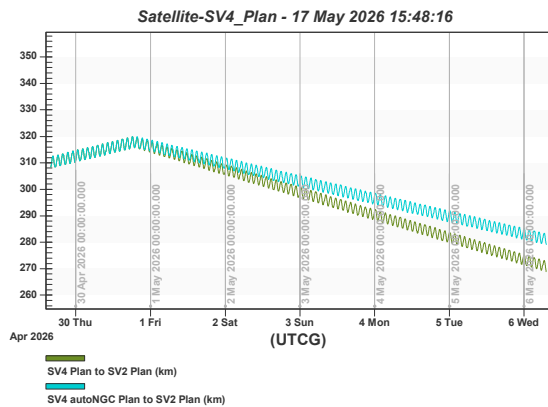


Figure 18 - Comparison of predicted SV4 to SV2 ISD trend if the maneuver planned by the Starling team is executed (green) to the predicted trend if the maneuver planned by autoNGC ground software is executed (blue). Maneuver execution scheduled for April 30th, 2026.

Flight Test Results

Processing the results of the shadow mode flight test involves examining the output of the maneuver planner using MATLAB and STK to predict the trajectory of SV4 if the maneuver planned onboard was executed. When the shadow mode test is completed, the autoNGC telemetry stored onboard the spacecraft is downlinked at the next available opportunity. This telemetry can be analyzed immediately if the planned maneuver is going to be executed onboard. However, the results shown here use a definitive ephemeris constructed one week after the shadow mode test as a reference for assessing autoNGC performance. The definitive ephemeris is used to create updated versions of the same plots shown in Figure 17 and Figure 18, and these new plots are shown in Figure 19 and Figure 20. One of the trends lines on each of these plots still shows the predicted ISD trend resulting from the maneuver planned by autoNGC during command validation prior to the flight test; however, the plots also show the predicted trends if the maneuver planned onboard was executed along with the actual ISD trend observed after SV4 executed the maneuver planned by the Starling team. In both Figure 19 and Figure 20 the ISD trend that results from implementing the maneuvers planned onboard by autoNGC are slightly steeper than the actual trend observed in the definitive ephemeris, but are overall very similar. Moreover, the trends resulting from the maneuver planned onboard are very similar to the trends predicted prior to the flight test. These similarities provided confidence that the method used to predict autoNGC performance is accurate, and that the maneuver planned onboard by autoNGC would lead to acceptable ISD trends. The reason that the maneuver planned onboard by autoNGC is different than the maneuver planned on the ground using the same command is because the DCO state and target state constructed onboard differ from what was used on the ground.

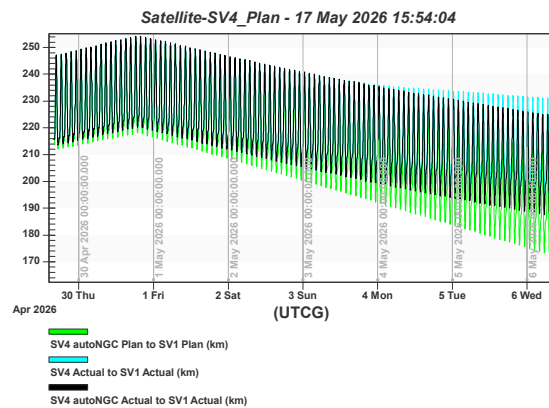


Figure 19 - Comparison of actual SV4 to SV1 ISD trend (cyan) to trend that would have resulted if the maneuver planned by autoNGC had been executed

(black). The predicted ISD trend resulting from the maneuver planned by autoNGC ground software prior to the flight test (green) is included as well. Maneuver execution occurred on April 30th, 2026.

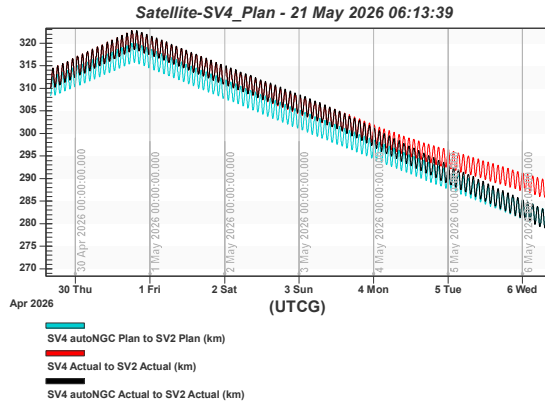


Figure 20 - Comparison of actual SV4 to SV1 ISD trend (red) to trend that would have resulted if the maneuver planned by autoNGC had been executed (black). The predicted ISD trend resulting from the maneuver planned by autoNGC ground software prior to the flight (blue) is included as well. Maneuver execution occurred on April 30th, 2026.

The performance of the onboard maneuver planner can also be assessed by examining how well the computed maneuvers deliver the spacecraft to the desired target state. Upon receiving the command to plan a maneuver, autoNGC constructed a target state onboard by propagating from the received DCO state and epoch to the provided target time and applying the desired change in SMA and eccentricity provided in the command payload. For the April 29th shadow mode test, these changes were approximately +76 km in SMA and 5.25E-06 in eccentricity. The latter change is effectively a request to keep the eccentricity constant. The target states constructed onboard are shown in Table 3. The definitive ephemeris and STK scenario are used to obtain the predicted state of SV4 at the target epoch had the maneuver planned onboard by autoNGC been executed. This predicted state and its differences relative to the desired target are also shown in Table 3. The onboard targeting algorithm was configured to satisfy the constraint tolerances to within 70 meters for SMA and 1E-8 eccentricity. This is a “loose” constraint on SMA when compared against the desired total change in this quantity. Future testing could benefit from an updated scaling approach; however, testing showed that the SMA constraint was consistently well within the tolerance. Constraint satisfaction, that is, the difference between the desired and achieved target state, is shown in Table 3. These numbers indicate that roughly 80% of the

desired change in SMA was achieved while the eccentricity was held steady, and this result was sufficient to lead to the close matching of ISD trends displayed in Figure 19 and Figure 20.

Table 3 - Comparison of Target and Achieved States for April 29th, 2026, shadow mode flight test. Target epoch is 2026 May 1 10:00:00 UTC.

Target	SMA [km]	ECC
Desired Target State	6918.399016	1.5778E-3
Predicted Achieved Target State if Onboard Maneuver Plan Was Executed	6918.383713	1.579E-3
Diff.	0.015303	-1.2E-6

While the results of a single flight test are shown here, this data is representative of the results of other recent tests. Early flight tests on SV2 used a different approach for formulating the target state provided to the autoNGC maneuver planner. The results of these tests motivated changes that were made to the target formulation process which led to the approach described in this work. As of writing, regular autoNGC maneuver planner flight testing continues, with the goal of demonstrating consistent performance in preparation for eventually executing station-keeping maneuvers planned onboard the Starling spacecraft.

CONCLUDING REMARKS

The Starling mission presented the first flight test opportunity for autoNGC. The tests conducted onboard the Starling spacecraft over the last year demonstrated three core capabilities of autoNGC: navigation, orbit prediction, and maneuver planning. Together these three functions provide a spacecraft with the tools needed to conduct more GN&C functions onboard, an essential steppingstone in the path to achieving greater spacecraft autonomy.

The autoNGC app `ngc_geons` provides an interface with the navigation flight software library GEONS, a tool with extensive flight heritage. The GEONS navigation filter onboard the Starling satellites provides position and velocity 1-sigma uncertainties of about 8-18 m and 8-20 mm/s respectively in each axis by processing only GPS pseudorange measurements. The accuracy of the filter could be significantly improved by turning off GPS receiver clock steering and retuning the filter. Clock steering introduces unknown clock dynamics, which necessitates a larger modeled process noise covariance for the clock offset, increasing the filter state estimation errors and covariance. Filter performance could also be improved by incorporating more measurement types such as GPS time-differenced carrier phase.

The autoNGC app `ngc_predict` uses the unscented transform to provide accurate propagation of the most recent mean state estimate and error covariance of the GEONS navigation filter. The GEONS linear covariance propagation was not used because the accuracy degrades significantly for propagation intervals greater than about 5 orbit periods. Simulation results indicate the GEONS linear covariance propagation could also provide good accuracy over many more orbit periods if the numerical integration approximation is replaced with a more accurate method, like one of the methods described in Section 3.2.2 of Carpenter 2025.³⁰ The ability to perform onboard state prediction is particularly important for swarm missions, like Starling, and for missions operating in the increasingly crowded Earth orbit domain where collision avoidance is a concern.

The autoNGC app, called `ngc_mp`, demonstrated the ability to use the onboard state estimate to compute maneuvers onboard that achieve desired trends in the inter-satellite distance between several spacecraft in the Starling swarm. A targeting algorithm was implemented that calculates the Δv magnitude in the velocity or anti-velocity direction that achieves a desired SMA and eccentricity at a target time. Tests were conducted on the ground, prior to command uplink, to assess whether a maneuver planner command was likely to produce a favorable result. A similar process was used to process the telemetry generated by running the maneuver planner onboard and determine whether the planned maneuver should be executed. Results from a flight test on April 29th, 2026, were presented, this test yielded favorable performance and was representative of most of the four flight tests conducted so far.

Collaboration between the Starling and autoNGC teams is set to continue with several upcoming tests. The autoNGC prediction capability will be further demonstrated by propagating the state of multiple resident space objects (RSOs) in the vicinity of the Starling spacecraft. This test represents another step towards using onboard state and covariance prediction to assess collision likelihood. Furthermore, future testing could involve the execution of a maneuver planned onboard. The process that will be used to verify and approve this maneuver was summarized in Figure 16. Enabling this test requires developing and testing interfaces between autoNGC and other components of the Starling spacecraft flight software, so that maneuver plans can be read and acted upon by the appropriate subsystems. Successful execution of a maneuver planned by autoNGC onboard one of the Starling spacecraft will further validate autoNGC capabilities and represent one more step towards enhanced spacecraft autonomy.

The autoNGC flight software suite is actively under development with the aim of enabling greater spacecraft autonomy and providing this capability to more missions across NASA and other government agencies. The cFS architecture upon which autoNGC is built is open-source, and the plug and play nature of this architecture enables it to be adapted to fit the needs of a particular mission. Several autoNGC apps are already available on the NASA software catalog for use by other government agencies. The flexibility of the autoNGC architecture facilitated its adaptation for a flight test in the first half of 2026 onboard the CAPSTONE spacecraft which operates in a near rectilinear halo orbit (NRHO) near the Moon.^{32,33} This test campaign required adding new capabilities to autoNGC including new measurement types, algorithms for optical navigation, and maneuver planning for NRHO station-keeping. Both the Starling and CAPSTONE autoNGC flight tests, along with additional flight opportunities planned in the years ahead, are built upon strong partnerships and the flexibility enabled by the widely used NASA cFS architecture. These qualities will continue to be at the core of achieving the mission of autoNGC.

References

1. S. Bhaskaran, N. Bosanac, L. Feaga, S. Jenkins, and T. Linn, "Report of the Autonomous Navigation Demonstration Relevance Assessment Team (ANDRAT)," National Aeronautics and Space Administration, Aug. 2024.
2. "NASA's Moon to Mars Strategy and Objectives Development," National Aeronautics and Space Administration, 2023.
3. M. D. Rayman, P. Varghese, D. H. Lehman, and L. L. Livesay, "Results from the Deep Space 1 Technology Validation Mission," in *50th International Astronautical Congress*, Amsterdam, The Netherlands, 1999, pp. 1–13.
4. M. D. Rayman and P. Varghese, "The Deep Space 1 Extended Mission," *Acta Astronautica*, vol. 48, no. 5–12, pp. 693–705, 2001, doi: [10.1016/S0094-5765\(01\)00044-3](https://doi.org/10.1016/S0094-5765(01)00044-3).
5. S. Bhaskaran *et al.*, "Orbit Determination Performance Evaluation of the Deep Space 1 Autonomous Navigation System," in *AAS/AIAA Space Flight Mechanics Meeting*, San Diego, CA: Univelt Inc., 1998.
6. S. Bhaskaran, "Autonomous Navigation for Deep Space Missions," in *SpaceOps 2012 Conference*, Stockholm, Sweden, 2012, pp. 1–13. doi: [10.2514/6.2020-4139](https://doi.org/10.2514/6.2020-4139).

7. B. Kennedy *et al.*, “Satellite-to-satellite imaging in support of LEO optical navigation, using the ASTERIA CubeSat,” in *CubeSats and SmallSats for Remote Sensing IV*, C. D. Norton, T. S. Pagano, and S. R. Babu, Eds., Online Only, United States: SPIE, Aug. 2020, p. 15. doi: [10.1117/12.2567794](https://doi.org/10.1117/12.2567794).
8. L. Fesq *et al.*, “Results from the ASTERIA CubeSat Extended Mission Experiments,” in *2021 IEEE Aerospace Conference (50100)*, Big Sky, MT, USA: IEEE, Mar. 2021, pp. 1–11. doi: [10.1109/AERO50100.2021.9438402](https://doi.org/10.1109/AERO50100.2021.9438402).
9. S. Bhaskaran and B. Kennedy, “Terminal guidance navigation for an asteroid impactor spacecraft,” in *AAS/AIAA Astrodynamics Specialist Conference*, Hilton Head, South Carolina, 2013, pp. 1–18.
10. J. E. Riedel *et al.*, “Configuring the Deep Impact AutoNav System for Lunar, Comet and Mars Landing,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, 2008, pp. 1–21. doi: [10.2514/6.2008-6940](https://doi.org/10.2514/6.2008-6940).
11. J. Bellerose *et al.*, “Double Asteroid Redirection Test (DART): Navigating to obliteration,” *Acta Astronautica*, vol. 219, pp. 417–427, 2024, doi: <https://doi.org/10.1016/j.actaastro.2024.02.021>.
12. D. Folta and A. Hawkins, “Results of NASA’s First Autonomous Formation Flying Experiment: Earth Observing-1 (EO-1),” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Monterey, California: American Institute of Aeronautics and Astronautics, Aug. 2002. doi: [10.2514/6.2002-4743](https://doi.org/10.2514/6.2002-4743).
13. S. De Florio, S. D’Amico, and J. S. Ardaens, “Flight Results from the Autonomous Navigation and Control of Formation Flying Spacecraft on the PRISMA Mission,” in *61st International Astronautical Congress*, Prague, Czech Republic, Sept. 2010.
14. S. D’Amico, J.-S. Ardaens, and R. Larsson, “Spaceborne Autonomous Formation-Flying Experiment on the PRISMA Mission,” *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 3, pp. 834–850, May 2012, doi: [10.2514/1.55638](https://doi.org/10.2514/1.55638).
15. R. Contreras, L. Peñín, V. Marco, A. Cropp, and T. Flinois, “Proba-3: High precision Formation Flying in HEO,” in *AAS Guidance, Navigation, and Control Conference*, Breckenridge, CO, 2017.
16. S. Tiraplegui, D. Serrano, L. F. Peñín, R. Contreras, and G. Rodríguez, “Proba-3: Challenges and Needs for Sub-Millimetre Autonomous Formation Flying,” in *8th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, Madrid, Spain, 2019.
17. G. Bonin, N. Roth, S. Armitage, J. Newman, B. Risi, and R. E. Zee, “CanX-4 and CanX-5 Precision Formation Flight: Mission Accomplished!,” in *29th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, Aug. 2015.
18. S. Miller *et al.*, “Starling CubeSat Swarm Technology Demonstration Flight Results,” in *38th Annual Small Satellite Conference*, Logan, UT, Aug. 2024, pp. 1–13.
19. N. Benz, “Starling CubeSat Swarm: Flight Results and Lessons Learned,” presented at the NASA Learn Forum, Oct. 02, 2024.
20. C. Adams *et al.*, “Advancing Autonomy in Distributed Space Systems: Results from the Distributed Spacecraft Autonomy Experiment on Starling 1.0,” in *38th Annual Small Satellite Conference*, Logan, UT, Aug. 2024.
21. J. Kruger, S. D. ’ Amico, and S. S. Hwang, “Starling Formation-Flying Optical Experiment: Initial Operations and Flight Results,” in *38th Annual Small Satellite Conference*, Logan, UT, Aug. 2024.
22. H. Sanchez *et al.*, “Starling1: Swarm Technology Demonstration,” in *32nd Annual AIAA/USU Conference on Small Satellites*, Logan, UT, Aug. 2018.
23. D. McComas, J. Wilmot, and A. Cudmore, “The Core Flight System (cFS) Community: Providing Low Cost Solutions for Small Spacecraft,” in *30th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, Aug. 2016.
24. L. B. Winternitz, W. A. Bamford, S. R. Price, J. R. Carpenter, A. C. Long, and M. Farahmand, “Global Positioning System Navigation Above 76,000 KM for NASA’S Magnetospheric Multiscale Mission,” *NAVIGATION: Journal of the Institute of Navigation*, vol. 64, no. 2, pp. 289–300, Jun. 2017, doi: [10.1002/navi.198](https://doi.org/10.1002/navi.198).
25. M. Farahmand, A. Long, and R. Carpenter, “Magnetospheric multiscale mission navigation performance using the Goddard enhanced onboard navigation system,” in *Proceedings of the 25th International Symposium on Space Flight Dynamics*, Munich, Germany, Oct. 2015, pp. 19–23.
26. “Aerospace and NASA Technologies Provide Novel Framework for Autonomous Navigation in Cislunar Space,” Aerospace Corporation. Accessed: May 15,

2026. [Online]. Available: <https://aerospace.org/article/aerospace-and-nasa-technologies-provide-novel-framework-autonomous-navigation-cislunar>
27. B. W. Ashman *et al.*, “Autonomous Navigation of a Lunar Relay Using GNSS and Other Measurements,” in *37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024)*, Baltimore, Maryland, Oct. 2024, pp. 3156–3173. doi: [10.33012/2024.19673](https://doi.org/10.33012/2024.19673).
28. S. Y. W. Low, T. Bell, and S. D’Amico, “Flight-Ready Precise and Robust Carrier-Phase GNSS Navigation Software for Distributed Space Systems,” in *AIAA SciTech Forum*, Orlando, FL, Jan. 2026.
29. N. Stacey, “Robust Autonomous Spacecraft Navigation and Environment Characterization,” Ph.D. Dissertation, Stanford University, 2022. doi: [10.22541/essoar.167214442.29146396/v1](https://doi.org/10.22541/essoar.167214442.29146396/v1).
30. J. R. Carpenter and C. N. D’Souza, “Navigation Filter Best Practices,” NASA Engineering and Safety Center, Hampton, VA, Technical Publication TP–2018–219822, Mar. 2025. [Online]. Available: <http://www.sti.nasa.gov>
31. T. Stevenson, “Flight Results and Lessons Learned from the Starling Propulsion System,” in *38th Annual Small Satellite Conference*, Logan, UT, Aug. 2024.
32. R. E. Pritchett *et al.*, “Navigation and Maneuver Planning Results from the AutoNGC CAPSTONE Flight Test,” in *AAS/AIAA Astrodynamics Specialist Conference*, Whistler, BC, CA, Jul. 2026.
33. A. J. Liounis *et al.*, “AutoNGC Flight OpNav Experiment Results on CAPSTONE,” in *40th Annual Small Satellite Conference*, Salt Lake City, UT, Aug. 2026.