(Preprint) AAS 20-634

GUIDANCE MODIFICATIONS AND ENHANCEMENTS FOR SPACE LAUNCH SYSTEM BLOCK-1 IN SUPPORT OF ARTEMIS I AND BEYOND

Matt Hawkins* and Naeem Ahmad†

NASA is currently building the Space Launch System (SLS) Block-1 launch vehicle for the Artemis I test flight. Design of the Artemis II mission, which will use the Block-1 vehicle to take astronauts around the moon for the first time in decades, is also underway. The Guidance, Navigation, and Controls (GN&C) algorithms will be largely similar for the two missions. However, the extensive simulation and testing campaign for Artemis I has revealed opportunities for improvements in the GN&C algorithms, allowing more effective use of the capabilities of the SLS vehicle, and enhancing safety for the astronauts aboard. This paper will describe several planned algorithm updates for the Artemis II mission. The updates enhance the Powered Explicit Guidance (PEG) algorithm and auxiliary guidance algorithms.

INTRODUCTION

The Space Launch System (SLS) guidance team is responsible for design and implementation of all guidance and steering algorithms for the SLS vehicle. While the focus has been on preparing for the Artemis I mission, design and analysis work has also been performed for Artemis II and beyond, including missions that use both the Block-1 and the Block-1B vehicles. Previous guidance design work includes a trade study to compare candidate guidance algorithms for the Block-1B vehicle,[1] and updates to guidance algorithms based on lessons learned during that trade study.[2]

During the extensive simulation and testing campaign for Artemis I, several opportunities for improvements to the guidance algorithms have been identified. Given a desire to finalize the Flight Software (FSW) design for Artemis I, most of the improvements described are scheduled to be included on Artemis II. However, one of the improvements was included in a late FSW algorithm change, and another improvement is primarily a process improvement and does not necessitate a change to the FSW.

For the SLS Artemis I mission, the onboard guidance system will be used for Boost Stage (BS) flight, defined as the portion of flight when the Solid Rocket Boosters (SRBs) are used and provide the majority of the thrust, through Core Stage (CS) flight, when the SRBs have separated and all thrust is provided by the four core engines. BS flight uses open-loop roll-pitch-yaw and throttle commands (called the Chi Table) as determined by the Chi Angle Optimizer (CHANGO)[3] tool. During

* PhD, Aerospace Engineer, EV42/Guidance, Navigation, and Mission Analysis Branch, Jacobs Space Exploration Group, Huntsville, AL 35806.

† Aerospace Engineer, EV42/Guidance, Navigation, and Mission Analysis Branch, NASA Marshall Space Flight Center, Huntsville, AL 35812.

CS flight, the Powered Explicit Guidance (PEG) algorithm is used. SLS uses an implementation of PEG derived from the Space Shuttle program.

In addition to Open-Loop Guidance (OLG), which uses the Chi Table, and Closed Loop Guidance (CLG), which uses PEG, the guidance system includes all steering functionality, and additional auxiliary algorithms. The steering system converts Chi Table and PEG commands into body rate and attitude commands for the Flight Control System (FCS). The auxiliary functions include a throttle manager, an acceleration limiter based on Shuttle's g-limiting algorithm,[4] an engine cutoff algorithm, a mass estimation algorithm, and a targeting algorithm for launch window effects.[5]

One of the most challenging scenarios for the guidance system is the loss of a single core engine, known as Engine Out (EO). In the event of an EO, the guidance system may not be able to reach the nominal target set. The desire is to continue on to the nominal Main Engine Cutoff (MECO) targets, called Press to MECO (PTM). If PTM is not feasible, two alternative target sets, High Energy Alternate MECO Target (AMT-HI) and Low Energy Alternate MECO Target (AMT-LO), are available for automatic downselection. The exact targets are designed to allow some test objectives to be completed while avoiding CS re-entry footprint concerns. For a BS EO, the pitch commands from the Chi Table will be augmented with an empirically determined delta polynomial. For a CS EO, only the potential downselect is needed, and PEG will re-target if needed. The AMT-HI and AMT-LO target sets are available for manual selection for contingency scenarios, however re-targeting a fully functional vehicle is not an especially stressing case for the guidance system.

IMPROVEMENTS

Launch Window

A Launch Opportunity is defined as a period of the year during which SLS is capable of completing Orion's primary test objective. For the Artemis I mission, this period of opportunity is divided into multiple feasible periods on up to seventeen consecutive days. For each day, there is feasible span of time during which launch must occur to satisfy the primary test objective. This span of time varies from twenty minutes to over four hours.

The launch window is the span of time within a given day when the vehicle can launch and achieve its objective: reaching a target orbit within an acceptable mass penalty. The planned ascent trajectory is designed for liftoff at a single point in time. To accommodate launching before or after the planned time, the launch window adjustment algorithm computes needed adjustments to the Right Ascension of the Ascending Node (RAAN) and inclination targets at insertion, and to the Chi Table Yaw profile. The Chi Table consist of throttle and attitude commands which are used to steer SLS during BS flight. These adjustments ensure that the vehicle reaches an acceptable orbit, including alignment for Earth-Moon missions or rendezvous missions, while maximizing the payload delivered.

Analysis of trajectories generated by Copernicus[6] and Program to Optimize Simulated Trajectories (POST),[7] two standard Three Degrees-of-Freedom (3-DOF) trajectory optimization tools, show that as the launch time is varied across the launch window, the ascent RAAN and inclination targets change in order to preserve optimality for the entire mission. Examination of the change in target data shows that data can be curve fitted by a polynomial function. For the baseline design, the quadratic fit was sufficient to model the trajectory, with assumptions that the launch window spans no more than two hours and the reference launch time (middle of launch window) always corresponds to a due-East launch azimuth (near 90 degrees). However, updates in recent design cycles

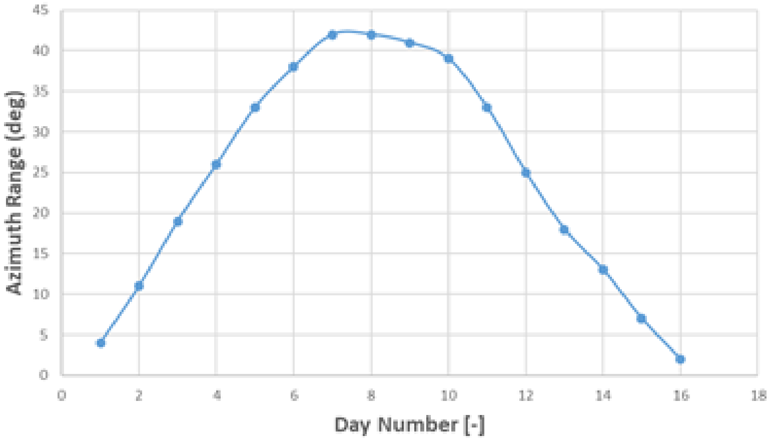

Figure 1: Variation in Azimuth Range Over a Launch Period Compared to Expected Minimum

rendered both of the assumptions invalid. Instead of a launch window spanning no more than two hours, launch windows could span longer than four hours. Moreover, the assumption of a due East launch at the middle of the launch window was no longer valid. In fact, some of the days did not include a due East launch at all. These changes essentially created a hole in the Guidance, Navigation, and Controls (GN&C) design process which needed to be addressed rapidly as the design has significantly matured since the original launch window algorithm was implemented. Moreover, the ability to use single set of Chi Table inputs for a launch opporunity was not feasible, as the reference azimuth target was not a constant for due East any more, and instead varied day to day. Figure 1 below shows the day to day variation in launch azimuth in a single launch period, referenced to a minimum expected azimuth.

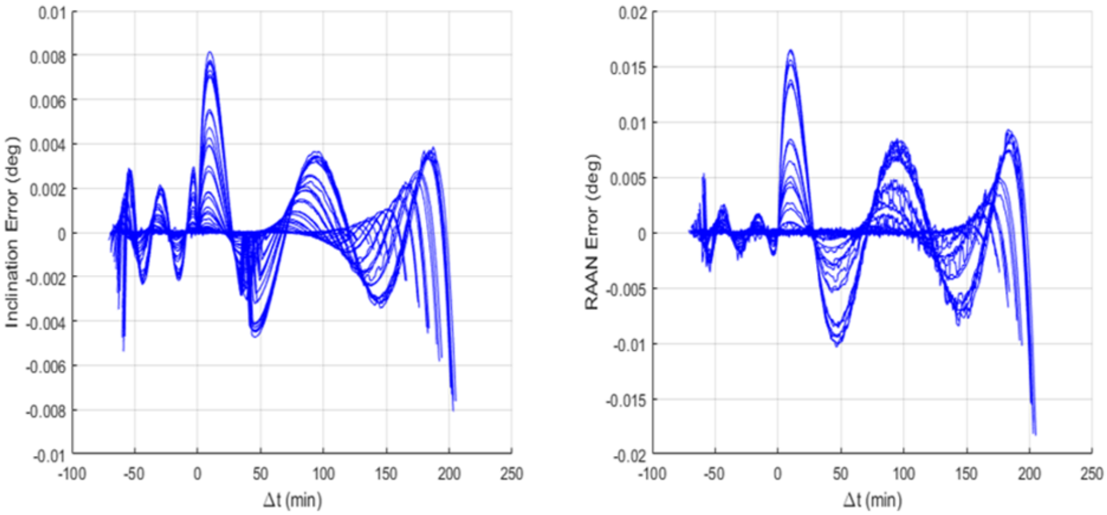

Figure 2: Polynomial Curve Fit Errors For First Four Launch Periods

Further data analysis has shown that at a least fifth-order polynomial is required to estimate extended launch window data. Furthermore, the reference point for the polynomial needs to be a launch azimuth which is approximately at the middle of azimuth range.The Chi Table also needs to use this reference azimuth. Lastly, to further reduce the polynomial estimation error, preserving the original implementation, two sets of fifth-order polynomials are used, each referenced to the middle of launch azimuth. Polynomial curve fitting azimuths that are lower than the reference launch azimuth are designated as “Northerly” curves and azimuths higher than the reference launch azimuth are designated as “Southerly” curves. A change was made to the FSW to implement fifth-order polynomials. Figure 2 shows the polynomial fit error for all launch windows in the first four periods of launch opportunity that span more than four hours.

Updating the polynomial order is a great solution that solved immediate needs, but future Artemis mission will need a table lookup approach. Although one negative side of table lookup is that it requires carrying extra information onboard, the great advantage to a table lookup is that the errors associated with polynomial fits are eliminated in place of simple interpolation error, which will result in improved overall performance. Furthermore, it removes the assumption that the optimal change in the RAAN and inclination profile can always be fit to a polynomial, ensuring suitability for other mission classes.

PEG Solution Hold

The PEG algorithm is robust for the class of missions using the Block-1 vehicle. As a predictor-corrector, it takes some number of iterations to reach a solution, known as the iteration count. The maximum iteration count before convergence failure is declared is 10, while in practice the maximum iteration count rarely exceeds 5 in Monte Carlo dispersions for nominal flight. Even in flights with significant failures, for example loss of a core stage engine, the iteration count rarely reaches 10. However, there are off-nominal scenarios for which the iteration count reaches 10. The baseline Artemis I behavior in the event that PEG unconverges is to transition to attitude hold steering, taking a snapshot of the vehicle attitude at the time of the transition and continuing to hold that attitude. Despite the intent of the attitude to have minimal impact, this behavior induces rates on the vehicle, as the turning rate commanded by PEG instantly becomes zero.

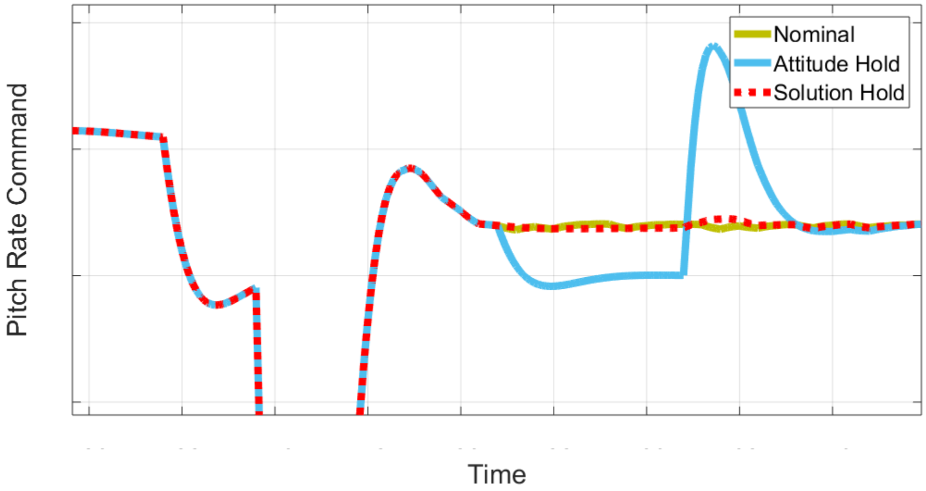

Figure 3: Pitch Rate Commands Around LAS Jettison

Given the wide variety of conditions that could cause PEG to become unconverged, there is no universal best response. The attitude hold can be characterized as a “first-do-no-harm” response.

However, the rates induced by the transition from the PEG solution to the frozen attitude can cause vehicle stress. Another option is to hold the last converged PEG solution and continue to follow it. This response assumes that the guidance system had better knowledge when it was converged, and that the vehicle should attempt to stay on the same path until convergence is restored. To demonstrate this idea, a nominal PEG simulation was run with the iteration count artificially set to 10 at shortly after SRB separation, for both the attitude hold and the solution hold. A nominal case was also simulated. Figure 3 shows the pitch rate commands around and after SRB separation. It can be seen that the solution hold is nearly identical to the nominal case, while the attitude hold causes significant pitch rate commanding at the beginning and end of the attitude hold.

The PEG solution hold is intended to handle short or intermittent non-convergence. If PEG remains unconverged for an extended amout of time, it is expected that corrective actions will be taken, including re-targeting or mission abort. However, a proof-of-concept test was performed showing admirable performance for an otherwise nominal flight with an artificial PEG solution hold.

The CS flight of SLS is divided into two segments, before and after Launch Abort System (LAS) jettison. The jettison event entails an instantaneous change of vehicle mass, and the iteration count typically increases for one or two guidance cycles at this staging event. After LAS jettison, the rest of the ascent flight is reasonably benign for the guidance system, with minimal aerodynamic effects and no further solution discontinuities. A test case was run for a nominal vehicle with no Monte Carlo-type dispersions. Shortly after LAS jettison, the iteration count was artificially set to 10 for the remainder of the flight, causing PEG to follow a static solution. Note that the g-limiting algorithm and the engine cutoff algorithm use measured states independent of PEG, so the vehicle will enter the g-limiting phase and command MECO regardless of stale PEG data.

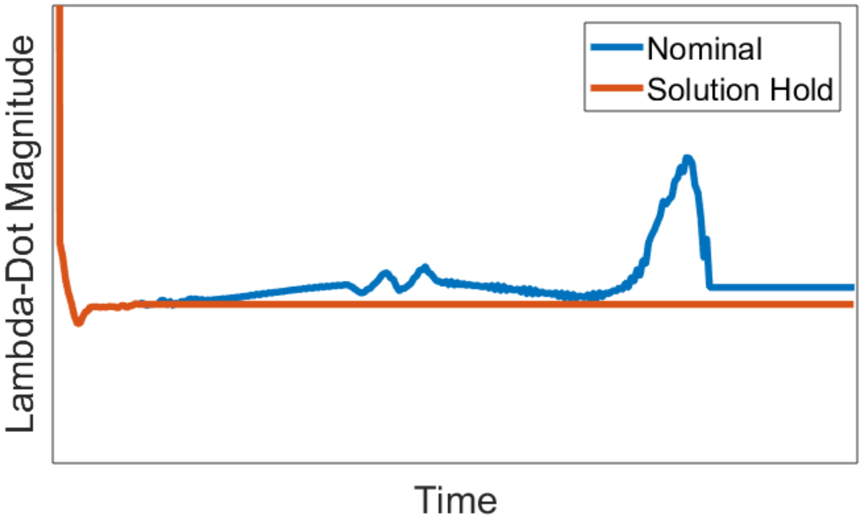

Figure 4: PEG’s Lambda Dot for Nominal Flight and Solution Hold

Figure 4 shows the calculated $\dot{\lambda}$ from PEG for a nominal run and the solution hold, starting shortly after SRB separation. The nominal flight shows signatures of a Programmed Test Input (PTI), a test input by the FCS to test the vehicle response, as well as the g-limiting phase and the release of PEG constraints. In contrast, the solution hold shows a constant $\dot{\lambda}$, as expected. Figure 3 shows pitch

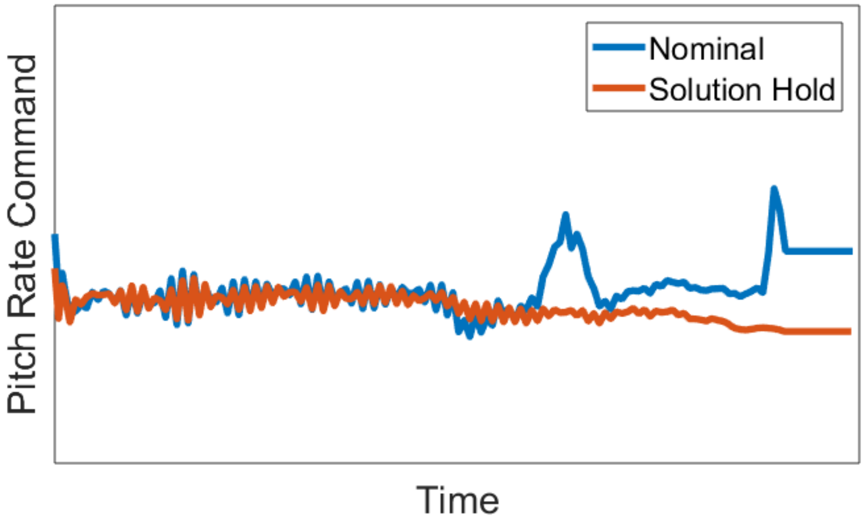

Figure 5: Pitch Rate Command for Nominal Flight and Solution Hold

Table 1: Insertion Errors for Nominal Case and Solution Hold

Case	Apogee Error, nmi	Perigee Error, nmi	Wedge Angle, Deg
Nominal	-0.721	0.038	0.0015
Solution Hold	-1.134	-0.478	0.0216

rate commands for a nominal run and the solution hold, starting after the PTI event. Since the pitch rate comes from the steering function, the pitch rate command is responding to flight conditions. Toward the end of flight, it is seen that the nominal flight acts on the PEG commands, while the solution hold is unaffected. Table 1 shows the insertion apogee, perigee, and wedge angle errors for the two runs. The errors are given relative to the navigated states, so they represent the errors due to the guidance system. Despite enacting the solution hold for a significant amount of the total flight, including no chance to recompute the solution near MECO, the solution hold for an otherwise nominal run results in insertion errors that are well within required values.

Spherical Trigonometry

On the launch day, at liftoff, FSW will receive the current Global Positioning System (GPS) time, which will be used along with the reference launch azimuth time to update RAAN and inclination targets to maximize payload delivery and ensure proper Earth-Moon alignment. To reach the new inclination and RAAN targets, the Chi Table yaw command is adjusted with a yaw bias. From a performance standpoint, it is better to yaw the vehicle early in the flight since less control authority will be required as vehicle has not gained as much momentum. If the Chi Table is not adjusted, all the yaw adjustment will be completed by the guidance system after BS flight, negatively impacting performance. A constant yaw bias is sufficient to improve performance.

The baseline approach to determine the yaw bias was to use a brute force method, simulating up to 25 different trajectories, with the assumption that these trajectories are representative of full launch period. The baseline approach was acceptable and performed as expected. However, due to

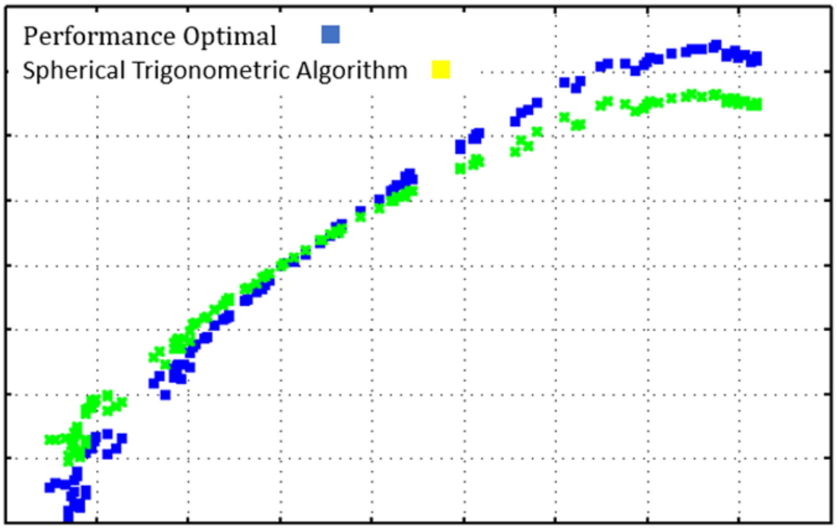

Figure 6: Comparison of Yaw Bias Estimate Using Spherical Trig and Brute Force

updated mission design, the brute force method needed to be run for each day in a period consisting of up to 17 days. As new launch periods open approximately monthly, this process is repeated every month, presenting a significant challenge for the Guidance team. The solution to process all this data was to replace the brute force method with a Spherical Trigonometric (ST) approach. An ST algorithm, an offline tool, is employed to determine the optimal yaw bias at any given launch time. Inputs to this algorithm are the target orbital plane and launch time for both the reference and actual launch times. This algorithm has tunable parameters, which have been tuned for the SLS Block-1 configuration, and have demonstrated near-optimal performance over a wide range of possible SLS target trajectories as shown in Figure 6. Figure 6 represents analysis of the first four launch periods, containing almost 70 days. For each day, two trajectories were simulated. The simulated trajectories correspond to the open and the close of the launch window, representing extreme performance cases. Output of the ST algorithm is used to fit fifth-order polynomial curves, representing the Chi Table yaw bias.

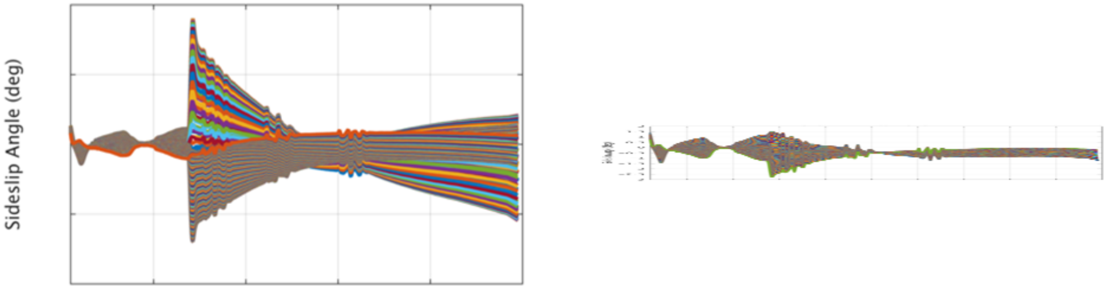

Figure 7: Sideslip Angles for Simulated Launches Every 1 Minute For a Single Launch Day Using Baseline (Left) and Spherical Trig (Right). Adjusted to Match Vertical Scale.

Evidence of the benefits of this updated implementation can also be seen Figure 7, showing that the sideslip angle has been significantly improved. Figure 7 has been adjusted so the the vertical scales are the same on the right-hand side and the left-hand side. The extreme adjustments necessary on the right-hand side show the dramatic improvement. A small overall sideslip angle means that Chi Table yaw bias was optimal.

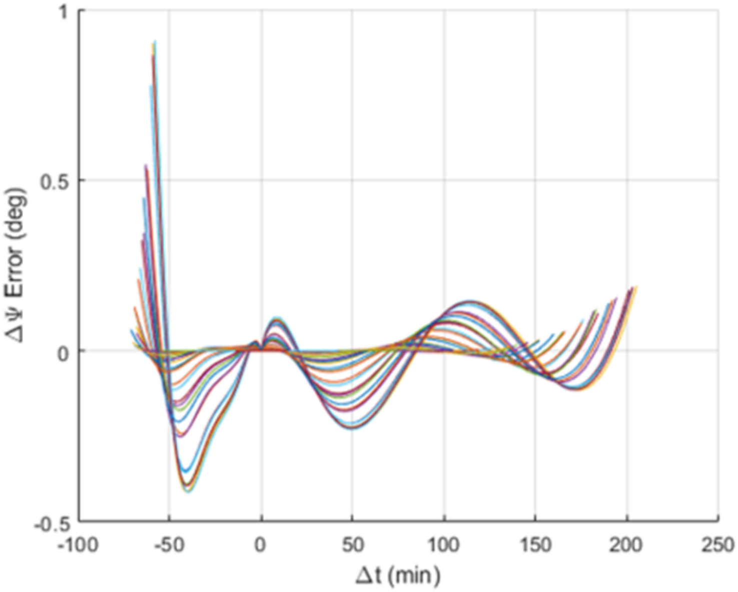

Figure 8: Chi Table Yaw Bias Error

This method worked well for a recent design cycle, however there is room for improvement as the polynomial fit introduces some amount of error, shown in Figure 8. For this reason, for future Artemis vehicles, the yaw bias polynomial will be replaced with an active ST algorithm.

Navigated Acceleration

The fundamental baseline SLS Guidance algorithms are inherited from the Shuttle program. Many of the core algorithms are implemented as they were exercised on Shuttle. Some examples include the steering filter, OLG, and CS guidance. These algorithms are further tailored to fit the SLS Block-1 configuration. However, as part of FSW Hardware In-The-Loop (HWIL) testing, rare and unexpected anomalies are revealed. This was the case for the onboard acceleration computation, which had been demonstrated to work reliably, until it was discovered that one odd case did not successfully maintain vehicle axial acceleration. Upon investigation of the root cause of the issue, it was determined that variation in estimated acceleration exceeded a threshold which resulted in a re-set of the acceleration limiting filter. Due to highly nonlinear dynamics, the filter was not able to converge back. The root cause was that the original algorithm, in various locations, used a finite difference form to estimate acceleration from velocity, as shown in Equation (1).

$$a_i = \frac{v_i - v_{i-1}}{\Delta t} \tag{1}$$

where v_i is the current sensed velocity from Navigation, v_{i-1} is the sensed velocity from Navigation from the previous cycle, and Δt t is the time elapsed from previous cycle.

In a single HWIL test, at a single time point, mis-synchronization between the elapsed time and

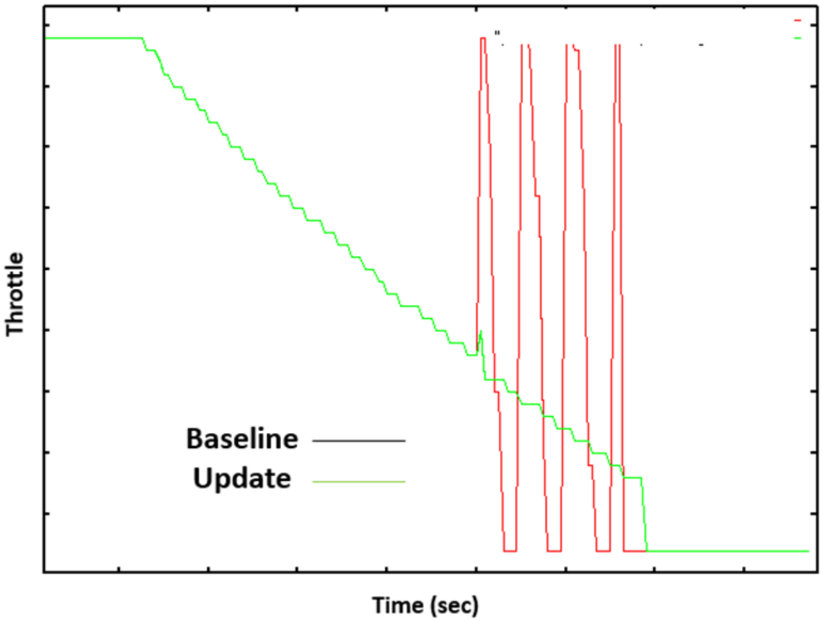

Figure 9: Throttle Response

state occurred, which caused an incorrect calculation of estimated acceleration and hence a catastrophic simulation failure. Since acceleration is one of the available signals from Navigation, the SLS guidance team replaced finite differences with the actual acceleration signal, filtered appropriately to remove high-frequency noise. This resolved the issue as shown in Figure 9, which shows updated throttle behavior.

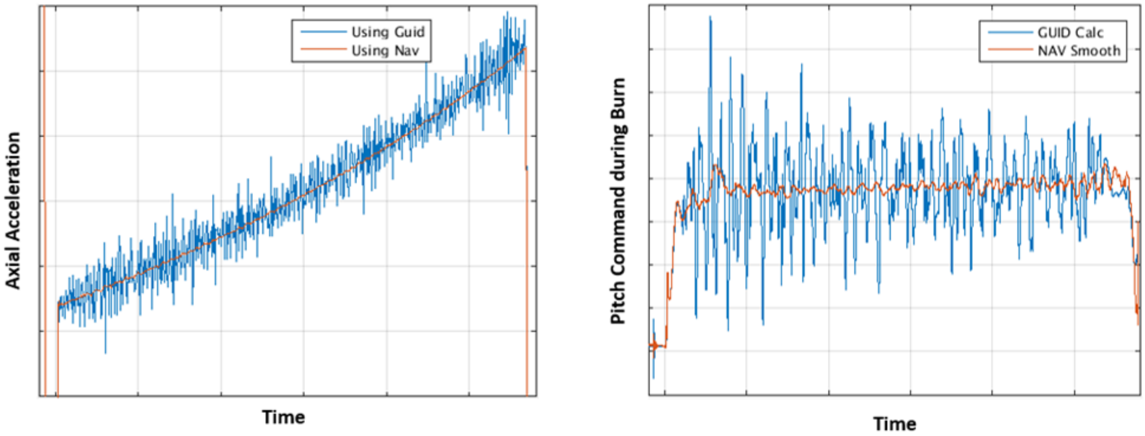

Figure 10: Acceleration Comparisons (L) and Resulting Pitch Commands (R)

Adding smooth acceleration greatly improved an attitude command oscillation for the SLS Block-1B cargo configuration mission, as shown in Figure 10. Figure 10 shows both the baseline acceleration and the Navigation-based acceleration and its effect on other parameters such as the pitch rate command.

Engine Out

The Block-1 SLS vehicle is capable of handling loss of thrust from a single core engine, called EO, at any time in ascent flight. EO design in CS flight consists solely of selecting the velocity boundary between AMT-HI and PTM, with the PEG algorithm automatically updating and flying to new targets if needed. The EO algorithm for BS flight is considerably more involved. In addition to the velocity boundary between AMT-LO and AMT-HI, two BS EO polynomial coefficients must be found.

The open-loop BS guidance algorithm uses a pre-determined roll-pitch-yaw program from the Chi Table. Examination of EO trajectories using POST showed that for an EO, only the pitch command needs to be modified. The roll command primarily ensures a heads-down orientation, and the yaw command affects the orbital plane. The FCS is able to overcome effects of thrust imbalance.

An EO can be considered a low-probability, high-impact event. During 135 flights in the Shuttle program, only one in-flight failure occurred, resulting in an Abort to Orbit (ATO),[8] analagous to the SLS Alternate MECO Targets (AMTs). However, proper EO algorithm design can ensure that vehicle loads are minimized and can maximize the availability of mission objectives. Given the balance between probability and impact, the design philosophy for EO is to make the FSW implementation as simple as possible, but no simpler. Each additional code path and input requires additional design and testing work by the guidance team, and additional code testing by the FSW team. At the same time, certain extra protections are necessary to prevent catastrophic failures at critical times, and well-designed algorithms can allow additional test objectives to be completed, retaining value that might otherwise be lost on a mission with a significant failure like an EO.

Experience during a vehicle analysis cycle presented a vivid demonstration of the value of good EO design. Early analysis cycles inlcuded a variety of vehicle configurations with different launch months, mass properties, and fuel loading. When a more flight-like vehicle was analyzed, it was discovered that AMT-HI was achievable with an EO at liftoff when using BS pitch augmentation. Without the pitch augmentation, AMT-HI could not be achieved for early EOs, and the AMT-LO option would have to be used. Based in part on this experience, new higher-energy AMTs were designed for the Artemis I mission which allowed more flight test objectives to be achieved.

The loss of a core engine represents a major step change in vehicle dynamics, and significantly affects the throttle manager, as a given throttle level now provides only three-quarters of the thrust. Several guidance algorithm updates to address throttle issues are described in Reference 9.

For a BS EO, the pitch from the chi table is augmented by a second-order polynomial, expressed as a function of altitude gained since the EO.

$$\theta_{New} = \theta_{Old} + \left[C_0 + C_1 \left(\Delta Alt_{curr} - \Delta Alt_{EO}\right) + C_2 \left(\Delta Alt_{curr} - \Delta Alt_{EO}\right)^2\right] \tag{2}$$

where θ_{New} is the new pitch command, θ_{Old} is the original pitch command from the Chi Table, $C_{0,\,1,\,2}$ are the polynomial coefficients, ΔAlt_{curr} is the current delta-altitude, and ΔAlt_{EO} is the delta-altitude recorded at the engine out event. The delta-altitude is the altitude gained since launch, and is the independent variable used by the Chi Table.[3]

A constant term in the pitch polynomial represents a step change in the pitch command, potentially causing significant rate changes as the vehicle initially responds to the EO. A linear blend was implemented to phase in the full polynomial augmentation. In addition to the coefficients, the blend time is another design parameter. Because the intent of the blend is to account for the general effects of the EO, only one blend time is specified which is applied to both AMT-HI and AMT-LO.

$$\theta_{cmd} = \begin{cases} \theta_{New} * \left(\frac{t_{nav} - t_{EO}}{t_{blend}}\right) + \theta_{Old} * \left(1 - \frac{t_{nav} - t_{EO}}{t_{blend}}\right) & (t_{nav} - t_{EO}) < t_{blend} \\ \theta_{New} & \text{otherwise} \end{cases} \tag{3}$$

where θ_{cmd} is the final pitch command with both the polynomial and blending applied, t_{nav} is the current time from Navigation, t_{EO} is the time of the EO event, and t_{blend} is the blend time parameter.

In general, the guidance team works closely with the trajectory team and the POST 3-DOF design tool, and EO is no exception. While guidance design is ultimately constrained by the FSW implementation, trajectory design by POST has fewer constraints. For BS EOs, POST is constrained to follow the original Chi Table until the EO occurs, but other constraints can be chosen based on design goals. In contrast, the guidance design must employ the second order pitch polynomial. POST can individually optimize the trajectory for each different EO time, while one common set of coefficients must be used for AMT-LO, and a separate set for AMT-HI. The differences in approach are well understood by both teams, and in general the POST simulations are not overly limited to mimic the guidance implementation, while taking into account that the guidance FSW cannot fully match the POST design for all cases.

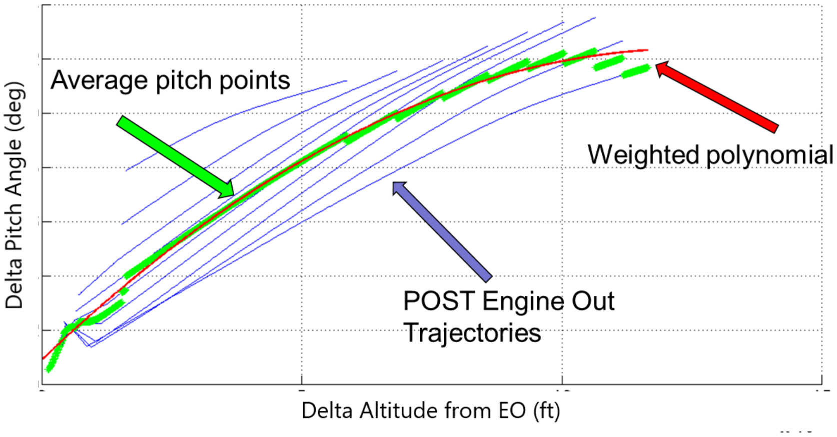

Figure 11: EO Curve Fit, Early Design Cycle

During early design cycles, it was discovered that EO polynomials designed by a curve fit to POST data typically worked well. As the vehicle design matured and the trajectory and guidance teams gained experience, the POST design was no longer conducive to the curve-fit approach. Figure 11 shows an example of the curve fit procedure for an early design cycle, and Figure 12 shows the same curve fit procedure for the FRAC-0 design cycle. Although the early curve fit doesn't cover some of the extreme cases, it can be seen that the general shape of the curve matches, and the polynomial is a reasonable middle-of-the-road fit to a variety of different cases. In contrast, the recent curve fit clearly misses the general shape. This is in part due to a desire to minimize additional maneuvering until Mach 2.0, which can be easily handled with POST but is much harder to implement with the guidance FSW. This limits the effectiveness of a simple curve fit to existing POST data.

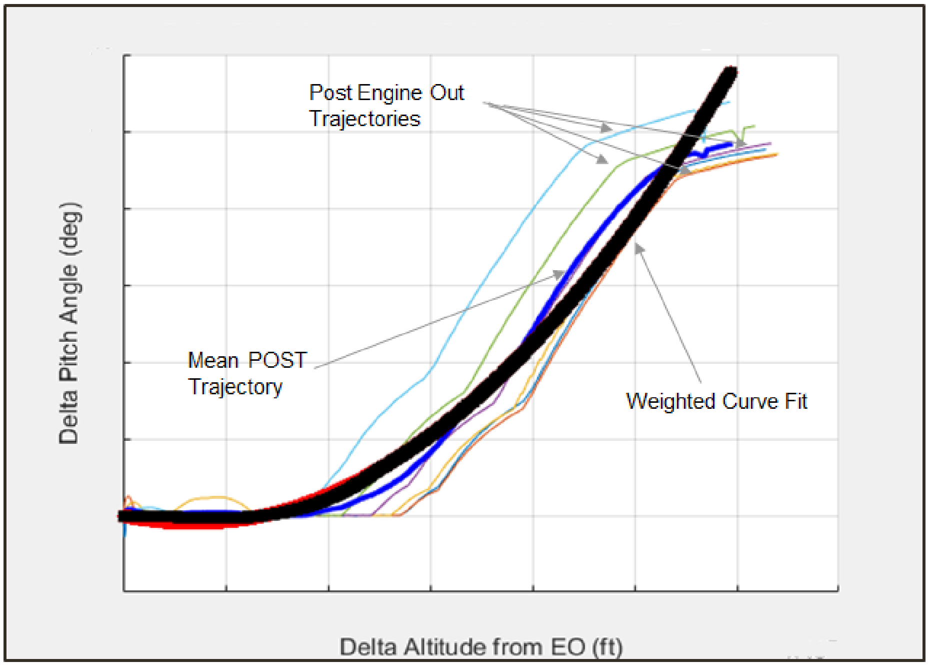

Figure 12: EO Curve Fit, Recent Design Cycle

In addition to the overall goal of increasing the availability of the AMT orbits, BS EO design must balance several competing demands. In general, a more aggressive polynomial design allows the AMT boundaries to shift earlier, allowing AMT-HI for earlier EOs. In addition, the total angle of attack in the transonic region needs to be minimized. The angle of attack at SRB separation must be kept within the bounds specified by the liftoff and separation team. Since an EO tends to increase the maximum Q - α (pitch dynamic pressure), the maximum Q - α must be monitored to stay under desired limits. Finally, any potential designs must be checked to ensure the re-entry footprint does not violate any keepout zones on or near populated landmasses.

Knowing the various constraints, a new approach was implemented for the FRAC-1 analysis cycle. Rather than attempting a curve fit to the POST EO pitch profile, instead the polynomials will be designed to achieve the desired macroscopic trajectory effects.

The full EO design problem, then, is to find seven BS parameters: Three polynomial coefficients for each AMT, as well as the single blend time. The velocity boundary between AMT-LO and AMT-HI also affects BS EO design, but it is typically not directly selected along with the polynomials. Instead, polynomials are designed to cover the expected range of times, with the exact boundaries selected later based on performance to orbit. Each polynomial design must work for EOs at any time in the selection range for the given AMT.

Because the exact trajectory effects of a polynomial design are not conducive to mathematical prediction, some amount of brute force processing is required, typicallly testing a grid of design points and examining the results. The full number of runs for a grid parametric study of the seven design parameters for a range of EO times would be prohibitive, as searching across only five values of each parameter for five different EO times would require approximately 400,000 runs.

To allow practical solution of the problem in a reasonable amount of time, several simplifying assumptions are made. The AMT-LO and AMT-HI polynomials can be designed separately. Other than the blend time, the two are decoupled. To support this assumption, the blend time will be determined for AMT-LO, and that value applied to AMT-HI. Because AMT-LO encompasses the earlier region of flight with more vehicle dynamics and loads considerations, it is the driving case for blend time. To further reduce the number of assessments, the only EO time assessed will be the earliest case, which is at liftoff for AMT-LO and is immediately after the switchpoint for AMT-HI. Performance tends to increase monotonically with later EO times, as evidenced by the need for lower-energy AMTs to begin with. When assessing performance to orbit, only EOs at the earliest time need to be simulated to insertion. Assessments of load effects can be done for a variety of EO times and only simulating until SRB separation.

The reduced AMT-LO problem now consists of finding the three coefficients and the blend time for different EO times. The Figures of Merit (FOMs) for this problem are:

1. Insertion mass for EO at liftoff
2. Maximum Q - α_{total} for different EO times
3. Maximum α_{total} in the transonic region for different EO times
4. Angle of attack at SRB separation

Some additional simplificaitons make the problem tractable. For any C_1 and C_2, there is a single C_0 that optimizes performance at liftoff. A common technique used by the guidance team for design is parabolic optimization. For many parameters, an optimal value maximizes insertion mass, with degrading performance on either side of the optimum. After some testing it was found that a near-optimal C_0 can typically be found with a single pass of parabolic optimization. For a given C_1 and C_2, three values of C_0 are tested for insertion mass and a parabola is fit. The optimal C_0 is selected as the one that maximizes the parabola. For each set of $C_{0,1,2}$, a small number of blend times and EO times are tested, and the blend time that minimizes the peak Q - α_{total} is selected as the blend time.

After following this procedure, each pair of C_1 and C_2 now has an optimal C_0 and blend time. The other FOMs can be plotted for each pair, and the optimal pair selected.

Annotated heatmaps of the four FOMs are shown in Figure 13. For the range of ceofficients tested, perfomance to orbit increases as the coefficient values increase. The various loads and trajectory constraints are expected to limit the feasible values of the coefficients, and this is indeed what is seen. The lines of mimimum peak Q - α_{total} and minimum peak α_{total} in the transonic region are roughly orthogonal to the direction of increasing performance. Finally, the hard constraint on angle of attack at SRB separation is illustrated. For this particular EO design, the SRB separation constraint is the active constraint. C_1 and C_2 are chosen to maximize performance to orbit along the SRB separation constraint.

To verify the design, a Monte Carlo simulation was run with the new parameters and compared to the parameters found by attempting to match the POST trajectory. Figure 14 shows the angle of attack near SRB separation, with the desired limit shown. While the previous design violated the constraint, the new design stays just under the constraint, as desired. Other simulation results, not shown here, also indicate overall improvements in peformance to orbit and re-entry footprint. AMT-HI design follows a similar procedure, with the exception that the same blend time as AMT-LO must be used.

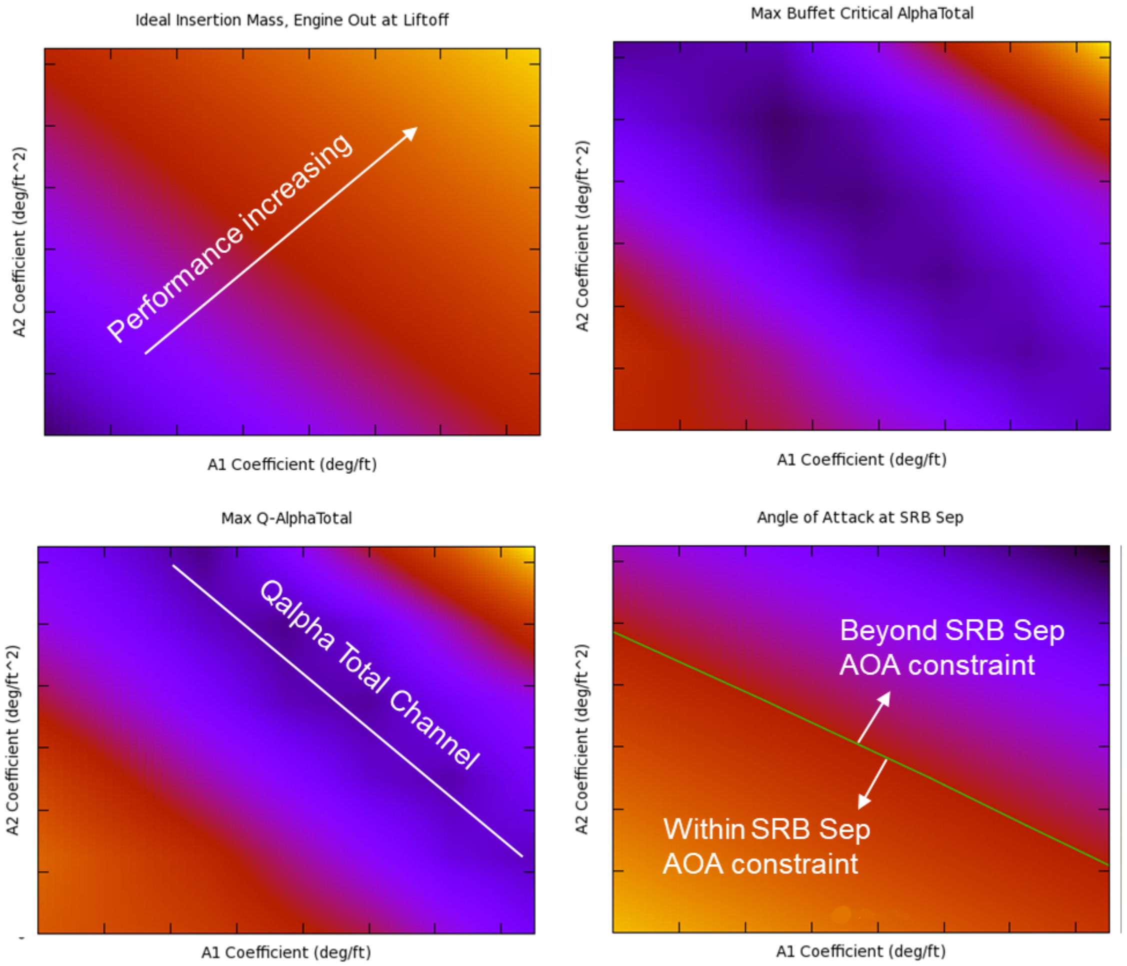

Figure 13: Heatmaps Showing Design Constraints

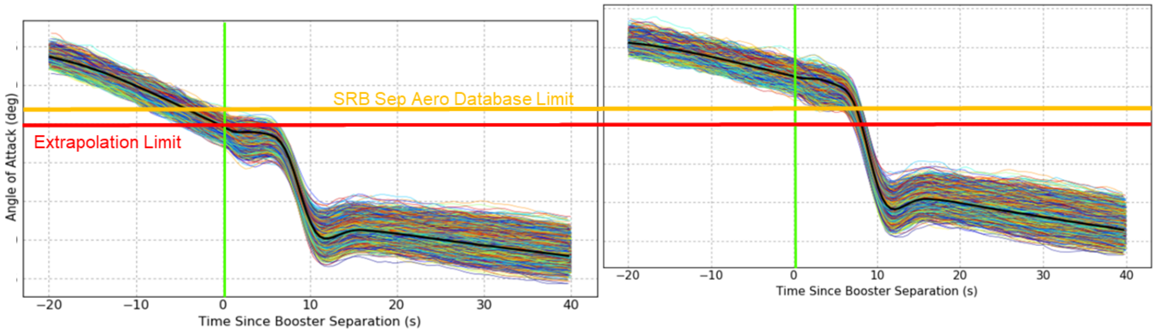

Figure 14: Angle of Attack at SRB Separation for Early (left) and Refined (right) Designs

CONCLUSION

As the SLS program evolves, the guidance FSW will be used for innovative and ambitious missions, and the guidance team will continue to apply lessons learned to improve and refine the guidance algorithms and design. The guidance improvements described in this paper are representative of the efforts of the SLS guidance team to prepare for Artemis I and subsequent missions. In addition to algorithm updates, process improvements will continue to be sought out. The guidance system is prepared for challenging aspects of Artemis I and future missions.

REFERENCES

[1] N. Ahmad, M. Hawkins, P. Von der Porten, R. Pinson, G. Dukeman, and T. Fill, "Closed Loop Guidance Trade Study for Space Launch System Block-1B Vehicle," *AAS 18-270*, 2018.

[2] P. Von der Porten, N. Ahmad, M. Hawkins, and T. Fill, "Powered Explicit Guidance Modifications and Enhancements for Space Launch System Block-1 and Block-1B Vehicles," *AAS 18-136*, 2018.

[3] M. Hawkins, N. Ahmad, and P. Von der Porten, "CHANGO: A Software Tool for Boost Stage Guidance of the Space Launch System Exploration Mission 1," *AAS 19-726*, 2019.

[4] J. Everett, J. Wall, and N. Ahmad, "Analysis and Application of Novel and Heritage Acceleration Limiting Algorithms for SLS on Ascent," *AAS 20-635*, 2020.

[5] A. Craig and N. Ahmad, "Space Launch System Launch Windows and Day of Launch Processes," *AAS 20-591*, 2020.

[6] J. Williams, *Copernicus Version 4.6 Users Guide*. NASA Johnson Space Center, Apr. 2018.

[7] S. Striepe, *Program to Optimize Simulated Trajectories (POST2), Vol 2: Utilization Manual, Ver 3.0*. NESC, NASA Langley Research Center, May 2014.

[8] R. Legler and F. Bennett, "Space Shuttle Missions Summary," Tech. Rep. TM-2011-216142, NASA Johnson Space Center, 2011.

[9] A. Craig, J. Hays, M. Hawkins, and J. Hauglie, "Space Launch System Engine Out Capabilities," *AAS 20-589*, 2020.