

HARDWARE VERIFICATION AND VALIDATION FOR A NAVIGATION SENSOR SOFTWARE MODEL IN SUPPORT OF FLIGHT VEHICLE PERFORMANCE ANALYSIS

Evan J. Anzalone,^{*} Nicholas Hoen,[†] Thomas Park[‡] , and Charles Weyandt[§]

... or, “It’s in the details, how to make complicated software perform like complicated hardware.” In attempts to minimize development time and quickly build an operational vehicle, NASA’s Space Launch System (SLS) has had to be intentional about integrated testing. Constraints on budget and schedule have required balance between testing needs and the desire for an integrated flight vehicle as soon as possible. To provide key insights early in design and analysis cycles, a large amount of effort has shifted into maturing and validating models at the component level with integrated testing as a means to validate their integration. In terms of SLS Navigation, this, and the model-based design approach have pushed explicit requirements for sensor models to be validated against flight hardware to high precision. This paper covers the approach taken to verify and validate the models for the two key navigation sensors on the SLS vehicle, the Redundant Inertial Navigation Sensor and the Rate Gyro Assembly. These models are used in performance evaluation, fault detection, and operations development extensively. Using a mix of data from hardware vendor documentation and testing reports, limited in-house testing, and integration activities, these models were able to be validated against flight hardware at multiple levels, from the internal software design to statistical behavior at the raw sensor and integrated box levels. The high level of insight into the hardware elements is instrumental to support flight certification activities and building confidence in SLS Navigation capability. Focused testing enabled additional insight and validation that proved invaluable and the resulting insights were used to focus and mature models. Additionally, of having validated performance-based hardware models enables a wide breadth of activities including detailed fault detection studies and integration into future vehicle frameworks, such as an upper stage and provide a valuable asset to continued SLS analysis and design.

INTRODUCTION

Inserting a payload accurately into a defined orbit is the major requirement of any launch vehicle system. For large vehicles that take years to design, build and fly, verifying this requirement is very complex, especially when test flight opportunities are limited or are not available. For the Space Launch System (SLS), a large number of requirements are verified through a combinations of hardware and software testing—both at the component level and at the integrated system level. Hardware characterization is used to feed the development of software models that are used in 6 degree of freedom simulation and hardware-in-the-loop testing.

^{*} SLS Navigation co-Lead, EV42 Guidance, Navigation, and Mission Analysis Branch, NASA/MSFC, Huntsville, AL 35801

[†] Aerospace Engineer, EV42 Guidance, Navigation, and Mission Analysis Branch, ESSCA Dynetics, Huntsville, AL 35801

^{‡‡} SLS Navigation co-Lead, EV42 Guidance, Navigation, and Mission Analysis Branch ESSCA/DCI, Huntsville, AL 35801

[§] Aerospace Engineer, EV42 Guidance Navigation and Mission Analysis Branch , ESSCA/Insight Global, Huntsville, AL, 35801

These hardware and software models are used to demonstrate system capability across a wide variety of vehicle configurations, launch environments, as well as robustness to off-nominal conditions. Cross-comparison between simulations at various level of hardware integration and with independent model implementations are used to provide the necessary evidence to ensure the models are a consistent and accurate representation of the system under design.

This approach is in contrast to that used on other programs, where the vehicle matures through cycles of increased capability, such as the Saturn program, and similar recent development programs, such as that of SpaceX^{**†} or Blue Origin[‡], which utilize extensive flight testing and demonstration as secondary missions to evolving integrated capability. Due to the vehicle complexity, schedule, and cost constraints, SLS is allocating one test flight, Artemis 1, to demonstrate integrated vehicle capability prior to human flight. As such, there is increased focus and attention to certification activities from individual hardware components up to an integrated stage, with testing focused on filling in analysis gaps, validating software models, and demonstrating integration performance prior to integrated vehicle flight.

This paper documents the approach taken to verify and validate the navigation subsystem of the SLS Block 1 vehicle. The two key elements of this are the navigation hardware and performance models. A description of the data products, their source, and their integration are included to provide insight into the system's capability. Both verification activities, which ensures that the model of the system matches the specification, and validation activities, which involves comparisons of the model performance to flight hardware data, detailed in this paper. Together these provide insight into a recommended approach for the usage of models in navigation system design for critical flight systems.

SLS MODEL-BASED ENGINEERING APPROACH

In order to streamline the development process and updates in computational models and process, the SLS program shifted away from a paperwork- and requirement-centric approach to systems capability definitions and vehicle design. Lessons learned from the Constellation program led to the implementation of a model-centric approach to vehicle design and internal collaboration among disciplines. In this definition, models can capture multiple facets and include such artifacts as a 3D model, a table of data, or executable code representing some portion of the vehicle. This is in contrast to the use of written documentation to pass requirements between systems and capture design elements. The impact of this is not to eliminate documentation, but to focus it and allow the models themselves to capture design elements and performance criteria. Similarly including traceability within models is important for documentation and later reference. Additionally, with the capturing of this information into executable models, it is also possible to gain more insight to integrated systems due to the increased capability and performance of subsystem models.

For such a large vehicle as SLS, flight testing can be schedule- and cost-limited due to available budget and desired system capability on-dates. This does not reduce the need for confidence in the design being safe, reliable, and meeting performance capabilities prior to flight, particularly for a manned vehicle; instead, it places additional burden on pre-flight analysis to ensure a successful mission. These models thus become key anchors of vehicle integrated analysis and confidence in vehicle performance. The maturity of these grow throughout the program to match hardware design maturity and eventually are used in integrated vehicle simulation models with as much hardware as possible to generate evidence to demonstrate expected vehicle performance. Similarly, additional burden must be taken to ensure the validity and official approval of these models and their results. For SLS, this encompasses the Design Math Model (DMM) processes for model acceptance and release among program elements. A description of this process in detail with regards to the SLS GNC is provided in Emerson et al.¹

^{*}<https://www.nextbigfuture.com/2018/12/reverse-engineering-spacex-to-learn-how-to-speed-up-technological-development.html>

[†] <https://www.airspacemag.com/space/is-spacex-changing-the-rocket-equation-132285884/?page=1>

[‡] <https://www.engadget.com/2018/02/28/blue-origin-spacex-space-race/>

A major model used to assess the launch vehicle insertion capability, from a guidance and navigation perspective, is the Inertial Navigation System model. This software model mimics the inertial sensor behavior, onboard software processing, and navigation algorithms that result in an integrated navigation state---position, velocity, and attitude---that is used in closed-loop guidance and steering algorithms to meet vehicle-level performance requirements. The development, verification, and validation of the SLS INS model has been ongoing for several years. This effort has recently completed in advance of Flight Readiness Analysis Cycle and Design Critical Review activities leading towards the Artemis-1 mission.

GNC MODEL DEVELOPMENT RESPONSIBILITIES

The SLS Stages Element is responsible for integrating with the Prime Contractor, Boeing, for the design and procurement of component-and system-level hardware based on NASA-provided requirements. The inertial sensors used in the GN&C architecture for the SLS Block 1 configuration are the Redundant Inertial Navigation Unit (RINU) and the Rate Gyro Assembly (RGA). The Stages Element also has the responsibility of developing any necessary models of the hardware and embedded software of these GN&C sensors. These models would be of sufficient fidelity for integration into Hardware in the Loop (HWIL) laboratories as well into vehicle 6-DOF analysis simulations.

Vehicle Management (VM) is responsible for the vehicle Mission and Fault Management algorithms, GN&C design, and integrated performance assessment. A major component of this performance assessment for GN&C is based on vehicle 6-DOF simulation analysis. It also is the responsibility of VM to integrate the GN&C sensor models, provided by the Stages element, into any analysis tools—including 6-DOF simulation tools—used in the performance assessments of the GN&C design.

Through the Stages Element, the hardware vendor provided to the SLS Navigation Team detailed documentation and insight on the design and functionality of the RINU and the RGA. This documentation included information on the different sensor error parameters modelled in the Vendor's own analysis tools as well as details on the contribution of each of these error sources on integrated navigation performance. The Navigation Team also had access to Vendor test data with which to anchor the sensor error models and the performance of the integrated navigation model. The Vendor shared with the team information on the implementation of algorithms used in the inertial measurement processing as well as in the navigation state estimation. This insight into the internal software of the GN&C sensors also including details on the fault-detection capability of the hardware. Through these data deliveries, along with numerous technical data exchanges, the Navigation team gained a level of insight into the hardware and software design, internal and external data interfaces, system performance, and integrated testing of the avionics components allowed the Navigation Team to develop high-fidelity performance models of the navigation sensors.

The Redundant Inertial Navigation Unit (RINU) is a one-fault tolerant Inertial Navigation System that produces an integrated navigation state---position, velocity, and attitude---as well as angular rate and linear acceleration data for use in the vehicle control software. To support the one-fault tolerance, the RINU contains redundant accelerometers and gyroscopes as well as Fault Detection, Isolation, and Recovery (FDIR) software used to detect failed sensors and prevents data from failed sensors from being used for any output data. To initialize the navigation system, the RINU performs Gyrocompass Alignment (GCA) to estimate its orientation with respect to an inertial frame. To support the navigation function, the RINU software contains modules for performing high-rate compensation of the inertial data, computing estimates of gravitational acceleration, and for integrating the compensated incremental angles and incremental velocities into an estimate of the vehicle position, velocity, and attitude. The RINU also contains hardware and software necessary to support communications with the SLS Flight Computers over redundant MIL-STD-1553B data busses, receiving and responding to commands and data loads, including time synchronization to vehicle time performing hardware built-in tests.

One key part of the RINU performance model is the sensor error model. Since the RINU contains redundant sensors, the Navigation team simulated each individually to reflect the geometry of the hardware sensor configuration. Stages and the hardware vendor provided NASA with a detailed characterization of the sensor errors and their impacts on the RINU's navigation performance. These were used to identify key performance metrics of the raw sensor capability and provided scope for primary drivers.

The hardware supplier also provided detailed descriptions of the algorithms used in the RINU in form of Software Description Documents (SDD). The level of detail in these software descriptions allowed the Navigation Team to develop high-fidelity models of various modules in the RINU software. Primarily, the team implemented models of the RINU's Gyrocompass Alignment filter, models of the inertial navigation algorithms, gravity model, and the high-rate compensations---coning and sculling---of the $d\theta$ s and dv s used in the navigation algorithms. Additionally, the team developed a detailed model of the FDIR algorithms used to detect and isolate failed sensors.

As a part of the RINU model, the Navigation Team developed a model of the RINU's capability to detect and isolate failed sensors. Specifically, the minimum number of sensors required to produce measurements in orthogonal axes is equal to the number of degrees of freedom for orthogonally arranged sensors; for sensors in a skewed configuration, the minimum is equal to the number of degrees of freedom plus one. The minimum number of sensors required to detect a dissimilar measurement in asymmetric (skewed) arrangement is equal to the number of degrees of freedom plus an additional sensor to resolve the measurement and another additional sensor to create an over-determined system capable of fault detection.

The RINU model is used in a variety of stand-alone and integrated 6-DOF tools for to develop several analysis products. The RINU model is integrated into MAVERIC, the vehicle 6-DOF simulation used at MSFC. Integrated GN&C performance is assessed with Monte Carlo runs of MAVERIC. For each vehicle configuration, the GN&C ascent performance is documented in the various volumes of SLS-RPT-080, Space Launch System Program (SLSP) Integrated Guidance, Navigation, and Control (GN&C) Performance Assessment.

Additionally, MAVERIC results are used to anchor our assessment of Hardware in the Loop (HWIL) test data. Operating as a stand-alone tool, the RINU model was used to perform an offline gyrocompassing analysis outside the MAVERIC simulation. The results of this analysis are used to initialize the RINU navigation model in the vehicle 6-DOF. When performing a large Monte Carlo set, a significant amount of computation time is saved by not having to wait on the RINU model to perform GCA in preparation for ascent flight. The RINU model is also used in developing the flight-specific set of parameters used in the RINU FDIR algorithms. Additionally, the model was used to determine the sensitivity analysis that was performed to determine the sensitivity of RINU FDIR thresholds and false positive failure probability due to increased sensor noise, flexible body dynamics and other un-modeled dynamics.

The Rate Gyroscope Assembly (RGA) is a set of three pairs of mutually orthogonal gyroscopes. The RGA samples the gyroscopes internally at a high rate, filters the measurements, and outputs filtered angular rate data on MIL-STD-1553B busses. While the RGA is internally redundant, each pair of rate gyros operates independently. The consolidation or down-selection of the angular rate data across the three channels is handled in the SLS Flight Software on the FCs. As such, the Navigation team chose to represent a single channel of RGA sensors and data processing in the model. Multiple instances of this single channel model were used to simulate outputs to match the hardware flight interface.

MODEL VERIFICATION APPROACH

Verification of the RINU and RGA performance models (DMM-CS-0411 and DMM-CS-0410) was performed according to the V&V requirements in Paragraphs 3.15 of NASA document SLS-RPT-080 Vol 3 Revision A, "Space Launch System Program (SLSP) Design Model Delivery Standard"², which states:

"Model Verification is the process of determining that a computational model accurately represents the underlying mathematical model and its solution from the perspective of the intended uses of the M&S [Model and Simulation]. It includes a set of input/output model test cases that can be used to verify that a model has been implemented correctly, and that span the range of validity of the model. Verification measures the correctness of (a) the mathematical understanding of the underlying physics, (b) the translation (i.e., program code that does the computational effort) of the physics-to-mathematics into a model, and (c) the effects and sensitivities of dependent and independent variables on other variables of the model, so that results are indicative of the hardware/system performance."

The Navigation Team developed several criteria in the verification of the RINU and RGA simulation models. These criteria included:

1. Trace of stochastic error parameters and modeling to vendor-provided documentation
2. Unit tests of individual error model implementations
3. Trace of algorithms and software used to the vendor documentation
4. Trace of isolator and latency characteristics to vendor-provided documentation
5. Monte-Carlo sets with the simulation model to quantify performance. For the RINU specifically, these Monte Carlo sets included both failed and non-failed sensor configurations.
6. Comparison with vendor analysis for RINU FDIR component of model.
7. Comparison with vendor performance data from the simulation used by the vendor to verify accuracy requirements.
8. Verification of frequency domain behavior of model components which affect the sensors' frequency domain.
9. Comparison with vendor analysis from Software Flight Qualification Test (FQT).

While there was no requirement for the Navigation team to independently verify the RINU and RGA models, members of the team that performed the V&V activity were not the same individuals who developed and implemented the model. As the V&V team discovered issues with the model, the Navigation Team peer-reviewed the findings and decided on actions necessary to resolve the discrepancy. The model developers would then implement the solution and present evidence to the V&V team. All V&V issues, along with dispositions were documented in the Verification Report that is delivered with the models.

For the individual sensor models in the RINU and RGA component models, the Navigation Team performed a code inspection to verify the correct implementation. The code inspection was accompanied by low-level unit tests to confirm the expected behavior of the model. For each function in the model, e.g., gyro and accel bias, scale factors, random walks, a test was developed to comparing the output of the component model to an analytically derived result. The data parameters used as inputs to the model were compared against vendor-provided to ensure that the error sources were modeled consistent with the values used by the vendor for verifying accuracy requirements. For error parameters used in the RGA model, comparisons were made to Stages-leveled requirements as well as to vendor-supplied capability estimates. For the RINU, the error parameters were compared back to the vendor-developed Navigation Error Budget which documented the vendor's analysis of the individual gyroscope and accelerometer error terms.

The Navigation Team performed a thorough inspection of the RINU model code compared to the vendor software descriptions. The Navigation V&V team annotated the model code with references to the vendor documentation. This allowed the team to trace, line-by-line, the functionality of the model code to RINU software documentation. Any discrepancies in the model were reported to the Navigation team for review and disposition. The model developers implemented the required changes. All modeling discrepancies were documented in the model verification report that is delivered along with the model.

To verify the navigation performance of the RINU model, the Navigation team performed several Monte Carlo sets, flying the RINU along a specified reference trajectory. These simulation runs were performed with and without sensor failures. The 1-sigma navigation errors from these analyses were compared to the capability estimates of the RINU navigation performance listed in the Navigation Error Budget. Additionally, the vendor supplied NASA with addition sets of Monte Carlo data used during Formal Qualification Test (FQT) of the RINU software. Comparisons of the one sigma navigation errors showed favorable results. In addition to verifying the performance of the RINU model by comparison to vendor test data, several analytical tests were performed by the Navigation team. Since heading error can be analytically predicted, the RINU model was run with a simulated East angular velocity bias. The RINU model's heading error tracked reasonably well with theoretical predictions. Assuming perfect sensors, the RINU's ability to track true attitude were verified simulating gyrocompassing under with no error sources enabled. A Schuler test was performed with the RINU model to verify that the navigation code in the model performed as expected when navigating in-place.

Frequency Response Verification

The Navigation team verified the frequency response of the sensor models by an analysis of the component models that effect the frequency content of the sensor model outputs. The team used vendor-provided data on the transfer functions that model the responses of the chassis and isolators to perform a code inspection the implementations of the models of the mechanical interfaces. Then Bode diagrams were used to verify the implementation of the filters matched vendor data. For each of the software components that effect the phase and gain response of the sensor outputs, the Navigation team used vendor-supplied data on the responses of each of filter in the sensors software. Code inspection showed that the models of filters were consistent with vendor documentation. Similarly, Bode diagrams demonstrated that the modeled and predicted responses were very similar. Where differences in the vendor's and the modeled responses differed, the Navigation team demonstrated the model produced a more conservative response—less attenuation but with more phase lag then the vendor analysis.

FDIR Verification Activities

Due to the unique sensor redundancy scheme, sensor failures have an impact on RINU performance and as a result, the modeling of the sensor pentad and FDIR logic has been included in DMM-CS-0411. Over the course of approximately a year beginning in mid-2013, the DMM-CS-0411 development team worked with the RINU hardware vendor to model RINU FDIR to a level of fidelity suitable for simulating RINU behavior and for FDIR threshold design. The Navigation team implemented the RINU FDIR logic based on the RINU Software Description document supplied by the vendor. The team regularly discussed the FDIR Implementation, configuration, and analysis-related details with the RINU vendor at various technical teleconferences. The primary analyst on the vendor side was brought in as a consultant to design FDIR thresholds for the RINU program and to discuss implementation details with the RINU model developers on the Navigation Team. The primary analyst had previously designed the FDIR thresholds for the heritage FTINU design for ULA's Atlas V EELV. In December of 2013 at a Navigation Technical Interchange Meeting the vendor's facility, the MSFC team presented the initial set of FDIR verification results vendor. A number of discrepancies were discovered, tracked, and subsequently resolved resulting in the presentation of detailed verification results at a follow-up Navigation TIM in May of 2014. At the end of the TIM the RINU vendor verbally concurred with the model developers' implementation of RINU FDIR, accuracy of results, and showed confidence in the transfer of knowledge required to design RINU FDIR thresholds.

MODEL VALIDATION APPROACH

For the purposes of this paper, validation analysis focuses on the comparison of the model to flight hardware. The intent of this analysis is to provide additional insight outside of algorithm and specification documentation. Similarly, this testing allows for an independent check on the assumptions used in interpreting design artifacts. The testing herein focused on the areas of performance that the GNC system is particularly sensitive to. The focus is on several key areas: demonstrated ability to gyrocompass under external dynamics, validation of noise characteristics, frequency response testing, integrated system latency, and validation of navigation performance between the model and a hardware development test unit. All of these tests utilized a flight-like hardware unit with actual sensors and flight software. This unit was produced in advance of flight units to allow for early testing and characterization. The primary difference between this and actual flight units is the durations and levels involved in acceptance testing at the vendor level. Due to its purpose as a test unit with limited exposure to high vibration or thermal variations, reduced testing was acceptable to meet the needs of the test events. Although the environmental testing was limited, the unit was still put through testing to establish its performance within requirements, focusing on calibration and navigation accuracy. As such, this platform provides an ideal platform for validation activities.

The validation testing suite utilized this flight hardware within controlled environments and provided targeted excitation to assess specific elements. These areas of focus can be captured in simulation and design documentation but are considered essential to the integrated GNC performance. For example, the key sensitivity of the navigation performance is to the initial attitude error at the end of gyrocompassing. As such, the ability of the unit to gyrocompass is critical. While the detailed models of SLS do enable a simulation of on-pad dynamics under various wind conditions, further hardware testing helps provide additional insight to

hardware robustness and allow for comparing the response to flight-like as well as worst case on-pad dynamics. Similarly, to meet the stability requirements of the vehicle, the controls algorithms are designed around the sensors' dynamics response. This includes both RINU and RGA frequency responses. Validation of these through hardware testing provides the additional confidence needed in assessing control margins and vehicle stability. Similarly, to meet insertion conditions, the onboard guidance algorithms are sensitive to integrated system latency, particularly at engine shutdown. Due to the high velocity of the vehicle, this latency must be taken into account when commanding this event to ensure the proper state is achieved to meet insertion requirements.

Gyrocompassing Alignment Testing Insights and Results

The first validation test performed focused on testing the RINU's ability to gyrocompass under simulated on-pad dynamics. This testing is detailed in Geohagan³, but is summarized here to provide context to the overall effort. For this test, the RINU was mounted onto a large 6 degree of freedom (DOF) table at MSFC and was gyrocompass'ed through a variety of dynamics. The table allowed for comparison between response of the flight hardware and the RINU model to a common set of dynamics. This included both the dynamics used in vendor specification and analysis expected on pad dynamics, as well as scaled up on-pad dynamics to validate robustness under much larger than expected on pad motion. Figure 1 below provides a picture of the RINU mounted onto the table as well as a close-up image of the unit under test.

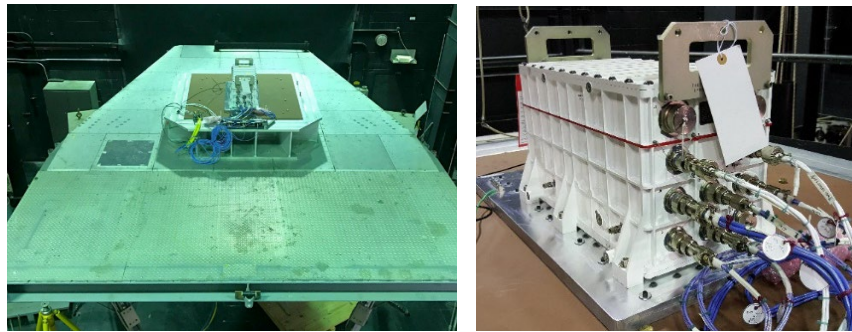


Figure 1. RINU and 6DOF Table (Left), Close-up View of Integrated RINU (Right)

In addition to validating the performance of the algorithms between the software and hardware models, this testing provided additional insight into several integration constraints, which caused some limitations in post-processing. Two areas of limitation in the test is due to having a multi-rate navigation systems, which is typical for integrated navigation boxes and laid out by sources such as Savage⁴. Based on the interface of the unit, only the low rate navigation data was available. As such, detailed validation of the sensor error models and processing was reduced. However, this data could be used to enable detailed checks of the onboard navigation software. This testing focused on validation of the implementation of the gyrocompassing algorithms in the RINU model. The other limitation of the test setup was manifested in the form of missed samples. Due to asynchronous nature of the 1553 bus, there is a potential for missed samples of data. While this does not affect in-flight performance since the latest state is always captured by flight software, this does limit the ability to re-create the entire software processing loop within the software model. For example, in Figure 2, missed samples can be seen to cause divergence between the unit under test and the software model. This plot shows the difference between the two solutions. In order to correct this, an interpolated data point was inserted to enable better matching performance, but an exact comparison was not able to be achieved. Even with this limitation though, the algorithms were able to be validated with this testing within expectations.

In addition, this test was used to capture static data over a long (24-hr) duration to enable noise characterization activities. Analysis of this testing provided insight into the readout noise of the RINU's sensors. The noise of the RINU model showed lower levels than captured with the actual hardware during the static testing. Analysis of this effect led to a model correction in terms of how noise is modeled and highlights potential areas for ambiguity in a sensor specification. For example, for an integrated navigation unit with multi-rate processing, the frequency commiserate with the noise specification is fundamental to the modeling

of the unit. In this scenario, the modeling of the noise relative to other onboard processing and other box-internal effects significantly changed the observed content. To capture this additional noise, the readout noise was increased to provide a better match. This update was then used in Monte Carlo analysis to validate integrated performance against the vehicle requirements. Figure 3 provides a comparison of the noise levels before and after this update with comparison to hardware testing results. The results of this had limited impact on insertion capability, but do have an effect on systems requiring tight attitude for such activities as docking to co-manifested payloads as in the Block 1B vehicle design⁵. Because of the value from this test, static testing was extended to 36 hours and repeated for the RINU and RGA in later testing. The results of this testing is include in the next section.

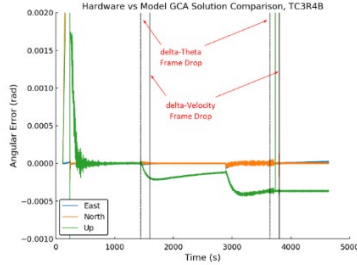


Figure 2: GCA Attitude Comparison w/ Frame Drops

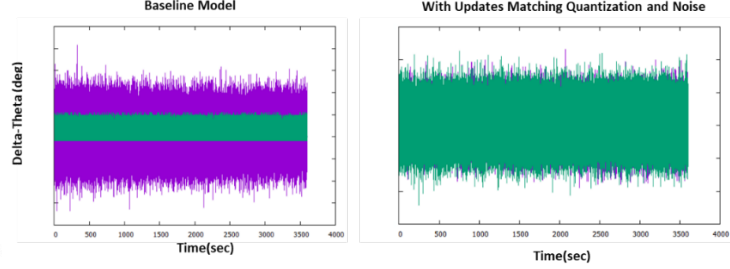


Figure 3: Noise Dynamics Before and After Update

Static Noise Characterization

A series of static and frequency response tests were performed during June 2018 to characterize the performance of the RINU and RGA inertial navigation sensors. Using data obtained from the static tests as input, an Allan Variance analysis was performed to determine the type of noise present in the hardware and to estimate the standard deviation needed to characterize these types of noise in the sensor simulation models. Allan variance is a time-domain analysis technique in which a data stream is divided into clusters of specified length, the average of each cluster is taken, and a two-point sample variance is calculated from the successive cluster averages⁶. Put simply, the Allan variance is a method of representing the root mean square (RMS) random-drift errors as a function of averaging times⁷.

Defined in terms of gyro output angle, the Allan variance can be calculated using the following equation:

$$\sigma^2(\tau) = \frac{1}{2\tau^2(N-2n)} \sum_{k=1}^{N-2n} (\theta_{k+2n} - 2\theta_{k+n} + \theta_k)^2 \quad (1)$$

where τ = averaging time, N = sample size, n = cluster size, θ = accumulated angle

Defined in terms of accelerometer output, accumulated velocity is substituted for accumulated angle. When plotted in logarithmic scales, the Allan Variance can discriminate different contributing error sources by simply examining the varying slopes on the Allan Deviation plot, as illustrated in Figure 3. This is achieved by relating the Allan Variance expressed in the frequency domain to the power spectral density of various error sources, so that these errors may be expressed in various powers of correlation time⁶.

Body rate and acceleration output obtained from the 36 to 48-hour static tests of the RINU and RGA inertial sensors was used as input into the Allan Variance analysis, and the noise present in the test data was identified in an Allan Deviation plot. This analysis was also performed on data from a 36-hr simulated static test of the RGA sensor model. Assuming statistically independent error sources, the Allan Variance can be expressed as a sum of these terms. Ignoring the sinusoidal and exponentially correlated noise terms for now, the Allan Variance is then of the form:

$$\sigma_{total}^2(\tau) = \sigma_Q^2(\tau) + \sigma_{rw}^2(\tau) + \sigma_B^2(\tau) + \sigma_{rrw}^2(\tau) + \sigma_{rr}^2(\tau) \quad (2)$$

$$= C_Q \tau^{-2} + C_{rw} \tau^{-1} + C_B \tau^0 + C_{rrw} \tau^1 + C_{rr} \tau^2$$

The method chosen to extract the noise terms from the Allan Variance consisted of a weighted least mean squares approximation:

$$\mathbf{C} = (\mathbf{V}^T \mathbf{W} \mathbf{V})^{-1} \mathbf{V}^T \mathbf{W} \sigma_A^2 \quad (3)$$

where the Vandermonde matrix $\mathbf{V}$ and weighting matrix $\mathbf{W}$ are given by:

$$\mathbf{V} = [\tau^{-2}, \tau^{-1}, \tau^0, \tau^1, \tau^2] \quad (4)$$

$$\mathbf{W} = \text{diag}(\sigma_A^2)^{-2} \quad (5)$$

The error coefficients for individual noise terms are then computed from the vector of least squares coefficients $\mathbf{C}$ as follows:

- Quantization Noise $Q = \sqrt{C_Q/3}$
- Angle/Velocity Random Walk $N = \sqrt{C_{rw}}$
- Bias Instability $B = 0.6648 \sqrt{C_B}$
- Rate/Acceleration Random Walk $K = \sqrt{3C_{rrw}}$
- Rate/Acceleration Ramp $R = \sqrt{2C_R}$

The Allan Deviation plot for the RGA hardware test and simulation model are shown in Figure 5. In the Allan Deviation for the hardware tests, there is clear evidence of sinusoidal noise on the left of the plot, and exponentially correlated noise in the peaks located just right of center. The sinusoidal noise appeared with greater amplitude and lower frequency in the simulation model. The greater prominence of sinusoidal noise in the model is likely due to the higher level of quantization in angle output, an order of magnitude greater than what was computed in the Allan Variance analysis for hardware, and the resulting interaction with internal filters.

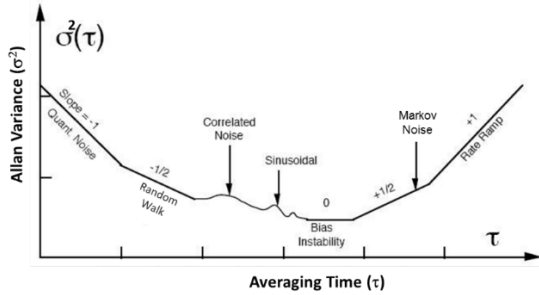


Figure 4: Typical characteristics of Allan Variance curve. Different contributing error sources are distinguished by different slopes along the curve.

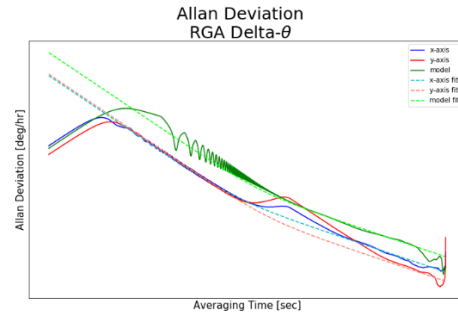


Figure 5: Allan Deviation plot for RGA hardware and simulated static test. The implementation of quantization induced sinusoidal noise in the Allan Deviation for the RGA model.

Approximate values for quantization noise, angle random walk, and bias instability were computed for the RGA hardware. The characteristic slopes for rate random walk and rate ramp did not appear for the 36-hour static test. The higher trend line for the model in the Allan Deviation plot indicates that the errors observed in hardware fall within the modeled error budget.

Frequency Response Testing

ATA Engineering, Inc., was contracted to perform Frequency Response tests (FRTs) on the RINU and RGA to provide the frequency response functions (FRFs) needed in verifying and validating the inertial

sensors used in the SLS GN&C system. Input dynamics describing base motion during the FRT were measured by six linear variable differential transducers (LVDTs) and eight accelerometers. An image of the test setup for the RINU taken from ATA's final report⁸ is shown in Figure 6.

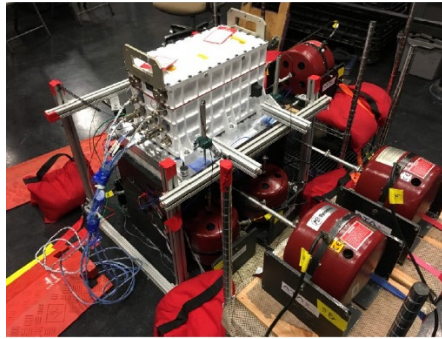


Figure 6: Overall view of test setup and test facility for the RINU.

RINU and RGA sensor data was logged on a separate system, so the FRT input and output data had to be time-aligned using a combination of digital signal processing and manual adjustment. Data obtained from the FRTs were also used to validate the inertial navigation sensor models. The input dynamics from the test were run through the navigation sensor simulation models in order to assess whether the model outputs provide a valid representation of the hardware. According to ATA's report, the LVDT measurements were most accurate at low frequencies, because displacements at higher frequencies became too small to measure accurately, while the accelerometer measurements were more accurate for high frequencies but experienced ringing in the lower-frequency range. Given these characteristics, the LVDT sensor data was used for generating FRFs below 2Hz, and the accelerometers were used for frequencies above 2Hz to reproduce the test inputs as accurately as possible. All FRFs were approximated using Fast Fourier Transformation (FFT) analysis.

Baseline FRFs for the navigation sensor models were generated using a combination of simulated continuous and stepped-sine inputs with modeled sensor errors disabled. The baseline response for the RGA is pictured in Figure 7, overlaying the design envelope. The baselines were compared against FRFs generated using the LVDT and accelerometer inputs with the models' sensor errors enabled. The model responses to test inputs were then compared against a reconstruction of the observed FRT response to these inputs as measured by the navigation sensors. Because the test data had been time-aligned, a constant time delay between input and navigation sensor output had been effectively removed. To aid in the comparison of test data and modeled frequency response, a constant delay was added to the phase response for each test run. The comparisons between the model response and the observed FRT response for one of the RGA gyros is displayed in Figure 8.

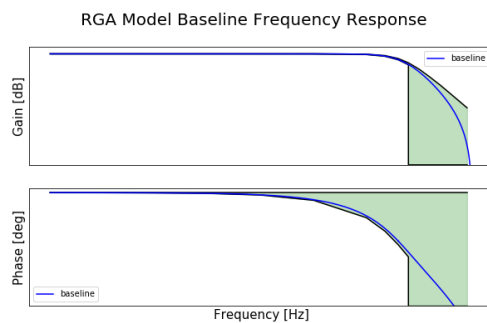


Figure 7: RGA model baseline frequency response and design envelope

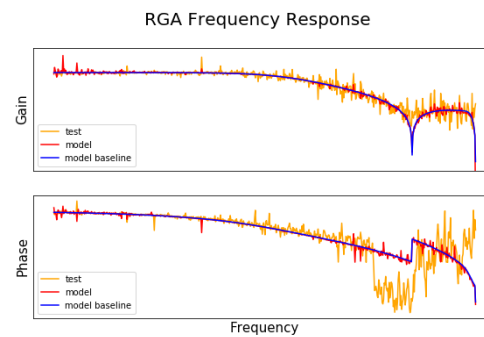


Figure 8: RGA hardware and model responses to FRT test input. Phase response in hardware test cannot be tracked accurately at the lower gains in high frequency region

In general, the FRF reconstructions demonstrated similar gain responses between the RINU and RGA hardware and their corresponding simulation models. The phase could be matched up reasonably by reintroducing a constant delay into the hardware response, though the phase could not be tracked well in the higher-frequency regions where gain begins to roll off. Overall, the precision with which FRF reconstructions of the hardware response could be produced appeared to be limited by physical limitations of the shaker table, particularly at low frequencies, the need to protect the hardware from excessive environments, and accelerometer or displacement sensor calibration errors.

Development Test Unit/Integrated Comparisons

Analysis of stand-alone hardware testing and detailed modeling is very important in being able to predict vehicle performance. Most hardware testing is performed with a test set interface with limited non-flight conditions, and usually performed with an engineering development unit (EDU). The next step in hardware testing is system integration testing, using HWIL, which plays an important part in the full testing program. For the SLS program, HWIL testing is done with a RINU development test unit (DTU), which has the same parts and software as a flight RINU, but with the sensors replaced with a PAM card, which provides the unit with an external interface to a dynamics simulation. The dynamic simulation (ARTEMIS) provides dynamics, actuator and some sensor modeling independent of MAVERIC.

But the benefits of HWIL testing is not just to ensure the hardware and software all interact correctly together, although that is also a critical need and invaluable risk mitigation tool. For us, this test environment provided opportunities to analyze, validate and improve our modeling of the sensor, system and how we evaluate them. Evaluating the performance of the HWIL sensors allowed us to validate and better understand the latency of the sensors. By using the ARTEMIS system, we were able to inject temporary biases into the sensors and measure the latency of the sensor response, using a bus monitor of the 1553 traffic. Because the RINU and RGAs both have internal filters that were not part of the latency budget, we had to compensate for that by analytically removing the filter response to determine the net latency. Latency is important for control system stability, but critical for affecting orbit injection and determining when to shut down the engines. With multiple runs, we were able to accurately measure actual latency and compare it with hardware specifications and expected variations due to asynchronous operation and variability.

Previous latency measurement methodologies at the system level focused on a method showing a phase lag response from the controller by injecting sinusoidal biases. Due to the system noise and sensitivity of the controller, as well as the 20 millisecond (msec) quantization of the controller process, preliminary testing in MAVERIC showed that this method would not be possible to resolve latency levels accurate enough to validate latency specifications. Thus, a new method was developed using larger, short duration, square wave biases that caused only temporary disturbances to the controllers and allowed much finer (<1 msec) measurement of the latency, as well as step-wise change evaluation through the flight computer and actuator commands at a system level. Figure 9 below provides a sample of the data used to perform the measurements/analysis.

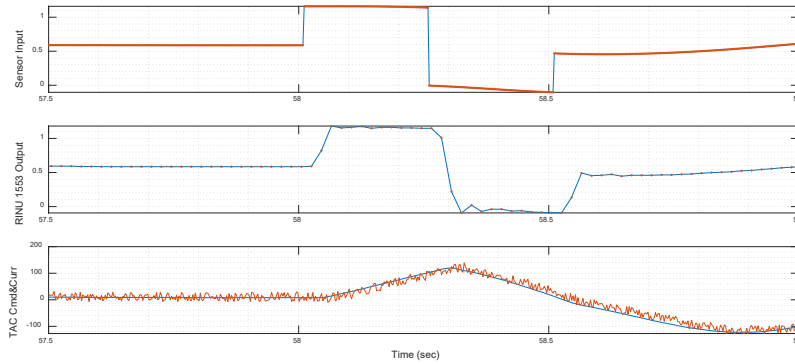


Figure 9: Time History Data of Induced Dynamics to Assess Latency at Input (Top), Sensor Output (Middle), and Actuator Output (Bottom)

This effort also involved spending time to better understand the internal working of the sensor and how it applied to latency that hadn't previously been studied to the same level of detail, and comparing this to the MAVERIC models. This resulted in a correction to the RINU navigation timing model where a time correction had been incorrectly applied. It also resulted in an RGA latency estimation change, even though the RGA is only a model in the hardware in the loop simulation, the additional focus and difference in what was used in MAVERIC prompted questions of the hardware vendor, which showed the latency specification included latency that was part of the 100 Hz update that was already modeled and thus was too large. The testing validated all other latency related values being used in the performance simulation.

The process of evaluating the sensors at the 1553 data rather than output from the flight computer, especially due to the large temporary biases used to determine latency, discovered that there were cancelling alignment transformation errors in the flight computer software and the ARTEMIS transformation to the RGA frame. This went undetected through other testing that focused on flight software output and simulation SLS frame states, rather than sensor states. While this did not affect the detailed modeling in MAVERIC (which was shown to be correct), this allowed us to discover and correct a flight software error that may not have been discovered until much later in the program, or not detected until flight, resulting in a catastrophic system failure. This provided further evidence of the importance of analyzing data at every interface and not just an end-to-end analysis.

One of the critical outputs of the RINU is the integrated navigation state during flight and the corresponding timetag that represents the time of validity. This data is used on SLS for both guidance and in determining the time to start engine shutdown, and is a primary contributor to orbit injection accuracy. The latency testing and analysis pointed out one error in the MAVERIC correction, where a correction to the timetag due to an internal RINU software correction was not being performed.

The comparison of the System-Integration-Lab (SIL) RINU DTU output, which includes the ARTEMIS simulation of the vehicle dynamics and actuator interaction to MAVERIC was also performed to validate the SIL RINU DTU navigation states were accurate enough to allow other system requirements verification using the flight software and avionics EDUs. This comparison has been used to identify several ARTEMIS issues during development. As the ARTEMIS models were corrected and improved, and the navigation errors and orbit insertion accuracies became closer, more accurate comparisons to MAVERIC outputs were performed. While it is understood that there are small inherent differences that the SIL introduces, we endeavored to understand what these differences were, and how much of a difference these made. In the process of doing this, the SLS navigation team provided insight into further differences. When a comparison of the improved ARTEMIS and RINU/DTU navigation state error (time aligned, navigation state versus dynamics model "truth" state for position, velocity and attitude) appeared to show better performance than the MAVERIC simulation was showing.

The detailed analysis into both the ARTEMIS and MAVERIC models provided additional validation of good correlation. It also brought focus onto the process of calculating the navigation error, which is simulation-only calculation (truth states of actual flight are not available, although post-processing using upper stage GPS data and ground tracking data can provide a better estimate for post-flight analysis). A comparison of the ARTEMIS team navigation error and navigation/MAVERIC team allowed us to discover minor errors on both sides. The ARTEMIS data did not provide propagation between the RINU navigation timetag based states and the states of the simulation truth. The simulation truth states are updated at 2.5 msec, and these steps are broken further into 1200 Hz incremental angle and velocity changes to feed into the RINU DTU. The states of the RINU are internally filtered, and the timetag also corrects for expected internal latencies, resulting in timetags that are not aligned closer than 2.5 msec. However, the expected accuracy of this results in small (<50 feet) errors. The errors between the two simulations was still several hundred feet. Independent determination of the navigation errors that performed separate time alignment and propagation for both simulations showed performance much closer than that shown in MAVERIC, which had previously been evaluated and determined to be accurate. This additional analysis provided the direction to evaluate the navigation error performed in MAVERIC. This resulted in two error findings. One was an error using the timetag from an incorrect RINU subaddress. This error was actually introduced due to the high level of modeling of the multiple, complex RINU outputs with similar mnemonics. Once discovered, it was also found that this time-tag mnemonic error propagated into the state used for guidance and engine shut-down determining. Correcting it improved the orbit injection accuracy.

A larger magnitude error was also discovered in the navigation error calculation that is performed each navigation step in the MAVERIC simulation. While this error did not affect injection performance, it materialized as a larger than expected navigation error. When the error was corrected, it resulted in much improved comparative results with the SIL results, as shown in Figure 10. The larger attitude errors in the SIL is traced directly to the 2.5 msec asynchronous operation of the ARTEMIS simulation and DTU, coupled with high roll and pitch-over maneuvers performed shortly after lift-off.

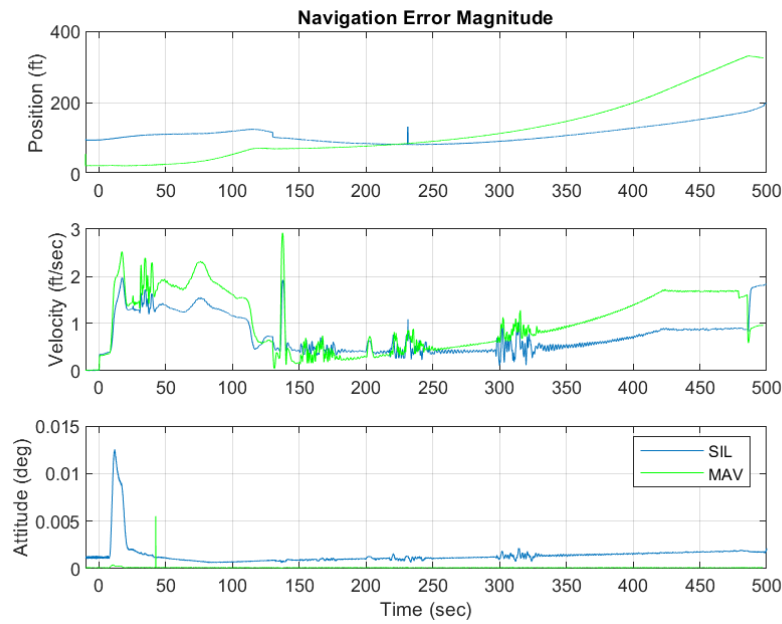


Figure 10: Navigation Error Comparisons between SIL and MAVERIC Models

CONCLUSIONS AND FUTURE WORK

This paper provides a summary of the approach taken by the SLS Navigation Team for verification and validation of sensor models in support of the SLS program. This extensive effort provided a high level of

confidence in understanding and assessing sensor and integrated performance across a wide variety of mission scenarios. The investment in this analysis results in mature analysis with a strong tie to hardware results, enabling a large amount of understanding and insight into the design for flight. Validation of the models prior to flight is a necessary step to assure key stakeholders and engineering management of the vehicle's capability in the path to flight. During previous design cycles, the INS model has been used to assess on-pad alignment under expected environments, in-flight navigation accuracy, and response to fault events⁹. Similarly, the RGA models have been used to validate controller response and stability to in-flight disturbances and have been used extensively in developing fault detection algorithms¹⁰. Additionally, this model was used to develop Mission and Fault Management criteria that drive autonomous vehicle operation. Lastly, the frequency response testing has informed SLS Control algorithm design and stability assessments, helping to anchor those results with hardware data.

Over the course of this analysis, several key items were identified that should be noted for future and similar efforts. It is important to be careful about readout noise specification and the rate at which that noise is applied in modeling. Hardware testing can easily validate this and help to match the model to hardware. While time-consuming and data-intensive, Allan Variance assessment of the sensors under static conditions proved very useful, allowing for a statistical assessment of the error behavior. Similarly, it is important to maintain focus on purpose of models, such as an interface emulator or a performance model. But it is also important to maintain an awareness of the sensor ICD and ensure matching of the sensor outputs ahead of FSW integration and testing, i.e. maintaining an output structure matching the ICD in terms of data content rather than specific message structure such as modeling all health and status information. During analysis for the vehicle, integrated latency became very important to characterize and understand due to its impact on controller lag and insertion accuracy. This testing of an integrated system can only be performed in a hardware-in-the-loop environment allowing inclusion of FSW and integrated flight hardware.

The biggest enabler of this effort is the large amount of integration and support between the NASA developers, the prime contractor, Boeing, and the hardware vendor, Honeywell. This team was well integrated throughout the hardware development and model integration process with Boeing and NASA Stages being the primary customers of the model. This model was used by NASA VM in performance assessment. As such, reviews after each test event and integrated validation and verification activity included key personnel across each organization to provide confidence and insight into the model and hardware. The amount of documentation provided enabled a large amount of insight into performance of the units and its capability. Additionally, use of data from Acceptance and Qualification testing is also a great asset, although its use may be limited if representative truth data is not available. A key lesson of this work is to focus on hardware testing to fill in known gaps and provide confidence in areas of high sensitivity. Due to flying heritage-based sensors, the hardware tests were designed to capture insight into expected performance in focused areas. As such, a flight-like development unit with live sensors was critical for GCA testing and detailed static analysis. A particular interest of these sensors is their frequency response at low frequencies. This proved difficult to assess, due to having to induce low frequency dynamics at low magnitudes to limit aging of the units. Even with this, the results still provided key insight into confirming the overall response and behavior of the units under test.

Another key takeaway from this work is the need for oversampling, particularly in the case of asynchronous hardware interfaces such as 1553. This problem exhibited itself in gyrocompassing testing as missed data frames and limited the ability to perform cross-verification between the model and hardware. To address this issue, the frequency response test utilized a higher sampling rate than the standard output of the unit under test to ensure no missed samples. While this did cause increased complexity in post-processing software, having the data on-hand is well worth the effort. This oversampling isn't needed in SLS Block 1 flight due to the vehicle always having access to the latest navigation state data, since the sensor observations are integrated within the RINU, rather than in SLS FSW.

The navigation team is taking these lessons learned and applying them to model development, verification, and validation for new sensor models needed for the B1B EUS, particularly GPS. A key area of difference is in terms of onboard software. For GPS receivers, the performance can ultimately be traced back to the tracking loops internal to the device. This is typically implemented in digital logic as opposed to typical software. To model these in real-time requires running the model at a very high rate in order to model the

high frequency radio signals involved. As such, the scope of modeling will need to be balanced between the needs of performance capability and runtime concerns in an integrated simulation. Conversely, through the use of RF simulators, extensive testing can be performed on flight-like units to provide a high level of insight into the operational characteristics of the device. Similarly, for a multi-sensor system, integrated system level verification and validation is necessary to validate performance and ensure operational capability. The lessons learned from the verification and validation has provided a large amount of benefit. The final result of this effort, the RINU and RGA models, have been tested to a high level of confidence prior to the first SLS flight, and allow for extensive analysis and development for future designs. Post-flight, these models will be reviewed to validate how well they match during flight conditions and provide further confidence. It is the hope of the authors that as the industry moves towards an ever increasing usage of software models and simulation to develop confidence and performance assessment, these lessons learned and approach can provide a starting or comparison point for other missions.

ACKNOWLEDGMENTS

The work included herein represents multiple years of effort and the contribution of several previous SLS navigation team members must be acknowledged. The majority of the work supporting gyrocompassing analysis was completed by William Bernard, Kevin Geohagan, and Dennis Strickland. Similarly, the frequency analysis testing was conducted under contract by ATA Technologies, whose results were used to in frequency response testing. NASA Flight Software and GNC personnel were essentially to this testing and analysis of the results, particularly Jeff Brouwer. The development team and program staff at Honeywell and Boeing were instrumental in providing the documentation, data artifacts, and test hardware used in the testing of the RINU and RGA units. Similarly, the team would like to acknowledge the support of SLS VM and Stages leadership, principally Joey Broome and Lisa Blue for their support of these hardware testing validation. Lastly, the team would like to acknowledge the former team lead, Emerson Oliver, who helped plan, acquire funding, defend, and execute these testing activities.

REFERENCES

- ¹ Oliver, T.E., et al, "SLS Model-Based Design: A Navigation Perspective," Proceedings of the AAS Guidance and Control Conference, Breckenridge, CO., 2018.
- ² NASA/SLS Program, "Space Launch Program Integrated Guidance, Navigation, and Control Performance Assessment, Volume 3: Appendices C-G, Verification, Validation, and Test", National Aeronautics and Space Administration, 2014.
- ³ Geohagan, K.W., Bernard, W.P., Oliver, T.E., et al, "6DOF Testing of the SLS Inertial Navigation Unit", Proceedings of the AAS Guidance and Control Conference, Breckenridge, CO., 2018.
- ⁴ Savage, Paul G. "Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms." *Journal of Guidance, Control, and Dynamics* 21.2 (1998): 208-221.
- ⁵ Oliver, T.E., Park, T., et al, "Space Launch System Block 1B Preliminary Navigation System Design", Proceedings of the AAS Guidance and Control Conference, Breckenridge, CO., 2018.
- ⁶ Ng, L. C.. "On the application of Allan variance method for ring laser gyro performance characterization (No. UCRL-ID-115695)", Lawrence Livermore National Lab., CA (United States), 1993.
- ⁷ El-Sheimy, N., Hou, H., Niu, X. "Analysis and modeling of inertial sensors using Allan variance", *IEEE Transactions on Instrumentation and Measurement*, 57(1), 140-149, 2007.
- ⁸ ATA Engineering, "Final Report on NASA SLS RGA and RINU Frequency Response Testing Revision 1", ATA Engineering, San Diego, CA, 2018.
- ⁹ Anzalone, E., "Optimization of Second Fault Detection Thresholds to Maximize Mission Probability of Success", Proceedings of the AAS Guidance and Control Conference, Breckenridge, CO., 2018.
- ¹⁰ Park, T., Oliver, T.E., Smith, A. "Sensor Data Quality and Angular Rate Down-Selection Algorithms on SLS EM-1," Proceedings of the AAS Guidance and Control Conference, Breckenridge, CO., 2018.