AAS 18-131

SLS MODEL BASED DESIGN: A NAVIGATION PERSPECTIVE

T. Emerson Oliver,[*] Evan Anzalone,[†] Thomas Park[‡], and Kevin Geohagan [§]

The SLS Program has implemented a Model-based Design (MBD) and Model-based Requirements approach for managing component design information and system requirements. This approach differs from previous large-scale design efforts at Marshall Space Flight Center where design documentation alone conveyed information required for vehicle design and analysis and where extensive requirements sets were used to scope and constrain the design. The SLS Navigation Team is responsible for the Program-controlled Design Math Models (DMMs) which describe and represent the performance of the Inertial Navigation System (INS) and the Rate Gyro Assemblies (RGAs) used by Guidance, Navigation, and Controls (GN&C). The SLS Navigation Team is also responsible for navigation algorithms. The navigation algorithms are delivered for implementation on the flight hardware as a DMM. For the SLS Block 1B design, the additional GPS Receiver hardware model is managed as a DMM at the vehicle design level. This paper describes the models, and discusses the processes and methods used to engineer, design, and coordinate engineering trades and performance assessments using SLS practices as applied to the GN&C system, with a particular focus on the navigation components.

NASA is designing and building the Space Launch System (SLS), an evolution of launch vehicles to enable the next era of human exploration of space.[1,2] NASA is currently in the process of verifying and building the Block 1 Launch Vehicle and designing the next evolution Vehicle, Block 1B. In the design of these new launch vehicles, the SLS Program has implemented a model based design approach. This paper describes implementation of a program driven systems engineering and integration (SE&I) approach. The purpose of this paper is not to debate or interpret how MBD should be implemented, but rather to describe how it has been implemented by the SLS Navigation Team. This paper will identify the merits of this approach, and provide some key lessons learned.

SLS was built on the Constellation Program (CxP) but it is a very different program with more efficient design practices and a different SE&I approach. The SE&I approach is described in great detail by Hutt et. al.[3] SLS emphasizes efficiency in requirements development and decomposition. As evidence of this, the Ares I project had five levels of requirements decomposition between the requirement for a launch vehicle with a specified insertion target and the inertial hardware specification. In contrast, the SLS program equivalent includes three levels with an increased level of design insight. Between the two programs, entire levels of functional requirements were decided to be unnecessary and thus were eliminated in the implementation of the SLS Program. Dissimilar to CxP and the Space Shuttle Launch Program (SSLP), SLS requirements are not intended to fully

---

[*] SLS Navigation Lead, EV42 Guidance, Navigation, and Mission Analysis Branch, ESSCA/Dynamic Concepts, Inc. Huntsville, AL 35806.
[†] PhD, Aerospace Engineer, EV42 Guidance, Navigation, and Mission Analysis Branch, NASA/MSFC, Huntsville, AL 35811.
[‡] PhD, Aerospace Engineer, EV42 Guidance, Navigation, and Mission Analysis Branch, ESSCA/Dynamic Concepts, Inc. Huntsville, AL 35806.
[§§] Aerospace Engineer, EV42 Guidance, Navigation, and Mission Analysis Branch, ESSCA/Jacobs. Huntsville, AL 35806.

describe the vehicle design but rather exist to constrain key design features with a high sensitivity to mission success and to facilitate a complete and thorough verification plan.

SLS design emphasizes the use of heritage hardware from SSLP, CxP, and other US launch vehicle systems. Hutt, Hanson, and Whitehead imply that given the selection of heritage hardware it is redundant to fully describe that hardware in requirements space.[3] The effort is better spent on characterizing the hardware, incorporating the knowledge gained from characterization into the design, and identifying key components which should be modified and/or verified.

A simplistic interpretation of the SLS program hierarchy is: Level I defines the vehicle and mission objectives, Level II is responsible for the integration and integrated systems such as GN&C and Integrated Avionics, and Level III consists of the elements responsible for managing the Prime Contractors and providing the component hardware that comprises the integrated vehicle. At the Program and Chief Engineer level, the delineation of responsibility between the levels is clear. Guidance, Navigation, and Controls (GN&C) is an integrated vehicle function. As such, SLS Navigation is a Level II integrated vehicle function. However, unlike Guidance for instance, Navigation has significantly more reliance on vehicle avionics and interfaces. This is particularly true on SLS where Navigation has the responsibility of managing all GN&C sensor redundancy functionality and interfaces including those used in the integrated navigation solution and not. Further, Navigation, as a discipline, uniquely contains the insight for the definition, specification, and modeling of navigation hardware. For this reason, the delineation of Navigation as completely belonging to Level II is peculiar and provides rationale for why SLS Navigation supports both Level II and Level III. Described below are models developed or seeded by SLS Navigation.

The models that control the design are denoted as Design Math Models (DMM). A DMM is a physical, mathematical, or otherwise logical representation of a system entity, phenomenon, or process.[4] A DMM is not specific to programming language. DMMs are implemented in a media convenient for the developer and user. The program controlled DMMs are subject to the flight certification process. The models must adhere to a model development, verification, and validation process that ensures they are adequate for design. Further, the models must be verified and validated to show that they adequately emulate the hardware or are otherwise of sufficient fidelity to emulate the subject being modelled.[3] Some of the benefits of the MBD approach to the SLS Program are detailed by Hanson.[4]

Requirements are primarily levied at the Level interfaces with respect to their responsibility to the Program. In general, Level I levies design objectives onto Level II, Level II levies vehicle hardware requirements onto Level III, and Level III levies component requirements onto hardware suppliers. Models are developed which explicitly describe the design and act as the systems engineering version of a transfer function between Level interfaces. Exceptions occur when design objectives or model behavior show high sensitivity to lower level model parameters or functions. Processes exists to elevate these exceptions.[4]

In theory, the Program Levels do not impose requirements on themselves. As an example, GN&C is part of Vehicle Management (VM), and GN&C develops the GN&C Model containing GN&C algorithms. The GN&C Model is implemented into the MAVERIC Model, a 6DOF vehicle simulation, with other component models. MAVERIC is used to show that VM (Level II) vehicle insertion accuracy requirements are met. GN&C does not decompose Guidance and Navigation accuracy from the vehicle insertion accuracy into requirement space, nor is there a strong justification to formally verify the navigation accuracy separate of the vehicle target accuracy requirement. That said, the rigor is not lost, but rather shifted to the model definition, verification, and validation.

The three sections below describe three models developed, or developed in part, by SLS Navigation. The first is the GN&C Model. Navigation is a contributor to the GN&C Model which is delivered to Flight Software (FSW) for implementation on the Flight Computers (FC). The second is the Inertial Navigation System Performance Model. The Inertial Navigation System (INS) Performance Model is developed, verified, and validated by the SLS Navigation Team for the SLS Stages Element, Level III. Level II Navigation integrates the INS Performance Model into standalone analysis tools and into MAVERIC for GN&C Model and vehicle performance requirements verification. The third model discussed is the GPS receiver Model. This is a new DMM for the evolved Block 1B vehicle design. SLS Navigation developed the initial framework for the model containing initial design assumptions associated with the performance and operation of the GPS receiver. The seed model was delivered to Level III Stages to convey initial assumptions and expected model form and function for use by the EUS Prime Contractor. Going forward, the EUS Prime Contractor will assume responsibility for updating, verifying, and validating the model against SLS GPS receiver hardware.

GN&C MODEL AND NAVIGATION SUB-MODEL

On the Ares I project under CxP, GN&C developed algorithms, implemented those algorithms in code for analysis, and then re-interpreted the implemented algorithms into a document which was used to develop FSW requirements and detailed design documents, from which code was developed for implementation on the FCs as FSW. Early in the development of SLS, GN&C and FSW were asked if they could define a more efficient approach with reduced requirements, verification, and systems engineering overhead. A new model based approach was piloted. The approach centered on the fact that the GN&C team was comprised of individuals that knew how to write C/C++ code and were already doing so for vehicle level analysis but without a strict set of coding standards or in a form that was readily portable. The solution was the use of a GN&C Model which would be developed to represent the design in a form that was directly usable by FSW. FSW then reviews the executable code and recommends changes back to GN&C. Following the integration effort, the resultant code is made suitable for implementation as FSW and used in GN&C analysis tools. The result of the pilot was a significantly more efficient method of advancing the design from GN&C designers to FC implementation with a common model. The process is illustrated by Figure 1. Development and verification of the GN&C design, represented by the GN&C model, occurs in simulation. The model is refined through integration between GN&C and FSW. Validation occurs after implementation within FSW in HWIL testing in the Systems Integration Laboratory (SIL).

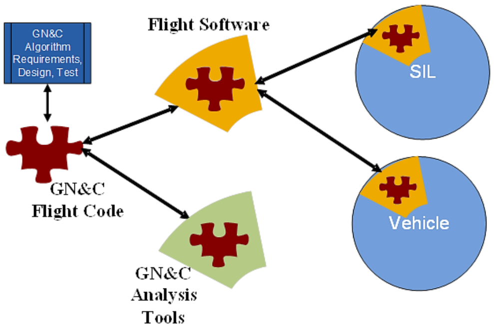

Figure 1: GN&C Common Model Illustration.

In order to convey the intent of the design, aid in review, and to maintain the rigor required for FSW and vehicle certification, a number of supplemental products were developed. First, the GN&C Model is delivered with a set of technical memorandums which describe the design and interface assumptions. Second, a comprehensive set of unit test cases, developed from simulation, accompanies the code to support proper implementation and test requirements development. Software requirements are still developed, but they are developed for the primary purpose of ensuring correct implementation and for defining FSW test and verification objectives versus fully defining the design. Hardware integrated closed loop testing is also supported with GN&C in an insight role and through the development of test appropriate success metrics. Third, the GN&C Model is delivered with a list of input and output variables, including those intended for telemetry. The interface is explicitly defined in code along with the GN&C parameters which drive the execution of the algorithms.

The purpose of the GN&C software is to fly the vehicle autonomously from liftoff to core stage Main Engine Cut-Off (MECO). The GN&C system is designed to provide vehicle stability, navigation solution, and guidance to the insertion target. The algorithms also provide data used during flight such as that used for abort trigger determination logic, sensor data quality information, and in separation events sequencing. Data pertinent to flight reconstruction is also supplied by GN&C algorithms through telemetry. Also provided by GN&C algorithms is a MECO command used to initiate Core Stage engine shutdown and upper stage separation timing, data for abort triggers and abort decisions, sensor data quality, and telemetry data.

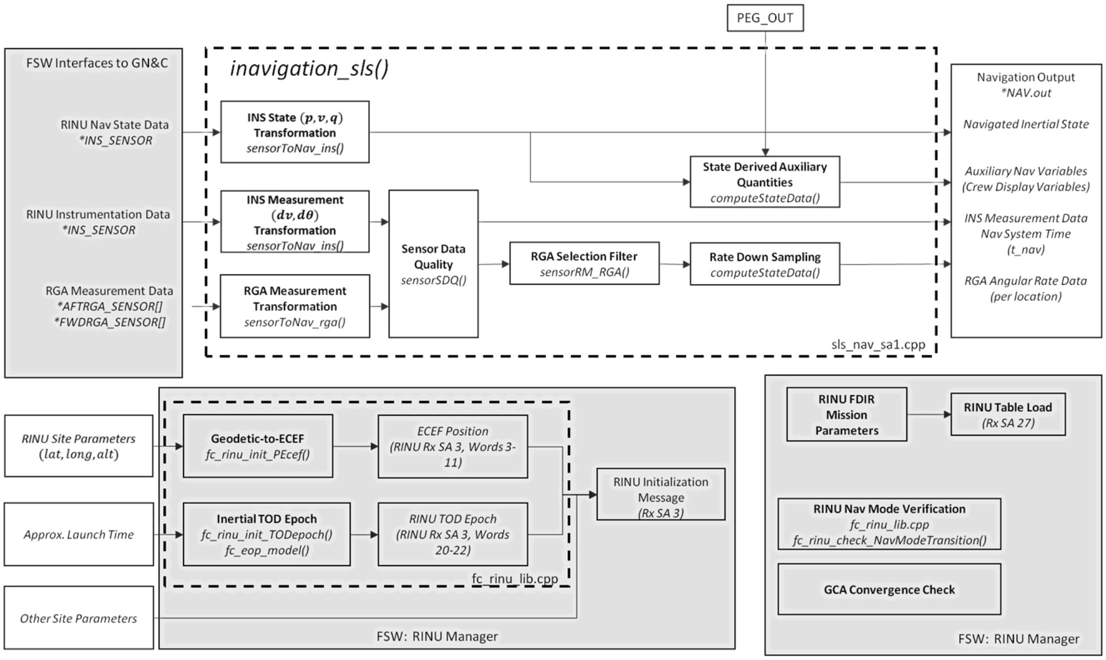

Figure 2: SLS Block 1 Navigation FSW Diagram.

The Navigation section of the integrated GN&C software is responsible for the state and state-derived data used by GN&C. This includes the processing and validating of data from the hardware managers associated with the Redundant Inertial Navigation Unit (RINU) and each Rate Gyro Assembly (RGA). The navigation code includes Sensor Data Quality (SDQ) algorithms responsible for the selection of specific RGA channels and checking the health and status of the RINU. The SDQ algorithms fill the role of redundancy management for the redundant RGA channels and ensure the quality of the measurements for additional fault tolerance down-stream of the navigation software and the use of the measurements in aborts logic. The navigation software is responsible for calculating the current state of the vehicle in terms of position, velocity, attitude, and other trajectory parameters for use in both the Controls and Guidance sections of the integrated GN&C

software. The navigation code is also responsible for computing several parameters for telemetry and crew displays. Additionally, the navigation code contains algorithms that, while not executing cyclically in the GN&C partition, are responsible for generating data used to initialize the RINU, as well as handle Fault Detection, Isolation, and Reconfiguration (FDIR) parameter loads. The RINU initialization code ensures that the error in the inertial navigation frame and the inertial frame assumed in the Guidance targets is minimized. The FDIR parameters, loaded during pre-launch, ensure proper operation of sensor redundancy within the RINU. An implementation-centric pictorial representation of the Navigation portion of the GN&C Model for the Block 1 vehicle is shown in Figure 2. For Block 1, unaided inertial navigation is used to track the vehicle state.[6] For Block 1B, the model is expanded to include aided inertial navigation and the associated interfaces and functional components.[7]

INS MODEL FRAMEWORK

One of the primary hardware models used by SLS Level II Navigation is the Redundant Inertial Navigation Unit (RINU) Performance Model, sometimes referred to as the Inertial Navigation System (INS) Performance Model. This DMM was developed to support GN&C Model development, navigation analysis, 6-DOF analysis, and Independent Verification and Validation (IV&V). Early in the SLS design process, the INS Model was chosen to pilot the MBD concept for component hardware employed on SLS. The implementation of the DMM concepts resulted in a reduction of 230 potential requirements which would have necessitated verification at the VM Level.[5] Further, the change resulted in reduced ambiguity in the specification, an extraordinarily high level of insight into the hardware design, and a forum for which to discuss and negotiate design details and data products with the hardware supplier.

In lieu of the Navigation Team specifying detailed inertial instrument and navigation system requirements, the requirements structure was simplified to only specify and verify the interface required by GN&C and the gross performance, treating the model in a “black box” fashion, illustrated in Figure 3. Key performance parameters include the specific interface comprised of data quantities and rates, navigation performance, coordinate frame definitions and alignment tolerances, and measurement frequency response characteristics. The performance was specified along a reference trajectory, consistent with an early SLS reference mission, and accuracy constraints on the interface. The error dynamics of an inertial navigation system are well known. By specifying performance in terms of integrated performance along with defining the dynamics, it is not necessary to maintain the individual error terms as requirements given that the general error dynamics are well known and the detailed error dynamics required for design will be explicitly described in the DMM. Even gyrocompassing accuracy (initial alignment) can be constrained through the allowable error in the insertion plane at the end of a reference trajectory. Likewise, an acceptable frequency response envelop was specified to ensure a clean low-band response for flight control and adequate anti-aliasing on non-navigation measurements.

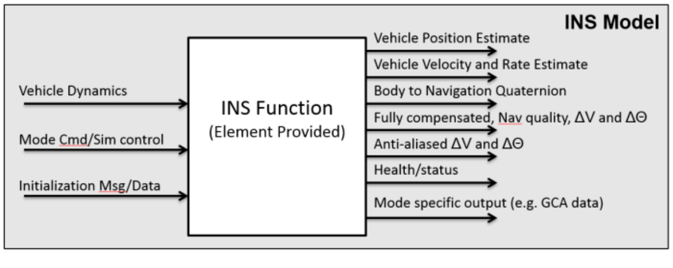

Figure 3. Level II INS Requirements Interface.

Model Capabilities and Approach

The SLS INS Model emulates the expected input/output and functional behavior of an independent INS for the SLS GN&C design. The primary function of the INS is to perform inertial navigation, a self-contained navigation technique in which the outputs of accelerometers and gyroscopes are used to track position, velocity, and orientation from a known set of initial conditions. The INS is a strap down inertial navigation system where the inertial sensor assembly are mounted rigidly to a non-rotating “platform”. As part of the primary function of the INS design, the INS is intended to autonomously align its self to an inertial frame provided the present location in the Earth Fixed Frame via gyrocompassing. As a secondary function, the INS samples the outputs of the accelerometers and gyroscopes, anti-aliases, and down-samples for use by the flight control and guidance systems. The design of the INS is intended to be fault tolerant and able to perform its primary and secondary functions with a single fault and retain the ability to detect a second fault. The SLS INS Model is intended to represent the aspects of software and hardware that impact navigation and GN&C performance. The intended use of the SLS INS Model is to perform time-domain analysis for the integrated GN&C design. The INS Model is not intended to capture the physical interfaces within the SLS Integrated Avionics Architecture (IAS), e.g. MIL-STD-1553 message and command composition.

Early analysis performed for trade and vehicle feasibility studies was conducted with a more generic INS Model. The model was matured using hardware Critical Design Review (CDR) documentation including review products developed specifically for use in model development. This allowed for the emulation of the actual algorithms implemented on the avionics unit and for the included complexity of modeling low and high-rate processing groups, internal data filtering and processing, as well as the exact navigation algorithm implementation. This level of precise modeling helps to generate a DMM that is easily verifiable. The overall functions and integration approach of the model is given in Figure 4. This describes the major components of the model including inertial sensor error budgets, navigation software, and modeling of the output interface (in terms of data generated, as opposed to direct implementation of the unit's ICD). This model is intended for integration into both standalone analysis tools and 6DOF simulation tools.

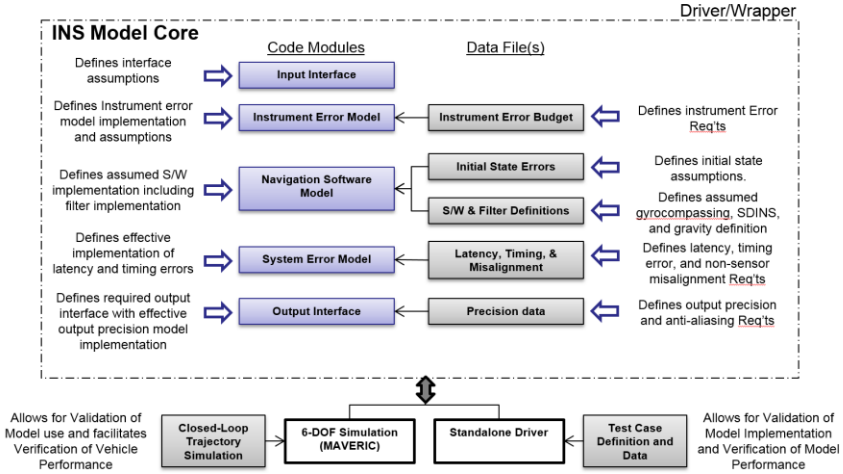

Figure 4: INS Model Functional Flow.

The model was implemented in C/C++ in order to match frameworks and tools already used by the team to support vehicle performance assessment. Another key design feature of the model is its inherent ability for the same code to be integrated into multiple environments and tools. Cross-verification of software between tools is used to ensure proper integration of the model into a new simulation package and to provide the necessary input to test and analyze other vehicle functions. The primary functionality and breakdown of the code is shown in Figure 5. This provides a high-level view of the individual algorithms implemented in the model. A listing of the primary capabilities and their rationale for inclusion is included in Table 1.

Table 1. INS Model Elements

Model Component	Rationale
Chassis and Isolator Dynamic Response Filters (High-Rate)	Modeling of vehicle's response to vibration and dynamic modes, allows for direct assessment of sensitivity to and adequacy of anti-aliasing filtering for downstream usage
Inertial Sensor Error Models (High-Rate)	Implementation of RINU Error Budget, capturing errors on inertial measurements, as well as noise and quantization modelling
Delay Model (High-Rate)	Captures latency of sensor measurements and software processes, applied at high rate to maximize precision
Dynamic Filtering (Low-Rate)	Models how the INS accumulates high-rate measurements at navigation rate and conversions to navigation frame measurements
Dynamic Compensation (Low-Rate)	Implementation of coning and sculling algorithms used to capture errors of independent measurement and navigation rate processes
Anti-Aliasing Filter (Low-Rate)	Implementation of SLS-defined filter to ensure dynamic response of sensor measurements is robust to aliasing risk for low-rate operations
FDIR (Low-Rate)	Modeling of algorithms and software used in detecting and isolated sensor faults, required for development of fault detection thresholds
Inertial Navigation	Implementation of Gyrocompassing algorithms, attitude and state integrations, initialization algorithms, frame definitions to assess integrated performance, sensitivity analysis, and integration

Model Verification and Validation

In contrast to the black-box approach taken with Level II requirements specification, model verification requires a high level of insight. Verification in part consisted of the inspection of individual function against the described hardware implementation in vendor documentation. Individual error models were unit tested and compared against expected behaviors. Further, test cases used by the INS vendor to qualify software were re-used to verify the model. Using the same input, very close comparisons were made between results recorded during INS Flight Software Qualification (FQT) and results from the model. During the V&V process, a distinction was made between developer and verifier to force independence between the model development and the model V&V. This allowed for higher confidence in the results and allows for individuals not involved with model development to gain insight into and understanding of the model. During the process, the primary model developer role shifted from active development to responding to comments and actions from the verification team. This process helped to provide additional insight into internal frame conventions, error models, and dynamic response. The summary of the verification effort was the primary product of this work and certified the model as verified for use in the SLS Vehicle Verification Analysis Cycle-1 Analysis. [8]

As the RINU program moved from hardware design to testing and qualification, test data recorded from development, qualification, and acceptance testing was made available to the model V&V team enabling validation efforts. Test data was also collected from a dynamic RINU gyrocompassing alignment test performed with a flight-equivalent RINU. The test was developed and performed by SLS Navigation at MSFC with cooperation from the Space Systems department.[9]

The modular design of the model and separation of error and navigation algorithms allows for direct validation of software algorithms, helping to decouple from sensor error uncertainties. As a result of this analysis, detailed validation was performed on the sensor's navigation algorithms.

Sensor calibration uncertainties are being used in statistical analysis to capture the robustness of the model and support error model validation efforts. An Allan Deviation assessment of the error models is being used to validate noise models, particularly in terms of sensor readout noise and quantization effects. Further model validation will be conducted as part of assessment of Green Run and a Frequency Response Test (FRT) to be performed at MSFC. The FRT will provide the data required to validate the frequency response portion of the INS Model, including anti-aliasing filter design and low frequency inertial sensor response. Once fully validated, the DMM may begin the flight certification process. Model validation is ongoing and is expected to be completed prior to SLS Flight Readiness Review.

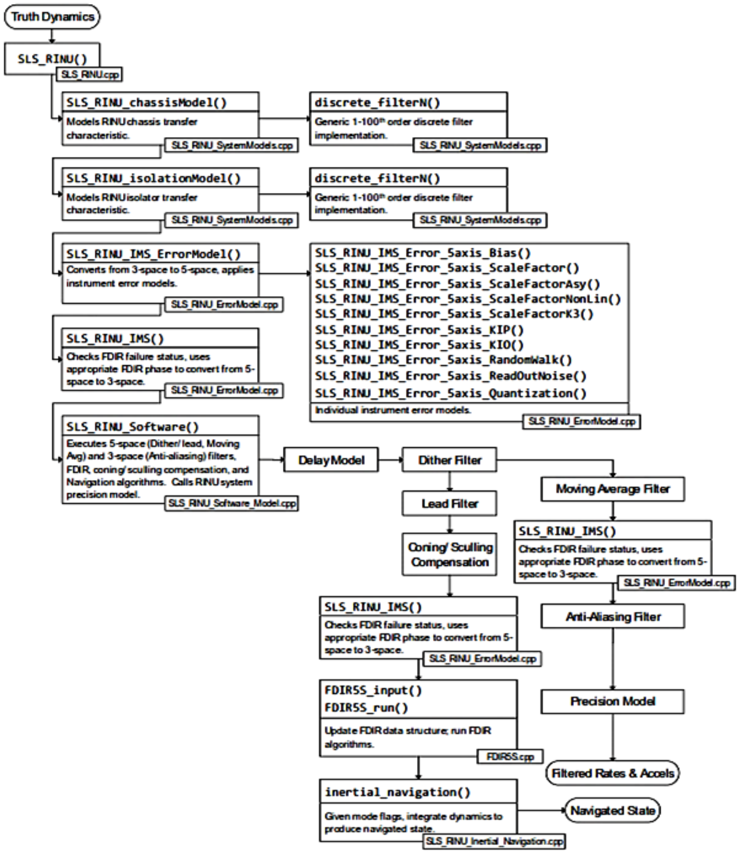

Figure 5. INS Model Code Structure

GPS MODEL FRAMEWORK

As part of the evolution from SLS Block 1 to Block 1B, the primary navigation component, the RINU, is being moved up to the EUS. A GPS receiver will be integrated onto the stage to support navigation aiding. The preliminary design and motivations are described by Oliver, et al.[7] Similar to the RINU, a detailed DMM for the GPS receiver is required to support the development process. The model is developed by the Level III Stages element to support Level II GN&C design and analysis. Additionally, the GPS Model must support implementation into simulation environments including Hardware- and Software-in-the loop simulation, MAVERIC to support vehicle level 6DOF analysis, 6DOF IV&V simulations, and standalone navigation design tools. As such, a modular model is required with a well-defined interface in order to support a wide variety of software frameworks.

To simplify the process and to convey model design expectations, the SLS Navigation team developed the Marshall Advanced GPS Model for Analysis (MAGMA) in C++ in order to provide a framework for capturing interface definitions and analysis functionality. The framework provides a definition of the input and output interfaces and breakdown of internal functionality. The model is an effective realization of model requirements conveying expectations in both form and function. As part of integration with industry partners, this framework was released with an open-use license to provide a starting point for model development. While the receiver-specific model is in development, the generic framework can be used to support software integration and testing. An additional benefit of seeding the model was in the communication of design assumptions used for early Design and Analysis Cycles (DAC).

MAGMA was originally implemented as a simulation asset used to provide insight into the GPS receiver-specific functionality and help inform sensitivities needed for requirements development. As such, the model includes the generic GPS functionalities required for modeling the onboard software and navigation processes used on a typical receiver. This includes both generic functionality for loading files, estimating SV states, inserting noise onto the measurements, and assessing link margin between the vehicle and individual GPS satellites. The model also has functions set aside to support modeling receiver specific algorithms, processes, and capabilities, specifically in terms of onboard error correction, navigation software, and output interface. As such, the generic MAGMA framework can be used to model any number of individual requirements within the same input and output interface and design. This allows for a common interface and helps to aid simulation integration. A common framework enables rapid development and reuse for future applications. The functional elements of the model architecture are illustrated in Figure 6. This diagram captures the high-level functionality and helps define the split between generic and receiver-specific capabilities.

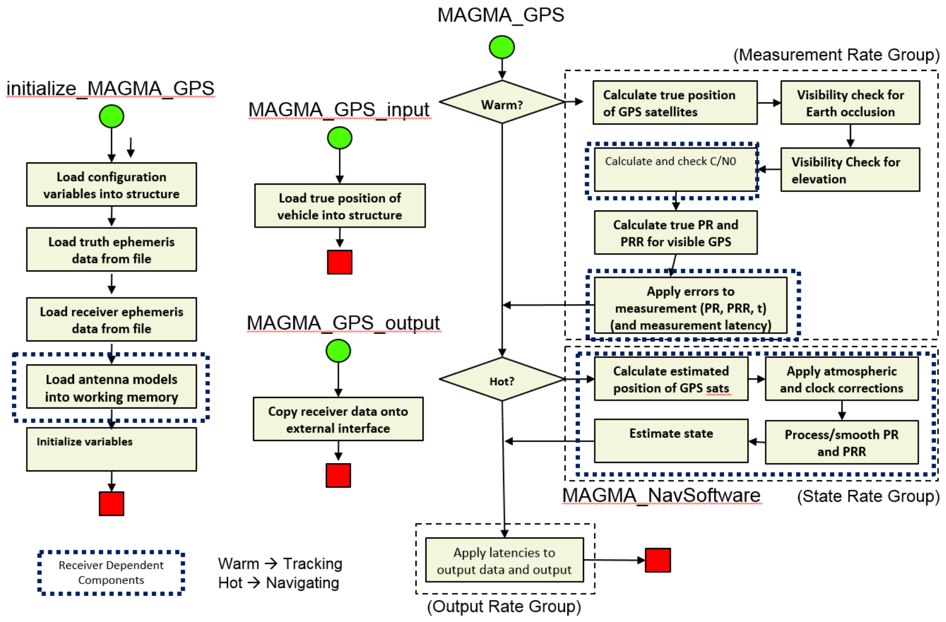

Figure 6. MAGMA Functionality.

This model seeks to capture the primary aspects of a GPS receiver to provide insight into system performance, algorithm capability, and navigation functionalities. The application of this model is focused on a multi-antenna system operating in an orbital environment. This is especially important as the model will be used to help assess link closure and GPS availability at high altitudes on a lunar-bound trajectory. As such, the model includes direct modeling of the individual antenna locations and orientations relative to the body frame. This allows for capturing moment arm effects, and is necessary when assessing the performance of the onboard solution when moment arms cannot be corrected algorithmically, i.e. when two RF inputs are joined prior to entering the receiver hardware. Additionally, a large amount of detail is provided to model antenna gain patterns both on the GPS SVs and vehicle. To accommodate this, the framework employs a modular interface to load in fixed or axisymmetric gain patterns via input files to allow for more detailed assessment. Full 2-d models are not included due to the uncertainty in GPS SV attitude. The framework allows for input of receiver losses, noise temperature, and any amplification on the RF input line to support calculation of signal power levels and carrier signal to noise ratio (C/No).

To support estimation of transmission losses and one-way light travel times, the MAGMA framework supports generic GPS almanac, broadcast ephemeris, and final trajectory data for constellation modeling. Newton interpolation was implemented to allow for use of the final trajectories to determine SV state at a defined time. The framework enables assessment of the effects of ephemeris aging with individual inputs for truth and onboard ephemeris in order to assess sensitivity. Combining the knowledge of constellation geometry, vehicle attitude and antenna placement, and receiver characteristics, it is possible to determine number of satellites in view and determine fix availability over the trajectory.

The core of the MAGMA framework is in the ability to model receiver latency and navigation processing. This includes assessing latency of measurement to truth, rate of updates, and bus latency. These aspects directly affect navigation filter development. The model provides functionality dedicated to modeling of the onboard receiver algorithms. Currently, the model implements a Least Squares solution to determine the vehicle state from GPS pseudorange and delta-range observations. This allows for direct emulation of the receiver's onboard algorithms, emulating direct simulation of raw measurements, noise sources, and integration routines, providing a much higher level of insight into onboard processes. With the framework, industry partners can include algorithms unique to specific receivers as raw code or from an embedded library, allowing for maintenance of proprietary information. Figure 7 shows position estimation errors from the MAGMA Model with a notional receiver. This provides insight into errors across the mission, and also identifies where vehicle attitude maneuvering affects the state solution.

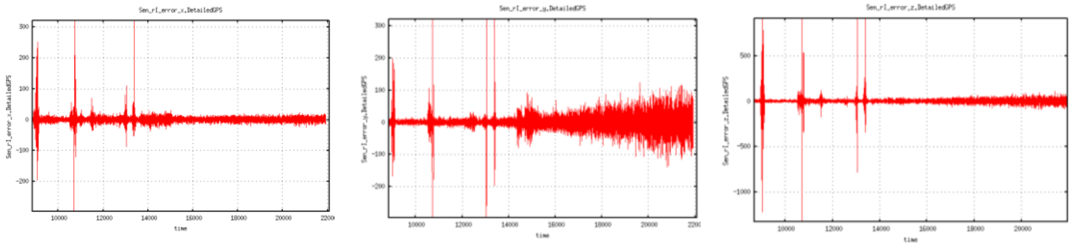

Figure 7. Notional MAGMA Errors through an In-space Trajectory

In addition to providing insight into receiver performance across a trajectory in terms of measurement accuracy, MAGMA also allows for evaluation and assessment of availability. This enables insight into sensitivities to receiver C/No parameters, orbital shape, and altitude over the trajectory. Due to its modular nature, parts of the framework can be called similarly to GPSTK. This includes loading RINEX, YUMA, or SEM files for simulation of the constellation. The broad functionality enabled rapid assessment of GPS availability across a wide range of metrics and scenarios. Figure 8 shows notional results for a EUS-like mission and was used to generate movies to support visualization of link closures over the reference trajectory. Another application is to assess availability across the Earth system. Figure 9 captures availability at the receiver given spacecraft location. Having a robust modeling framework allows for a rapid assessment for a variety of locations, allowing for a parametric trade on position in Earth orbit, focusing on signal power levels received at the spacecraft at specific times. These plots show slices at individual altitudes relative to the equator with x, and y axis representing location on the plane. The colors in the charts represent number of satellites in view, and provides insight into sensitivities to constellation geometry. The red sphere represents Earth, with the black lines representing potential altitude operational requirements, i.e. 2500, 5000, or 8000 kilometers.

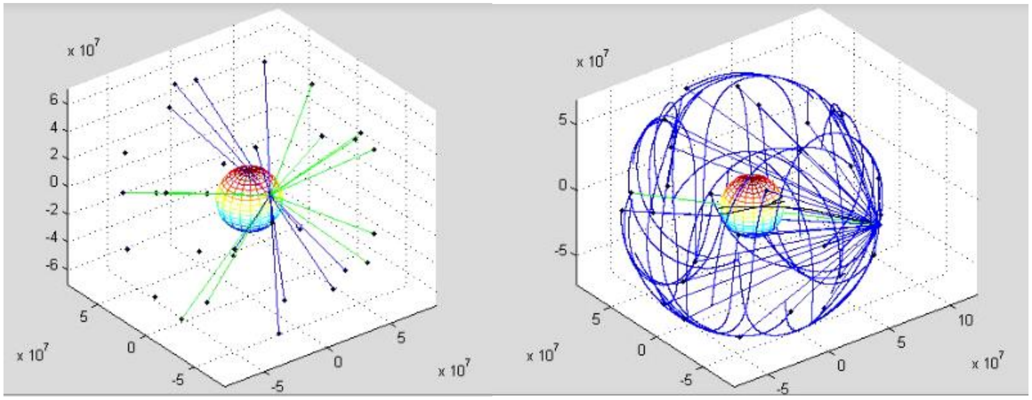

Figure 8: GPS Dynamic Simulation: at Launch (L) and post-TLI Maneuver (R)

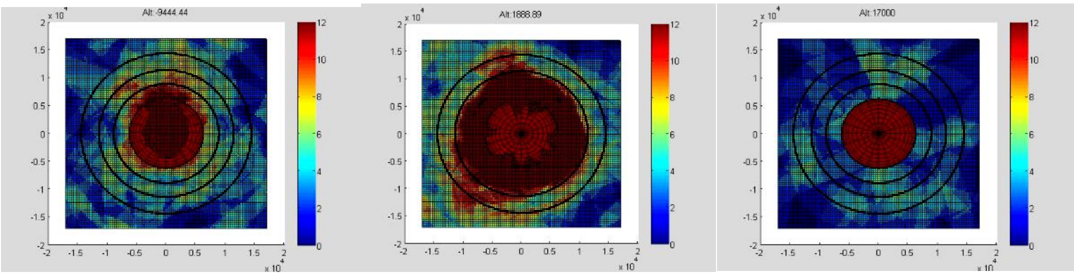

Figure 9: GPS Availability (Number of Satellites in View) as a Function of Altitude

Utilization of a common framework has allowed SLS Navigation to perform detailed sensitivity analysis and performance evaluation, while EUS GPS Receiver design trades are being completed. This has allowed for development of onboard navigation filtering algorithms to support vehicle CDR-level design. Additionally, this allowed the prime contractor to rapidly develop and begin work on receiver-specific functionality and deliver a version of the framework back to NASA teammates for reintegration and development. Post- GPS receiver development and integration, this re-delivered model will act as a DMM for the specific hardware in use, with algorithms and performance verified and validated against vendor designs and testing data thus allowing for a high level of confidence in integrated performance.

CONCLUSIONS

From a SLS Navigation perspective, the MBD approach has been hugely successful in focusing design efforts and providing a mechanism for insight required for design and analysis. Along the way some key lessons have been learned. Each Model implementation has been unique and required flexibility to meet the developer, user, and program needs for the specific application.

With the Block 1 GN&C Model, the effort required to implement software to FSW standards was not fully appreciated. This has been rectified by focusing effort early in designing code according to standard and by incorporating updates to GN&C Model into the design process. Additionally, the paradigm shift from traditional software requirements to the MBD approach has conflicted with established CMMI certified processes. A FSW Requirements Document is required as a basis for a FSW test program, but there's a balance between too much detail requiring increased overhead associated with requirements modification and not enough detail resulting in inadequate test coverage. Also, if the approach is culturally different than what has been done before, effort is required to defend the approach.

Very detailed Design Data Requirements and very tight integration with the INS vendor were key to enabling the detailed development of the INS Performance Model. Very early in the procurement process, data required for model development was specifically described and the INS supplier was contractually obligated to deliver that data. Additionally, test data that will be used for model validation from the hardware supplier needs to be defined early in contractual negotiations to ensure data availability in sync with model verification and validation efforts. Similarly, key areas of sensitivity should be defined early. If the areas of sensitivity are not adequately covered by vendor testing, then vendor test plans should be augmented or supplemental testing planned.

In the development of the INS Model, responsibility for the model intended for use in HWIL simulations and the performance model intended for vehicle analysis was split, resulting in two different models with different capabilities and different performance predictions. In the initiation of the GPS Receiver DMM development, a plan was developed to use the same core model supplemented with use-specific code to avoid a cross-validation effort later. Lastly, model V&V required more effort than originally expected. Scheduling was also more difficult than expected due to repeated hardware and testing delays. This led to Program hesitance to accept further in-house modeling responsibility. Ultimately, the increased insight helped to better inform the Navigation Team resulting in an improved design at a significant cost savings. This level of insight is needed to provide confidence in hardware testing results and expected flight performance.

REFERENCES

[1] W. Gerstenmaier. "HEO Status Report". NASA. November 29th, 2017.

[2] W. Gerstenmaier. "Progress in Defining the Deep Space Gateway and Transport Plan". Presentation to the NASA Advisory Council. NASA. March 28th, 2017.

[3] J. Hutt, J., Whitehead, & J. Hanson,. (2017). "NASA's Space Launch System: Systems Engineering Approach for Affordability and Mission Success". In *AIAA SPACE and Astronautics Forum and Exposition* (p. 5330).

[4] Department of Defense, Directive 5000.59, M&S Management, 1998.

[5] J. Hanson. "SLS Vehicle Modeling and Simulation". *NDIA Annual Systems Engineering Conference; 18th*; 26-29 Oct. 2015; Springfield, VA; United States

[6] MSFC SLS Navigation Working Group. "SLS Navigation Flight Software Description". February 13th, 2017.

[7] T. E. Oliver, et al. "Space Launch Systems Block 1B Preliminary Navigation System Design", *Proceedings of the AAS Guidance and Control Conference*, Breckenridge, CO., 2018.

[8] Bernard, W. & Geohagan, K. (2016) "Verification of the Redundant Inertial Navigation Unit (RINU) Model, DMM-CS-0411". Report included with delivery of DMM-CS-0411 v3.0.

[9] K. Geohagan, et al. "6DOF Testing of the SLS Inertial Navigation Unit", *Proceedings of the AAS Guidance and Control Conference*, Breckenridge, CO., 2018.