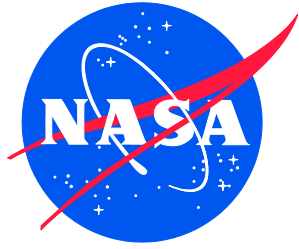


NASA/TM-2020-5008164
NESC-RP-18-01312



Development and Verification of a Pipeline for Modeling Flexible Multibody Dynamics

*Tannen S. VanZwieten/NESC
Langley Research Center, Hampton, Virginia*

*Aaron D. Schutte, Abhinandan Jain, and Kimberly A. Simpson
Jet Propulsion Laboratory, Pasadena, California*

*Jeffrey R. Brouwer
Troy 7 Inc., Huntsville, Alabama*

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

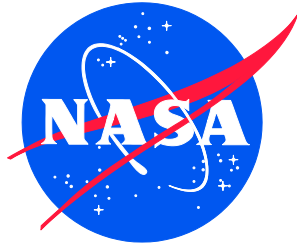
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to help@sti.nasa.gov
- Phone the NASA STI Information Desk at 757-864-9658
- Write to:
NASA STI Information Desk
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

NASA/TM-2020-5008164
NESC-RP-18-01312



Development and Verification of a Pipeline for Modeling Flexible Multibody Dynamics

*Tannen S. VanZwieten/NESC
Langley Research Center, Hampton, Virginia*

*Aaron D. Schutte, Abhinandan Jain, and Kimberly A. Simpson
Jet Propulsion Laboratory, Pasadena, California*

*Jeffrey R. Brouwer
Troy 7 Inc., Huntsville, Alabama*

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

September 2020

Acknowledgments

The NESC assessment team gratefully acknowledges the peer reviewers of this report. They include Steve Gentz, Ben Jimenea, Dexter Johnson, Jeb Orr, and Eric Stoneking.

The use of trademarks or names of manufacturers in the report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500



NASA Engineering and Safety Center Technical Assessment Report

Development and Verification of a Pipeline for Modeling Flexible Multibody Dynamics

August 11, 2020

Report Approval and Revision History

NOTE: This document was approved at the August 11, 2020, NRB. This document was submitted to the NESC Director on August 19, 2020, for configuration control.

Approved:	<i>Original Signature on File</i>	8/20/20
	NESC Director	Date

Version	Description of Revision	Office of Primary Responsibility	Effective Date
1.0	Initial Release	NESC Principal Engineer's Office	August 11, 2020

Table of Contents

1.0	Notification and Authorization	5
2.0	Signature Page	6
3.0	Team List	7
3.1	Acknowledgments	7
4.0	Executive Summary	8
5.0	Assessment Plan	10
6.0	Background	11
6.1	Challenges in Modeling Flexible Multibody Dynamics for GN&C	12
6.2	The Multibody Dynamics Approach	14
6.3	Background on Relevant DARTS/Dshell Tools	16
7.0	Assessment Summary	17
7.1	The Pipeline	19
7.1.1	The Process from FEM to GN&C Model	19
7.1.2	The FModal FEM-to-Multibody Dynamics Tool	20
7.1.3	Dshell Interface and Linearization	21
7.1.4	Application of the Pipeline	22
7.2	Improving GN&C Flexible Body Dynamics Model Fidelity	23
7.3	MSFC/JPL Model Development and Comparisons	26
7.3.1	MSFC Model Description and Heritage	26
7.3.2	Overview of Results	28
7.4	Demonstration for Large Space-Based Observatory Study	32
8.0	Findings and NESC Recommendations	33
8.1	Findings	33
8.2	NESC Recommendations	33
9.0	Alternative Viewpoints	34
10.0	Other Deliverables	34
11.0	Lessons Learned	34
12.0	Recommendations for NASA Standards and Specifications	34
13.0	Definition of Terms	34
14.0	Acronyms and Nomenclature	34
15.0	References	36
Appendices		37
Appendix A	Component Body Modal Analysis	38
A.1	Modal Models for an Unconstrained Body	38
A.2	Modal Models for an Attached Body	39
A.2.1	Component Mode Synthesis	39
A.2.2	Normalizing the Component Modes	40
A.3	Five-Particle Spring-Mass Example	41
Appendix B	Flexible Body Dynamics Modeling Approaches for GN&C	45

	B.1	Nonlinear Rigid/Flexible Multibody Dynamics	45
	B.1.1	Minimal Coordinate Rigid/Flexible Multibody Dynamics	45
	B.1.2	DARTS Rigid/Flexible Multibody Dynamics Software	47
	B.2	Alternative Fallback Solutions	47
	B.2.1	Linear Superposition Approach	47
	B.2.2	Rigid Bodies with Springs Approach	48
Appendix C		FModal: A FEM-to-GN&C Flexible Multibody Modeling Pipeline Tool	49
	C.1	Software Description	49
	C.2	Normal Modes Analysis of a 10-Node Symmetric Beam	49
	C.3	HDF5 Output File Definitions	54
	C.4	Craig-Bampton Models	56
	C.5	Iterating FEM Models Using FModal	57
Appendix D		System-Level Modeling and Simulation	58
	D.1	Dshell System-Level Modeling Framework	58
	D.2	SLS Modeling and Simulation Using Dshell	60
Appendix E		Modal Integrals for Flexible Bodies	64
	E.1	Equations of motion for the k^{th} body	64
	E.1.1	Equations of motion for the single \mathcal{O}_k^j node (derivatives in the \mathcal{O}_k^j node frame)	64
	E.1.2	Equations of motion for the single \mathcal{O}_k^j node (derivatives in the k^{th} body frame)	66
	E.1.3	Modal representation at the node level	68
	E.1.4	Modal representation at the body level	68
	E.1.5	Equations of motion for the k^{th} flexible body	69
	E.2	Expressions for $M_{fl}(k)$	70
	E.2.1	Expression for $M_{fl}^{ff}(k)$ using modal integrals	71
	E.2.2	Expression for $M_{fl}^{tt}(k)$ using modal integrals	71
	E.2.3	Expression for $M_{fl}^{tf}(k)$ using modal integrals	72
	E.2.4	Overall structure of $M_{fl}(k)$	73
	E.2.5	Equations of motion in modal coordinates	74
	E.3	Special case: Point mass nodes	75
	E.3.1	Expression for $b_{fl}(k)$ (point mass nodes)	76
	E.4	Definition of Terms	76
Appendix F		MSFC/JPL Launch Vehicle Model Validation	80
	F.1	Summary	80
	F.2	Introduction	80
	F.3	Airframe Model Context	81
	F.4	Airframe Modeling Assumptions	81
	F.5	Tool Development and Implementation	85
	F.6	Test Cases	86
	F.7	Summary and Conclusions	97
	F.8	Test Case Definitions	98
	F.9	Test Case 1: Pitch Prescribed Engine Angular Acceleration Input	99
	F.10	Test Case 2: Pitch Engine Gimbal Torque Input	103
	F.11	Test Case 3: Pitch Prescribed Engine Angular Acceleration Input	105
	F.12	Test Case 4: Yaw Engine Gimbal Torque Input	109
	F.13	Test Case 5: Pitch Engine Gimbal Torque Input with Yaw Prescribed Input (Repeat of Case 2 with Yaw Prescribed)	111
	F.14	Test Case 6: Yaw Engine Gimbal Torque Input with Pitch Prescribed Input (Repeat of Case 4 with Pitch Prescribed)	113

F.15	Test Case 7: Pitch Engine Gimbal Torque Input, Thrust = 0 (Repeat of Case 2 with Zero Thrust)	115
F.16	Test Case 8: Yaw Engine Gimbal Torque Input, Thrust = 0 (Repeat of Case 4 with Zero Thrust)	117
F.17	Bonus Test Case: $\ddot{\delta}_{p0}$ and $\ddot{\delta}_{y0}$ Sensitivity Study Repeat of Test Case 4, Yaw Engine Gimbal Torque Input	120
Appendix G	Flexible Spacecraft Modeling for Microthruster Assessment	122
G.1	Problem Description	122
G.2	LUVOIR-A FEM	122
G.3	GN&C Actuators and Sensors	124
G.4	Building the State Space Model Using FModal and Dshell	125

1.0 Notification and Authorization

This project was proposed by Jet Propulsion Laboratory (JPL) personnel in response to a call from the NASA Engineering and Safety Center (NESC) Chief Engineer at JPL for discipline-advancing work. The work described in this NESC Final Assessment Report was requested to improve the efficiency and accuracy of analyses to support the dynamic modeling of complex aerospace systems.

The members of the NESC assessment team have extensive experience in the development and independent verification and validation (IV&V) of analysis tools related to guidance, navigation, and control (GN&C) of compliant structures. The requested work is intended to reduce the need for iterations of structural models between dynamicists and GN&C practitioners, a process that often extends the duration of design cycles. In addition, the finite element method (FEM) models delivered for use in GN&C analyses are often optimized for other applications, such as loads or stress analyses. This work was requested to tailor the process of extracting relevant FEM model information for use specifically in GN&C analyses.

The key stakeholders for this effort are the structural dynamicists responsible for FEM development and the GN&C teams using these models.

2.0 Signature Page

Submitted by:

Team Signature Page on File - 8/31/20

Dr. Tannen S. VanZwieten	Date
--------------------------	------

Significant Contributors:

Dr. Aaron D. Schutte	Date
----------------------	------

Dr. Abhinandan Jain	Date
---------------------	------

Mr. Jeffrey R. Brouwer	Date
------------------------	------

Ms. Kimberly A. Simpson	Date
-------------------------	------

Signatories declare the findings, observations, and NESC recommendations compiled in the report are factually based from data extracted from program/project documents, contractor reports, and open literature, and/or generated from independently conducted tests, analyses, and inspections.

3.0 Team List

Name	Discipline	Organization
Core Team		
Tannen VanZwieten	NESC Lead	KSC
Kim Simpson	NESC Co-Lead	JPL
Aaron Schutte	Technical Lead	JPL
Abhinandan Jain	Multibody Dynamics	JPL
Jeff Brouwer	SLS GN&C	MSFC/ESSCA/Troy 7 Inc.
Consultants		
Neil Dennehy	NASA Tech Fellow for GN&C	GSFC
Aron Wolf	NASA Deputy Tech Fellow for GN&C	JPL
Joel Sills	NASA Deputy Tech Fellow for Loads and Dynamics	JSC
Victor J. Cabezas-Tapia	SLS GN&C	MSFC
Carl Leake	Multibody Dynamics	JPL
Business Management		
Kay Little	Program Analyst	LaRC/MTSO
Assessment Support		
Linda Burgess	Planning and Control Analyst	LaRC/AMA
Jenny DeVasher	Technical Editor	LaRC/AS&M
Jocelyn Santos	Project Coordinator	LaRC/AMA

3.1 Acknowledgments

The NESC assessment team gratefully acknowledges the peer reviewers of this report. They include Steve Gentz, Ben Jimenea, Dexter Johnson, Jeb Orr, and Eric Stoneking.

4.0 Executive Summary

This NESC assessment focused on developing and verifying a toolchain to improve the process of generating and integrating structural dynamics data for use in multibody aerospace system models. Flexible multibody dynamics modeling is often critical for the design and analysis of GN&C systems used to achieve desired closed-loop system performance. Examples include large flexible spacecraft that may require high-precision pointing, flexible launch vehicles that use gimballed engines for thrust vector control, rotorcraft systems where rotor and blade dynamics can impact closed-loop stability, and even small, highly maneuverable missiles where the inertial loading of a high bandwidth control system can excite structural bending.

The Structures and Loads & Dynamics disciplines also rely on structural dynamics models for vehicle design and to analyze loads and stresses. While these disciplines and GN&C share a need for high-fidelity structural models to predict dynamic behavior, fragmented modeling approaches have historically persisted because the needs of the disciplines differ. The Structures discipline typically makes extensive use of the FEM to model the dynamics of structures with complex geometry and material properties. A FEM model is used to assess the structural response of individual structural elements to applied forces and moments, and structural designs are refined based on load path optimization. A FEM model used for stress analyses will typically incorporate much denser grids in localized areas to capture stress/strain concentrations on elements of interest. On the other hand, a FEM model optimized for GN&C requires only enough detail to capture the dynamic responses in the sensor-to-actuator path of the vehicle during flight. FEM models developed to evaluate loads and stress typically contain far more detail and complexity than required for GN&C modeling efforts. Unfortunately, converting a high-resolution FEM model into a reduced-order modal model for GN&C remains a challenging task, especially for multibody systems. Some of the challenges arise from the cross-disciplinary nature of the problem, while others are due to a lack of toolchains that bridge the gap between the disciplines. Common issues include time-consuming design iterations, a lack of GN&C-focused sanity checks prior to model delivery, and reliance on custom tool sets that can be difficult to modify and often incorporate complex dynamics in an ad-hoc manner.

This NESC assessment addresses common issues in building a GN&C flexible multibody dynamics model from a FEM model by developing a FEM-to-GN&C modeling pipeline using a general multibody dynamics framework. This work resulted in a tool called FModal, which streamlines the processing between Structures and GN&C models. The tool interfaces with the widely used FEM analysis tool NASA STRucture ANalysis (NASTRAN), and generates non-standard NASTRAN outputs for use in GN&C flexible multibody dynamics models and simulations. A common portable output file based on the Hierarchical Data Format (HDF5) file format is generated by FModal to support GN&C simulation development. The FModal tool also interfaces with the JPL Dynamics and Real-Time Simulation (DARTS) Lab software and simulation interface (Dshell) to quickly generate flexible body models from large FEM datasets that can be incorporated into linear and nonlinear GN&C tools. The pipeline development included DARTS linearization capability that extracts linear models for frequency-domain design and analysis directly from the full-fidelity nonlinear multibody time-domain simulation.

The pipeline was demonstrated using a FEM with on the order of 10^5 nodes originally developed for the Space Launch System (SLS) Core Stage Integrated Modal Test (IMT). This model included the rocket body and four integrated core-stage engines. FModal was used to generate and incorporate the flexible body data into the GN&C multibody dynamics model within DARTS/Dshell. This process was used to compare the effects of integrating the engines into a single FEM model versus creating a FEM model for a core stage and then attaching the engines. The newly developed DARTS/Dshell linearization capability was also exercised to allow for frequency-domain design and analysis. A Dshell time-domain model simulated closed-loop control of the flexible launch vehicle from launch to T+300 seconds.

DARTS/Dshell models were also compared against models developed by Marshall Space Flight Center (MSFC) Engineering using an intentionally asymmetric multi-body FEM designed to emphasize dynamic coupling between bodies. This test configuration represents a flexible launch vehicle with a single gimbaled engine and can be configured to demonstrate the different methods of dynamically coupling the vehicle and engine. A series of eight increasingly complex test problems was created, and responses between the models produced by the FEM-to-GN&C modeling pipeline and MSFC tools were compared. The effort demonstrated excellent comparisons in the frequency domain for all dynamic responses. This provided a high level of confidence that the FEM-to-GN&C modeling pipeline could be used to generate models for analysis of launch vehicle flight control systems.

In addition to the work with MSFC, the team used the FEM-to-GN&C modeling pipeline to develop GN&C flexible body models for the NESC Assessment TI-18-01371, "Microthruster Systems for Low-Jitter Space Observatory Precision Attitude Control." This assessment required flexible body models to assess the relative pointing stability performance of microthrusters and reaction wheels. The microthruster team benefited from the FEM-to-GN&C modeling pipeline, which quickly delivered a portable flexible body model of the 15-meter Large Ultraviolet Optical Infrared Surveyor (LUVOIR) telescope. The resulting model was implemented in a GN&C simulation developed by The Aerospace Corporation for the microthruster study.

The NESC assessment team also developed an integrated approach to improve flexible multibody model fidelity by incorporating higher-order modal integral terms that couple rigid and flexible body dynamics. These effects are often neglected, but can be significant for systems that experience large deflections or angular rates. The effects of these modal integral terms are rarely assessed quantitatively since the parameters can be difficult to extract from a FEM model and incorporate into a GN&C model. The team included these effects in the pipeline by developing computational methods in FModal for modal integral data extraction from a FEM model and extended DARTS to correctly include modal integral effects in the multibody dynamics model.

In summary, this assessment developed and verified a multibody FEM-to-GN&C pipeline to rapidly produce GN&C dynamics models. The performance of this pipeline was demonstrated for two representative aerospace systems. The tool improves model fidelity by capturing modal integral terms that can be significant for systems that experience large deflections or angular rates. Projects should consider early in their life cycles whether a multibody dynamics approach using this FEM-to-GN&C pipeline is a good fit for their system. Benefits of this approach include improved efficiency when developing and validating system models and when incorporating updates as the project evolves. Several key findings emerged during this project when comparing the results of two independent multibody dynamics models: Analysts responsible for performing linearization of a nonlinear system model with flexible body dynamics should (1) take into consideration the impact of trim initial conditions, (2) pre-load with an appropriate acceleration vector, and (3) ensure adequate data precision in the FEM output file. Finally, to accurately evaluate the interactions between the bodies in a multibody model, tool comparisons benefit from testing using both prescribed motion of control inputs and test cases that capture closed-loop flexible-body interactions between elements of a multibody system induced by control inputs.

5.0 Assessment Plan

This assessment was broken into two phases. Phase 0 used a representative ascent phase launch vehicle model to compare airframe models from the DARTS tool with models from MSFC. DARTS is a general-purpose flexible multibody dynamics simulation tool used for a wide variety of aerospace programs and missions. The MSFC toolset includes a high-fidelity structural dynamics model that incorporates multiple launch vehicle dynamics formulations. The Phase 0 baseline models paved the way for incorporating the flexible body dynamics enhancements developed in the follow-on phases.

During Phase 1, the assessment was split into four tracks:

- Track 1 - Development of the pipeline (FEM-to-GN&C software tool) (Appendix C)
- Track 2 - Increasing fidelity of flexible multibody modeling with DARTS/Dshell (Appendix D and Appendix E)
- Track 3 - MSFC model development and comparison with DARTS/Dshell (Appendix F)
- Track 4 - Development of a flexible body model for the NESC microthruster assessment TI-18-01371 (Appendix G)

Track 1 developed a FEM-to-GN&C modeling pipeline, with emphasis on the development of a tool called FModal to streamline the process of incorporating data from NASTRAN FEM models into GN&C system level flexible multibody models. In addition, linearization capability was added to DARTS/Dshell. Linearization capability is heavily relied on for GN&C design workflows. The use of FModal and a capable flexible multibody dynamics simulation tool like DARTS/Dshell could help eliminate common modeling errors caused by miscommunication between the technical disciplines and streamline the modeling pipeline for increased efficiency and reduced program cost. In Track 2, the NESC assessment team focused on increasing flexible multibody model fidelity beyond what is traditionally implemented for space vehicle applications. This was accomplished by extending FModal to extract modal integrals from NASTRAN models and by incorporating the resulting data products into DARTS/Dshell. Modal integrals can be significant in situations where there are large angular rates or when body deformation causes changes in inertia. Track 3 verified the FEM-to-GN&C modeling pipeline by comparing flexible multibody dynamics launch vehicle models developed using FModal and DARTS/Dshell with equivalent models developed at MSFC. Finally, in Track 4, FModal and DARTS/Dshell were used to develop flexible body dynamics models for use in NESC Assessment TI-18-01371 to assess closed-loop pointing stability of large space-based observatories using microthrusters.

6.0 Background

Modeling and simulation (M&S) is one of the central activities in aerospace engineering. Flexible multi-body dynamics modeling in particular is a critical component in the GN&C analysis of aerospace vehicle systems. These systems include large spacecraft that require high-precision bus and payload pointing and launch vehicles that use gimballed engines for thrust vector control where large propellant slosh masses may be present. These flexible-body effects may even arise on small, highly maneuverable missiles where the inertial loading of a high bandwidth control system can excite structural bending, or in rotorcraft systems where rotor and blade dynamics can impact closed-loop stability. Flexible body dynamics modeling plays an important role in the broader control-structure interaction (CSI) problem, which can involve many technical disciplines, such as controls, structures, optics, thermal, and flight software. The GN&C analysis of these systems must consider the complex large-scale motion of the overall vehicle where interaction from effects such as flexible body dynamics, articulating component bodies (e.g., solar arrays, parachutes, and gimballed appendages), time-varying masses due to propellant depletion, closed-loop control, and configuration changes are all at play.

High-fidelity dynamics models are heavily used in the GN&C technical discipline to design and predict closed-loop system performance. The structures discipline also relies on dynamics models to design structures and analyze vehicle loads and stresses. These two disciplines share a common need for high-fidelity models that accurately predict vehicles' dynamic behavior. Historically, fragmented modeling approaches have persisted because the needs of the structures and GN&C disciplines differ. The structures discipline typically makes extensive use of the FEM to model the dynamics of structures with complex geometry and material properties. A FEM model is used to assess the structural response of individual structural elements to applied forces and moments, and structural designs are refined based on load path optimization. A FEM model used for stress analyses will typically incorporate much denser grids for localized areas to capture stress/strain concentrations on elements of interest. On the other hand, a FEM model optimized for GN&C must focus on the detail required to capture the dynamic responses in the sensor-to-actuator path of the vehicle during flight. Both loads and stress FEM models typically contain far more detail and complexity than required for GN&C modeling efforts. To account for flexibility in the dynamics model, the GN&C discipline often seeks to capitalize on existing structures FEM modeling efforts. Unfortunately, incorporating FEM data into the dynamics model in a reduced-order modal model for GN&C remains a challenging task, especially for multi-body systems. Some of the challenges arise from the cross-disciplinary nature of the problem, while others are due to a lack of toolchains that bridge the gap between the disciplines.

Reduced-order flexible multibody dynamics models must be developed to capture key features for the specific system under consideration. For example, the presence of large angular rates, large articulations, or time-varying mass properties must be considered. Other system-specific considerations include appropriate mode shape retention, effector modeling implementation, discontinuous structural properties (e.g., across separation events), and appropriate structural damping based on material selection or fabrication techniques. Another complicating factor may be that FEM models of various system elements may be provided by multiple vendors. The quality and format of these models can vary greatly. In many cases, the GN&C model is based on a custom toolset that is difficult to extend or modify. For example, in the case of a hand-derived linearized model, the coupling effects are decided upon a priori. If any effects are missing, model iteration is required to incorporate them. This approach can lead to a constant state of model development.

An interesting example of communication challenges between structural dynamicists and GN&C practitioners arises when describing the form of the FEM used to describe a multi-body system. In the late 1960s, Arthur Greensite was responsible for developing a seminal series of 16 volumes intended to illustrate the methods used for the design and analysis of space vehicle flight control systems. This work was performed

by General Dynamics/Convair for NASA/MSFC. In “Volume XV–Elastic Body Equations” [1], Greensite popularized the “engines-in-the-modes” (EITM) nomenclature to describe a FEM that includes the mass of the core vehicle as well as the masses of the engines and sloshing masses. By extension, an “engines-out-of-the-modes” (EOTM) FEM includes the total system mass minus the mass of the moving bodies. In an interesting twist, this nomenclature persisted on Air Force programs such as the Titan IV launch vehicle while NASA/MSFC generally abandoned these terms, instead using “Integrated Body Model” in place of EITM and “Reduced Body Model” in place of EOTM.

6.1 Challenges in Modeling Flexible Multibody Dynamics for GN&C

As shown in Figure 6.1.1, the conventional approach for building GN&C flexible multibody dynamics mod-

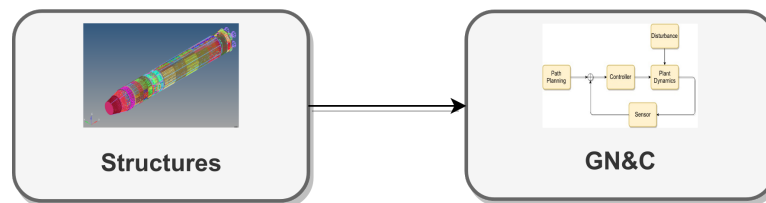


Figure 6.1.1. Structures to GN&C Workflow

els begins with the structures team. Structures analysts typically work with static configurations of the vehicle, where loads and stress analyses are the primary focus. The data required for these analyses are typically derived from FEM-based structural models using commercial off-the-shelf tools, such as NAS-TRAN. A FEM model is the foundational data source for building a GN&C flexible multibody dynamics model. However, the modeling needs of the structures analysts do not precisely align with those of the GN&C team. The models used for stress analysis typically focus on high-resolution modeling of individual components and are often impractically large to incorporate into a global vehicle model. Likewise, the FEM models used for loads analyses may contain thousands of modes that have no influence on the global low-frequency vehicle-level response critical to GN&C design. Specialized conditioning of the FEM model is required for application of these models for use in a GN&C flexible multibody dynamics model, especially when inter-body coupling constraints or dynamic actuator elements must be incorporated. The gap between the static, linear stress or loads analysis models developed by the structures analysts and the time-varying, possibly large articulation, flexible multibody dynamics models required by GN&C analysts is large. As a result, the state of practice in the development of GN&C models is less than desirable and fraught with challenges. Key challenges of modeling flexible dynamics for space vehicles are described below:

1. **Multiple disciplines:** The structures and GN&C disciplines require different skills and expertise. It can be challenging for GN&C analysts to effectively communicate their needs and usage to the structures team, which has very different analysis goals and needs.

Common cross-disciplinary issues in developing GN&C dynamics models include:

- Design and analysis iterations between structures and GN&C are time-consuming.
- Model iteration between disciplines is often complicated by miscommunication of GN&C requirements due to the use of different nomenclature or different assumptions about modeling details.
- Sanity checks are not performed or completed before delivery of structural datasets.
- Component structural models are not available; only the system model is available.

- Code development is required to interface structural datasets with GN&C tools.

Some of these issues can result in months of delays, which reduces the time available to design and refine the models for GN&C.

Good communication and interface definition is essential for the exchange of correct and properly formatted data needed for the GN&C models.

2. **Complexity of structures models:** Structures models can involve many component parts and millions of degrees of freedom and can be time-consuming to develop. Specific design decisions and assumptions used to develop a structures model can have significant impact on GN&C users, though the subtle aspects and impacts may not be evident to the GN&C user who tends to work with modal-based flexible body dynamics models. This can be an especially difficult issue when FEM models include data from proprietary models with “opaque” modeling approaches that may be used to protect classified or proprietary hardware models. Navigating the best path to work around such limitations is non-trivial when extracting GN&C relevant model data.

Another common complexity arises when component models are combined within the FEM model. While this may be adequate for structures analysis, it can be a problem for GN&C analysis when the component bodies can undergo large relative motion or when active effectors can transmit forces between nodes. One example is the EITM versus EOTM launch vehicle modeling issue. With EITM models, actuator dynamics are difficult to model properly for GN&C analysis.

3. **Specialized tools and inadequate tool-chains:** The tools and design techniques used by the structures and GN&C disciplines are different enough that there is little chance for cross-training. It is rare to find engineers who are comfortable using tools and manipulating models across the disciplines. The more typical practice is to process and hand off data products from one discipline to the other, with little ability to revise or rework upstream assumptions. Defining and transmitting model data can take a long time because of the gaps in expertise across the disciplines. It is easy for assumptions built into the models and data to be misunderstood, leading to the use of incorrect data or misuse of model data.

Large amounts of transformed and specialized data needs to be computed from the FEM model for the GN&C model. Examples of such data are mode shapes, grid point locations, and modal integrals. Without a proper toolchain, it is not possible to implement sanity checks throughout the process to catch errors that may creep in. Due to these difficulties, the path often taken is to degrade the model data to simplify the data transfer process. The diverse nature of the tools across the disciplines means that in the absence of toolchains to bridge the gap, data definition and handoff remains a manual process. This is expensive and time-consuming, and provides limited ability to regenerate the data to correct shortcomings. Since models are provided to GN&C analysts on the back end of the model development, when there is typically little time remaining for model revisions, GN&C analysts may not have the opportunity to provide feedback to structures design teams to help iteratively improve the design.

4. **Fragmented approach to model development:** Systems like space platforms, with extensible solar panels or effectors, may require models that can simulate large articulation angles. Integrating static, linear FEM models into flexible multibody dynamics models with large rigid-body motion and articulation required for GN&C analysis is a challenging and non-trivial task. Structures models can be large and intractable for direct use in GN&C analysis, making model reduction essential. This leads to multiple problems. One problem is the proliferation of ad-hoc models that tend to have limited applicability with less than desired fidelity. These models can also take significant time and effort to tune and substantial resources to validate. This can lead to the fragmentation of modeling efforts through the development of multiple independent models for different configurations and purposes. The development of these types of narrow models at times requires specialized techniques and can

become a niche art-form by itself. Validating and comparing results across such models can be a Herculean task. Since it is non-trivial to define the range of validity of a model, there is the constant danger of using the models outside their range of validity.

An example of such fragmentation is in the development of linearized state-space models commonly required for control system design. Since linear models are integral for flight control design, there are efforts to directly develop them to meet GN&C needs. However, in the absence of a robust methodology for linearization, it can be unclear which effects are included in the linearized model. In addition, there are extensibility concerns, i.e., questions as to whether the linearized model can extend to handle newer aspects of the vehicle model, such as additional bodies or degrees of freedom.

A fragmented modeling approach has been common across the aerospace domain, leading to a tendency to continue to develop ad-hoc models rather than develop a more comprehensive alternative solution.

5. **Inadequate model fidelity:** Due to the complexity of modeling rigid/flexible dynamics coupling correctly through large motions, there is a tendency to develop and use models of reduced fidelity. Examples include models where effects such as the rigid/flexible body dynamic coupling, large angle motion and articulation, and/or the inherent nonlinear nature of the dynamics are ignored. Inadequate fidelity or the extension of low-fidelity approaches to more complex scenarios can adversely affect GN&C design and performance.

The challenges increase when spacecraft undergo large time-varying changes to their properties, such as mass property changes from fuel consumption, body deployment, or jettisoning of bodies. In some specific cases, additional dynamics effects need to be taken into account that go beyond the linear elastic theory typically incorporated into GN&C models. Examples include the effects of changing inertia properties from deformation or the geometric stiffening when a body undergoes large displacements or accelerations (e.g., rotor blades). Capturing these effects is not possible in the absence of a foundationally high-fidelity model.

6. **System-level models:** In addition to modeling basic rigid/flexible dynamics, vehicle models at the system level also must include sensors, actuators, and environment interactions that are tightly coupled with underlying vehicle dynamics and system-level dynamics, such as aero-elastic and servo-elastic effects. The challenges of developing the core rigid/flexible dynamics model and the fragmentation of such models makes it difficult to properly disseminate and integrate the additional models needed to evaluate GN&C performance.
7. **Poor computational efficiency:** A key part of the GN&C design and performance evaluation process is closing the loop between the control system and the dynamics model. To be effective, the model should be computationally efficient when implemented in time-domain simulations. Given the complexity of dynamics models, obtaining adequate computational speed is non-trivial.

Due to these challenges, the modeling state of practice in the aerospace community remains less than ideal, with a multitude of one-off, ad-hoc, and low-fidelity solutions filling the gap. This inefficient mode of operation can also negatively impact program design risk and extend design cycles. As models in aerospace systems continue to grow in complexity and variety, the community could benefit from a standardized high-fidelity modeling approach.

6.2 The Multibody Dynamics Approach

A paradigm that offers a solution to the problems described in Section 6.1 is to bring the discipline of multibody dynamics into the mix. Adding a new discipline as a remedy may appear counterintuitive, given the existing challenges of dealing with the structures and GN&C disciplines. As shown in Figure 6.2.1, the rationale for this approach is that multibody dynamics can provide solutions for converting the static

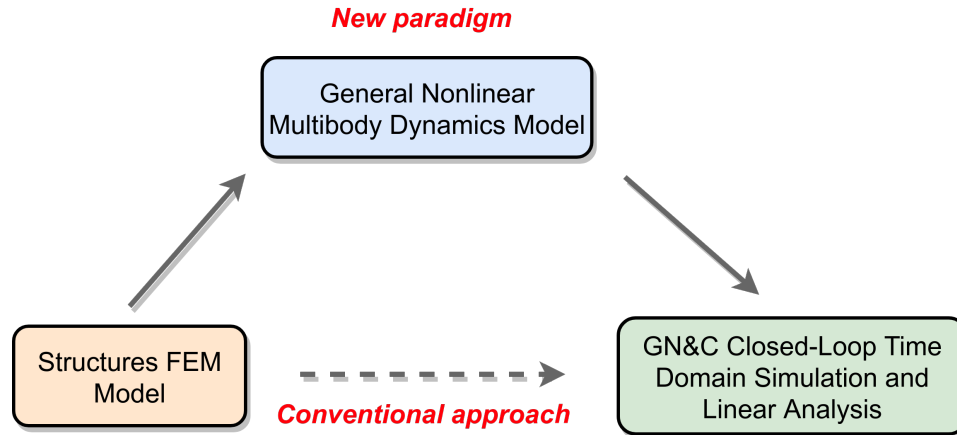


Figure 6.2.1. New paradigm for Linking General Multibody Dynamics Modeling with Structures and GN&C Modeling Efforts

and linear FEM models originating from structural dynamics into full-scale dynamics models that properly handle large-motion, articulation, and rigid/flexible dynamics coupling that may be important to GN&C modeling needs. Multibody dynamics has been around for many decades, and the solutions it has traditionally offered often serve to make the problem even worse. Before proceeding further with flexible body modeling, a look at the multibody dynamics field can provide perspective.

The discipline of multibody dynamics provides methodologies for systematically modeling the dynamics of articulated, rigid/flexible multibody systems. Multibody dynamics models incorporate overall system dynamics involving large-body motions and their coupling to the deformation degrees of freedom (DOF). An accurate multibody dynamics model facilitates the design and development of GN&C algorithms, as well as their performance evaluation and test in closed-loop simulation. While the underlying foundations of dynamics modeling from Lagrangian mechanics [2] have been well known for more than two centuries, the complexity of models for systems of even small to moderate size has led the field to develop its own discipline, with a proliferation of modeling and algorithmic approaches often based on advances in the field of analytical dynamics [3–5].

Using well-established principles, multibody dynamics offers a way of developing high-fidelity, nonlinear dynamics models of the vehicle platform, including large articulation motion and rigid/flexible dynamics coupling. Once a high-fidelity model is available, system-level vehicle models can be developed by interfacing this model with actuator, sensor, and environmental interaction models. With such a model available, it is conceptually easier to derive other models for GN&C, such as reduced-order models; linearized models; and models to support different vehicle configurations, closed-loop and V&V test-beds, etc. A comprehensive multibody solution can dispense with the need for ad-hoc modeling approaches. Even where multiple reduced-scope models are needed, they would be traceable and comparable to a high-fidelity multibody dynamics gold-standard model.

The field of multibody dynamics has failed to live up to its promise, mainly because of the sheer complexity of the dynamics models for even modest-sized systems. During the early years, the focus of the discipline was on the development of modeling approaches that correctly modeled the nonlinear and coupled nature of multiple, inter-connected rigid/flexible body dynamics. These methods were motivated by the significant complexity in the equations of motion from a direct and naive application of Lagrangian principles. A major advance was Kane's approach [6], which provided a systematic and scalable procedure for assembling the equations of motion of complex systems by decoupling the kinematics and dynamics aspects of the problem. Furthermore, the approach used minimal coordinate representations that are well-suited for

GN&C dynamics models. While this approach provided a path to developing correct models, the models remained complex and had to be assembled manually for each system. Moreover, the eventual cost of solving the equations of motion grew as the cube of the number of degrees of freedom. Scaling this approach for multibody systems remained a problem, and the model complexity required multibody experts to create the models. Software to help with model development led to tools using symbolic methods and auto-code generators for the equations of motion [7]. These tools were typically limited to rigid bodies and could not handle flexible-body dynamics.

Another methodology that has become popular is one using non-minimal absolute coordinates [4, 8–10]. In this approach, modeling complexity is reduced by treating a multibody system as a collection of independent bodies. The bodies are then assembled into a multibody system by adding constraints to model the inter-body coupling. Setting up the system dynamics is therefore simple, and it is left to a numeric solver to use the constraints to compute the correct solution. The non-minimal coordinates approach requires the use of more complex differential-algebraic solvers and constraint stabilization methods. While the inherent cost of solving the equations of motion still remains at cubic order, specialized methods have been developed to reduce the cost by exploiting the sparse structure of the underlying matrices.

Eventually, recursive methods emerged as a new approach from the robotics domain. Recursive algorithms provided remarkable gains in efficiency by using minimal coordinate models that reduced the computational cost for evaluating equations of motion from cubic to linear growth with the number of degrees of freedom. The new recursive methods solved the equations of motion exactly, but at much lower computational cost [11–13]. Despite their benefits, the sheer analytical complexity of minimal coordinate models has led to few adopters or tools that follow this path, and the ones that do are largely limited to rigid body dynamics models.

In summary, the key challenges in multibody dynamics modeling are:

- Accurately capturing the dynamic complexity in the model, particularly in regard to the interfaces between individual bodies.
- Models may not match GN&C needs.
- Available modeling tools are often not extensible to body flexibility.
- The computational cost of solving the equations of motion can be high.
- Models are often hand-crafted by a resident multibody dynamics expert, thereby requiring extensive validation efforts to gain confidence in the model.
- Changes to multibody dynamics models typically require the support of a multibody dynamics expert and can be time-consuming.
- Developing models to support multiple vehicle configurations remains non-trivial.
- The interfaces between the dynamics and device models for integrated system level models can be complex and difficult to extend, especially as project needs evolve.

6.3 Background on Relevant DARTS/Dshell Tools

Development and demonstration of a new and improved pipeline for building GN&C flexible multibody dynamics models and simulations is the primary goal of this NESC assessment. To this end, the NESC assessment team pursued the multibody paradigm described in Section 6.2, but used the DARTS/Dshell multibody modeling approach to overcome the typical challenges associated with using the multibody approach in GN&C modeling. This toolkit has been used to model a variety of complex aerospace systems, including Cassini, SLS/Multi-Purpose Crew Vehicle (MPCV), Mars 2020, Mars Helicopter, Mars Science

Laboratory, Phoenix, InSight, and Sierra Nevada's DreamChaser [14, 15]. The following is an overview of the DARTS/Dshell multibody capability:

- **Spatial operator algebra (SOA) methodology:** Underpinning the DARTS/Dshell toolkit is SOA theory and methodology. SOA provides mathematical and analytical tools for effectively working with the inherent complexity of minimal coordinate multibody models for small to large systems. The approach exploits underlying structure to carry out analysis and develop recursive low-cost algorithms whose complexity grows only linearly with the number of degrees of freedom. Moreover, the methods apply to rigid as well as flexible multibody systems. The minimal coordinate models are especially well-suited for GN&C models and allow the use of standard solvers since they do not have to work with explicit constraints. The SOA methodology is described in Appendix B.1.1.
- **DARTS multibody engine:** The DARTS multibody dynamics software is an object-oriented implementation of the linear cost SOA algorithms for rigid/flexible multibody dynamics. DARTS is designed to handle the nonlinear dynamics of systems with arbitrary branching topologies and size, smooth and non-smooth dynamics, and run-time configuration changes, and also provide a full complement of computational algorithms needed for dynamics analysis and model-based control with fast computational performance [16]. While the DARTS object-oriented implementation is in C++, a Python interface is available for all the classes and methods in the system. This allows users flexibility to directly define and configure the model as desired and even to modify the model topology and properties without requiring the involvement of a multibody dynamics expert. The bodies can be added, removed, or reattached at run-time. The DARTS multibody engine is described in Appendix B.1.2.
- **Dshell system level modeling framework:** While DARTS handles the articulated body dynamics of a vehicle, Dshell provides a component model-based simulation framework to add and couple parameterized models for system-level models to support the incorporation of servo-elasticity, aero-elasticity, actuator/sensor devices, and environment interactions. Thus it provides the ability to model rigid/flexible body dynamics as well as system-level dynamics effects that are important for GN&C analysis. Dshell has provisions for closing the loop with GN&C software for time-domain simulations. The Dshell system-level modeling framework is described in Appendix D.1.

The DARTS/Dshell framework is general-purpose and can model rigid/flexible multibody systems with an arbitrary number of bodies and branching topology without modifying the software. This middleware can be reused as-is across testbeds, projects, and model configurations. The low cost of SOA's recursive dynamics algorithms makes this framework especially suitable for closed-loop and even real-time simulations. Furthermore, the structure-based algorithms adapt to run-time configuration changes in the vehicle without requiring configuration-specific code changes. The use of minimal coordinates allows simpler and standard dynamics solvers for propagating the system state. The Dshell toolkit includes standard simulation services for data logging, three-dimensional visualization, units checking, etc. The Python interface allows users to configure and interact with the simulation without requiring experience as a C++ programmer. This toolkit addresses the shortcomings of conventional multibody approaches described in the previous section [17].

7.0 Assessment Summary

The primary goals of the assessment were to improve the fidelity of flexible body GN&C models and simplify the process of creating these models from the FEM developed by structures analysts. Section 6.0 described the key challenges of coordinating the requirements of the structures and GN&C disciplines and discussed an approach for integrating the discipline of multibody dynamics to address these hurdles. The NESC assessment team pursued the approach of developing and demonstrating a novel pipeline to standardize and streamline the process of integrating the structures, multibody dynamics, and GN&C technical

disciplines.

A capable flexible multibody dynamics tool is a key component of this approach. For this assessment the team selected JPL's DARTS/Dshell toolkit. As discussed in Section 6.3, the DARTS/Dshell toolkit was developed to address the many pitfalls of a multibody approach (outlined in Section 6.2). This tool also provides a foundation for developing and demonstrating a new GN&C flexible body modeling pipeline. Though this assessment utilized the DARTS/Dshell toolkit, the ideas and developments are not limited to this environment and can be integrated into other tools. After selecting a dynamics modeling tool as the baseline for development, gaps were identified in the process or pipeline between FEM model development by structures engineers and the utilization of the resulting data by GN&C engineers. This assessment focused on filling these gaps to create a pipeline prototype, improve model fidelity, and complete a number of case studies and demonstrations to verify and validate this process. The overall pipeline is illustrated in Figure 7.0.1. An overview of the specific efforts carried out in this assessment follows.

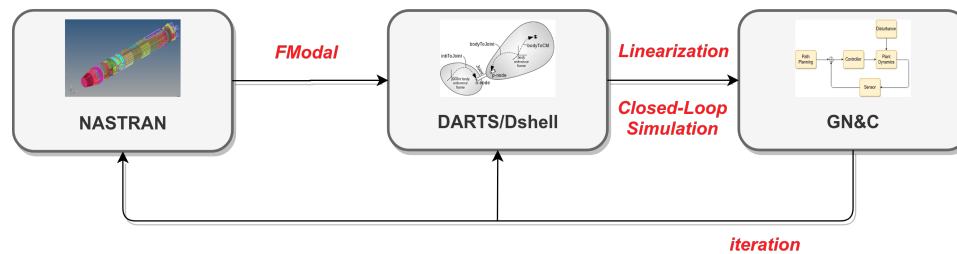


Figure 7.0.1. FEM-to-GN&C Modeling Pipeline

Improving GN&C Flexible Body Dynamics Model Fidelity: While the multibody approach provides a systematic and rigorous way to model the rigid/flexible coupled dynamics of vehicle systems, in practice, many tools (including DARTS/Dshell) neglect some of the higher-order terms present in rigid/flexible dynamics coupling. The reason for such neglect is typically a process issue. Including these effects in the multibody model requires the computation and extraction of a number of additional modal integral terms from a structural dynamics model. In the absence of a smooth process for computing and transferring this data, the expedient approach is to simply ignore these terms. However, these effects can be important when changes in body inertia due to body deformation must be captured. Section 7.2 provides an overview of the work done in this assessment to extend the DARTS multibody tool to correctly include the modal integral effects.

The FModal FEM-to-GN&C Tool: As mentioned above, multibody dynamics tools often implement reduced-fidelity flexible body models to reduce the data extraction burden on the structural dynamics models since the process can be quite complex even for basic flexible body dynamics data needs. The NESC assessment team developed the FModal software tool for simplifying and streamlining the process of generating modal data (including modal integrals) from component NASTRAN structural dynamics models for use in flexible multibody dynamics tools. FModal provides an output in the form of a portable HDF5 file that is a hierarchical, well-organized, and labeled dataset that can be interfaced with multibody tools to automate and streamline the model data transfer process. The team also developed a DARTS interface to directly populate multibody model data from the HDF5 file generated by FModal for defining component flexible bodies within DARTS. Section 7.1.2 provides an overview of the FModal tool and its use in bridging the gap between NASTRAN structural dynamics and multibody dynamics models. Additional details are provided in Appendix C.

Bridging the Gaps Between Multibody Dynamics and GN&C Modeling: With FModal bridging the gap between structural and multibody dynamics, the NESC assessment team also considered the gaps between the multibody dynamics and GN&C domains (see Figure 7.0.1). The team focused on two primary needs: (a) the need to generate system-level linearized state space models (including servo-

and aero-elastic effects) to support GN&C algorithm design and development, and (b) the need for building time-domain simulations with high-fidelity, nonlinear rigid/flexible multibody models to assess the performance of the GN&C system. The team added an automated linearization capability to the DARTS/Dshell tool to meet need (a) while meeting need (b) with existing DARTS/Dshell capabilities. The team demonstrated the pipeline by creating a representative launch vehicle model and simulation in DARTS/Dshell using the SLS integrated modal test (IMT) FEM. A summary of the study is provided in Section 7.1.1.

MSFC Model Development and Comparison with DARTS/Dshell: To validate the new DARTS pipeline, the NESC assessment team carried out a study to compare this model with models developed by MSFC Engineering. The MSFC tool includes a high-fidelity airframe dynamic model with legacy to several flight programs. This effort and modeling approach is summarized in Section 7.3, and further details are provided in Appendix F. The airframe model is defined in [18].

Demonstration for Large Space-Based Observatory Study: As a demonstration of the generality of the pipeline, this NESC assessment team supported the NESC Assessment TI-18-01371, “Microthruster Systems for Low-Jitter Space Observatory Precision Attitude Control” (assessment still active as of the time of this report). Specifically, flexible body models of a large space-based observatory were developed and delivered. The pipeline was exercised in this study by using an FEM model of the LUNAR-A spacecraft to create linear state space models for use in a GN&C simulation constructed by The Aerospace Corporation. While the NESC assessment team for TI-18-01371 struggled to acquire a representative flexible body model from other sources, using the pipeline they were able to quickly obtain a portable FEM-based flexible body model. Section 7.4 summarizes the work executed in this study. Additional information is provided in Appendix G.

7.1 The Pipeline

In addition to improving GN&C model fidelity, another goal of the assessment was to improve the modeling pipeline between structural dynamicists and the GN&C practitioners who use FEM model data in a so-called FEM-to-GN&C pipeline. A key enabling component of this pipeline includes the use of a general multibody dynamics framework. Several solutions were developed in the assessment to bridge the identified disciplinary gaps and challenges, as described in the following sections.

7.1.1 The Process from FEM to GN&C Model

The FEM-to-GN&C modeling pipeline uses the multibody approach, and in this assessment specifically, the DARTS/Dshell rigid and flexible multibody dynamics modeling capability. A workflow is outlined here for how the NESC assessment team exercised the pipeline to bridge the multibody dynamics and GN&C modeling gaps:

1. The new FModal capability simplifies the process of extracting the necessary model data from a component FEM model. The extraction process produces a portable HDF5 file that encapsulates the data needed to generate a GN&C flexible body model. The HDF5 file can be used by any multibody dynamics toolset. The extraction process is easily repeated as needed to regenerate the data when modeling assumptions change. Minimal involvement of the structures analyst is required in this process, removing the need for the time-consuming and expensive coordination traditionally required when iterating the model.
2. Once HDF5 files are generated for the individual bodies, the multibody dynamics tool (DARTS in this case) can be exercised to directly import the modal data into its multibody dynamics model for each of the bodies.

3. Within Dshell, sensor and actuator models attached to flexible bodies are affected by body deformation, allowing full capture of servo- and aero-elastic effects in the system model.
4. At this stage, a high-fidelity rigid/flexible system-level model of the vehicle is available for use in nonlinear time-domain simulation and linear analysis.
 - (a) The model linearization capability built into Dshell can then be used to automatically create linear state space models of the vehicle system around a user-desired configuration and for any user-selected combination of inputs and outputs. The linearized models are used by GN&C analysts to design control system algorithms.
 - (b) Alternatively, the GN&C software can be used to close the loop around the high-fidelity DARTS/Dshell system-level vehicle model to evaluate performance against realistic time-varying conditions affecting the vehicle.
 - (c) The DARTS/Dshell model can also be delivered and shared with other V&V and hardware-in-the-loop project testbeds as needed.

The new pipeline simplifies the processing, data extraction, and creation of high-fidelity nonlinear rigid and flexible dynamics models for GN&C analysis. This pipeline can be easily exercised to iterate between modeling assumptions that can have a significant impact on GN&C analysis and performance. For example, the use of scalable multibody and system-level modeling capability such as DARTS/Dshell simplifies the development of complex models (since no code-changes are required) and furthermore removes the need for development of ad-hoc lower-fidelity and narrow models and the fragmentation of modeling efforts. The use of the state-of-the-art low-cost recursive algorithms provides the computational speed necessary for closed-loop simulations.

7.1.2 The FModal FEM-to-Multibody Dynamics Tool

The NESC assessment team developed a software tool called FModal to simplify the model data extraction process from NASTRAN. From the GN&C perspective, this tool is a key gap filler for enabling an iterative GN&C flexible body modeling and simulation workflow. The tool was designed to allow a GN&C analyst to directly process a NASTRAN model in bulk data format for GN&C modeling purposes. This eliminates the need for specialized data deliveries by the structures analyst and reduces the chances of error due to miscommunication between the structures and GN&C disciplines.

The FModal software is written in Python and C++. The software interfaces with FEM models constructed in NASTRAN to generate non-standard NASTRAN outputs for use in GN&C flexible multibody modeling and simulation. The software also interfaces with JPLs DARTS/Dshell software to allow flexible multibody dynamics simulation development in DARTS/Dshell. The team also developed a portable output file in HDF5 format to assist simulation development in other environments. The software simplifies computation and extraction of the elements needed to build a GN&C flexible multibody dynamics model. Some of the GN&C relevant data products produced by FModal include:

- Computation and extraction of modal integrals
- Craig-Bampton models
- Normal mode models
- Nodal mass, damping, and stiffness matrices in sparse matrix format
- Selection of interface and output grid points
- Output coordinate transformations

FModal accomplishes data extraction from NASTRAN by first importing a NASTRAN model in bulk data format. A custom NASTRAN run deck is then automatically created, and includes a GN&C-centric Direct

Matrix Abstraction Program (DMAP). NASTRAN is executed using the custom run deck to generate the relevant GN&C data using the NASTRAN OUTPUT2 (OP2) file format. The OP2 data is converted to a standardized HDF5 file format. Figure 7.1.1 illustrates the basic FModal processing flow. The team tested

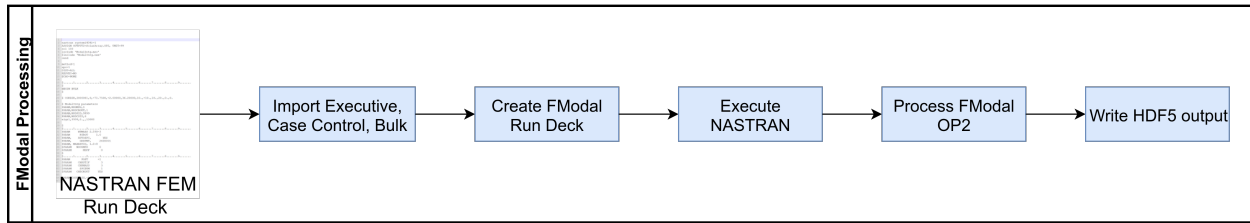


Figure 7.1.1. FModal Processing Diagram

the software by creating regression tests to exercise software functionality. The FModal output data was extensively used in regression testing and in the construction of GN&C flexible multibody models using DARTS/Dshell. Figure 7.1.2 shows an example output HDF5 file generated by FModal. Among other data,

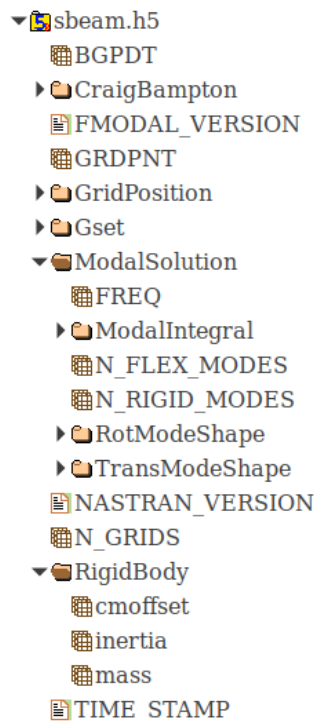


Figure 7.1.2. FModal HDF5 Output File

the example file includes rigid body mass properties, normal mode shapes, modal integrals, and a reduced Craig-Bampton model. Further details about the FModal software and its use in building GN&C flexible multibody models are provided in Appendix C.

7.1.3 Dshell Interface and Linearization

To enable pipeline capability within DARTS/Dshell, an interface was developed from the FModal output (currently in HDF5 file format) to Dshell. Specifically, Dshell classes and methods were created to generate

a DARTS flexible body based on FModal output. Additionally, Dshell linearization was generalized to allow arbitrary specification of system inputs and outputs for linear state space modeling. Methods for trim conditioning of pre-loaded flexible body models were also developed for system linearization.

The linearization capability includes two-point and four-point central difference schemes as well as Richardson extrapolation [19], a sequence acceleration method that can be used to further reduce the error in each numerical difference scheme. The finite difference step sizes used in the linearization process are selected by a user-defined relative error tolerance. The algorithm works with the general Dshell machinery and enables linearization of complex models that can include, for example, the underlying multibody dynamics, actuators, sensors, gravity, slosh, and aerodynamics. Capability was also developed to export the linear model as a state space model in HDF5 format. This enables a common framework for completing nonlinear time-domain simulation and linear analysis activities. The linearization algorithms were verified by comparing Dshell linearized systems with analytically derived linear models, as described in Section 7.3.

7.1.4 Application of the Pipeline

The utility of the flexible body data generated by FModal and eventual use in a system-level GN&C simulation was demonstrated by developing a DARTS/Dshell simulation of a representative flexible launch vehicle based on the SLS IMT FEM [20]. The NESC assessment team used the simulation to demonstrate how to carry out time-domain simulation and linear analysis using the pipeline. As depicted in Figure 7.1.3, the launch vehicle model included the Launch Vehicle Stage Adaptor (LVSA), the core stage, and four

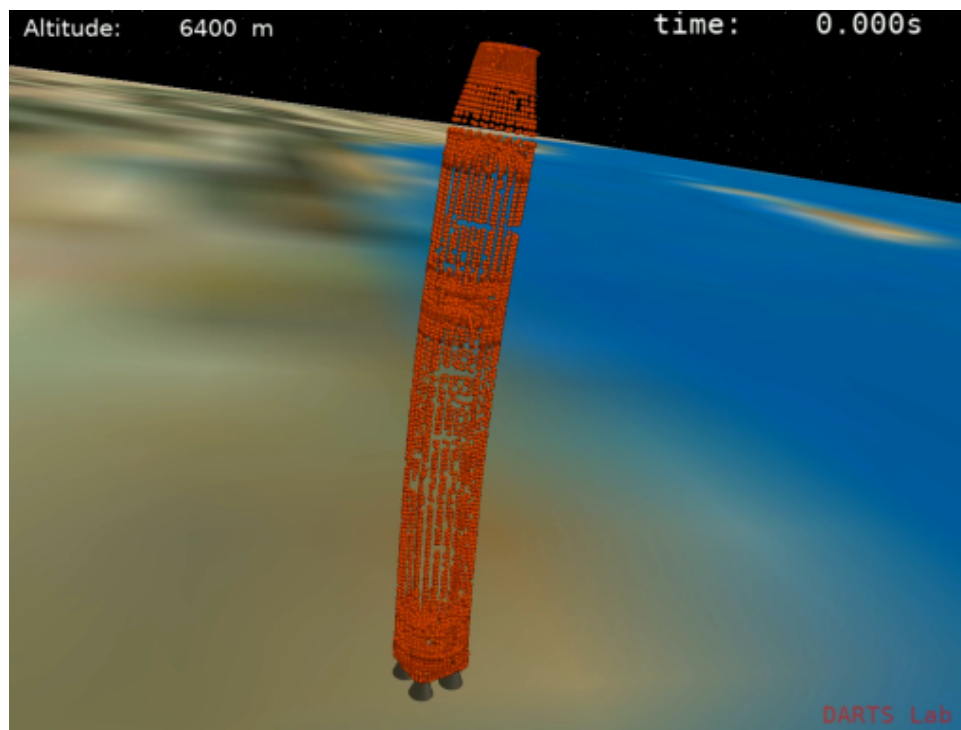


Figure 7.1.3. First Bending Mode of a Representative Launch Vehicle Model

core stage engine bodies from the SLS IMT FEM. For simplicity, ideal actuators were assumed at the engine joints (i.e., actuator dynamics not included, resulting in force/torque from engine being applied directly to the vehicle). In Dshell, two multibody models were constructed that included EITM and EOTM mode sets.

The Dshell simulation is highly reconfigurable. Coupled with the pipeline, it can be used to execute quick studies of different core body mode sets and engine attachment assumptions. Updates to include effects from fuel slosh, aero-elastic forces, and higher fidelity engine actuator models are also straightforward. Appendix D provides more detail about this demonstration.

The NESC assessment team has developed and demonstrated a solution that includes tools, toolchains, and processes that the team believes can address GN&C model development challenges. While the assessment used the DARTS/Dshell toolkit, most of the ideas and toolchain components are readily portable and usable with other multibody tools and environments—with the expectation that they offer the key architectural features from DARTS/Dshell that are vital to pipeline success.

7.2 Improving GN&C Flexible Body Dynamics Model Fidelity

The equations of motion of a GN&C flexible body dynamics model for an aerospace vehicle are typically derived by assuming that the elastic motion of the body is linear and the elastic displacements can be represented using modal coordinates (see Appendix A). As discussed in Appendix B, multibody dynamics provides a rigorous approach to deriving the nonlinear rigid/flexible dynamics equations of motion for vehicles. Alternative methods, such as linear superposition and rigid bodies with springs, while simpler to implement, can fail to fully capture the nonlinear coupling between the rigid body and linear-elastic motion.

The modal data required by flexible multibody dynamics models is derived by processing FEM structural dynamics models. This data is typically limited to mode shapes, modal frequencies, and grid points for the component bodies. This limited data set is often the only data available due to the large size of the modal datasets and the complexity of the extraction process. While properly capturing the first-order nonlinear rigid multibody dynamics coupling with linear flexible dynamics, the limited data set ignores higher-order dynamics effects, such as the deformation-dependent variation of the first and second moments of inertia of a flexible body. The additional modal data needed to capture such higher-order dynamics effects are the full set of modal integrals. Beyond the variation in body inertia properties, modal integrals are needed to capture additional coupling between the rigid and deformation dynamics of a flexible body. This can be observed in the following partitioned form of the mass matrix of the k^{th} flexible body in a multibody system when using the floating frame of reference approach [21]:

$$M_{\text{fl}}(k) = \begin{pmatrix} G^k & [F^k]^* & [E^k]^* \\ F^k & \mathcal{J}(k) & m(k) \tilde{p}(k) \\ E^k & -m(k) \tilde{p}(k) & m(k) I_3 \end{pmatrix}, \quad (7.2.1)$$

where $m(k)$ is the body mass, $\mathcal{J}(k)$ is the deformation-dependent moment of inertia tensor of the body, and G^k is the generalized mass matrix corresponding to the modal degrees of freedom. The partitioning separates the elastic, rotational, and translational degrees of freedom for the body. Each of the G^k , F^k , E^k , $\mathcal{J}(k)$, and $\tilde{p}(k)$ matrices are deformation dependent, with the variation being defined by the modal integrals.

In particular, the matrices F^k and E^k define the coupling between the elastic and rigid degrees of freedom within the mass matrix. When the modal integrals are ignored, i.e., assumed to be zero, the body mass

matrix reduces to the following form:

$$M_{fl}(k) = \begin{pmatrix} G_0^k & 0 & 0 \\ 0 & \mathcal{J}_0(k) & m(k) \tilde{p}_0(k) \\ 0 & -m(k) \tilde{p}_0(k) & m(k) \mathbf{I}_3 \end{pmatrix} \quad (7.2.2)$$

This mass matrix is block-diagonal, and all of its entries are constant. In this form, the coupling between the elastic and rigid degrees of freedom within the mass matrix is lost. Other coupling terms in the equations of motion also vanish when the modal integrals are ignored. Additional details and derivation of the modal integrals are provided in Appendix E. For this assessment, the DARTS multibody capability was extended to include modal integral effects.

The significance of the higher-order terms depends on the simulation model and scenario. For example, dynamics models of flexible bodies undergoing large rotational rates or large translational accelerations underpredict body stiffness since geometric stiffening cannot be modeled without modal integrals. Realistically, assessment of whether higher-order effects are significant can be achieved only with a side-by-side comparison of models with and without the modal integral terms. In practice, however, the primary reason modal integral data is not included is the large size of the modal dataset and the complexity of extracting modal integral data from FEM models. One goal of this assessment was to simplify the extraction of modal integral data so complete modal datasets are available for use within flexible multibody models.

To illustrate the coupling of elastic and rigid-body motions due to modal integrals, a DARTS/Dshell simulation was constructed using a NASTRAN FEM model of a tapered beam. The modal integrals of the body were computed using the FEM-to-GN&C tool FModal (see Appendix C) and used in the DARTS multibody dynamics model. The mode shapes Π of the tapered beam were computed using the normal mode basis such that

$$\Pi^T M \Pi = I, \quad (7.2.3)$$

where M is the FEM mass matrix and I is the identity matrix. As a result of the orthogonality in Eq. 7.2.3, it can be shown that the modal integrals p_1^k , E_0^k , and F_0^k described in Appendix E.3 are zero. This simplifies the flexible body equations of motion and eliminates some of the rigid-elastic coupling. Additionally, this indicates that the ensuing elastic motion does not shift the center of mass of the body from its undeformed location. However, the remaining modal integrals \mathcal{J}_1^k , \mathcal{J}_2^k , and F_1^k (see Appendix E.3) are generally non-zero. Therefore, rigid-elastic coupling effects due to inertia, Coriolis, and centrifugal forces remain (see Appendix E.3.1) under this assumption.

Figure 7.2.1 shows the node locations of the tapered beam model in a DARTS/Dshell viewport. The red, green, and blue axes correspond to the roll, pitch, and yaw axes, respectively. The green node indicates the location of the output sensor node. The motion of the flexible body was simulated with no external excitation for 10 sec in two test cases. The first case assumes the initial state of the beam frame is stationary while the first symmetric bending mode is excited. As expected, the body is free to deform while the beam frame remains stationary as shown in Figure 7.2.2. In the second case, the initial rotational rate of the beam frame is $\omega = [5, 5, 5]$ deg/sec, and the translational and modal states are initially stationary. In this case, the deformation at the output sensor node is zero if the modal integrals are removed from the model. However, the plots in Figure 7.2.3 demonstrate the non-zero deformation of the freely rotating flexible body when modal integrals are included in the model. This simple simulation example of a freely rotating flexible

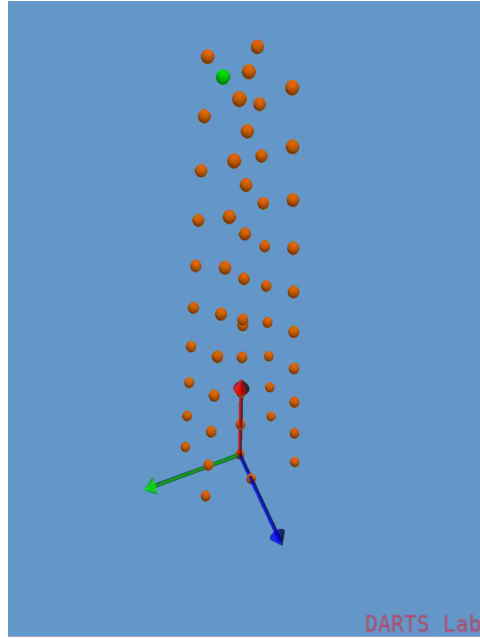
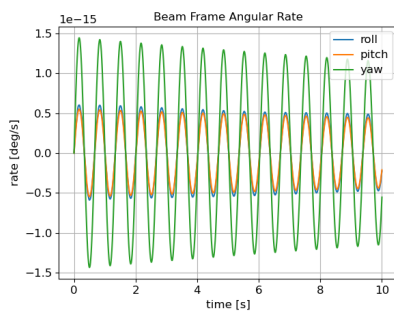
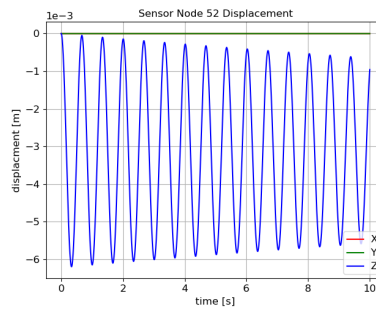


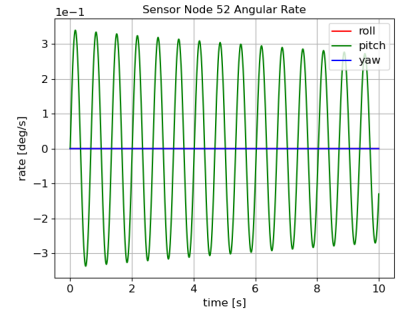
Figure 7.2.1. Tapered Beam Model in DARTS/Dshell



(a) Beam Frame Angular Rate



(b) Sensor Node Displacement



(c) Sensor Node Angular Rate

Figure 7.2.2. Simulation of a Freely Vibrating Tapered Beam Model Using Modal Integrals

body confirms that modal integrals must be included to fully capture the coupling between the elastic and rigid-body motion since, in the absence of modal integral terms, the body behaves as a rigid body and will not deform unless external forces are applied.

The correctness of the modal integrals implemented in the DARTS multibody tool were verified by independently constructing a modal integral-based flexible body model using computer algebra techniques. Given a nonzero initial state, the acceleration of the two models was compared and seen to numerically agree to within 10^{-8} difference. Regression tests were also constructed to ensure the modal integral features continue to produce the expected results as the software evolves.

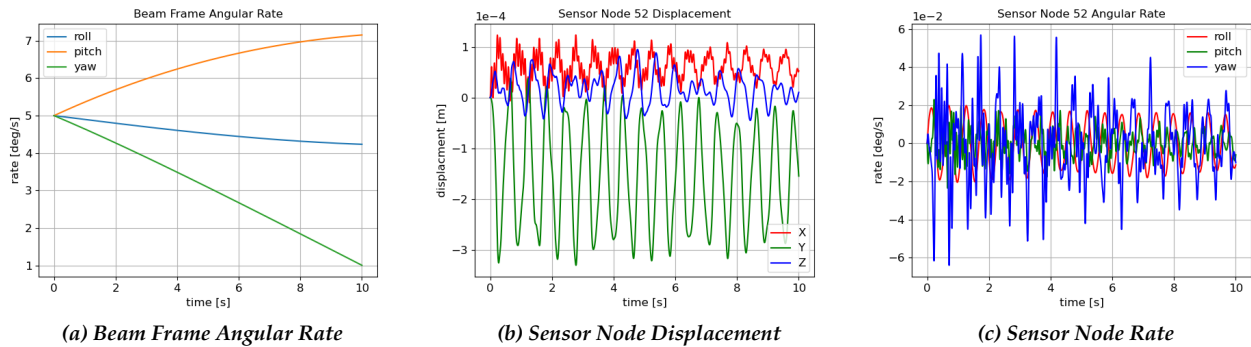


Figure 7.2.3. Simulation of a Freely Rotating Tapered Beam Model Using Modal Integrals

7.3 MSFC/JPL Model Development and Comparisons

To demonstrate the FEM-to-GN&C modeling pipeline approach, the team required a relevant evaluation test problem. Ideally this problem would be representative of a wide range of design challenges NASA encounters and could be configured to incrementally increase model complexity. A launch vehicle was proposed as a test configuration, since this is perhaps one of the simplest examples of a two-body problem when configured as a single missile body and gimballed engine. Problem complexity can be increased by incorporating additional bodies, such as multiple engines, slosh masses, and solid rocket motors.

7.3.1 MSFC Model Description and Heritage

To provide confidence in comparison results, a model traceable to a legacy program was desired. The GN&C structural dynamics model used for the SLS program was initially considered. This model was ultimately rejected to minimize support required from SLS engineers and use a more generic model. Instead, a model developed previously by an MSFC Engineering member of the NESC assessment team was utilized; this model has traceability to large and small launch vehicle programs and has been demonstrated in a tightly coupled structures/GN&C pipeline toolset.

The airframe modeling approach implemented in this tool was first developed in support of an IV&V of the Titan IV flight control system [22]. A Newtonian model was derived from first principles and used to certify each vehicle for flight. The flight control system gain and phase margins generated using this approach were compared to those generated by the prime contractor and by an Air Force IV&V contractor. The tools used by these contractors incorporated different methods of combining the moving masses of the engines into the structural dynamics model.

The next iteration of the dynamics model was derived to support autopilot design and testing for a family of one, two, and three-stage rockets created from surplus Intercontinental Ballistic Missile (ICBM) motors and excess avionics and sensors. These systems were often quite flexible, since they were a mash-up of parts from different vehicles being used to lift payloads much heavier than flown on the original systems. An analysis toolchain was created to support a range of motor configurations, reentry vehicles, and actuation and sensor technologies. These vehicles were launched from ground, sea, and air-based platforms; to support air-launched configurations, the analysis models and flight software were updated to handle significant periods of flight at angles of attack approaching 180 degrees. Since these vehicles often flew over populated areas, the analysis models were extensively tested and verified. The configuration of these systems varied wildly, and the design cycle from program Authority to Proceed (ATP) to flight was typically between 12 and 14 months. Several launch campaigns demonstrated several launches in a single month.

This operations tempo required a tightly integrated toolchain between the structural dynamicists and flight controls teams.

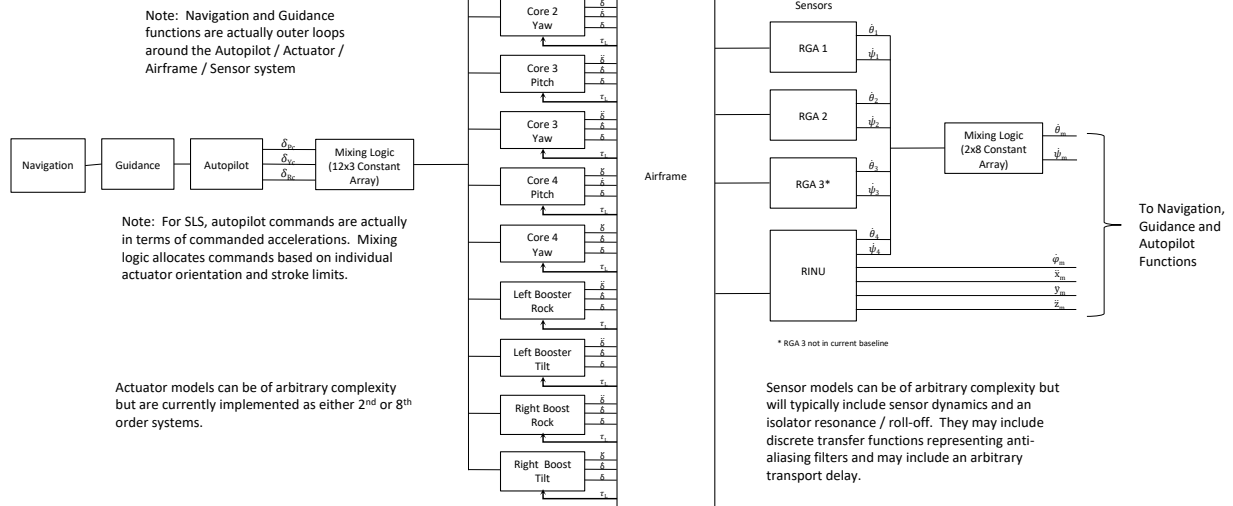
As an exercise to verify understanding of the SLS servo-bending-coupling approach, the MSFC primary investigator began deriving and documenting a structural dynamics model that incorporated lessons learned from the programs described above as a personal project. Because that effort meshed with current work objectives, the tool was enhanced during this NESC assessment for comparison with the DARTS results. Because a high-fidelity comparison was desired, several model updates were required. Minor updates included updating reference frames and sign conventions to match JPL conventions. A more significant update was the inclusion of axial flexible body dynamics and modeling the perturbational changes on system moments of inertia due to engine articulation. In addition, gyroscopic forces due to modeling engine motion in a rotating body frame were incorporated, and the impacts of a steady-state engine deflection on both rigid-body and bending dynamics were included. This model is documented in reference [23].

A launch vehicle is a complex system of interconnected components, including a flexible airframe, gimballed engines, actuation systems, and inertial sensors, all of which are controlled via flight control, navigation, and guidance software. To assess the stability of the vehicle during flight, the performance of each component must be modeled and the effects of external influences such as aerodynamics, thrust, and gravity must be simulated. The airframe model is the focus of this work, but its performance can be assessed only in the context of the overall system. A notional block diagram of some of the important system elements is shown in Figure 7.3.1. Each of the blocks in this figure represents a subsystem that may be described by an arbitrarily complex linear or nonlinear model, and the system may be modeled in either the time or frequency domains. This figure illustrates the system in an open-loop sense: in general, the inertial sensor outputs will be fed back into the blocks representing the Navigation, Guidance, and Autopilot functions. The airframe model used must support the architecture described in the block diagram. One of the most common areas of miscommunication between disciplines is the interface between the actuation system(s) and the airframe; this interface looks simple on a block diagram, but actually involves dynamically coupling three mechanical elements (engine, actuator, and airframe) and the joints between them. Modeling this interface correctly was at the heart of the assessment goals. As explained in more detail in Appendix F, a key detail affecting the interface between GN&C and structural dynamic models is whether the engines (or other bodies) are included in the missile FEM model. There are valid reasons for formulating a FEM with or without the engines included, but it is important that both disciplines use the same assumptions.

As described in Appendix F, it is important to verify the operation of the airframe model with prescribed engine motion (load torque feedback open) and commanded torque inputs (load torque feedback closed). The prescribed engine motion case assumes the actuators are capable of generating arbitrarily large torques as required to obtain a desired engine excitation. This test case is useful for verifying the airframe model in isolation since the engine drives the airframe but not vice-versa. The torque input test case is useful for verifying proper modeling of the interactions between the airframe and engines, since in this case the gimbal acts as a frictionless bearing to disturbances and any airframe modeling error will drive engine motion.

To focus on the critical features of the coupled engine/airframe system, the configuration used to evaluate the models is illustrated in Figure 7.3.2. The system is evaluated as a single-input-multiple-output system. For pitch test cases, the yaw command is set to zero and vice versa. The load torque feedback connections are turned on or off using the gains K_{LT_p} and K_{LT_y} .

NOTE: This is not an official SLS model or analysis tool. It represents a tool I have used to design, analyze and simulate other flexible vehicles in the past. Previous implementations have been validated on other programs.



The diagram illustrates a flight control system with the following components and signal flow:

- Pitch Engine:** Receives a **Pitch Cmd** input. It contains a feedforward path with gain $\frac{1}{I_{gtm}}$ and two feedback paths with integrators $\frac{1}{s}$. The output is δ .
- Yaw Engine:** Receives a **Yaw Cmd** input. It contains a feedforward path with gain $\frac{1}{I_{gtm}}$ and two feedback paths with integrators $\frac{1}{s}$. The output is δ .
- Airframe:** Receives δ from both engines. It outputs a vector of states: $\bar{\theta}, \dot{\psi}, \ddot{\psi}, \ddot{\theta}, \psi, \dot{\phi}, \theta, \psi, \varphi, \ddot{x}, \ddot{y}, \ddot{z}, \dot{x}, \dot{y}, \dot{z}, x, y, z$.
- Feedback Loops:**
 - The **roll rate** $\dot{\psi}$ is measured and fed back to the Pitch Engine.
 - The **yaw rate** $\dot{\phi}$ is measured and fed back to the Yaw Engine.
 - The **roll rate** $\dot{\psi}$ is also fed back to the Yaw Engine.
 - The **yaw rate** $\dot{\phi}$ is also fed back to the Pitch Engine.
- Control Gains:**
 - K_{LTP} (Roll Rate Feedback Gain) is applied to the roll rate feedback signal.
 - K_{LTY} (Yaw Rate Feedback Gain) is applied to the yaw rate feedback signal.
- Disturbance:** **Airframe Load Torque** is an input to the system, which is fed back to the Yaw Engine.
- Other Signals:**
 - $\delta_{uc} = 0$ is an input to the Airframe.
 - τ_{LP} and τ_{LY} are outputs from the Airframe.

7.3.2 Overview of Results

Restricted distribution to NESC and designated team members until approved by the NRB.

frequency domain. The most complex test case simulated a highly asymmetric flexible structure driven by an engine with large center of gravity offsets. Transfer functions examined were body linear and angular accelerations and rates as a function of engine motion. A metric of ± 0.001 dB was established as the desired matching criterion. Based on team members' experience with IV&V for GN&C, independent tools that compare to within 0.1 dB are considered to have an excellent match. Because this assessment is considering potentially small modeling differences such as the inclusion of modal integrals, a matching tolerance of 100x better than this was selected, or 0.001 dB. This level of accuracy is far higher than that required for autopilot design and analysis. Most responses were well within this tolerance, and any exceptions explained. This effort provides a high level of confidence that the tool can be used to generate high-fidelity structural dynamics models for use in launch vehicle control systems analyses. A complete summary of the tool comparison results and findings can be found in Appendix F, and some representative results are presented here.

Figure 7.3.3 is an overlay of results generated by JPL DARTS/Dshell and MSFC for the primary responses of Test Case 2 (see Table F1 for case definition). MSFC results are plotted with a solid blue line, JPL DARTS/Dshell results with a dotted red line. The responses match closely enough that the plot appears as a single purple trace.

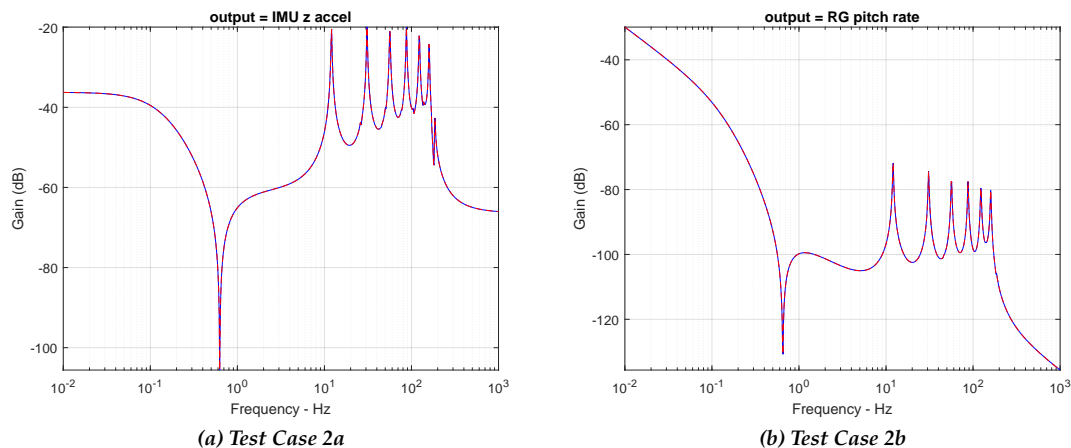


Figure 7.3.3. Overlay of DARTS and MSFC Acceleration and Rate Responses at Sensor Location, Test Cases 2a and 2b

Figure 7.3.4 is a plot of the differences between the JPL DARTS/Dshell and MSFC results; note the y axis limits of $\pm 2 \times 10^{-3}$ dB. This quality of matching is common to most of the responses, as discussed in detail in Appendix F. Of note is the spike in response error at the airframe tail-wags-dog zero and the slight (2×10^{-4} dB) low-frequency error due to small differences in calculated initial conditions.

During the JPL and MSFC tool comparison effort, it was observed that to achieve the desired matching criterion of 0.001 dB it was important to accurately model initial conditions on both rigid body and modal states. This effect is especially prominent in the complex torque-input test cases since the frictionless gimbal results in a relative steady-state angular acceleration between missile body and engine. In addition, the off-axis responses are much more sensitive to this effect; i.e. the yaw and roll responses to a pitch plane input. Figure 7.3.5 illustrates the effect of the initial engine angular accelerations on the x-acceleration and roll rate response to a pitch torque input (Table F1, Case 2). The solid line is an overlay of JPL and MSFC results with the engine angular acceleration state properly initialized. The dashed red line illustrates the MSFC results with initial engine angular accelerations zeroed. Even small errors in the initial state resulted in violations

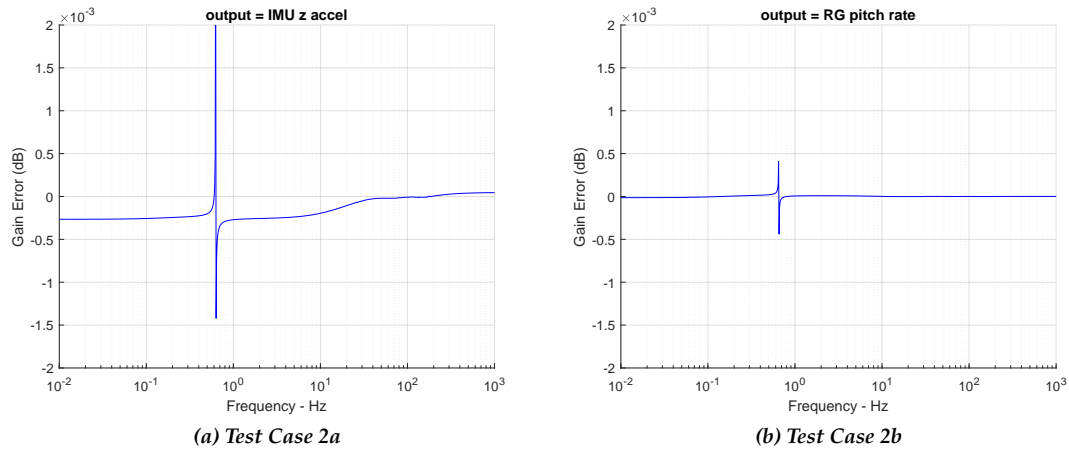


Figure 7.3.4. Zoomed Differences of DARTS and MSFC Acceleration and Rates Responses at Sensor Location, Test Cases 2a and 2b

of the 0.001 dB matching criterion. Appendix F includes additional discussion of the initial conditions on rigid body responses.

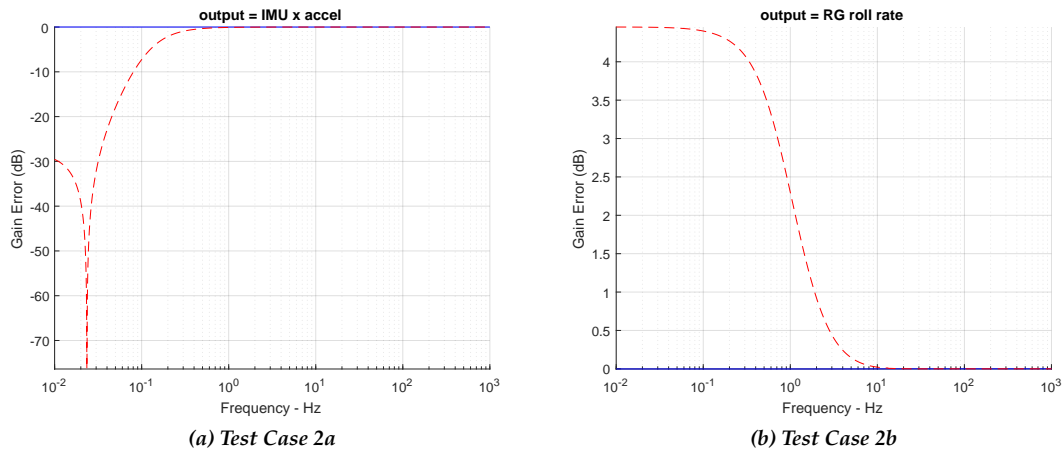


Figure 7.3.5. Comparison of the Effect of Neglecting Initial Engine Accelerations on Off-axis Responses, Test Case 2

Perhaps a more interesting observation is the effect of modal state initialization as illustrated in Figure 7.3.6, which depicts the response of the pitch gimbal angle to a pitch torque input. The figure on the left illustrates the response with the flexible launch vehicle initially undeformed while the figure on the right shows the response with the airframe initialized in a steady-state condition, compressed by the propulsive thrust force of the engine. When initialized in the undeformed state, the linear model exhibits large resonant engine dynamics, but when it is initialized in the deformed state this behavior is not present. Fundamentally, this behavior is a result of linearizing the system at different operating points such that the system has non-zero steady-state modal accelerations when initialized in the undeformed state while the initial modal accelerations are zero in the deformed state. This illustrates the importance of trim initial condi-

tions when developing linear flexible body dynamics models, since small deviations in modal coordinate states can lead to large variations in computed linear models. Note that this caution applies to models that do not incorporate a modelling assumption that the system bending dynamics are in a steady-state before computing the linear model parameters.

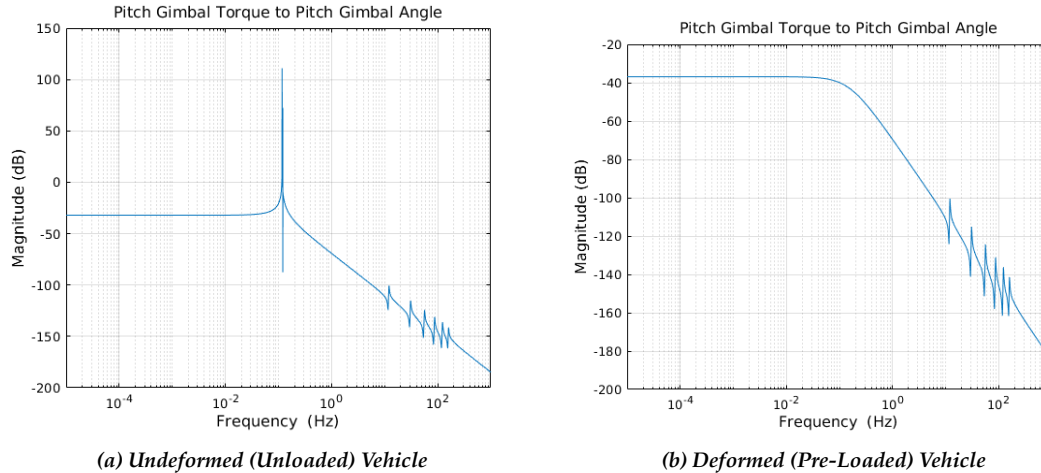


Figure 7.3.6. Comparison of Engine Dynamics FRF in an Undeformed and Deformed Vehicle State

In addition to careful consideration of frequency response functions (FRF) initial conditions, the JPL and MSFC comparison effort found that truncating NASTRAN modal data to single precision resulted in larger than expected frequency response errors. This issue can crop up when, for example, modal data is extracted from a NASTRAN .f06 ASCII output file. The default .f06 output file will report modal data accurate to 6 decimal places (single precision). Figure 7.3.7 illustrates frequency response errors due to the use of single precision modal data. For this example, the error due to the use of single precision modal data (red curve) exceeded the ± 0.001 dB error tolerance metric. Therefore, modal data precision should be retained whenever possible.

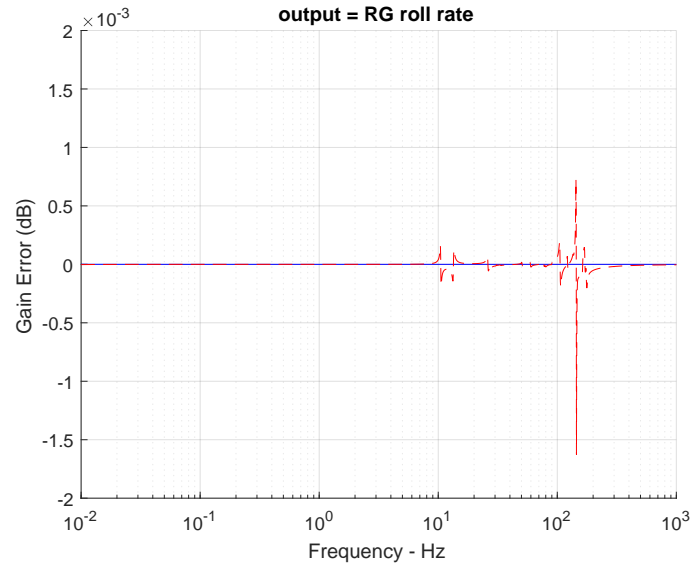


Figure 7.3.7. Comparison of Single and Double Precision Modal Data for Yaw Engine Torque Input to Roll Rate Output (the blue curve is the error with double precision modal data; the red curve is the error with single precision modal data)

7.4 Demonstration for Large Space-Based Observatory Study

The FEM-to-GN&C modeling pipeline was used to construct linear state space models in support of NESC Assessment TI-18-01371, “Microthruster Systems for Low-Jitter Space Observatory Precision Attitude Control.” That team required a representative flexible body model of a space telescope to assess the feasibility of using microthrusters as a pointing actuator. The team used the pipeline to rapidly produce linear-state space models using FModal to integrate an FEM model of the LUVOIR-A 15 m concept spacecraft, and DARTS/Dshell to add flexible body dynamics, actuator dynamics (reaction wheels and thrusters), and sensor outputs (i.e., inertial measurement unit, or IMU).

The FEM model of the LUVOIR-A spacecraft was delivered by GSFC [24] in NASTRAN bulk data format. This preliminary model contained no actuator models or even grid point definitions for actuator/sensor locations. To realize a representative GN&C system, 4 reaction-wheel and 16 thruster models were created in a Dshell simulation along with the flexible body dynamics generated using the output data from FModal. The Dshell simulation was used to linearize the system, assuming proportional damping ratios of 0.0025, 0.005, and 0.01 across the structure. In the undamped case, the linearized Dshell model was compared with the NASTRAN model as shown in Figure 7.4.1. The difference between the NASTRAN and DARTS model system frequencies was commensurate with the finite difference truncation error used to linearize the DARTS model. An example frequency response of an applied force at thruster 1 to a pitch angular rate at the boresight node is also shown in Figure 7.1.1.

Three linear state space models were delivered to The Aerospace Corporation for use in a MAT-LAB/Simulink GN&C closed-loop simulation to assess the relative pointing stability performance of space-craft systems using microthrusters and reaction wheels. Appendix G describes the details of the microthruster study and the application of the pipeline to convert the LUVOIR-A FEM to a GN&C model.

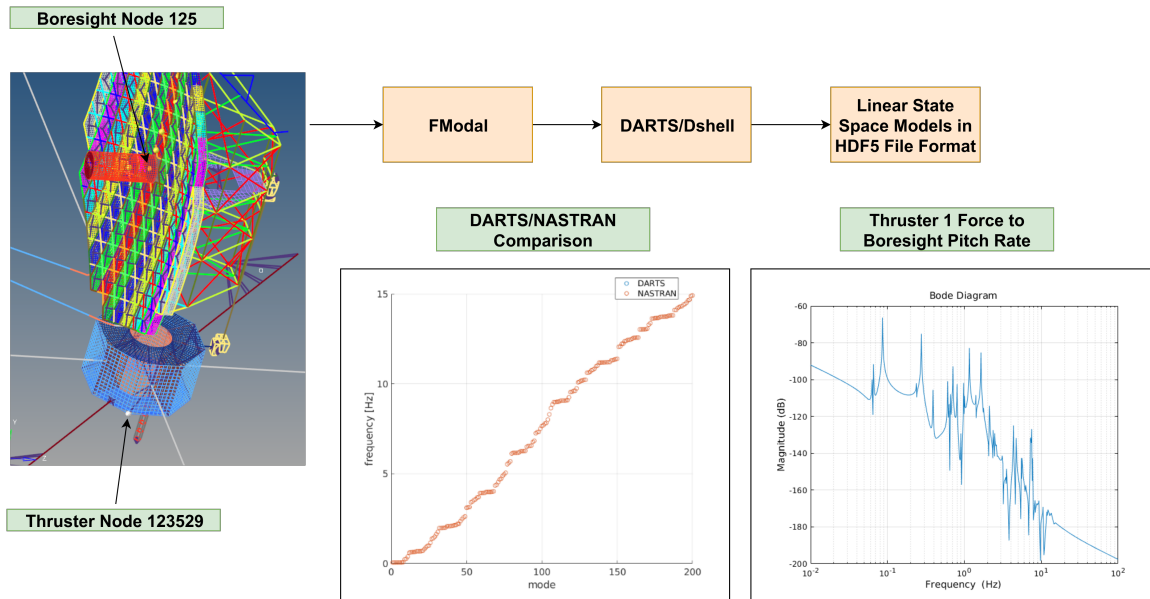


Figure 7.4.1. GN&C Linear State Space Modeling Using the LUVOIR-A FEM

8.0 Findings and NESC Recommendations

8.1 Findings

The following findings were identified:

- F-1** A multibody dynamics approach with the FEM-to-GN&C pipeline developed under this assessment demonstrated its ability to rapidly produce GN&C system-level models.
- F-2** Small deviations in trim initial conditions can lead to larger than expected variations in the linearization of multibody systems with coupled nonlinear rigid and linear flexible body dynamics.
- F-3** Loss of data precision in a FEM output file can result in significant numerical errors in the GN&C model.
- F-4** Frequency response functions for GN&C flexible multibody dynamics models can be sensitive to geometry, mass properties, or where a control loop is broken. Careful consideration of these sensitivities aid in understanding GN&C flexible multibody dynamics tool differences.
- F-5** To fully exercise the interactions between bodies in the GN&C flexible body dynamics tool comparison, it is necessary to test the tools using both prescribed motion and torque inputs.

8.2 NESC Recommendations

The following NESC recommendations are directed toward analysts responsible for providing, integrating, or working with flexible body dynamics models.

- R-1** Consider use of a multibody dynamics approach with the FEM-to-GN&C pipeline early in a project's life cycle to mitigate the need for project-specific and repeated model development, updates, and validation efforts as the project evolves. (**F-1**)

R-2 Incorporate flight-like initial conditions (i.e., pre-load launch vehicle model with the nominal acceleration vector) into nonlinear rigid multibody model with linear flexible body dynamics when conducting numerical linearization. (F-2)

R-3 Provide modal data in a file format that ensures sufficient numerical precision is preserved. (F-3)

9.0 Alternative Viewpoints

No alternative viewpoints were identified during the course of this assessment by the NESC assessment team or the NRB quorum.

10.0 Other Deliverables

The analysis models and associated software developed for FModal and DARTS/Dshell during the assessment will be made available via JPL standard software release processes.

Flexible body dynamics models were delivered in support of NESC Assessment TI-18-01371, “Microthruster Systems for Low-Jitter Space Observatory Precision Attitude Control.” The filenames of the delivered models are listed below:

- luvoira_linear_0p0025_06032019.h5
- luvoira_linear_0p005_06032019.h5
- luvoira_linear_0p01_06032019.h5

11.0 Lessons Learned

No lessons learned were identified during the course of this assessment.

12.0 Recommendations for NASA Standards and Specifications

No recommendations for NASA standards and specifications were identified as a result of this assessment.

13.0 Definition of Terms

Finding: A relevant factual conclusion and/or issue that is within the assessment scope and that the team has rigorously based on data from their independent analyses, tests, inspections, and/or reviews of technical documentation.

Recommendation: A proposed measurable stakeholder action directly supported by specific Finding(s) and/or Observation(s) that will correct or mitigate an identified issue or risk.

14.0 Acronyms and Nomenclature

ATP Authority to Proceed

CSI	Control-Structure Interaction
dB	Decibel
DARTS	Dynamics and Real-Time Simulation
DMAP	Direct Matrix Abstraction Program
DOF	Degrees of Freedom
Dshell	DARTS Lab software and simulation interface
EITM	Engines-In-The-Modes
EOTM	Engines-Out-of-The-Modes
FEM	Finite Element Method
FRF	Frequency Response Functions
GN&C	Guidance, Navigation and Controls
GRAM	Global Reference Atmospheric Model
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
HDF5	Hierarchical Data Format 5
Hz	Hertz
ICBM	Intercontinental Ballistic Missile
ICPS	Interim Cryogenic Propulsion Stage
IMU	Inertial Measurement Unit
IMT	Integrated Modal Test
IV&V	Independent Verification and Validation
JPL	Jet Propulsion Laboratory
kg	kilogram
KSC	Kennedy Space Center
LaRC	Langley Research Center
LH2	Liquid Hydrogen
LIDAR	Light Detection and Ranging
LO2	Liquid Oxygen
LQR	Linear Quadratic Regulator
LUVOIR	Large Ultraviolet Optical Infrared Surveyor
LVSA	Launch Vehicle Stage Adaptor
m	meter
MATLAB	Matrix Laboratory
MPCV	Multi-Purpose Crew Vehicle

M&S	Modeling and Simulation
MSFC	Marshall Space Flight Center
N	Newton
NASA	National Aerospace and Space Administration
NESC	NASA Engineering and Safety Center
NASTRAN	NASA STRucture ANalysis
OP2	Output 2
POST	Program to Optimize Simulated Trajectories
SLS	Space Launch System
SOA	Spatial Operator Algebra
STS	Space Transportation System
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
WFIRST	Wide Field Infrared Survey Telescope

15.0 References

- [1] A. Greensite. "Analysis and Design of Space Vehicle Flight Control Systems, Vol. XV-Elastic Body Equations". Convar Division Report GDC-DDE65-025, August, 1967.
- [2] J.L. Lagrange. *Mecanique Analytique*. Paris: Mme Ve Courcier, 1787.
- [3] J.G. Papastavridis. *Analytical Mechanics*. Oxford University Press, 2002.
- [4] F.E. Udwadia and R.E. Kalaba. *Analytical Dynamics: A New Approach*. Cambridge University Press, 1996.
- [5] D.T. Greenwood. *Advanced Dynamics*. Cambridge University Press, 2003.
- [6] T.R. Kane and C.F. Wang. "On the derivation of equations of motion". In: *Journal of the Society of Industrial and Applied Mechanics* 13 (2 1965), pp. 487–492.
- [7] M.G. Hollars, D.E. Rosenthal, and M.A. Sherman. *SD/FAST User's Guide B.2: Symbolic Dynamics, Inc.* 1994.
- [8] A.D. Schutte and F.E. Udwadia. "New approach to the modeling of complex multibody dynamical systems". In: *Journal of Applied Mechanics* 78 (2 2011).
- [9] A.A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, 2013.
- [10] R.A. Wehage and E.J. Haug. "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of of Constrained Dynamic Systems". In: *ASME J. Mech. Des.* 104 (1 1982), pp. 247–255.
- [11] Guillermo Rodriguez, Abhinandan Jain, and K Kreutz-Delgado. "A spatial operator algebra for manipulator modeling and control". In: *INTRR* 10.4 (1991), p. 371. URL: <http://ijr.sagepub.com/cgi/content/abstract/10/4/371>.
- [12] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer Verlag, 2008.
- [13] Abhinandan Jain. *Robot and Multibody Dynamics: Analysis and Algorithms*. Springer, 2011, p. 512. ISBN: 1441972668. DOI: 10.1007/978-1-4419-7267-5. URL: <http://www.springerlink.com/content/978-1-4419-7266-8/contents/>.
- [14] J. Cameron et al. "DSEND: Multi-mission Flight Dynamics Simulator for NASA Missions". In: (Long Beach, CA). AIAA SPACE Conference and Exposition, 2016. DOI: 10.2514/6.2016-5421.

- [15] H. Grip et al. "Flight Control System for NASA's Mars Helicopter". In: (San Diego, CA). AIAA SciTech, 2019. DOI: 10.2514/6.2019-1289 .
- [16] A. Jain. "DARTS - Multibody Modeling, Simulation and Analysis Software". In: (Duisburg, Germany). Proceedings of the 9th ECCOMAS Thematic Conference on Multibody Dynamics, 2019. DOI: 10.1007/978-3-030-23132-3_52 .
- [17] A. Jain. "Unified Formulation of Dynamics for Serial Rigid Multibody Systems". In: *Journal of Guidance, Control, and Dynamics* 14 (1991), pp. 531–542.
- [18] J.R. Brouwer and V.J. Cabezas-Tapia. "The Equations of Motion of a Flexible Missile with Two Different Structural Model Implementations". Version 1.8, November 5, 2019.
- [19] L.F. Richardson. "The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam". In: *Philosophical Transactions of the Royal Society of London Series A* 210 (459-470 1911), pp. 307–357.
- [20] J. Sills. "DTaMSS R0 Model - Integrated Modal Test (IMT) Configuration". FEM model generated at JSC Sep 21 2017.
- [21] A.A. Shabana and R. Schwertassek. "Equivalence of the floating frame of reference approach and finite element formulations". In: *Intl. J. Non-Linear Mechanics* 33.3 (1998), pp. 417–432.
- [22] J.R. Brouwer, T.W. Fey, and W. Kelly. "Equations of Motion of Flexible Missile". Martin Marietta Missile Systems Report ANA-102000010-013A, Orlando, FL, March, 1993.
- [23] J.R. Brouwer. "The Rigid Body Equations of Motion of a Missile with a Gimballed Engine". Version 1.7, December 11, 2019.
- [24] C. Collins. "LUVOIR-A 15-m Jitter Monolith FEM". FEM model generated at GSFC Oct 17 2018.
- [25] R.R. Craig and M.C. Bampton. "Coupling of substructures for dynamic analyses". In: *AIAA Journal* 14 (1968), pp. 1313–1319.
- [26] F.E. Udwardia and A.D. Schutte. "Equations of motion for general constrained systems in Lagrangian mechanics". In: *Acta Mechanica* 213 (1 2010), pp. 111–129.
- [27] Abhinandan Jain. "Structure Based Modeling and Computational Architecture for Robotic Systems". In: *2013 IEEE International Conference on Robotics and Automation*. 2013.
- [28] Abhinandan Jain and Guy K Man. "Real-Time Simulation of the Cassini Spacecraft Using DARTS: Functional Capabilities and the Spatial Algebra Algorithm". In: *5th Annual Conference on Aerospace Computational Control*. Jet Propulsion Laboratory, Pasadena, CA. 1992.
- [29] *Dynamics and Real-Time Simulation (DARTS) Lab*. \url{http://dartslab.jpl.nasa.gov/}. 2019. URL: <http://dartslab.jpl.nasa.gov> .
- [30] *The HDF Group*. <https://www.hdfgroup.org/> . Accessed: 2020-01-21.
- [31] J. Orr, J. Wall, and T. Barrows. "Simulation-Based Analysis and Prediction of Thrust Vector Servoelastic Coupling". In: (Breckenridge, Colorado). Proceedings of the 43rd Annual AAS Guidance, Navigation and Control Conference, 2020.
- [32] A. Jain. *Robot and Multibody Dynamics: Analysis and Algorithms*. Springer, 2011.
- [33] John W. Eaton. *GNU Octave*. <https://www.gnu.org/software/octave/> .

Appendices

- A. Component Body Modal Analysis
- B. Flexible Body Dynamics Modeling Approaches for GN&C
- C. FModal: A FEM-to-GN&C Flexible Multibody Modeling Pipeline Tool
- D. System-Level Modeling and Simulation
- E. Modal Integrals for Flexible Bodies
- F. MSFC/JPL Launch Vehicle Model Validation
- G. Flexible Spacecraft Modeling for Microthruster Assessment

Appendix A. Component Body Modal Analysis

Aerospace vehicles are often composed of multiple bodies, each of which may demonstrate coupling between rigid and flexible motion. Nonlinear multibody models for these systems require flexible body models for each of the flexible component bodies. This section describes common methods for modeling the deformation of a typical component body within the multibody system, beginning with floating frame models consisting of differential masses (also referred to as nodes) typically derived using finite element methods. Since such nodal models can be very large, assumed modes are typically used for model reduction for GN&C analysis. We describe the modal approach initially for an unconstrained body, followed by the component mode synthesis approach for handling the coupling constraints on the component bodies in a multibody system.

The finite element method is an essential technique used in structural engineering to discretize the governing continuous system that describes the time and spatial evolution of a structure. The resulting FEM model is typically large since the number of finite elements used to mesh the spatial domain must be large enough to capture the complex geometry of the structure. It is assumed that the body deformation is small so that linear elasticity theory can be used to derive a linear multi-DOF system. The linear, or small, deformation assumption assumes that the stress-strain behavior of the structure material remains linear over the expected loading conditions.

A.1 Modal Models for an Unconstrained Body

Using a floating frame of reference approach [21], the equations of motion of an undamped n_{nd} -node FEM model for a single flexible body take the following form

$$M\ddot{u}_{nd} + Ku_{nd} = F, \quad (A.1.1)$$

where M and K are $6n_{nd}$ by $6n_{nd}$ matrices and u_{nd} and F are $6n_{nd}$ -vectors. The physical coordinates u_{nd} contain the 6-dimensional rotational and positional deformation values for each of the nodes. Eq. A.1.1 is generally coupled since either M or K can be non-diagonal. Unfortunately, direct use of such FEM models is not practical for most GN&C analyses since the model can contain millions of DOF. Furthermore, the FEM model is valid only for a single configuration. This poses a challenge for GN&C design and analysis activities, especially in a multibody system where a large range of configurations must be analyzed. The non-tractability of the full FEM model is typically overcome using modal analysis. Modal analysis is used to significantly reduce the number of DOF of the full FEM model. However, the utility of the modal based model must be considered with care in the multibody context due to the coupling constraints between the bodies.

Commonly, the multibody dynamics formulation used by a GN&C analyst will typically assume that the deformation of the component bodies under consideration follows linear elasticity theory. In this case, the deformation of the body is approximated using modal analysis where it is assumed that the elastic deformations of the j -th differential masses (or nodes) on the k^{th} body take the form

$$u_{nd}(\mathcal{O}_k^j) \stackrel{E.1.19}{=} \Pi(\mathcal{O}_k^j) \eta(k) \approx \sum_{r=1}^{n_{md}(k)} \Pi_r(\mathcal{O}_k^j) \eta_r(k), \quad (A.1.2)$$

where \mathcal{O}_k^j is the origin of the j -th node of body k . Coincident with the j -th differential mass, $\Pi(\mathcal{O}_k^j)$ is a spatially dependent 6 by $n_{md}(k)$ shape matrix of basis vectors, and η is a time-dependent $n_{md}(k)$ -vector of generalized modal coordinates. This is the classical Rayleigh-Ritz discretization, which allows use of the admissible vector functions $\Pi_r(\mathcal{O}_k^j)$, $r = 1, \dots, n_{md}(k)$ to approximate continuous systems. By admissible

functions, it is assumed that $\Pi_r(\mathcal{O}_k^j)$, $r = 1, \dots, n_{md}(k)$, (1) forms a linearly independent set, (2) has continuous derivatives up to the order appearing in its corresponding strain energy relation, and (3) satisfies all geometric boundary conditions. We call $\Pi_r(\mathcal{O}_k^j)$ the r -th spatial mode shape and η_r the r -th modal coordinate of the j -th differential mass on the k^{th} body. Modal coordinates have significant benefits in terms of model reduction. Once in modal form, the model is readily reduced by simply removing the modes that are not significant for the motion under consideration. For large FEM models, this reduction can be greater than 10^6 DOF. Extraction of the mode shapes and corresponding modal frequencies from the FEM model is one of the primary tasks in developing a multibody dynamics model for GN&C analysis. In the rest of this appendix, most references to the k^{th} body will be dropped to simplify notation, keeping in mind that the material is general and is for a representative body.

The normal mode transformation

$$u_{nd} = \Pi \eta \quad (A.1.3)$$

can be used to diagonalize the system, where the normal "free-free" mode shapes Π are found as the eigenvector solutions to the eigenvalue problem

$$(K - \zeta_i M)[\Pi]_i = 0, \quad i = 1, \dots, 6n_{nd}. \quad (A.1.4)$$

In Eq. A.1.4, the i -th normal mode shape $[\Pi]_i$ corresponds to the i -th eigenvalue ζ_i , where ζ_i is the square of the i -th modal frequency. The normal mode shapes diagonalize the system in Eq. A.1.1 so that

$$\Pi^T M \Pi \ddot{\eta} + \Pi^T K \Pi \eta = \Pi^T F \quad (A.1.5)$$

is a set of $6n_{nd}$ state independent equations of motion. The mode shapes Π are typically normalized so that

$$\Pi^T M \Pi = I \quad (A.1.6)$$

and

$$\Pi^T K \Pi = \zeta, \quad (A.1.7)$$

where ζ is a diagonal matrix containing the squares of the modal frequencies. This well-known transformation is widely used in GN&C analysis (when normal free-free modes are valid) to generate a reduced-order flexible body model, i.e., Eq. A.1.5 is readily reduced by simply removing the modal DOF that contain frequency content above a pre-determined cutoff frequency or have small influence coefficients for the input/output behavior of interest.

A.2 Modal Models for an Attached Body

In the context of a multibody system, the coupling between individual component bodies must be considered. This coupling requires the satisfaction of inter-body constraints. Due to the coupling constraints, normal mode shapes with free-free modal basis are not well-suited since a large number of such modes are required to account for the attachment constraints. The following sections describe the component mode synthesis alternative approach to selecting modes that are suited to handling body attachment constraints.

A.2.1 Component Mode Synthesis

Component mode synthesis uses modes that satisfy attachment constraints and allows the development of multibody models with articulated appendages without having to re-run a normal modes analysis as bodies articulate. The Craig-Bampton transformation [25] is one of the methods developed for component mode synthesis. This particular transformation permits the generation of flexible body models with a small number of modal DOF, where the model can capture the critical low-frequency modes of the structure while leaving the boundary (interface) nodes in physical coordinate form. This is convenient for component FEM modeling since the interface DOF are readily available to couple with other substructures where clamped

boundary conditions are assumed at the interface nodes. The Craig-Bampton transformation is obtained by partitioning the n_{nd} -node FEM model as

$$\begin{bmatrix} M_e & M_{eb} \\ M_{be} & M_b \end{bmatrix} \begin{bmatrix} \ddot{u}_{nd}^e \\ \ddot{u}_{nd}^b \end{bmatrix} + \begin{bmatrix} K_e & K_{eb} \\ K_{be} & K_b \end{bmatrix} \begin{bmatrix} u_{nd}^e \\ u_{nd}^b \end{bmatrix} = \begin{bmatrix} F_e \\ F_b \end{bmatrix} \quad (A.2.1)$$

such that

$$u_{nd} := \begin{bmatrix} u_{nd}^e \\ u_{nd}^b \end{bmatrix} = \begin{bmatrix} \Pi_e & \Pi_b \\ 0 & I \end{bmatrix} \begin{bmatrix} \eta \\ u_{nd}^b \end{bmatrix} := \Pi_c u_{nd}^c. \quad (A.2.2)$$

In Eq. A.2.2, u_{nd}^e are the interior DOF, u_{nd}^b are the boundary DOF, η are the modal coordinates, Π_e are the fixed-base mode shapes and Π_b are the so-called constraint modes. The constraint modes Π_b are obtained by evaluating the static configuration ($\ddot{u}_{nd}^e = \ddot{u}_{nd}^b = \eta = 0$) of Eq. A.2.1 so that

$$u_{nd}^e \Big|_{\ddot{u}_{nd}^e = \ddot{u}_{nd}^b = \eta = 0} = -K_e^{-1} K_{eb} u_{nd}^b := \Pi_b u_{nd}^b. \quad (A.2.3)$$

The fixed-base mode shapes in the columns of Π_e are obtained by constraining the boundary DOF ($\ddot{u}_{nd}^b = u_{nd}^b = 0$) and by assuming there are no applied forces on the interior DOF ($F_e = 0$). This yields the equation of motion

$$M_e \ddot{u}_{nd}^e + K_e u_{nd}^e = 0. \quad (A.2.4)$$

The columns of Π_e are then the eigenvector solutions to the eigenvalue problem

$$(K_e - \zeta_i M_e) [\Pi_e]_i = 0, \quad i = 1, \dots, n_{md}(k), \quad (A.2.5)$$

where the i -th mode shape $[\Pi_e]_i$ corresponds to the i -th eigenvalue ζ_i . These mode shapes are typically normalized so that

$$\Pi_e^T M_e \Pi_e = I \quad (A.2.6)$$

and

$$\Pi_e^T K_e \Pi_e = \zeta_e, \quad (A.2.7)$$

where ζ_e is a diagonal matrix containing the squares of the fixed-base modal frequencies. Given Π_e and Π_b , the Craig-Bampton coordinate transformation (Eq. A.2.2) yields the equations of motion

$$\Pi_c^T M \Pi_c \ddot{u}_{nd}^c + \Pi_c^T K \Pi_c u_{nd}^c = \Pi_c^T F, \quad (A.2.8)$$

where M , K and F are the mass matrix, stiffness matrix and force vector in Equation A.1.1.

A.2.2 Normalizing the Component Modes

It is important to recognize that Eqs. A.1.1, A.1.5, and A.2.8 are all equivalent if all modal DOF are retained in the model. However, in contrast to Eq. A.1.5, modal reduction of Eq. A.2.8 yields a model that can be constrained at the interface nodes with fixed boundary conditions. Unfortunately, though, the form of Eq. A.2.8 is not amenable to integration into a multibody dynamics formulation because the interface nodes remain in physical coordinate form and the rigid body modes are not readily removable. Assuming the columns of Π_c have been truncated to reduce the number of fixed-base mode shapes, a normalization of Eq. A.2.8 can be obtained by solving the eigenvalue problem

$$(K_n - \zeta_i M_n) [\hat{\Pi}_n]_i = 0, \quad i = 1, \dots, m, \quad (A.2.9)$$

where

$$M_n = \Pi_c^T M \Pi_c \quad (\text{A.2.10})$$

and

$$K_n = \Pi_c^T K \Pi_c. \quad (\text{A.2.11})$$

The coordinate transformation

$$u_{nd} = \Pi_n \eta := \Pi_c \hat{\Pi}_n \eta \quad (\text{A.2.12})$$

then yields

$$\Pi_n^T M \Pi_n \ddot{\eta} + \Pi_n^T K \Pi_n \eta = \Pi_n^T F, \quad (\text{A.2.13})$$

where

$$\Pi_n^T M \Pi_n = I \quad (\text{A.2.14})$$

and

$$\Pi_n^T K \Pi_n = \zeta_n. \quad (\text{A.2.15})$$

Since Eq. A.2.13 is diagonal, the rigid body modes are readily removed and the mode shapes Π_n can be used to constrain the fixed interface nodes.

A.3 Five-Particle Spring-Mass Example

A simple spring-mass system is presented in this section to supplement the component body modal analysis theory given in the previous two sections. Consider a one-dimensional, unconstrained spring-mass system consisting of five masses and four springs with no external forces, as shown in Figure A.3.1 . The

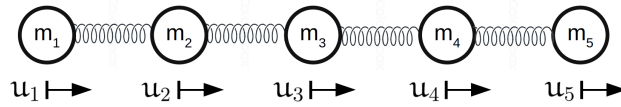


Figure A.3.1. Spring mass system

coordinates u_1 through u_5 give the displacement of the masses relative to their starting positions, which are chosen such that all springs begin unstretched. The equations of motion for this system are written concisely in Eq. A.3.1.

$$M \ddot{u}_{nd} + K u_{nd} = 0$$

$$M = mI, \quad K = k \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad (\text{A.3.1})$$

where for this example $n_{nd} = 5$, $u_{nd} = \{u_1, u_2, u_3, u_4, u_5\}^T$, $m = 2$ is the mass of each of the five masses, and $k = 1$ is the spring constant (stiffness) for each spring. The free-free mode shapes for this system can be obtained by solving the eigenproblem from Eq. A.1.4.

$$(K - \zeta_i M) [\Pi]_i = 0, \quad i = 1, \dots, 5.$$

Further, suppose that in the flexible dynamics simulation the system is to be simulated with the constraint $u_1 = 0$. The fixed-base modes for the constrained system are obtained from the eigenproblem,

$$(K_e - \zeta_i M_e) [\Pi_e]_i = 0, \quad i = 1, \dots, n_{md}(k),$$

where $n_{md}(k) = 4$,

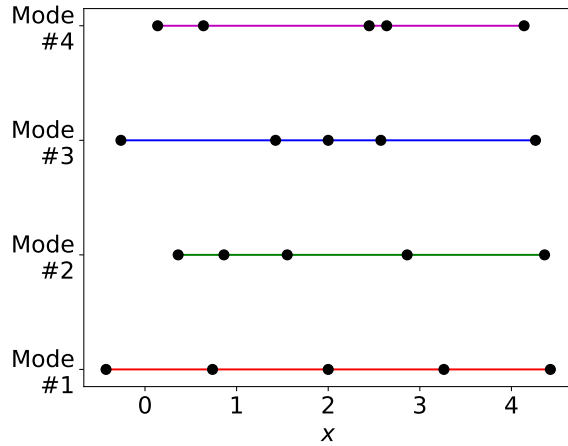
$$M_e = mI, \quad \text{and} \quad K_e = k \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

As explained in section A.2, the component modes and normalized component modes for the system can be acquired via the Craig-Bampton transformation. Moreover, as stated previously, model reduction can be performed on the normalized component mode system, Eq. A.2.13, and the result can still be constrained at the interface nodes. To demonstrate the utility of the normalized component mode system, four comparisons are made:

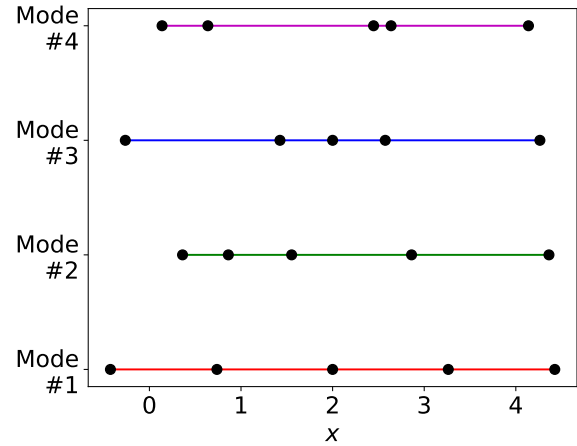
1. The free-free modes of the original system are compared to the free-free modes of a normalized component mode system with no model reduction. This comparison shows that the free-free mode shapes of both systems are identical. Figures A.3.2(a) and A.3.2(b) show the mode shape comparison for this case.
2. The fixed-base modes of the original system are compared to the fixed-base modes of a normalized component mode system with no model reduction. This comparison shows that the fixed-base mode shapes of both systems are identical. Figures A.3.3(a) and A.3.3(b) show the mode shape comparison for this case.
3. The free-free modes of the original system are compared to the free-free modes of a normalized component mode system that has undergone model reduction; in this case, the highest frequency flexible mode has been removed. This comparison shows that the free-free modes of the two systems are similar, but not identical, despite the fact that the normalized component mode system was created from a truncated set of modes. Figures A.3.4(a) and A.3.4(b) show the mode shape comparison for this case.
4. The fixed-base modes of the original system are compared to the fixed-base modes of a normalized component mode system that has undergone model reduction; as in the previous cases, the highest frequency flexible mode has been removed. This comparison shows that the fixed-base modes of the two systems are identical, despite the fact that the normalized component mode system was created from a truncated set of modes. This behavior is expected, because when the normalized component modes are constrained, the modes of the resulting system are exactly the original fixed-base modes. Figures A.3.5(a) and A.3.5(b) show the mode shape comparison for this case.

Figures A.3.2(a) through A.3.5(b) show the mode shapes of both systems for each of the four comparisons. The x-axes of these figures show the locations of the five nodes for each of the mode shapes. Note that these figures only show the flexible modes of the systems. Hence, Mode #1 refers to the first flexible mode, Mode #2 refers to the second flexible mode, and so on. Table A1 gives a quantitative summary of the four comparisons, and it presents the order of magnitude difference between the two systems for the modal frequencies and elements of the modal matrix.

In comparisons 1 and 2, the normalized component mode system is constructed using all mode shapes of the original system (i.e. no model reduction is performed). In these comparisons, the mode shapes and

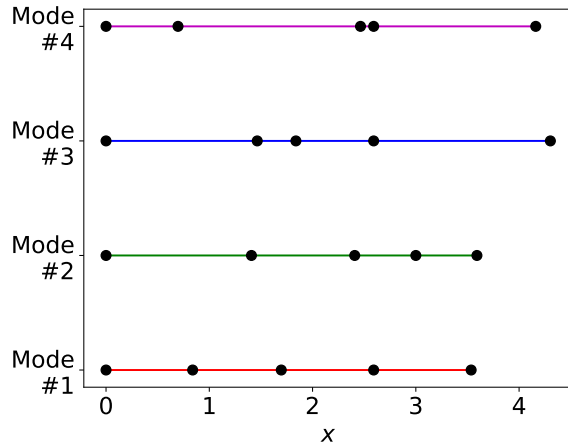


(a) Free-free mode shapes of the original system

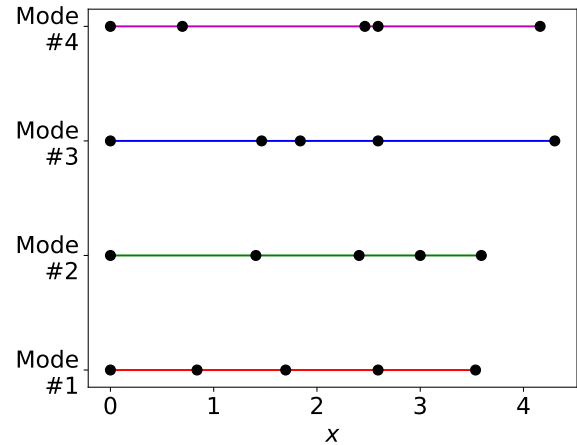


(b) Free-free mode shapes of the normalized component mode system

Figure A.3.2. Comparison # 1



(a) Fixed-base mode shapes of the original system



(b) Fixed-base mode shapes of the normalized component mode system

Figure A.3.3. Comparison # 2

modal frequencies of the two systems are identical; the difference in modal frequencies and mode shapes between the two systems is on the order of 10^{-16} , the precision of double-precision floating-point numbers. In contrast, in comparison 3, where model reduction is performed, the modal frequencies and mode shapes of the two systems are very similar but not identical. The differences between the two systems arises from the model reduction. However, in comparison 4, even though model reduction has been performed, the modal frequencies and mode shapes of the two systems are again identical. The reason the fixed-base systems match even though model reduction has been performed is when all rigid DOF in the normalized component mode system are constrained, the normalized component mode system reduces to the fixed-base mode shapes of the original system.

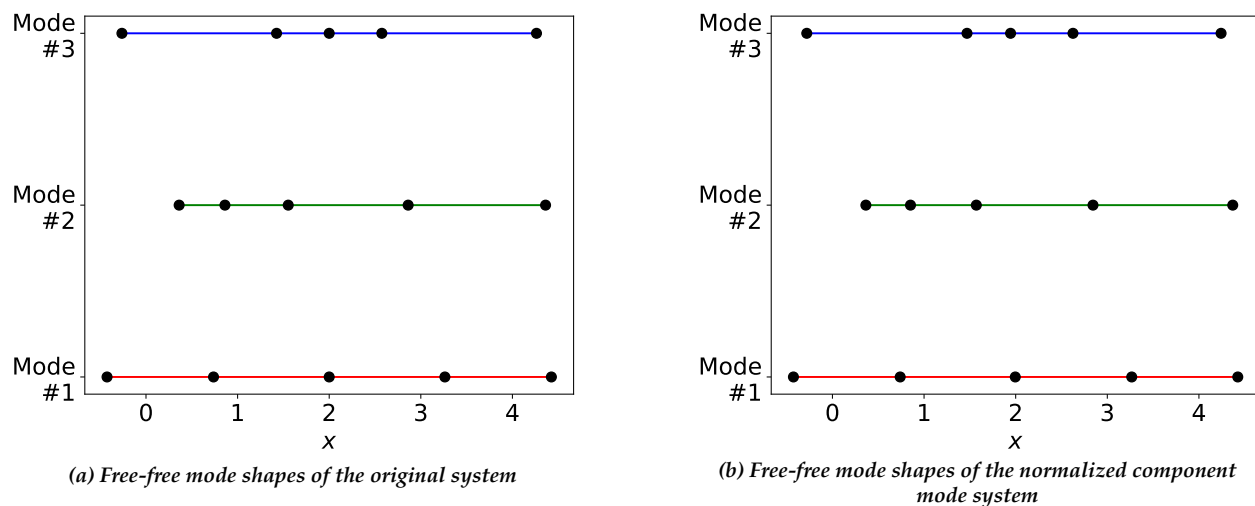


Figure A.3.4. Comparison # 3

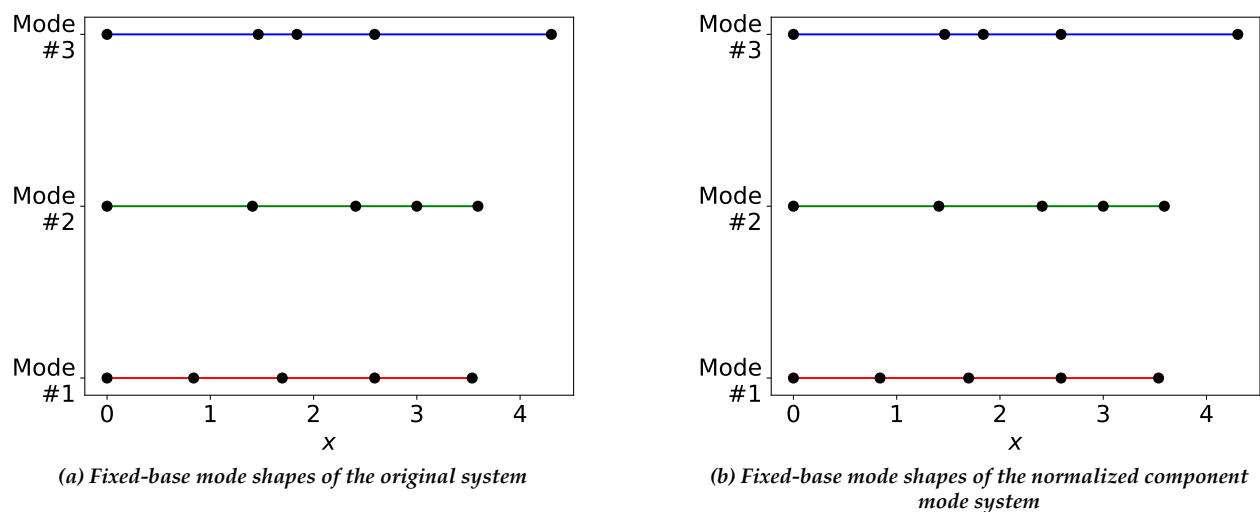


Figure A.3.5. Comparison # 4

Comparison #	Modal Frequency Difference (Order of Magnitude)	Modal Matrix Element Difference (Order of Magnitude)
1	10^{-16}	10^{-16}
2	10^{-16}	10^{-16} – 10^{-18}
3	10^{-3} – 10^{-4}	10^{-2} – 10^{-3}
4	10^{-16}	10^{-16} – 10^{-18}

Table A1. Quantitative Summary of Mode Shape Comparison

Restricted distribution to NESC and designated team members until approved by the NRB.

Appendix B. Flexible Body Dynamics Modeling Approaches for GN&C

Aerospace vehicle platforms typically involve multiple coupled bodies undergoing large articulation motions, with component bodies being rigid and flexible. The underlying dynamics of such systems are nonlinear. Physics-based models of such platforms require multibody dynamics modeling methods to capture system nonlinearities, provide adequate fidelity, and function across time-varying system configurations. This section first describes the multibody dynamics approach used for modeling such systems, and concludes by mentioning fallback lower-fidelity approaches commonly used when a general purpose multibody modeling solution is unavailable.

B.1 Nonlinear Rigid/Flexible Multibody Dynamics

Multibody dynamics theory is a well-studied and -researched topic within the overarching study of dynamical systems. The theory is commonly used to model a wide range of engineering systems, including aerospace and ground vehicles, robotic manipulators, and molecular dynamics. Two primary approaches

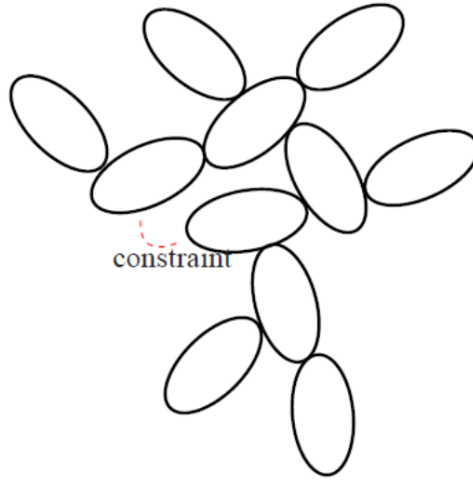


Figure B.1.1. Tree-structured Multibody System

should be considered when developing multibody dynamics algorithms for modeling and simulation purposes, which include minimal-coordinate methods [12, 13] and constraint-based methods [8, 9, 26]. The minimal coordinate methods are more complex, but lead to recursive methods that are generally less expensive and better behaved. Constraint-based methods, on the other hand, are simpler to implement, but are computationally inefficient and require more complex differential algebraic solvers to manage constraint errors. Due to the advantages, this assessment focused on the minimal coordinates approach, which reduces or avoids the need for explicit constraints, DAE integration methods, and constraint stabilization techniques.

B.1.1 Minimal Coordinate Rigid/Flexible Multibody Dynamics

In a multibody system, rigid and flexible body motions are fully coupled and lead to nonlinear dynamics. In this case, the equations of motion have the form

$$\mathcal{T} = \mathcal{M}(\vartheta)\ddot{\vartheta} + \mathcal{C}(\vartheta, \dot{\vartheta}) \quad (\text{B.1.1})$$

Restricted distribution to NESC and designated team members until approved by the NRB.

where $\mathcal{M}(\vartheta)$ denotes the mass matrix and $\mathcal{C}(\vartheta, \dot{\vartheta})$ the Coriolis term.

The ϑ generalized coordinates contain the hinge coordinate as well as deformation coordinates for all the bodies in the system. The rigid body and elastic DOF are fully coupled. Equation B.1.1 provides an exact model valid for large nonlinear rigid body motion and assumed linear elastic motion. This model requires derivation of the so-called modal integrals, which are additional mass invariant terms that arise because of the rigid-flexible body coupling (see Appendix E). In the context of an FEM model, derivation of the modal integrals requires additional information about the FEM model beyond what is normally included in a typical FEM data delivery for GN&C models.

Formulating and solving the equations of motion in Eq. B.1.1 can be complex in principle. The SOA framework for multibody dynamics [11, 13] provides a methodology for exploiting the underlying structure of the system dynamics to provide elegant and low-cost solutions. SOA is based on a minimal coordinates dynamics representation, and uses a family of spatial operators to elegantly describe multibody quantities, reveal their underlying structural patterns [27], and carry out analysis and computations with them. For example, SOA operators can be used to show that the coordinate-dependent mass matrix \mathcal{M} for any tree topology flexible multibody system can be expressed as [13]

$$\mathcal{M} = H\phi M\phi^*H^* \quad (\text{B.1.2})$$

where the block-diagonal H and M operators contain hinge articulation and body spatial inertia elements respectively, while the elements of ϕ define transformations between pairs of bodies. Beyond the elegance and compactness of such expressions, the spatial operators have important analytical properties that allow further mathematical analysis to be carried out. In particular, the ϕ operator has the form $\phi = (I - \mathcal{E}_\phi)^{-1}$, where \mathcal{E}_ϕ is another operator whose non-zero block elements correspond to the adjacency matrix for the directed graph that describes the multibody system's topology. Using SOA, analytical operator expressions can be derived for the factorization of the mass matrix and its inverse for arbitrary tree multibody systems. These expressions form the basis for a broad range of low-cost computational algorithms, including the well-known $O(N)$ recursive methods for the low-cost solution of the equations of motion for tree-topology systems [13]. Specifically, SOA shows that the mass matrix can be analytically inverted as illustrated by the following expressions [13]:

$$\begin{aligned} \mathcal{M} &= H\phi M\phi^*H^* \\ &= [I + H\phi\mathcal{K}]\mathcal{D}[I + H\phi\mathcal{K}]^* \\ [I + H\phi\mathcal{K}]^{-1} &= I - H\psi\mathcal{K} \\ \mathcal{M}^{-1} &= [I - H\psi\mathcal{K}]^*\mathcal{D}^{-1}[I - H\psi\mathcal{K}] \end{aligned} \quad (\text{B.1.3})$$

The final expression above is an explicit, analytical expression for the inverse of the mass matrix that applies to arbitrary size and branching structure. This expression avoids the expensive need for mass matrix computation and inversion and is the basis of SOA's low-cost algorithms for solving the equations of motion.

While such analytical structure of system-level dynamics properties can be studied via operators, another benefit of operator expressions is that they lead directly to low-order recursive computational algorithms. The adjacency matrix structure of the \mathcal{E}_ϕ and \mathcal{E}_ψ operators allows expressions such as ϕx , ϕ^*x , ψx and ψ^*x to be carried out via $O(N)$ recursive algorithms without requiring the explicit computation of ϕ or ψ ! The well known $O(N)$ articulated body recursive algorithm for solving the equations of motion is a direct consequence of using this property with the Eq. B.1.3 expression for the mass matrix inverse. Similarly, low-order algorithms for the operational space inertia, mass matrix sensitivity, etc. have been derived using spatial operators [13].

B.1.2 DARTS Rigid/Flexible Multibody Dynamics Software

This section describes the DARTS general-purpose software for multibody dynamics modeling, analysis, and simulation [28] used throughout this project. DARTS uses minimal coordinate models based on the

SOA methodology [13] described in the previous section and implements its low-cost recursive computational algorithms for solving the equations of motion.

DARTS is designed to handle rigid/flexible multibody dynamics, arbitrary system topologies, smooth and non-smooth dynamics, and run-time configuration changes, and also provide a full complement of computational algorithms for dynamics analysis and model-based control with fast computational performance. While the DARTS object-oriented implementation is in C++, a rich Python interface is available for all classes and methods in the system. This allows users full flexibility in defining and configuring the model as desired and even to modify the model topology and properties during run-time. DARTS is used for dynamics simulations for aerospace, ground vehicle, robotics, and multi-scale molecular applications [29].

DARTS computational algorithms are structure-based and consist of scatter/gather recursions that proceed across the bodies in the system topology. This allows DARTS to be a general-purpose tool requiring no change to the software to model multibody systems with arbitrary numbers of bodies and branching structure. This property also allows DARTS to easily handle run-time structural changes in the system topology, such as the attachment/detachment and addition/deletion of bodies. Such structural changes are common in aerospace separation and deployment scenarios, during robotics manipulation, and in model-coarsening strategies for large-scale molecular dynamics simulations. The algorithms accommodate such changes with recursions simply following the new system topology.

B.2 Alternative Fallback Solutions

Since general purpose rigid/flexible multibody modeling tools are often unavailable, simpler, lower-fidelity approaches are often used as a fallback for GN&C analyses. We describe two such methods: the linear superposition and rigid bodies with springs approaches.

B.2.1 Linear Superposition Approach

The linear superposition method is primarily designed to handle single flexible body models, and cannot model the dynamics of body articulation within the system. The approach works with frozen system configurations, requiring the use of a large number of such configurations to cover the full range of system configurations. This approach assumes that the nonlinear rigid-body motion is completely decoupled from the flexible body motion, such that the equations of motion of a single flexible body are approximated from Eq. E.1.35 by assuming

$$M_{fl}(k) = \begin{pmatrix} M_{fl}^{ff}(k) & 0 \\ 0 & M_{fl}^{rr}(k) \end{pmatrix} \quad (B.2.1)$$

and

$$\mathbf{b}_{fl}(k) = \begin{bmatrix} 0 \\ \mathbf{b}_{\omega}^k \\ \mathbf{b}_v^k \end{bmatrix}. \quad (B.2.2)$$

The modal DOF in Eq. E.1.35 are then independent of one another if proportional damping is assumed and $\Pi(\mathcal{O}_k^j)$ are the normal mode shapes. The deformed state of the j -th differential mass of the body is obtained by superimposing the decoupled rigid and flexible body motions. This type of modeling is limited to problems where body angular rates and articulated motions remain small.

At the system level, the bodies could be frozen, or locked, at a pre-determined configuration so the analysis reduces to a single-body system. To assess controller performance using these models, a normal

modes analysis is conducted for what can be a large number of system configurations. Each model is then used independently to characterize control system stability and performance over the entire configuration space. This requires a complex bookkeeping strategy to manage (1) FEM normal modes analyses for a pre-determined range of configurations and (2) conversion of each FEM normal modes analysis into a GN&C model. On top of the added bookkeeping complexity, this approach could lead to missed controller instability assessments if the chosen configuration space is too coarse.

B.2.2 Rigid Bodies with Springs Approach

Modeling flexible-body systems as a system of rigid bodies connected by linear springs is another commonly employed modeling approach. This heuristic approach begins with a multibody model consisting of rigid bodies connected via springs. A tuning process is used to approximate the mode shapes and frequencies from the FEM modal analysis. The stiffness of the springs and hinge locations of the multibody system are tuned so the resulting mode shapes and frequencies of the system of rigid bodies connected by linear springs will match the mode shapes and frequencies of the FEM model. The method produces an equation of motion that considers only the rigid-body DOF of Eq. E.1.35; i.e., the elastic DOF are completely removed. This permits large angular rates and articulated motions. However, the tuned system can only approximate system flexibility unless a large (possibly unrealistic) number of springs are used in the tuning process. This process can be time-consuming and complex, and fragile to configuration changes.

Like the linear superposition approach, the rigid body with springs approach suffers from inexact dynamics. While the linear superposition approach can be adapted only to problems that are close to stationary, the rigid body with springs approach should be used cautiously since it only approximates the flexibility of the structure predicted by the FEM model. Moreover, both approaches lack the generality of the full non-linear multibody modeling approach, and can lead to time-consuming and tedious manual model development and validation efforts.

Appendix C. FModal: A FEM-to-GN&C Flexible Multibody Modeling Pipeline Tool

C.1 Software Description

The DARTS/FModal software is a toolset written in C++ and Python that simplifies NASTRAN execution and data extraction for GN&C flexible body dynamics modeling. NASTRAN is an industry standard FEM modeling tool that is widely used in aerospace structural engineering. FModal provides functionality for GN&C analysts to work directly with NASTRAN bulk datasets for the purposes of building a GN&C flexible body dynamics model. Figure C.1.1 illustrates a typical FModal workflow. FModal imports and reads

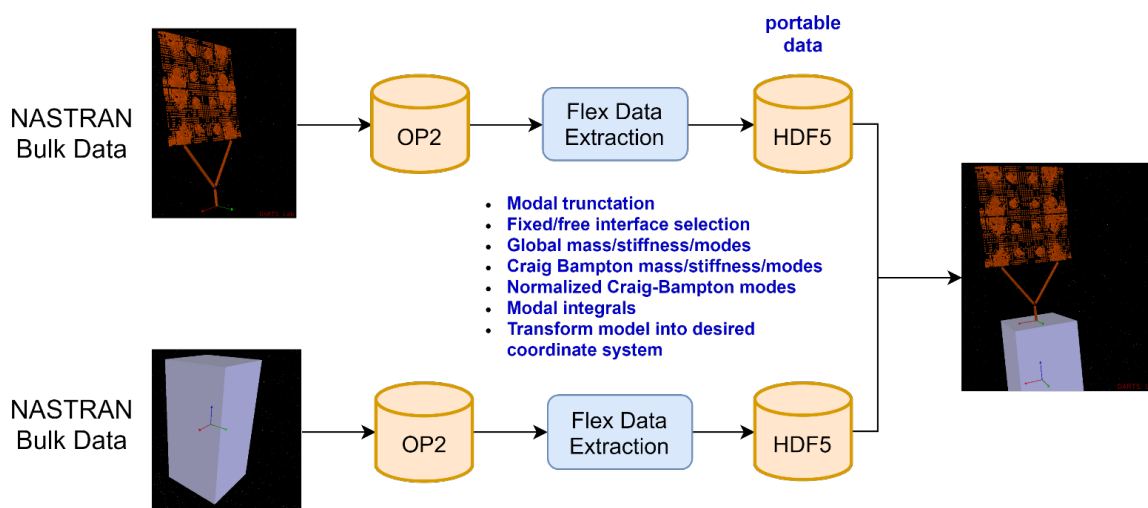


Figure C.1.1. Component Synthesis of Spacecraft Bus and Solar Array Models Using FModal

NASTRAN FEM models by parsing the executive, case control, and bulk data sections of a NASTRAN run deck. This import can consist of complex NASTRAN run decks with many "include" statements. After importing the run deck, FModal creates a custom NASTRAN run deck used to generate the flexible body data needed for GN&C flexible body modeling. This includes mode shapes and frequencies for free-free and fixed-free boundary conditions, grid point locations, pose of all coordinate systems, modal integrals, rigid body mass properties, coupled damping matrices, Craig-Bampton models, and the global physical model in sparse matrix format. FModal output data is stored in the HDF5 common open source file format [30], a portable file format well-supported by other standard software languages (e.g., Matlab, Python, Mathematica). This file format allows scaling to large data files and is self-documenting. An HDF5 interface to DARTS/Dshell has also been constructed to allow fast configuration of flexible multibody models based on Spatial Operator Algebra (SOA) techniques (see Appendix E).

The data encapsulated by the HDF5 output file can be used to configure both single body and multibody GN&C flexible body dynamics models. In the multibody context, FModal can be used to process multiple component FEMs, which can later be used to build a linear or nonlinear multibody dynamics model. In either case, FModal simplifies the iteration process for building models based on modal analysis techniques.

C.2 Normal Modes Analysis of a 10-Node Symmetric Beam

The following SimpleBeam.dat file, listed below verbatim, is a NASTRAN run deck of a 10-node symmetric beam model.

Restricted distribution to NESC and designated team members until approved by the NRB.

sol 103
cend

ECHO = NONE
DISP = ALL

method=1

BEGIN BULK

PARAM, AUTOSPC, NO
PARAM, POST, -1
PARAM, COUPMASS, -1
PARAM, MAXRATIO, 1.0+8
PARAM, grdpnt, 11
param, wtmass, 1.0

EIGRL,1,0.0,,10,,MASS

QSET1,,1001,THRU,1100
SPOINT,1001,THRU,1100

GRID	1	0	0.0	0.0	0.0	0
GRID	2	0	1.0	0.0	0.0	0
GRID	3	0	2.0	0.0	0.0	0
GRID	4	0	3.0	0.0	0.0	0
GRID	5	0	4.0	0.0	0.0	0
GRID	6	0	5.0	0.0	0.0	0
GRID	7	0	6.0	0.0	0.0	0
GRID	8	0	7.0	0.0	0.0	0
GRID	9	0	8.0	0.0	0.0	0
GRID	10	0	9.0	0.0	0.0	0
GRID	11	0	4.5	0.0	0.0	0

MAT1 1200.00+7 .29000008000.0003.6100-621.85000 0.0+
+ 1.5000+91.5000+96.8000+7

PBEAM, 50002, 1, 1.5625E-02, 2.0345E-05, 2.0345E-05, 0.0000E+00, 3.4383E-05, 0.0000E+00
, 6.2500E-02, 6.2500E-02, 6.2500E-02 -6.2500E-02 -6.2500E-02 -6.2500E-02 -6.2500E-02, 6.2500E-02
, 8.3333E-01, 8.3333E-01, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00
, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00

CBEAM, 11, 50002, 1, 2, 0.0000E+00, 1.0000E+00, 0.0000E+00
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00

CBEAM, 12, 50002, 2, 3, 0.0000E+00, 1.0000E+00, 0.0000E+00
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00

CBEAM, 13, 50002, 3, 4, 0.0000E+00, 1.0000E+00, 0.0000E+00
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00

CBEAM, 14, 50002, 4, 5, 0.0000E+00, 1.0000E+00, 0.0000E+00
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00

CBEAM, 15, 50002, 5, 6, 0.0000E+00, 1.0000E+00, 0.0000E+00
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00

CBEAM, 16, 50002, 6, 7, 0.0000E+00, 1.0000E+00, 0.0000E+00
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00

CBEAM, 17, 50002, 7, 8, 0.0000E+00, 1.0000E+00, 0.0000E+00

```
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00
CBEAM, 18, 50002, 8, 9, 0.0000E+00, 1.0000E+00, 0.0000E+00
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+0
CBEAM, 19, 50002, 9, 10, 0.0000E+00, 1.0000E+00, 0.0000E+00
, 0, 0, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+0
ENDDATA
```

The model represents a single non-articulated free-free body. Figure C.2.1 illustrates the 10-node model. An

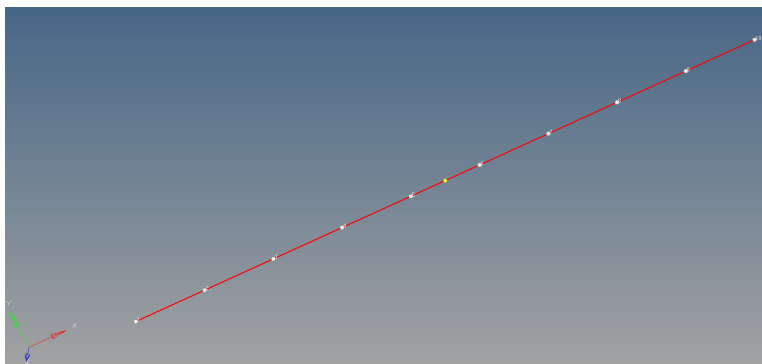


Figure C.2.1. Simple 10-Node Symmetric Beam Model

additional yellow node is shown at the undeformed center of mass for reference. The NASTRAN run deck listed above is set up to execute a normal modes solution using the NASTRAN solution sequence SOL 103. NASTRAN programs are executed using solution sequences, and SOL 103 is the standard solution sequence for computing mode shapes and frequencies. A structures analyst executes the run deck using NASTRAN and obtains the relevant GN&C data using a set of NASTRAN output files. Standard NASTRAN output files used in this process can include

- .f06 - a text file containing diagnostics and some results of the solution sequence
- .op2 - a binary file used to store matrices
- .op4 - a text file used to store matrices
- .pch - a text file containing tabulated data

The .f06 output file may not record all the necessary data needed for a GN&C model by default; e.g., Craig Bampton models and detailed information at specific grid points such as lumped mass and undeformed displacement. A structures analyst will typically have to modify the run deck and will also likely have to add Direct Matrix Abstraction Program (DMAP) code to get the relevant GN&C data, which could include, for example, the normal mode shapes Π used in Eq. A.1.5, the corresponding modal frequencies, and the direction cosine matrices and undeformed displacements corresponding to the nodes of interest. DMAP code is inserted in the executive section of the run deck, which allows the user to change the default behavior of the NASTRAN solution sequence.

Execution of the SOL 103 sequence and extraction of all relevant GN&C data from a NASTRAN run deck is a key FModal feature. The Python script listed below shows how FModal can be used to run and extract GN&C data from a normal modes analysis using the run deck SimpleBeam.dat .

```

1  import os
2  from FModal.Nastran import Nastran
3
4  # get the FModal Nastran object
5  nast = Nastran()
6
7  # read the NASTRAN model run deck
8  nast . read( "SimpleBeam.dat" )
9
10 # point FModal to the NASTRAN executable
11 nast . executable_path = os . getenv( "MSC_NASTRAN_2019_PATH" )
12
13 # run a SOL 103 solution to get 30 normal modes
14 nast . run_103(n_modes = 30 )
15
16 # load the custom FModal OP2 data into Python workspace
17 nast . load_op2(modal_integral_flag =True ,
18               rigid_body_mode_thresh =1e-4 )
19
20 # write the relevant data to an HDF5 file
21 nast . write_hdf5( "sbeam.h5" ,
22                 write_gset =True ,
23                 write_fset =True ,
24                 uset =[])

```

The above script can be used to specify the number of output mode shapes in ascending order, calculation of modal integrals, output of NASTRAN G-set and F-set matrices in sparse matrix format, and mode shapes corresponding to the nodes of interest. The prototypes of the functions listed in the above Python script are listed below:

```

def read ( self ,
           fpath):      # path to the main NASTRAN run deck file

def run_103 ( self ,
             n_modes =7,      # number of mode shapes in ascending order
             csid =0,         # output coordinate system
             aset ={},        # dictionary of fixed-interface nodes;
                               # e.g., {101 : 123456, 102 : 123, 103 : 456}
             grdpnt =None):   # reference node for rigid body mass properties

def load_op2 ( self ,
              fop2 ="",       # path to FModal OP2 file (optional)
              grdpnt =0,      # reference node for rigid body mass properties
              modal_integral_flag =False , # flag for computing modal integrals
              rigid_body_mode_thresh =1e-4 ): # threshold for computation of rigid body modes

def write_hdf5 ( self ,
               fh5,           # HDF5 output file name
               write_gset =False , # flag to output NASTRAN G-set matrices
               write_fset =False , # flag to output NASTRAN F-set matrices
               write_modal_matrix =False , # flag to output modal matrix
               uset =[]):      # list of nodes to output corresponding mode shapes and
                               # grid point locations, an empty list will output mode
                               # shapes and grid points for all nodes

```

After importing the entire NASTRAN run deck, FModal parses the executive, case control, and bulk data sections and creates a custom run deck that is printed to a single file. NASTRAN is then executed using the custom run deck. An analysis directory is created with the following output files:

/working_directory

└─ analysis_SimpleBeam					
└─ FModal_SimpleBeam.op2	...	custom	FModal	output	file
└─ SimpleBeam.dat	...	custom	FModal	run	deck
└─ SimpleBeam.f04	...	standard	NASTRAN	output	file
└─ SimpleBeam.f06	...	standard	NASTRAN	output	file
└─ SimpleBeam.log	...	standard	NASTRAN	output	file
└─ SimpleBeam.op2	...	standard	NASTRAN	output	file

The file FModal_SimpleBeam.op2 listed above contains the relevant GN&C data. The data can be used within a Python workspace or exported to the portable HDF5 file. The structure of the HDF5 file generated for the 10-node symmetric beam is shown in Figure C.2.2. The data from this file is easily im-

BGPDT	HDF5 Dataset
FMODAL_VERSION	HDF5 Dataset
Fset	HDF5 Group
GRDPNT	HDF5 Dataset
GridPosition	HDF5 Group
Gset	HDF5 Group
ModalSolution	HDF5 Group
NASTRAN_VERSION	HDF5 Dataset
N_GRIDS	HDF5 Dataset
RigidBody	HDF5 Group
TIME_STAMP	HDF5 Dataset

Figure C.2.2. FModal Output File in HDF5 Format

ported in other commonly used software environments; e.g., Fortran, C/C++, Matlab, Python, Mathematica. DARTS/Dshell can directly process the HDF5 file to extract flexible body properties for use in creating a nonlinear multibody model in DARTS for time domain and linear analysis simulation. Figure C.2.3 illustrates a view of the first four mode shapes of the symmetric beam within DARTS/Dshell.

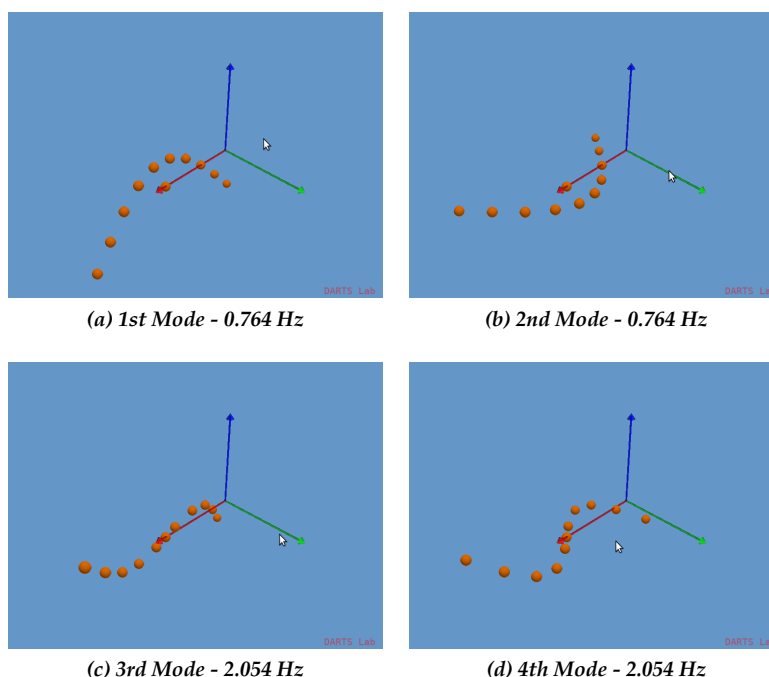


Figure C.2.3. First Four Flexible Normal Mode Shapes of 10-Node Symmetric Beam Illustrated Using DARTS/Dshell

C.3 HDF5 Output File Definitions

A complete description of the data recorded in the HDF5 output file is listed below:

```

/ sbeam.h5 ... FModal HDF5 output file
├── BGPDPT ... Basic Grid Point Definition Table
├── CoordinatePosition ... coordinate frame displacements relative to
│   ├── 1 global frame in global frame
│   └── :
├── CoordinateRotation ... coordinate frame direction cosine matrices from
│   ├── 1 local frame to global frame
│   └── :
├── CraigBampton ... Craig-Bampton model
│   ├── FREQ ... fixed-interface modal frequencies
│   ├── MassMatrix ... Craig-Bampton mass matrix
│   ├── StiffnessMatrix ... Craig-Bampton stiffness matrix
│   ├── ViscousDamping ... Craig-Bampton viscous damping matrix
│   ├── StructuralDamping ... Craig-Bampton structural damping matrix
│   └── USET ... coordinate indices for grids in Craig-Bampton model
├── FMODAL_VERSION ... FModal version used to create this file
└── Fset ... NASTRAN F-set model

```


MassMatrix ...	F-set mass matrix in sparse matrix format
StiffnessMatrix ...	F-set stiffness matrix in sparse matrix format
ViscousDamping ...	F-set viscous damping matrix in sparse matrix format
StructuralDamping ...	F-set structural damping matrix in sparse matrix format
USET ...	coordinate indices for grids in Fset model
GRDPNT ...	rigid body mass properties reference grid
GridPosition ...	displacements of each grid point relative to GRDPNT
1	
:	
Gset ...	NASTRAN G-set model
MassMatrix ...	G-set mass matrix in sparse matrix format
StiffnessMatrix ...	G-set stiffness matrix in sparse matrix format
ViscousDamping ...	G-set viscous damping matrix in sparse matrix format
StructuralDamping ...	G-set structural damping matrix in sparse matrix format
USET ...	coordinate indices for grids in G-set model
ModalSolution ...	normal mode solution data
FREQ ...	normal mode frequencies
ModalMatrix ...	normal mode shape matrix
ModalDampingMatrix ...	modal damping matrix
N FLEX MODES ...	number of flexible body mode shapes, $n_{md}(k)$
N RIGID MODES ...	number of rigid body mode shapes
RotModeShape ...	$3 \times n_{md}(k)$ rotational mode shapes for each grid point, λ_r^j (Eq. E.1.19)
1	
:	
TransModeShape ...	$3 \times n_{md}(k)$ translational mode shapes for each grid point, γ_r^j, λ_r^j (Eq. E.1.19)
1	
:	
ModalIntegral ...	modal integrals (see Appendix E)
P0 ...	$n_{md}(k) \times 3 \times 3$ array, $\mathcal{J}_1^k(r)$ (Eq. E.3.1)
P1 ...	$3 \times n_{md}(k)$ array, $m(k)p_1^k(r)$ (Eq. E.3.1)
P2 ...	$n_{md}(k) \times 3 \times 3$ array, $\mathcal{J}_1^k(r) + [\mathcal{J}_1^k(r)]^*$ (Eq. E.3.1)
P3 ...	$n_{md}(k) \times n_{md}(k) \times 3 \times 3$ array, $\mathcal{J}_1^k(r)$ (Eq. E.3.1)
P4 ...	$3 \times n_{md}(k)$ array, $F_0^k(r)$ (Eq. E.3.1)
P5 ...	$n_{md}(k) \times 3 \times n_{md}(k)$ array, $F_0^k(r, s)$ (Eq. E.3.1)
P6 ...	$n_{md}(k) \times n_{md}(k)$ array, G_0^k (Eq. E.3.1)
NASTRAN_VERSION ...	NASTRAN version used to create this file
N GRIDS ...	number of grid points in model
RigidBody ...	rigid body mass properties about GRDPNT in CSID or global frame
cmoffset ...	3×1 center of mass offset, p_0^k , (Eq. E.2.4)
inertia ...	3×3 inertia matrix, \mathcal{J}_0^k , (Eq. E.2.4)
mass ...	total mass, $m(k)$, (Eq. E.2.3)

TIME_STAMP ... time stamp of file creation

C.4 Craig-Bampton Models

FModal can be used to extract both forms of Craig-Bampton modeling in Eq. A.2.8 and Eq. A.2.13. FModal will record both Craig-Bampton models in the HDF5 output file when run_103 is executed with nodes specified in the aset option as shown in the function prototype listing above. The nodes listed in the aset option are the fixed-interface nodes.

The utility of this type of modeling is demonstrated by considering the spacecraft bus and solar array models shown in Figure C.1.1. The bus model is assumed rigid while the solar array is a component NASTRAN FEM model. At the system level, the solar array is attached to the rigid bus model with fixed-interface boundary conditions.

A DARTS/Dshell simulation (see Appendix D) was constructed where the solar array model is imported using the FModal HDF5 output file and constrained to a rigid bus model. Likewise, an equivalent NASTRAN system model was constructed where the solar array component FEM model is attached to the rigid body bus model. Figure C.4.1 shows system frequency comparisons of the DARTS and NASTRAN lin-

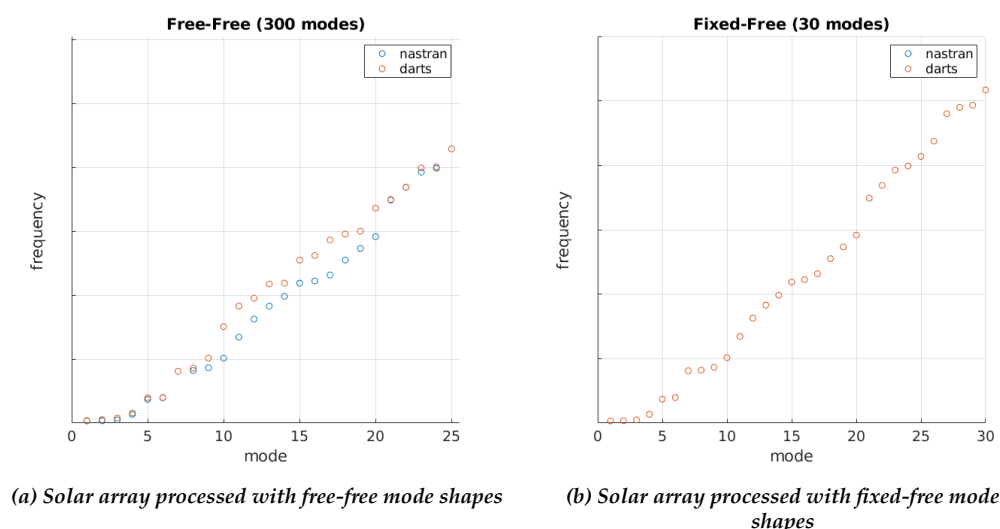


Figure C.4.1. System Frequency Comparison Between DARTS and NASTRAN Using Free-Free and Fixed-Free Mode Shapes

ear models using free-free and fixed-free (Craig-Bampton) mode shapes for the solar array model. The DARTS model should re-create the NASTRAN system frequencies. The comparison clearly illustrates that the Craig-Bampton model is better suited to model the solar array attachment to the bus. Figure C.4.2 shows the difference in system frequency for the fixed-free mode shape comparison. The difference is on the order of 10^{-5} Hz, which is commensurate with the truncation error used to linearize the inherently nonlinear DARTS model. In this example, the GN&C analyst is free to articulate the solar array in DARTS and re-run the GN&C linear analysis without the need to run the full NASTRAN system model for every solar array angle under consideration.

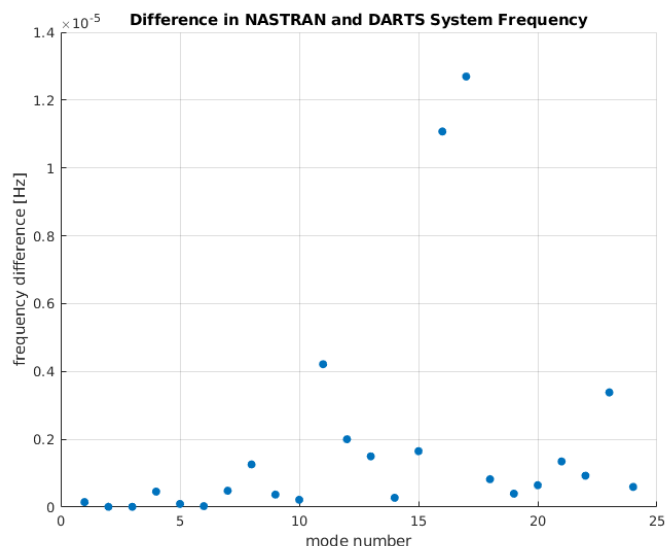


Figure C.4.2. Comparison of NASTRAN and DARTS System Frequencies for Flexible Solar Array Attached to Rigid Bus

C.5 Iterating FEM Models Using FModal

Traditionally, iteration of NASTRAN FEM models for GN&C purposes requires a structures analyst to modify the run deck to analyze new assumptions. A new dataset is then re-generated and re-delivered by the structures analyst. However, with FModal, a GN&C analyst can independently accomplish this iteration for common cases, e.g., when the FEM model is updated, a new cutoff frequency is desired, modal integrals need to be evaluated, a different node set is required for mode shape output, or even if a different version of NASTRAN needs to be used. However, some coordination should be established with the structures analyst responsible for the FEM model. At a minimum, the FEM model should be executable using NASTRAN without FATAL terminations. Additionally, NASTRAN results can change version to version and the FEM model may only compile with a specific version of NASTRAN. Currently, FModal supports MSC NASTRAN 2019, 2017, 2016, and NX NASTRAN 12, and 7.1.

Appendix D. System-Level Modeling and Simulation

Full-scale integrated modeling and simulation of a vehicle system requires a software environment that goes beyond multibody dynamics modeling. While the multibody dynamics modeling handles the underlying interbody interactions, other structural-environmental interactions due to, e.g., aeroelasticity, gravity, or electromagnetism may need to be assessed. Models of vehicle actuator and sensor components, typically distinct from the multibody formulation, must also be included to capture the large-scale behavior of the vehicle, which could include, e.g., servo-elastic effects. In addition, the simulation framework must support nonlinear time-domain and linearization to support controller design and system performance assessments.

This project leveraged and extended the DARTS/Dshell simulation framework to demonstrate full-scale modeling and simulation in the context of a FEM-to-GN&C modeling pipeline. Figure D.0.1 illustrates the

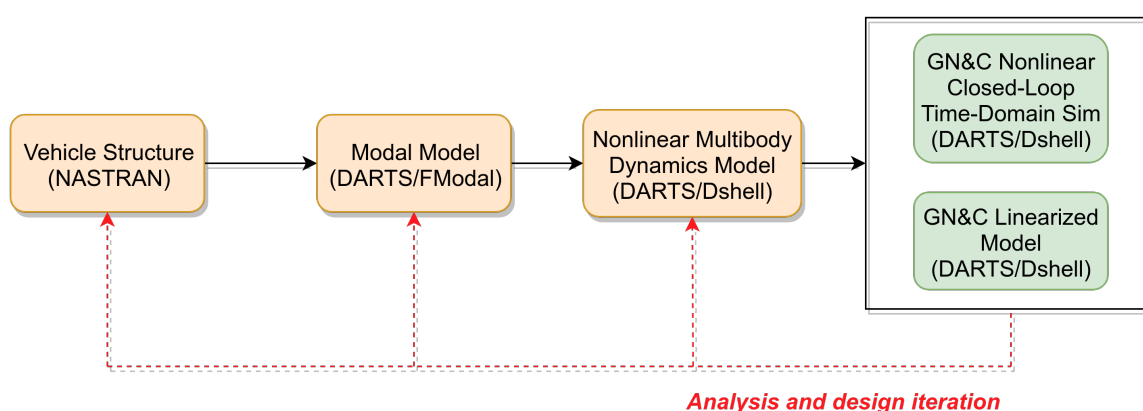


Figure D.0.1. Flowchart Illustrating How DARTS/Dshell Enables FEM-to-GN&C Pipeline for System-Level Modeling and Simulation

key components that form an enabling framework for executing analysis and design iterations between the structural and GN&C disciplines. In this case, the vehicle structural model, which is distinct from the multibody formulation, is adapted to the DARTS/Dshell simulation environment using DARTS/FModal (see Appendix C). The framework improves the process for developing and validating system performance requirements by tying together modeling and simulation techniques from the structural engineering, multibody dynamics, and GN&C technical disciplines.

D.1 Dshell System-Level Modeling Framework

While DARTS handles the articulated body dynamics of a vehicle, Dshell provides a component model-based simulation framework to add and couple parameterized models for actuator/sensor devices and environment interactions needed for modeling the overall physics of a vehicle (see Figure D.1.1). The reusability of component models is a key to Dshell's usage across multiple platform applications, and even platform domains (e.g., aerospace, UGVs, UAV, robotics). Dshell's component models library includes sensors (e.g., IMUs, cameras, LIDARs), actuators (e.g., motors, engines), and environment interactions (e.g., gravity, aerodynamics, terramechanics) that can be used out of the box, adapted, or added to from third party sources as needed. For instance, the aerodynamics models rely on planetary atmosphere (GRAM models) and wind models to compute the aerodynamics forces on a flight vehicle. The typical use of Dshell is in closed-loop simulation with autonomy, navigation, guidance and control software. Dshell can simulate

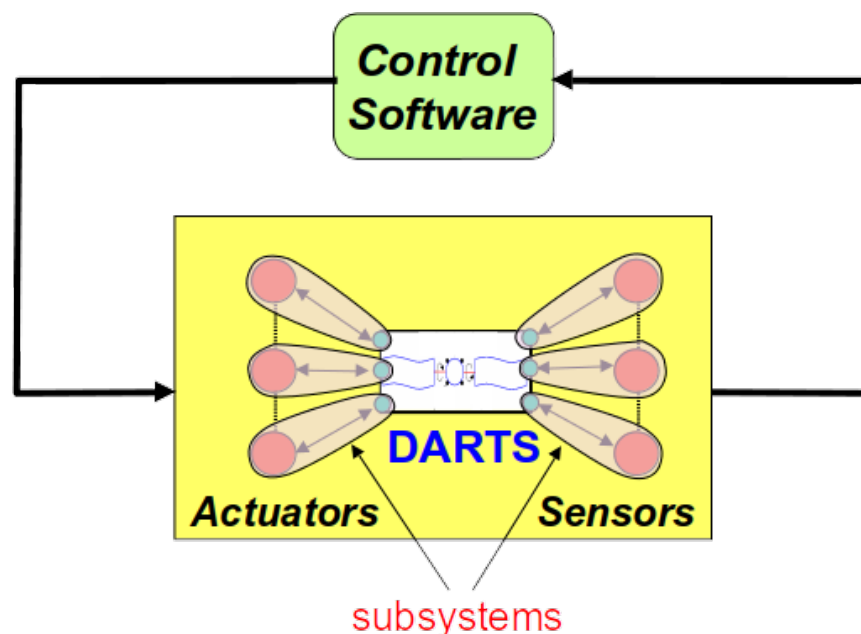


Figure D.1.1. Dshell Simulation Framework

single and multiple heterogeneous vehicles.

The high-fidelity IMU model accounts for quantization, angle random walk, rate random walk, bias stability, drift rate (rate ramp), and angle white noise effects as well as scale factor error, sensor misalignment, and the resolution of the IMU integer counter. Third-party models for sensors and other devices can be easily included by developing Dshell model wrapper interfaces for them. This is fairly common, since projects often develop highly detailed models for project-specific hardware which are then plugged into the simulation via Dshell model wrappers.

Environmental interaction models include multiple flavors of terrain mechanics models for ground (e.g., Bekker/Wong, Terzhagi) interaction. Aerodynamics models are available for aerial rotorcraft, balloon, and lander simulations, which use vehicle-derived aerodatabase parameters to compute forces on the vehicle based on the atmospheric conditions and wind models.

Vehicle simulations are often complex and can involve large numbers of models, coupled interactions between system elements, and complex environmental interactions. Managing this complexity is critical to the use, sustainability, robustness, and extensibility of simulations. The Dshell framework uses the notion of hierarchical assemblies to create and organize simulation sub-systems (e.g., wheels, limbs, cameras and even whole vehicles). Assemblies can be reused to support repeated sub-system patterns (e.g., multiple limbs, cameras) often found within platform models. Assemblies can be reused from one platform simulation to another. Dshell assemblies allow the scaling up of simulations to provide the highly complex engineering analysis models needed to meet the high bar of real-world application, while still benefiting from reuse.

Dshell includes a large family of fixed and variable step integrators for simulation time-stepping as well as the use of multi-rate models. There is also support for exact zero-finding during time-stepping to ensure that the simulation state does not step over key events (e.g., ground contact). The standard Dshell framework also includes 3D visualization, GUIs, data logging, and large-scale Monte Carlo simulation capabilities. Moreover, Dshell tracks physical units of all simulation parameters and includes input data

checking for catching user errors early.

D.2 SLS Modeling and Simulation Using Dshell

Dshell was used to develop a launch vehicle simulation based on the SLS Integrated Modal Test (IMT) NASTRAN FEM model [20] (see Figure D.2.1). The purpose of the simulation was to demonstrate the FEM-

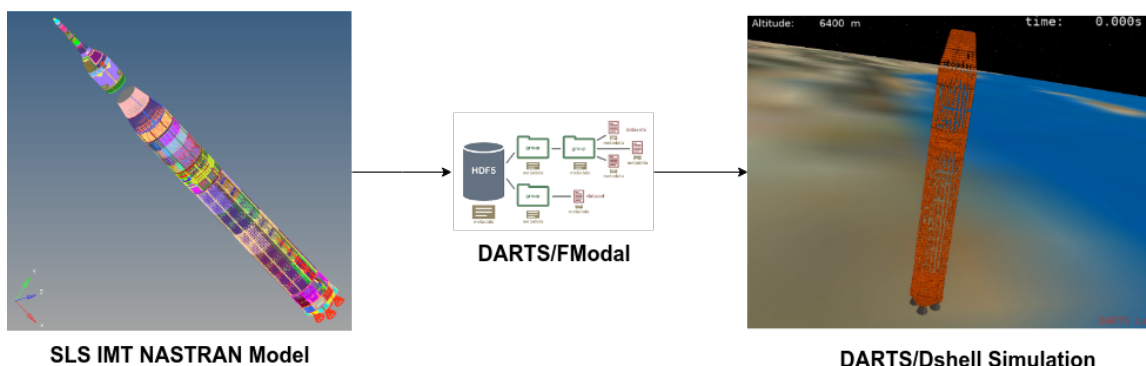


Figure D.2.1. FEM-to-GN&C Pipeline Demonstration Using SLS IMT FEM

to-GN&C pipeline capability using a full-scale vehicle model. For simplicity, the SLS IMT FEM included the Launch Vehicle Stage Adapter (LVSA), the core stage, and four core stage engine models. The upper stage vehicle components and all reduced order models (e.g., ICPS, boosters, and MPCV) were ignored. The engines were modeled as lumped rigid bodies in NASTRAN using the CONM2 card. Ideal actuators at the engine gimbal joints were considered.

The IMT FEM model was delivered with four engines connected to the thrust structure, as shown in Figure D.2.2. In Figure D.2.2, the node numbers correspond to Engine 1 structural components, which include

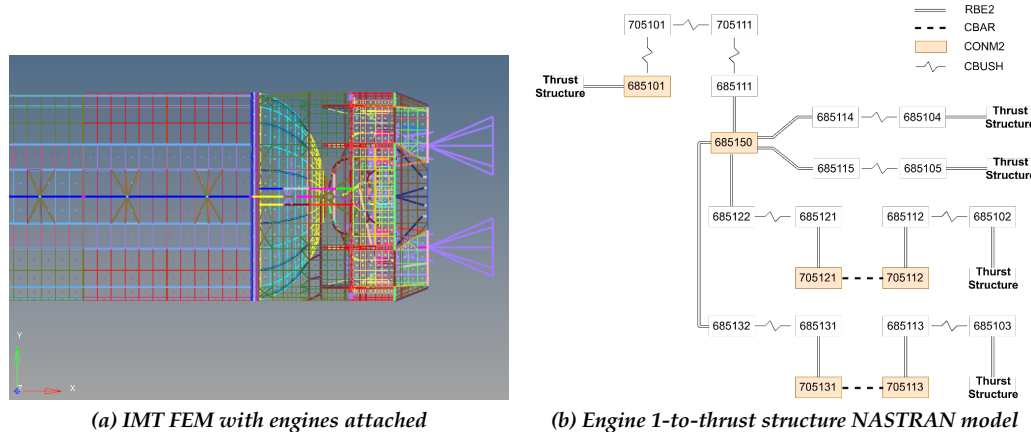


Figure D.2.2. IMT FEM Engine Model

the engine inertia (685150), gimbal bearing (685111), LO2 feedline (685114), LH2 feedline (685115), pitch actuator (685122), and yaw actuator (685132). The other three engines were assembled in the same manner. Modal analysis of the FEM in this form would yield a set of mode shapes with EITM. Construction of a GN&C flexible body dynamics model using EITM mode shapes will lead to double-booking of engine inertia since the GN&C multibody model will need to attach an articulated body at the gimbal node to account

for the motion of the engine body with respect to the core stage body. While the rigid body properties of the engines could be removed from the overall system, the system mode shapes will still double-book the engine inertias. Another option would involve delivery of a FEM model with EOTM, as shown in Figure D.2.3.

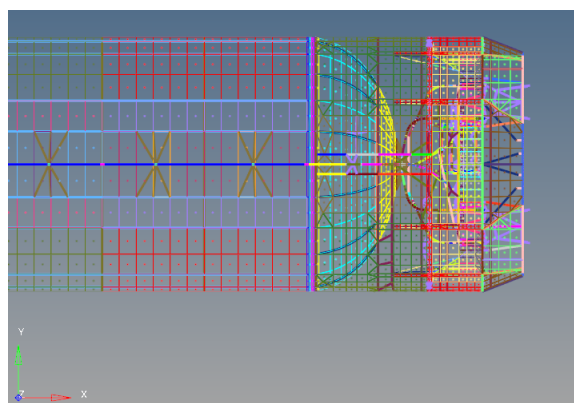


Figure D.2.3. IMT FEM Without Engines

Two FEM models were constructed to compare the use of EITM and EOTM mode sets with the GN&C multibody model. Using FModal (see Appendix C), flexible body models for EITM and EOTM mode sets were constructed in Dshell. The EOTM model was constructed by removing the engine components for each engine in the IMT FEM (e.g., engine inertia, pitch actuator, yaw actuator). Four articulated engine bodies were added to EITM and EOTM models in Dshell. U-joints with prescribed motion were configured at the engine joints to enable pitch and yaw motion for each of the engines in both models. The rigid body properties of the engines were removed from the EITM model.

Once configured, the Dshell simulation provides the facility to conduct linear analysis and nonlinear time-domain simulation. Given user specified inputs and outputs, a linear state space model is constructed using finite differences and stored in the HDF5 file format shown in Figure D.2.4. Indices corresponding

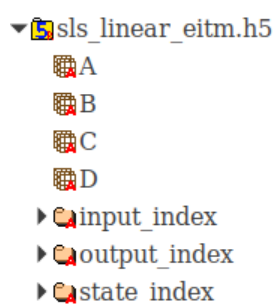
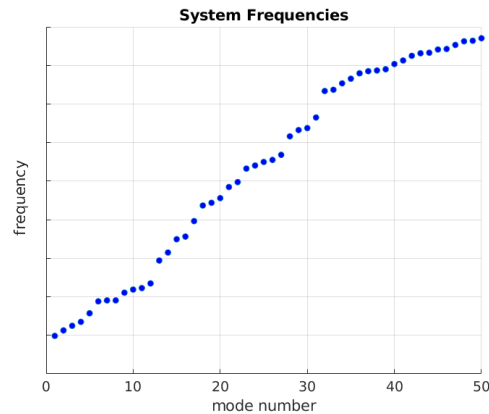


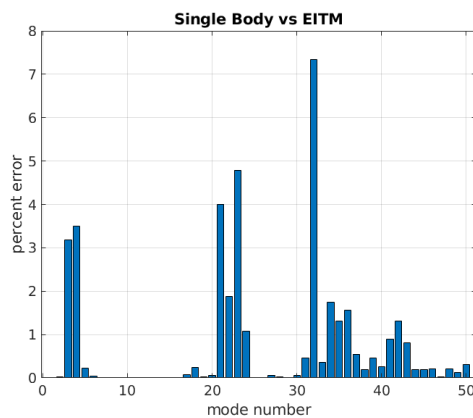
Figure D.2.4. Linear State Space Model Output Stored in HDF5 File Format

to inputs, outputs, and states are provided for reference. Using linearization, system frequencies for EITM and EOTM GN&C models were compared with the IMT FEM single body case. Figure D.2.5 shows the error in system frequency between the EITM, EOTM, and IMT FEM single-body models using an 11 Hz cutoff frequency. The EOTM GN&C model more closely matches the IMT FEM system frequencies since the core stage engine bodies are not double-booked, as they are in the EITM GN&C model.

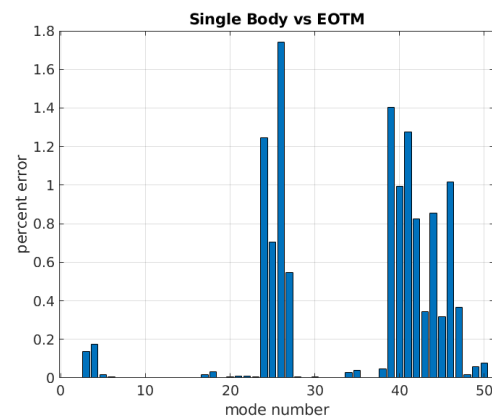
Efficient iteration of studies like the EITM/EOTM comparison is enabled by the FModal and Dshell toolsets. In terms of linearization, note that the linear state space models are constructed from full model fidelity,



(a) Single-body IMT FEM system frequencies



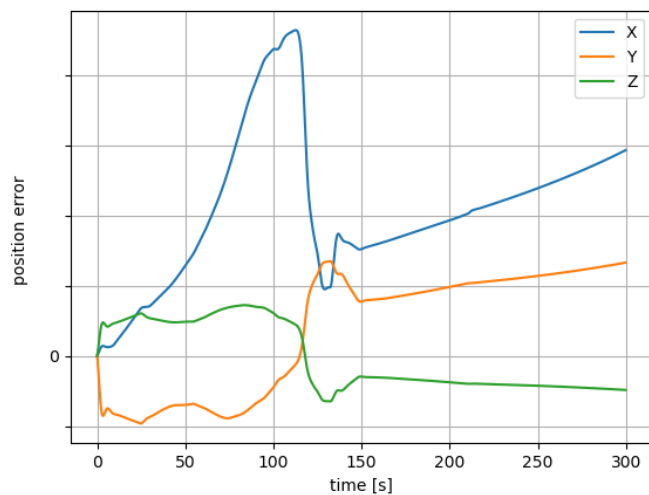
(b) System frequency error between IMT FEM and EITM models.



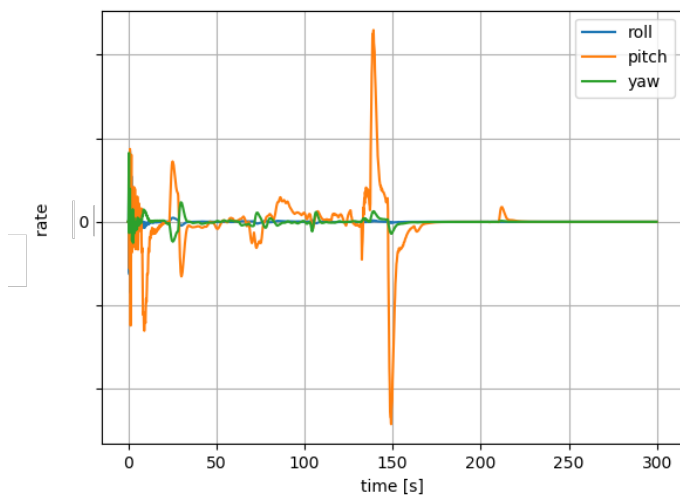
(c) System frequency error between IMT FEM and EOTM models

Figure D.2.5. System Frequency Comparison Among IMT FEM, EITM, and EOTM Models

which includes nonlinear flexible body dynamics and any additional environmental, actuator, and sensor models that have been added to the Dshell simulation. In terms of nonlinear time-domain simulation, the Dshell environment provides the facility to carry out full end-to-end simulation. As a demonstration, a VAC-1R POST trajectory was used to command the vehicle from launch to T-plus 300 sec. The linearized EOTM model was used to design a Linear Quadratic Regulator (LQR) controller to achieve closed-loop control. Allocation of the engine torque demand for each engine throughout the trajectory was achieved using a least-squares approach. Figure D.2.6 shows the controller performance in terms of inertial position error and a time history plot of the angular rates at the IMU node with respect to the core stage body frame.



(a) Inertial position error between actual and commanded trajectory



(b) Angular rate measurement at IMU node with respect to core stage body frame

Figure D.2.6. Dshell Time-Domain Plots

Appendix E. Modal Integrals for Flexible Bodies

Appendix A focused on an approach using assumed modes to model component body deformation—this approach is especially suited to model reduction and for articulated multibody systems with attachment constraints. The system FEM is often created for the primary purpose of structural (loads, stress) analyses. The structural effects of articulation between component bodies are usually assessed by creating multiple FEMs with different articulation states and repeating the required analyses. Modal data for the individual component bodies are extracted from the FEM models and provided to other disciplines as required.

This approach simplifies the FEM development by eliminating the complexities associated with a dynamics model that incorporates articulation between bodies. However, this approach does not capture the variation in inertia properties due to deformation, which may be important for some systems. Including these missing effects requires additional modal integral data to be extracted from the FEM models. In this appendix we describe these additional modal integrals. The FModal tool described in Appendix C is able to extract these modal integrals into an HDF5 file, and the DARTS multibody engine (Section B.1.2) has an interface to import this modal integral data for use in dynamics computations.

Appendix B discusses the application of this approach to the key role of nonlinear dynamics modeling of flexible multibody system for GN&C design and V&V applications.

E.1 Equations of motion for the k^{th} body

In a lumped mass model of a flexible body, the body can be viewed as a collection of independent rigid nodes interconnected by springs. We begin with developing the equations of motion for a single node, with the ultimate goal of developing the modal equations of motion for a flexible body and the required modal integrals.

Spatial notation works with rotational and translation quantities together to simplify the dynamics expressions, especially when working with quantities offset from the center of mass. Thus spatial velocities are 6-vectors containing angular and linear velocity components, while spatial forces are 6-vectors containing moment and force terms. A 6×6 spatial inertia combines a body's mass, center of mass offset, and inertia matrix within a single quantity. Additional discussion on the benefits of spatial notation can be found in reference [13].

E.1.1 Equations of motion for the single \mathcal{O}_k^j node (derivatives in the \mathcal{O}_k^j node frame)

We focus here on the k^{th} flexible body. The reference frame for the body is denoted \mathbb{B}_k . Let us assume that the k^{th} body has $n_{\text{nd}}(k)$ nodes and that the j^{th} node of the k^{th} body is denoted as \mathcal{O}_k^j . Note that the node is not considered a point mass; it may have a center of gravity offset and moment of inertia from/about \mathcal{O}_k^j . The translational vector from \mathbb{B}_k to the \mathcal{O}_k^j node reference frame is denoted $\mathbf{l}(k, \mathcal{O}_k^j) \in \mathcal{R}^3$, as illustrated in Figure E.1.1. $\mathbf{l}(k, \mathcal{O}_k^j)$, is deformation dependent and is given by:

$$\mathbf{l}(k, \mathcal{O}_k^j) = \mathbf{l}_0(k, \mathcal{O}_k^j) + \delta_{\mathbf{l}}(\mathcal{O}_k^j) \in \mathcal{R}^3 \quad (\text{E.1.1})$$

Here $\mathbf{l}_0(k, \mathcal{O}_k^j)$ denotes the undeformed vector and $\delta_{\mathbf{l}}(\mathcal{O}_k^j)$ the translational deformation for the \mathcal{O}_k^j node. Let $\delta_{\mathbf{r}}(\mathcal{O}_k^j)$ denote the relative attitude deformation matrix of the node. Also, let $\mathbf{M}_{\text{nd}}(\mathcal{O}_k^j) \in \mathcal{R}^{6 \times 6}$ denote

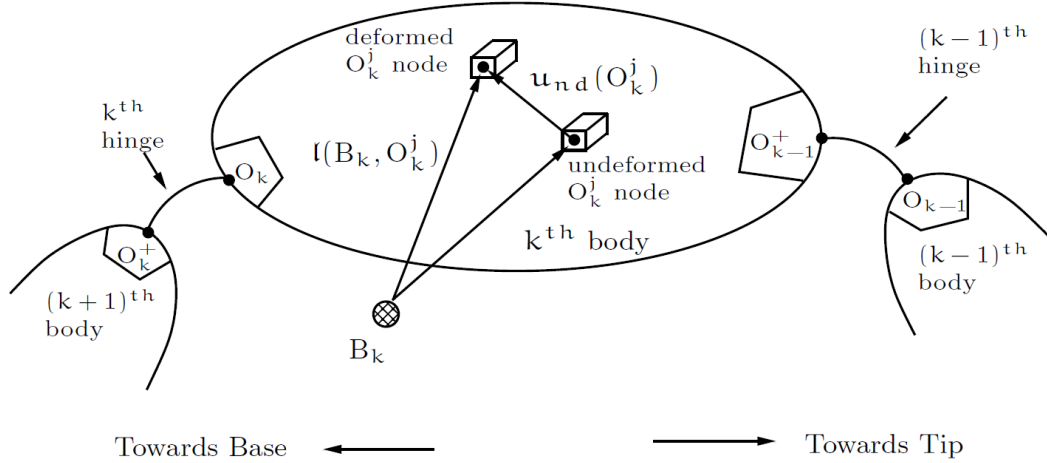


Figure E.1.1. Illustration of Links and Hinges in a Flexible Serial Multibody System

the spatial inertia of the \mathcal{O}_k^j node (in the local node frame)

$$M_{nd}(\mathcal{O}_k^j) = \begin{pmatrix} \mathcal{J}(\mathcal{O}_k^j) & m(\mathcal{O}_k^j)\tilde{p}(\mathcal{O}_k^j) \\ -m(\mathcal{O}_k^j)\tilde{p}(\mathcal{O}_k^j) & m(\mathcal{O}_k^j)\mathbf{I}_3 \end{pmatrix} \in \mathcal{R}^{6 \times 6} \quad (\text{E.1.2})$$

where $\mathcal{J}(\mathcal{O}_k^j) \in \mathcal{R}^{3 \times 3}$, $p(\mathcal{O}_k^j) \in \mathcal{R}^3$ and $m(\mathcal{O}_k^j)$ denote the rotational inertia, the center of mass offset and the mass of node \mathcal{O}_k^j respectively in the node frame. Note that we use the notation $\tilde{p}(\mathcal{O}_k^j)$ to denote the skew-symmetric cross-product matrix associated with $p(\mathcal{O}_k^j)$. The $M_{nd}(\mathcal{O}_k^j)$ node spatial inertia is constant in the \mathcal{O}_k^j frame. $\underline{M}_{nd}(\mathcal{O}_k^j) \in \mathcal{R}^{6 \times 6}$ denotes the spatial inertia of node \mathcal{O}_k^j in the k^{th} (parent) body frame and is given by

$$\underline{M}_{nd}(\mathcal{O}_k^j) = \Delta_r(\mathcal{O}_k^j)M_{nd}(\mathcal{O}_k^j)\Delta_r^*(\mathcal{O}_k^j) \in \mathcal{R}^{6 \times 6} = \begin{pmatrix} \underline{\mathcal{J}}(\mathcal{O}_k^j) & m(\mathcal{O}_k^j)\underline{\tilde{p}}(\mathcal{O}_k^j) \\ -m(\mathcal{O}_k^j)\underline{\tilde{p}}(\mathcal{O}_k^j) & m(\mathcal{O}_k^j)\mathbf{I}_3 \end{pmatrix} \quad (\text{E.1.3})$$

$$\text{with } \underline{\mathcal{J}}(\mathcal{O}_k^j) \triangleq \delta_r(\mathcal{O}_k^j)\mathcal{J}(\mathcal{O}_k^j)\delta_r^*(\mathcal{O}_k^j) \quad \text{and} \quad \underline{\tilde{p}}(\mathcal{O}_k^j) \triangleq \delta_r(\mathcal{O}_k^j)p(\mathcal{O}_k^j)$$

and

$$\Delta_r(\mathcal{O}_k^j) \triangleq \begin{pmatrix} \delta_r(\mathcal{O}_k^j) & 0 \\ 0 & \delta_r(\mathcal{O}_k^j) \end{pmatrix} \in \mathcal{R}^{6 \times 6} \quad (\text{E.1.4})$$

We will now introduce some of the notation used in the SOA approach defined in [13]. This approach allows for the methodical, compact expression of flexible multibody dynamics. While not the focus of this appendix, the more important advantage of this approach is that it allows for extremely efficient calculation of the mass matrix and its inverse for large systems. See [13] for these details.

The spatial velocity of node \mathcal{O}_k^j is given by $\mathcal{V}(\mathcal{O}_k^j)$, and the spatial velocity of frame B_k is given by $\mathcal{V}(k)$. Both quantities are inertially referenced and resolved into B_k . We can use the rigid body spatial transformation

operator to transform $\mathcal{V}(k)$ to the undeformed location of node \mathcal{O}_k^j , and we then add the deformation spatial velocity to obtain $\mathcal{V}(\mathcal{O}_k^j)$:

$$\mathcal{V}(\mathcal{O}_k^j) = \Phi^*(k, \mathcal{O}_k^j) \mathcal{V}(k) + \delta_{nd}^{\mathcal{V}}(\mathcal{O}_k^j) \in \mathcal{R}^6, \quad (\text{E.1.5})$$

where

$$\Phi(k, \mathcal{O}_k^j) \triangleq \begin{pmatrix} \mathbf{I} & \tilde{l}(k, \mathcal{O}_k^j) \\ 0 & \mathbf{I} \end{pmatrix} \in \mathcal{R}^{6 \times 6} \quad (\text{E.1.6})$$

and

$$\mathcal{V}(k) = \begin{bmatrix} \omega(k) \\ v(k) \end{bmatrix} \in \mathcal{R}^6. \quad (\text{E.1.7})$$

The variables $\omega(k)$ and $v(k)$ are the angular and linear velocities of body k with respect to the inertial frame. With $\delta_{\omega}(\mathcal{O}_k^j)$ denoting the deformation angular velocity of \mathcal{O}_k^j and $\delta_v(\mathcal{O}_k^j)$ denoting the deformation linear velocity, i.e., the time derivative of $\delta_l(\mathcal{O}_k^j)$, the deformation spatial velocity $\delta_{nd}^{\mathcal{V}}(\mathcal{O}_k^j)$ of the \mathcal{O}_k^j with respect to the \mathbb{B}_k body frame takes the form

$$\delta_{nd}^{\mathcal{V}}(\mathcal{O}_k^j) = \begin{bmatrix} \delta_{\omega}(\mathcal{O}_k^j) \\ \delta_v(\mathcal{O}_k^j) \end{bmatrix} \in \mathcal{R}^6 \quad (\text{E.1.8})$$

The equations of motion of a single node expressed in the k^{th} body frame \mathbb{B}_k using its (node frame derivative) spatial acceleration $\alpha_{nd}(\mathcal{O}_k^j)$ are given by [13]:

$$\mathbf{f}_{nd}(\mathcal{O}_k^j) = \underline{\mathbf{M}}_{nd}(\mathcal{O}_k^j) \alpha_{nd}(\mathcal{O}_k^j) + \mathbf{b}(\mathcal{O}_k^j) + \mathbf{f}_{nd}^{st}(\mathcal{O}_k^j) \quad \text{where} \quad \mathbf{b}(\mathcal{O}_k^j) = \overline{\mathbf{V}}(\mathcal{O}_k^j) \underline{\mathbf{M}}_{nd}(\mathcal{O}_k^j) \mathcal{V}(\mathcal{O}_k^j) \quad (\text{E.1.9})$$

$\mathbf{f}_{nd}^{st}(\mathcal{O}_k^j)$ represents the spatial forces arising from structural stiffness and damping forces imposed by adjacent nodes, and $\mathbf{f}_{nd}(\mathcal{O}_k^j)$ the external spatial force on the node. $\mathcal{V}(\mathcal{O}_k^j)$ and $\mathbf{b}(\mathcal{O}_k^j)$ denote the spatial velocity of \mathcal{O}_k^j with respect to the inertial frame and the gyroscopic spatial force for the \mathcal{O}_k^j node. All these quantities are resolved in the \mathbb{B}_k frame.

Note that in this development, the spatial acceleration $\alpha_{nd}(\mathcal{O}_k^j)$ is calculated by transforming the body-frame quantity $\mathcal{V}(\mathcal{O}_k^j)$ into the node frame, taking the derivative in the node frame, then transforming the result back into the body-frame. Thus, $\alpha_{nd}(\mathcal{O}_k^j) = \Delta_r(\mathcal{O}_k^j) \frac{d}{dt} [\Delta_r^*(\mathcal{O}_k^j) \mathcal{V}(\mathcal{O}_k^j)]$.

In the development above we have used the notation $\widetilde{(\quad)}$ and $\overline{(\quad)}$ for a spatial vector z as follows:

$$\text{For } z \triangleq \begin{bmatrix} x \\ y \end{bmatrix}, \quad \widetilde{z} \triangleq \begin{pmatrix} \widetilde{x} & \mathbf{0}_3 \\ \widetilde{y} & \widetilde{x} \end{pmatrix} \in \mathcal{R}^{6 \times 6} \quad \text{and} \quad \overline{z} \triangleq \begin{pmatrix} \widetilde{x} & \widetilde{y} \\ \mathbf{0}_3 & \widetilde{x} \end{pmatrix} \quad (\text{E.1.10})$$

E.1.2 Equations of motion for the single \mathcal{O}_k^j node (derivatives in the k^{th} body frame)

The equations of motion of a node in Eq. E.1.9 are using the node frame derivative spatial acceleration. To combine the equations of motion of all the nodes on the body, we need to express them via the body spatial acceleration—and we do so below in the \mathbb{B}_k body frame.

Recall that we defined the spatial acceleration $\alpha_{nd}(\mathcal{O}_k^j)$ as

$$\alpha_{nd}(\mathcal{O}_k^j) = \Delta_r(\mathcal{O}_k^j) \frac{d}{dt} \left[\Delta_r^*(\mathcal{O}_k^j) \mathcal{V}(\mathcal{O}_k^j) \right]. \quad (\text{E.1.11})$$

The coordinate transformation $\Delta_r(\mathcal{O}_k^j)$ is defined in E.1.4 and $\delta_r(\mathcal{O}_k^j)$ denotes the relative attitude deformation matrix of \mathcal{O}_k^j . With $\delta_\omega(\mathcal{O}_k^j)$ denoting the deformation angular velocity of \mathcal{O}_k^j , we have

$$\dot{\delta}_r(\mathcal{O}_k^j) = \tilde{\delta}_\omega(\mathcal{O}_k^j) \delta_r(\mathcal{O}_k^j). \quad (\text{E.1.12})$$

With this definition, we can take the derivative in Equation E.1.11 to find an expression for $\alpha_{nd}(\mathcal{O}_k^j)$ using body frame derivatives:

$$\alpha_{nd}(\mathcal{O}_k^j) = \overset{\circ}{\mathcal{V}}(\mathcal{O}_k^j) - \begin{bmatrix} \tilde{\delta}_\omega(\mathcal{O}_k^j) & 0 \\ 0 & \tilde{\delta}_\omega(\mathcal{O}_k^j) \end{bmatrix} \mathcal{V}(\mathcal{O}_k^j) \quad (\text{E.1.13})$$

We can now take the derivative of Eq. E.1.5 in the body frame to find an expression for $\overset{\circ}{\mathcal{V}}(\mathcal{O}_k^j)$. Substituting into Equation E.1.13 yields the following expression relating the $\alpha(k)$ body acceleration (body frame derivative) and the $\alpha_{nd}(\mathcal{O}_k^j)$ node acceleration (node frame derivative):

$$\begin{aligned} \alpha_{nd}(\mathcal{O}_k^j) &= \Phi^*(k, \mathcal{O}_k^j) \alpha(k) + \overset{\circ}{\delta}_{nd}^{\mathcal{V}}(\mathcal{O}_k^j) + \mathbf{a}_f(k, \mathcal{O}_k^j) \\ \text{where } \mathbf{a}_f(k, \mathcal{O}_k^j) &\triangleq \tilde{\mathcal{V}}(\mathcal{O}_k^j) \delta_{nd}^{\mathcal{V}}(\mathcal{O}_k^j) - \begin{bmatrix} 0 \\ \tilde{\delta}_\omega(\mathcal{O}_k^j) \delta_v(\mathcal{O}_k^j) \end{bmatrix} = \begin{bmatrix} \tilde{\omega}(k) \delta_\omega(\mathcal{O}_k^j) \\ \tilde{\omega}(k) \delta_v(\mathcal{O}_k^j) + \tilde{\mathcal{V}}(\mathcal{O}_k^j) \delta_\omega(\mathcal{O}_k^j) \end{bmatrix} \end{aligned} \quad (\text{E.1.14})$$

$\overset{\circ}{\delta}_{nd}^{\mathcal{V}}(\mathcal{O}_k^j) \in \mathcal{R}^6$ denotes the time derivative of the $\delta_{nd}^{\mathcal{V}}(\mathcal{O}_k^j)$ node deformation spatial velocity with respect to the k^{th} body frame. $\mathbf{a}_f(k, \mathcal{O}_k^j)$ denotes the Coriolis spatial acceleration for the \mathcal{O}_k^j node.

Using Eq. E.1.14, the equations of motion for the \mathcal{O}_k^j node in terms of the k^{th} body's spatial acceleration $\alpha(k)$ and the node spatial acceleration with respect to the body frame $\overset{\circ}{\delta}_{nd}^{\mathcal{V}}(\mathcal{O}_k^j)$ are thus given by

$$\mathbf{f}_{nd}(\mathcal{O}_k^j) \stackrel{\text{E.1.14, E.1.9}}{=} \underline{\mathbf{M}}_{nd}(\mathcal{O}_k^j) \left[\Phi^*(k, \mathcal{O}_k^j) \alpha(k) + \overset{\circ}{\delta}_{nd}^{\mathcal{V}}(\mathcal{O}_k^j) \right] + \mathbf{Q}(\mathcal{O}_k^j) + \mathbf{f}_{nd}^{\text{st}}(\mathcal{O}_k^j) \quad (\text{E.1.15})$$

where

$$\mathbf{Q}(\mathcal{O}_k^j) \triangleq \underline{\mathbf{M}}_{nd}(\mathcal{O}_k^j) \mathbf{a}_f(k, \mathcal{O}_k^j) + \mathbf{b}(\mathcal{O}_k^j) = \begin{bmatrix} \tilde{\omega}(\mathcal{O}_k^j) \underline{\mathcal{J}}(\mathcal{O}_k^j) \omega(\mathcal{O}_k^j) + \underline{\mathcal{J}}(\mathcal{O}_k^j) \tilde{\omega}(k) \delta_\omega(\mathcal{O}_k^j) \\ + \mathbf{m}(\mathcal{O}_k^j) \underline{\mathbf{p}}(\mathcal{O}_k^j) \tilde{\omega}(k) \left(\mathbf{v}(\mathcal{O}_k^j) + \delta_v(\mathcal{O}_k^j) \right) \\ \mathbf{m}(\mathcal{O}_k^j) \left[-\underline{\mathbf{p}}(\mathcal{O}_k^j) \tilde{\omega}(k) \delta_\omega(\mathcal{O}_k^j) - \tilde{\omega}(\mathcal{O}_k^j) \underline{\mathbf{p}}(\mathcal{O}_k^j) \omega(\mathcal{O}_k^j) \right. \\ \left. + \tilde{\omega}(k) \left(\mathbf{v}(\mathcal{O}_k^j) + \delta_v(\mathcal{O}_k^j) \right) \right] \end{bmatrix} \quad (\text{E.1.16})$$

This form of the equations of motion for a node is useful since it is about the k^{th} body's frame, which can be used for all the nodes on the body.

E.1.3 Modal representation at the node level

With the small deformation assumption, the relative attitude rotation matrix for the \mathcal{O}_k^j with respect to the k^{th} body, $\delta_r(\mathcal{O}_k^j)$ is given by

$$\delta_r(\mathcal{O}_k^j) \triangleq \exp \left[\tilde{\delta}_q(\mathcal{O}_k^j) \right] \approx \mathbf{I}_3 + \tilde{\delta}_q(\mathcal{O}_k^j) \in \mathcal{R}^{3 \times 3} \quad (\text{E.1.17})$$

where $\delta_q(\mathcal{O}_k^j) \in \mathcal{R}^3$ denotes a three-parameter representation of orientation such as Euler angles or a Rodriguez vector. In the above, the last expression on the right is obtained by ignoring the higher-order terms based on the small deformation assumption. The six-dimensional spatial deformation (slope plus translational) of node \mathcal{O}_k^j (with respect to frame \mathcal{B}_k) is denoted $\mathbf{u}_{\text{nd}}(\mathcal{O}_k^j) \in \mathcal{R}^6$, and has the form:

$$\mathbf{u}_{\text{nd}}(\mathcal{O}_k^j) = \begin{bmatrix} \delta_q(\mathcal{O}_k^j) \\ \delta_l(\mathcal{O}_k^j) \end{bmatrix} \in \mathcal{R}^6 \quad (\text{E.1.18})$$

A modal representation represents that deformation of the k^{th} body as a linear combination basis mode shapes with the r^{th} mode shape denoted $\Pi_r(k) \in \mathcal{R}^{6n_{\text{nd}}(k)}$. The coefficients of the mode basis functions are denoted $\eta(k)$ and are referred to as modal coordinates. In control analysis, model reduction from $6n_{\text{nd}}(k)$ nodal degrees of freedom to a smaller size is performed in modal space by retaining only a much smaller number $n_{\text{md}}(k)$ of the mode shapes. In the modal representation, the $\mathbf{u}_{\text{nd}}(\mathcal{O}_k^j)$ spatial deformation of the \mathcal{O}_k^j node can be expressed using the $\eta(k)$ modal coordinates as follows:

$$\mathbf{u}_{\text{nd}}(\mathcal{O}_k^j) \stackrel{\text{E.1.18}}{=} \begin{bmatrix} \delta_q(\mathcal{O}_k^j) \\ \delta_l(\mathcal{O}_k^j) \end{bmatrix} = \sum_{r=1}^{n_{\text{md}}(k)} \Pi_r(\mathcal{O}_k^j) \eta_r(k) \quad \text{where} \quad \Pi_r(\mathcal{O}_k^j) \triangleq \begin{pmatrix} \lambda_r^j(k) \\ \gamma_r^j(k) \end{pmatrix} \in \mathcal{R}^6 \quad (\text{E.1.19})$$

$\Pi_r(\mathcal{O}_k^j)$ denotes the contribution of the r^{th} mode to the spatial deformation for the \mathcal{O}_k^j node and $\eta_r(k)$ denotes the r^{th} element of $\eta(k)$. The $\lambda^j(k) \in \mathcal{R}^{3 \times n_{\text{md}}(k)}$ and $\gamma^j(k) \in \mathcal{R}^{3 \times n_{\text{md}}(k)}$ are the modal slope displacement influence vector and the modal translational displacement influence vector, respectively, for node \mathcal{O}_k^j . They define the contribution of the various modes to the slope (or differential change in orientation) and translational deformation for the \mathcal{O}_k^j node on the k^{th} body.

Eq. E.1.19 can be re-expressed as

$$\mathbf{u}_{\text{nd}}(\mathcal{O}_k^j) = \Pi(\mathcal{O}_k^j) \eta(k) \quad \text{where} \quad \Pi(\mathcal{O}_k^j) \triangleq \left[\Pi_1(\mathcal{O}_k^j), \dots, \Pi_{n_{\text{md}}(k)}(\mathcal{O}_k^j) \right]_{r=1}^{n_{\text{md}}(k)} \in \mathcal{R}^{6 \times n_{\text{md}}(k)} \quad (\text{E.1.20})$$

where $\Pi(\mathcal{O}_k^j)$ is referred to as the modal spatial displacement influence vector for node \mathcal{O}_k^j and are assumed to be constant and not depend on the deformation coordinates. Using modal velocity coordinates, the deformation spatial velocity $\delta_{\text{nd}}^v(\mathcal{O}_k^j)$ of the \mathcal{O}_k^j node with respect to the \mathcal{B}_k body frame can be obtained by taking the time derivative of Eq. E.1.19 to obtain the expression

$$\delta_{\text{nd}}^v(\mathcal{O}_k^j) \stackrel{\text{E.1.20, E.1.8}}{=} \Pi(\mathcal{O}_k^j) \dot{\eta}(k) \quad (\text{E.1.21})$$

E.1.4 Modal representation at the body level

Define the overall deformation field, $\mathbf{u}_{\text{nd}}(k)$, and nodal spatial velocity, $\mathcal{V}_{\text{nd}}(k)$, stacked vectors for the k^{th} body as

$$\begin{aligned} \mathbf{u}_{\text{nd}}(k) &\triangleq \text{col} \left\{ \mathbf{u}_{\text{nd}}(\mathcal{O}_k^j) \right\}_{j=1}^{n_{\text{nd}}(k)} \in \mathcal{R}^{6n_{\text{nd}}(k)} \\ \mathcal{V}_{\text{nd}}(k) &\triangleq \text{col} \left\{ \delta_{\text{nd}}^v(\mathcal{O}_k^j) \right\}_{j=1}^{n_{\text{nd}}(k)} \in \mathcal{R}^{6n_{\text{nd}}(k)} \end{aligned} \quad (\text{E.1.22})$$

With $\Pi(k)$ modal matrix for the k^{th} body defined as

$$\Pi(k) \triangleq \text{col} \left\{ \Pi(\mathcal{O}_k^j) \right\}_{j=1}^{n_{nd}(k)} \in \mathcal{R}^{6n_{nd}(k) \times n_{nd}(k)} \quad (\text{E.1.23})$$

we have

$$\mathbf{u}_{nd}(k) \stackrel{\text{E.1.20}}{=} \Pi(k)\dot{\eta}(k) \quad \text{and} \quad \delta_{nd}^{\mathcal{V}}(k) = \Pi(k)\dot{\eta}(k) \quad (\text{E.1.24})$$

The r^{th} column of $\Pi(k)$ is denoted $\Pi_r(k) \in \mathcal{R}^{6n_{nd}(k)}$, and is the mode shape for the r^{th} assumed mode for the k^{th} body.

Now define the stacked vector of the spatial velocities of all the nodes on the body as:

$$\mathcal{V}_{nd}(k) \triangleq \text{col} \left\{ \mathcal{V}(\mathcal{O}_k^j) \right\}_{j=1}^{n_{nd}(k)} \in \mathcal{R}^{6n_{nd}(k)} \quad (\text{E.1.25})$$

It follows from Eq. E.1.5 that

$$\mathcal{V}_{nd}(k) = \mathcal{B}^*(k)\mathcal{V}(k) + \delta_{nd}^{\mathcal{V}}(k) \quad (\text{E.1.26})$$

where

$$\mathcal{B}(k) \triangleq \left[\phi(k, \mathcal{O}_k^1), \phi(k, \mathcal{O}_k^2), \dots, \phi(k, \mathcal{O}_k^{n_{nd}(k)}) \right] \in \mathcal{R}^{6 \times 6n_{nd}(k)} \quad (\text{E.1.27})$$

Thus

$$\begin{aligned} \mathcal{V}_{nd}(k) = Y(k)\mathcal{V}_{fl}(k) \in \mathcal{R}^{6n_{nd}(k)} \quad \text{with} \quad \mathcal{V}_{fl}(k) &\triangleq \begin{bmatrix} \dot{\eta}(k) \\ \mathcal{V}(k) \end{bmatrix} \in \mathcal{R}^{\tilde{N}(k)} \\ \text{and} \quad Y(k) &\triangleq [\Pi(k), \mathcal{B}^*(k)] \in \mathcal{R}^{6n_{nd}(k) \times \tilde{N}(k)} \end{aligned} \quad (\text{E.1.28})$$

where $\Pi(k)$ and $\mathcal{B}^*(k)$ are as defined in Eq. E.1.23 and Eq. E.1.27 respectively. $\mathcal{V}_{fl}(k)$ represents the combined rigid and flex generalized velocities for the k^{th} body. $Y(k)$ defines the mapping between the generalized velocities for the k^{th} body and the deformation spatial velocities for all the nodes on the body. Note that $Y(k)$ can be expressed as

$$\begin{aligned} Y(k) &= \text{col} \left\{ Y(\mathcal{O}_k^j) \right\}_{j=1}^{n_{nd}(k)} \\ \text{where} \quad Y(\mathcal{O}_k^j) &\triangleq [\Pi(\mathcal{O}_k^j), \phi(k, \mathcal{O}_k^j)] \in \mathcal{R}^{6 \times \tilde{N}(k)} \end{aligned} \quad (\text{E.1.29})$$

E.1.5 Equations of motion for the k^{th} flexible body

For a single free-floating body, we would like to derive the equations of motion using the $\mathcal{V}_{fl}(k)$ modal velocity coordinates instead of the $\mathcal{V}_{nd}(k)$ nodal velocity coordinates.

Now let $\underline{M}_{nd}^k(\mathcal{O}_k^j)$ denote the spatial inertia of the \mathcal{O}_k^j node about the k^{th} body's reference frame, which by the parallel axis theorem for spatial inertias leads to

$$\underline{M}_{nd}^k(\mathcal{O}_k^j) \triangleq \phi(k, \mathcal{O}_k^j) \underline{M}_{nd}(\mathcal{O}_k^j) \phi^*(k, \mathcal{O}_k^j) = \begin{pmatrix} \underline{\mathcal{J}}^k(\mathcal{O}_k^j) & \mathbf{m}(\mathcal{O}_k^j) \underline{\tilde{p}}^k(\mathcal{O}_k^j) \\ -\mathbf{m}(\mathcal{O}_k^j) \underline{\tilde{p}}^k(\mathcal{O}_k^j) & \mathbf{m}(\mathcal{O}_k^j) \mathbf{I} \end{pmatrix} \quad (\text{E.1.30})$$

with

$$\begin{aligned}\underline{\mathcal{J}}^k(\mathcal{O}_k^j) &\triangleq \underline{\mathcal{J}}(\mathcal{O}_k^j) - \mathbf{m}(\mathcal{O}_k^j) \left[\tilde{\mathbf{l}}(\mathcal{O}_k^j) \tilde{\mathbf{p}}(\mathcal{O}_k^j) + \tilde{\mathbf{p}}(\mathcal{O}_k^j) \tilde{\mathbf{l}}(\mathcal{O}_k^j) + \tilde{\mathbf{l}}(\mathcal{O}_k^j) \tilde{\mathbf{l}}(\mathcal{O}_k^j) \right] \\ \underline{\mathbf{p}}^k(\mathcal{O}_k^j) &\triangleq \mathbf{l}(\mathcal{O}_k^j) + \underline{\mathbf{p}}(\mathcal{O}_k^j)\end{aligned}\quad (\text{E.1.31})$$

Unlike $\mathbf{M}_{\text{nd}}(\mathcal{O}_k^j)$, which is constant, $\underline{\mathbf{M}}_{\text{nd}}(\mathcal{O}_k^j)$ and $\underline{\mathbf{M}}_{\text{nd}}^k(\mathcal{O}_k^j)$ are both deformation-dependent. The overall, deformation-dependent, spatial inertia of the k^{th} body about its body frame is simply

$$\mathbf{M}(\mathbf{k}) = \sum_{j=1}^{n_{\text{nd}}(\mathbf{k})} \underline{\mathbf{M}}_{\text{nd}}^k(\mathcal{O}_k^j) \quad (\text{E.1.32})$$

The k^{th} body kinetic energy is

$$\mathcal{K}_e(\mathbf{k}) = \frac{1}{2} \mathcal{V}_{\text{fl}}^*(\mathbf{k}) \mathbf{M}_{\text{fl}}(\mathbf{k}) \mathcal{V}_{\text{fl}}(\mathbf{k}) \quad (\text{E.1.33})$$

with k^{th} body mass matrix

$$\mathbf{M}_{\text{fl}}(\mathbf{k}) = \mathbf{Y}^*(\mathbf{k}) \underline{\mathbf{M}}_{\text{nd}}(\mathbf{k}) \mathbf{Y}(\mathbf{k}) \in \mathcal{R}^{\tilde{\mathcal{N}}(\mathbf{k}) \times \tilde{\mathcal{N}}(\mathbf{k})} \quad \text{where} \quad \underline{\mathbf{M}}_{\text{nd}}(\mathbf{k}) \triangleq \text{diag} \left\{ \underline{\mathbf{M}}_{\text{nd}}(\mathcal{O}_k^j) \right\} \in \mathcal{R}^{6n_{\text{nd}}(\mathbf{k}) \times 6n_{\text{nd}}(\mathbf{k})} \quad (\text{E.1.34})$$

The diagonal structure of $\underline{\mathbf{M}}_{\text{nd}}(\mathbf{k})$ is a consequence of the lumped-mass structural model assumption for the body.

The equations of motion for the k^{th} flexible body (by itself) take the form

$$\begin{aligned}\mathcal{T}_{\text{fl}}(\mathbf{k}) &\triangleq \mathbf{Y}^*(\mathbf{k}) \text{col} \left\{ \underline{\mathbf{f}}_{\text{nd}}(\mathcal{O}_k^j) \right\}_{j=1}^{n_{\text{nd}}(\mathbf{k})} \stackrel{\text{E.1.34, E.1.15}}{=} \mathbf{M}_{\text{fl}}(\mathbf{k}) \boldsymbol{\alpha}_{\text{fl}}(\mathbf{k}) + \mathbf{b}_{\text{fl}}(\mathbf{k}) + \mathbf{Y}^* \mathbf{f}_{\text{nd}}^{\text{st}}(\mathcal{O}_k^j) \\ \mathbf{b}_{\text{fl}}(\mathbf{k}) &\triangleq \mathbf{Y}^* \mathbf{Q}(\mathcal{O}_k^j) \stackrel{\text{E.1.16}}{=} \mathbf{Y}^* \left[\underline{\mathbf{M}}_{\text{nd}}(\mathcal{O}_k^j) \mathbf{a}_f(\mathbf{k}, \mathcal{O}_k^j) + \mathbf{b}(\mathcal{O}_k^j) \right]\end{aligned}\quad (\text{E.1.35})$$

The $\mathbf{Y}^*(\mathbf{k})$ and $\underline{\mathbf{M}}_{\text{nd}}(\mathbf{k})$ matrices in Eq. E.1.34 can be large, and evaluating the expressions using them can be computationally expensive. However, the use of modal integrals that can be computed offline from the structural dynamics model can significantly reduce computation cost. The use of modal integrals to examine the structure of the $\mathbf{M}_{\text{fl}}(\mathbf{k})$ body mass matrix is discussed in Section E.2, with expressions for its block components derived in Exercises E.2.1, E.2.2 and E.2.3. The expressions for $\mathbf{b}_{\text{fl}}(\mathbf{k})$ using modal integrals are derived in Section E.2.5.

E.2 Expressions for $\mathbf{M}_{\text{fl}}(\mathbf{k})$

From Eq. E.1.28 and Eq. E.1.34 the modal mass matrix of the k^{th} body is given by

$$\begin{aligned}\mathbf{M}_{\text{fl}}(\mathbf{k}) &= \begin{pmatrix} \Pi^*(\mathbf{k}) \\ \mathbf{B}(\mathbf{k}) \end{pmatrix} \underline{\mathbf{M}}_{\text{nd}}(\mathbf{k}) [\Pi(\mathbf{k}), \mathbf{B}^*(\mathbf{k})] \\ &= \begin{pmatrix} \Pi^*(\mathbf{k}) \underline{\mathbf{M}}_{\text{nd}}(\mathbf{k}) \Pi(\mathbf{k}) & \Pi^*(\mathbf{k}) \underline{\mathbf{M}}_{\text{nd}}(\mathbf{k}) \mathbf{B}^*(\mathbf{k}) \\ \mathbf{B}(\mathbf{k}) \underline{\mathbf{M}}_{\text{nd}}(\mathbf{k}) \Pi(\mathbf{k}) & \mathbf{B}(\mathbf{k}) \underline{\mathbf{M}}_{\text{nd}}(\mathbf{k}) \mathbf{B}^*(\mathbf{k}) \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{M}_{\text{fl}}^{\text{ff}}(\mathbf{k}) & \mathbf{M}_{\text{fl}}^{\text{fr}}(\mathbf{k}) \\ \mathbf{M}_{\text{fl}}^{\text{rf}}(\mathbf{k}) & \mathbf{M}_{\text{fl}}^{\text{rr}}(\mathbf{k}) \end{pmatrix} \in \mathcal{R}^{\tilde{\mathcal{N}}(\mathbf{k}) \times \tilde{\mathcal{N}}(\mathbf{k})}\end{aligned}\quad (\text{E.2.1})$$

The $\underline{\mathbf{M}}_{\text{nd}}(\mathcal{O}_k^j)$ diagonal block elements of $\underline{\mathbf{M}}_{\text{nd}}(\mathbf{k})$ are the spatial inertias of the individual nodes in the k^{th} body frame defined in Eq. E.1.3 on page 65.

E.2.1 Expression for $M_{fl}^{ff}(k)$ using modal integrals

Define the matrix $G^k \in \mathcal{R}^{n_m(k) \times n_m(k)}$ with $(r, s)^{th}$ element denoted $G^k(r, s)$. It can be shown that the $M_{fl}^{ff}(k)$ modal/modal sub-block of the body mass matrix can be expressed as:

$$M_{fl}^{ff}(k) = G^k \quad \text{where}$$

$$G^k(r, s) \triangleq G_0^k(r, s) + \sum_{l=1}^{n_m(k)} [G_1^k(r, s, l) + G_1^k(s, r, l)] \eta_l(k) + \sum_{l=1}^{n_m(k)} \sum_{m=1}^{n_m(k)} G_2^k(r, s, l, m) \eta_l(k) \eta_m(k) \in \mathcal{R}^1 \quad (E.2.2)$$

with

$$G_0^k(r, s) \triangleq \sum_{j=1_k}^{n_{nd}(k)} [\lambda_r^j(k)]^* \mathcal{J}(\mathcal{O}_k^j) \lambda_s^j(k) + m(\mathcal{O}_k^j) \left\{ [\lambda_r^j(k)]^* \tilde{p}(\mathcal{O}_k^j) \gamma_s^j(k) + [\lambda_s^j(k)]^* \tilde{p}(\mathcal{O}_k^j) \gamma_r^j(k) + [\gamma_r^j(k)]^* \gamma_s^j(k) \right\} \in \mathcal{R}^1$$

$$G_1^k(r, s, l) \triangleq \sum_{j=1_k}^{n_{nd}(k)} [\lambda_r^j(k)]^* \tilde{\lambda}_l^j(k) \mathcal{J}(\mathcal{O}_k^j) \lambda_s^j(k) + m(\mathcal{O}_k^j) [\lambda_r^j(k)]^* [\tilde{\lambda}_l^j(k) p(\mathcal{O}_k^j)]^\times \gamma_s^j(k) \in \mathcal{R}^1$$

$$G_2^k(r, s, l, m) \triangleq - \sum_{j=1_k}^{n_{nd}(k)} [\lambda_r^j(k)]^* \tilde{\gamma}_l^j(k) \mathcal{J}(\mathcal{O}_k^j) \tilde{\gamma}_m^j(k) \lambda_s^j(k) \in \mathcal{R}^1$$

Note that

$$G^k(r, s) = G^k(s, r)$$

$G_0^k(r, s)$, $G_1^k(r, s, l)$ and $G_2^k(r, s, l, m)$ are the first of several modal integrals that we will encounter. Modal integrals are constants that do not depend on the deformation coordinates and can be pre-computed from the structural deformation model data and used directly while solving the equations of motion.

E.2.2 Expression for $M_{fl}^{rr}(k)$ using modal integrals

The $M_{fl}^{rr}(k)$ rigid/rigid sub-block of the body mass matrix can similarly be shown to be expressed as:

$$M_{fl}^{rr}(k) = \begin{pmatrix} \mathcal{J}(k) & m(k) \tilde{p}(k) \\ -m(k) \tilde{p}(k) & m(k) \mathbf{I}_3 \end{pmatrix} \quad \text{where}$$

$$m(k) = \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j), \quad p(k) = p_0^k + \sum_{r=1}^{n_m(k)} p_1^k(r) \eta_r(k) \quad (E.2.3)$$

$$\mathcal{J}(k) = \mathcal{J}_0^k + \sum_{r=1}^{n_m(k)} \left[\mathcal{J}_1^k(r) + [\mathcal{J}_1^k(r)]^* \right] \eta_r(k) + \sum_{r=1}^{n_m(k)} \sum_{s=1}^{n_m(k)} \mathcal{J}_2^k(r, s) \eta_r(k) \eta_s(k)$$

and with

$$p_0^k \triangleq [1/m(k)] \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) [p(\mathcal{O}_k^j) + l_0(k, \mathcal{O}_k^j)] \in \mathcal{R}^3$$

$$\begin{aligned}
 \mathbf{p}_1^k(\mathbf{r}) &\triangleq [1/m(k)] \sum_{j=1_k}^{n_{nd}(k)} \mathbf{m}(\mathcal{O}_k^j) \left[\gamma_r^j(k) + \tilde{\lambda}_l^j(k) \mathbf{p}(\mathcal{O}_k^j) \right] \in \mathcal{R}^3 \\
 \mathcal{J}_0^k &\triangleq \sum_{j=1_k}^{n_{nd}(k)} \mathcal{J}(\mathcal{O}_k^j) - \mathbf{m}(\mathcal{O}_k^j) \left[\tilde{l}_0(k, \mathcal{O}_k^j) \tilde{l}_0(k, \mathcal{O}_k^j) \right. \\
 &\quad \left. + \tilde{\mathbf{p}}(\mathcal{O}_k^j) \tilde{l}_0(k, \mathcal{O}_k^j) + \tilde{l}_0(k, \mathcal{O}_k^j) \tilde{\mathbf{p}}(\mathcal{O}_k^j) \right] \in \mathcal{R}^{3 \times 3} \\
 \mathcal{J}_1^k(\mathbf{r}) &\triangleq \sum_{j=1_k}^{n_{nd}(k)} \left[\tilde{\lambda}_r^j(k) \mathcal{J}(\mathcal{O}_k^j) - \mathbf{m}(\mathcal{O}_k^j) \left\{ \tilde{\gamma}_r^j(k) \left[\tilde{l}_0(k, \mathcal{O}_k^j) + \tilde{\mathbf{p}}(\mathcal{O}_k^j) \right] \right. \right. \\
 &\quad \left. \left. + \tilde{l}_0(k, \mathcal{O}_k^j) \left[\tilde{\lambda}_r^j(k) \mathbf{p}(\mathcal{O}_k^j) \right]^\times \right\} \right] \in \mathcal{R}^{3 \times 3} \\
 \mathcal{J}_2^k(\mathbf{r}, \mathbf{s}) &\triangleq - \sum_{j=1_k}^{n_{nd}(k)} \left[\tilde{\lambda}_r^j(k) \mathcal{J}(\mathcal{O}_k^j) \tilde{\lambda}_s^j(k) + \mathbf{m}(\mathcal{O}_k^j) \left\{ \tilde{\gamma}_r^j(k) \tilde{\gamma}_s^j(k) \right. \right. \\
 &\quad \left. \left. + \tilde{\gamma}_r^j(k) \left[\tilde{\lambda}_s^j(k) \mathbf{p}(\mathcal{O}_k^j) \right]^\times + \left[\tilde{\lambda}_s^j(k) \mathbf{p}(\mathcal{O}_k^j) \right]^\times \tilde{\gamma}_r^j(k) \right\} \right] \in \mathcal{R}^{3 \times 3}
 \end{aligned} \tag{E.2.4}$$

Note that

$$\mathcal{J}_2^k(\mathbf{r}, \mathbf{s}) = [\mathcal{J}_2^k(\mathbf{s}, \mathbf{r})]^*$$

$\mathbf{m}(k)$, \mathbf{p}_0^k , $\mathbf{p}_1^k(\mathbf{r})$, \mathcal{J}_0^k , $\mathcal{J}_1^k(\mathbf{r})$ and $\mathcal{J}_2^k(\mathbf{r}, \mathbf{s})$ are additional instances of modal integrals that can be pre-computed from structural deformation model data.

E.2.3 Expression for $M_{fl}^{rf}(k)$ using modal integrals

The $M_{fl}^{rf}(k)$ rigid/modal sub-block of the body mass matrix can be expressed as:

$$\begin{aligned}
 M_{fl}^{rf}(k) &= \begin{pmatrix} \mathbf{F}^k \\ \mathbf{E}^k \end{pmatrix} \quad \text{where} \quad \mathbf{F}^k \triangleq \begin{pmatrix} [\mathbf{F}^k(1), \dots, \mathbf{F}^k(n_m(k))] \\ [\mathbf{E}^k(1), \dots, \mathbf{E}^k(n_m(k))] \end{pmatrix} \quad \text{where} \\
 \mathbf{F}^k(\mathbf{r}) &= \mathbf{F}_0^k(\mathbf{r}) + \sum_{s=1}^{n_m(k)} \mathbf{F}_1^k(\mathbf{r}, s) \eta_s(k) + \sum_{s=1}^{n_m(k)} \sum_{l=1}^{n_m(k)} \mathbf{F}_2^k(\mathbf{r}, s, l) \eta_s(k) \eta_l(k) \in \mathcal{R}^3 \\
 \mathbf{E}^k(\mathbf{r}) &= \mathbf{E}_0^k(\mathbf{r}) + \sum_{s=1}^{n_m(k)} \mathbf{E}_1^k(\mathbf{r}, s) \eta_s(k) \in \mathcal{R}^3 \quad \text{with} \\
 \mathbf{F}_0^k(\mathbf{r}) &\triangleq \sum_{j=1_k}^{n_{nd}(k)} \left[\mathcal{J}(\mathcal{O}_k^j) - \mathbf{m}(\mathcal{O}_k^j) \tilde{l}_0(k, \mathcal{O}_k^j) \tilde{\mathbf{p}}(\mathcal{O}_k^j) \right] \lambda_r^j(k) \\
 &\quad + \mathbf{m}(\mathcal{O}_k^j) \left[\tilde{l}_0(k, \mathcal{O}_k^j) + \tilde{\mathbf{p}}(\mathcal{O}_k^j) \right] \gamma_r^j(k) \in \mathcal{R}^3 \\
 \mathbf{F}_1^k(\mathbf{r}, s) &\triangleq \sum_{j=1_k}^{n_{nd}(k)} \left[\tilde{\lambda}_s^j(k) \mathcal{J}(\mathcal{O}_k^j) - \mathcal{J}(\mathcal{O}_k^j) \tilde{\lambda}_s^j(k) \right] \lambda_r^j(k) \\
 &\quad + \mathbf{m}(\mathcal{O}_k^j) \left\{ \tilde{\gamma}_r^j(k) \left[-\gamma_s^j(k) + \tilde{\mathbf{p}}(\mathcal{O}_k^j) \lambda_s^j(k) \right] \right. \\
 &\quad \left. - \tilde{l}_0(k, \mathcal{O}_k^j) \left[\tilde{\lambda}_s^j(k) \mathbf{p}(\mathcal{O}_k^j) \right]^\times \lambda_r^j(k) - \left[\tilde{\lambda}_r^j(k) \mathbf{p}(\mathcal{O}_k^j) \right]^\times \gamma_s^j(k) \right\} \in \mathcal{R}^3
 \end{aligned} \tag{E.2.5}$$

$$\begin{aligned}
 F_2^k(r, s, l) &\triangleq - \sum_{j=1_k}^{n_{nd}(k)} \left[\tilde{\lambda}_s^j(k) \mathcal{J}(\mathcal{O}_k^j) \tilde{\lambda}_l^j(k) + m(\mathcal{O}_k^j) \tilde{\gamma}_s^j(k) \left[\tilde{\lambda}_l^j(k) p(\mathcal{O}_k^j) \right]^\times \right] \lambda_r^j(k) \in \mathcal{R}^3 \\
 E_0^k(r) &\triangleq \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \left[\gamma^j(k) - \tilde{p}(\mathcal{O}_k^j) \lambda_r^j(k) \right] \stackrel{E.2.4}{=} m(k) p_1^k(r) \in \mathcal{R}^3 \\
 E_1^k(r, s) &\triangleq - \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \left[\tilde{\lambda}_s^j(k) p(\mathcal{O}_k^j) \right]^\times \lambda_r^j(k) \in \mathcal{R}^3
 \end{aligned}$$

$E_0^k(r)$, $E_1^k(r, s)$, $F_0^k(r)$, $F_1^k(r, s)$, and $F_2^k(r, s, l)$ are additional modal integrals.

E.2.4 Overall structure of $M_{fl}(k)$

The overall structure of the $M_{fl}(k)$ k^{th} body mass matrix is clearer to see in the following block-partitioned form:

$$\begin{aligned}
 M_{fl}(k) &= \left(\begin{array}{c|c|c} G^k & [F^k]^* & [E^k]^* \\ \hline F^k & \mathcal{J}(k) & m(k) \tilde{p}(k) \\ \hline E^k & -m(k) \tilde{p}(k) & m(k) I_3 \end{array} \right) = \underbrace{\left(\begin{array}{c|c|c} G^k & [F_0^k]^* & [E^k]^* \\ \hline F_0^k & \mathcal{J}_0^k & m(k) \tilde{p}_0^k \\ \hline E^k & -m(k) \tilde{p}_0^k & m(k) I_3 \end{array} \right)}_{M_{fl}^0(k)} \quad (E.2.6) \\
 &+ \underbrace{\left(\begin{array}{c|c|c} \sum_{l=1}^{n_m(k)} G_1^k(r, s, l) \eta_l(k) & \left[\sum_{r=1}^{n_m(k)} F_1^k(r) \eta_r(k) \right]^* & 0 \\ \hline \sum_{r=1}^{n_m(k)} F_1^k(r) \eta_r(k) & \sum_{r=1}^{n_m(k)} \left[\mathcal{J}_1^k(r) + [\mathcal{J}_1^k(r)]^* \right] \eta_r(k) & m(k) \left[\sum_{r=1}^{n_m(k)} p_1^k(r) \eta_r(k) \right]^\times \\ \hline 0 & -m(k) \left[\sum_{r=1}^{n_m(k)} p_1^k(r) \eta_r(k) \right]^\times & 0 \end{array} \right)}_{M_m^1(k)} \\
 &+ \underbrace{\left(\begin{array}{c|c|c} \sum_{l=1}^{n_m(k)} \sum_{m=1}^{n_m(k)} G_2^k(r, s, l, m) \eta_l(k) \eta_m(k) & 0 & 0 \\ \hline 0 & \sum_{r=1}^{n_m(k)} \sum_{s=1}^{n_m(k)} \mathcal{J}_2^k(r, s) \eta(r) \eta(s) & 0 \\ \hline 0 & 0 & 0 \end{array} \right)}_{M_m^2(k)}
 \end{aligned}$$

The $i = 0, 1, 2$ subscripts in $M_{fl}^i(k)$ terms denote the order of dependency of the terms on the deformation-generalized coordinates.

E.2.5 Equations of motion in modal coordinates

In partitioned form, $\mathbf{b}_{fl}(\mathbf{k})$ can be expressed as

$$\mathbf{b}_{fl}(\mathbf{k}) \stackrel{\text{E.1.35}}{=} \mathbf{Y}^* \mathbf{Q}(\mathcal{O}_k^j) = \begin{bmatrix} \mathbf{b}_\eta^k(1) \\ \vdots \\ \mathbf{b}_\eta^k(n_m(\mathbf{k})) \\ \hline \mathbf{b}_\omega^k \\ \hline \mathbf{b}_v^k \end{bmatrix} \quad (\text{E.2.7})$$

We now show expressions for $\mathbf{b}_\eta^k(\mathbf{r})$, \mathbf{b}_ω^k and \mathbf{b}_v^k below. Using Eq. E.1.16, $\mathbf{Q}(\mathcal{O}_k^j)$ can be re-expressed as:

$$\mathbf{Q}(\mathcal{O}_k^j) = \frac{\begin{bmatrix} \left\{ \tilde{\omega}(\mathbf{k}) \left(\underline{\mathcal{J}}(\mathcal{O}_k^j) - \mathbf{m}(\mathcal{O}_k^j) \tilde{\mathbf{l}}(\mathcal{O}_k^j) \tilde{\mathbf{p}}(\mathcal{O}_k^j) \right) + \tilde{\delta}_\omega(\mathcal{O}_k^j) \underline{\mathcal{J}}(\mathcal{O}_k^j) - \underline{\mathcal{J}}(\mathcal{O}_k^j) \tilde{\delta}_\omega(\mathcal{O}_k^j) \right. \\ \left. - \mathbf{m}(\mathcal{O}_k^j) \tilde{\mathbf{p}}(\mathcal{O}_k^j) \left(\tilde{\mathbf{v}}(\mathbf{k}) + 2\tilde{\delta}_v(\mathcal{O}_k^j) \right) \right\} \omega(\mathbf{k}) \\ \left. + \tilde{\omega}(\mathbf{k}) \underline{\mathcal{J}}(\mathcal{O}_k^j) \delta_\omega(\mathcal{O}_k^j) + \tilde{\delta}_\omega(\mathcal{O}_k^j) \underline{\mathcal{J}}(\mathcal{O}_k^j) \delta_\omega(\mathcal{O}_k^j) \right]}{\begin{bmatrix} -\mathbf{m}(\mathcal{O}_k^j) \left[\left\{ \tilde{\omega}(\mathbf{k}) \left(\tilde{\mathbf{p}}(\mathcal{O}_k^j) + \tilde{\mathbf{l}}(\mathcal{O}_k^j) \right) + \left(\tilde{\mathbf{v}}(\mathbf{k}) + 2\tilde{\delta}_v(\mathcal{O}_k^j) \right) \right\} \omega(\mathbf{k}) \right. \\ \left. + \left(2\tilde{\omega}(\mathbf{k}) + \tilde{\delta}_\omega(\mathcal{O}_k^j) \right) \tilde{\mathbf{p}}(\mathcal{O}_k^j) \delta_\omega(\mathcal{O}_k^j) \right]} \end{bmatrix}} \quad (\text{E.2.8})$$

1. The following is the expression for the $\mathbf{b}_\eta^k(\mathbf{r})$ modal gyroscopic term in Eq. E.2.7:

$$\mathbf{b}_\eta^k(s) = \sum_{j=1_k}^{n_{nd}(\mathbf{k})} \Pi_s^*(\mathcal{O}_k^j) \mathbf{Q}(\mathcal{O}_k^j) \quad (\text{E.2.9})$$

2. With $M_R(\mathbf{k})$ denoting the deformation dependent instantaneous spatial inertia of the k^{th} body, the rigid-body \mathbf{b}_ω^k and \mathbf{b}_v^k gyroscopic terms are given by:

$$\begin{bmatrix} \mathbf{b}_\omega^k \\ \mathbf{b}_v^k \end{bmatrix} = \sum_{j=1}^{n_{nd}(\mathbf{k})} \phi(\mathbf{k}, \mathcal{O}_k^j) \mathbf{Q}(\mathcal{O}_k^j) \quad (\text{E.2.10})$$

$$= \bar{\mathbf{V}}(\mathbf{k}) \mathbf{M}(\mathbf{k}) \mathbf{V}(\mathbf{k}) + \sum_{j=1}^{n_{nd}(\mathbf{k})} \begin{bmatrix} \left[\tilde{\delta}_\omega(\mathcal{O}_k^j) \underline{\mathcal{J}}(\mathcal{O}_k^j) - \underline{\mathcal{J}}(\mathcal{O}_k^j) \tilde{\delta}_\omega(\mathcal{O}_k^j) \right] \omega(\mathbf{k}) \\ + \tilde{\omega}(\mathbf{k}) \underline{\mathcal{J}}(\mathcal{O}_k^j) \delta_\omega(\mathcal{O}_k^j) \\ + \tilde{\delta}_\omega(\mathcal{O}_k^j) \left(\underline{\mathcal{J}}(\mathcal{O}_k^j) \right. \\ \left. - \mathbf{m}(\mathcal{O}_k^j) \tilde{\mathbf{p}}(\mathcal{O}_k^j) \tilde{\mathbf{l}}(\mathcal{O}_k^j) \right) \delta_\omega(\mathcal{O}_k^j) \\ - \mathbf{m}(\mathcal{O}_k^j) \left\{ 2\tilde{\mathbf{p}}^k(\mathcal{O}_k^j) \tilde{\delta}_v(\mathcal{O}_k^j) \omega(\mathbf{k}) \right. \\ \left. + 2\tilde{\mathbf{l}}(\mathcal{O}_k^j) \tilde{\omega}(\mathbf{k}) \tilde{\mathbf{p}}(\mathcal{O}_k^j) \delta_\omega(\mathcal{O}_k^j) \right\} \\ \left. - \mathbf{m}(\mathcal{O}_k^j) \left\{ 2\tilde{\delta}_v(\mathcal{O}_k^j) \omega(\mathbf{k}) \right. \right. \\ \left. \left. + \left(2\tilde{\omega}(\mathbf{k}) + \tilde{\delta}_\omega(\mathcal{O}_k^j) \right) \tilde{\mathbf{p}}(\mathcal{O}_k^j) \delta_\omega(\mathcal{O}_k^j) \right\} \right] \end{bmatrix}$$

The first term on the right above reflects the rigid body contribution for the k^{th} body.

E.3 Special case: Point mass nodes

We now make the commonly used assumption that each node is a point mass, and hence $\mathcal{J}(\mathcal{O}_k^j) = 0$ and $p(\mathcal{O}_k^j) = 0$ for all nodes. Using this in the modal integral expressions obtained so far simplify to

$$\begin{aligned}
 p_0^k &\triangleq [1/m(k)] \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) l_0(k, \mathcal{O}_k^j) \in \mathcal{R}^3 \\
 p_1^k(r) &\triangleq [1/m(k)] \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \gamma_r^j(k) \in \mathcal{R}^3 \\
 \mathcal{J}_0^k &\triangleq - \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \tilde{l}_0(k, \mathcal{O}_k^j) \tilde{l}_0(k, \mathcal{O}_k^j) \in \mathcal{R}^{3 \times 3} \\
 \mathcal{J}_1^k(r) &\triangleq - \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \tilde{\gamma}_r^j(k) \tilde{l}_0(k, \mathcal{O}_k^j) \in \mathcal{R}^{3 \times 3} \\
 \mathcal{J}_2^k(r, s) &\triangleq - \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \tilde{\gamma}_r^j(k) \tilde{\gamma}_s^j(k) \in \mathcal{R}^{3 \times 3} \\
 F_0^k(r) &\triangleq \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \tilde{l}_0(k, \mathcal{O}_k^j) \gamma_r^j(k) \in \mathcal{R}^3 \\
 F_1^k(r, s) &\triangleq - \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \tilde{\gamma}_r^j(k) \gamma_s^j(k) \in \mathcal{R}^3 \\
 F_2^k(r, s, l) &\triangleq 0 \in \mathcal{R}^3 \\
 E_0^k(r) &\triangleq \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) \gamma^j(k) \stackrel{E.2.4}{=} m(k) p_1^k(r) \in \mathcal{R}^3 \\
 E_1^k(r, s) &\triangleq 0 \in \mathcal{R}^3 \\
 G_0^k(r, s) &\triangleq \sum_{j=1_k}^{n_{nd}(k)} m(\mathcal{O}_k^j) [\gamma_r^j(k)]^* \gamma_s^j(k) \in \mathcal{R}^1 \\
 G_1^k(r, s, l) &\triangleq 0 \in \mathcal{R}^1 \\
 G_2^k(r, s, l, m) &\triangleq 0 \in \mathcal{R}^1
 \end{aligned} \tag{E.3.1}$$

These modal integrals are computed by the FModal tool described in Appendix C from the finite-element structural analysis data for the component bodies.

E.3.1 Expression for $\mathbf{b}_{fl}(\mathbf{k})$ (point mass nodes)

For the simplified case where all the nodes on the k^{th} flexible body are point masses, i.e. their first and second moments of inertia are zero. The expression for \mathbf{Q} in Eq. E.2.8 simplifies to

$$\mathbf{Q}(\mathcal{O}_k^j) = \left[\frac{0}{\mathbf{m}(\mathcal{O}_k^j) \tilde{\omega}(\mathbf{k}) \left\{ \tilde{\omega}(\mathbf{k}) \mathbf{I}(\mathcal{O}_k^j) + \mathbf{v}(\mathbf{k}) + 2\delta_v(\mathcal{O}_k^j) \right\}} \right] \quad (\text{E.3.2})$$

Substituting this in Eq. E.2.7 and simplifying using the modal integrals in Section E.3 lead to the following expressions for the $\mathbf{b}_\eta^k(\mathbf{r})$, \mathbf{b}_ω^k and \mathbf{b}_v^k gyroscopic terms in Eq. E.2.7.

$$1. \quad \mathbf{b}_\eta^k(\mathbf{r}) = -\omega^*(\mathbf{k}) \left[\mathbf{m}(\mathbf{k}) \tilde{\mathbf{p}}_1^k(\mathbf{r}) \mathbf{v}(\mathbf{k}) + 2\mathbf{F}_1^k(\mathbf{r}) \dot{\eta}(\mathbf{k}) + [\mathcal{J}_1^k(\mathbf{r}) + \mathbf{X}^k(\mathbf{r}, \eta)] \omega(\mathbf{k}) \right] \quad (\text{E.3.3})$$

where

$$\mathbf{X}^k(\mathbf{r}, \eta) \triangleq \sum_{s=1}^{n_m(\mathbf{k})} \mathcal{J}_2^k(\mathbf{r}, s) \eta_s(\mathbf{k}) \in \mathcal{R}^{3 \times 3} \quad (\text{E.3.4})$$

2.

$$\begin{bmatrix} \mathbf{b}_\omega^k \\ \mathbf{b}_v^k \end{bmatrix} = \bar{\mathbf{V}}(\mathbf{k}) \mathbf{M}(\mathbf{k}) \mathbf{V}(\mathbf{k}) + \frac{\begin{bmatrix} \sum_{r=1}^{n_m(\mathbf{k})} (\mathcal{J}_1^k(\mathbf{r}) + \mathbf{X}(\mathbf{r}, \eta) + [\mathcal{J}_1^k(\mathbf{r}) + \mathbf{X}(\mathbf{r}, \eta)]^*) \dot{\eta}_r(\mathbf{k}) \omega(\mathbf{k}) \\ -\mathbf{m}(\mathbf{k}) \tilde{\mathbf{v}}(\mathbf{k}) \sum_{r=1}^{n_m(\mathbf{k})} \mathbf{p}_1^k(\mathbf{r}) \dot{\eta}_r(\mathbf{k}) \end{bmatrix}}{\begin{bmatrix} 2\mathbf{m}(\mathbf{k}) \tilde{\omega}(\mathbf{k}) \sum_{r=1}^{n_m(\mathbf{k})} \mathbf{p}_1^k(\mathbf{r}) \dot{\eta}_r(\mathbf{k}) \end{bmatrix}} \quad (\text{E.3.5})$$

Note that the last term on the right of Eq. E.3.5 is the standard gyroscopic force term for a rigid body with $\mathbf{M}(\mathbf{k})$ body spatial inertia.

The DARTS multibody engine (Section B.1.2) has an interface to import the modal integral data from the HDF5 file created by FModal and use it within its dynamics computations for solving the equations of motion.

E.4 Definition of Terms

\dot{x} denotes the time derivative of x with respect to the body-fixed (rotating) frame \mathbb{B}_k

$\tilde{x} = [x]^\times \in \mathcal{R}^{3 \times 3}$ and denotes the skew-symmetric cross product matrix associated with the 3-dimensional vector x

We use the following notation $\widetilde{(\quad)}$ and $\overline{(\quad)}$ notation for a spatial vector $z \triangleq \begin{bmatrix} x \\ y \end{bmatrix}$ as follows:

$$\tilde{z} \triangleq \begin{pmatrix} \tilde{x} & \mathbf{0}_3 \\ \tilde{y} & \tilde{x} \end{pmatrix} \in \mathcal{R}^{6 \times 6} \quad \text{and} \quad \bar{z} \triangleq \begin{pmatrix} \tilde{x} & \tilde{y} \\ \mathbf{0}_3 & \tilde{x} \end{pmatrix} \quad (\text{E.4.1})$$

ϕ^* = the transpose of matrix ϕ

$\mathcal{V}(k)$ = the spatial velocity of the k^{th} body frame \mathbb{B}_k with respect to the inertial frame and expressed in \mathbb{B}_k

$$\mathcal{V}(k) = \begin{bmatrix} \omega(k) \\ v(k) \end{bmatrix}$$

$\omega(k)$ = the rigid body angular velocity of body k with respect to the inertial frame and expressed in \mathbb{B}_k

$v(k)$ = the rigid body linear velocity of body k with respect to the inertial frame and expressed in \mathbb{B}_k

$\alpha(k)$ = time derivative of the rigid body spatial velocity $\mathcal{V}(k)$ with respect to \mathbb{B}_k and expressed in \mathbb{B}_k

$$\alpha(k) = \begin{bmatrix} \dot{\omega}(k) \\ \dot{v}(k) \end{bmatrix}$$

$\phi(x, y) = \begin{bmatrix} I_{3 \times 3} & \tilde{l}(x, y) \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}$ = the rigid body spatial transformation operator which transforms spatial velocities and forces between points x and y

\mathcal{O}_k^j = the origin of the j^{th} node of body k . Note that in general, body k is flexible and that \mathcal{O}_k^j is not necessarily at the node's center of gravity.

$\mathcal{V}(\mathcal{O}_k^j)$ = the spatial velocity of \mathcal{O}_k^j with respect to the inertial frame and expressed in \mathbb{B}_k

$$\mathcal{V}(\mathcal{O}_k^j) = \begin{bmatrix} \omega(\mathcal{O}_k^j) \\ v(\mathcal{O}_k^j) \end{bmatrix}$$

$\omega(\mathcal{O}_k^j)$ = the angular velocity of node \mathcal{O}_k^j with respect to the inertial frame and expressed in \mathbb{B}_k

$v(\mathcal{O}_k^j)$ = the linear velocity of node \mathcal{O}_k^j with respect to the inertial frame and expressed in \mathbb{B}_k

$u_{nd}(\mathcal{O}_k^j)$ = the spatial displacement of node \mathcal{O}_k^j with respect to $\mathbb{B}_k = \begin{bmatrix} \delta_q(\mathcal{O}_k^j) \\ \delta_l(\mathcal{O}_k^j) \end{bmatrix} = \Pi(\mathcal{O}_k^j) \eta(k)$

$\delta_{nd}^v(\mathcal{O}_k^j)$ = the time derivative of $u_{nd}(\mathcal{O}_k^j)$ with respect to $\mathbb{B}_k = \begin{bmatrix} \delta_\omega(\mathcal{O}_k^j) \\ \delta_v(\mathcal{O}_k^j) \end{bmatrix} = \Pi(\mathcal{O}_k^j) \dot{\eta}(k)$

$\Pi(\mathcal{O}_k^j)$ = the modal spatial displacement influence vector for node $\mathcal{O}_k^j = \left[\Pi_1(\mathcal{O}_k^j), \dots, \Pi_{n_{md}(k)}(\mathcal{O}_k^j) \right]_{r=1}^{n_{md}(k)}$

$\eta(k)$ = the modal representation time-domain basis functions (or modal coordinates)

$\delta_q(\mathcal{O}_k^j)$ = the attitude deformation of the \mathcal{O}_k^j node on the k^{th} flexible body with respect to \mathbb{B}_k

$\delta_l(\mathcal{O}_k^j)$ = the translational deformation of \mathcal{O}_k^j with respect to \mathbb{B}_k and expressed in \mathbb{B}_k

$\delta_\omega(\mathcal{O}_k^j)$ = the deformation angular velocity of \mathcal{O}_k^j with respect to the body frame \mathbb{B}_k and expressed in \mathbb{B}_k

$\delta_v(\mathcal{O}_k^j)$ = the deformation linear velocity of \mathcal{O}_k^j with respect to \mathbb{B}_k and expressed in $\mathbb{B}_k = \overset{\circ}{\delta}_l(\mathcal{O}_k^j)$

$l_0(k, \mathcal{O}_k^j)$ = the vector from \mathbb{B}_k to the location (before deformation) of \mathcal{O}_k^j

$l(k, \mathcal{O}_k^j)$ = the vector from \mathbb{B}_k to the location (after deformation) of $\mathcal{O}_k^j = l_0(k, \mathcal{O}_k^j) + \delta_l(\mathcal{O}_k^j)$

Using the five definitions above,

$$\begin{aligned}\omega(\mathcal{O}_k^j) &= \omega(k) + \delta_\omega(\mathcal{O}_k^j) \\ v(\mathcal{O}_k^j) &= v(k) - \tilde{l}(k, \mathcal{O}_k^j) \omega(k) + \delta_v(\mathcal{O}_k^j)\end{aligned}$$

Using the definition of the rigid body spatial transformation operator, we can express the two equations above as

$$\mathcal{V}(\mathcal{O}_k^j) = \Phi^*(k, \mathcal{O}_k^j) \mathcal{V}(k) + \delta_{nd}^{\mathcal{V}}(\mathcal{O}_k^j)$$

C_a^b = coordinate transformation that pre-multiplies a vector in frame a to transform it into frame b . If A = a vector in a , $B = C_a^b A$ = the vector represented in frame b .

$$\delta_r(\mathcal{O}_k^j) = C_{\mathcal{O}_k^j}^{\mathbb{B}_k}$$

$\omega_{\mathcal{O}_k^j \mathbb{B}_k}$ = angular rate of frame \mathbb{B}_k with respect to frame $\mathcal{O}_k^j = -\delta_\omega(\mathcal{O}_k^j)$

$\alpha_{nd}(\mathcal{O}_k^j)$ = time derivative of the spatial velocity $\mathcal{V}(\mathcal{O}_k^j)$ of the \mathcal{O}_k^j node with respect to the \mathcal{O}_k^j frame. The resulting spatial acceleration being transformed back into the body frame \mathbb{B}_k

$$\begin{aligned}\text{The linear part of } \alpha_{nd}(\mathcal{O}_k^j) &= C_{\mathcal{O}_k^j}^{\mathbb{B}_k} \frac{d}{dt} [C_{\mathbb{B}_k}^{\mathcal{O}_k^j} v(\mathcal{O}_k^j)] = \delta_r(\mathcal{O}_k^j) \frac{d}{dt} [\delta_r^*(\mathcal{O}_k^j) v(\mathcal{O}_k^j)] \\ &= C_{\mathcal{O}_k^j}^{\mathbb{B}_k} [C_{\mathbb{B}_k}^{\mathcal{O}_k^j} [\overset{\circ}{v}(\mathcal{O}_k^j) + [\omega_{\mathcal{O}_k^j \mathbb{B}_k}]^\times v(\mathcal{O}_k^j)]] = \delta_r(\mathcal{O}_k^j) [\delta_r^*(\mathcal{O}_k^j) [\overset{\circ}{v}(\mathcal{O}_k^j) - \tilde{\delta}_\omega(\mathcal{O}_k^j) v(\mathcal{O}_k^j)]] \\ &= \overset{\circ}{v}(\mathcal{O}_k^j) + [\omega_{\mathcal{O}_k^j \mathbb{B}_k}]^\times v(\mathcal{O}_k^j) \\ &= \overset{\circ}{v}(\mathcal{O}_k^j) - \tilde{\delta}_\omega(\mathcal{O}_k^j) v(\mathcal{O}_k^j)\end{aligned}$$

Likewise, the rotational part of $\alpha_{nd}(\mathcal{O}_k^j) = \overset{\circ}{\omega}(\mathcal{O}_k^j) - \tilde{\delta}_\omega(\mathcal{O}_k^j) \omega(\mathcal{O}_k^j)$.

More generally, if $\Delta_r(\mathcal{O}_k^j) = \begin{bmatrix} \delta_r(\mathcal{O}_k^j) & 0 \\ 0 & \delta_r(\mathcal{O}_k^j) \end{bmatrix}$,

$$\begin{aligned} \alpha_{nd}(\mathcal{O}_k^j) &= \Delta_r(\mathcal{O}_k^j) \frac{d}{dt} \left[\Delta_r^*(\mathcal{O}_k^j) \mathcal{V}(\mathcal{O}_k^j) \right] \\ &= \mathring{\mathcal{V}}(\mathcal{O}_k^j) - \begin{bmatrix} \tilde{\delta}_\omega(\mathcal{O}_k^j) & 0 \\ 0 & \tilde{\delta}_\omega(\mathcal{O}_k^j) \end{bmatrix} \mathcal{V}(\mathcal{O}_k^j) \end{aligned}$$

$f_{nd}(\mathcal{O}_k^j)$ = sum of the external spatial forces on node \mathcal{O}_k^j expressed in the body frame \mathbb{B}_k

$b(\mathcal{O}_k^j)$ = the gyroscopic spatial force for the \mathcal{O}_k^j node expressed in the body frame \mathbb{B}_k

$f_{nd}^{st}(\mathcal{O}_k^j)$ = sum of the spatial forces on node \mathcal{O}_k^j arising from structural stiffness and damping forces imposed by adjacent nodes. Expressed in the body frame \mathbb{B}_k

$a_f(k, \mathcal{O}_k^j)$ = velocity-dependent Coriolis-like spatial acceleration terms expressed in the body frame \mathbb{B}_k

$u_{nd}(k) = \text{col} \left\{ u_{nd}(\mathcal{O}_k^j) \right\} \in \mathcal{R}^{6n_{nd}(k)}$ - the stacked vector of spatial displacements for all nodes on the k^{th} flexible body.

Appendix F. MSFC/JPL Launch Vehicle Model Validation

F.1 Summary

The NASA Jet Propulsion Laboratory (JPL) Dynamics and Real-Time Simulation (DARTS) Lab is developing advanced modeling capabilities to analyze the dynamics of flexible multi-body structures in challenging environments. To help verify these capabilities, Marshall Space Flight Center (MSFC) EV-41 supplied a high-fidelity analysis tool that models the complex structural interactions between a flexible launch vehicle and other bodies such as engines and slosh masses. A series of increasingly complex test problems was created and tool responses were compared in the frequency domain. The most complex test case simulated a highly asymmetric flexible structure driven by an engine with large center of gravity offsets. Transfer functions examined were body linear and angular accelerations and rates as a function of engine motion. A metric of ± 0.001 dB was established as the desired matching criterion. Most responses were well within this tolerance, and any exceptions were explained. This effort has provided a high level of confidence that the JPL tool can generate high-fidelity structural dynamics models for use in launch vehicle control systems analyses.

The dynamic models used in the majority of this work focus on FEM models developed for a structure alone; engines and slosh masses are removed and added back mathematically by the analysis tools. Both JPL and MSFC tools can utilize FEMs constructed with engines included, but verification of this approach has been identified as future work.

F.2 Introduction

The JPL DARTS Lab provides simulation support for a wide variety of complex systems such as launch vehicles, space observatories, landers, rovers and helicopters. Many of these systems have modes of operation that violate the constraints of the linear analysis tools typically used to model them. One example is the MarsHeli, which uses high-rpm counter-rotating blades to maneuver in the thin atmosphere of Mars. Proper modeling of the blade dynamics requires inclusion of geometric stiffening and modal integral terms into the FEM representation of this system. A flexible robotic arm that experiences large joint excursions is another example of a system that is challenging to model using a traditional FEM because the mass distribution of the system varies significantly during operation. In response to these challenges, the Mobility and Robotic Systems Section of the DARTS lab is developing a tool to simplify the process of including these effects into system FEMs. In addition to improving the fidelity of structural models, another goal of this effort is to improve the pipeline of models between structural dynamicists and the GN&C practitioners who use these models. In 2018, the NESC recognized the engineering value of these goals and provided funding to help develop this capability.

To help assess the DARTS analysis tool capabilities, MSFC was asked to demonstrate and document a high-fidelity airframe model that could be used to evaluate the DARTS model. The model delivered by MSFC is implemented in a linear analysis toolkit suitable for flight control system design and analysis. This toolkit now serves as a common baseline to model vehicle dynamics; the results can be compared with DARTS results to evaluate the sensitivity of system stability margins to flexible body modal integral terms and geometric stiffening effects. These effects are not usually significant for typical launch vehicles, but the models are generic and applicable to a wide range of systems, some of which may be sensitive to these effects. This report describes the design and testing approach of the airframe model implemented in this tool.

F.3 Airframe Model Context

A launch vehicle is a complex system of interconnected components, including a flexible airframe, gimbaled engines, actuation systems and inertial sensors, all of which are controlled via autopilot, navigation, and guidance software. To assess vehicle stability during flight, the performance of each of these components must be modeled and the effects of external influences such as aerodynamics, thrust and gravity must be simulated. The airframe model is the focus of this work, but its performance can be assessed only in the context of the overall system. A notional block diagram of some of the important system elements is shown in Figure F.3.1. Each block in this figure represents a subsystem that may be described by an arbitrarily complex linear or nonlinear model, and the system may be modeled in either the time or frequency domain. This figure illustrates the system in an open-loop sense; in general, the outputs of the inertial sensors will be fed back into the blocks representing the Navigation, Guidance and Autopilot functions. The airframe model used in this study must support the architecture described in the block diagram. One of the most common areas of miscommunication between disciplines is the interface between the actuation system(s) and the airframe. This interface looks simple on a block diagram but actually involves dynamically coupling three different mechanical elements (engine, actuator and airframe) and the joints between them. Modeling this interface correctly is at the heart of this effort and will be discussed in more detail below.

The MSFC airframe model used for this task is derived in nonlinear form from first principles. The inputs and outputs of the model are compatible with the block diagram of Figure F.3.1. To support linear analysis tools commonly used for autopilot design, the airframe model is linearized, written in second-order polynomial matrix form and then transformed to a first-order state-space implementation. In addition, the model can be included in a time domain simulation for comparison with a full nonlinear implementation. Using this approach, the linear model can be verified against a time domain simulation using step or impulse responses. If the nonlinear tool is capable of simulating the assumptions used in the linear analysis, the frequency response of the time domain simulation can be derived via frequency analysis of a digital impulse excitation. This approach provides a cross-verification of the linear analysis tools and the linearized response of the time domain simulation.

F.4 Airframe Modeling Assumptions

Across the aerospace industry, there are two common approaches to describe complex multi-body dynamical systems. The first uses Lagrangian mechanics to develop the equations of motion based on the kinetic and potential energy of the system. The second approach uses Newtonian mechanics to model each body of the system and the interactions between these bodies. Each approach may lead to a preferred FEM description for the component bodies.

For a launch vehicle, the simplest system might be made up of two primary bodies: an airframe and an engine, each of which may be rigid or flexible. The Newtonian approach is naturally described with a FEM for each body in the system and a description of the interactions between these bodies. This type of FEM is known as an EOTM or “Reduced Body” model; the vehicle FEM is created without including the engine moving masses. (Note that the FEM must still provide proper interfaces so that NG&C models can reconnect the engine to the vehicle model.) A rocket stage with a liquid engine and a low-friction gimbal bearing is an example of a system that might be most readily modeled using this technique. The Lagrangian approach might more naturally be implemented with a FEM that includes both vehicle and engine. The engine is attached to the vehicle through a perfectly rigid gimbal joint or with an element that simulates the compliance of an actuator or joint. This type of FEM is known as an EITM or “Integrated Body” model. The energy due to the additional engine degrees of freedom is then added into the Lagrangian computation. A booster with a flex-seal bearing might be most readily modeled using this approach, since the motor and engine are essentially bonded together with the flex-seal bearing, which acts as a distributed spring. Because one objective of the NESC Flex Modeling project is to improve the model flow between structural dynamicists and the users of these models, the airframe model used in this project should accept either

Typical system model for a system such as SLS.

NOTE: This is not an official SLS model or analysis tool. It represents a tool I have used to design, analyze and simulate other flexible vehicles in the past. Previous implementations have been validated on other programs.

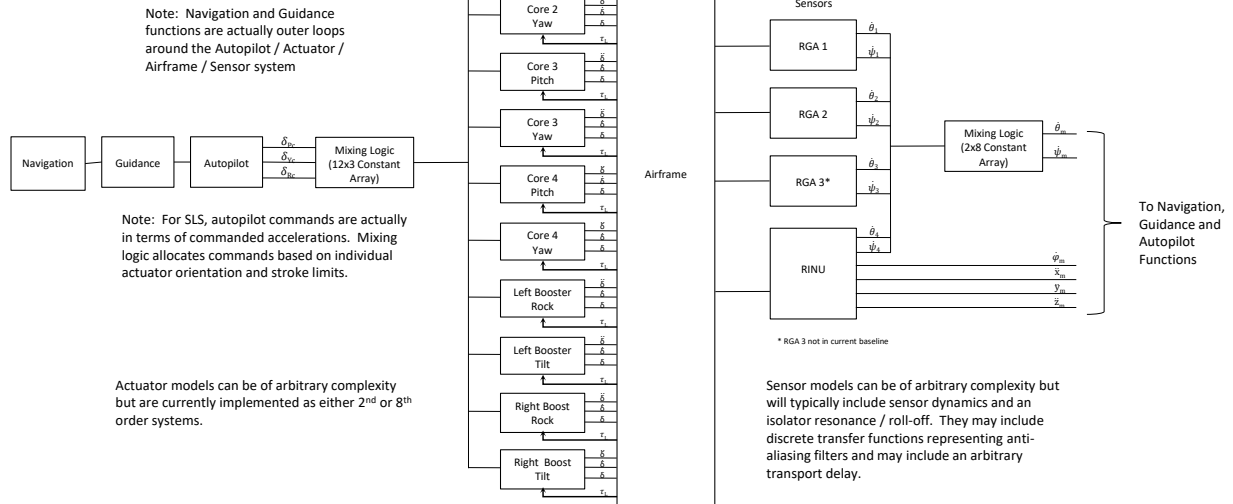


Figure F.3.1. Notional Flight Vehicle Block Diagram

EOTM or EITM FEMs. Since a valid representation of the system can be achieved with either modeling approach, another objective is to clarify the impacts of the airframe structural modeling assumptions on the other subsystem models.

To illustrate how the airframe modeling approach interacts with other subsystem models, consider Figure F.4.1, which is a simplification of Figure F.3.1 and illustrates a single channel of the coupled airframe/actuation system with a single engine. The quantity τ_{Act} is the torque applied by the actuator about the gimbal, and the load torque is defined as the torque applied by the airframe on the engine about the gimbal. The load torque expression is simply derived as an implementation of the engine moment equation $I_{gim} \ddot{\delta} = \tau_{Act} + \tau_{Load}$, where δ is the angle between engine and airframe. Consider the situation where $\tau_{Act} = 0$. In this case, the actuator will not inhibit engine motion about the gimbal and the engine will flop around a frictionless joint under the influence of any disturbance. If the actuator load torque loop is broken, $\tau_{Act} = 0$ implies that the engine angular acceleration, angular rate, and angle are zero (assuming the system is initially in a relaxed condition); this is equivalent to a perfectly rigid gimbal joint.

Now consider how this engine/airframe system will respond when the airframe is modeled using an EOTM versus EITM model. The EOTM airframe model is created without the inertial effects of an engine—the structural dynamics response of this type of model is (almost) equivalent to the response of an engine/airframe system connected with a frictionless gimbal joint. With the actuator load torque feedback path closed, the system should respond to an external disturbance with a modal response that reflects the frequencies contained in the EOTM modal data set. With the load torque loop open, the gimbal joint is rigidized and the EOTM airframe model must couple the engine into the airframe mode. In this case, the system response frequencies should shift down to reflect the response of a coupled system. This behavior is demonstrated in Figure F.4.2. The plot with the smaller low-frequency gain reflects the torque input response obtained with the load torque feedback path closed (overlaid responses from both tools). The modal frequencies displayed in the plot reflect those in the EOTM modal data set. The plot with the higher low-frequency gain

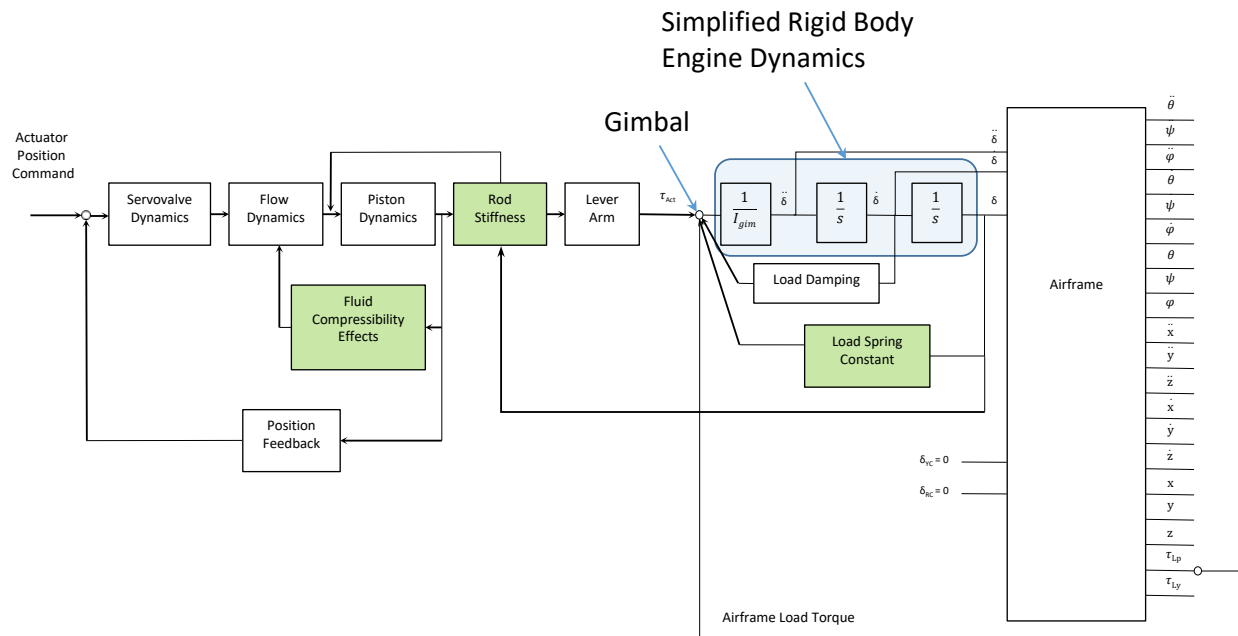


Figure F.4.1. Planar Coupled Actuator/Airframe System

reflects the response of the system with the load torque feedback broken. The mode frequencies in this case are clearly lower in frequency since the math model has coupled the engine into the system modal response as would be expected with a rigid gimbal joint. (Note that the DC gain of the second response was adjusted to more clearly illustrate the modal response.)

For an EITM airframe model implementation, the situation is reversed. If the actuator load torque loop is broken, the gimbal joint is again rigidized, matching the conditions used to generate the system modal data. When subjected to a disturbance, the system should respond at the frequencies contained in the modal data set. If the actuator load torque loop is closed, the EITM airframe model must remove the effects of the engine from the coupled modal data. If the model is implemented correctly, a disturbance in this condition will cause the airframe to respond at frequencies higher than those in the modal data set. The frequencies should, in fact, match those from an EOTM FEM model.

To use an electrical analogy, the actuator acts as an ideal voltage source when the load torque loop is broken; it will force a given voltage (actuator position) regardless of the current (torque) required. This configuration is often a good approximation for a system such as a launch vehicle, where the actuator has enough torque capability for the limited bandwidth required for control. When the load torque loop is closed, the actuator acts as an ideal current source—it will supply the commanded current (torque) regardless of the load voltage (engine position). This is not a common situation for a launch vehicle since it would imply some kind of clutch connection between engine and airframe. To fully characterize the plant model, however, it is best to verify performance for both high and low impedance inputs. The value of testing with torque inputs was demonstrated in this effort, as will be described in a later section.

Note that in some cases a very accurate representation of the load path between the engine and flexible airframe is required; see [31] for an interesting example. The principles above still apply but the engine loads are distributed to the airframe FEM using multiple grid points to more accurately capture local compliance effects.

The preceding discussion illustrates the importance of coordinating airframe FEM modeling assumptions

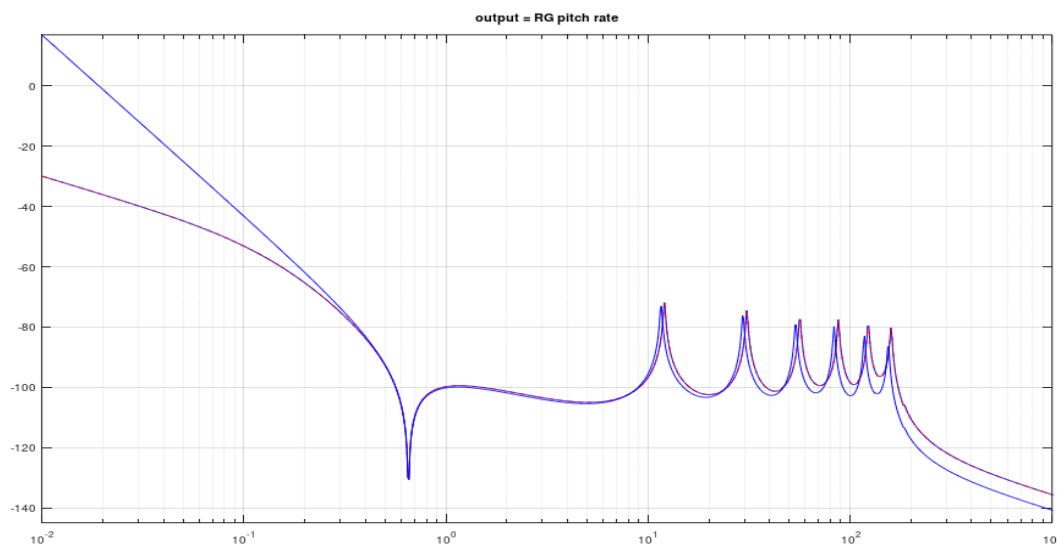


Figure F.4.2. Modal Response Frequencies Shifted as a Result of Engine Coupling

between those developing and those using the model data. Figure F.4.1 also illustrates some additional modeling details that must be coordinated. The boxes shaded in green indicate spring elements embedded in the actuator model. The specific location of these elements will vary based on the type of actuator employed but the figure is generally representative of a wide range of effectors. The “Load Spring Constant” block is representative of a spring that might represent a flex-seal nozzle connection, nozzle boot or engine duct. This effect may be included in an external actuator model as shown or may be embedded in the airframe model using static properties derived from a FEM. If modeled in the airframe model, this effect should be included in the Airframe Load Torque and removed from the actuator block diagram. In this case, an approach to evaluating variations in this spring constant should be coordinated with the appropriate SMEs.

In an EITM FEM, the “Rod Stiffness” and “Fluid Compressibility” elements are often used to calculate an actuator compliance that is used to restrain the engines. A difficulty in this approach is that these effects must be backed out of the actuator model used for controls analyses to avoid duplicating the effects. It may be possible to incorporate dynamic actuator models into the FEM, but these actuator models can require very high frequency elements (multiple kHz) to capture relevant dynamics. Another effect to consider is that a steady-state engine deflection will result in a variation in system mass properties, and this may be modeled in the FEM or in the analytical model. Any of these approaches may be valid when developing the FEM, but it is clear that the assumptions used to develop the FEM must match those used to develop the models for control system design.

The preceding discussion illustrates the importance of communicating FEM modeling assumptions between structural dynamicists and control systems practitioners. The different modeling options have both been used successfully on a wide range of aerospace systems. The Titan IV launch vehicle used the EOTM approach. The Hera family of rockets used an EOTM approach, but used FEMs validated via modal tests performed with engines included. The SLS system uses both EITM and EOTM approaches, with the EITM method used for the engine masses in the the core stage FEM while the EOTM method is used for booster engine masses and core stage slosh. The Saturn program used EITM models while the Space Transportation System (STS) program used an EOTM implementation for both slosh and engine masses. To accommodate the various modeling approaches, the actuator/airframe model defined in this project supports a wide range of modeling assumptions. For the current project phase, an EOTM assumption was primarily used

for comparison with DARTS models. The airframe model used by MSFC for this task is documented in Reference [23]. The model implemented in the DARTS toolkit is derived using a relatively new approach using Spatial Operator Algebra; this approach is described in Reference [32], a textbook written by one of the co-authors. Reference [18] documents a comparison of the rigid body equations developed by JPL and those described in Reference [23].

F.5 Tool Development and Implementation

To expedite airframe model testing, a simple analysis framework was written using Octave, a scientific programming language that is largely compatible with Matlab (Reference [33].) Octave is free software licensed under the GNU General Public License, and was used to avoid tying up SLS Matlab licenses. For the convenience of those who do not have access to Octave, Octave-specific language extensions were avoided so the software will execute in a Matlab environment as well. One complication is that Octave does not yet implement all of the functionality of the Matlab Control System Toolbox, so some utility commands were manually coded or avoided. Various versions of the models described above have been used for control system design and analysis on programs ranging from multi-million pound launch vehicles to sub one-pound flying inverted pendulums. In all cases, the analysis flow and implementation is similar.

1. Define test case parameters in an input data file or script. For this work, a script was created for each analysis test case describing the desired input channel (pitch, yaw or roll); desired input signal (e.g. prescribed engine angular acceleration or torque, body-frame force or moment at the gimbal); and the desired output response (e.g. measured rate or acceleration at the sensor location, load torque).
2. Define airframe parameters at the desired flight condition. Such data includes aerodynamic parameters, mass properties, modal parameters, flight conditions, and slosh parameters. In many cases, this step is performed as a scheduled task during a 6DoF simulation run. For this effort, the airframe mass properties and modal data are read from an HDF5 file generated from a FEM and provided by JPL. Other parameters such as engine mass properties are defined in the input data file.
3. Initialize airframe models with appropriate initial conditions, such as initial linear and angular accelerations and velocities of all bodies, including the core stage, engines and slosh masses. In a typical analysis flow, this would be performed as part of the previous step. For testing, we need to calculate initial conditions which are consistent with the desired test case. As an example, for a prescribed engine input, we assume a rigid body with locked engine gimbal and compute initial body (and engine) linear and angular accelerations based on thrust and cg offsets. (These assumptions are used only to calculate initial conditions.) In the case of a torque input, the gimbal joint is assumed frictionless and the initial linear and angular accelerations of both body and engine must be calculated. The current process is rather awkward and is implemented in the scripts "calc_initial_conditions_prescribed_input.m", "calc_initial_conditions_torque_input.m", etc. Calculation of simulation variables based on these initial conditions is performed in "calculate parameters.m".
4. Populate the airframe math model second-order-polynomial matrices using the script "airframe.m". The coefficients of these matrices are defined in Appendix A of Reference [23].
5. Convert the airframe model to state-space form and connect with linearized state-space models of the pitch, yaw and roll actuators using the script "linear_ActAF.m". The "connect" command is used to couple the airframe load torque outputs into the actuator model if required.
6. Create the desired frequency response data using the Matlab/Octave "bode" command. Previous generations of Matlab (and Octave) required alternate algorithms for calculating transfer function responses in certain cases, but the current versions seem quite robust. Typically it is quite obvious if the algorithm runs into numerical difficulty when calculating frequency response data. If this occurs, it is fairly straightforward to implement an alternate algorithm.

Recall that the DARTS project objectives include reducing the turnaround time for delivery of data products and demonstrating how the model pipeline between structural dynamicists and controls engineers can be improved. In the long term, this goal will be achieved by training controls engineers to generate and manipulate FEMs and resulting modal data using the DARTS toolset. While this long-term goal is outside the scope of the current project phase, the team did develop and demonstrate a process for rapidly exchanging and documenting FEM modal data and models. Initial modal data exchange was via NASTRAN .f06 data files. This approach was cumbersome and required processing with external Python scripts to extract the required information. This approach also resulted in truncation of the modal data due to wordlength constraints; this truncation was clearly visible in tool comparisons. To improve the data exchange process, the HDF5 file protocol was utilized. The Hierarchical Data Format version 5 (HDF5) is an open-source file format that supports large, complex, heterogeneous data such as a complete FEM, output data sets, and documentation or other metadata describing each of the data groups. Both Matlab and Octave have the capability to read HDF5 files. Octave implements a simpler method for accessing HDF5 files that is not compatible with the Matlab approach; in this case, separate Matlab and Octave subroutines were created.

Once the HDF5 data transfer approach was implemented, the process of exchanging system models was greatly simplified. During multiple model debugging sessions the team was able to exchange and evaluate updated models in a matter of minutes. In addition to speeding the data exchange process, the HDF5 file format serves to consolidate a variety of model data and documentation for improved configuration control. A single file can contain a FEM, metadata, documentation, and supporting data such as a state-space description of the simulated system. (These state-space models were used for tool comparison purposes.) The format also allows data to be retained in full precision, eliminating earlier difficulties experienced with truncation of the modal data as contained in NASTRAN .f06 files. All the software described above is included with the delivery of this report.

F.6 Test Cases

Simple Test Cases: A series of test cases of increasing complexity was coordinated with JPL. The first test case was a simple uniform, flexible beam. A NASTRAN FEM was created by JPL and a standard .f06 output file was provided to MSFC. This model was used to ensure MSFC and DARTS/DShell tools use consistent units and sign conventions. After aligning some basic modeling assumptions, an exact match was obtained for both step and frequency responses. The next step in the testing process added a gimballed engine to the simple flexible beam. Yet another level of complexity was added by modeling a non-uniform asymmetric beam with 24 bending modes. Test results for all of these models showed excellent correlation and were documented in a series of status reports. The simplified test cases will not be discussed further, since the complex test cases were much more challenging and will be discussed in detail.

Complex Test Cases: The complex system configuration used for detailed testing is summarized in Figure F.6.1. The airframe model was created by JPL with an intentionally asymmetric FEM. For model comparison, 24 flexible modes were retained with a maximum frequency of 30 Hz. One percent damping was assumed for all bending modes. The airframe mass was 1695 kg and a relatively massive engine of 50 kg was used. The engine axial CG was located 1m aft of the gimbal and relatively large y and z CG offsets of 10 cm and 20 cm were modeled. To exaggerate coupling between channels, large cross products of inertia were specified. The body frame origin was arbitrarily selected as the CG of the missile body (neither MSFC nor JPL tools require this assumption.) The vectors from the origin to the sensor and gimbal locations are defined in Figure F.6.1. Both gimbal and sensor are offset from the missile centerline to provide an additional degree of cross-axis coupling. A thrust level of 1000 N and zero initial body rates were used to define a reasonable set of initial conditions. To provide better resolution of modal responses, the HDF5 file format was used to transfer FEM data for greater numerical resolution.

As described earlier in the discussion of Figure F.4.1, it is important to verify the operation of the airframe model with both prescribed engine motion (load torque feedback open) and commanded torque inputs

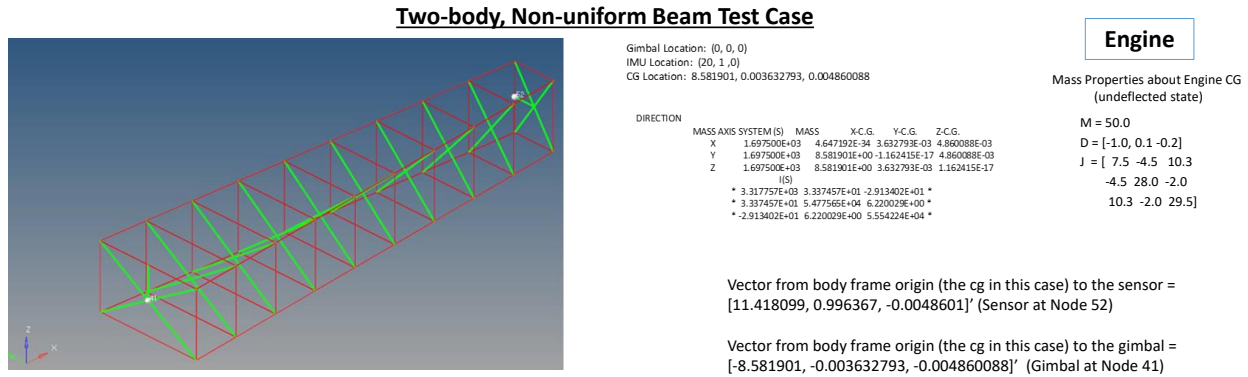


Figure F.6.1. Non-Uniform Beam Test Case Definition

(load torque feedback closed.) To focus on the critical features of the coupled engine/airframe system, the configuration used to evaluate the models is illustrated in Figure F.6.2. The system is evaluated as a single-input-multiple-output system. For pitch test cases, the yaw command is set to zero and vice versa. The load torque feedback connections are turned on or off using the gains K_{LTp} and K_{LTy} .

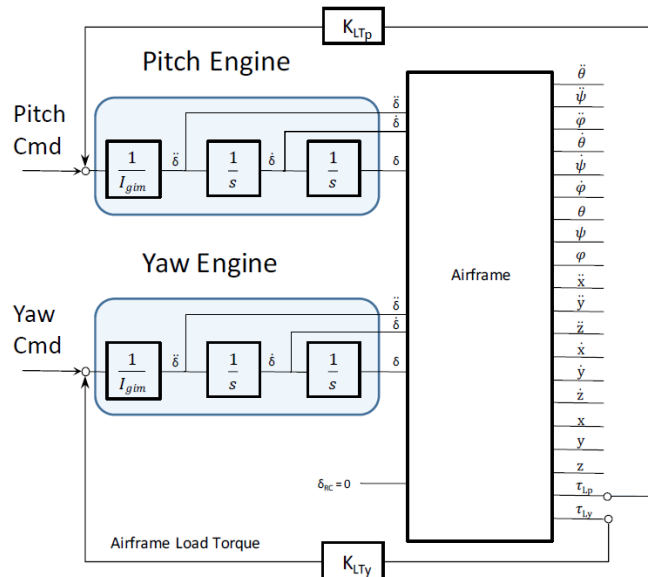


Figure F.6.2. Test Case Configuration

Table F1 illustrates the test case matrix used to compare MSFC and DARTS/DShell results; all of these test cases refer to the Complex Test Case described above. Eight test cases were defined and multiple output variables were examined for each case. Test Case 1 uses a prescribed pitch actuator acceleration input. The primary outputs examined were the Z acceleration at the sensor location, as well as measured pitch, yaw, and roll rates. "Prescribed Motion" is obtained for both channels by opening the "Load Torque Feedback" loop as shown in Figure F.6.2 by setting $K_{LTp} = 0$ and $K_{LTy} = 0$. Case 2 is similar, but the input is specified as a pitch gimbal torque. The only difference between this case and Test Case 1 was that the Load Torque

Feedback path for both channels was closed and thus the feedback gains set to 1 for both channels. Test Cases 3 and 4 were identical to Cases 1 and 2 except they use a yaw engine input and the primary evaluation outputs focused on yaw parameters instead of pitch. Test Cases 5–8 were added late in the program and were designed to help isolate modeling discrepancies noted in Cases 2 and 4. Cases 2 and 4 modeled both pitch and yaw gimbal joints as frictionless; it was observed that this condition magnifies any minor model differences and is very sensitive to initial conditions. Cases 5 and 6 were duplicates of Cases 2 and 4, except the off-axis channel was modeled with a prescribed input. Cases 7 and 8 are again duplicates of Cases 2 and 4 except that thrust was set to zero. Thus, both gimbals operate as frictionless but the initial conditions due to thrust and engine/airframe CG offsets have been eliminated. To more logically isolate the minor modeling differences between JPL and MSFC models, the test cases will be examined in order of complexity.

Test Case	Input	K_{LT_p}	K_{LT_y}	Thrust (N)
1	Pitch ($\ddot{\delta}_{p_c}$)	0	0	1000
2	Pitch (τ_{p_c})	1	1	1000
3	Yaw ($\ddot{\delta}_{y_c}$)	0	0	1000
4	Yaw (τ_{y_c})	1	1	1000
5	Pitch (τ_{p_c})	1	0	1000
6	Yaw (τ_{y_c})	0	1	1000
7	Pitch (τ_{p_c})	1	1	0
8	Yaw (τ_{y_c})	1	1	0

Table F1. Evaluation Test Case Definitions

Test Cases 7 and 8: The Complex Test Case was defined with exaggerated missile body and engine CG offsets and large engine cross products of inertia. If the system is initially relaxed with zero engine deflections, both body and engine will experience initial conditions on linear and angular accelerations if external forces are applied. In early testing it was observed that these initial conditions were responsible for some of the most significant differences between JPL and MSFC results. Test Cases 7 and 8 were created to verify the inertial coupling between the engines and missile body modal motion without the complication of these initial conditions by setting the axial thrust to zero. These test cases also verify the proper modeling of the frictionless gimbal joints. Key findings will be discussed here, but the full set of acceleration and rate responses can be found in Appendices F.15 and F.16.

Figure F.6.3 is an overlay of responses generated by JPL and MSFC for Case 8. MSFC results are plotted with a solid blue line and JPL results with a dotted red line. The responses match closely enough that the plot appears as a single purple trace.

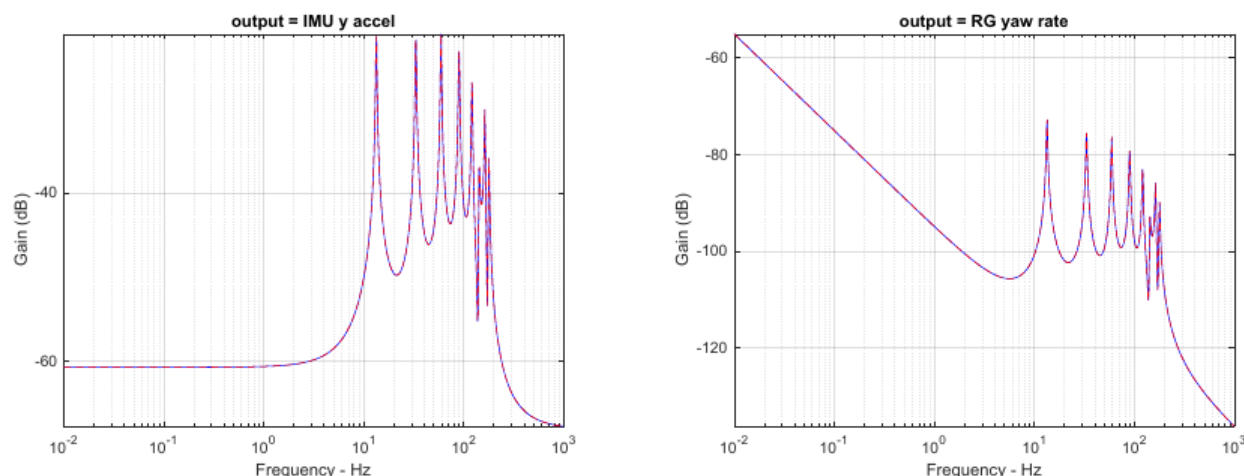


Figure F.6.3. Overlay of DARTS and MSFC Responses

Figure F.6.4 is a plot of the difference between the JPL and MSFC results presented in Figure F.6.3; note the y axis limits of $\pm 2 \times 10^{-6}$ dB. This quality of matching is common across all the Case 7 and Case 8 responses. While this match is far better than the established objective of ± 0.001 dB, some additional work was performed to establish what limited this performance.

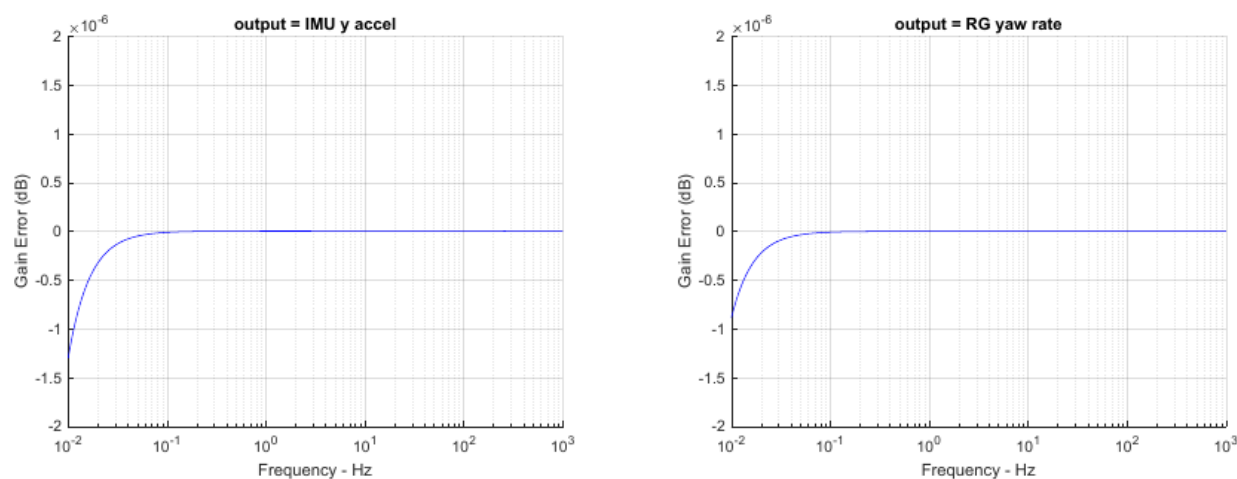


Figure F.6.4. DARTS and MSFC Response Differences; Zoomed Y Axis (Matlab)

Given that most of the complexity in the airframe equations (see Appendix A of Reference [23]) is in the implementation of the bending models, it was somewhat surprising to see a better match at bending frequencies than at rigid body frequencies. This finding is illustrated in Figure F.6.5, where the match is actually better than $\pm 5 \times 10^{-9}$ dB for the sensed IMU acceleration and $\pm 5 \times 10^{-12}$ dB for the sensed body rates over the frequency range spanning all of the bending modes.

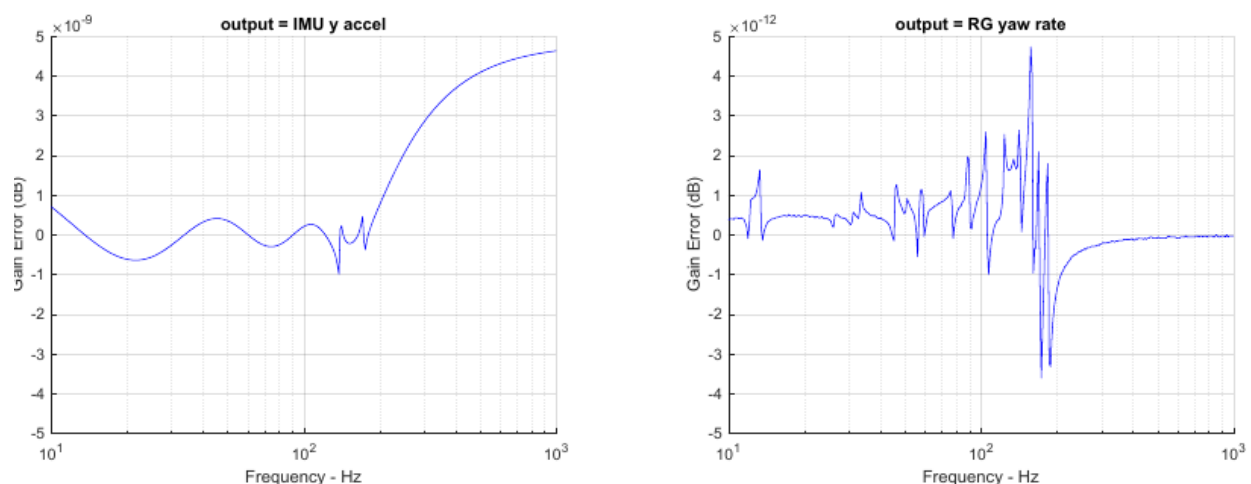


Figure F.6.5. DARTS and MSFC Response Differences; Extreme Zoom of Y Axis

Investigating further, Figure F.6.6 illustrates the sensed gyro yaw rate response as calculated by Matlab (right) versus Octave (left.) It is clear that for this test configuration, the JPL and MSFC implementations match to within the numerical precision of the Matlab and Octave algorithms for the rate responses over the frequency range of the bending data. The sensed accelerations match to a slightly lower resolution of $\pm 5 \times 10^{-9}$ dB. It should be noted that the reported accuracy of the JPL finite difference approach to calculating the system matrices is only expected to result in matches on the order of $\pm 5 \times 10^{-5}$ dB.

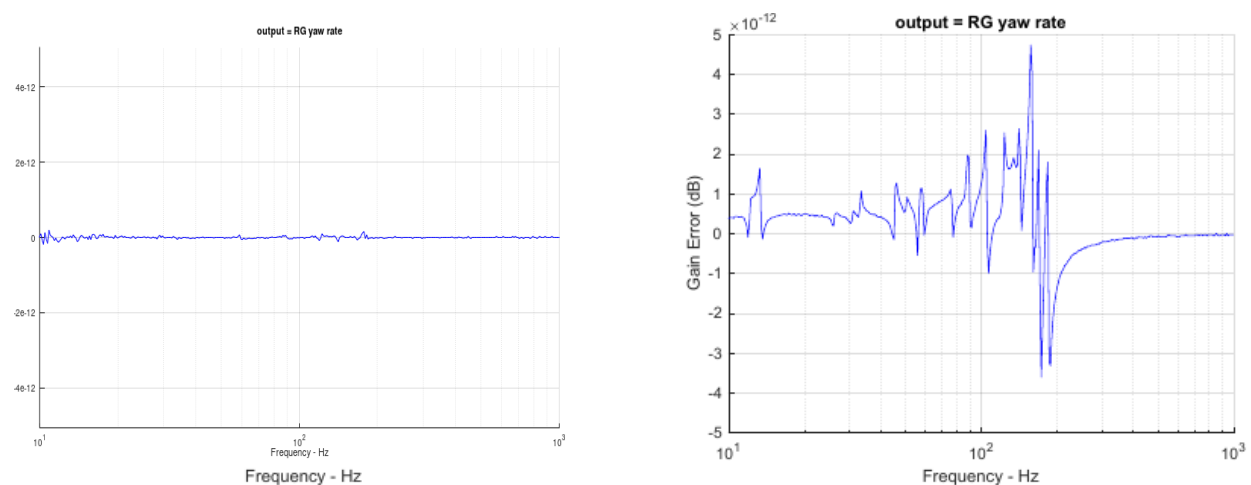


Figure F.6.6. Octave (left) versus Matlab (right) Response Errors, 10-1000 Hz

Finally, the low frequency response differences of Figure F.6.4 were investigated. Figure F.6.4 was created with Matlab and shows a smooth increase in response errors below 0.1 Hz. This plot was recreated using Octave as shown in Figure F.6.7. The effects of numerical precision are obvious in the response errors below 0.1 Hz. It is not clear whether the responses of Figure F.6.4 or Figure F.6.7 are more accurate, but it is apparent that the calculations become numerically challenging below 0.1 Hz. We conclude that Cases 7 and 8 match to within the reliable numerical accuracy of the linear analysis libraries contained in Matlab and Octave. (Note that errors of this magnitude are insignificant for practical control system design purposes.)

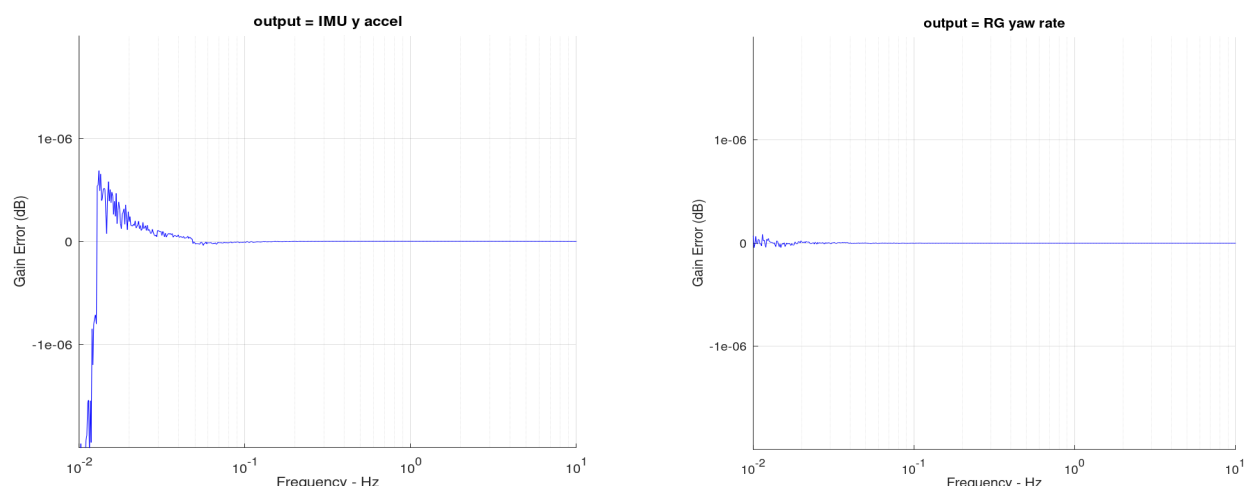


Figure F.6.7. DARTS and MSFC Response Differences; Zoomed Y Axis (Octave)

Test Cases 1 and 3: Test Cases 1 and 3 represent the next step up in complexity. These are prescribed engine acceleration cases in the Pitch and Yaw channels respectively. An extensive set of responses for each of these cases is provided in the Appendix. This section focuses on the primary responses for each case. Test Case 1 is driven by a prescribed pitch engine acceleration, thus the primary responses are the missile body Z acceleration and the body angular rate about the Y axis. There are noticeable responses in the yaw and roll body rates due to CG offsets and engine cross products of inertia. Likewise, the primary responses for the Yaw Test Case 3 are the body Y axis acceleration and angular rate about the Y axis. There are again significant rates in the other channels due to inertial coupling.

These cases both include a steady-state thrust of 1,000 N. The addition of thrust will introduce zeros into some of the responses where the inertial effects of engine motion cancel the forces and moments due to thrust. When comparing two different simulations, it is extremely unlikely that the responses will match in the immediate vicinity of a response zero. Figure F.6.8 illustrates the sensed pitch rate response in the vicinity of the airframe tail-wags-dog (TWD) zero. (The TWD zero is defined as the frequency where the moment about the system cg due to engine angular acceleration about the gimbal is cancelled by the moment due to vectored engine thrust.) The plot on the right is an overlay of JPL and MSFC responses between the frequencies of 0.65253 and 0.65254 Hz; the gain in this frequency is lower than -160 dB. As the plot resolution is increased, this response should ideally go to $-\infty$ dB. Any slight numerical error will cause differences in the predicted TWD zero frequency or cause added numerical damping. The plot on the left of Figure F.6.8 shows that even though the JPL and MSFC responses match remarkably closely, the magnitude of the error at the TWD zero location peaks at approximately 0.6 dB.

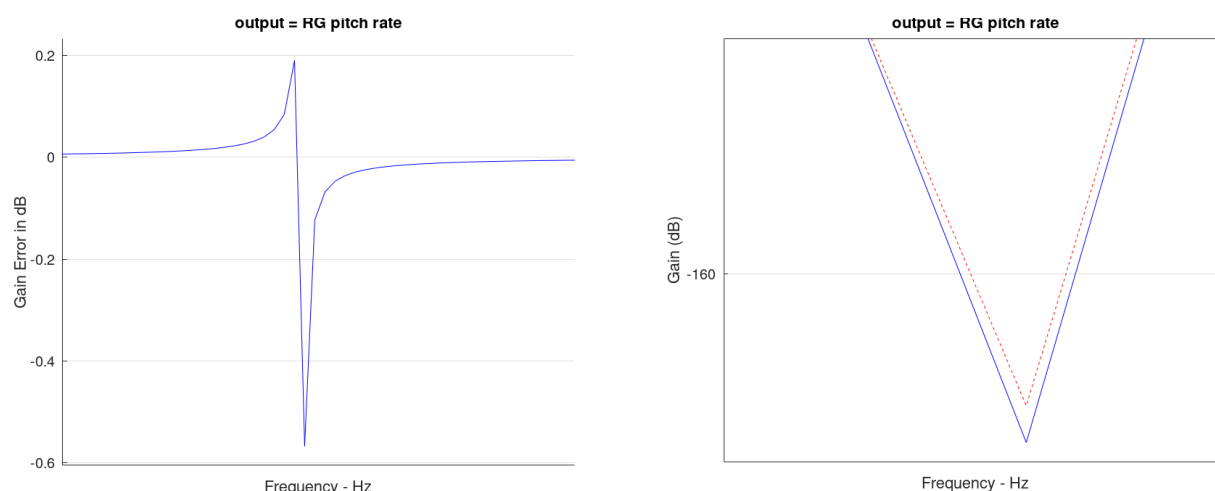


Figure F.6.8. DARTS and MSFC Responses near the Tail-Wags-Dog Airframe Zero

Another example of this effect is seen in the sensed roll rate response of Test Case 3, as shown in Figure F.6.9. In this case, the sign of the engine Z cg offset results in a series of zeros in the roll modal response. Recognizing that it is not practical to expect the same degree of response accuracy in the vicinity of response zeros, we will relax our matching objective. In the immediate vicinity of response zeros we will note any response errors but will not consider these differences a violation of our goal.

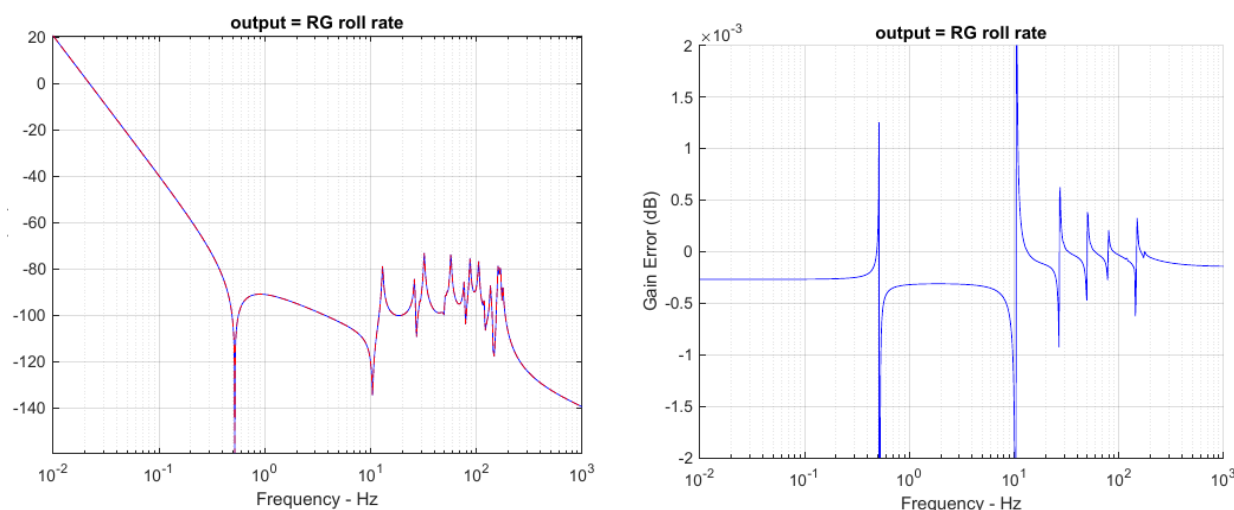


Figure F.6.9. Test Case 3 Roll Rate Response Zeros

Cases 1 and 3 introduce another interesting, somewhat unexpected observation. The match between tools at rigid body frequencies is actually more challenging than at modal frequencies. An example of this can be seen in the low-frequency response of the roll rate shown in Figure F.6.9. This effect will be described in more detail when describing Cases 2 and 4, since this effect is even more pronounced there. All of the Case 1 and Case 3 frequency responses are presented in the Appendix. All primary responses are accurate to well within the the goal of ± 0.001 dB and most were 10x better than this. In addition, the secondary responses also met the evaluation goal with the single exception of the axial acceleration response of Test Case 1.

At the lowest frequency of 0.01 Hz this case matched to within 0.01 dB. We will explore the challenges of matching at very low frequencies and relate this to numerical sensitivity to initial conditions in a following section. As a final note, this frequency range is extremely low for the small test system; for a test case representative of a system such as the SLS, we would expect this discrepancy to appear at least another decade lower in frequency.

Test Cases 5 and 6: As defined in Table F1, Test Cases 5 and 6 are single-channel torque-input cases; the off-axis channels have the load torque feedback path broken and thus the off-axis gimbal is locked. These cases are used to verify the modeling of the frictionless gimbal without the complexity of cross-axis coupling in the gimbal joint. The primary acceleration responses and all rate responses can be found in the Appendix. While the responses display the expected low frequency differences due to the closed torque feedback loops, the error response plots are almost indistinguishable from those of Cases 1 and 3 and are well within the desired tolerance of ± 0.001 dB. These results indicate that the gimbal joint is being modeled correctly for both torque and prescribed engine motion inputs.

Test Cases 2 and 4: As defined in Table F1, Test Cases 2 and 4 are torque-input cases with the load torque feedback paths closed for both pitch and yaw channels; this connection forces the joints to act as frictionless bearings to disturbances. These cases are very sensitive to initial conditions on body and engine linear and angular accelerations. The response errors for primary accelerations and both pitch and yaw rate responses are well within the desired tolerance of ± 0.001 dB while the roll rate responses are accurate to within ± 0.002 dB. The following paragraphs illustrate the reasons for the roll responses being slightly outside the desired tolerance and more generally illustrate the impact of initial conditions on low-frequency response errors.

The complete set of responses for Cases 2 and 4 can be found in the Appendix. The Case 4 errors were slightly higher so we will focus on this test case. Figure F.6.10 illustrates the secondary rate responses and response errors for this yaw-input test case. Of note are the low-frequency errors; the pitch rate error has almost reached the maximum error tolerance of ± 0.001 dB and the roll rate error has exceeded this value.

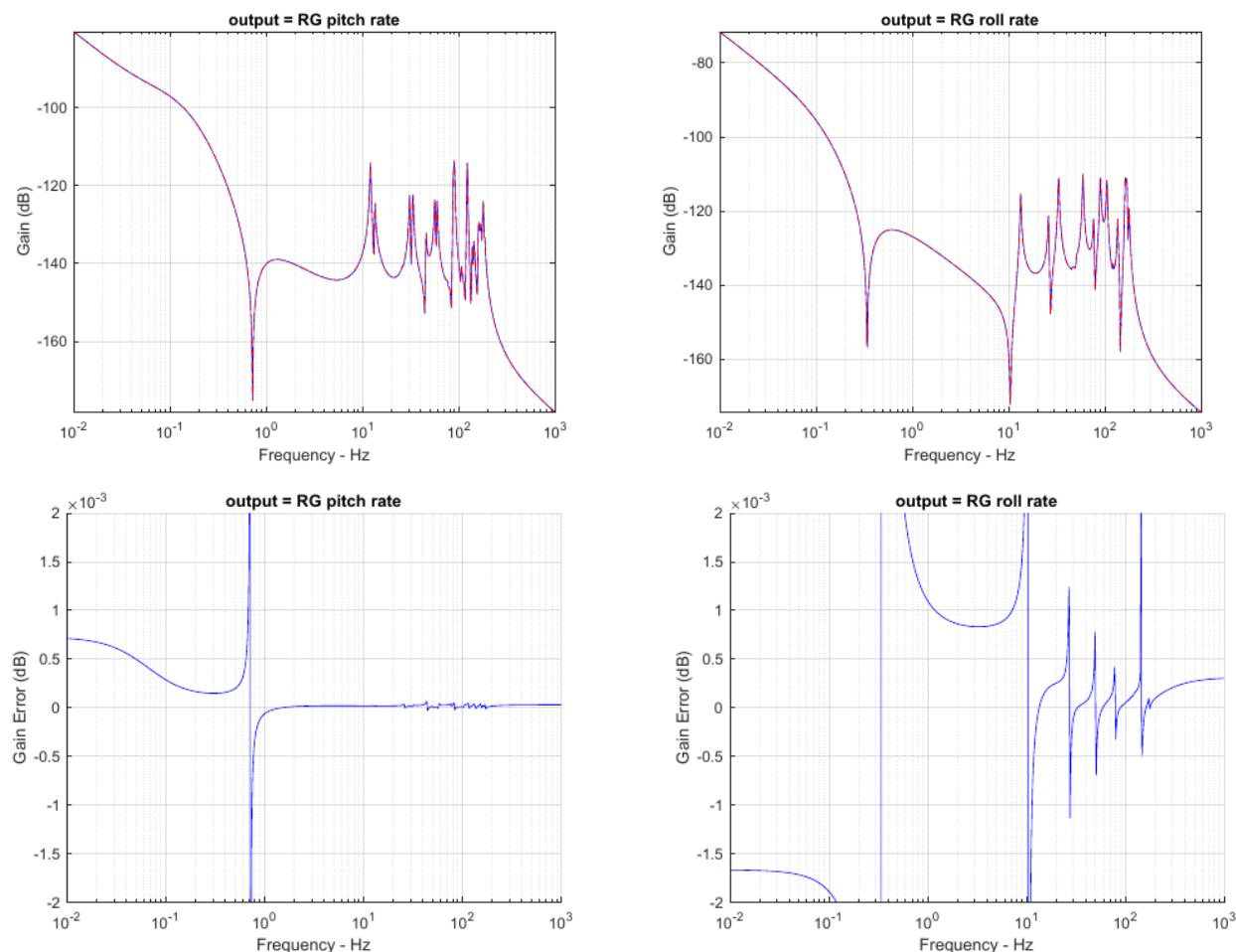


Figure F.6.10. Test Case 4 Sample Responses and Errors

Figure F.6.11 illustrates the nominal error responses in blue along with the effect of varying the initial angular accelerations of the engine about the gimbal by $\pm 0.025\%$. For this test case, $\ddot{\delta}_{p0} = 4.4 \text{ deg/s}^2$ and $\ddot{\delta}_{y0} = 2.2 \text{ deg/s}^2$ and a 0.025% variation is less than 0.0015 deg/s^2 . The response variations exhibit an almost extreme sensitivity to this minuscule variation in initial conditions, indicating that the response errors are almost chaotic at these low levels. Note that somewhere between these $\pm 0.025\%$ variations, both error plots would have a low-frequency error of zero. As another check of these responses, the results were duplicated using Octave.

Figure F.6.12 illustrates that the responses predicted by Octave look similar to the Matlab results except for the low frequency gain, which is significantly different. We have previously demonstrated that this problem becomes numerically challenging at very low frequencies for both Matlab and Octave, and the low-frequency response difference is a symptom of this. In addition, we have just seen that these test case responses are very sensitive to initial conditions. It seems reasonable to relax the objective for the roll rate response tolerance to $\pm 0.002 \text{ dB}$ for this test case at low frequencies.

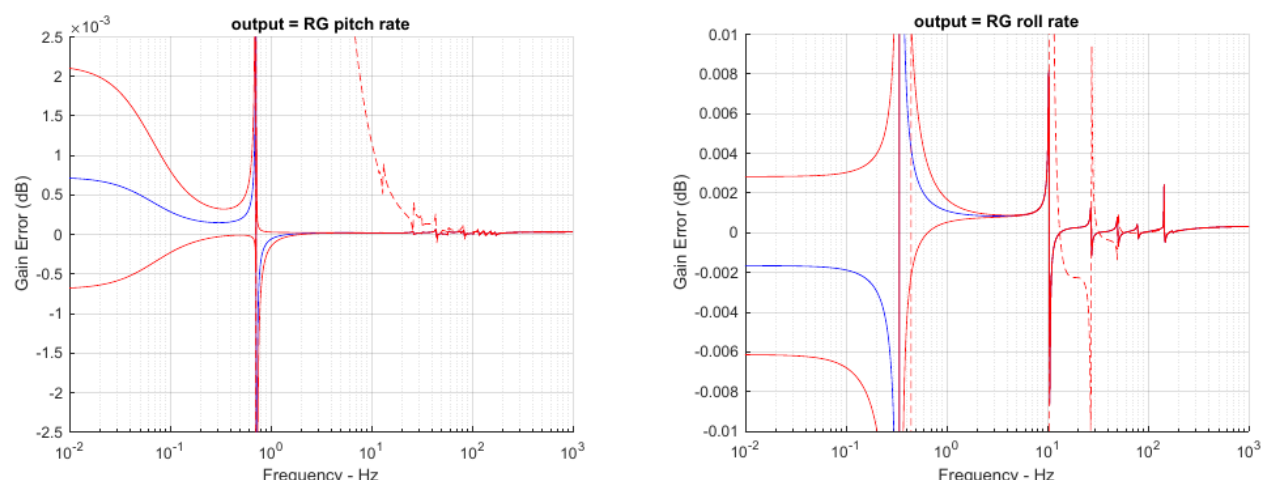


Figure F.6.11. Error Sensitivity to $\pm 0.025\%$ Variation in Initial Engine Accelerations, Matlab

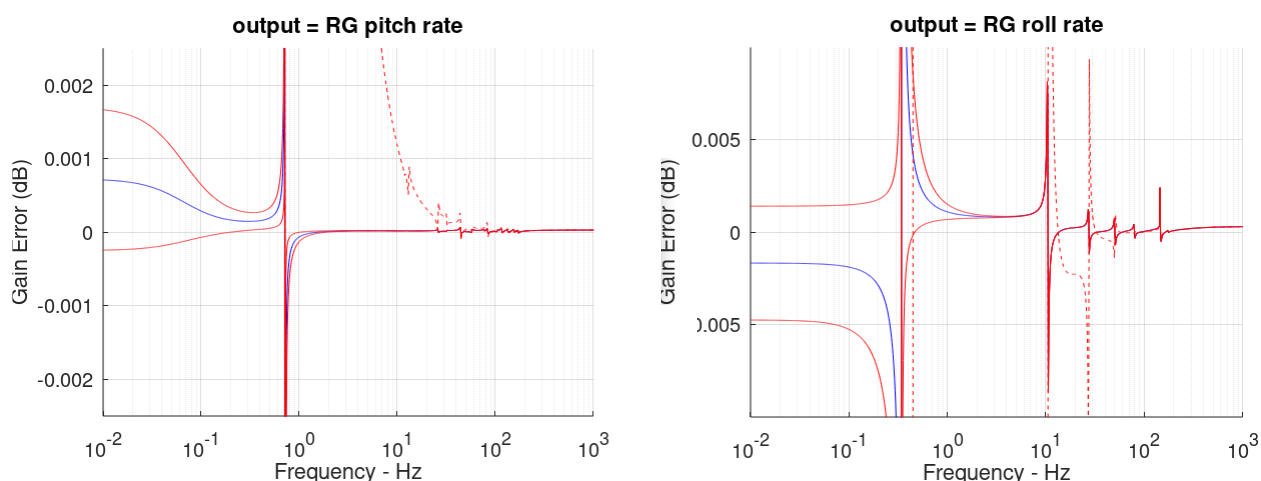


Figure F.6.12. Error Sensitivity to $\pm 0.025\%$ Variation in Initial Engine Accelerations, Octave
(Note the difference in low frequency response compared with Figure F.6.11)

A final observation on the effect of initial conditions on low-frequency responses is illustrated in Figure F.6.13. The solid lines represent the nominal response and the responses with $\pm 0.025\%$ variations in initial engine angular accelerations. The dashed red lines indicate the response if the initial engine angular accelerations are neglected (set to zero.) It is apparent that modeling these initial conditions is essential to meet the matching objectives even for the primary responses; without these effects the response error is approximately three times higher than the desired matching criterion. The lower two plots of Figure F.6.13 illustrate how sensitive the off-axis responses are to these effects; if these initial conditions are neglected the pitch rate response error is greater than 5 dB and the roll rate response error is almost 8 dB at low frequencies. An important conclusion is that if accurate prediction of very low-frequency and/or off-axis system responses is important, then initial conditions on linear and angular accelerations must be modeled.

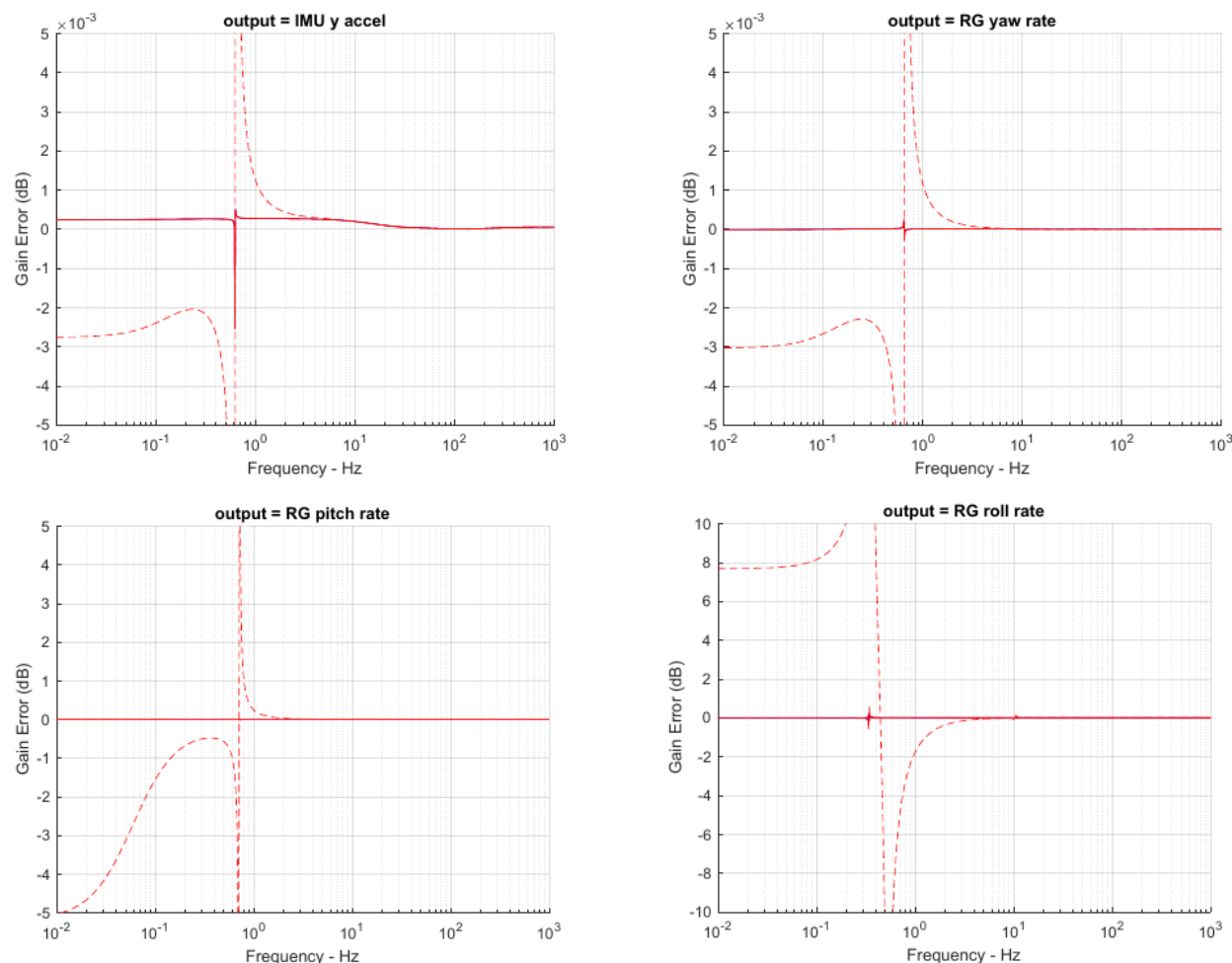


Figure F.6.13. Effect of Initial Engine Accelerations on Response Errors

Summary of Test Cases: Table F1 describes the test cases used to compare the JPL DARTS airframe model against an independently derived MSFC model. The metric used for comparison is the error in transfer function magnitudes for the responses important for control system design. An extremely aggressive goal of ± 0.001 dB was established to allow a reliable assessment of the effects of phenomena such as modal integrals and geometric stiffening, which are expected to be small for our familiar evaluation system. The primary responses for all test cases met the matching goal and, with one minor exception, all secondary responses met this goal as well. Note that the matching goal was relaxed in the immediate vicinity of response zeros and the goal for Case 2 and 4 low frequency roll rate was relaxed to ± 0.002 dB due to the extreme sensitivity to initial conditions. Test Case 8 illustrates that without the initial conditions caused by axial thrust and CG offsets, the responses matched to better than $\pm 1 \times 10^{-6}$ dB at low frequencies and this performance was limited by numerical precision of the Matlab and Octave algorithms. At modal frequencies, the match was better than $\pm 5 \times 10^{-9}$ dB.

E.7 Summary and Conclusions

A goal of the DARTS program is to demonstrate the effects of different FEM modeling assumptions on the modal data delivered for autopilot design and to develop a framework to communicate these assumptions. In this assessment, the approach to demonstrate this objective was to develop a high-fidelity tool that can be used to verify the response predictions of the DARTS toolkit for a complex sample problem. The airframe model incorporated into the JPL DARTS toolkit was evaluated against an independently derived MSFC model, which has legacy to several programs. To provide an intuitive framework for comparison, an evaluation system was created to be representative of a flexible launch vehicle driven by a gimballed engine. A comprehensive set of test cases was established, and important responses of the two models were compared. These test cases included prescribed engine inputs and torque-input cases as required to support different FEM assumptions. An aggressive goal of ± 0.001 dB was established as a matching criterion for the transfer function magnitude error. This goal was met for all primary responses with one exception: the roll rate response for Cases 2 and 4 is extremely sensitive to initial conditions, and the low frequency goal was relaxed to ± 0.002 dB for these cases. In addition, all secondary responses met the matching goal with the exception of the low-frequency axial response for one test case that matched to ± 0.01 dB. This level of fidelity is not required to support control system design, but does verify the objective of developing a high-fidelity dynamic airframe/engine model.

F.8 Test Case Definitions

Figure F.8.1 documents the set of test cases selected during the first phase of this task. All cases use the system definition known as the “Complex Test Case” described previously. Test Cases 1 and 3 represent prescribed motion with Pitch and Yaw channel inputs respectively. Test Cases 2 and 4 represent torque

Phase 1 Test Cases:

Test Case 1a: Tapered beam, Thrust = 1000, Load Torque = 0 Input: Pitch torque about the gimbal Output: Z acceleration at the sensor	Test Case 1b: Tapered beam, Thrust = 1000, Load Torque = 0 Input: Pitch torque about the gimbal Output: Pitch rate at the sensor	Test Case 1c: Tapered beam, Thrust = 1000, Load Torque = 0 Input: Pitch torque about the gimbal Output: Yaw rate at the sensor	Test Case 1d: Tapered beam, Thrust = 1000, Load Torque = 0 Input: Pitch torque about the gimbal Output: Roll rate at the sensor
Test Case 2a: Tapered beam, Thrust = 1000, Load Torque = 1 Input: Pitch torque about the gimbal Output: Z acceleration at the sensor	Test Case 2b: Tapered beam, Thrust = 1000, Load Torque = 1 Input: Pitch torque about the gimbal Output: Pitch rate at the sensor	Test Case 2c: Tapered beam, Thrust = 1000, Load Torque = 1 Input: Pitch torque about the gimbal Output: Yaw rate at the sensor	Test Case 2d: Tapered beam, Thrust = 1000, Load Torque = 1 Input: Pitch torque about the gimbal Output: Roll rate at the sensor
Test Case 3a: Tapered beam, Thrust = 1000, Load Torque = 0 Input: Yaw torque about the gimbal Output: Y acceleration at the sensor	Test Case 3b: Tapered beam, Thrust = 1000, Load Torque = 0 Input: Yaw torque about the gimbal Output: Yaw rate at the sensor	Test Case 3c: Tapered beam, Thrust = 1000, Load Torque = 0 Input: Yaw torque about the gimbal Output: Pitch rate at the sensor	Test Case 3d: Tapered beam, Thrust = 1000, Load Torque = 0 Input: Yaw torque about the gimbal Output: Roll rate at the sensor
Test Case 4a: Tapered beam, Thrust = 1000, Load Torque = 1 Input: Yaw torque about the gimbal Output: Y acceleration at the sensor	Test Case 4b: Tapered beam, Thrust = 1000, Load Torque = 1 Input: Yaw torque about the gimbal Output: Yaw rate at the sensor	Test Case 4c: Tapered beam, Thrust = 1000, Load Torque = 1 Input: Yaw torque about the gimbal Output: Pitch rate at the sensor	Test Case 4d: Tapered beam, Thrust = 1000, Load Torque = 1 Input: Yaw torque about the gimbal Output: Roll rate at the sensor

- Notes:
- 1) 4 different test cases; “a”, “b”, “c”, “d” variants are just different outputs (Same system matrices)
 - 2) The “c” and “d” variants are designed to capture cross-axis responses
 - 3) Test case with non-zero thrust should exercise both force and moment forcing functions
 - 4) Test cases use engine mass props with non-zero engine cross products and cg offsets

Figure F.8.1. Initial Set of Test Cases for Phase 1

Table F2 documents the expanded set of test cases used during the second phase of this task. Cases 1–4 match the Phase 1 cases. Cases 5–8 are similar, but add intermediate steps of complexity to aid in tool evaluation. Evaluation criterion are the same as for Phase 1.

Test Case	Input	K_{LT_p}	K_{LT_y}	Thrust (N)
1	Pitch ($\ddot{\delta}_{P_c}$)	0	0	1000
2	Pitch (τ_{P_c})	1	1	1000
3	Yaw ($\ddot{\delta}_{Y_c}$)	0	0	1000
4	Yaw (τ_{Y_c})	1	1	1000
5	Pitch (τ_{P_c})	1	0	1000
6	Yaw (τ_{Y_c})	0	1	1000
7	Pitch (τ_{P_c})	1	1	0
8	Yaw (τ_{Y_c})	1	1	0

Table F2. Expanded Set of Test Cases for Phase 2

F.9 Test Case 1: Pitch Prescribed Engine Angular Acceleration Input

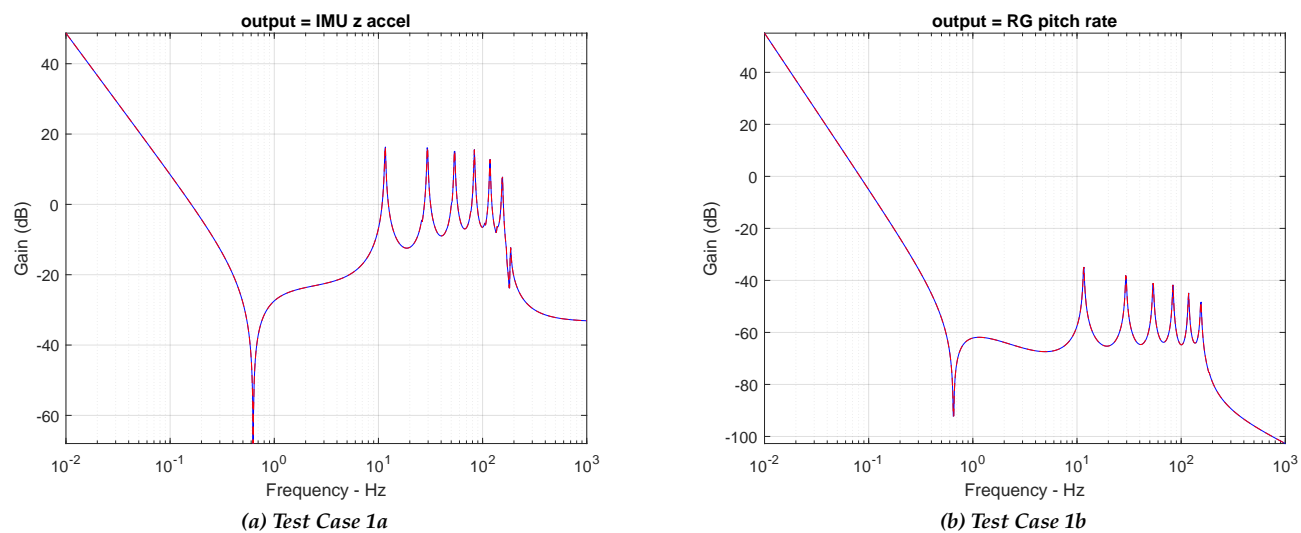


Figure F.9.1. Overlay of DARTS and MSFC Responses, Test Cases 1a and 1b

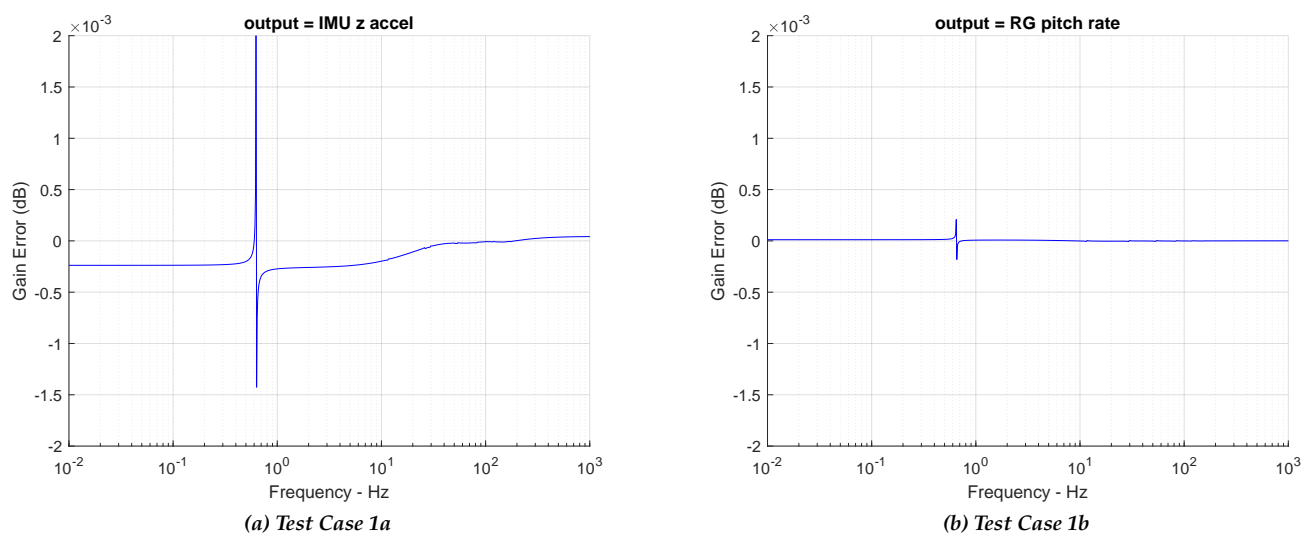


Figure F.9.2. Zoomed Differences

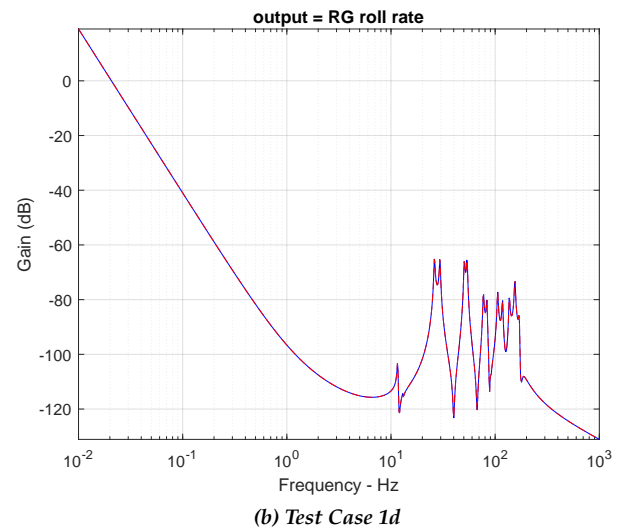
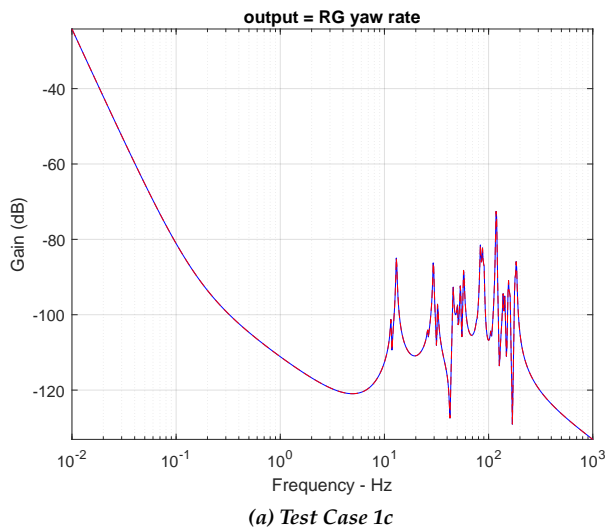


Figure F.9.3. Overlay of DARTS and MSFC Responses, Test Cases 1c and 1d

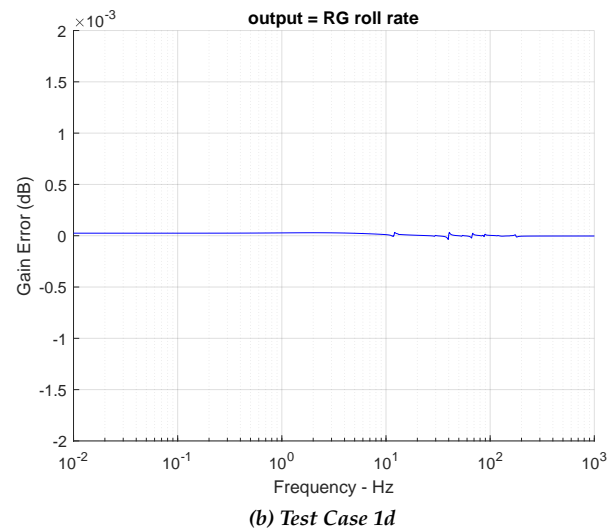
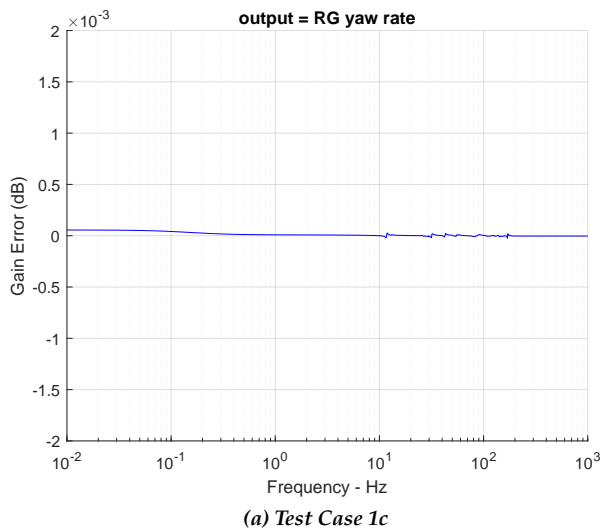
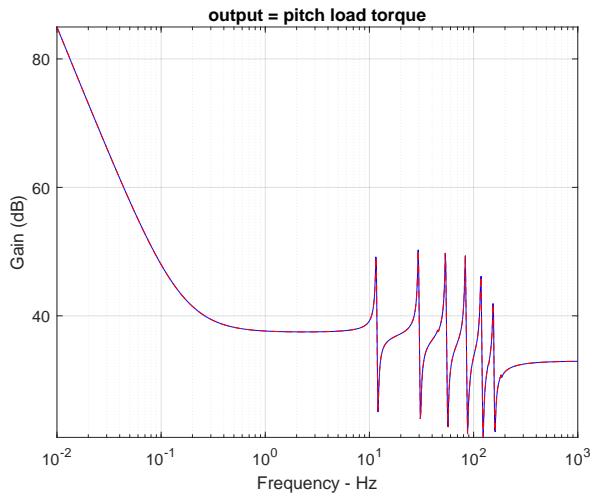
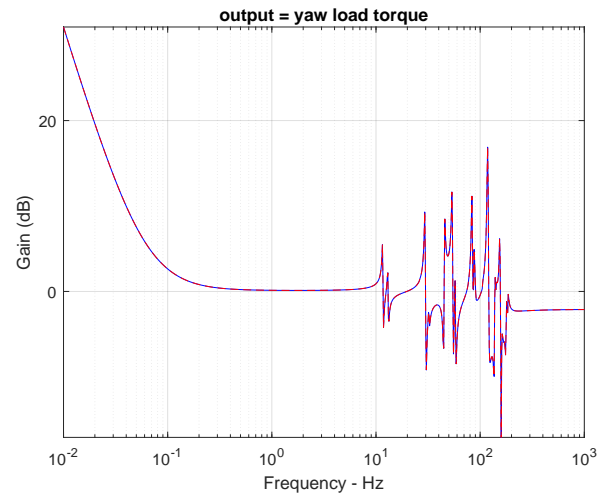


Figure F.9.4. Zoomed Differences

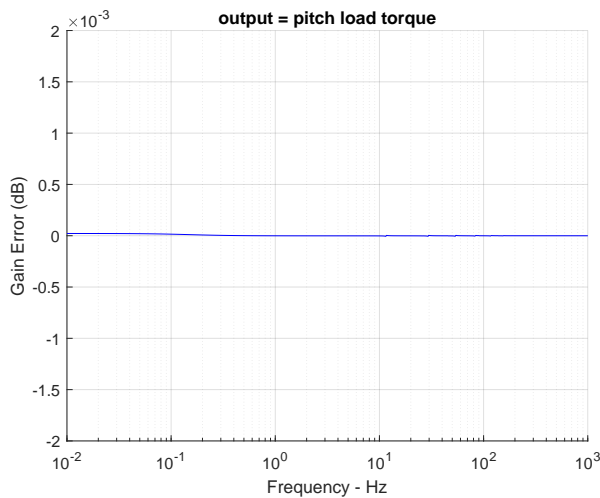


(a) Test Case 1e

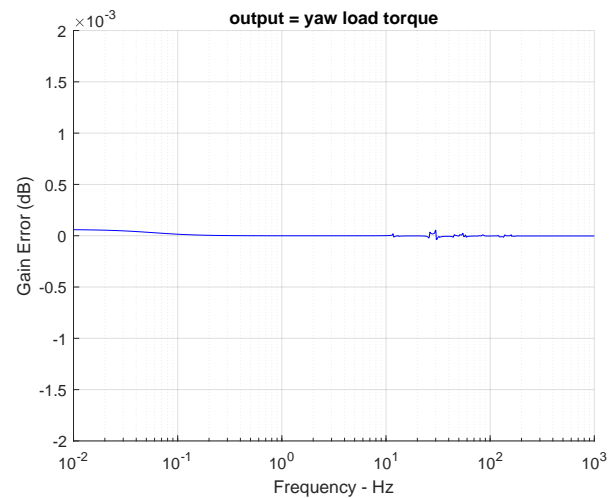


(b) Test Case 1f

Figure F.9.5. Overlay of DARTS and MSFC Responses, Test Cases 1e and 1f

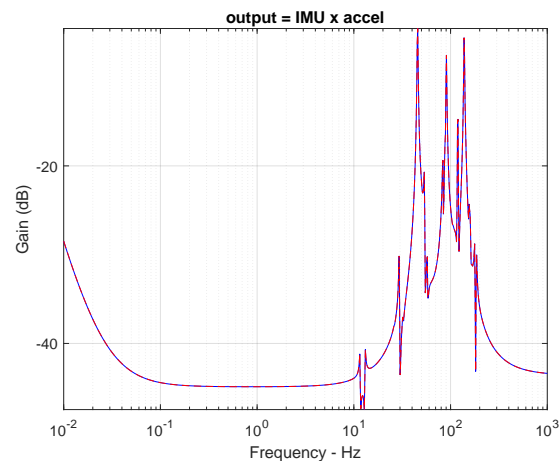


(a) Test Case 1e

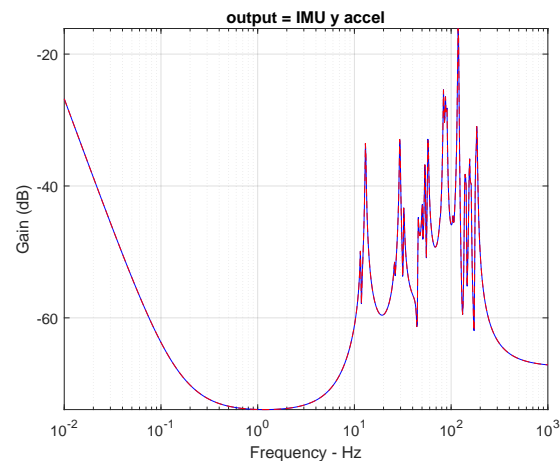


(b) Test Case 1f

Figure F.9.6. Zoomed Differences

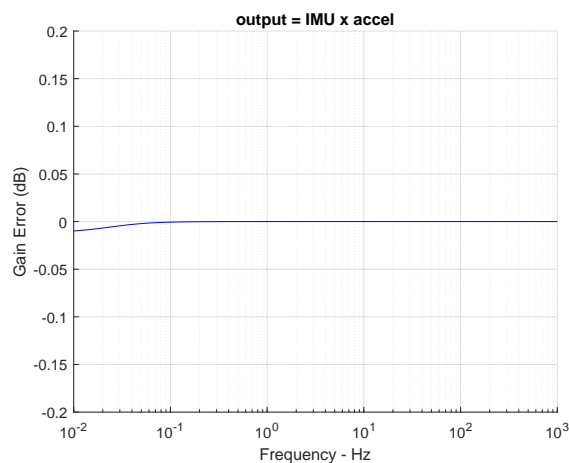


(a) Test Case 1g

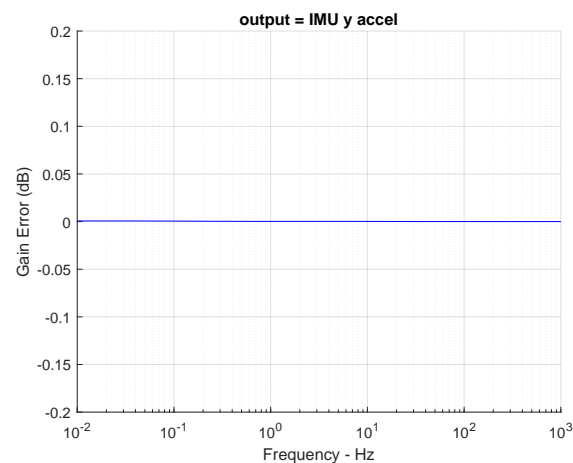


(b) Test Case 1h

Figure F.9.7. Overlay of DARTS and MSFC Responses, Test Cases 1g and 1h

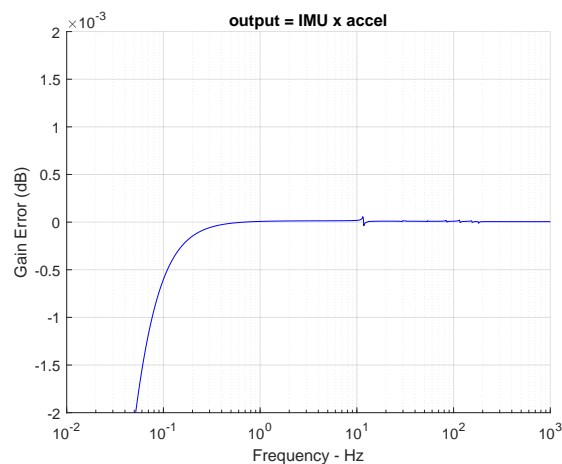


(a) Test Case 1g

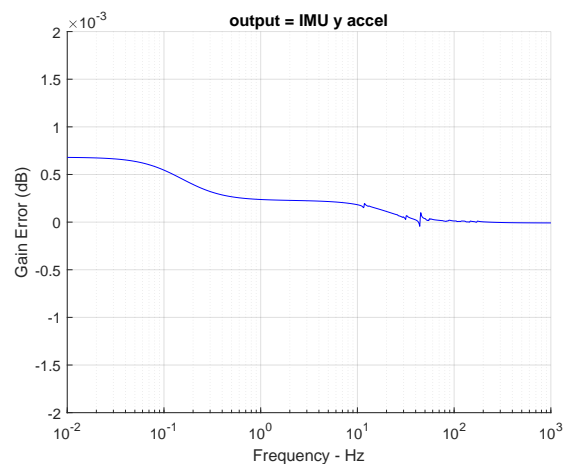


(b) Test Case 1h

Figure F.9.8. Difference Between DARTS and MSFC Responses



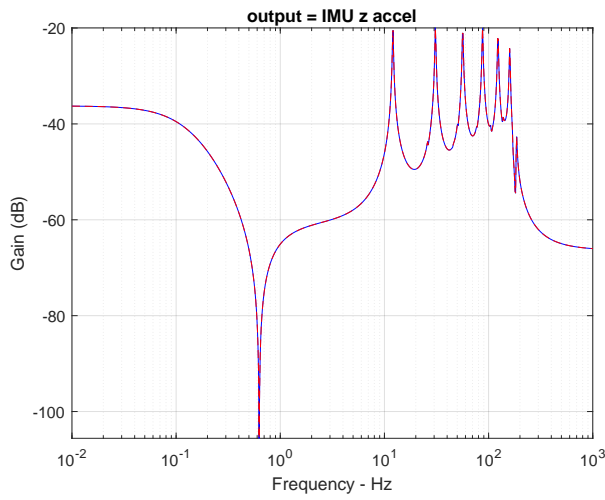
(a) Test Case 1g



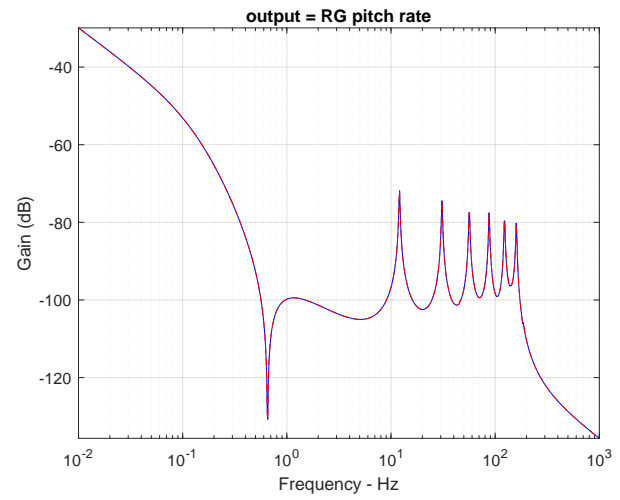
(b) Test Case 1h

Figure F.9.9. Zoomed Differences

F.10 Test Case 2: Pitch Engine Gimbal Torque Input

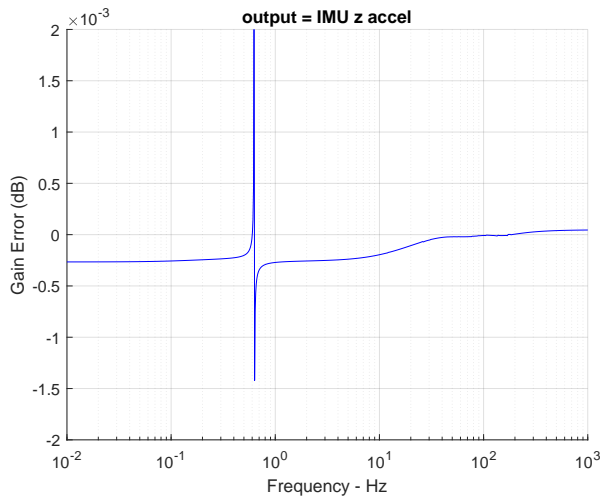


(a) Test Case 2a

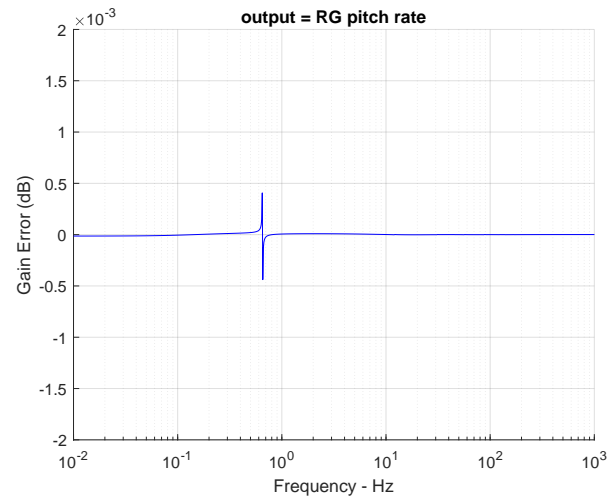


(b) Test Case 2b

Figure F.10.1. Overlay of DARTS and MSFC Responses, Test Cases 2a and 2b



(a) Test Case 2a



(b) Test Case 2b

Figure F.10.2. Zoomed Differences

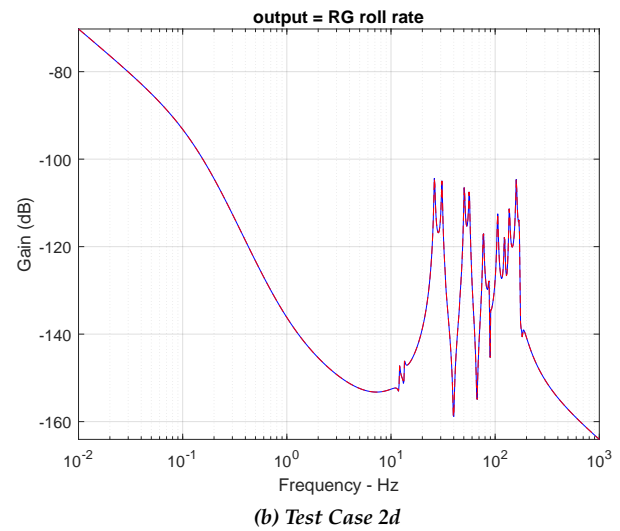
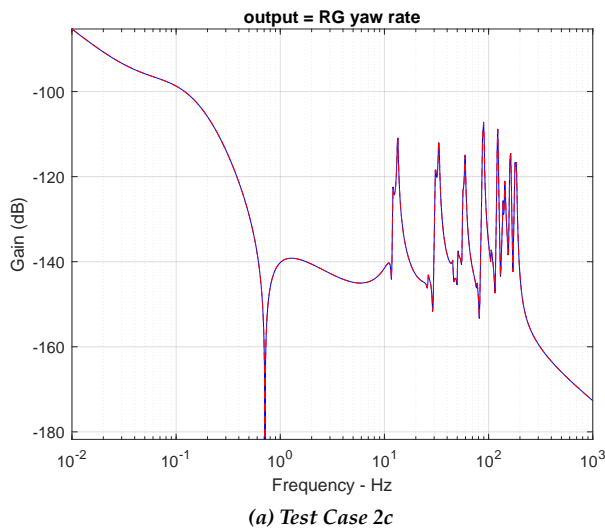


Figure F.10.3. Overlay of DARTS and MSFC Responses, Test Cases 2c and 2d

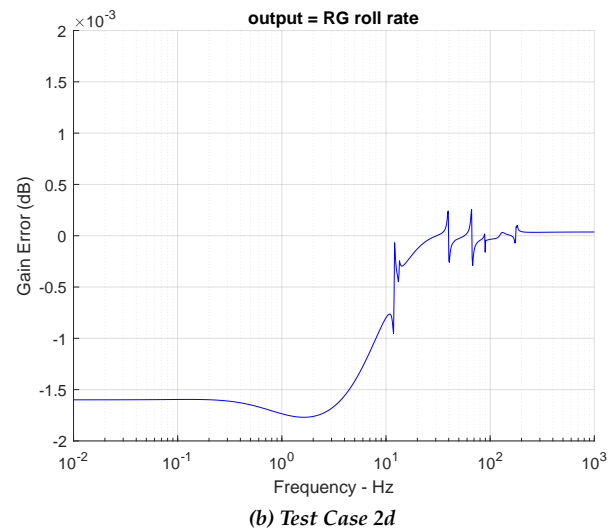
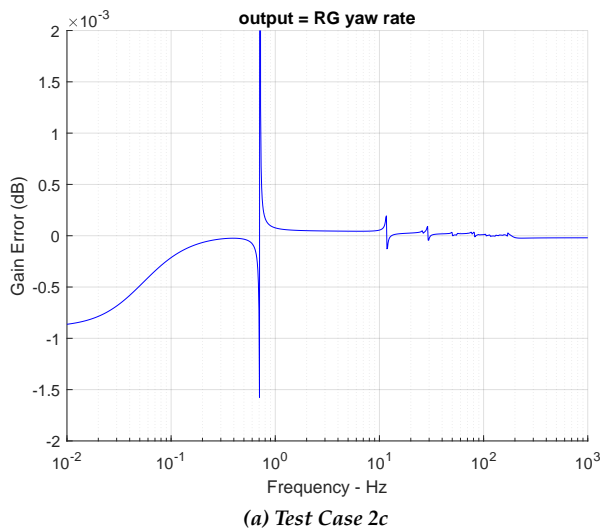


Figure F.10.4. Zoomed Differences

F.11 Test Case 3: Pitch Prescribed Engine Angular Acceleration Input

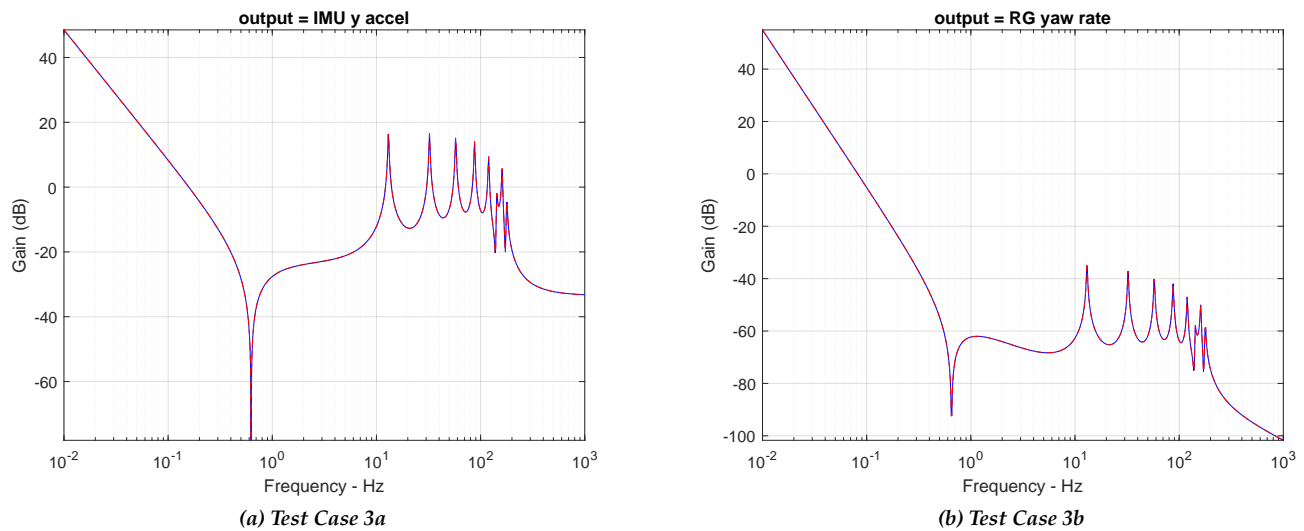


Figure F.11.1. Overlay of DARTS and MSFC Responses, Test Cases 3a and 3b

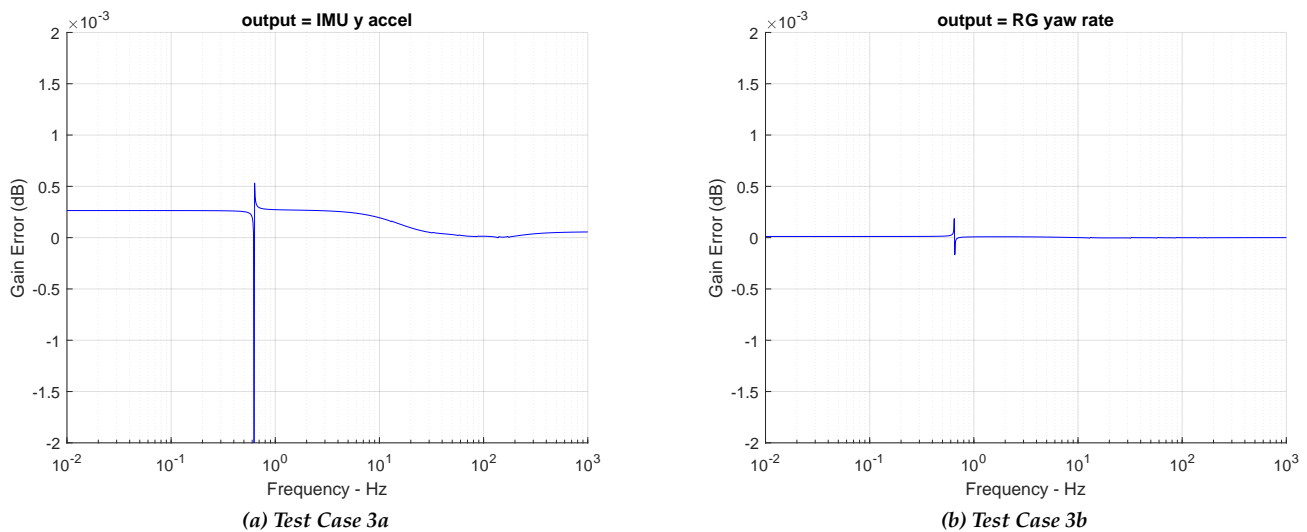


Figure F.11.2. Zoomed Differences

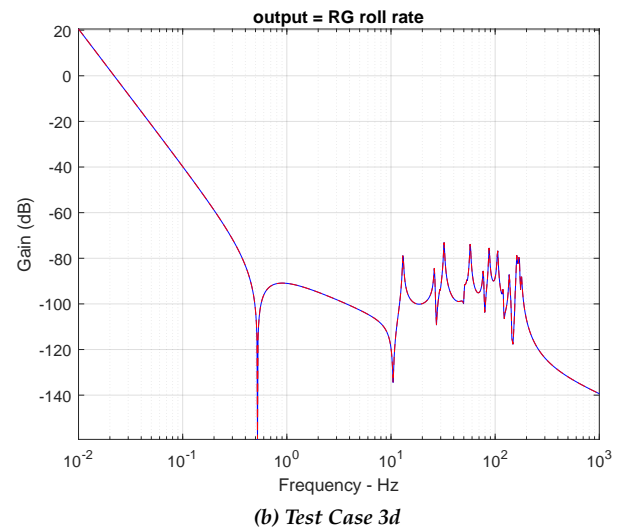
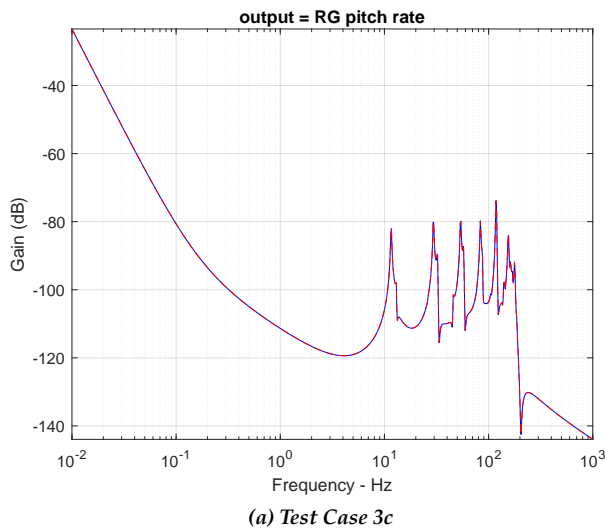


Figure F.11.3. Overlay of DARTS and MSFC Responses, Test Cases 3c and 3d

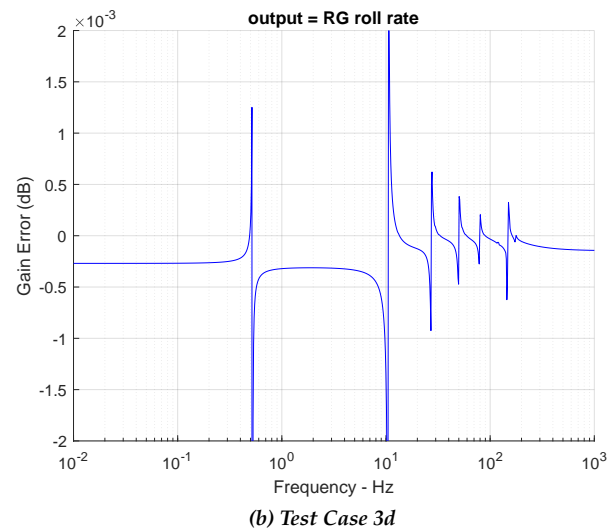
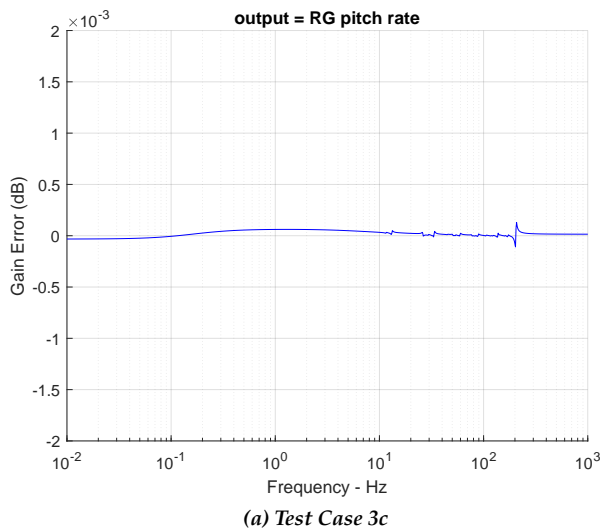
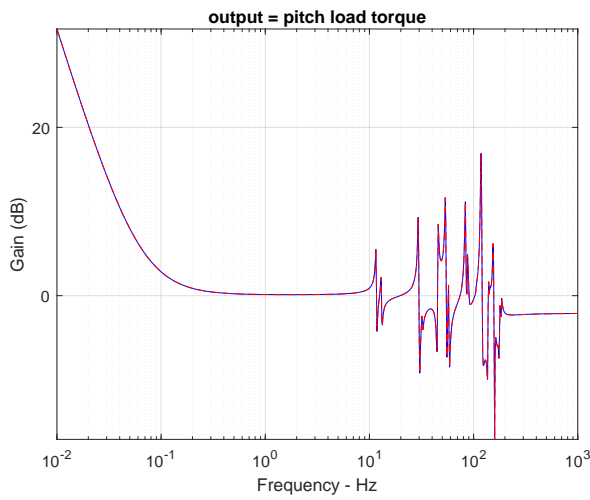
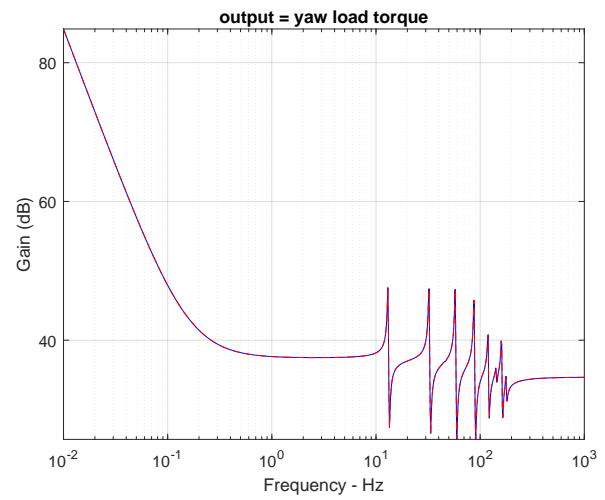


Figure F.11.4. Zoomed Differences

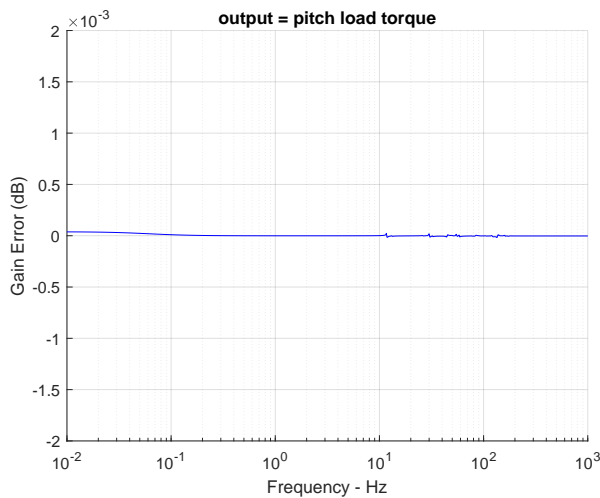


(a) Test Case 3e

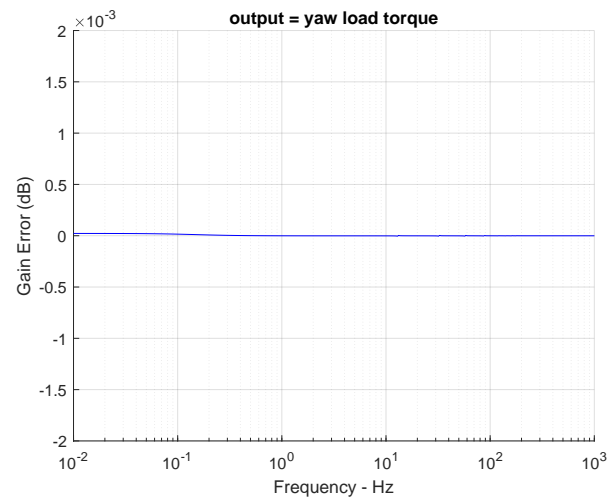


(b) Test Case 3f

Figure F.11.5. Overlay of DARTS and MSFC Responses, Test Cases 3e and 3f

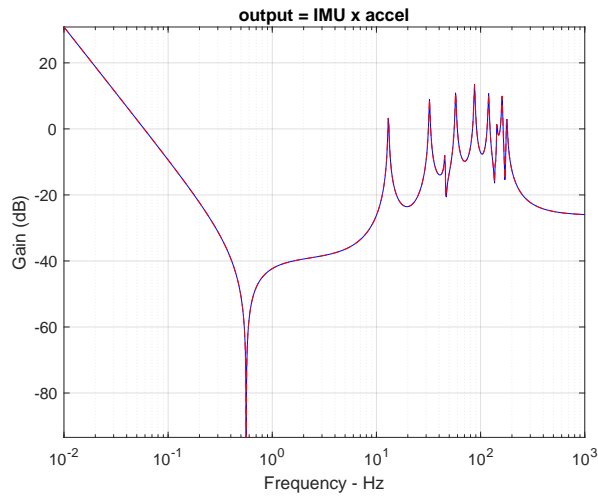


(a) Test Case 3e

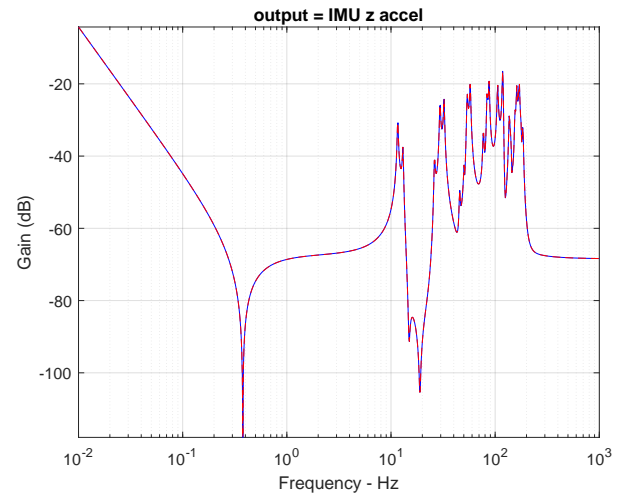


(b) Test Case 3f

Figure F.11.6. Zoomed Differences

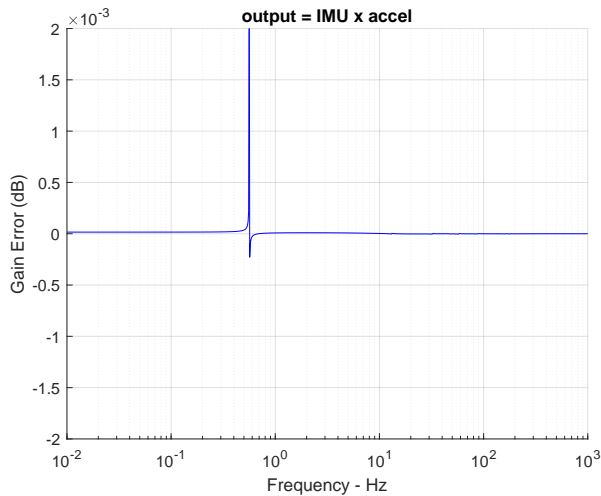


(a) Test Case 3g

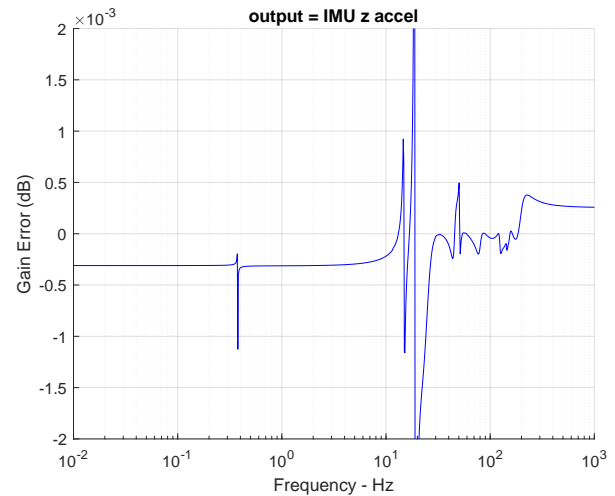


(b) Test Case 3h

Figure F.11.7. Overlay of DARTS and MSFC Responses, Test Cases 3g and 3h



(a) Test Case 3g



(b) Test Case 3h

Figure F.11.8. Zoomed Differences

F.12 Test Case 4: Yaw Engine Gimbal Torque Input

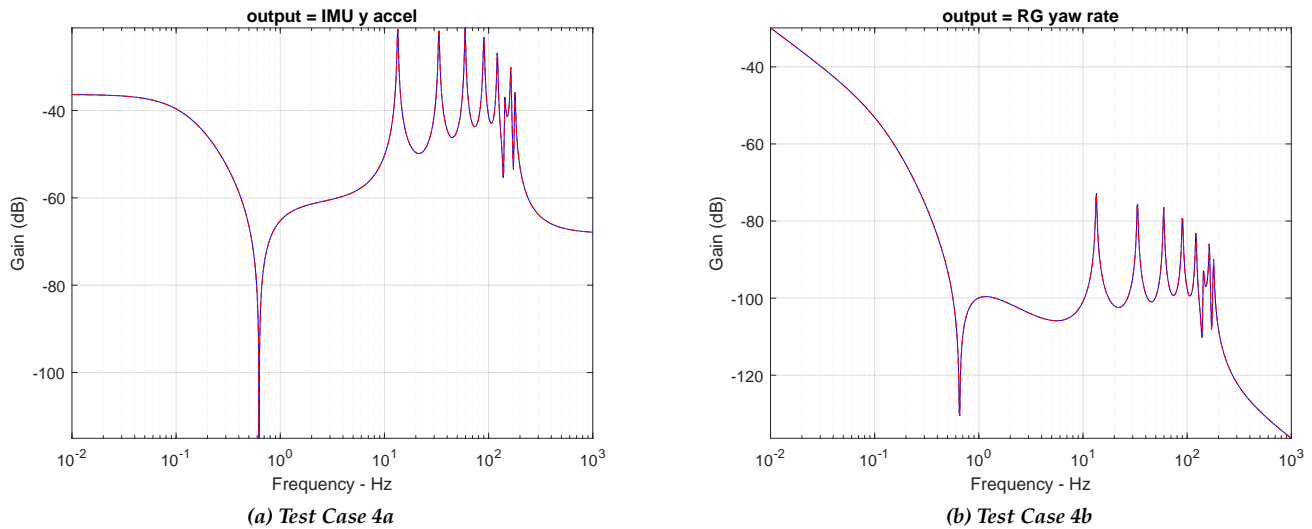


Figure F.12.1. Overlay of DARTS and MSFC Responses, Test Cases 4a and 4b

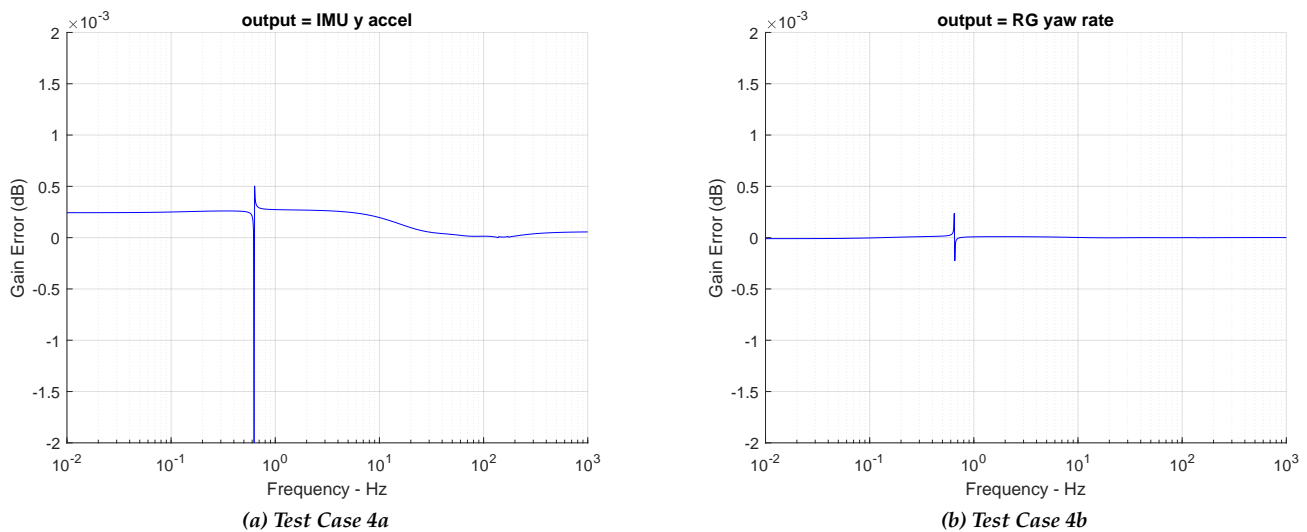


Figure F.12.2. Zoomed Differences

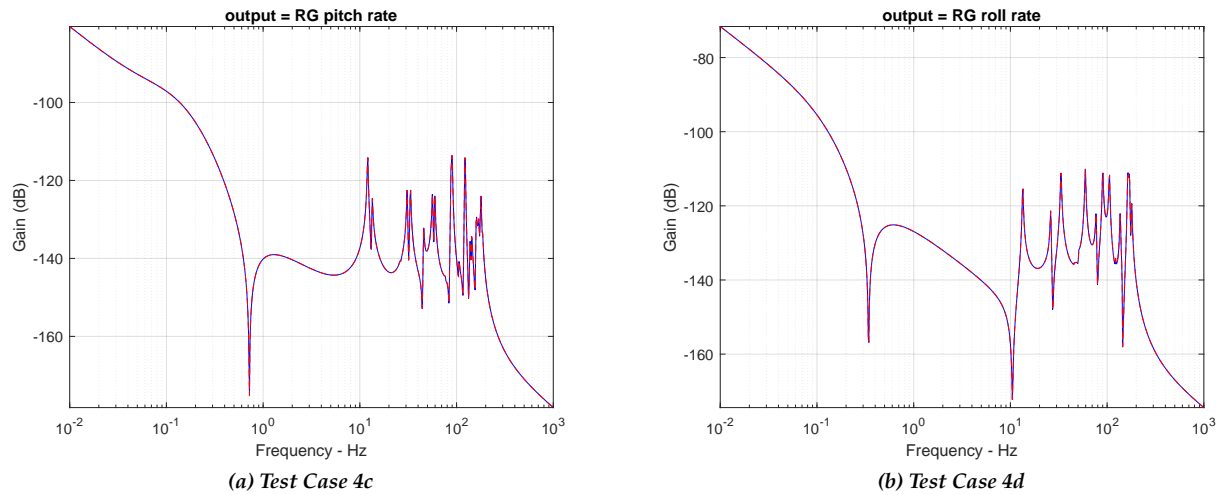


Figure F.12.3. Overlay of DARTS and MSFC Responses, Test Cases 4c and 4d

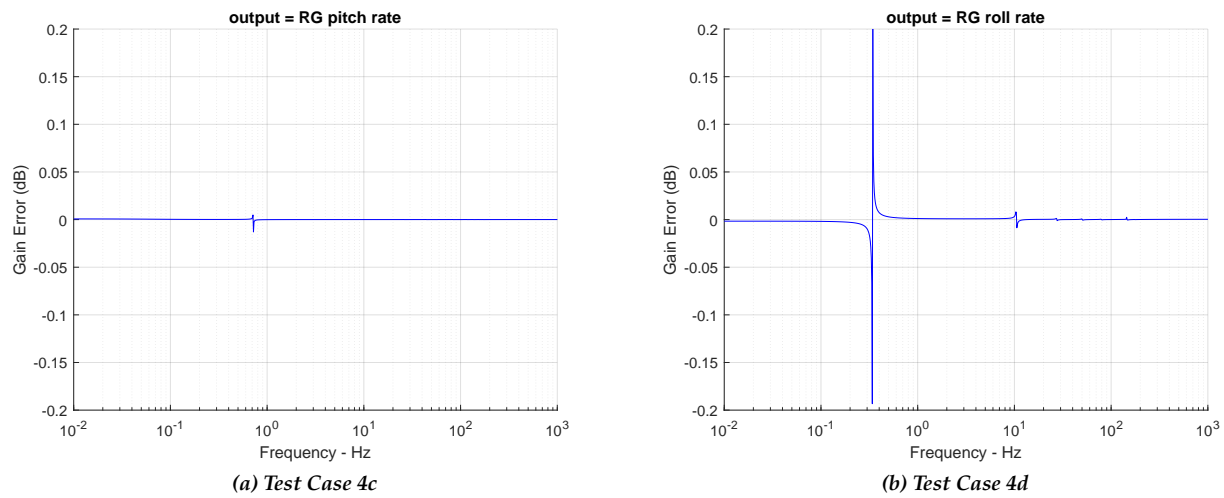


Figure F.12.4. Difference Between DARTS and MSFC Responses

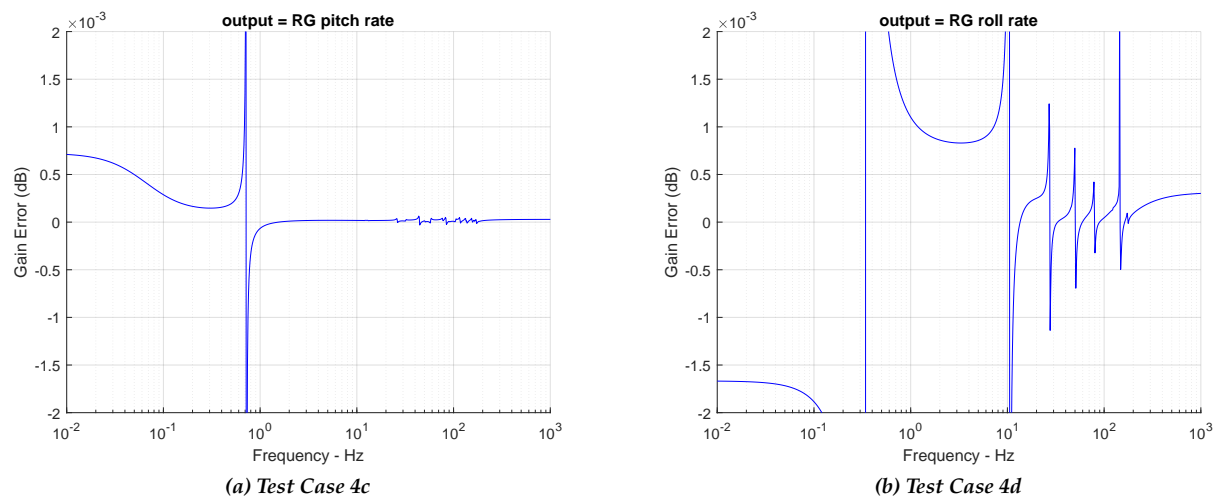
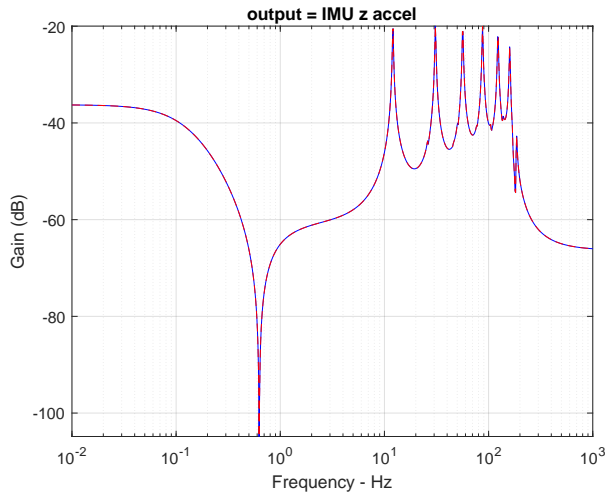
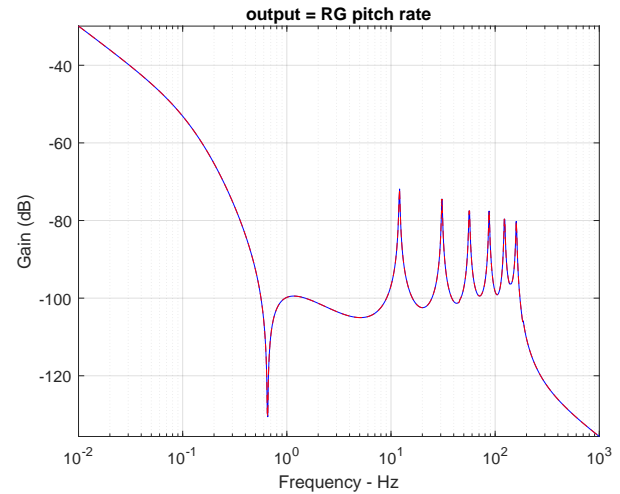


Figure F.12.5. Zoomed Differences

F.13 Test Case 5: Pitch Engine Gimbal Torque Input with Yaw Prescribed Input (Repeat of Case 2 with Yaw Prescribed)

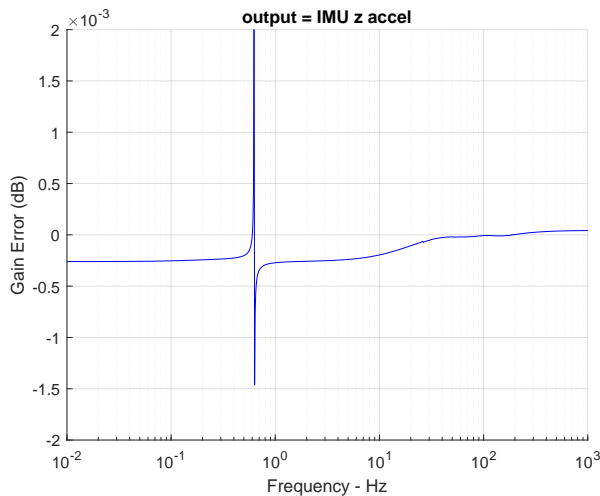


(a) Test Case 5a

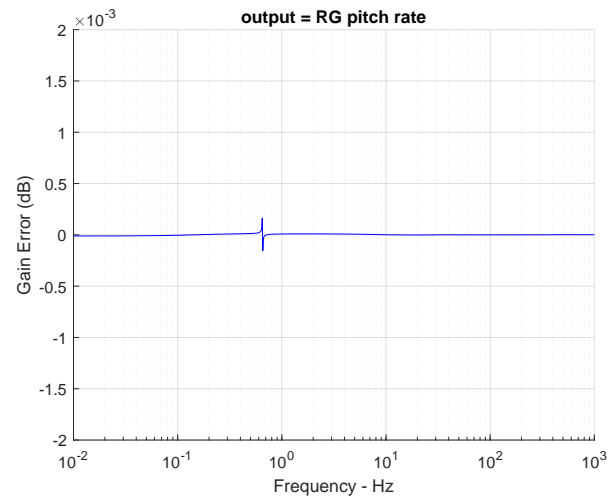


(b) Test Case 5b

Figure F.13.1. Overlay of DARTS and MSFC Responses, Test Cases 5a and 5b



(a) Test Case 5a



(b) Test Case 5b

Figure F.13.2. Zoomed Differences

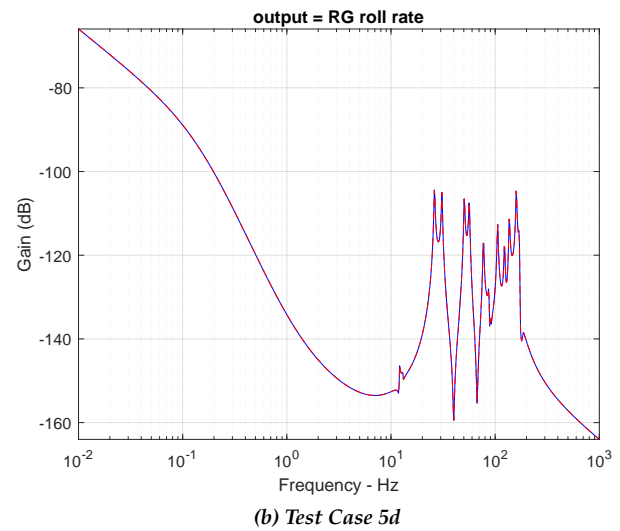
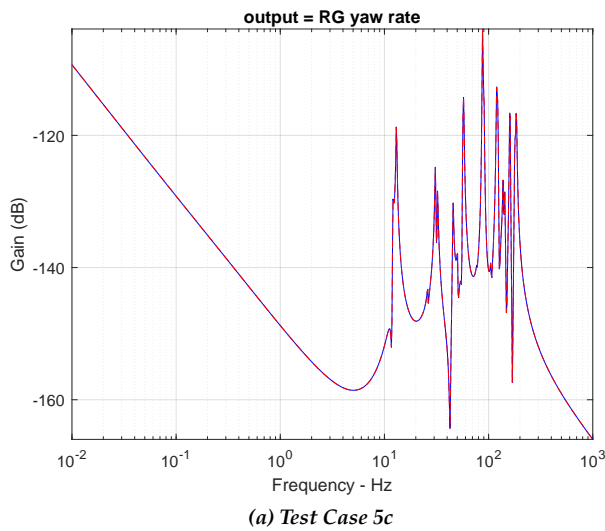


Figure F.13.3. Overlay of DARTS and MSFC Responses, Test Cases 5c and 5d

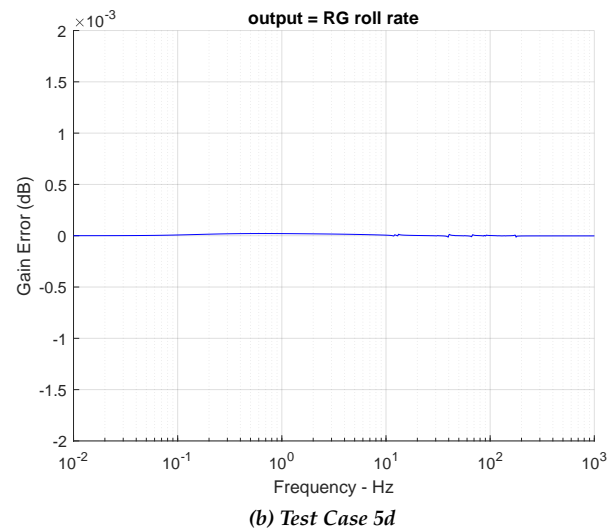
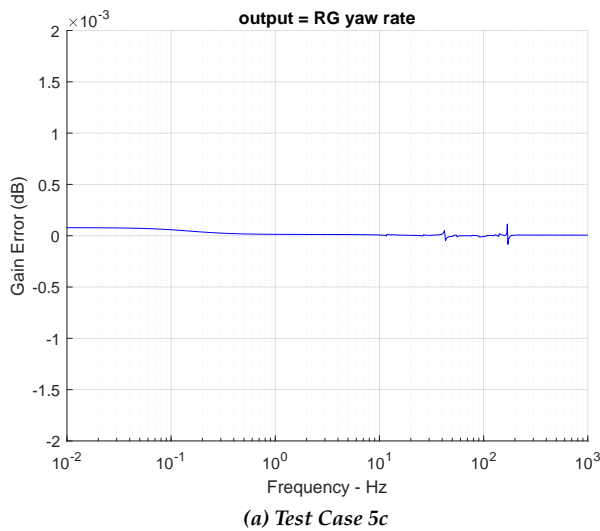
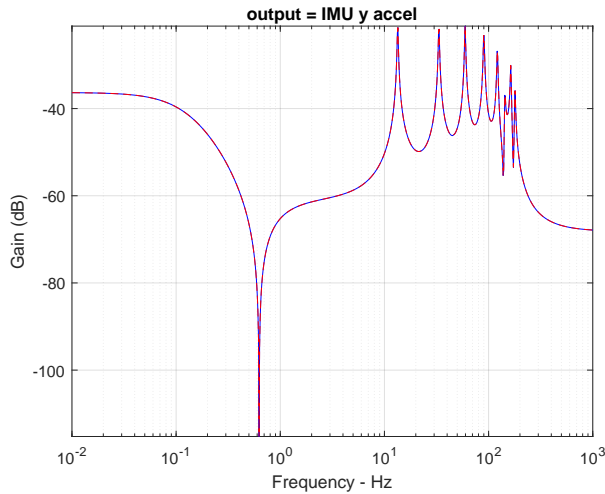
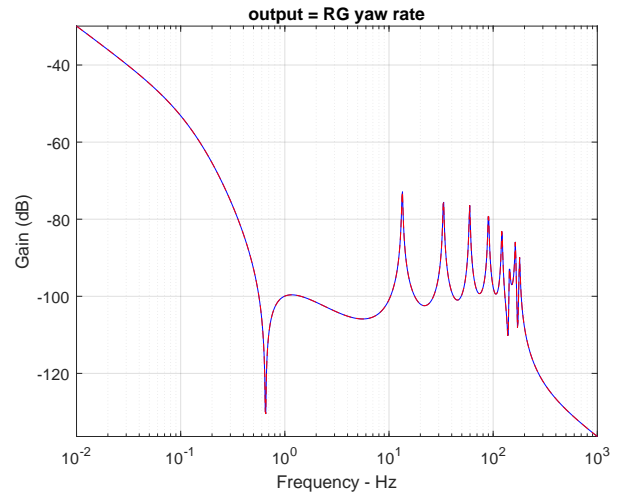


Figure F.13.4. Zoomed Differences

F.14 Test Case 6: Yaw Engine Gimbal Torque Input with Pitch Prescribed Input (Repeat of Case 4 with Pitch Prescribed)

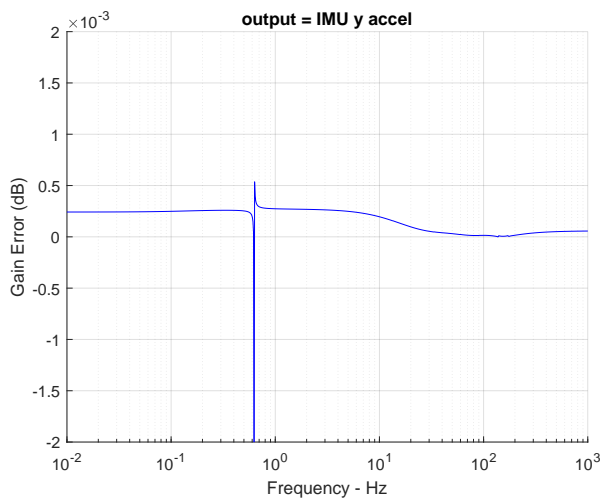


(a) Test Case 6a

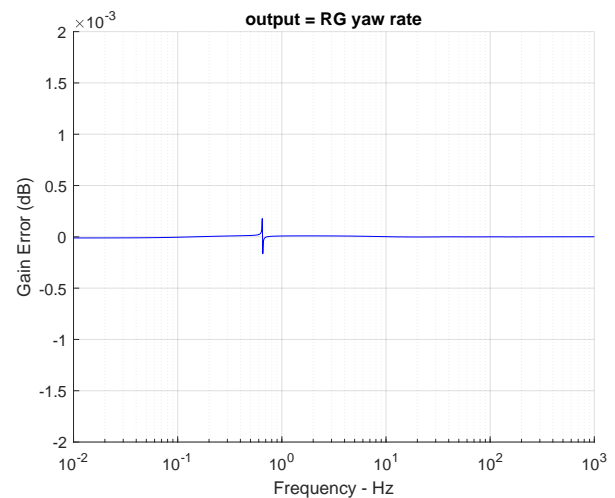


(b) Test Case 6b

Figure F.14.1. Overlay of DARTS and MSFC Responses, Test Cases 6a and 6b

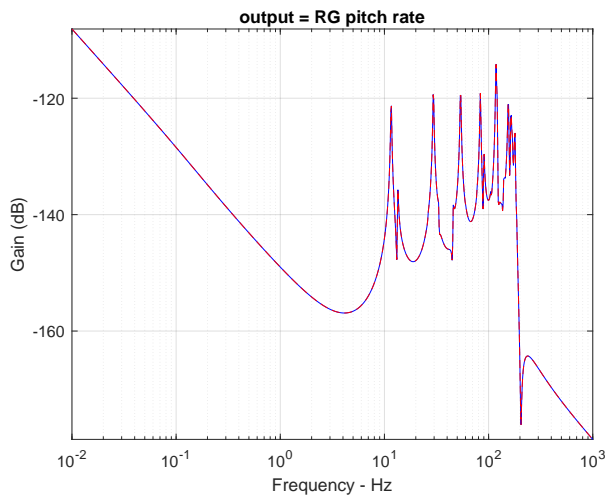


(a) Test Case 6a

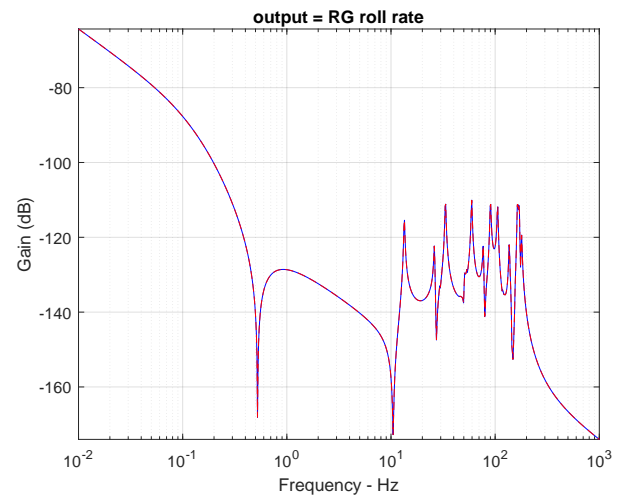


(b) Test Case 6b

Figure F.14.2. Zoomed Differences

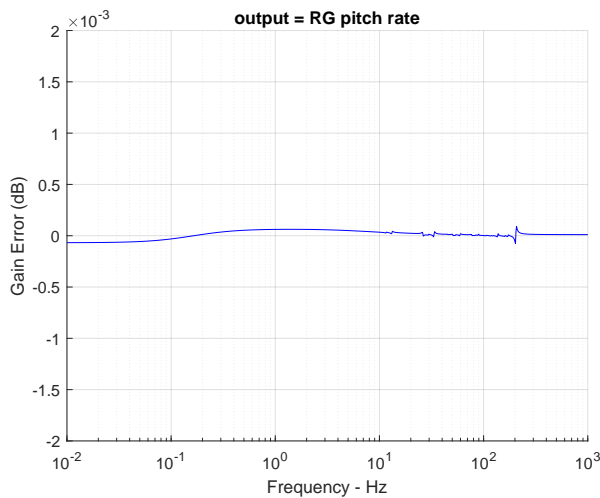


(a) Test Case 6c

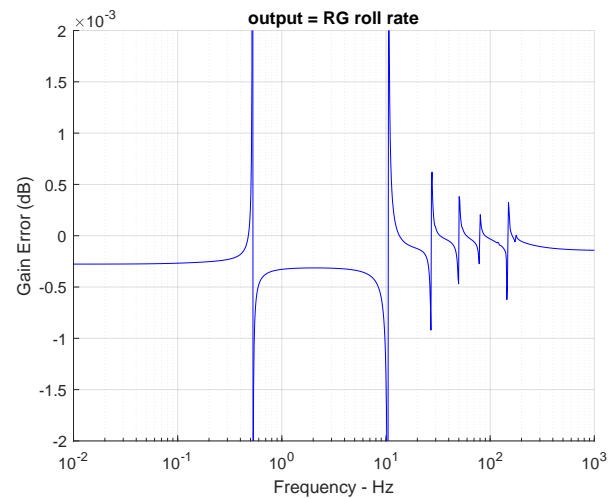


(b) Test Case 6d

Figure F.14.3. Overlay of DARTS and MSFC Responses, Test Cases 6c and 6d



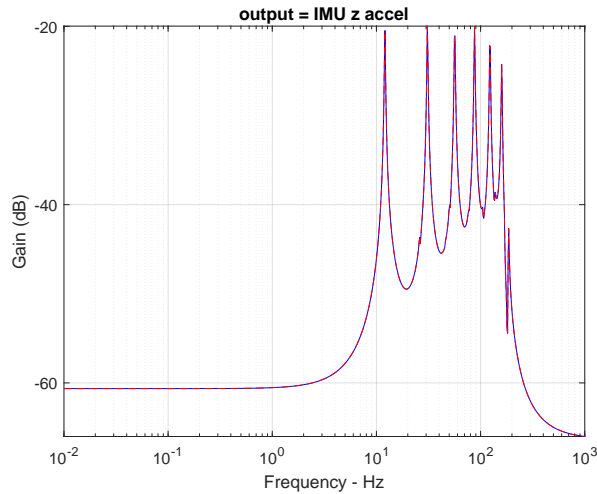
(a) Test Case 6c



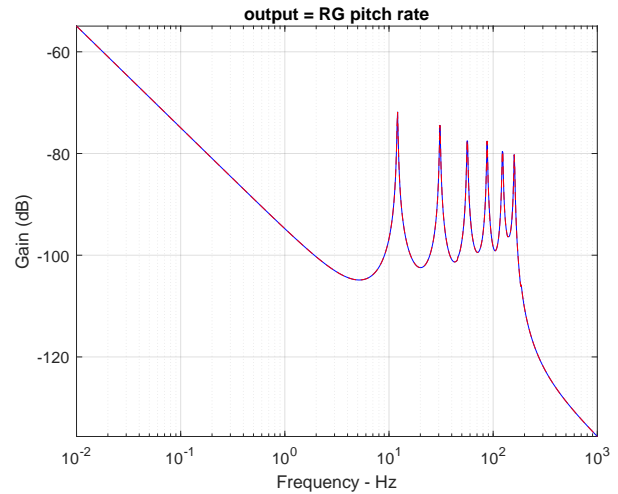
(b) Test Case 6d

Figure F.14.4. Zoomed Differences

F.15 Test Case 7: Pitch Engine Gimbal Torque Input, Thrust = 0 (Repeat of Case 2 with Zero Thrust)

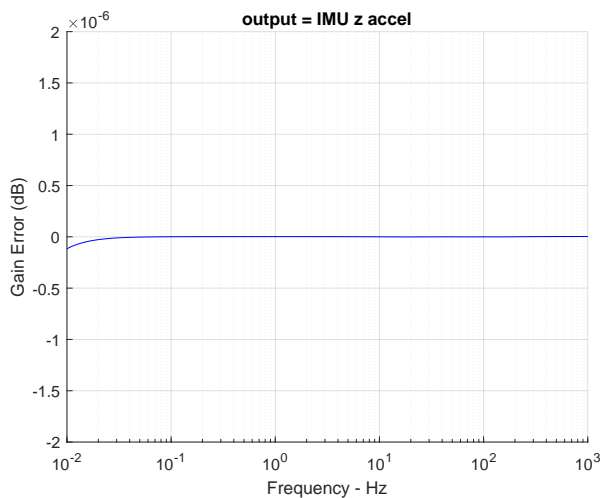


(a) Test Case 7a

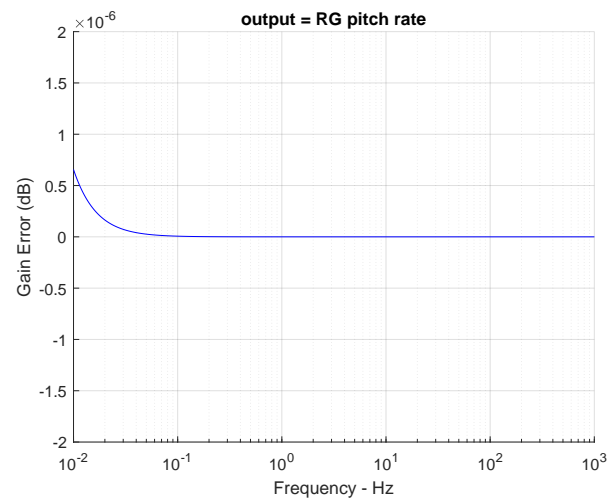


(b) Test Case 7b

Figure F.15.1. Overlay of DARTS and MSFC Responses, Test Cases 2a and 2b, $T = 0$



(a) Test Case 7a



(b) Test Case 7b

Figure F.15.2. Zoomed Differences (Note Scale of $\pm 2 \mu\text{dB}$)

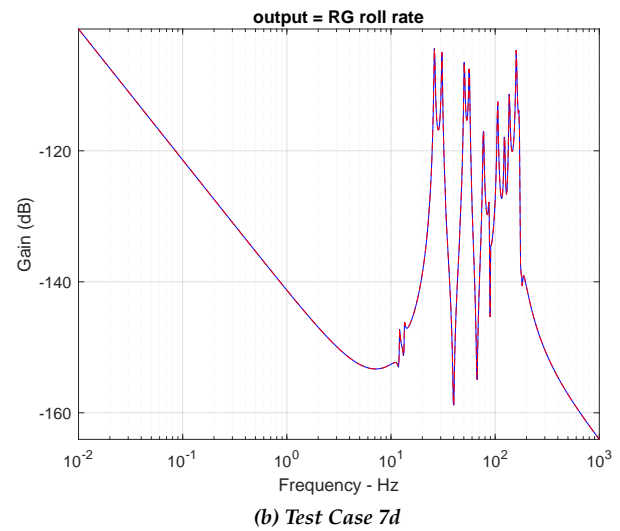
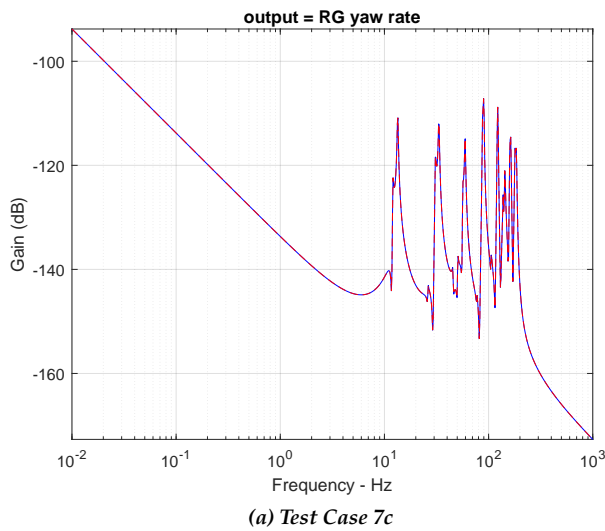


Figure F.15.3. Overlay of DARTS and MSFC Responses, Test Cases 2c and 2d, $T = 0$

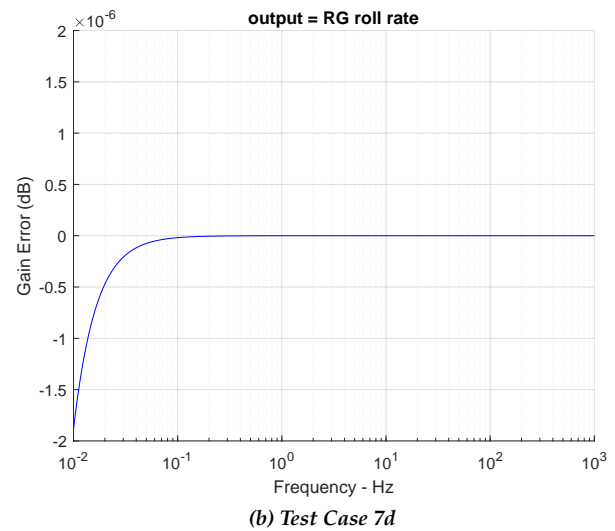
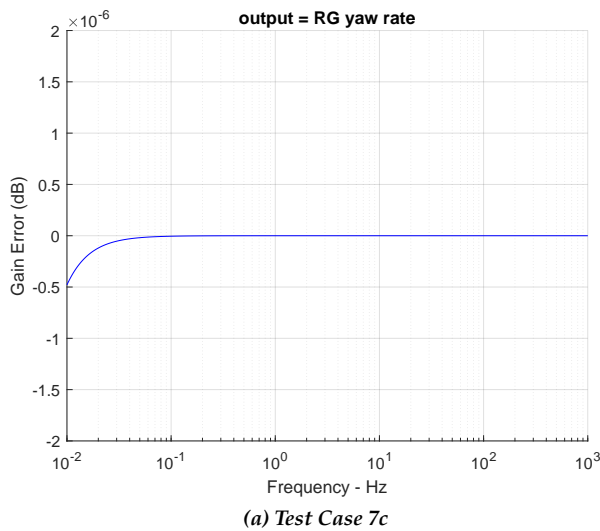


Figure F.15.4. Zoomed Differences (Note Scale of $\pm 2 \mu\text{dB}$)

F.16 Test Case 8: Yaw Engine Gimbal Torque Input, Thrust = 0 (Repeat of Case 4 with Zero Thrust)

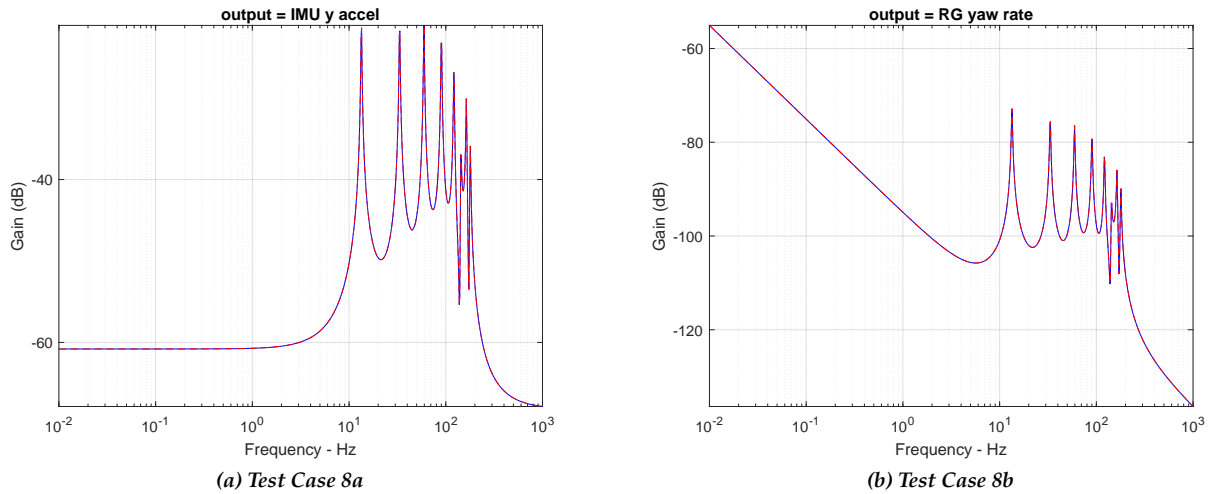


Figure F.16.1. Overlay of DARTS and MSFC Responses, Test Cases 4a and 4b, T = 0

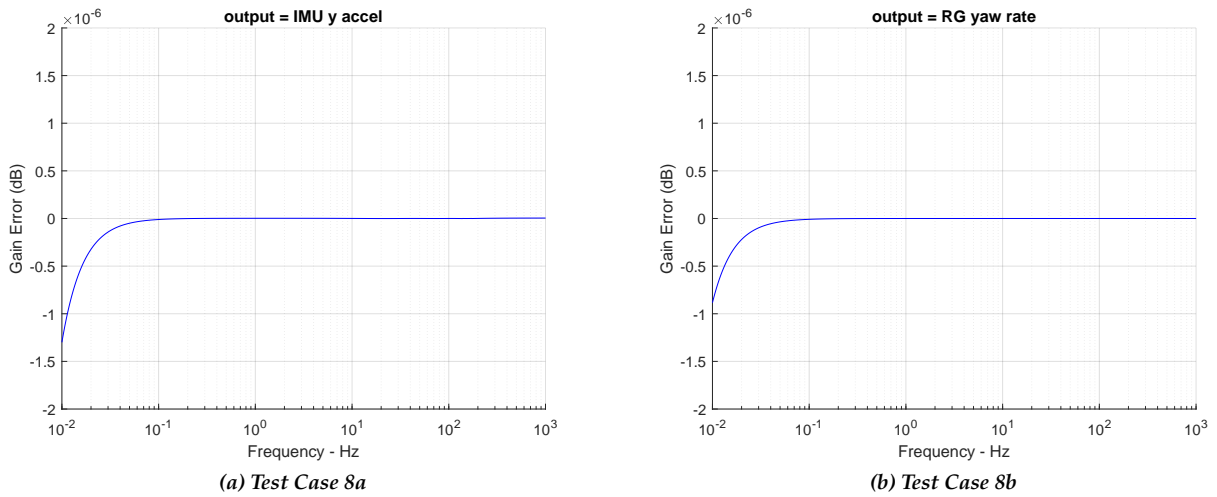


Figure F.16.2. Zoomed Differences (Note Scale of +/- 2 μ dB)

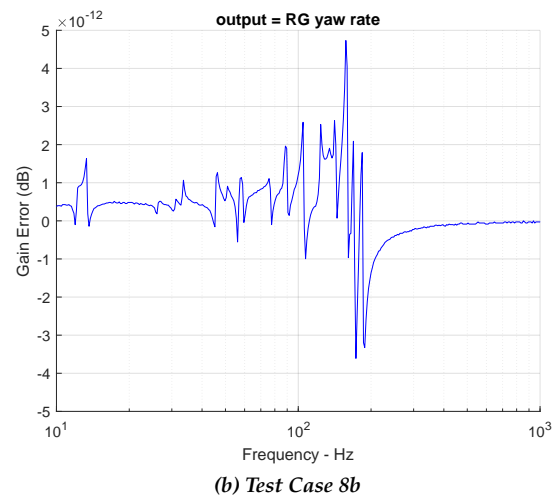
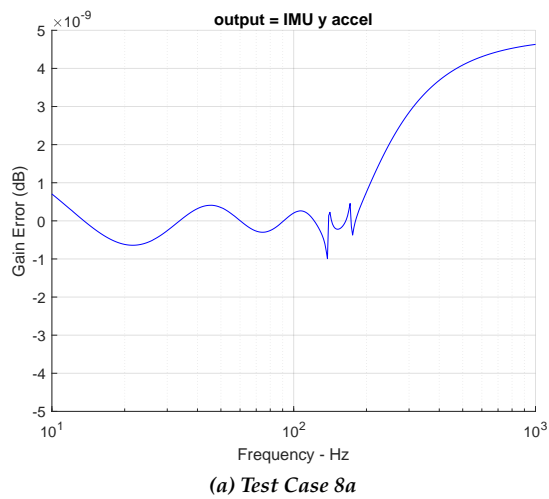


Figure F.16.3. Zoomed Differences (Note frequency range and vertical scales)

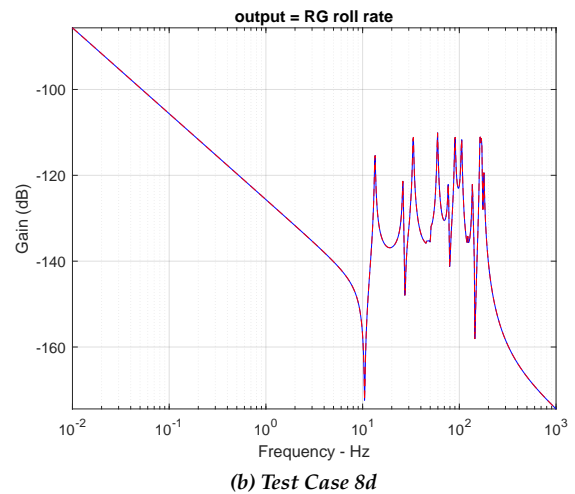
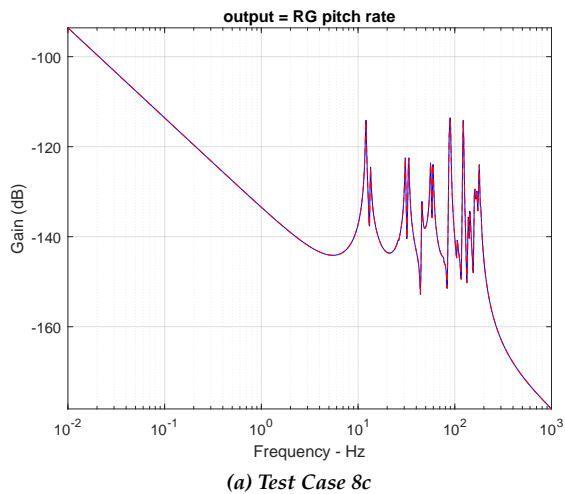
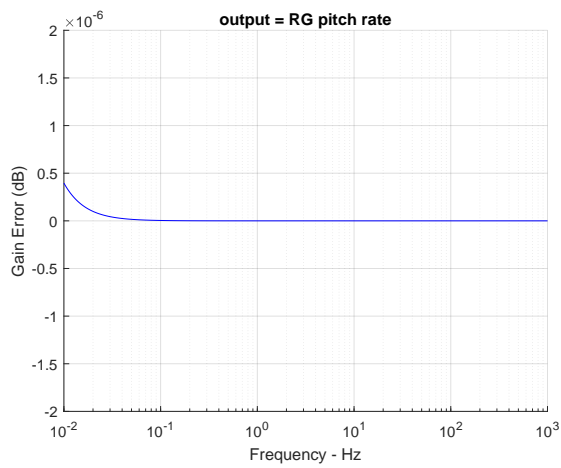
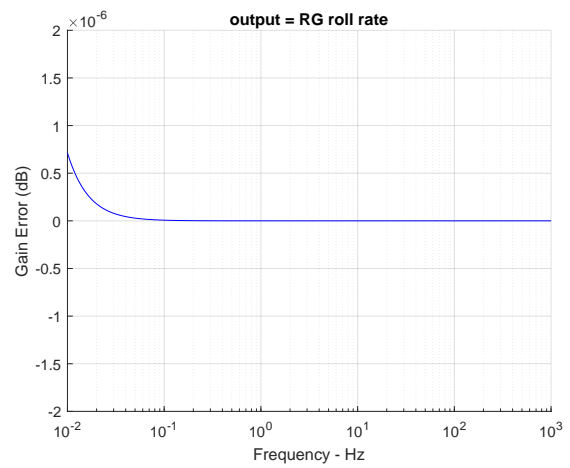


Figure F.16.4. Overlay of DARTS and MSFC Responses, Test Cases 4c and 4d, $T = 0$

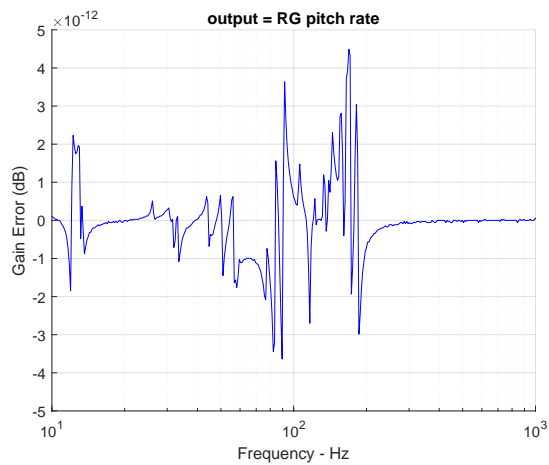


(a) Test Case 8c

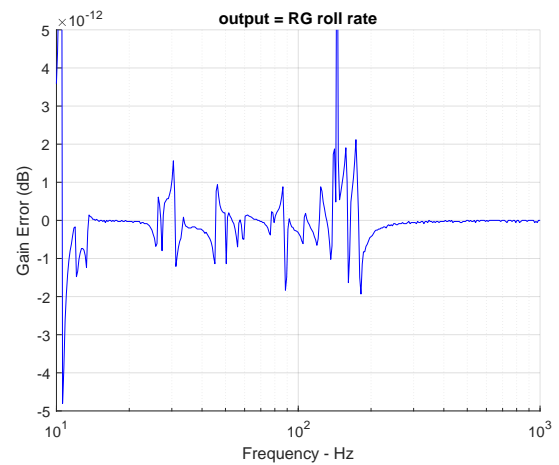


(b) Test Case 8d

Figure E.16.5. Zoomed Differences (Note Scale of $\pm 2 \mu\text{dB}$)



(a) Test Case 8c



(b) Test Case 8d

Figure E.16.6. Zoomed Differences (Note frequency range and vertical scales)

F.17 Bonus Test Case: $\ddot{\delta}_{p_o}$ and $\ddot{\delta}_{y_o}$ Sensitivity Study Repeat of Test Case 4, Yaw Engine Gimbal Torque Input

Blue line: Nominal Case 4 response

Dashed red line: Both $\ddot{\delta}_{p_o}$ and $\ddot{\delta}_{y_o}$ set to zero

Solid red lines: Multipliers of $\pm 0.025\%$ on both $\ddot{\delta}_{p_o}$ and $\ddot{\delta}_{y_o}$

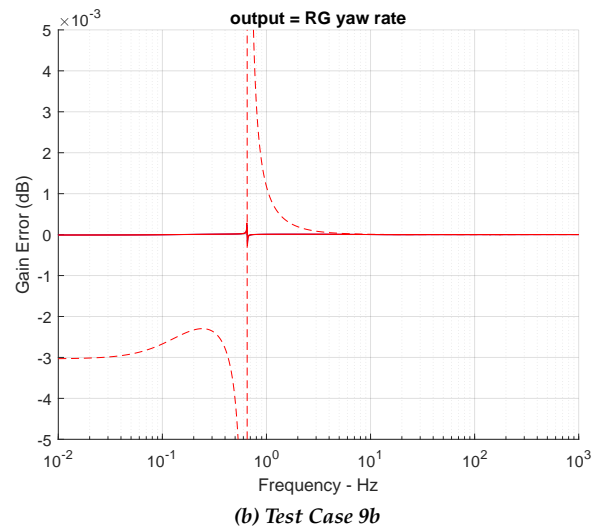
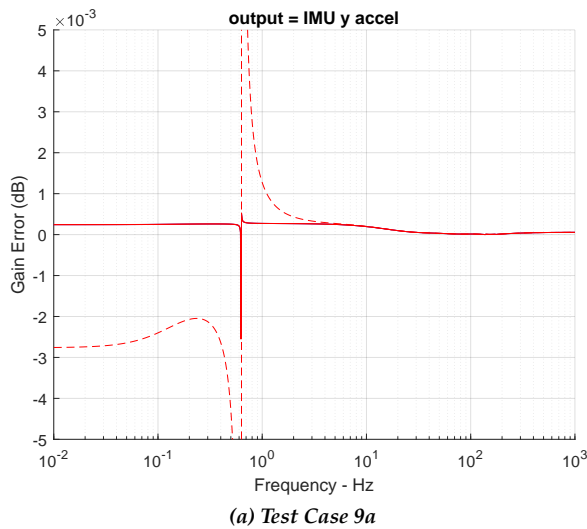


Figure F.17.1. Overlay of DARTS and MSFC Responses, MSFC Results with Variations in $\ddot{\delta}_{p_o}$ and $\ddot{\delta}_{y_o}$

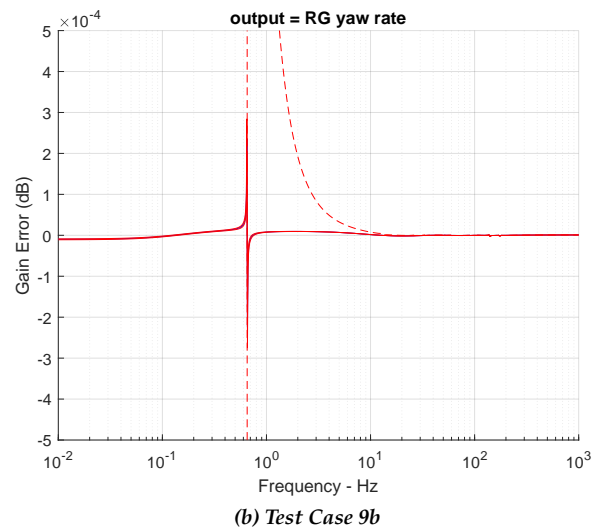
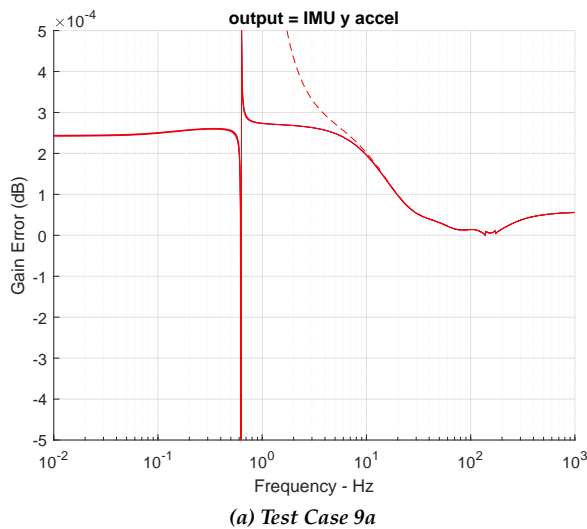
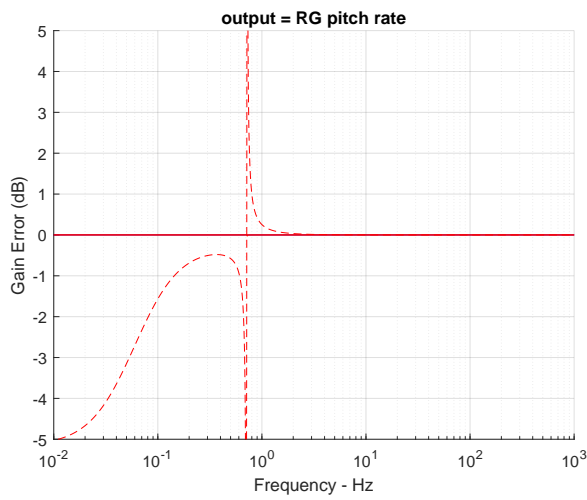
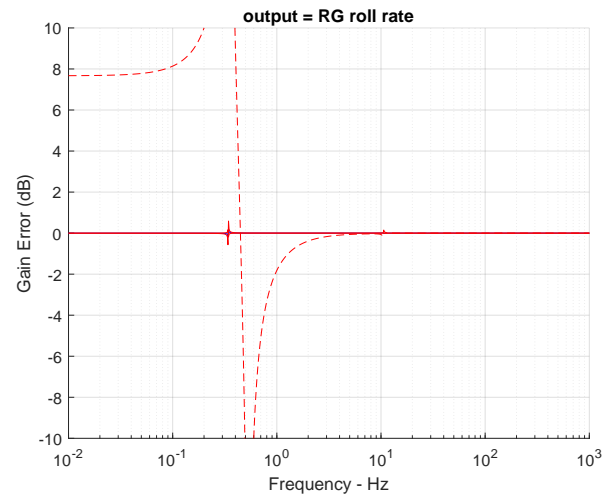


Figure F.17.2. Zoomed Differences

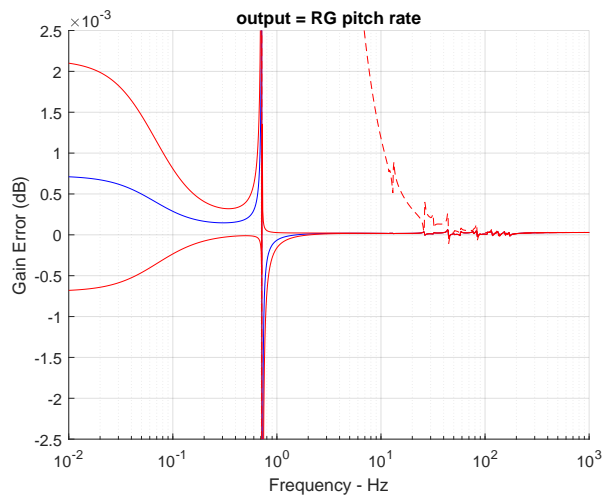


(a) Test Case 9a

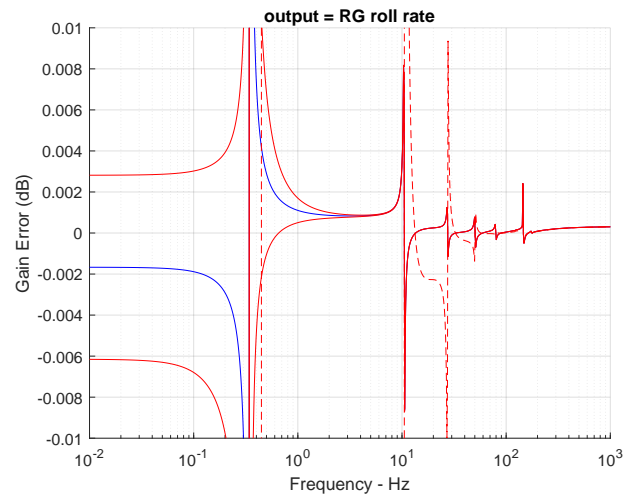


(b) Test Case 9b

Figure F.17.3. Overlay of DARTS and MSFC Responses, MSFC Results with Variations in $\ddot{\delta}_{p0}$ and $\ddot{\delta}_{y0}$



(a) Test Case 9a



(b) Test Case 9b

Figure F.17.4. Zoomed Differences

Appendix G. Flexible Spacecraft Modeling for Microthruster Assessment

G.1 Problem Description

Pointing requirements for large space-based observatories are difficult to meet with current reaction wheel-based architectures. Reaction wheels are typically the largest pointing disturbance on the observatory due to wheel imbalances and bearing disturbances. As observatory systems demand higher pointing performance, use of reaction wheels as a primary fine pointing actuator requires increasing system complexity (e.g., addition of vibration isolation), and at some performance level may no longer be feasible. Alternatively, microthrusters can be used as a fine pointing actuator with very small pointing disturbances ($\sim 1 \mu\text{N}$).

NESC assessment TI-18-01371, "Microthruster Systems for Low-Jitter Space Observatory Precision Attitude Control" seeks to evaluate the potential benefits of using cold-gas or colloidal microthrusters instead of traditional reaction wheel-based systems. One element of the assessment includes flexible body dynamics modeling for assessing settle time after slewing to a target and pointing stability in steady state. The Aerospace Corporation was contracted to develop the GN&C simulation, while the NESC assessment "Flexible Multibody Dynamics of Space Vehicles" was tasked with building the GN&C flexible body dynamics model. Figure G.1.1 illustrates the GN&C simulation architecture that was used to assess system performance.

Two space observatory-type structures were considered for GN&C analysis, which included (1) an antenna-shaped observatory with a segmented primary mirror and (2) a barrel-shaped telescope observatory. Flexible body dynamics models for GN&C analysis and simulation are normally built from an existing FEM developed by a structural engineering group. A FEM representative of an antenna-shaped observatory with segmented mirror was obtained from the LUVOIR concept study team [24]. FModal and DARTS were leveraged in this study to convert this FEM model into a notional GN&C flexible-body model in linear state space form that included reaction wheels, thrusters, and an IMU.

Implementation of this GN&C flexible body models differed in important respects from that of LUVOIR; including (but not limited to) addition of microthrusters. Consequently, performance was not intended or expected to be identical to that of LUVOIR. In the Microthruster Systems assessment, the LUVOIR-derived model is referred to as "Observatory 1". A GN&C flexible-body model representative of a barrel-shaped telescope observatory was obtained from the WFIRST project and similarly modified—the WFIRST-derived model is referred to as "Observatory 2."

G.2 LUVOIR-A FEM

Figure G.2.1 illustrates a deployed view of the LUVOIR-A 15-meter FEM model. The model's major components include the segmented primary mirror, the bus compartment, and the sunshade boom beams. The FEM model did not include actuator or reduced Craig-Bampton models. For the purposes of this study, the model was processed in a single configuration as a single body using FModal (see Appendix C) and Dshell (see Appendix D).

The as-delivered FEM model was processed using NX NASTRAN 12. Table G1 shows the rigid body mass properties of the vehicle about grid point 125 as computed by NASTRAN, where grid point 125 is the location of the telescope boresight node. For controller design, a cutoff frequency of 15 Hz was selected by The Aerospace Corporation. Using NASTRAN, a total of 212 mode shapes were computed for a 15 Hz cutoff frequency as shown in Figure G.2.2.

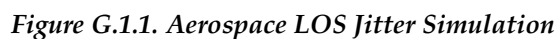


Table G1. LUVUOIR-A 15-meter Rigid Body Mass Properties About Grid Point 125

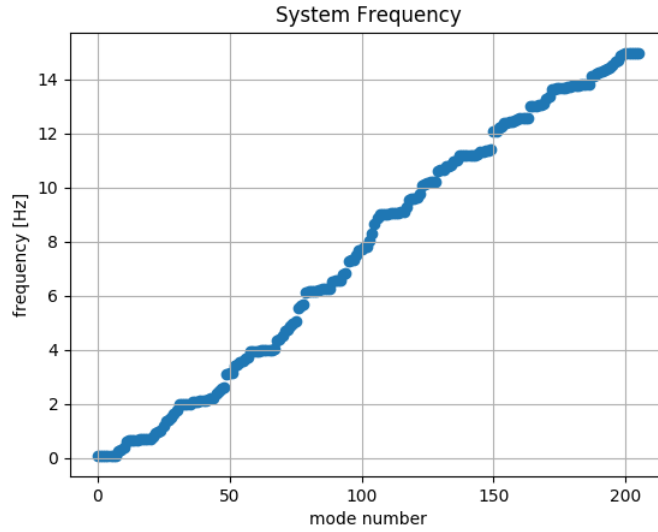


Figure G.2.2. LUVOR-A 15-meter Modal Frequencies Computed by NASTRAN

G.3 GN&C Actuators and Sensors

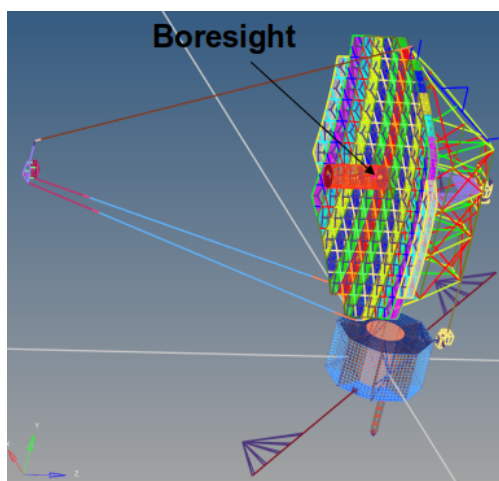
The as-delivered FEM model did not include models (or even nodes) for reaction wheels, thrusters, or an IMU sensor. The design of the notional system includes 4 reaction wheels, 16 thrusters, and an IMU to accomplish closed-loop control. The layout chosen for the reaction wheels and the IMU is shown in Figure G.3.1. The boresight location is also shown in Figure G.3.1 for reference. The reaction wheel angular momentum unit-vectors in the LUVOR-A body frame are given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0.5774 \\ 0 & 1 & 0 & 0.5774 \\ 0 & 0 & 1 & 0.5774 \end{bmatrix} h, \quad (G.3.1)$$

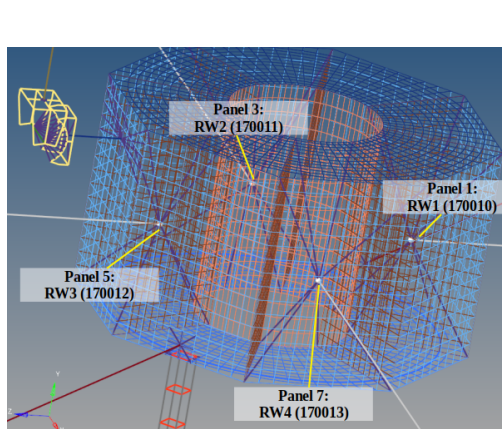
where the 3-vector H is the total reaction wheel angular momentum in the LUVOR-A body frame and the 4-vector h are the reaction wheel angular momentum components in the reaction wheel frames. The inertia of the reaction wheel bodies were selected as $20 \text{ kg} \cdot \text{m}^2$ to provide a reasonable amount of momentum storage for such a large spacecraft.

Thruster nodes were selected on each of the eight panels located on the bus compartment. Two thrusters per panel were assumed for a total of sixteen thrusters. The thruster force to LUVOR-A body frame transformation is given by

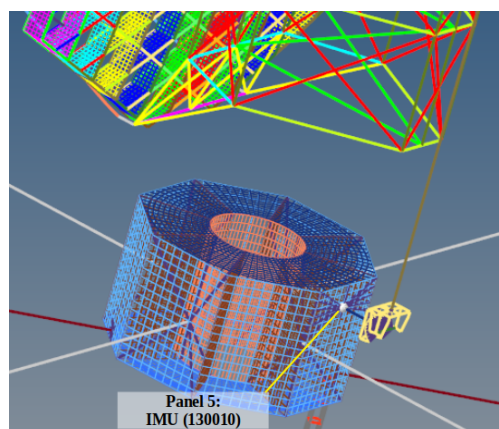
$$F^T = f^T \begin{bmatrix} -0.7071 & 0.5 & 0.5 \\ 0.7071 & -0.5 & 0.5 \\ -0.1464 & 0.5 & 0.8536 \\ 0.8536 & -0.5 & -0.1464 \\ 0.5 & 0.5 & 0.7071 \\ 0.5 & -0.5 & -0.7071 \\ 0.8536 & 0.5 & 0.1464 \\ -0.1464 & -0.5 & -0.8536 \\ 0.7071 & 0.5 & -0.5 \\ -0.7071 & -0.5 & -0.5 \\ 0.1464 & 0.5 & -0.8536 \\ -0.8536 & -0.5 & 0.1464 \\ -0.5 & 0.5 & -0.7071 \\ -0.5 & -0.5 & 0.7071 \\ -0.8536 & 0.5 & -0.1464 \\ 0.1464 & -0.5 & 0.8536 \end{bmatrix}, \quad (G.3.2)$$



(a) Boresight location



(b) Reaction wheel locations



(c) IMU location

Figure G.3.1. Actuator and Sensor Layout

where \mathbf{f} is a 16-vector of thruster magnitudes and the 3-vector \mathbf{F} is the total thruster force in the LUVOIR-A body frame.

G.4 Building the State Space Model Using FModal and Dshell

While FModal was used to obtain the relevant FEM model data for use in construction of a GN&C flexible body model, Dshell was used to assemble the flexible body dynamics, reaction wheels, thrusters, and IMU into a system-level simulation. Using the Dshell simulation, three linear state space models were generated with proportional modal damping ratios of 0.0025, 0.005, and 0.01 for delivery to The Aerospace Corporation. Figure G.4.1 shows the difference in computed modal frequencies between NASTRAN and the Dshell-generated state space models. The difference is on the order of the truncation error used to linearize the Dshell model. The format and description of the HDF5 delivery file is shown below for the linear state space model with 0.0025 damping ratio. The flexible body dynamics model encapsulated by the HDF5 delivery file was used in the GN&C simulation developed by The Aerospace Corporation to assess

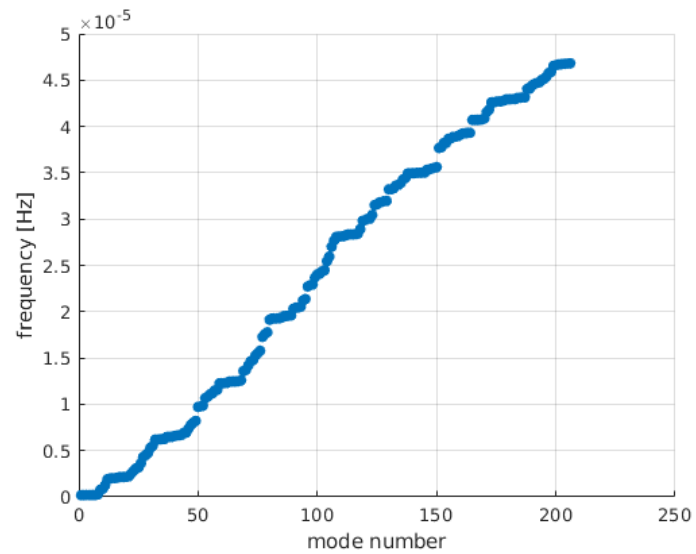


Figure G.4.1. Difference in Modal Frequency Between NASTRAN and Dshell Models

and compare system pointing stability performance between reaction wheel, colloidal microthruster, and cold-gas microthruster based architectures.


```

/ luvoira_linear_0p0025_06032019.h5 ... linear state space delivery file
├─ A ... state space A matrix ( 432 × 432)
├─ B ... state space B matrix ( 432 × 44)
├─ C ... state space C matrix ( 10 × 432)
├─ D ... state space D matrix ( 10 × 44)
├─ grid_positions ... grid positions relative to node 125
│   ├── 125 ... boresight node
│   ├── 123529 ... thruster 1,2 node
│   ├── 123654 ... thruster 3,4 node
│   ├── 123780 ... thruster 5,6 node
│   ├── 123906 ... thruster 7,8 node
│   ├── 124032 ... thruster 9,10 node
│   ├── 124157 ... thruster 11,12 node
│   ├── 123377 ... thruster 13,14 node
│   ├── 123403 ... thruster 15,16 node
│   ├── 130010 ... IMU node
│   ├── 170010 ... reaction wheel 1 node
│   ├── 170011 ... reaction wheel 2 node
│   ├── 170012 ... reaction wheel 3 node
│   └── 170013 ... reaction wheel 4 node
├─ input_index ... input matrix index map
│   ├── .rw1_dist_force ... force at RW1 node
│   ├── .rw1_dist_torque ... torque at RW1 node
│   ├── .rw1_motor_torque ... RW1 wheel torque
│   ├── .rw2_dist_force ... force at RW2 node
│   ├── .rw2_dist_torque ... torque at RW2 node
│   ├── .rw2_motor_torque ... RW2 wheel torque
│   ├── .rw3_dist_force ... force at RW3 node
│   ├── .rw3_dist_torque ... torque at RW3 node
│   ├── .rw3_motor_torque ... RW3 wheel torque
│   ├── .rw4_dist_force ... force at RW4 node
│   ├── .rw4_dist_torque ... torque at RW4 node
│   ├── .rw4_motor_torque ... RW4 wheel torque
│   └── .thruster_force ... thruster force inputs
├─ output_index ... output matrix index map
│   ├── .rw1_rate ... RW1 rate
│   ├── .rw2_rate ... RW2 rate
│   ├── .rw3_rate ... RW3 rate
│   ├── .rw4_rate ... RW4 rate
│   ├── .sn125_boresight_angRates ... inertial angular rates at boresight node
│   └── .sn130010_imu_angRates ... inertial angular rates at IMU node
├─ reaction_wheel_unit_vectors ... reaction wheel unit vectors in body frame
│   ├── 1 ... RW1 unit vector
│   ├── 2 ... RW2 unit vector
│   ├── 3 ... RW3 unit vector
│   └── 4 ... RW4 unit vector
├─ rigid_body_mass_properties ... rigid body mass properties about node 125
│   ├── mass ... total mass
│   └── cmoffset ... center of mass offset

```

```

└─ inertia ...      inertia      matrix
└─ state_index ...  state        space  A matrix  index  map
└─ .bus_modal_pos ... bus modal displacement dof
└─ .bus_modal_vel ... bus modal velocity  dof
└─ .bus_rotational_pos ... bus angular dof
└─ .bus_rotational_vel ... bus angular velocity dof
└─ .bus_trans_pos ... bus translational dof
└─ .bus_trans_vel ... bus translational velocity dof
└─ .rw1_angle ...   RW1 angle
└─ .rw1_rate ...    RW1 rate
└─ .rw2_angle ...   RW2 angle
└─ .rw2_rate ...    RW2 rate
└─ .rw3_angle ...   RW3 angle
└─ .rw3_rate ...    RW3 rate
└─ .rw4_angle ...   RW4 angle
└─ .rw4_rate ...    RW4 rate
└─ thruster_unit_vectors ... thruster unit vectors in body frame
└─ 1
└─ :
└─ 16

```

REPORT DOCUMENTATION PAGE					Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>						
1. REPORT DATE (DD-MM-YYYY) 09/29/2020		2. REPORT TYPE Technical Memorandum			3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Development and Verification of a Pipeline for Modeling Flexible Multibody Dynamics				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) VanZwieten, Tannen S.; Schutte, Aaron D.; Jain, Abhinandan; Simpson, Kimberly A.; Brouwer, Jeffrey R.				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER 860921.01.07.01.01		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199				8. PERFORMING ORGANIZATION REPORT NUMBER NESC-RP-18-01312		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001				10. SPONSOR/MONITOR'S ACRONYM(S) NASA		
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/TM-2020-5008164		
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category Space Transportation and Safety Availability: NASA STI Program (757) 864-9658						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT This project was proposed by Jet Propulsion Laboratory (JPL) personnel in response to a call from the NASA Engineering and Safety Center (NESC) Chief Engineer at JPL for discipline-advancing work. The work described in this NESC Final Assessment Report was requested to improve the efficiency and accuracy of analyses to support the dynamic modeling of complex aerospace systems.						
15. SUBJECT TERMS Dynamic Modeling; Flexible Multibody; Finite Element Model; NASA Engineering and Safety Center						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)	
U	U	U	UU	135	19b. TELEPHONE NUMBER (Include area code) (443) 757-5802	