Development of Zero-Gravity Structural Control Models from Analysis and Ground Experimentation

by

Roger M. Glaese

B.S., Aerospace Engineering
Virginia Polytechnic Institute & State University, 1991

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aeronautics and Astronautics

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February, 1994

Signature of Author _____
Department of Aeronautics and Astronautics
January 18, 1994

Certified by _____
Dr. David W. Miller
Principal Research Scientist, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by _____
Professor Harold Y. Wachman
Chairman, Department Graduate Committee

Development of Zero-Gravity Structural Control Models from Analysis and Ground Experimentation

by

Roger M. Glaese

Submitted to the Department of Aeronautics and Astronautics
on January 18, 1994, in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Aeronautics and Astronautics

Abstract

Future spacecraft will require high authority control because of higher performance requirements. This control will be active structural control because of higher payload mass fractions and overlap of the control with spacecraft bus and solar array dynamics. Prior to flight, no on-orbit experimental data is available, and consequently all models of the zero-gravity (0-g) behavior of the spacecraft must be derived from analytical methods. However, these analytical models tend to be inaccurate unless they have been correlated with experimental data. Ground tests of the spacecraft will be corrupted by the suspension system and gravity effects, making direct correlation of ground data with the analytical 0-g model difficult. Thus, a modeling approach must be developed for acquiring an accurate 0-g model, along with its error, prior to flight.

This thesis describes a modeling approach for deriving 0-g structural dynamic models, useful for control design, from a combination of analysis and ground experimentation. This approach relies on the Finite Element Method to derive a model of the one-gravity (1-g) behavior of a spacecraft. This finite element model is improved through comparison with ground data in a process called open- and closed-loop updating. Then, the suspension and gravity effects are removed to yield an updated prediction of the 0-g behavior. Since this model will still be in error when compared with flight data, estimates of these errors must be determined.

This approach is applied to the Middeck Active Control Experiment (MACE), an MIT and NASA space shuttle flight experiment intended to demonstrate high authority structural control in zero-gravity conditions. An input-output model, derived from the 1-g finite element model, is developed. It is found that parameters external to the finite element method contribute significantly to the error in the model. An update parameter selection criterion is presented which determines the uniqueness of the parameter sensitivity directions and their influence on the cost. It is found that the frequency-domain, logarithmic cost is best suited to the MACE problem. Automated updating results in a 28% reduction in this cost, with the remaining frequency errors being less than 3% in lightly damped modes. Closed-loop experiments using this model have achieved a performance improvement of 24 dB.

Thesis Supervisor: Dr. David W. Miller
Principal Research Scientist, Department of Aeronautics and Astronautics

Acknowledgments

I would like to thank my friend and advisor, Dave Miller, for his guidance and support through the course of my research.

I would also like to thank the members of the MACE team for their insights and suggestions. In the control group, Mark Campbell, Simon Grocott, and Dr. Jonathan How and in the identification group, Dr. Ketao Liu.

A special thanks goes to Erik Saarmaa. Without your help and excellent initial models, this thesis would have taken much longer and been much harder.

Another special thanks goes to Dan Rey, whose insights helped tremendously.

I would also like to thank all my friends in SERC, who make working here a pleasure. A special thanks goes to the Ear crowd, who help keep me sane in this pressure cooker.

Thanks goes to my roommates, past and present. Mark, Carl, Simon, thanks for putting up with me.

A special thanks goes to Brett. Thanks for being my best friend.

Most importantly, I would like to thank my parents for allowing me the freedom to choose my own path in life and the thirst for knowledge that makes anything possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Future spacecraft, in order to meet stringent performance specifications, will require active structural control. In order to design controllers for these spacecraft, a good understanding of the structural dynamics is required. Prior to flight, however, these controllers may only be analyzed and tested on the ground. Ground testing necessitates suspending the spacecraft in order to approximate the conditions of flight. However, this suspension system, along with other gravity effects, will couple with the dynamics of the structure. This coupling will corrupt the ground tests making verification of the flight controllers difficult at best. In order for the designers of these spacecraft to have confidence that the controllers will work satisfactorily once the spacecraft in orbit, a good prediction of the eventual zero-gravity (0-g) behavior of the structure is needed prior to launch. The objective of this thesis is to develop and demonstrate an approach for developing a predictive model of 0-g behavior, and an estimate of its uncertainty, which has been updated through analysis and correlation with one-gravity (1-g) open- and closed-loop test results.

Obtaining this predictive model of the on-orbit behavior of a structure requires a modeling approach that combines both analytical modeling techniques with ground based experimentation. Such a modeling approach, proposed in Chapter 2, combines the analytical power of the Finite Element Method [1, 2, 3] with advanced structural dynamic testing and model update methods. In this modeling approach, the Finite Element Method is used to create a 1-g finite element model of the suspended

structure. This 1-g finite element model consists of the free-free structure, to which a model of the suspension system and gravity effects are added. Since the finite element method does not match the input-output behavior of the test article (*i.e.*, the behavior as seen by the controlling computer), the finite element model must be augmented. This necessitates the detailed modeling of the actuators and sensors, both mechanically and electronically, as well as any signal conditioning electronics included in the control hardware. Since many spacecraft with pointing payloads have local servo loops implemented on these payloads, any servo loops included on the structure must also be captured in this 1-g finite element input-output model.

Once the 1-g input-output model has been developed, it is compared with experimental data. Inevitably, some errors will exist. Thus, the correlation of the finite element model with the data must be improved through a process called updating. The first phase of the updating process is to examine the model for errors in the model parameters and inconsistencies in the modeling assumptions. This examination is called engineering insight, because the engineer relies on experience to make the judgments necessary to improve the model. Also included in engineering insight is the measurement of all possible model parameters: including lengths; masses; and rotary inertias.

The next phase in the process is called automated updating, because the model/data correlation relies on optimization techniques to update the physical parameters. Physical parameters represent the physical attributes of the structure, such as mass, stiffness, length, *etc.* The optimized parameters must be physical in order to stay within the framework of the finite element method. This allows the suspension and gravity effects to eventually be removed from the updated model. If the modal parameters of the model were updated, this would jump outside of the finite element method and there would be no way to extract the 0-g behavior of the structure.

Thus far in the approach, the model has been updated based upon some open-loop criterion. However, the finite element model will eventually be used to design control for the spacecraft in 0-g. Some large open-loop errors may have little impact on the stability and performance of the closed-loop system, while others might have

serious implications. Therefore, it would be beneficial to rescale the importance of the various errors from a closed-loop perspective. This can be achieved by evaluating the performance of controllers designed using the 1-g finite element model and implemented on the suspended hardware. Assuming that bandwidths and performance levels are comparable to those expected on orbit, 1-g model errors that require the design of robust control will most likely pose robustness problems on orbit. Such errors would warrant further update. In this manner, the finite element model can be closed-loop verified prior to flight.

Once the 1-g finite element model has passed through the open- and closed-loop updating phases, it should match the open-loop experimental data quite closely. Controllers designed using the updated model should achieve performance comparable to that achievable using a measurement model [4]. To arrive at predictions of the 0-g behavior of the structure, the suspension and gravity effects are analytically removed from the updated 1-g finite element model. The resulting finite element model is simply a model of the free-free structure. This 0-g model differs from the initial free-free model in that it has been updated to better match the ground experimental data. Assuming that the suspension and gravity effects were accurately captured in the 1-g model, this resulting 0-g model should provide accurate predictions of the orbital behavior of the structure.

Regardless of how accurate the 1-g model is, the resulting 0-g model will still be in error. All that the 1-g updating has done is to reduce the size of these remaining errors. High authority control will drive the closed-loop system unstable if these modeling errors are not taken into account in the design. If, however, an estimate of the errors can be determined, robust control techniques can derive controllers that will be stable even in the face of the anticipated modeling errors. Thus, a very important part of the proposed modeling approach is the determination of this uncertainty model. Since no on-orbit experimental data exists prior to flight, the uncertainty estimate must be made using knowledge of the residual errors in the 1-g finite element model and knowledge of how the dynamics of the structure change from 1-g to 0-g.

Thus, the result of the proposed modeling approach is a 0-g model that is suitable for control design and implementation on the structure in flight. This 0-g model consists of a nominal model and estimates of the remaining uncertainties in the model. The nominal 0-g model is a finite element input-output model that has been updated using ground experimental data and verified through closed-loop experimentation on the structure suspended in 1-g. The uncertainty estimates for this nominal model are derived from an extrapolation of the 1-g uncertainty estimates into 0-g.

In order to demonstrate the details of the proposed modeling approach, it will be applied to the Middeck Active Control Experiment (MACE) test article [5, 6]. MACE is a NASA In-Step/Control-Structure Interaction Office funded MIT space shuttle flight experiment. The goal of MACE is to develop a set of Controlled Structures Technologies (CST) that give designers of future spacecraft, which cannot be adequately tested on the ground, confidence in the eventual on-orbit performance. The methodology used in MACE is to design a flexible structure that can be closed-loop tested both on the ground and in orbit in the middeck of the space shuttle. The test article is designed to couple strongly with the suspension system during ground testing. This coupling, along with the performance specifications, make testing of flight controllers on the suspended test article impossible and also represents a real challenge for the proposed modeling approach. To test the end result of the proposed modeling approach, one phase of the flight testing is to implement controllers designed from models of the predicted 0-g behavior of the test article before any flight experimental data is available.

This thesis builds on previous work done at MIT's Space Engineering Research Center (SERC) on the modeling of controlled structures. Rey's work on the identification of suspension and gravity effects on controlled structures [7] forms the theoretical basis for the inclusion of these effects into a 1-g finite element model. In his work, Rey identified five primary categories of suspension and gravity effects: suspension effects; geometric stiffening effects on the structure; initial deformation effects on the structure; gravity effects on sensors and actuators; and nonlinear suspension and gravity effects. The first three of these effects are included in the MACE 1-g finite

element model. Due to the nature of the sensors and actuators used on the MACE test article, the gravity effects on sensors and actuators are not needed. Nonlinear gravity effects are not treated due to their complexity and the difficulty of including them in a linear model.

Barlow's work on the modeling of the structural test article of MIT SERC's previous flight experiment, the Middeck 0-gravity Dynamics Experiment (MODE), serves as a guideline for the development of a 1-g finite element model and predicting 0-g behavior of a structure based on that finite element model. This work differs from Barlow's in several aspects. For MODE, the structural dynamics of interest were only the first few structural modes. For the MACE test article, the bandwidth of interest consists of the structural modes up to 250 Hz, encompassing nearly 35 modes in 0-g, including the rigid body modes. A second difference is in the complexity of the models required for the two experiments. For MODE, only a simple input-output model was required. For MACE, the input-output model becomes quite complex due to the diversity of actuators and sensors present on the test article, the addition of various signal conditioning electronic filters on the sensor outputs, and the local servo loops closed on the structure to provide low bandwidth pointing control of the multiple payloads. Although Barlow formed a 1-g finite element model for MODE, he made no attempts to adjust this model to better match the ground experimental data. The goal of the MACE program, on the other hand, requires this adjustment based on ground experimental data in order to improve the predictions of the 0-g behavior of the test article.

The final set of work, that forms the basis for this thesis, is that done in the area of finite element model updating. Updating finite element models has been around nearly as long as the finite element method, and is a widely practiced skill in industry. Various commercial updating codes are available [8, 9]. At SERC, the most recent work was performed by Balmès [10] who worked with hybrid measurement/finite element modeling techniques. Although not used directly in this thesis, his work provided inspiration for finite element model updating.

Chapter 2 outlines a general modeling approach that can be used to develop a

high-precision 0-g model that is useful for control design. This modeling approach relies on the finite element method to analytically capture the suspension and gravity effects in a 1-g model. Once the 1-g model is formed, it is improved using open- and closed-loop ground tests to arrive at an 'updated' 1-g model. The suspension and gravity effects that are present in the ground tests are then turned off analytically using the finite element method to form a model of the 0-g behavior of the structure.

Chapter 3 traces the development of the initial 1-g model for MACE. Since this model will be improved using experimental results, the frequencies and modeshapes that result from the finite element method must be augmented to match the input-output behavior of the test article. The frequencies and modeshapes are augmented with with experimentally determined damping. Other parts of this augmentation include adding mechanical and electronic models of the actuators and sensors such that their inputs and outputs are in the proper units: volts. Any additional signal conditioning filters, as well as any local servo controllers present on the test article, are also included in this model.

Chapter 4 discusses the model updating effort directed at improving the correlation of the 1-g finite element input-output model with ground experimental data. Generally, the model is updated using engineering insight and automated updating techniques. Engineering insight refers to the changing of model parameters and assumptions based on engineering judgment rather than some optimization scheme. Automated updating, on the other hand, uses optimization techniques to make improvements in the model. Since the 0-g models developed using the proposed modeling approach will be used for control design, it would be highly desirable to verify the 1-g model through closed-loop ground tests. Problems that arise during this closed-loop experimentation can be used to indicate further improvements in the finite element model that may be necessary.

Chapter 5 is the culmination of the proposed modeling approach, the derivation of the 0-g model from the updated 1-g model. This derivation consists of analytically turning off the suspension and gravity effects in the finite element model and adding the input-output aspects of the structure. This 0-g finite element input-output

model will be in error when compared to flight experimental data, and consequently controllers designed using the 0-g model are likely to be unstable. Robust control techniques can compensate for these model errors if a model of the uncertainty is known. For this reason, a model of the uncertainty present in the 0-g input-output model must be derived along with a nominal 0-g model.

Chapter 2

Modeling Approach

The objective of the modeling effort for MACE is to provide a model with sufficient accuracy to design high authority control of the test article for implementation in zero-gravity. Two methods are used to generate dynamic models for structures, models based exclusively on experimental data, measurement models [4], or through the use of an analytical procedure such as the Finite Element Method [1, 2, 3]. Generally, measurement models are more accurate than finite element models because they are fit to the same experimental data that is used to assess accuracy. The finite element model, on the other hand, is derived from the material and geometric properties, the so-called physical properties.

While this would seem to suggest that there is no point in modeling structural dynamics via the Finite Element Method (FEM), measurement based models do have their limitations. The FEM is useful for creating a model before hardware exists, investigating sensor/actuator placement, investigating the impact of component modifications, and modeling something in an environment which is different from that in which it is tested. The last of these is particularly important for spacecraft, where all preliminary testing must be conducted on the ground, necessitating a suspension system which introduces gravity coupling. Since spacecraft will eventually be used in the zero-gravity environment, where ground based measurement models are invalid, a finite element model must be constructed in order to predict the on-orbit behavior of the structure. This model is necessary for developing *a priori* confidence that the

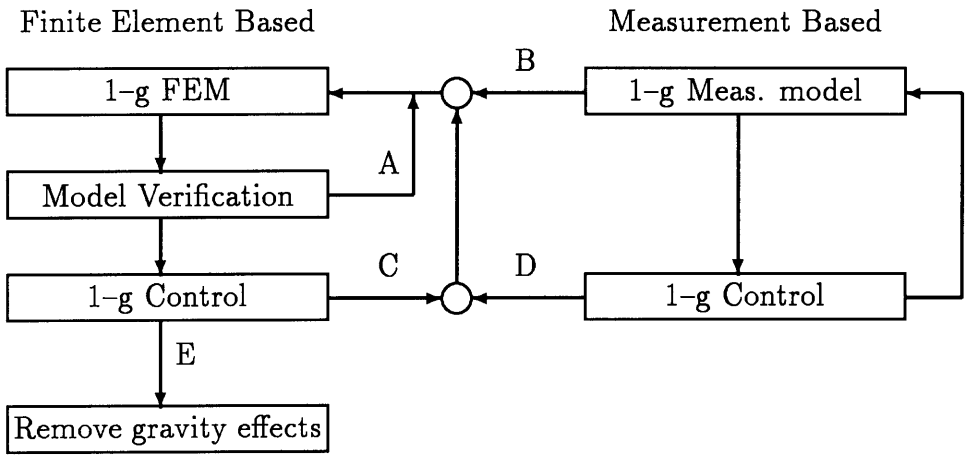


Figure 2.1: Modeling Approach for the Middeck Active Control Experiment

controlled structure will meet performance specifications in its operational environment. Once on orbit, a measurement model can be constructed for further open-loop analysis.

The proposed modeling approach for MACE [11] is summarized in Figure 2.1. As indicated in the figure, this approach consists of both open- and closed-loop testing on the ground using models derived from both the finite element method and measurement based techniques. The first step in this approach is the creation of a 1-g model which allows updating using ground test results. Since the model is to be used for control design, this approach updates the model using not only open-loop ground test results but also closed-loop results. If modal parameters were used in the updates, there would be no way to remove the suspension and gravity effects to form a 0-g model. For this reason, physical parameters of the finite element model are used in the updates. The following sections describe each step of this approach in detail.

2.1 Initial 1-g model

The first step in the modeling process, detailed for MACE in Chapter 3, is to obtain an initial 1-g model of the input-output behavior of the structure. This model consists of the structure as it would behave in 0-g, the so-called free-free model, along with a model of the suspension system and gravity effects. To form the input-output model,

the mechanical and electronic attributes of the actuators and sensors are added, including signal conditioning filters. Since typical spacecraft may have local pointing control on the attached payloads and instruments, the local control aspects of the structure, including digital implementation of the controls, must also be included in the model.

The initial free-free finite element model is based upon engineering drawings for dimensions, material handbooks for elasticity and density properties, and measurements of the actual hardware components, if they exist. The nodal points used in the finite element model are taken from engineering drawings. Most of these nodal points are distributed along the flexible parts of the structure to provide resolution of the flexible modeshapes. Other nodal points are located at the centers of mass of components that are complicated or have large mass. The last type of nodal points are located at the sensor and actuator locations, to provide modeshape components for the model input and output matrices. Once the nodal points have been defined, the interconnecting mass and stiffness properties must be defined.

Stiffness is given to the structural model by connecting the nodal points with elastic finite elements. Depending on the configuration of the component, the element may be a truss, beam, plate, shell, *etc.* While requiring different specific parameters, all elements require material and cross-sectional properties. Typical material properties are the Young's modulus, shear modulus, and Poisson ratio. The cross-sectional properties allow the calculation of areas and area moments of inertia. These material and cross-sectional properties are initially taken from handbooks and engineering drawings, respectively, but are eventually updated by measurement of the actual hardware.

Two types of mass may be used in the finite element method: distributed and concentrated. For distributed masses, the finite element code computes the mass matrix for an element from a given material density, initially taken from handbooks and eventually updated with component measurements, and cross-sectional shape. Distributed mass is typically used for very simple components where the cross-section and material density are easily described, such as truss members, beams, and plates.

Concentrated mass is generally reserved for more complicated components, where the cross-section changes radically along the element. For concentrated masses, the total mass and rotary inertias of the component are lumped at the center of mass. Since the model may be constructed prior to hardware fabrication, the masses, rotary inertias, and centers of mass must initially be computed analytically. For simple components, these computations may be carried out by hand. But for more complex components, a solid body modeling program, such as I-DEAS [12], may be used to compute the mass, rotary inertias, and center of mass, using the component geometry and material properties. Once hardware exists, the concentrated mass properties can be verified by component measurement. Concentrated masses are also used for small, rigid components where the dimensions are small when compared with the nodal spacing.

Once the initial free-free finite element model has been constructed, a model of the suspension system and gravity effects must be added to arrive at the initial 1-g finite element model. The model of the suspension system can be simple or complex depending on the level of fidelity desired in the model. If only the bounce and pendular behaviors of the suspension system are desired, the suspension model might simply consist of springs and rigid suspension cables. For higher order suspension dynamics, such as suspension violin and compression modes, the suspension model would consist of springs, hinges, and beam element suspension cables with the proper mass and stiffness properties.

Once the suspension system has been modeled, the direct effects of gravity on the structure need to be included in the model. The structural gravity effects include pre-deformation, pre-loading of the members, stiffening of rigid body behavior, and pendular effects on articulating payloads [13]. The structural gravity effects are captured through a nonlinear stress-stiffening procedure in the finite element code. In this procedure, the gravity load is incrementally applied to the structure and the resulting stresses are used to modify the stiffness matrix using the geometric stiffness matrix. The resulting modified stiffness matrix is used, along with the unchanged mass matrix, to generate the 1-g modal frequencies and modeshapes. These 1-g

modal frequencies are combined with arbitrary damping ratios, initially taken to be 1%, to form the finite element system matrix. Gravity also affects the behavior of proof-mass actuators and accelerometers and can be captured in the model input and output matrices.

The final step is to add the actuation and sensing capacities of the structure. These are added through the input and output matrices, which are formed by selecting the proper combination of modeshape components. For a point force or moment input, or displacement, rate, or acceleration output, the value of the proper modeshape component at the actuator or sensor nodal point is used. For a relative force or moment input, or relative displacement or rate output, the values of the proper modeshape components at two collocated actuator or sensor nodal points are differenced to provide the relative input or output. The resulting matrices make up a state-space model of the structure in which the actuator inputs are in their physical units, such as Newtons and Newton-meters, and the outputs are in their physical units, such as meters, radians, and strain (*i.e.*, meters/meter) and their temporal derivatives. Since the input and output units are in physical units, this model is called the physical state-space model of the system.

Although this model does provide the correct input-output structure of the system, the model is to be used for control design, and must therefore match the input-output behavior of the hardware as seen by the controller. Thus this physical model must be transformed into a so-called volts-volts model, in which the inputs and outputs are both in volts.

Actuators and sensors rely on some physical mechanism to convert the electrical signal to a physical force, or *vice versa*. In the case of relative moment actuators, such as a gimbal or reaction wheel, a motor and power amplifier are used to convert the applied voltage to an applied torque. For an active strut, a piezo-electric material is used to convert the applied voltage to an applied strain that induces bending in the strut. The sensors behave oppositely in the sense that the proper structural displacement, rate, acceleration, or strain is converted to a voltage.

Relationships can be written between the physical unit sensed or actuated and the

voltage measured or applied. Often, the electrical signals of the actuators or sensors are amplified to provide ease of measurement or actuation. The hardware gains of the actuators or sensors are combined with any amplification factors to make up the total gains, which are applied to the physical input and output matrices to convert them to volts.

Since the actuators and sensors rely on physical mechanisms to provide the conversion between the physical and electrical domains, these actuators and sensors may exhibit resonances and other dynamic effects. The input and output signals may also be passed through electronic filters to provide better signal conditioning (noise filtering). These filters may be high or low-pass filters, anti-aliasing filters, such as a Bessel filter, or notch filters to reduce the effects of sensor dynamics. Since the control computer sees the structure through these filters, they must be included in the model. These filters may be accounted for in either a coarse or detailed fashion. If the filter dynamics are sufficiently high in frequency, the effects of the filter may be approximated by including the DC-gain of the filter and an equivalent time delay to account for the phase lag. If the sensor, actuator, or filter dynamics are within the frequency range of interest, detailed state-space descriptions must be included in the model.

If any initial controllers, such as pointing servos, are implemented on the structure, they must be included in the model as well. If these controllers are implemented using analog circuitry, the loop is closed between the proper sensors and actuators using the control gains. Some of these controllers may be implemented using a digital computer. In order to stay in the continuous time domain, the model must account for the sampling and computational delays of the computer. The most convenient way of including these time delays is through the use of a Pade approximation [14], which is appended to all controlled actuators. The servo loop is then closed around the model containing these time delays.

The resulting finite element model includes the free-free structure, a model of the suspension system, and gravity effects. The input-output behavior of the structure is captured through models of the sensors and actuators, including sensor and

actuator dynamics, electronic gains, and electronic filters. Once the general input-output behavior has been captured, any preliminary control, either analog or digital, is captured, including any computational and sampling time delays appropriate for the application of control. This results in an initial prediction of the 1-g behavior of the structure.

2.2 Model Updating

Once the initial 1-g finite element input-output model has been created, the rest of the modeling approach is geared toward improving the model and deriving a 0-g model from the improved 1-g model. This process of improving the model is called updating. In the proposed modeling approach, model updating occurs using both open- and closed-loop testing in 1-g, detailed for MACE in Chapter 4.

2.2.1 Open-loop updating

The first step in the update process is to verify the model (Step A in Figure 2.1) by measuring modeled properties on the hardware and verifying modeling assumptions. Experimental data may also be used to provide insights into errors contained in the model. As an example, comparison of analytical predictions of the transfer functions with experimental transfer functions is used to identify errors in the model and suggest improvements. This updating, called engineering insight, comes from the experience the engineer brings to the process. Once the model has been verified through engineering insight, it can be further refined using optimization techniques, called automated updating.

Engineering insight, often little more than an educated guess, covers all aspects of the modeling and updating process and is used to make coarse adjustments to the model. The first step is to carefully examine the finite element input deck and post processing code for any typographical errors. The second step is to obtain as much information as possible about the components of the structure without getting into the input-output behavior. This step includes measuring component masses, centers

of mass, rotary inertias, lengths, and cross-sections of assembled components. It also includes measuring the stiffness properties of components, if possible, and measuring electronic and control properties, such as hardware gains, filter transfer functions, and computer time delays. Once this step is complete, the finite element mass matrix should be correct and all electronic and control aspects of the structure should be known.

All remaining errors in the model should be due to erroneous modeling assumptions, errors in the stiffness parameters of the model, and nonlinearities in the structure. The predictions of the 1-g finite element input-output model are compared to the ground experimental data. Regions of large error are examined to see if erroneous modeling assumptions could be causing the error. An example of this type of insight is the examination of the beam element formulation in the model. Perhaps there are large frequency errors in the higher frequency modes. Their analytical modeshapes can be examined for large amounts of shear. If shear is seen, the beam element formulation should be changed from Bernoulli-Euler to Timoshenko.

In automated updating, the computer is allowed to make changes in model parameters, such as the stiffnesses of the beam elements, to minimize some cost based either on measured modal parameters or the raw experimental data. Engineering insight is used to select the update parameters, define the cost, and place bounds on the extent to which the parameters will be allowed to vary. The result of automated updating is a model that is more closely correlated to the experimental 1-g data.

Closely associated with model updating is the development of an input-output model based exclusively on the ground experimental data, called a measurement model. The measurement model is useful because it represents the best approximation to the data that is possible using a finite order, linear model because all parameters in the state-space model are used to fit the data. Using the measurement model, estimates of the modal frequencies and damping ratios can be obtained, which can be used to update the finite element model (Step B in Figure 2.1) [9].

It is interesting to note that the finite element model can also be used to improve the measurement model by indicating the minimum order of the model, as

measurement models are prone to retain multiple modes to represent a single mode that has slightly different frequencies depending on which input/output channel is measured. The finite element model can also be used to determine whether lightly damped transfer function zeros are minimum or nonminimum phase. This is a problem because these zeros have low magnitude and are often below the noise floor of the sensor causing the data to possibly indicate a nonminimum phase zero where the finite element model indicates a minimum phase zero.

The raw experimental data can also be used to update the model. In this case, the model predictions are compared using some cost function. This type of updating is desirable because of difficulties encountered in matching the modal parameters of a measurement model. These problems are closely spaced and repeated modes. In these cases, the wrong finite element modes may be paired with the measured modes causing a poor update. Using the raw data eliminates these problems because the modes will naturally sort themselves out to pair with the proper experimental modes.

In theory, it would seem that a progression from initial engineering insight, through matching the frequencies and damping of a measurement model, to automated updating based solely on the experimental data would give good results. In practice, though, this is hardly the case. Often, coarse engineering insight adjustments come about because of difficulties encountered during automated updating, the model error simply will not budge using the chosen parameters. In this case, engineering insight must be used to verify that some erroneous modeling assumption is not dominating the errors or that the chosen parameters are sufficient to allow a good update.

2.2.2 Closed-loop updating

The eventual goal of this modeling approach is to derive a 0-g input-output model of the structure useful for control design. Because of this, it is advantageous to verify the model in closed-loop before flight. Thus, the second half of the update process is based on closed-loop results. Two avenues for updating are exploited in closed-loop updating: implementation of controllers based on the 1-g finite element model and comparison of these closed-loop results with measurement based control results.

In comparing the finite element model to open-loop experimental data, all the errors in the model are shown in detail. To examine which of these errors are most important for closed-loop experiments, controllers are designed based on the 1-g finite element model. These errors will cause the actual performance to differ from predictions, possibly leading to instability, when the controller is implemented on the hardware (Step C in Figure 2.1). The closed-loop results cannot, as yet, be used to explicitly update the model. Rather, the closed-loop results indicate which modes went unstable first, and thus, which modes should be emphasized in further updates.

The closed-loop updating step could continue *ad infinitum*, as long as a controller of high enough authority leads to instability. At some point, though, it is desirable to end the update process and test the structure on orbit. Thus, some test of the quality of the finite element model is needed. As stated previously, the measurement model is the best finite order, linear approximation to the data, and as such, controllers designed using this model should give the best closed-loop performance. Consequently, a good test of the quality of the finite element model would be to compare the closed-loop finite element results with those of the measurement model (Step D in Figure 2.1). This comparison could be done in two ways: direct comparison of the measurement and finite element based control results and comparison of the finite element based control results with a standardized design curve.

The first method of comparison would simply be to compare the performance improvement achieved by controllers designed using the two models. In order to provide a fair comparison, though, both measurement and finite element results should be derived from controllers with the same control authorities, topologies, and design techniques. Differences in any of these areas would cause the comparison to be unfair and not provide the proper insight. The way around this leads to the second method of comparison.

This method eliminates these problems by standardizing the comparison in the form of a design curve. To form this curve, the closed-loop cost, computed based on the nominal model, is plotted against the amount of control used for some set of 'standard' weightings, topology, and control design technique. This design curve is

derived using the measurement model in order to minimize the number of times the curve must be recomputed and to make the measurement model results the basis for comparison with the finite element results. With the design curve in hand, both the measurement and finite element based experimental control results are compared with the design curve. The measurement based results are plotted to give an indication of what is actually possible, as even the measurement model is in error and will lead to instability at some level of control authority. The finite element based control results are compared to the design curve to determine how far away the finite element model is from the theoretical best, and compared to the measurement results to determine how far away the finite element model is from the achievable best.

These comparisons of the finite element-based closed-loop results with the measurement model-based closed-loop results, although not directly used in updating the finite element model, provide an indication of when the finite element model is 'good enough' and updating can be terminated. This final 1-g model very closely matches the 1-g open-loop experimental data and has been closed-loop verified such that it achieves good performance when compared with measurement based controllers. The next step is to form the 0-g model.

2.3 0-g Prediction

To form the 0-g model, detailed for MACE in Chapter 5, the final 1-g model is used as a starting point. Since the transformation from 1-g to 0-g is not known for the measurement model, because the gravity effects are buried within the modal parameters, the finite element method must be used to generate the 0-g model. In the 1-g finite element model, the suspension system is removed and the gravity effects are analytically turned off, leaving simply the updated free-free model.

From this point, the development of the 0-g model closely parallels the development of the initial 1-g model. The 0-g system matrix is formed by taking the 0-g frequencies and the damping ratios for the corresponding modes from the 1-g model. The 0-g input and output matrices are formed in the same fashion as in the 1-g model,

except using the 0-g modeshapes. If the ground test electronics are the same as the flight hardware, the same electronic gains and filters are used in the 0-g volts-volts model. If different electronics are used on the ground and for flight, the appropriate gains and filters for the flight electronics are used to form the volts-volts model. The appropriate flight controllers are also included in the input-output model. The resulting model is a nominal model of the structure as it will behave in 0-g.

Controllers designed using modern control theory do not tolerate model errors very well, leading to instability at low control authority. If model errors are known, however, robust control theory allows the control designer to take these errors into account, allowing controllers of much higher control authority to be stable. For this reason, it is desired to have some knowledge of what errors are likely to be present in the model when compared with flight experimental data. Since flight data does not exist, this 0-g error model must come from a combination of the remaining errors present in the 1-g model and possible errors in the transformation of the model from 1-g to 0-g. Probably the most important piece of information in the 0-g prediction is the generation of this measure of uncertainty. Thus, the final result of this modeling approach is a nominal model of the on-orbit behavior of the structure and a measure of how the model is likely to be in error when compared with flight experimental data.

Chapter 3

1-g MACE Model Development

The modeling approach described in the previous chapter will be applied to the Middeck Active Control Experiment (MACE) test article. The MACE program calls for three generations of hardware: the Development Model (DM) to test preliminary hardware designs, the Engineering Model (EM) to finalize flight hardware designs, and the Flight Model (FM) which will actually fly on the shuttle. In order to be most representative of the Flight Model, the modeling approach will be applied to the EM hardware, which has undergone extensive testing and analysis.

The first step in the modeling approach is the development of an initial 1-g input-output model of the MACE structure. This chapter details the development of the initial 1-g model for the Engineering Model version of the MACE hardware (Figure 2.1).

3.1 Hardware Description

The MACE EM test article, illustrated in Figure 3.1, consists of a segmented straight tubular Lexan^{TM} bus with a two-axis pointing/scanning payload at the right end. A second two-axis gimbal, considered the disturbance source, is mounted on the left end. Each two-axis gimbal allows actuation of rotation about the X- and Z-axes via two DC torque motors. A reaction wheel assembly is attached at the center node to provide attitude control torques. The reaction wheel assembly is comprised of

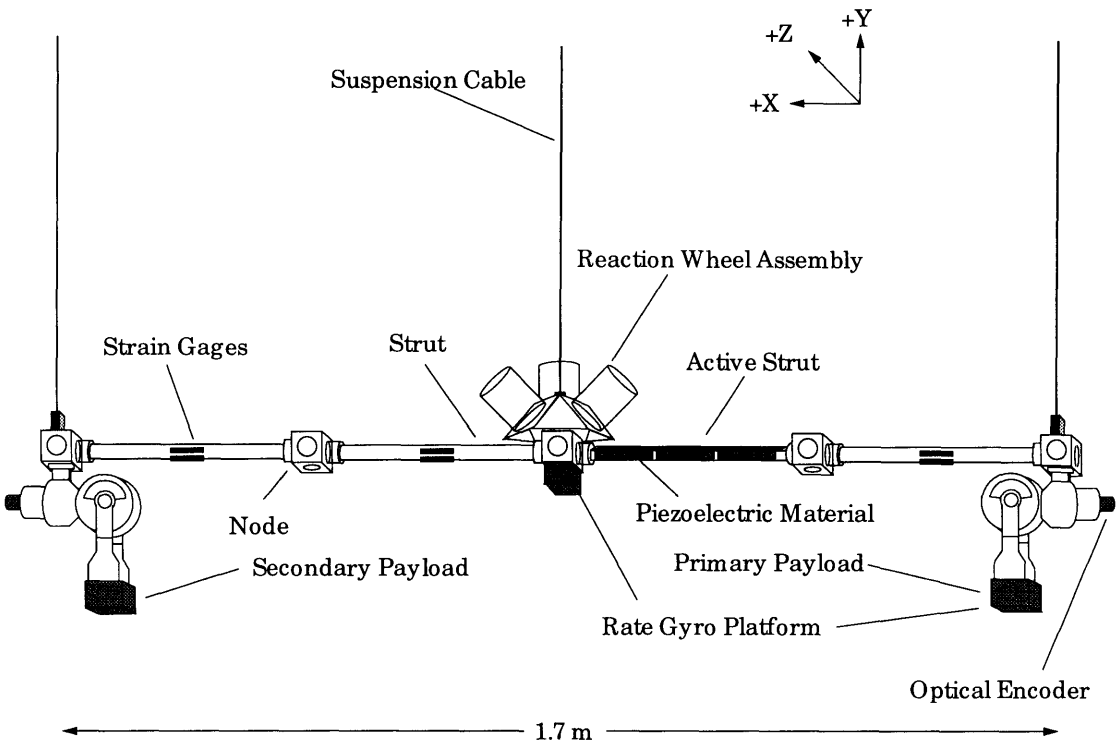


Figure 3.1: Middeck Active Control Experiment (MACE) Engineering Model test article suspended in 1-g

three orthogonally mounted DC servo motors with an inertia wheel mounted on each. Speed-control of the reaction wheel assembly is performed to attenuate the effects of friction and to avoid wheel runaway. One segment of the structural bus consists of an active member, which allows the actuation of bending strain through the use of piezoelectric materials. The hardware is suspended using three pneumatic/electric low frequency suspension devices [15].

The test article is outfitted with a variety of sensors used for control and identification. Each gimbal axis has a laser rotary encoder to measure relative angular rate of the gimbal. A two-axis rate gyro platform is mounted in the payload of the primary gimbal, providing measures of inertial X- and Z-axis angular rates. A three-axis rate gyro platform is mounted beneath the reaction wheel assembly, providing measures of inertial X-, Y-, and Z-axis angular rates. A tachometer is mounted on each reaction wheel motor to provide a measurement of relative wheel speed used in the speed-control servos. Eight strain gage pairs (2 per strut) are used to provide a

measurement bending strains about the Y- and Z-axes . Anti-aliasing Bessel filters are included on all analog sensor channels.

Since many spacecraft use low frequency servo control systems to point attached payloads, such a servo control is also used on the MACE hardware. All four gimbal motors have proportional-integral servo loops closed around them, using the encoder rate and integrated rate signals for feedback. The result is that below a frequency of 3 Hz, the input signal to each gimbal commands the relative angular position of the payload with respect to the bus. These servos and the structural control algorithms are implemented on a real-time computer operating at a 500 Hz sampling rate. The computational delay, zero-order hold, and Bessel filters introduce a substantial net time delay into the transfer functions necessitating their incorporation into the 1-g model.

The first step in deriving the initial input-output model is development of a 1-g finite element model, consisting of the free-free model to which a suspension model and gravity effects are added. The resulting 1-g frequencies and modeshapes are used to form a physical input-output model by including models of the actuators and sensors. This model is physical because the inputs and outputs are in physical units. Since the control computer inputs and outputs are in volts, the electronic aspects of the structure, such as hardware gains, amplifiers, and signal conditioning filters, are added to the physical input-output model to transform it into a volts-volts model. The reaction wheel speed control and gimbal pointing servos are added to the volts-volts model to complete the formation of the initial 1-g input-output model.

3.2 Initial Free-Free Model

The initial free-free EM finite element model is generated using a combination of engineering drawings, material properties from handbooks, and experience gained from previous versions of the hardware. Nodal points are distributed across the flexible portion of the structural bus to provide spatial resolution of the bending modeshapes. (In order to distinguish between the structural nodes and the finite

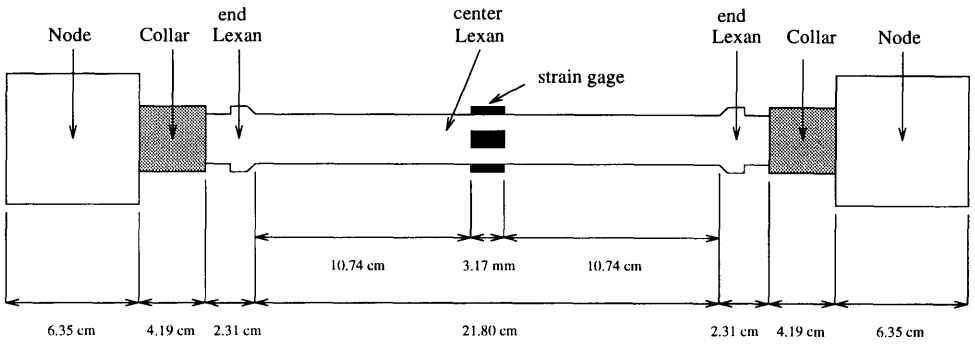


Figure 3.2: Passive strut dimensions

element nodes, the structural nodes will be referred to simply as 'nodes' and the finite element nodes as 'nodal points'.) Some of these nodal points are located at component connections, such as the node-collar and the collar-strut interfaces, see Figures 3.2 and 3.3. Other nodal points are located at the centers of mass of components with large mass, such as the nodes, gimbal stages (Figure 3.4), reaction wheel assembly stages (Figure 3.5), and rate gyro packages (Figure 3.6). The centers of mass for these components, in their relative coordinate frames from Figures 3.4, 3.5, and 3.6 are given in Table 3.1. All centers of mass are referenced from the node interface (*i.e.*, where the component attaches to the structural node). The table shows both the payload rate gyro package ("Gimbal Payload") and the dummy rate gyro package ("Dummy Payload"). For the primary gimbal, attached to the right end of the test article in Figure 3.1, the payload rate gyro package is used, while for the secondary gimbal, rotated 180° from the primary gimbal and attached at the opposite end of the test article, the dummy rate gyro package is used. The center of mass of the node, which is a 6.35 centimeter cube (see Figure 3.2) is assumed to be located at the center of the cube.

The last set of nodal points are located at the actuators (*i.e.*, the gimbals and reaction wheels) and sensors (*i.e.*, the rate gyros, encoders, tachometers, and strain gages). For actuators and sensors whose action is a relative torque or rotation about a single point (*i.e.*, everything but the rate gyros, active strut, and strain gages) two nodal points are superimposed. The free rotation is defined using constraint equations. The encoder-gimbal and tachometer-reaction wheel sensor-actuator pairs

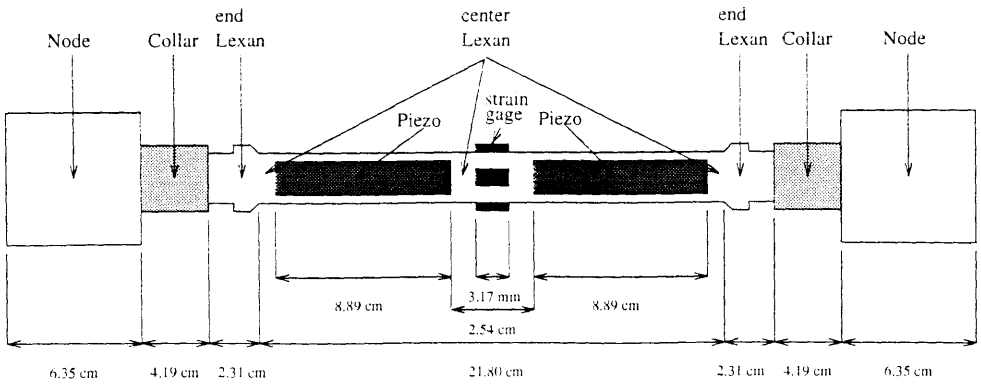


Figure 3.3: Active strut dimensions

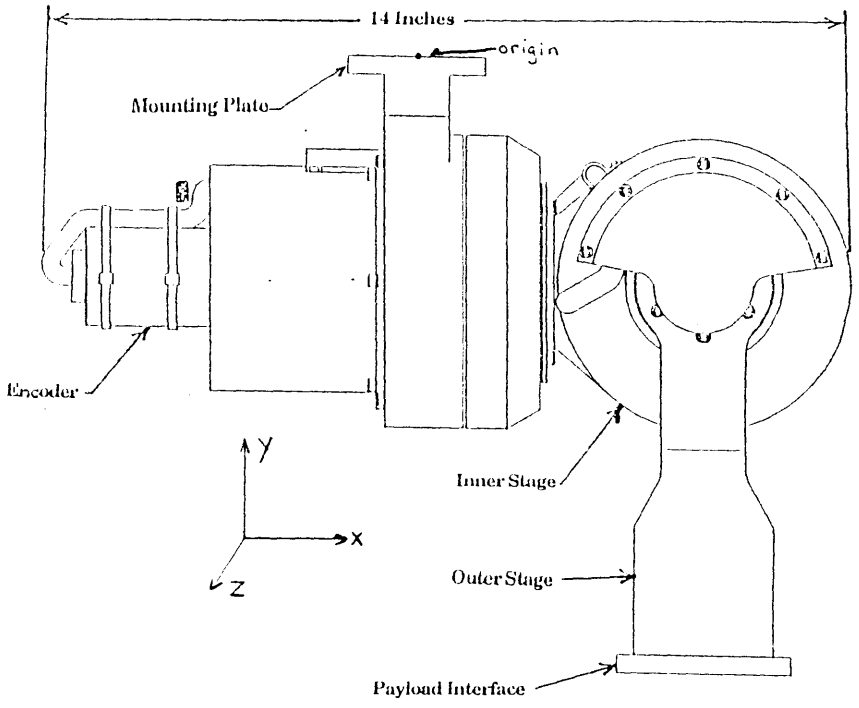


Figure 3.4: Gimbal

are collocated and consequently, use the same nodal points. The location of the gimbal axes, in the coordinate frame of Figure 3.4, is given in Table 3.1. The rotation axes of the reaction wheel motors are assumed to be located at the center of mass of the wheels as given in Table 3.1. In the case of the rate gyros, the location of the rate gyros contained in the package is assumed to be coincident with the center of mass of the rate gyro package. For the strain gages, see Figures 3.2 and 3.3, two nodal points are used, located at the two ends of the strain gage. For the free-free model of the

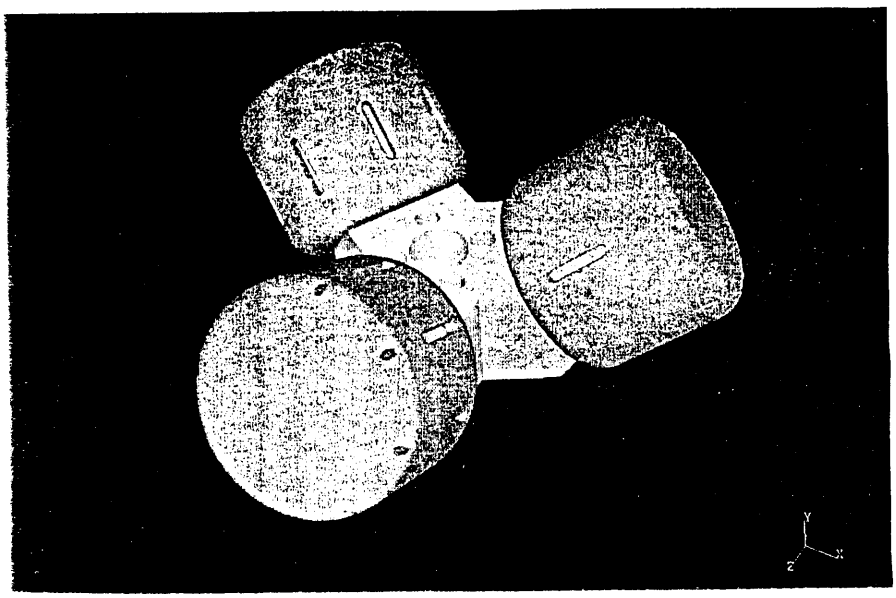


Figure 3.5: Reaction wheel assembly

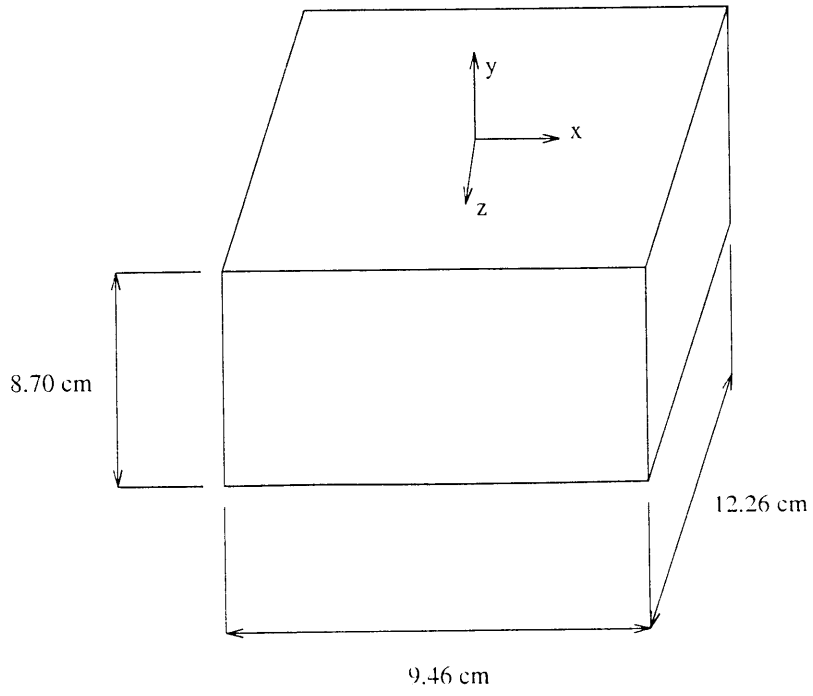


Figure 3.6: Rate gyro package

Table 3.1: Component centers of mass and actuator locations

Component Stage	x (cm)	y (cm)	z (cm)
Gimbal Base	-0.281	-7.257	-0.116
Gimbal Inner (X-Axis) Stage	6.661	-7.824	-0.437
Gimbal Outer (Z-Axis) Stage	10.593	-11.401	-0.005
Gimbal Payload	10.508	-26.307	0.000
Dummy Payload	10.508	-24.498	0.000
Gimbal Axes	10.508	-7.988	0.000
RWA Base	0.000	6.279	0.000
RWA Wheel 1	0.000	8.067	10.629
RWA Wheel 2	9.205	8.067	-5.314
RWA Wheel 3	-9.205	8.067	-5.314
Rate Gyro Package	0.000	-4.350	0.000

Table 3.2: Material parameters

Material	Young's Modulus (N/m^2)	Poisson's Ratio	Density (kg/m^3)
$Lexan_{end}$	2.794×10^9	0.37	2021.77
$Lexan_{center}$	2.626×10^9	0.37	2021.77
Stainless Steel	1.305×10^{11}	0.30	
Aluminum	6.890×10^{10}	0.30	
Active $Strut_{end}$	2.794×10^9	0.37	1838.18
Active $Strut_{piezo}$	3.986×10^9	0.37	1968.12
Active $Strut_{center}$	2.626×10^9	0.37	1838.18

MACE EM, 78 nodal points are used to model the test article, resulting in a total of 433 degrees of freedom, after the constrained degrees of freedom had been removed.

These nodal points are connected using 32 Timoshenko and 38 Bernoulli-Euler beam elements with 7 different sets of material and cross-section properties, summarized in Tables 3.2 and 3.3. The Lexan™ materials are used for the passive struts, stainless steel for the collars, Lexan™ and piezo-electric materials for the active strut, and aluminum is used as a rigid material for the nodes, gimbals, reaction wheels, and rate gyros. In some cases, such as the passive and active struts, multiple materials are used to describe a single component.

Due to the varying cross-section properties at the ends of the exposed portion of the passive strut and the complexity of the strut-collar connection, the strut is broken

Table 3.3: Element sections

Material	Outer Diameter (cm)	Inner Diameter (cm)	Shear Factor
Lexan_{end}	2.883	2.248	0.5492
Lexan_{center}	2.883	2.248	0.5492
Stainless Steel	4.445	0.000	
Aluminum	7.282	4.026	
Active Strut_{end}	2.883	2.248	0.5492
Active Strut_{piezo}	3.410	2.353	0.5646
Active Strut_{center}	3.410	2.353	0.5646

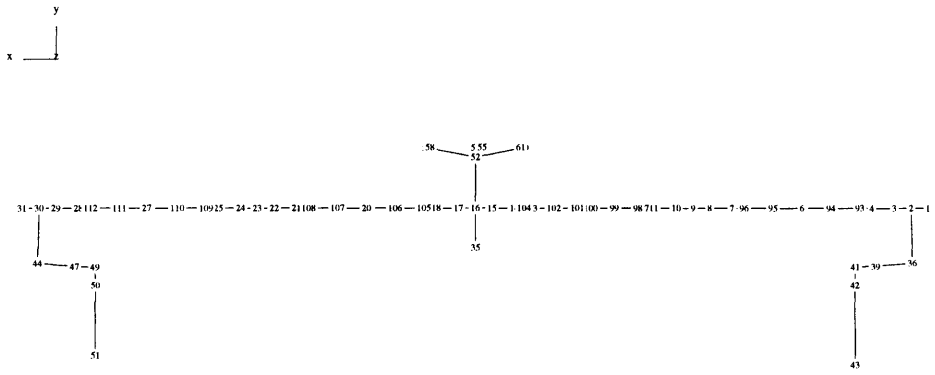


Figure 3.7: MACE finite element model

into two parts (see Figure 3.2). The portion near the strut-collar connection is called Lexan_{end} and the center of the strut is called Lexan_{center}.

Since the active strut (see Figure 3.3) is similar to the passive strut, with the exceptions being the cross section of the center part of the strut and the piezo-electric material covering portions of the center part of the active strut, it is broken into three parts. Two of these parts are the same as the passive strut, namely the end and center portions. The additional material is a Lexan^{TM} and piezo-electric composite, where the material stiffness properties are computed to match the effective bending stiffness of the actual strut determined from the true cross-section and material properties.

Figure 3.7 shows the finite element nodal point locations and connectivities for the MACE free-free model. For reference, the finite element mesh is oriented with the primary gimbal on the right and the secondary gimbal on the left, as in Figure 3.1.

Each number in the drawing is a nodal point and each line is an element. The representations of the gimbals and reaction wheels can be clearly seen connected to the long, straight structural bus.

With all the stiffness properties of the structure defined, the only thing left to complete the free-free finite element model is to define the mass properties of the structure. Two types of mass are used in the finite element model, distributed and concentrated mass. For distributed mass, the finite element code simply uses the material density of the element, along with the previously defined element length and cross-section dimensions, to compute the mass properties. This type of mass is used for the elastic components of the structure, such as the passive and active struts. The MACE structure has electrical wires running through the hollow interior of the struts and nodes, the mass of which are included in the model. To capture this mass, the passive and active struts with wires and plugs in place are weighed, with effective densities computed by dividing the mass of the strut by the modeled volume.

Concentrated masses are used for more complex components, where the modeled cross-section may not adequately capture the true geometry of the component, or for components that are assumed rigid. This is the case for the nodes, rate gyro packages, gimbals, and reaction wheel assembly. For lumped mass, the component mass, obtained from weighing the component, and rotary inertias are lumped at a single nodal point, usually the center of mass. Several methods can be used to find the component rotary inertias. For simple components, a material density can be assumed and the rotary inertia computed analytically using standard integral formulas.

The second method, which can be used for more complex components, is experimental in nature. The rotary inertia, J, can be computed from the frequency of a rotary pendulum, Figure 3.8, using the well-defined relationship [16]

$$J = \frac{mgd^2}{4f^2l} \tag{3.1}$$

where m is the component mass, g is the gravitational constant, d is the cable separation, f is the pendulum frequency in radians/second, and l is the pendulum length. This equation also assumes that the component center of mass is directly centered

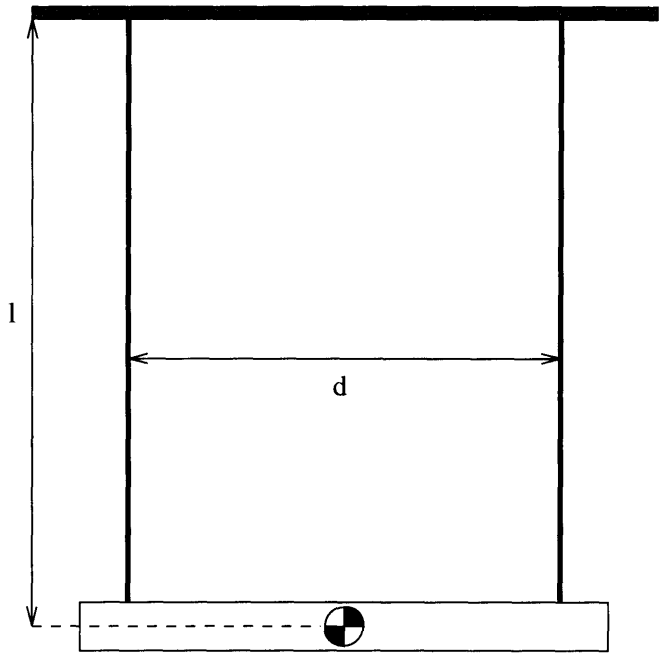


Figure 3.8: Rotary pendulum

Table 3.4: Component mass properties

Component	mass (kg)	J_{xx} (kgm^2)	J_{yy} (kgm^2)	J_{zz} (kgm^2)
Node 1	0.8300	.00099	.00122	.00130
Node 2	0.7692	.00097	.00112	.00060
Node 3	0.7328	.00095	.00077	.00087
Node 4	0.7167	.00067	.00082	.00059
Node 5	0.8300	.00099	.00122	.00130
Bus Rate Gyro	1.3243	.00198	.00186	.00182
Payload Rate Gyro	1.1973	.00195	.00185	.00135
Dummy Rate Gyro	1.2729	.00226	.00241	.00068

between the ends of the two cables. The rotary pendulum can only be used for single stage components, such as the nodes and rate gyro packages. Table 3.4 summarizes the lumped mass properties of these components obtained from the rotary pendulum tests. Since the dummy rate gyro package consists of two aluminum bars, its rotary inertias were computed analytically.

The third method is reserved for very complex or multi-stage components, which cannot be tested using the rotary pendulum. In this method, a solid model of the component is generated using a solid modeling program such as I-DEAS [12]. Exam-

Table 3.5: Reaction wheel mass properties

RWA Component	mass (kg)	J_{xx} (kgm^2)	J_{yy} (kgm^2)	J_{zz} (kgm^2)
Base	6.0618	.03555	.05497	.03555
Wheel (principal)	3.4168	.00456	.00456	.00569

Table 3.6: Inertia wheel Euler angles

Wheel Number	Θ_1 (deg)	Θ_2 (deg)	Θ_3 (deg)
1	-35.258	0.000	0.000
2	-15.792	108.438	-41.804
3	164.208	108.438	138.196

ples of very complex components which cannot be tested using the rotary pendulum, due to their multi-stage nature, are the reaction wheel assembly and gimbals. The mass properties of the reaction wheels are given in Table 3.5 for the base stage and the inertia wheels. The inertias of the inertia wheels are about the wheel principal axes, which are coincident with the rotation axes. The Euler angles used to define these skew coordinate systems, whose z-axes are aligned with the rotation axes, are given in Table 3.6. The Euler angles are defined in a 1-2-1 sequence, such that Θ_1 is about the x-axis, Θ_2 is about the rotated y-axis, and Θ_3 is about the doubly-rotated x-axis. The gimbal mass properties are given in Table 3.7 [17]. The listed gimbal mass properties are about the individual stage centers of gravity. The secondary gimbal rotary inertias differ from those of the primary gimbal only in the signs of some of the off-diagonal terms, due to the 180° rotation between the primary and secondary gimbals. The rotary inertias, except the inertia wheels, of these two tables are in the global coordinate frame defined in Figure 3.1. Inertias of the inertia wheel are given in a local coordinate system. With the description of the lumped mass properties of the structure, the free-free finite element model is complete.

Table 3.7: Gimbal mass properties

Gimbal Stage	mass (kg)	J_{xx} (kgm^2)	J_{yy} (kgm^2)	J_{zz} (kgm^2)	J_{xy} (kgm^2)	J_{xy} (kgm^2)	J_{xz} (kgm^2)
Base	1.3370	.00258	.00304	.00354	-.00002	.00005	-.00002
Inner Stage	2.0757	.00295	.00703	.00678	.00011	-.00032	-.00006
Outer Stage	1.2309	.00505	.00165	.00424	-.00009	.00000	.00006

3.3 Suspension and Gravity Effects

Once the free-free model is completed, the final step in generating the 1-g finite element model is to include the gravity effects, which come in five primary categories: suspension effects, geometric stiffening effects, initial static deformation effects, gravity effects on sensors and actuators, and nonlinear gravity and suspension effects. This section discusses these gravity effects to the extent that they manifest themselves in the finite element modeling process. For a more extensive treatment of suspension and gravity effects see Rey's work in References [7, 13, 18].

Three of these gravity effects are of importance in MACE: suspension effects; geometric stiffening effects; and initial static deformation effects. The gravity effects on sensors and actuators are not important because the only sensors and actuators found to be susceptible to gravity are accelerometers and proof-mass actuators, both of which are not found on the MACE test article. The nonlinear gravity and suspension effects are not treated because there is no easy way to incorporate them into a linear model and are most often negligible due to small angles.

In order to conduct ground testing, the structure must be suspended. To minimize the interaction of the suspension with the structure's dynamics, the bounce and pendular frequencies of this suspension system are desired to be at least one decade below the fundamental mode of the structure. Because of these requirements, the suspension system must have very low stiffness to minimize the bounce frequency and be very long to minimize the pendular frequencies. If simple springs were used for the suspension system, their lengths would be many meters long, which is impractical. For this reason, alternate mechanical suspension devices have been developed which have low stiffness but are small enough to allow ground testing without the

requirement of extremely high ceilings.

The MACE EM test article is supported by three pneumatic-electric suspension devices, shown in schematic in Figure 3.9, and three 4.7 meter long graphite-epoxy composite cables attached to the test article end nodes and center node with small universal joints. The pneumatic-electric devices behave like low stiffness springs in the vertical (gravity) direction with a bounce frequency of approximately 0.25 Hz. The upper end of each suspension cable is connected to a suspension carriage which is constrained to only move in the vertical direction by low friction air bearing shafts. Each suspension carriage is attached to a piston which travels in one end of an air filled chamber that is connected to a large volume air tank. Each tank is pressurized to provide the correct force on the piston to offload the weight of the structure. The combination of tank pressures for the three devices is used to level the structure for ground testing. The electric portion of the suspension devices consist of a displacement sensor and motor to actively control piston position. Most of the vertical stiffness of the device is provided by the large air tank, with an additional smaller stiffness and damping provided by the electric portion of the device [15].

The pneumatic-electric suspension devices are modeled in the finite element model as a simple mass and spring, as shown in Figure 3.9. The mass used in the finite element model, implemented as a lumped mass located at the top nodal point of the suspension cable, is simply the mass of the suspension carriage, given in Table 3.8. The bounce stiffness of the suspension devices is the sum of the pneumatic stiffness, k_p, and the electro-mechanical stiffness, k_{em}, which are represented in the finite element model by a single stiffness, $k_{suspension}$. Since computing k_p and k_{em} analytically is very difficult, an approximation to this sum is found using the finite element model. In this approximation, a static solution is performed in which the reaction forces, F_R, at each suspension device are calculated. Then the stiffness of each suspension device is computed assuming a bounce frequency, f_b, of 0.25 Hz, according to the following equation

$$k_{suspension} = \frac{4\pi^2 f_b^2 F_R}{g} \approx k_p + k_{em} \tag{3.2}$$

where f_b is in Hertz and g is the gravitational constant. Table 3.8 gives the spring

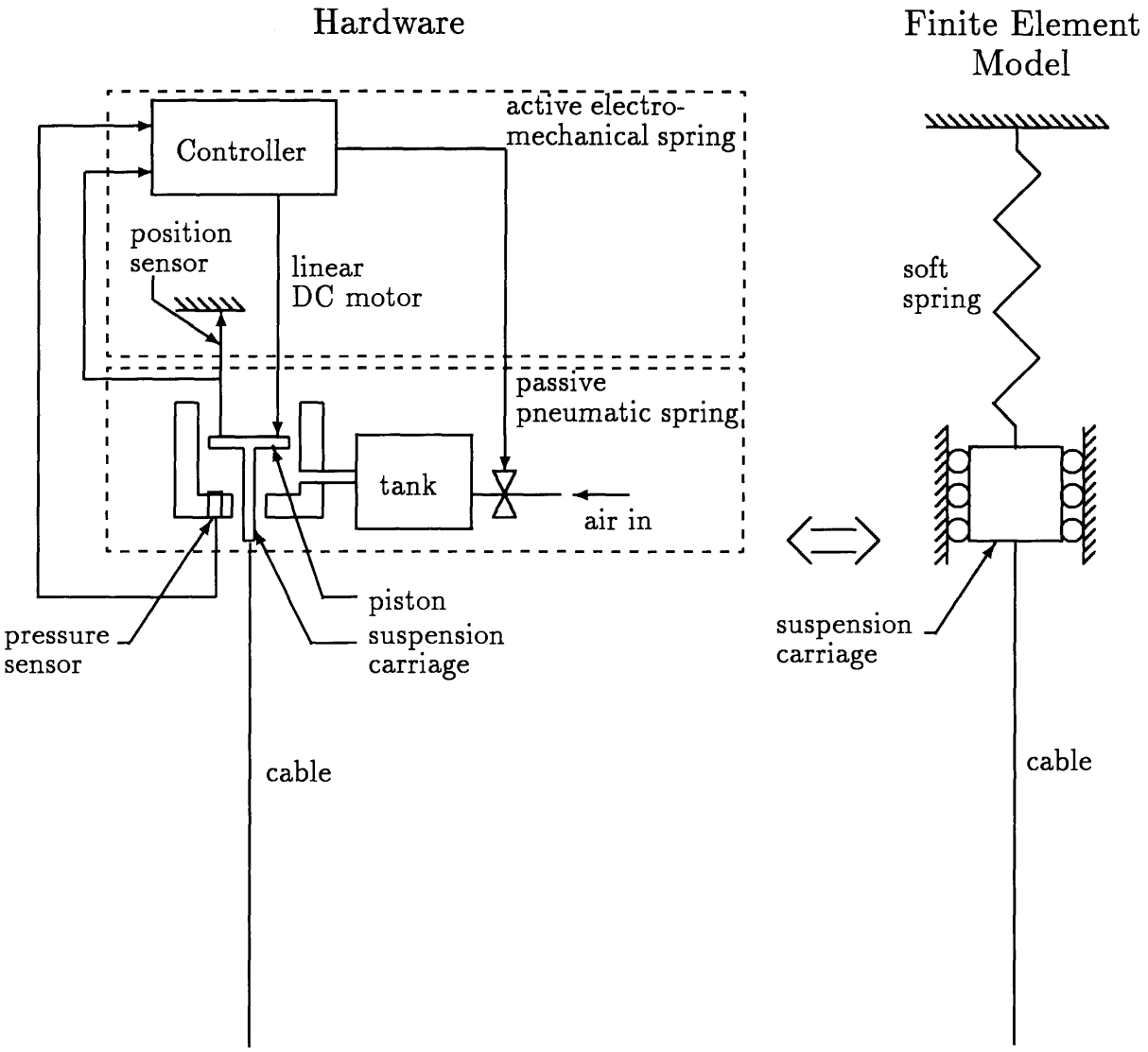


Figure 3.9: Suspension schematic

stiffnesses for the MACE EM finite element model, where suspension device number 1 is connected to the primary payload end of the test article, device number 2 is connected to the center node, and device number 3 is connected to the secondary payload end of the test article (Figure 3.1).

Several representations of the suspension cables can be used in the finite element model, depending on the level of fidelity desired and the degree of coupling exhibited between higher order suspension dynamics and the structure's dynamics. If only the pendulum effect of the suspension cables is desired in the model, each cable need only

Table 3.8: Suspension device parameters

Suspension Device Number	Carriage Mass (kg)	$k_{suspension}$ (N/m)
1	1.3132	16.988
2	1.3132	56.282
3	1.3132	17.038

Table 3.9: Suspension cable parameters

Cable Number	Density (kg/m^3)	Young's Modulus (N/m^2)	Poisson's Ratio	Cable Diameter (mm)	Cable Length (m)
1	1991.37	1.655×10^{11}	0.30	3.175	4.7
2	1991.37	1.655×10^{11}	0.30	3.175	4.7
3	1991.37	1.655×10^{11}	0.30	3.175	4.7

be represented by two nodal points, one at the top and one at the bottom, connected using a single rigid element. If, on the other hand, higher order suspension dynamics, such as compression and transverse violin modes become significant (*i.e.*, they have frequencies in the bandwidth of interest for the structure), the full mass and stiffness properties of the cables, summarized in Table 3.9, should be included in the finite element model. To determine if the compression and violin effects do need to be modeled, the fundamental frequencies of these modes are examined.

The compression mode fundamental frequency in Hertz, f_s, is given by [16]

$$f_s = \frac{1}{2l}\sqrt{\frac{E}{\rho}} \tag{3.3}$$

where l is the cable length, E is the cable's Young's modulus, and ρ is the cable density. For the MACE suspension cable properties, this equation gives a fundamental compression frequency of 970 Hz, several times higher than the 250 Hz Nyquist frequency set by the control computer, indicating that the compression effects do not need to be included.

The fundamental violin frequency in Hertz, f_v, is given by [16]

$$f_v = \frac{1}{2l}\sqrt{\frac{F_R}{\rho A}} \tag{3.4}$$

where F_R is the cable tension and A is the cable cross-sectional area. Using the MACE suspension properties, the fundamental violin frequencies are: 7.0 Hz, 12.7 Hz, and 7.0 Hz for cables 1, 2, and 3, respectively. Since the fundamental violin frequencies lie well below the 250 Hz Nyquist frequency, the mass and stiffness properties of the suspension cables must be included in the model. A concern in modeling the violin modes is that their frequency depends on the cable tension. No easy method for capturing this tensioning effect exists in the finite element method. Fortunately, as will be seen shortly, capturing this tensioning effect is one of the fundamental aspects of capturing the gravity effects on the structure, and as such, the suspension violin modes are simply a by-product of the structural gravity effects modeling.

Once it is known that the higher order suspension dynamics must be included in the model, a decision must be made regarding the number of nodal points to use to represent the cables. Care must be taken in making this decision because, on the one hand, the more nodal points used, the higher the violin mode that can be captured. On the other hand, the more nodal points used, the more degrees of freedom in the model, greatly increasing computation time in the finite element solution. Keeping these factors in mind, 9 nodal points are used to model each of the suspension cables, enabling the 4th violin mode in each of the cables to be captured without significantly increasing the size of the model.

One last aspect of the suspension system must be addressed, namely the universal joints at the bottom of the suspension cables (Figure 3.10). These universal joints connect the suspension cable to a steel rod which is itself attached to the structure at one of the nodes. The universal joints are modeled similar to the gimbals or reaction wheels, namely through the use of coincident nodal points and constraint equations, except in this case only 4 constraint equations are needed because of the two free rotations. Since the steel connecting rod has mass, this mass must be included somewhere in the model. One convenient location is at the node to which the rod is connected. If the mass and rotary inertias are lumped at the structural node, care must be taken to remove them when the suspension system is removed to form the 0-g model.

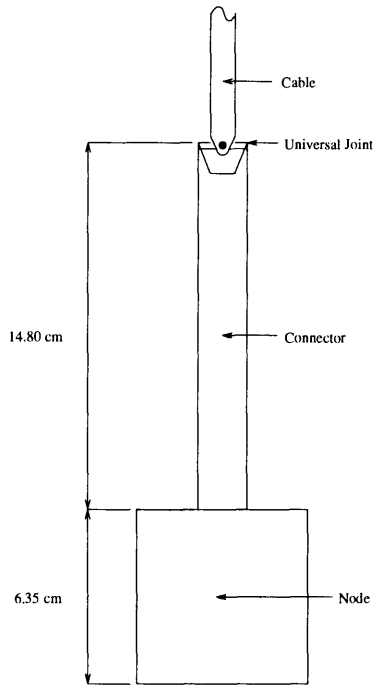


Figure 3.10: Universal joint and connecting rod

With the addition of the universal joint, the finite element representation of the 1-g structure is complete. This model consists of a total of 108 nodal points, resulting in a total of 586 degrees of freedom with the constraint equations and fixed degrees of freedom removed. These 108 nodal points are connected with a total of 32 Timoshenko and 65 Bernoulli-Euler beam elements, and 3 linear spring elements for the suspension devices. The only thing left to complete the 1-g finite element model is to incorporate the structural gravity effects.

The structural gravity effects are: geometric stiffening effects and initial static deformation effects. Geometric stiffening effects arise from initial internal stresses due to the gravity loading. These effects manifest themselves by changing the stiffness characteristics of the system. An analogy to this would be the change in bending frequency of a beam due to an axial loading. Initial static deformation effects arise when the initially undeformed structure undergoes deformation due to the gravity loading. These effects manifest themselves by changing the spatial distribution of the mass and stiffness of the system. An analogy to this would be the change in bending frequency of a beam due to some initial curvature [7, 13]. These effects change the

mass and stiffness characteristics of the structure and are the result of structural sag, pendular effects on articulating appendages, stiffening of rigid body behavior (such as the suspension pendulum behavior), and suspension violin behavior.

The structural gravity effects are captured through a nonlinear large-displacement, stress-stiffening finite element procedure. In this procedure, the gravity load is applied incrementally to the structure and the resulting deformations and stresses at each increment are used to modify the stiffness matrix. Although the initial deformation effects cause changes in the mass matrix due to changes in the spatial distribution of the mass, at present, no procedure exists in any commercial finite element code that takes these changes into account. Due to the small initial deformations seen in the MACE structure, these mass perturbations are negligible. Once the full gravity load has been applied, the final modified stiffness matrix is used along with the unmodified mass matrix in an eigensolution to generate the 1-g modal frequencies and modeshapes.

The 1-g finite element model developed up to this point is analyzed using the ADINA finite element code [19]. Where the free-free model requires only 35 modes to cover the frequency range from 0 to 250 Hz, the 1-g model requires more than twice that number, 80 modes, to cover the same frequency range. Table 3.10 compares the frequencies for the 0-g and 1-g models up to 27 Hz. From the table, the effects of gravity and the suspension system are seen. First, it is seen that the number of zero frequency modes goes from 13 (6 rigid body and 7 articulation modes) to 3 (articulation modes) because all the rigid body modes have been constrained by the suspension system and the gimbal mechanisms have become gravity stiffened pendula. The only remaining zero frequency modes are the reaction wheel articulation modes (motion of reaction wheel and motor armature relative to the motor housing) which are unaffected by gravity. Another effect is seen in the change in frequency of the flexible modes where the frequencies change due to the additional mass and stiffness of the suspension system and the additional stiffness of the geometric stiffening and initial deformation effects. Notice in the table that most of the modes decrease in frequency from 0-g to 1-g, indicating that the mass effects of the suspension system

Table 3.10: Natural frequencies for initial 0-g and 1-g finite element models

Description	0-g (Hz)	1-g (Hz)
reaction wheel 1	0.00	0.00
reaction wheel 2	0.00	0.00
reaction wheel 3	0.00	0.00
X suspension pendulum (X translation)	0.00	0.23
Z suspension pendulum (Z translation)	0.00	0.23
Y suspension twist (Y rotation)	0.00	0.23
bounce (Y translation)	0.00	0.23
Z tilt (Z rotation)	0.00	0.44
X gimbal pendulum	0.00	1.09
X gimbal pendulum	0.00	1.16
Z gimbal pendulum	0.00	1.18
Z gimbal pendulum	0.00	1.27
X twist (X rotation)	0.00	1.67
1st Z bending	2.32	2.11
1st Y bending	3.42	3.72
1st out-of-plane violin, cables 1 and 3		6.76
1st in-plane violin, cables 1 and 3		6.82
1st out-of-plane violin, cables 1 and 3		6.84
1st in-plane violin, cables 1 and 3		6.84
2nd Z bending	9.56	8.96
2nd Y bending	9.77	9.12
torsion, secondary end	11.51	10.78
3rd Z bending	11.32	11.22
torsion, primary end	14.24	12.24
1st in-plane violin, cable 2		13.17
1st out-of-plane violin, cable 2		13.21
2nd in-plane violin, cable 1		13.75
2nd in-plane violin, cable 3		13.81
2nd out-of-plane violin, cable 3		13.94
2nd out-of-plane violin, cable 1		14.03
4th Z bending	16.71	14.30
3rd Y bending	18.85	18.95
3rd in-plane violin, cable 1		20.73
3rd in-plane violin, cable 3		20.77
3rd out-of-plane violin, cable 1		20.77
3rd out-of-plane violin, cable 3		20.81
4th Y bending	27.64	23.22
2nd out-of-plane violin, cable 2		26.31
2nd in-plane violin, cable 2		26.31

dominate over the suspension stiffness and structural gravity effects. A third effect is seen in the suspension cable violin modes, which significantly increase the number of modes in the bandwidth of interest. These violin modes also couple with the structural modes, changing their frequencies. With the 1-g frequencies and modeshapes in hand, the next step is to generate a state-space model of the test article.

3.4 Physical Input-Output Model

Since modern control techniques operate on state-space models, the 1-g finite element frequencies and modeshapes must be converted to a modal state-space model of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3.5}$$

where $x = [\eta\ \dot{\eta}]^T$, η are the modal displacements, u are the actuator inputs, and y are the sensor outputs. The A, B, C, and D matrices are formed according to the following equations

$$\begin{aligned} A &= \begin{bmatrix} 0 & I \\ -\Omega^2 & -2\zeta\Omega \end{bmatrix} \\ B &= \begin{bmatrix} 0 \\ \Phi^T b \end{bmatrix} \\ C &= \begin{bmatrix} c_d\Phi & 0 \\ 0 & c_r\Phi \end{bmatrix} \\ D &= \begin{bmatrix} 0 \end{bmatrix} \end{aligned} \tag{3.6}$$

where Ω^2 is a diagonal matrix with the square of the modal frequencies, ω_i^2, on the diagonal. $\zeta\Omega$ is a diagonal matrix with the product of the damping ratio and modal frequency, $\zeta_i\omega_i$, on the diagonal. Φ is the matrix of modeshapes and b and c describe the behavior of the actuators and sensors in the finite element coordinate system.

The D matrix is zero because no feedthrough is present in any of the sensors of the MACE test article. If the sensor suite included accelerometers, a non-zero D term would be required. Some feedthrough is seen in the active strut to strain gage transfer functions, whose D matrix can be determined empirically.

Forming the structure's A, or system matrix requires only the modal frequencies and damping ratios. The damping ratios, ζ, used in the system matrix are obtained from experimental data of an earlier version of the hardware. While it was not expected that these would be the correct damping ratios, it was expected that using these damping ratios would be closer than simply using 1% for all the modes. In forming the system matrix, a choice can be made regarding which modes to retain in the model. Some modes may be unobservable in all the transfer functions, so carrying around those modes would only increase the order of the model and not provide any better accuracy. Another case is the construction of a model of only the vertical or horizontal dynamics of the structure, in which case only the modes with large vertical or horizontal deformations should be retained, respectively. While the system matrix relies solely on the frequencies and damping ratios, the input and output matrices, B and C, rely solely on the modeshapes.

As indicated in Equation 3.6, with the modeshape matrix Φ known, forming the input and output matrices is simply a matter of determining the input relations, b, and output relations, c. Since the sensors on MACE are either generalized displacement or generalized rate sensors, the sensor relations, c, are broken into two parts: c_d for the displacement sensors and c_r for the rate sensors. The MACE test article has a total of 9 actuators, consisting of 4 gimbals, 3 reaction wheels, and an active strut which actuates vertical and horizontal bending strain. The test article is instrumented with a total of 20 sensors, consisting of 8 strain gage pairs, 4 angular rate encoders, 5 rate gyros, and 3 tachometers. The rest of this section will describe how the actuators and sensors work and how they are modeled in the B and C matrices.

The finite element method assumes that inertial forces and moments can be applied at each nodal point. When defining the input (B) matrix for the system, the subset of inertial forces and moments, which combine to define the actual actuator

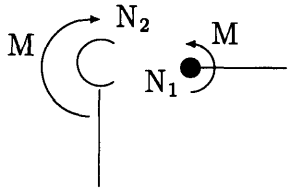


Figure 3.11: Gimbal schematic

behaviors, must be defined. This involves defining a pointing matrix, b, which describes the inertial forces and moments at every nodal point, f, as a function of the actual inputs

$$f = bu = \begin{bmatrix} b_1 & b_2 & \cdots & b_m \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \tag{3.7}$$

The row dimension of b equals the number of modeled degrees of freedom. The column dimension corresponds to the number of inputs, 9 in this case. The following describes how each of the b_i's are selected.

Figure 3.11 shows a schematic of the gimbal. Each of the two DC motors in each gimbal provides a relative moment between the two stages of the gimbal to which it is attached. In the finite element model, this motor is modeled by superimposing two nodal points and constraining all the degrees of freedom of one nodal point to be equal to those of the second nodal point except for the desired rotation. This creates a mechanism in the finite element model. Since the center of mass of the second gimbal stage is not collocated with the hinge point of the motor, this causes the gimbals to be susceptible to gravity, where in 1-g they will behave as pendula. Since the moment applied by the motor is internal, a moment, M, is placed on one nodal point and an equal and opposite moment, M, is placed on the other nodal point, making the net external moment equal to zero. The magnitude of this moment for each mode is found by differencing the values of the eigenvector for the proper degrees of freedom as in

$$M = \theta_2 - \theta_1 \tag{3.8}$$

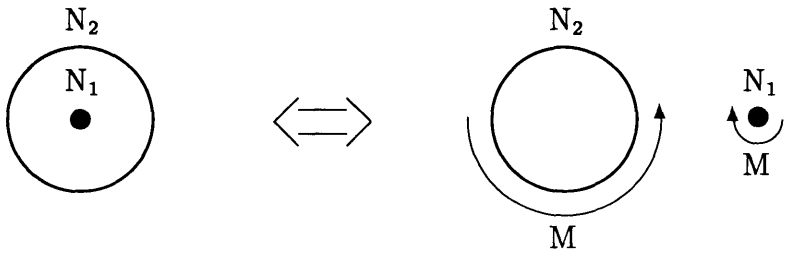


Figure 3.12: Reaction wheel schematic

where M is the resulting modal moment. In this case, the b_{gimbal} matrix has the form,

$$b_{gimbal} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \tag{3.9}$$

where the -1 and 1 entries are in the proper locations for θ_1 and θ_2, respectively. Thus, the product $\Phi^T b$ forms the forcing matrix for each gimbal and all the retained modes.

The DC motor in each reaction wheel provides a relative moment between the inertia wheel and the reaction wheel base by changing the speed of the spinning inertia wheel. Since the axis of rotation of each of the reaction wheels is not aligned with the global coordinate system, skew coordinate systems, with one axis aligned with the motor axis, are used for each reaction wheel. Since the center of mass of the wheel is located on the motor axis, the reaction wheels are not susceptible to gravity and therefore have a zero frequency mechanism mode even in 1-g. As for the gimbal, the moment of each reaction wheel is internal, resulting in the same equation as for

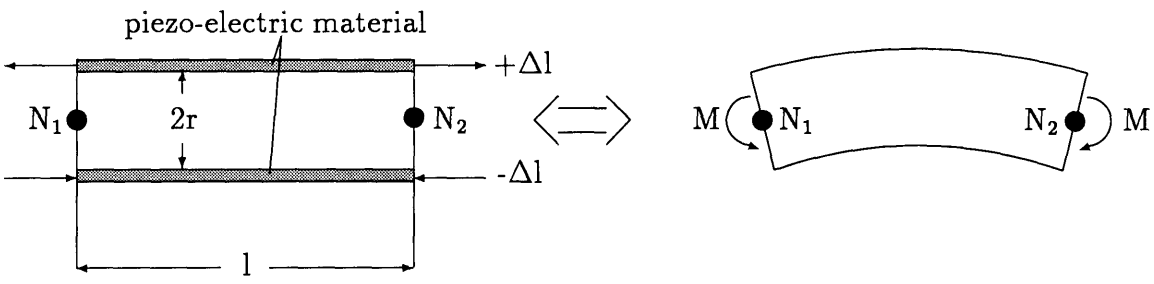


Figure 3.13: Active strut schematic

the gimbal, where in this case the θ's correspond to the local skew coordinate system.

$$M = \theta_2 - \theta_1 \tag{3.10}$$

Due to the similarity of the reaction wheels and gimbals, the $b_{reaction\ wheel}$ matrix is formed in a similar manner as b_{gimbal}, except the reaction wheel degrees of freedom are used.

Since it is desirable to have the net reaction wheel moments aligned with the global coordinate system, a transformation must be made to take the moment of each reaction wheel and convert it into its global X, Y, and Z components. This geometric transformation for the MACE reaction wheel assembly is given in the following equation:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 0 & 0.7071 & -0.7071 \\ 0.5773 & 0.5773 & 0.5773 \\ -0.8165 & 0.4083 & 0.4083 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \tag{3.11}$$

The active strut, shown in Figure 3.13, actuates a bending strain through the use of piezo-electric materials. In the schematic, the voltage applied to the active strut causes the upper surface to expand and the bottom surface to contract. This results in a net bending strain, shown in the first part of Equation 3.12. In the finite element model, each piezo-electric patch is modeled using two nodal points connected by a beam element. The bending strain caused by the piezo-electric elements is equivalent to the bending strain caused by equal and opposite moments, M, applied at each end of the beam element [20]. Since bending strain is the second derivative of displacement, or the first derivative of rotation, the modal magnitude of these

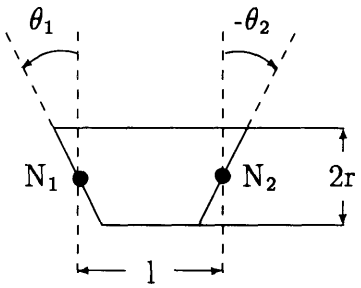


Figure 3.14: Strain gage schematic

moments can be related to the nodal rotations by a central difference approximation, resulting in the second part of Equation 3.12

$$M = \frac{\Delta l_{upper} - \Delta l_{lower}}{l} = \frac{2r}{l}(\theta_2 - \theta_1) \tag{3.12}$$

where $2r$ is equal to 2.96 centimeters and l is equal to 8.89 centimeters. This equation is then applied to both piezo-electric patches. Since the active strut equation is similar to the gimbal equation, the resulting $b_{active\ strut}$ matrix is also similar to the b_{gimbal} matrix, except $b_{active\ strut}$ is scaled by $\frac{2r}{l}$ and uses the active strut degrees of freedom.

With the description of the active strut, the complete B matrix can be assembled, with the resulting actuator inputs in units of Newton-meters. To complete the state-space model of the structure, descriptions of the displacement sensors (*i.e.*, strain gages), and rate sensors (*i.e.*, angular rate encoders, rate gyros, and tachometers) need to be found. These sensor descriptions are made easier by the fact that many of the sensors use the same degrees of freedom as the actuators.

The strain gages (Figure 3.14) are connected in pairs with each gage mounted on opposite sides of the strut, such that they sense bending strain. In the strain gages, this strain, given by the first part of Equation 3.13, can be related through a central difference approximation to give the second part of Equation 3.13.

$$y = \frac{\Delta l_{upper} - \Delta l_{lower}}{l} = \frac{2r}{l}(\theta_1 - \theta_2) \tag{3.13}$$

The gage length of all the strain gages is 3.2 millimeters. For the strain gages located on the passive struts, $2r$ is 2.88 centimeters and for the active strut, $2r$ is 2.96 centimeters.

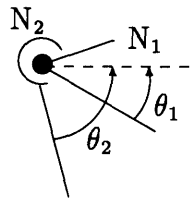


Figure 3.15: Encoder schematic

The encoder, Figure 3.15, is collocated with the gimbal and senses the relative angular rate between the two gimbal stages. Since the encoder is collocated with the gimbal motor, the nodal points used for the motor are again used for the encoder. The modal magnitude of the encoder output is found by taking the difference between the angular rates at each nodal point, shown in Equation 3.14.

$$y = \dot{\theta}_2 - \dot{\theta}_1 \tag{3.14}$$

A problem arises here because the finite element code generates displacement modeshapes, not rate modeshapes. Two solutions to this problem exist which rely on the fact that for the modal decomposition, the displacement modeshapes, Φ_d, and the rate modeshapes, Φ_r, are related by:

$$\Phi_r = j\Phi_d\Omega \tag{3.15}$$

where j is $\sqrt{-1}$ and Ω is a diagonal matrix with the modal frequencies along the diagonal. The first solution is simply to select the proper rotation degree of freedom for each mode, multiply it by $j\omega_i$, and place it in the displacement part of the C matrix, but this leads to a complex C matrix. The second solution relies on the fact that the state vector already includes the $\dot{\eta}_i$ degree of freedom which is related to the η_i degree of freedom by $\dot{\eta}_i = j\omega_i\eta_i$. This equation also contains the $j\omega_i$ term needed to form the rate modeshapes. Thus the best solution is to select the proper degrees of freedom from Φ_d and place them in the rate part of the C matrix, keeping the C matrix real.

The rate gyro, Figure 3.16, differs from all the other sensors in that it is an inertial sensor, rather than a relative sensor, and that it is not collocated with an actuator. As

Figure 3.16: Rate gyro schematic

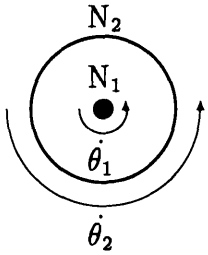


Figure 3.17: Tachometer schematic

such, only one nodal point is required to model a rate gyro. The rate gyro measures the inertial angular rate at the point at which it is attached, shown by the following equation

$$y = \dot{\theta} \tag{3.16}$$

The tachometer, Figure 3.17, is collocated with the reaction wheel motor and measures the relative angular rate between the reaction wheel motor and the inertia wheel. The tachometer is not a sensor to be used in control design, but still must be modeled because they are used in reaction wheel speed control servos. Since the motor and tachometer are collocated, the tachometer uses the same nodal points and degrees of freedom as the reaction wheels. The modal magnitude of this sensor is found by taking the difference of the angular rates of the two nodal points for each mode, shown by the following equation:

$$y = \dot{\theta}_2 - \dot{\theta}_1 \tag{3.17}$$

Since the reaction wheel inputs have already been aligned with the global coordinate system, the tachometer outputs must also be aligned with the global coordinate system using the same transformation as for the reaction wheels.

Incorporating the sensors into the C matrix follows a path similar to incorporating the actuators into the B matrix. Equation 3.13 is used to form the c_d matrix and

Equations 3.14, 3.16, and 3.17 are used to form the c_r matrix, which are combined with the modeshapes, Φ, as in Equation 3.6, to form the complete C matrix.

With a description of the input and output relations complete, any actuator or sensor dynamics must be considered. These dynamics arise from the physical phenomena that allow the actuators and sensors to convert electrical energy to physical energy and *vice versa*. Other effects are caused by interactions with gravity, and would be included here. The only identified dynamics are a resonance in the rate gyros. This resonance is near 275 Hz, at the extreme upper end of the the bandwidth of interest, and consequently is not captured in the finite element model. Although in some cases actuator and sensor dynamics are important, for the case of the MACE EM, they can be neglected.

Since the actuator and sensor dynamics do not need to be included in the model, the generation of a physical input-output state-space model of the structure is complete. This physical model has inputs with units of Newton-meters and outputs with units of radians, strain, and radians/second. The inputs and outputs of the structure as seen by the control computer, though, are in volts. Thus, the next step in the generation of the input-output model used for control design is to find the electronic gains, which turn the physical model into a volts-volts model, and model any filters which might be included on any of the input or output channels.

3.5 Electronics

Since the actuator inputs and sensor outputs as seen by the control computer are not really moments and angles, but are instead in volts, the units of the inputs and outputs in the finite element model must be converted to volts. To attenuate noise and other effects, these voltage inputs and outputs may be passed through electronic filters. The dynamics of these filters are likely to affect the measured input-output behavior of the structure in the laboratory, and as such, models of these filters must be included in the finite element model.

Converting the inputs and outputs to volts is simply a matter of finding the

Table 3.11: Actuator gains

Gimbals	Reaction Wheels	Active Strut
0.2625 Nm/V	0.1031 Nm/V	0.1223 Nm/V

Table 3.12: Sensor gains

Encoders	Strain Gages	Rate Gyros	Tachometers
1 V/(rad/sec)	0.5 mV/μstrain	12.319 V/(rad/sec) 8.213 V/(rad/sec)	0.0934 V/(rad/sec)

electronic gains, which are a combination of the hardware sensitivities and a net amplifier gain. The hardware sensitivities are the gains of the hardware without any amplification, such as the motor constants of the gimbals and reaction wheels. The sensitivities are generally taken from supplier specification and testing sheets. The signals that come from the sensors or go to the actuators generally must be amplified to provide the proper voltage and current levels. This amplification comes from two sources, amplifiers hard wired into the signal path and software selectable amplification. The net amplification is the product of the hardware amplifier gains and the software selectable amplifier gains. Tables 3.11 and 3.12 summarize the actuator and sensor electronic gains for the MACE EM actuators and sensors.

It should be pointed out that the gain listed for the active strut is empirically determined by matching the magnitude of the finite element model transfer functions with those obtained from experiment. Note, also, that two gains are listed for the rate gyros. The first of these gains is the gain determined solely from the known hardware sensitivity and amplifier gains. When the model transfer functions using this gain were compared with experimental transfer functions, it was found that the model transfer functions were larger in magnitude by a factor of 1.5 than the experimental transfer functions, for which no explanation can be found. The smaller value, which reflects the extra gain, is used in the subsequent volts-volts model.

With the electronic gains determined, the model inputs and outputs are scaled by these gains resulting in a volts-volts model. This model captures the essential structural and input-output behavior of the test article. Other factors, such as signal

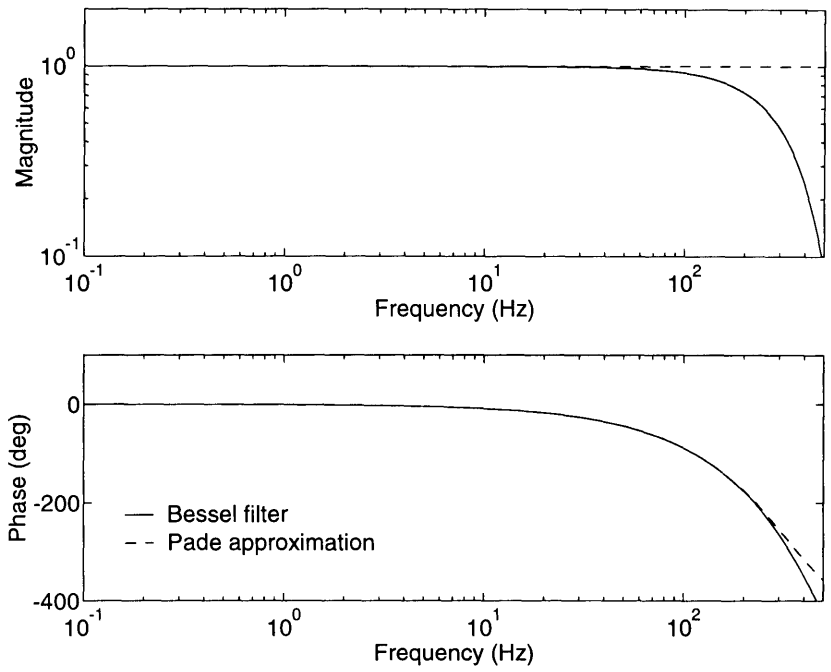


Figure 3.18: Transfer functions of a Bessel filter and 3rd order Pade approximation to a 2.44 millisecond time delay

conditioning filters and local servo controls, may also affect the input-output behavior of the test article. These other factors also need to be considered and their effects included in the model. The signal conditioning filters will be treated in the remainder of this section and the servo controllers in the next section.

The only signal conditioning filters used on the MACE test article are anti-aliasing Bessel filters on all analog outputs, and notch filters on the rate gyro channels to reduce the effect of rate gyro sensor dynamics. Both these filters serve to reduce the effects of high frequency factors which can cause problems in the frequency response of the test article.

The Bessel filters used on the MACE test article are of the 8 pole variety, with the -3 dB point at 200 Hz. Figure 3.18 shows the transfer function of one of these Bessel filters. Since the Bessel filter is a well-known filter, state-space descriptions of these filters are easy to obtain. In looking at the Bessel filter transfer function, it is seen that the magnitude effects are very small up to about 100 Hz, but the phase

effects are quite significant even at low frequency. Since the effects of the Bessel filter are significant in the bandwidth of interest, a representation of the filter must be included in the model. Upon closer inspection of the phase transfer function, it is seen that the phase portion of the transfer function looks like a time delay. This indicates that two methods of incorporating the Bessel filter into the finite element model are possible: including the state-space description of the filter and replacing the filter with a Pade approximation of the equivalent time delay [14], 2.44 milliseconds, shown in Figure 3.18 for a third order approximation. As can be seen from the phase transfer function, the Pade approximation provides a very good approximation to the Bessel filter phase, deviating only at very high frequencies. The magnitude transfer functions show that the Pade approximation has unity gain for all frequencies, indicating that capturing the magnitude effect of the Bessel filter is impossible with the Pade approximation.

Two issues need to be addressed when considering which model of the filter to incorporate into the finite element model. The first of these is the magnitude effects of the Bessel filter. If, in the frequency range of interest, the magnitude effects of the filter become important, then the full state-space description of the Bessel filter should be used. The second issue is the number of states each model uses to describe the filter. This issue is important because the larger the final input-output model, the more unwieldy it becomes for control design, where most techniques generate controllers of the same order as the model from which they are designed. The full state-space description of the Bessel filter requires 8 states per filter. With 13 analog output channels, that means 104 extra states in the model just for the filters. The third order Pade approximation requires only 3 states per filter, adding just 39 states to the model. Thus, if the magnitude effects of the Bessel filter are not important, it is best to model the filter as a time delay. An alternate solution to the magnitude problem might be to fit a reduced order state-space system to the filter so that the magnitude effects can be captured with very little penalty in terms of the number of states added to the model. Regardless of which model of the filter is used, the filter is cascaded through the model using a series connection between the full model and

Table 3.13: Notch filter DC gains for the listed rate gyros

Payload X	Payload Z	Bus X	Bus Y	Bus Z
2.30	2.70	2.59	2.32	2.39

the filter model for each analog output channel.

The second type of signal conditioning filter used on the MACE test article are notch filters on the rate gyro channels, which are used to attenuate the effects of the rate gyro resonance. As for the resonances they are designed to counteract, the dynamics of the notch filters are located near 275 Hz, at the upper end of the frequency range of interest, and can be neglected. The only complication of the notch filters is that their DC gain is not unity. These non-unity DC gains, given in Table 3.13, cause magnitude errors in the rate gyro transfer functions. These gains are incorporated into the finite element model by multiplying the rate gyro electronic gain from Table 3.12 by the additional notch filter gain for each channel.

More phase delay was seen in the experimental data for the rate gyros than was expected in the finite element model. This extra phase delay is likely an effect of the resonance mentioned above. As for the Bessel filter, this extra delay is also modeled as a third order Pade approximation. If the Pade approximation to the Bessel filter is used, this extra delay is added to the filter delay and the net delay is used to compute the Pade approximation. Otherwise, the third order Pade approximation is added to the model along with the full state-space approximation of the Bessel filter.

With these signal conditioning filters included in the model, the model now matches the hardware up to the connection with the control computer. The next step in the generation of the 1-g finite element input-output model is to capture any factors inside the control computer which might affect the input-output behavior of the structure. These factors are any initial local control loops implemented on the structure.

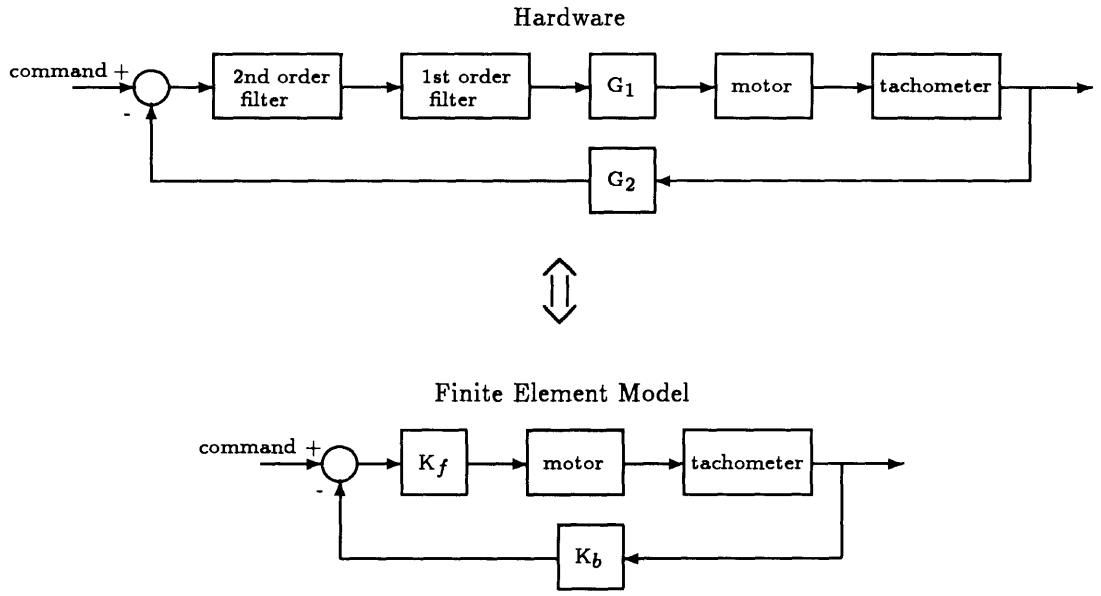


Figure 3.19: Reaction wheel speed control servo block diagram

3.6 Servo Controls

This section discusses the implementation of the reaction wheel speed control and gimbal pointing servos. Although, the speed control servos are implemented in an analog fashion, they are included here because of their similarity to the gimbal servos. Since the gimbal servos are implemented using a digital computer, this introduces a time delay due to the computational and sampling delays of the computer. As for previous time delays, these computational time delays are incorporated into the model using a Pade approximation.

The reaction wheel speed control servos are used to overcome the effects of friction in the bearings of the reaction wheels and avoid runaway in the reaction wheels. A block diagram of the speed control servos is shown in Figure 3.19. The speed control uses the tachometer output to compute an input to the reaction wheel motor such that the motor speed tracks a commanded motor speed. In the hardware, a speed control loop is implemented on each reaction wheel individually. In the finite element model, though, the reaction wheel inputs and tachometer outputs have been transformed to the global coordinate system, and the individual wheel servo loops become equivalent

to loops around each of the coordinate axes. This is satisfactory as long as the reaction wheel motor constants and tachometer sensitivities are the same for each reaction wheel. If they are different, the proper method is to close the servo loop around each wheel individually and then to transform the wheel inputs, now speeds instead of moments, to the global coordinate system.

In the hardware, the forward path contains a first and a second order filter along with a gain, G_1, while the backward path contains only a gain, G_2. These first and second order filters have their dynamics at very high frequency, $\sim$ 250 Hz, and consequently only their DC gains are of importance. The DC gains of these filters, along with the gain G_1, combine to give a forward loop gain, K_f, equal to 4.3. The gain G_2 becomes the backward loop gain, K_b, and is equal to 1.0. With these gains defined, the closed-loop system and input matrices, A_{cl} and B_{cl}, are obtained from the volts-volts state-space model by:

$$\begin{aligned} \tilde{B} &= B_{reaction\ wheel} K_f \\ A_{cl} &= A - \tilde{B} K_b C_{tachometer} \\ B_{reaction\ wheel\ cl} &= \tilde{B} \end{aligned} \tag{3.18}$$

where the reaction wheel subscript refers to the reaction wheel columns of the B matrix and the tachometer subscript refers to the tachometer rows of the C matrix.

The main effect of the speed control servo is to move the zero frequency reaction wheel rate pole to the left in the complex plane. Also note that since the reaction wheel angle pole is unobservable, it remains at the origin and should be truncated from the model. Note that the columns of the closed-loop input matrix, B_{cl}, corresponding to the reaction wheels have been scaled by the forward loop gain, K_f, changing the magnitude of the reaction wheel input-output relationships.

The gimbal servos are used to provide initial pointing control of the two payloads relative to the bus, as would be present on a typical spacecraft with pointing payloads. A side benefit of the gimbal servos is that they are used to overcome friction and stiction effects in the gimbal motors. The gimbal servos use the encoder outputs to generate measures of encoder angles and rates, from which a gimbal torque input is computed. The gimbal servos are implemented on the digital control computer,

which introduces other effects which must be modeled before servo loop closure can occur in the finite element model. Since the computer requires sensor measurements at discrete time points, the computer samples the output channels at a certain frequency, introducing an effective time delay equal to one-half of the sample period. Once the computer has the measurements, it uses them to compute the actuator commands, introducing a further time delay equal to an integral number of sample periods. For most cases the computational delay is equal to a single sample period, making the net time delay due to the digital computer equal to one and one-half sample periods. For some unknown reason, the model implementation of the MACE servos that best matches the experimental data requires a time delay of two and one-half sample periods, 5 milliseconds (for a 500 Hz sample frequency). This computer time delay is again approximated using a third order Pade approximation and incorporated into the model using the following equations:

$$\begin{aligned}
\tilde{A} &= \begin{bmatrix} A & B_{gimbal}C_{td} \\ 0 & A_{td} \end{bmatrix} \\
\tilde{B} &= \begin{bmatrix} B_{gimbal}D_{td} \\ B_{td} \end{bmatrix} \\
\tilde{C} &= \begin{bmatrix} C & D_{gimbal}C_{td} \end{bmatrix} \\
\widetilde{D} &= \begin{bmatrix} D_{gimbal}D_{td} \end{bmatrix}
\end{aligned} \tag{3.19}$$

where A_{td}, B_{td}, C_{td}, and D_{td} are the state-space description of the Pade approximation. This inclusion of the time delay into the servos is very important, as the model implementation of the servos will not work properly without the time delay in the loop.

A block diagram of the gimbal servo loops is shown in Figure 3.20. In the computer, the encoder signals are passed through a digital integrator to provide measures of the encoder angles in addition to the encoder angular rates. This digital integrator is nothing more than a counter, and as such, does not introduce any dynamics into the

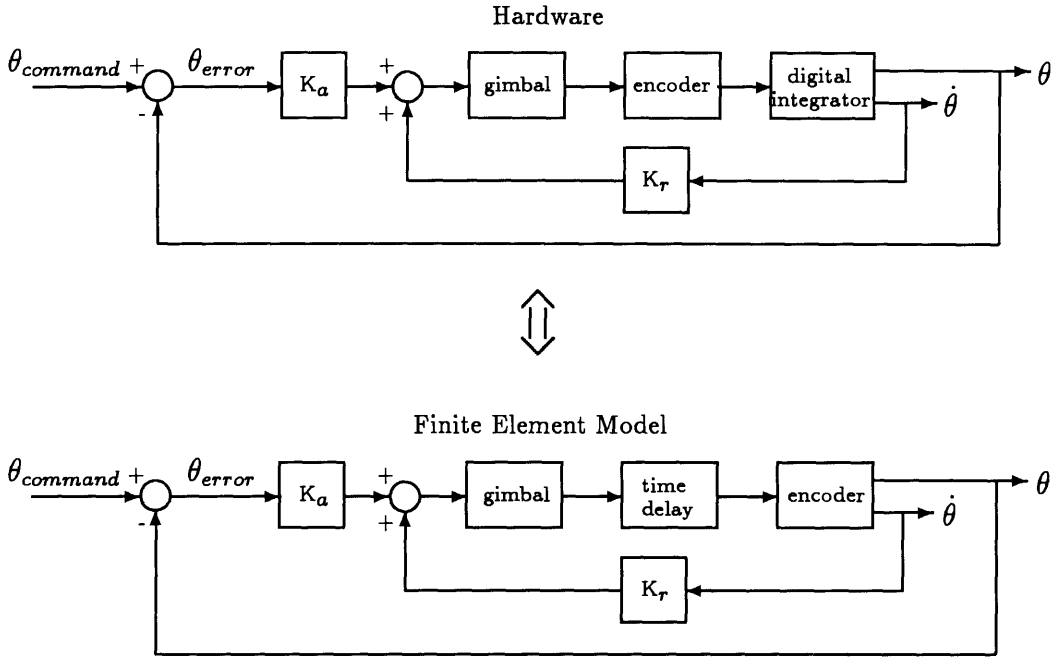


Figure 3.20: Gimbal servo block diagram

encoder signal. Consequently, in the finite element model, the encoder is assumed to provide both angle and angular rate. Obtaining the encoder angle is simply a matter of taking the rate part of the C matrix for the encoder rate and placing it in the displacement part of a new row of the C matrix. The digital part of the integrator is included in the finite element model as the computer time delay. With measurements of the encoder angles and rates, forming the closed-loop system and input matrices, $\tilde{A}_{cl}$ and $\tilde{B}_{cl}$, is simply a matter of applying the angle gain, K_a, and the rate gain, K_r, according to the following equations

$$\begin{aligned} \tilde{A}_{cl} &= \tilde{A} - \tilde{B}_{gimbal}(K_a \tilde{C}_{encoder\ angle} + K_r \tilde{C}_{encoder\ rate}) \\ \tilde{B}_{gimbal\ cl} &= \tilde{B}_{gimbal} K_a \end{aligned} \tag{3.20}$$

where K_a is equal to -2.0, K_r is equal to -0.1, and the $\sim$'d matrices are from the volts-volts model and include the time delays. These gains cause the servo to have a bandwidth of approximately 3 Hz. With the closure of the gimbal servo, the input changes from being a motor torque to a commanded gimbal angle relative to the bus. The principal effect of the servo is to significantly increase the damping of the lower frequency modes. Note that the columns of the closed-loop input matrix, B_{cl},

corresponding to the gimbals have been scaled by the angle gain, K_a, changing the magnitude of the gimbal input-output relationships.

3.7 Initial 1-g Model

With the closure of the servo loops, the generation of the initial 1-g finite element volts-volts, input-output model is complete. This model includes 80 modes from the finite element model. The input-output model includes all actuators and sensors, resulting in a 160 state system (2 states for every mode) with 9 inputs and 20 outputs. The full state-space descriptions of the Bessel filters are included, adding 104 states to the model. The gimbal and speed control servos are included in the model, adding 36 additional states for time delays. The total size of this model is 300 states, which can be significantly reduced by reducing the number of modes retained and not including the full models of the Bessel filters.

Table 3.14 shows a comparison of the modal frequencies and damping ratios for the ground test results and the initial 1-g finite element input-output model. The experimental modal values were obtained from a measurement model that was fit to the transfer function data. This table shows that this initial finite element model is pretty good with many frequency errors less than 10%. But there are still some large frequency errors in the model, most notably the 3rd and 4th Y-axis bending modes. As expected, since the damping ratios used in the finite element model were for a previous set of hardware, the predicted closed-loop damping ratios of the finite element model are significantly in error for many modes. Fortunately, the damping ratios are very easy to update and will improve dramatically. The comparison of the finite element model modal frequencies and damping ratios with the experimental values gives only part of the story.

What is most important for control design is the quality of the prediction of the input-output behavior of the structure. To examine this, the finite element predictions of the transfer functions are compared with experimental transfer functions. Figures 3.21 and 3.22 show predictions of the disturbance to performance transfer

Table 3.14: Comparison of frequencies and damping ratios for ground experimental data and initial 1-g input-output model

Description	Measurement Model		Finite Element Model		Errors	
	Frequency (Hz)	Damping ratio	Frequency (Hz)	Damping ratio	Frequency (%)	Damping (%)
Y twist	0.24	0.100	0.23	0.081	-4.2	-19.0
Z tilt	0.44	0.038	0.45	0.037	2.3	-2.6
X rotation	1.39	0.036	1.40	0.034	0.7	-5.6
1st Z bending	1.96	0.022	2.00	0.028	2.0	27.3
1st Y bending	3.38	0.038	3.38	0.039	0.0	2.6
gimbal pend.	4.57	0.220	4.26	0.257	-6.8	16.8
gimbal pend.	4.79	0.179	5.37	0.458	12.1	155.9
gimbal pend.	5.93	0.414	6.17	0.427	4.0	3.1
gimbal pend.	6.75	0.271	6.50	0.279	-3.7	3.0
1st Violin	6.76	0.033	6.84	0.021	1.2	-36.4
1st Violin	7.14	0.024	6.84	0.015	-4.2	-37.5
1st Violin	7.19	0.022	6.85	0.015	-4.7	-31.8
1st Violin	7.21	0.024	6.87	0.013	-4.7	-45.8
2nd Y bending	9.29	0.011	9.57	0.022	3.0	100.0
2nd Z bending	11.34	0.105	11.45	0.379	1.0	261.0
3rd Z bending	12.11	0.103	12.77	0.204	5.5	98.1
2nd Violin	12.71	0.010	13.21	0.012	3.9	20.0
2nd Violin	14.29	0.017	13.77	0.017	-3.6	0.0
2nd Violin	14.47	0.010	13.79	0.010	-4.7	0.0
2nd Violin	16.88	0.014	13.80	0.014	-18.2	0.0
3rd Y bending	17.06	0.009	18.95	0.010	11.1	11.1
3rd Violin	21.54	0.017	20.73	0.012	-3.8	-29.4
3rd Violin	21.71	0.026	20.74	0.017	-4.5	-34.6
4th Y bending	22.63	0.070	27.56	0.011	21.8	-85.7
time delay	29.88	0.458	30.83	0.486	3.2	6.1
time delay	32.24	0.538	32.57	0.482	1.0	-10.4
4th Z bending	39.05	0.021	40.47	0.016	3.6	-23.8
5th Z bending	45.41	0.015	45.75	0.012	0.7	-20.0
5th Y bending	48.55	0.016	51.77	0.022	6.6	37.5
6th Y bending	53.98	0.022	55.88	0.035	3.5	59.1

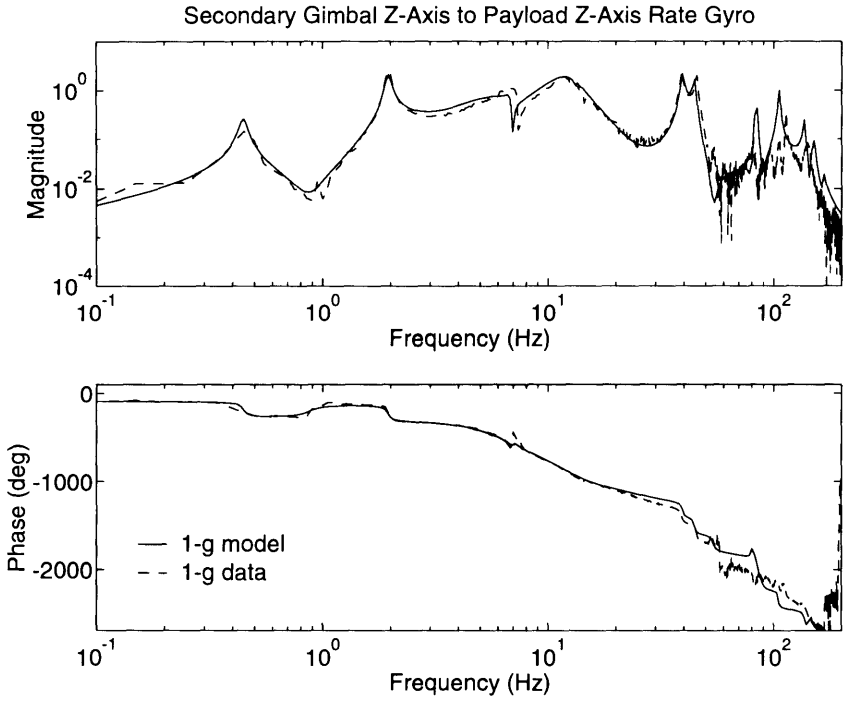


Figure 3.21: Typical in-plane transfer function for initial 1-g input-output model

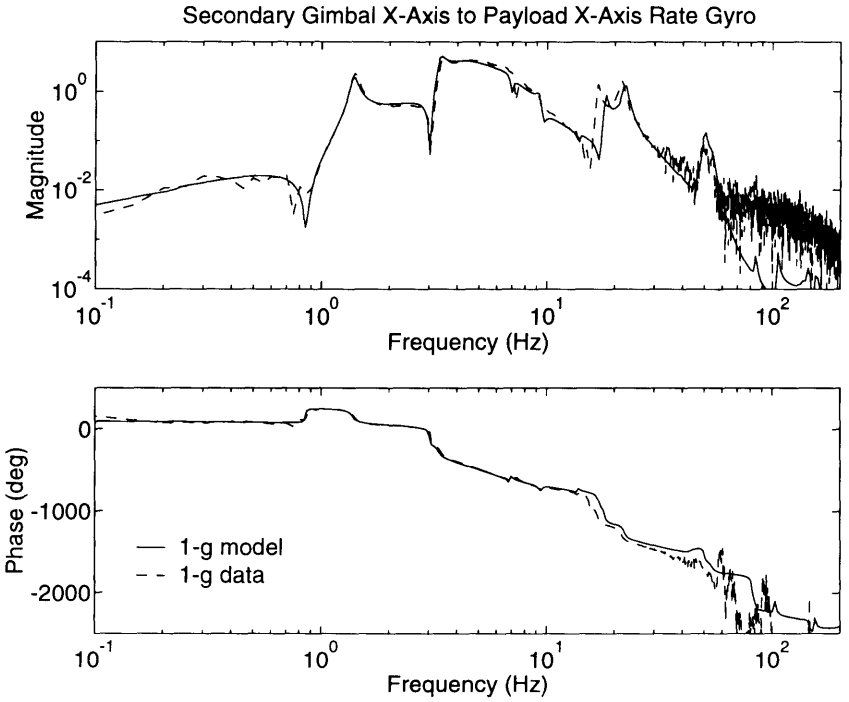


Figure 3.22: Typical out-of-plane transfer function for initial 1-g input-output model

Table 3.15: Free-Free test article properties

mass (kg)	length (m)	J_{xx} (kgm^2)	J_{yy} (kgm^2)	J_{zz} (kgm^2)	ω_f (Hz)
36.64	1.710	0.664	9.104	9.557	2.10

functions using this initial 1-g model. These transfer functions show that although the general characteristics of the transfer functions are good (*i.e.*, all the modes seen in the data also appear in the model), the frequencies and damping ratios of some of the modes are significantly in error. These transfer functions also show that, in general, the in-plane predictions, where in-plane refers to the plane of the suspension cables, of the finite element model are better than the predictions of the out-of-plane behavior of the test article. This is because the effects of gravity are smaller in the in-plane behavior than in the out-of plane. In order to reduce the errors in the predicted transfer functions, the model must be updated, which will be discussed in the next chapter.

Table 3.15 summarizes some of the most frequently used properties of the free-free test article. These properties are the mass, length, and rotary inertias, which are referenced about the center node. The free-free test article fundamental frequency, ω_f, is also included.

Chapter 4

Finite Element Model Updating

Since the initial finite element model of a structure is often formed without the benefit of hardware, many aspects of the finite element model are idealizations of the structure based on engineering drawings and material handbooks. Once hardware exists, experimental data can be taken and compared to the finite element predictions. Seldom, if ever, do the initial finite element predictions overlay the experimental data to a degree that high authority structural control may be performed using the finite element model as a design model. For this reason, the finite element predictions of the structure must be improved, in relation to the data, through a process called updating.

Finite element model updating, detailed in this chapter, comes in several categories: engineering insight, automated updating, and closed-loop updating. The first of these is the updating of the model using the engineer's previous modeling experience. This experience, called engineering insight, leads the engineer to make changes in the model that are based on the structure, the data, and assumptions made in the initial model, often making changes that are little more than an educated guess. This insight often accounts for $\sim$90% of the total improvement in the model in the updating process.

Once the model has been verified using insight, it becomes time to turn the rest of the updating over to computer optimization techniques. In this type of updating, called automated updating, parameters that still remain uncertain are chosen and

optimized to fit the finite element model to the experimental data. Previous work in the area of automated updating has been centered around modal analysis. In modal analysis [21], numerous accelerometers are distributed across the structure and the resulting frequency responses to several different shaker inputs are recorded. Here two distinct groups emerge in the field of automated updating. In the first of these, the experimental data is used to determine experimental modal parameters, which are then used to update the finite element model [22, 23, 24]. Most of the techniques have used only the modal frequencies in the update, but recent work [24] has been done in using the modeshapes in the update as well. The second group operates directly on the frequency response functions [25, 26, 27]. In the majority of both groups, the sensitivities of the predicted modal parameters to the model parameters are captured through finite difference approximations, which have limitations for structures with repeated modes and high modal densities. Because of these limitations, recent work [28] has been done in using reanalysis techniques to capture the model sensitivities.

The automated updating work performed on MACE differs from previous work primarily due to the complexity of the model that is updated. Although complicated structures which include gravity and suspension effects have been updated before [29], the previous work was done using modal analysis, which requires instrumenting the test article with numerous sensors (essentially one sensor per finite element nodal point). This method is impractical for MACE, due to the large cost and time required to implement. Consequently, an alternate approach has been adopted, namely to use only the sensors and actuators already present on the structure in a system-identification type approach [4]. Because of this approach, all the input-output aspects of the structure must be added to the finite element model, which includes filters and local servos. Prior to the work on MACE, all updating has been performed on the bare finite element model without the added burden of the input-output behavior of the test article.

The parameters used in automated updating are the so-called physical parameters, such as masses, stiffnesses, lengths, time delays, *etc.* Another choice would be to

update the modal parameters of the input-output model. The problem with this latter choice is that there would be no way to predict the 0-g behavior because the suspension and gravity effects would be buried in the modal parameters. Updating the physical parameters avoids this problem by staying within the framework of the finite element method. Once the physical parameters have been updated, the suspension and gravity effects can be removed from the finite element model to arrive at the 0-g predictions.

Because the automated updating step is performed on an input-output model which will eventually be used for control design, a closed-loop update step is now possible, which in all previous work was impossible. In this step, the finite element model is used for control design and the resulting controllers implemented on the test article. Remaining problem areas in the finite element model will cause instabilities in the closed-loop experiments. These problem areas are evaluated to see how they can be improved in the finite element model. Once satisfactory closed-loop results have been achieved, the model updating process is complete and the result is a finite element model that is closely correlated with experimental data and has been verified through closed-loop experimentation.

4.1 Engineering Insight

Engineering insight enters in the modeling and updating process at every phase. It is most important in the model verification phase (Step A in Figure 2.1). In the initial modeling process, the engineer must make numerous assumptions about the structural parameters and the structure's behavior, all of which come from the engineer's experience and insight. In updating, engineering insight leads to the inspection of the model input file for typographical and computational errors. Once this stage has been passed, and the hardware exists, the next stage in the update is to verify the mass, stiffness, and dimensional properties used in the model by measuring them on the structure. After these properties have been verified, the remaining errors in the model are due to erroneous modeling assumptions and mismodeled parameters. Thus,

the next phase of the update process is to verify the modeling assumptions in light of the modeling errors. Once these modeling errors have been removed, the entire model can be verified by repeating the model in a different finite element code. If the two models compare favorably, either code may be used. The remaining errors are due to mismodeled parameters which can be updated in an automated fashion. Otherwise, the better of the two models should be used for further updating and modeling.

First, the model input file should be scrutinized carefully for typographical errors and all parameters that are computed should be checked for computational errors. These errors may be found in the nodal point locations, element connectivity, material properties, masses, and coordinate systems. Significant improvements can often be made through this careful examination.

The structural components are initially modeled according to engineering drawings and material handbooks. The mass and rotary inertia properties for these components are found from idealizations of the components, which may be significantly in error. When there is hardware, there is no excuse not to have the mass properties of the components, and consequently the mass matrix, essentially correct. This is because the assumed mass properties can be verified by measuring their actual values. For lumped mass components, the component can be weighed and the rotary inertia measured using a rotary pendulum. For distributed mass components, the component can be weighed and the assumed density adjusted to match the component mass, given the volume of the component. With these measurements, the mass properties used in the model have been verified and corrected.

Initially, the stiffness properties of the structure are the least well known of the model parameters. Some of the component stiffness properties can be verified by measuring them on the hardware. The most likely components for this verification are the elastic components. The stiffness properties can be determined from extension, bending, and torsion tests. Due to the complex geometries of some of the components, these tests may only give effective stiffness properties for the entire component, not for specific parts. The boundary conditions of these tests may not be the same as the assembled structure, affecting the measured stiffness properties. Because of

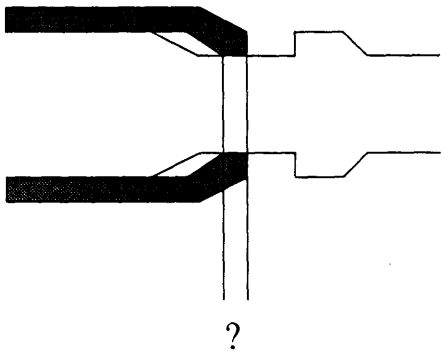


Figure 4.1: Strut-collar joint

these reasons, the stiffness properties are the most difficult properties to verify by measurement.

As for the mass and stiffness properties, dimensional properties of the structure can also be verified by measurement. For example, the dimensions used to compute the section properties for the elements, such as diameters, can be measured. Another example is the length of components, initially taken from drawings, which can also be verified through measurement. Other examples are components that are not found on engineering drawings. An example of this is the suspension attachment fixtures of MACE, long threaded rods with a universal joint on one end and connected to a node at the other end, Figure 3.10. The length of these components would not be known without measurement. Verification of the dimensional properties of the structure completes the component measurement phase of engineering insight.

Comparison of the initial model with experimental data might show that certain frequency regions are significantly in error. These frequency regions, and the modeling assumptions which could cause such errors, can be analyzed to improve the model. The proper assumptions to analyze come from experience gained in modeling structures. Some of the modeling assumptions that were examined for MACE were the effective bending length of the Lexan struts and the type of beam formulation used for the struts.

In previous versions of the hardware, the detail of the strut-collar joint, Figure 4.1, was examined to see exactly where the collar clamps down on the strut, determining the effective bending length of the strut. In the previous versions of the hardware, it

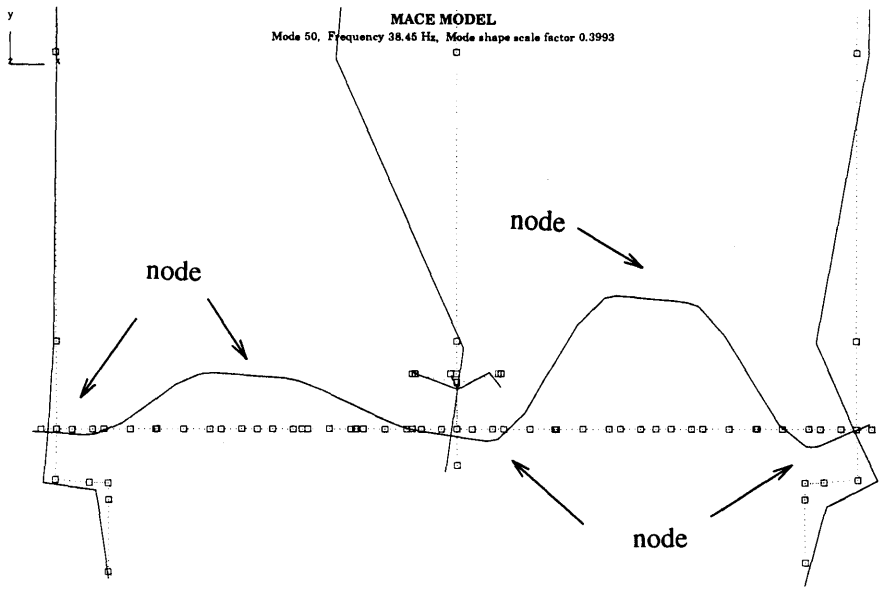


Figure 4.2: Modeshape showing large shear

was determined that the effective bending length of the strut was 6% longer than the exposed length of the strut [30]. Since the strut-collar joint did not change substantially in the different versions of the hardware, this 6% extension was carried over to the EM model. This was implemented by shifting the strut-collar joint nodal points to make the exposed strut length equal to the effective bending length, conserving the total test article length. Alternatively, this strut collar combination could be modeled in detail (3-D) and an equivalent element could be derived from this model.

Another assumption that was examined was the type of beam element formulation used for the struts. Several bending modes, one of which is shown in Figure 4.2, were found to have frequencies significantly higher than in the experimental data. An examination of these modeshapes showed a large amount of apparent shear in the struts. Note the large relative translation of the nodes without large node rotations (Figure 4.2). Initially, the struts were modeled using the Bernoulli-Euler beam formulation, which does not account for shear effects. With the indication of a large amount of shear in the struts, the strut beam element formulation was changed to a Timoshenko formulation which does account for local shear effects. To implement the Timoshenko formulation in the finite element code, all that was required was

the computation of a nondimensional shear factor, κ, which when multiplied by the cross-sectional area gives the effective shear area for the cross-section. These shear factors were computed using the following equation [16]:

$$\kappa = \frac{6(1+\nu)(1+m^2)^2}{(7+6\nu)(1+m^2)^2 + (20+12\nu)m^2} \tag{4.1}$$

where ν is Poisson's ratio and m is the ratio of the inner and outer diameters for a circular cross-section. When the Timoshenko beam formulation was used, the problem modes were significantly reduced in frequency.

These are not the only assumptions that were evaluated for validity, they are simply the ones that had a significant impact on the model results. Often this type of evaluation comes about in response to automated updating. For example, the automated updating routine will update to a certain point, beyond which it will not budge. It is at these points that evaluation of the modeling assumptions can have a significant impact on the modeling results. This is shown dramatically in the choice of finite element codes.

The initial model was generated using the ADINA [19] finite element code. During the course of automated updating, it was found that a couple of the modes that were significantly in error would simply not move. The modeling assumptions used in ADINA were examined again, but no invalid assumptions could be found. It was decided to verify the ADINA model by constructing an identical model in a different finite element code, NASTRAN [31]. When the NASTRAN model was constructed, a dramatic shift in the problem modes was seen, such that the frequencies of the model were nearly identical to those of the experimental data. It is unclear why this difference between finite element codes happened, as both codes use the same nonlinear, stress-stiffening procedure implemented in slightly different manners. Since the NASTRAN model is a better model than the ADINA model, it was decided to continue using NASTRAN.

Table 4.1 shows the frequencies of the model after engineering insight type updates have been implemented and before any automated updating has taken place. Note that the largest errors are in the suspension or gravity affected modes, such as the

Table 4.1: Comparison of frequencies for ground experimental data and NASTRAN 1-g input-output model after verification.

Description	Measurement Model Frequency (Hz)	Finite Element Model Frequency (Hz)	Frequency errors (%)
Y twist	0.24	0.23	-4.2
Z tilt	0.44	0.45	2.3
X rotation	1.39	1.40	0.7
1st Z bending	1.96	1.98	1.0
1st Y bending	3.38	3.35	-0.9
gimbal pend.	4.57	4.42	-3.3
gimbal pend.	4.79	5.31	17.1
gimbal pend.	5.93	6.06	2.2
gimbal pend.	6.75	6.56	-2.8
1st Violin	6.76	7.11	5.2
1st Violin	7.14	7.11	-0.4
1st Violin	7.19	7.15	-0.6
1st Violin	7.21	7.17	-0.6
2nd Y bending	9.29	9.23	-0.6
2nd Z bending	11.34	11.14	-1.8
3rd Z bending	12.11	12.84	0.6
2nd Violin	12.71	12.94	1.8
2nd Violin	14.29	14.31	0.1
2nd Violin	14.47	14.32	-1.0
2nd Violin	16.88	14.34	-15.0
3rd Y bending	17.06	17.23	1.0
3rd Violin	21.54	21.55	0.0
3rd Violin	21.71	21.55	-0.7
4th Y bending	22.63	21.99	-2.8
time delay	29.88	31.69	6.1
time delay	32.24	33.23	3.1
4th Z bending	39.05	39.26	0.5
5th Z bending	45.41	45.58	0.4
5th Y bending	48.55	48.66	0.2
6th Y bending	53.98	54.09	0.2

violin modes, with the largest error in a structural mode being less than 3%. The errors in the suspension modes are less significant because they will not be present in the 0-g model. These good results, before any automated update has been performed, show that further improvements using automated updating should result in excellent agreement with the experimental data.

At this point, the mass, dimensional properties, and major modeling assumptions of the finite element model have been verified. Engineering insight has not been exhausted though, as engineering insight is also used in the automated update techniques. Engineering insight is used to determine which parameters are uncertain and which optimization cost is most appropriate. Further engineering insight must be exercised to determine what size parameter changes are reasonable, as automated techniques might change parameters to unreasonable values.

4.2 Automated Updating

All automated update techniques consist of two parts: a cost statement and the sensitivity of the cost with respect to the parameters to be updated. The cost statement generates a scalar quantity which reflects the degree to which the model, formulated from a particular set of parameters, matches some desired behavior, such as the measured input-output response of the hardware. The sensitivities describe how each parameter influences the magnitude of the cost. Therefore, the sensitivities provide cost gradient information which is used in the update procedure to find the set of parameters which minimizes the cost, thereby creating the best match of the model to the desired behavior.

The eligible update parameters include all the parameters that make up the input-output model such as structural parameters, gains, time delays, *etc.* Engineering insight is used to determine which of these parameters are most uncertain and therefore should be included in the update procedure. Often, due to computational and storage constraints, many more possible parameters are identified than can actually be used. Therefore a method of reducing the number of update parameters, by ex-

amining the influence of the parameters on the cost and their orthogonality to each other, is desired.

4.2.1 Update Techniques

Automated updating can be performed using several different costs. These costs are differentiated by what sort of experimental data the finite element model is compared to: measurement model modal parameters [9], time responses, or frequency responses [32] (Step B in Figure 2.1). Each cost defines a different technique used in automated updating because the cost dictates what form of finite element model predictions will be updated: modal parameters; time histories; or frequency responses.

The first of these techniques is based on matching the finite element model modal parameters to those of a measurement model. Matching the modal parameters is useful because the frequencies and damping ratios are discrete parameters used to describe a dynamic model. The cost for modal frequencies is given in the following equation:

$$J = \sum_i \left(\frac{\omega_{identified_i} - \omega_{model_i}}{\omega_{identified_i}} \right)^2 \tag{4.2}$$

In this technique, a measurement model is fit to the experimental time or frequency response and the resulting modal parameters, usually frequencies and damping ratios but also possibly residues, are compared to the corresponding finite element modal parameters. The advantage of this technique is that it eliminates the need for computationally expensive time or frequency response calculations using the finite element model. One of the drawbacks is that the finite element modal parameters must be compared with the proper measurement model parameters, which may be difficult to determine for closely spaced modes that have different directions, such as suspension violin modes. A second drawback is that there is no guarantee that matching only the frequencies and damping ratios of a measurement model will cause the finite element model to better predict time or frequency responses.

The other two update techniques eliminate the need for a measurement model by operating on the raw time or frequency domain experimental data. The second

technique uses the time response of the system to a known input command. In this technique, the known command is used as an input to a time simulation using the finite element model. The resulting simulated time response is compared with the experimental time response to compute a cost given in the following equation:

$$J = \sum_{i=1}^{n} (y_{data}(t_i) - y_{model}(t_i))^2 \tag{4.3}$$

The advantage of this technique is that it operates directly on the experimental data, eliminating errors that might be present in a measurement model. Another advantage is that the problem with closely spaced modes in different directions has been eliminated, because the correct modes will appear in the proper time trace. One drawback of the time domain technique is that, in order to capture low frequency dynamics, long duration time responses may be required which take up large amounts of memory and require large computational effort to simulate. Another drawback is that measurement noise will be in the data traces.

The final technique is based on the frequency response of a system. In this technique the finite element predicted frequency response is compared to the experimental frequency response. A cost can be computed based simply on the difference in magnitude between the experimental and predicted frequency responses:

$$J = \sum_{i=1}^{n} \|G_{data}(j\omega_i) - G_{model}(j\omega_i)\|^2 \tag{4.4}$$

The problem with this cost, though, is that it weights the large magnitude portions of the frequency response (poles) very heavily and weights the low magnitude portions (zeros) hardly at all.

Since zeros are important for control design, they should be weighted as heavily as the poles. This is effected in a second cost based on the complex logarithm:

$$J = \sum_{i=1}^{n} \left\| \log \left(\frac{G_{data}(j\omega_i)}{G_{model}(j\omega_i)} \right) \right\|^2 \tag{4.5}$$

This second cost has a side benefit, namely that due to the nature of the complex logarithm, the phase of the transfer functions is also included in the cost.

The advantages of the frequency domain technique are the same as for the time domain technique. An additional advantage is that the amount of data required in the

frequency domain is much less than the time domain. This happens because to obtain the frequency domain data, the full length of time domain data from above is broken down into segments. An FFT is performed on each segment. These segments are then averaged, resulting in a reduction of the amount of data. A side effect of the averaging process is that the effects of noise present in the time response have been reduced. The drawback to this, though, is that the experimental frequency response functions are distorted by the windowing and averaging. This distortion usually manifests itself as an apparent increase in the damping of the poles and zeros [33].

Once the cost has been defined, the influence of the update parameters on the cost must be determined, which can be related to how the update parameters affect the input-output model. The influence on the cost can be determined by computing the updated model and then computing the updated cost. In the case of the structural parameters, they are included in the updated model by using some measure of how the update parameters affect the modal frequencies and modeshapes. The modal parameters are recomputed based on the updated parameters and then substituted into the input-output model. The non-structural parameters, such as damping ratios, are included directly into the input-output model. Two methods determining the effects of the structural parameters on the finite element modal parameters have been found: recomputation using an eigensolution and recomputation using linear sensitivities.

The first method is to perform a complete modal solution at each iteration, with the recomputed frequencies and modeshapes being combined with damping ratios to form the updated model. This method works well for small problems, but is inefficient when only a few modes are retained in the model and hence, need to be recomputed. The size of the eigensolution can be reduced by using a reanalysis approach to compute the updated frequencies and modeshapes [28].

In reanalysis, the full mass and stiffness matrices are updated using first order mass and stiffness matrix sensitivities:

$$\widetilde{M} = M_0 + \sum_i \Delta\alpha_i \Delta M_i \tag{4.6}$$

$$\widetilde{K} = K_0 + \sum_i \Delta\alpha_i \Delta K_i \tag{4.7}$$

where $\widetilde{M}$ and $\widetilde{K}$ are the updated mass and stiffness matrices, M_0 and K_0 are the nominal mass and stiffness matrices, and ΔM_i and ΔK_i are the mass and stiffness matrix sensitivities due to parameter α_i. The ΔM_i and ΔK_i matrices are obtained from a finite difference technique by varying the α_i parameter in the finite element model, forming the new mass and stiffness matrices, and differencing them with the nominal mass and stiffness matrices.

$\widetilde{M}$ and $\widetilde{K}$ are used to compute estimates of the updated modal frequencies:

$$(-\tilde{\Omega}\,\Phi_0^T \widetilde{M} \Phi_0 + \Phi_0^T \widetilde{K} \Phi_0)\Phi_R = 0 \tag{4.8}$$

$$\tilde{\Phi} \cong \Phi_0 \Phi_R \tag{4.9}$$

where $\tilde{\Omega}$ is the matrix of updated eigenvalues, ω^2, Φ_0 is the matrix of retained nominal modeshapes, Φ_R is the matrix of Ritz vectors, and $\tilde{\Phi}$ is the matrix of updated modeshapes. The resulting eigensolution has been reduced in size to the number of retained modes. The accuracy of this approximation increases as the number of retained modes increases.

The second method is to recompute the frequencies and modeshapes using first order sensitivities computed from finite differences, as in the following equation:

$$\begin{aligned} \tilde{\omega} &= \omega_0 + \sum_i \frac{\partial \omega}{\partial \alpha_i} \Delta\alpha_i \\ \tilde{\phi} &= \phi_0 + \sum_i \frac{\partial \phi}{\partial \alpha_i} \Delta\alpha_i \end{aligned} \tag{4.10}$$

where $\tilde{\omega}$ and $\tilde{\phi}$ are the updated frequencies and modeshapes, ω_0 and ϕ_0 are the frequencies and modeshapes of the initial model, and $\frac{\partial \omega}{\partial \alpha_i}$ and $\frac{\partial \phi}{\partial \alpha_i}$ are the finite difference approximations to the frequency and modeshape sensitivities for update parameter α_i. These sensitivities are computed from the finite element model by changing the α_i parameter in the model and recomputing the modal frequencies and modeshapes. These are differenced with the nominal frequencies and modeshapes and divided by $\Delta\alpha_i$ to obtain the sensitivities. One problem with the modeshape sensitivities is that due to the closely spaced nature of the modes, the ordering of the modes may change,

giving erroneous sensitivities. This can be solved by keeping the parameter changes in the model small. Another problem is that the perturbed modeshape may have changed sign, giving a sensitivity approximately equal to twice the original modeshape, which is clearly wrong. This is because the sign of the modeshape is arbitrary. A final problem is found in repeated modes, where any combination of a set of modeshapes is also a valid modeshape, making sensitivities for these modes meaningless. These difficulties are troublesome, but not insurmountable.

Both reanalysis and modal sensitivities have good and bad points. While reanalysis does tend to be more accurate than modal sensitivities, the need for the full order nominal and sensitivity mass and stiffness matrices requires them to be stored, taking up large amounts of active memory on the computer which may or may not be available. Another problem with reanalysis is simply obtaining the mass and stiffness matrices, which are often difficult to obtain from the finite element code. While the use of modal sensitivities eliminates the need to obtain and store the mass and stiffness matrices, the modal sensitivities often must be recomputed several times during the course of the update process before arriving at a final solution. Thus, the computational expense of recomputing the modal sensitivities must be weighed with the expense of a modal solution and storage of large matrices in order to determine which method to use.

In both the reanalysis and modal sensitivity methods, all the time and frequency responses of the finite element model must be recomputed each time the optimization makes a change in the parameters. This becomes very expensive when many iterations are required for convergence. A way around this expense leads to a method for including the updated parameters directly in the time or frequency domain costs: the cost sensitivity method. The chosen cost is written in vector form:

$$j = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_n \end{bmatrix} \tag{4.11}$$

where J_i is the cost evaluated at each time or frequency point. This vector cost can

be expanded to include multiple actuators and sensors by stacking the vectors for each actuator and sensor pair. With this definition of the cost, the scalar cost can be written as $j^T j$.

This method uses the perturbed finite element models, used to find ΔM_i and ΔK_i in reanalysis and compute the modal sensitivities. Once these perturbed finite element models have been run, input-output models using the perturbed frequencies and modeshapes are generated. The cost vector in Equation 4.11 is evaluated for the nominal model, j_0, and each perturbed model, j_i, with the corresponding sensitivity computed according to:

$$\frac{\Delta j}{\Delta \alpha_i} = \frac{(j_i - j_0)\alpha_i}{\Delta \alpha_i} \tag{4.12}$$

The resulting vector cost is updated by:

$$j = j_0 + \sum_i \frac{\Delta j}{\Delta \alpha_i} \Delta \alpha_i \tag{4.13}$$

Thus, the cost sensitivity method reduces the number of time or frequency response computations to be equal to just the number of update parameters plus one for the nominal model. Although the initial computation of the cost sensitivity method is much larger than for reanalysis and modal sensitivities, it is computationally more efficient in the long run. One of the added benefits of this method is that obtaining and storing the mass and stiffness matrices in the reanalysis method have been eliminated. In addition, all problems with the modeshape sensitivities have been eliminated. The drawback of this method, though, is that several recomputations of the cost sensitivities may be necessary during the course of the update.

From examining the costs, it is unclear which cost (measurement model, time domain, or logarithmic frequency domain) should be used. In order to make this determination, they should be tested on a sample problem to see which works best. In deciding which method of including the update parameters in the finite element model to use, either reanalysis or modal sensitivities can be used for small systems that do not have a lot of modal overlap. For larger systems that have repeated modes, the cost sensitivity method should be used to avoid problems with storage and modeshape sensitivities in the reanalysis or modal sensitivity methods.

4.2.2 Parameter Selection

Some care must be exercised when choosing which parameters to use in the update. First, some parameters have similar effects on the predictions of the model making the solution non-unique. Second, it is likely that only a subset of the available parameters will affect the cost in a significant manner. Third, the computational expense increases with the number of parameters. Therefore, it is desirable to reduce the number of update parameters by examining how each of them affects the cost.

The first step is to determine which model parameters have significant uncertainty. Following that, the cost sensitivities computed above can also be used to reduce the number of candidate parameters. These sensitivities describe how a change in a particular parameter changes the cost. Since these sensitivities are vectors, standard vector manipulation, such as dot products, can be used to indicate the proper parameters to use in the automated update. The sensitivities can be used to show the orthogonality of the sensitivity vectors between parameters. This orthogonality shows if the parameters have similar influences on the cost. If several parameters have sensitivities in the same direction, the updating procedure will not work very well as there is not a unique set of parameters that minimizes the cost, and some of the parameters should be discarded. The alignment of the sensitivity vectors with the initial cost vector, j_0, is used to show the magnitude of the parameter's influence on the cost. If a parameter's sensitivity is orthogonal to the initial cost or the parameter is aligned with the initial cost but has only a small influence on the cost, this parameter should be discarded.

To see these properties, begin by forming the matrix of sensitivities, ΔJ:

$$\Delta J = \left[\begin{array}{cccc} \frac{\Delta j}{\Delta \alpha_1} & \frac{\Delta j}{\Delta \alpha_2} & \cdots & \frac{\Delta j}{\Delta \alpha_N} \end{array} \right] \tag{4.14}$$

To see the orthogonality between the sensitivities of the update parameters, the matrix of sensitivities needs to be normalized so that the sensitivities have the same length:

$$\Delta \overline{J} = \left[\begin{array}{cccc} \frac{\Delta j}{\Delta \alpha_1} / \| \frac{\Delta j}{\Delta \alpha_1} \| & \frac{\Delta j}{\Delta \alpha_2} / \| \frac{\Delta j}{\Delta \alpha_2} \| & \cdots & \frac{\Delta j}{\Delta \alpha_N} / \| \frac{\Delta j}{\Delta \alpha_N} \| \end{array} \right] \tag{4.15}$$

Now examine the matrix $\left| \Delta \overline{J}^T \Delta \overline{J} \right|$. This matrix represents the dot products of each

parameter sensitivity with every other parameter sensitivity. The diagonal of this matrix will be unity, since the sensitivity vectors have been normalized, and the magnitude of the off-diagonals will lie between zero and unity. Off-diagonal values near zero indicate that the sensitivities of the two parameters are nearly orthogonal. The ideal situation would be that this matrix is the identity matrix, indicating that the parameters are perfectly orthogonal to each other. If some of the off-diagonal values are near unity, this indicates that the sensitivities of the two corresponding parameters are aligned in the same direction. If this turns out to be the case, the best course of action is to choose the parameter that has the most influence on the initial cost, to choose the more uncertain parameter, or combine them into a single update parameter if they are connected.

The second test of the update parameters is to compare them with the initial cost, j_0, to see whether they will influence this cost in the update procedure. To do this, take the dot product of the initial cost vector with each sensitivity vector:

$$\|\Delta J_0\| = \left| j_0^T \Delta J \right| \tag{4.16}$$

This test includes both orthogonality and parameter strengths. If a parameter is orthogonal to the initial cost, the value of the dot product will be small and the parameter should be discarded. If the parameter is aligned with the initial cost but has a very weak effect on the cost, the value of the dot product will again be small and the parameter should be discarded. Thus large values of $\|\Delta J_0\|$ indicate the parameters that should be retained. However, it may be useful to normalize the parameter sensitivities by their maximum allowable perturbations in order to properly scale this test.

Engineering insight provides a list of candidate update parameters based on which parameters are most suspect in the input-output model. Using the above tests, the candidate parameters can be ranked based on their ability to influence the cost and their orthogonality. The best of these parameters should be retained for the automated update and the rest should be discarded.

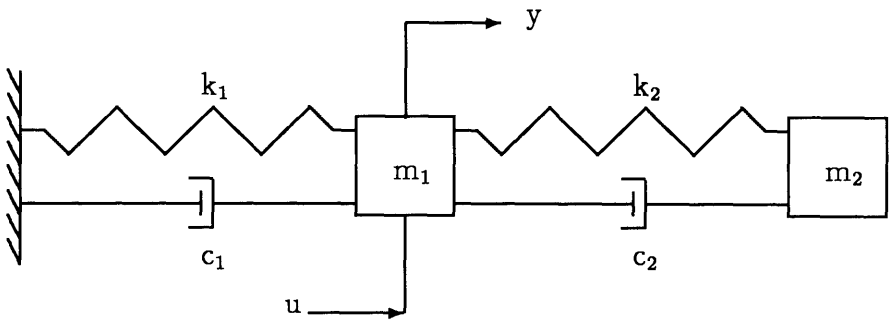


Figure 4.3: Collocated mass-spring-damper system

4.3 Automated Updating Sample Problem

Before attempting automated updating on the full MACE model, some thought must be put into which update technique to use. For this reason, the automated update techniques from the previous section are implemented on a low order problem in order to find out which update techniques give the best results. The system shown in Figure 4.3 is used with uncertain spring stiffnesses. These spring stiffnesses, with 'actual' values of k_1=0.75 and k_2=1.20, are initially modeled incorrectly with initial values of k_1=k_2=1. This system is the lowest order system with all the essential features of a typical structural transfer function, namely interlaced poles and zeros. Thus, this system allows the effects of transfer function zeros on the update process to be examined.

Five cases were considered:

1. Measurement model cost with modal sensitivities
2. Time domain cost with
   a. reanalysis, identify ζ
   b. modal sensitivities, identify ζ
3. Frequency domain cost with
   a. reanalysis, identify ζ
   b. modal sensitivities, identify ζ

In case 1, experimental values for the modal frequencies and damping ratios (ζ) are determined by fitting a measurement model to an 'experimental' transfer function. This 'experimental' transfer function is determined from the input (u) and output (y) time response of the exact system (Figure 4.3) with additive white noise. The experimental modal frequencies are used to update the analytical modal frequencies. Once updated, they are combined with the experimental damping ratios to form the

updated input-output model.

In case 2, a white noise input (u) is put into the model of the exact system and the resulting output time response (y) is computed. To this time response, a white measurement noise is added, simulating an experimental time response. To update the model, the same white noise signal is put into the current analytical model and the predicted time response is computed. The error between the two time responses is used to update the model.

In case 3, the experimental transfer function of case 1 is used again. In this case, though, the raw 'experimental' transfer function is used to update the model by minimizing the difference from the analytical transfer functions as calculated using the logarithmic cost in Equation 4.5.

In these cases, two methods are used to include the updated parameters in the analytical model. The first method is a different form of reanalysis from that given in Equation 4.8. This form differs from normal reanalysis [28] in that, due to the low order nature of the problem, the mass and stiffness matrix sensitivities are not used to update the mass and stiffness matrices. The resulting frequencies and modeshapes are combined with damping ratios from a measurement model fit to form the updated model. The second method is to recompute the frequencies and modeshapes using modal sensitivities approximated using a finite difference method. Typically, the sensitivities are computed, then an update is performed. Once the update is complete, the finite differences are recomputed about the updated parameters and the system is updated again. This process is repeated until the change in updated parameters is small. Again, the damping ratios are taken from a measurement model. Because of the small size and low modal overlap of this problem, the sensitivities of the costs to the update parameters were not considered.

The results of these cases will be discussed from two viewpoints: comparison of cost statements and comparison of the methods for including the updated parameters. Table 4.2 lists the actual and updated parameters, including the spring stiffnesses, k_1 and k_2, resulting modal frequencies, ω_1 and ω_2, identified damping ratios, ζ_1 and ζ_2, and zero frequency and damping ratio, ω_z and ζ_z. The table shows that all the

Table 4.2: Converged parameters

Case	Actual	1	2a	2b	3a	3b
k_1 (N/m)	0.7500	0.7487	0.7483	0.7483	0.7535	0.7535
k_2 (N/m)	1.2000	1.2000	1.1998	1.1988	1.1894	1.1894
ω_1 (rad/sec)	0.5637	0.5633	0.5632	0.5632	0.5644	0.5644
ω_2 (rad/sec)	1.6829	1.6827	1.6825	1.6825	1.6774	1.6774
ζ_1	0.0035	0.0101	0.0034	0.0034	0.0106	0.0106
ζ_2	0.0077	0.0086	0.0077	0.0077	0.0090	0.0090
ω_z (rad/sec)	1.0954	1.0955	1.0954	1.0954	1.0906	1.0906
ζ_z	0.0046	0.0080	0.0053	0.0053	0.0084	0.0084

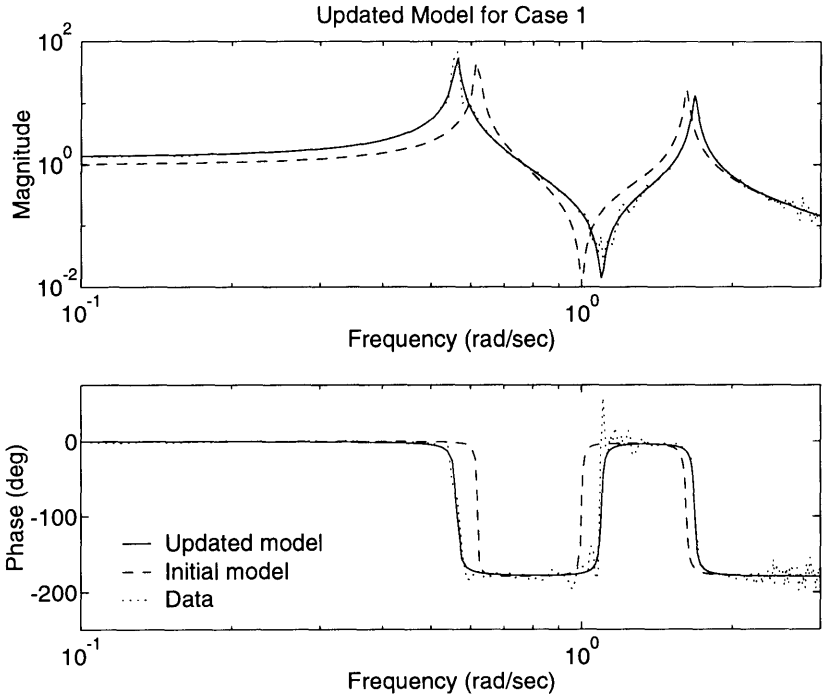


Figure 4.4: Updated transfer function for case 1

updates worked well. Figures 4.4 through 4.6 show the results for cases 1, 2b, and 3b. These cases cover each of the costs with updated model parameters included using modal sensitivities. As expected, the transfer functions all show excellent agreement with the experimental data.

In comparing the three cost statements, the table shows that all three give very good results for the springs. The measurement model (1) and time domain cases (2) tend to underestimate k_1, while the frequency domain cases (3) tend to overestimate

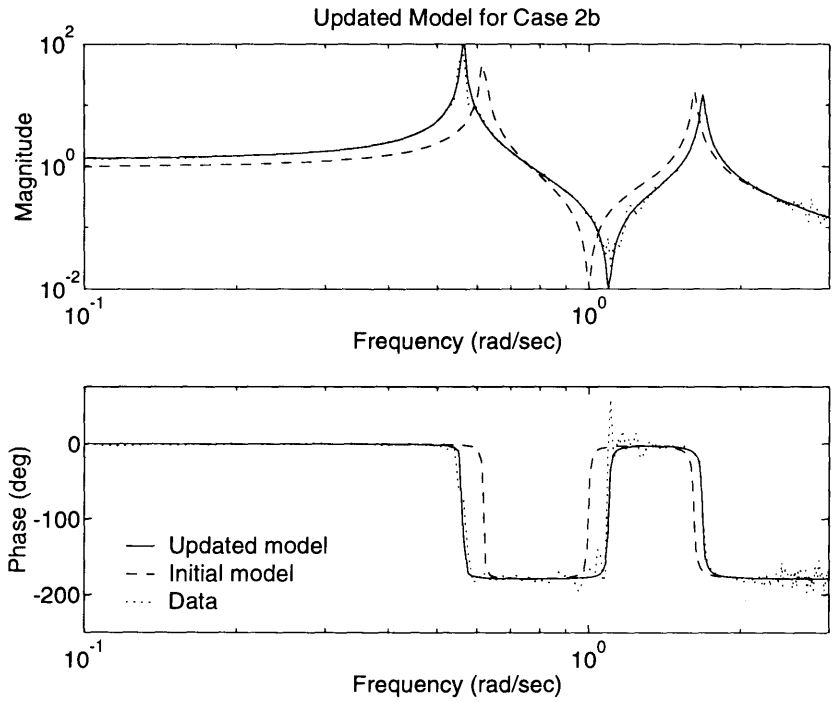


Figure 4.5: Updated transfer function for case 2b

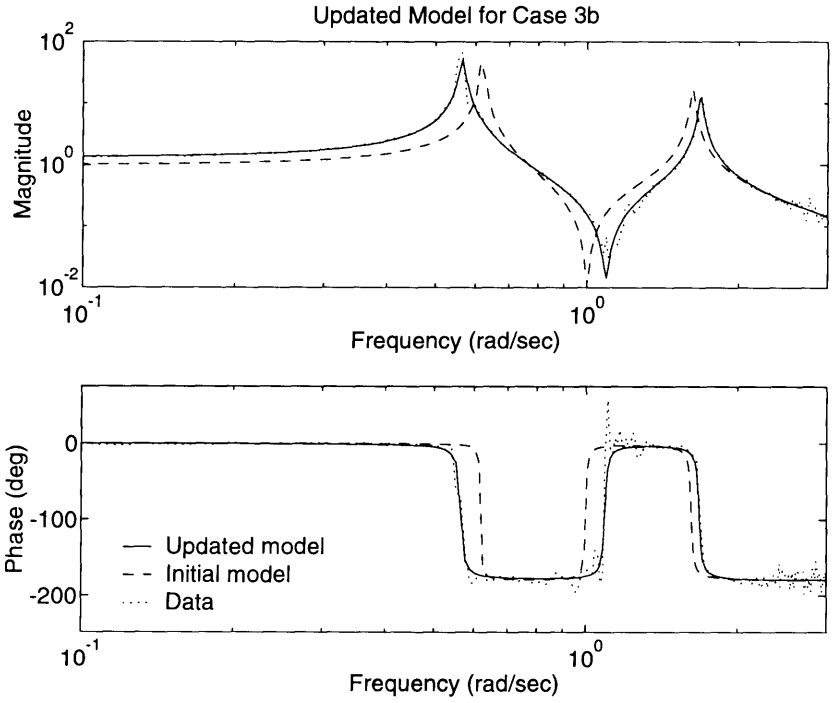


Figure 4.6: Updated transfer function for case 3b

k_1. All cases tend to underestimate k_2, with the frequency domain case underestimating k_2 more than the other two cases. These same trends are also seen in the analytical predictions of the modal frequencies, ω_1 and ω_2. The big difference between the cases is seen in the identified damping ratios. The table shows that the time domain cases all identified ζ_1 and ζ_2 nearly exactly, while the measurement model and frequency domain cases identify too much damping.

It is interesting to note that in both of these costs, the damping ratios are identified from experimental transfer functions. Windowing and averaging effects often cause an apparent increase in the damping of experimental transfer functions [33]. Another effect that can increase the apparent damping in the poles and zeros is the density of data points at the poles and zeros. Unless a data point lies exactly at the pole or zero frequency, the damping will always appear too large and may also cause an apparent shift in the pole or zero frequency [33]. Since the identified damping of both the poles and the zero are consistently too high, this indicates that these effects are indeed being seen in the measurement model and frequency domain cases.

These results indicate that it does not really matter which cost is used, as they all achieve very good results. The results seem to indicate that the time domain case is best suited for identifying the damping ratios. The time domain case has some drawbacks, though, the first of which is that the time cost does not necessarily place equal emphasis on the poles and zeros, as does the frequency domain case. Another factor is that most control design techniques are frequency domain techniques, and as such, closed-loop stability will be predicted from experimental transfer functions, not experimental time responses. Thus, it is best to match the experimental transfer functions. Since the measurement model is based on the experimental transfer function, the measurement model and frequency domain cases are recommended for further consideration.

In order to determine which method works best for including the updated parameters into the model, Cases 3a and 3b are considered. The table shows that both of these methods give excellent results, with reanalysis and modal sensitivities giving identical results. This indicates that reanalysis and modal sensitivities are equivalent

Table 4.3: Converged parameters with separate stiffness and damping updating

Case	Actual	3a	3b
k_1 (N/m)	0.7500	0.7535	0.7536
k_2 (N/m)	1.2000	1.1893	1.1893
ω_1 (rad/sec)	0.5637	0.5644	0.5644
ω_2 (rad/sec)	1.6829	1.6774	1.6774
ζ_1	0.0035	0.0106	0.0106
ζ_2	0.0077	0.0090	0.0090
ω_z (rad/sec)	1.0954	1.0906	1.0906
ζ_z	0.0046	0.0084	0.0084

methods.

One final issue in automated updating that needs to be addressed is whether the stiffness and damping must be updated together or whether they may be updated separately. To examine this, cases 3a and 3b are repeated with the stiffnesses and damping ratios updated separately (Table 4.3). As can be seen in the table, the resulting values are nearly identical to the previous cases. These results indicate that updating the stiffness and damping separately is a satisfactory method. This is good because the most significant errors in the model are frequency errors which can be handled separately from damping errors, since the latter are normally taken directly from the measurement model. The only complication to this situation comes when the system to be updated is actually a closed-loop system and the open-loop damping ratios are to be identified, as is the case for the MACE model. In this case, errors in the open-loop damping can cause errors in the closed-loop frequencies and *vice versa*. Normally, though, stiffness dominates the closed-loop frequencies and open-loop damping along with the control dominate the closed-loop damping. This can be verified by computing the sensitivities of the closed-loop frequencies and damping ratios to the open-loop damping ratios and stiffnesses, respectively. If these sensitivities are small, the stiffness and damping can be updated separately. If not, they need to be updated together.

The implication of this sample problem for MACE is to conduct the update by either matching a measurement model and/or matching experimental frequency re-

sponses. Since both reanalysis and modal sensitivities give the same answer, either method is suitable for MACE, subject to storage and computational expense considerations. It is unclear which method is computationally more efficient, as this depends on the size of the model, the number of retained modes, and the number of parameters. In fact, neither method may actually be used because the cost sensitivity method avoids problems that are inherent in both methods. Finally, the stiffness and damping should be updated separately in order to speed up the stiffness update by reducing the total number of parameters. The easiest method for updating the damping ratios is to take them directly from the measurement model. This is most efficient as it eliminates all the transfer function computations.

4.4 Nonlinearity

A nonlinearity has been discovered in the MACE test article which affects the automated updating effort. This nonlinearity is centered on the actuators and the gimbal servos. The actuators are classified according to their ability to excite the test article. These groups are, in order of decreasing strength: the gimbals, the reaction wheels, and the active strut.

The nonlinearity was first noticed in the active strut transfer functions, Figure 4.7. When the experimental transfer functions were compared with the input-output model containing the gimbal servos (solid line), much more damping was seen in the modeled transfer functions than was evident in the experimental transfer functions, especially in the 9 to 20 Hz range. Since the damping of these heavily damped modes was dominated by the gimbal servos, it was thought that the experimental transfer functions looked like they were unservoed, which was not the case. This suggested that the mechanism for the nonlinearity was friction in the gimbals. The active strut, since it is so weak, is incapable of exciting the gimbals enough to overcome friction, effectively locking the gimbals. Since the gimbals are not moving, the servos do nothing and the resulting transfer functions appear lightly damped. To verify this theory, the 1-g finite element results were recomputed with the gimbals

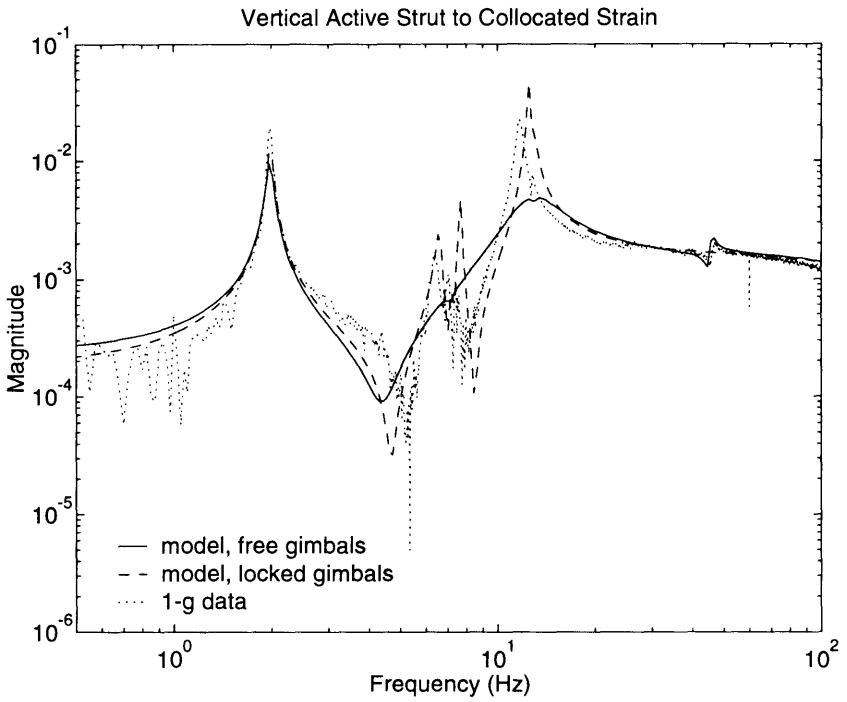


Figure 4.7: Active strut transfer function

locked and an input-output model generated. This resulting model is also plotted in Figure 4.7 as the dashed line. The figure shows that the model with locked gimbals has the same damping attributes as the data, confirming the theory about the effect of the nonlinearity.

Since this nonlinearity is due to friction in the gimbals, it should also be evident in the reaction wheel transfer functions. Since the reaction wheels are capable of much more structural excitation than the active strut, they should be able to overcome the gimbal friction. Thus, to investigate the effect of the nonlinearity in the reaction wheel transfer functions, the reaction wheels were excited at several input levels. One of these sets of transfer functions is shown in Figure 4.8 for the X-axis reaction wheel to payload X-axis rate gyro. In these transfer functions, heavily damped behavior is seen in the 6 to 9 Hz range, indicating that the servos are, indeed, having an effect. The transfer functions for the three excitation levels show that the lowest excitation level has the lightest damping behavior and the highest excitation level has the heaviest damping behavior. Since lower excitation levels are expected to produce less motion in

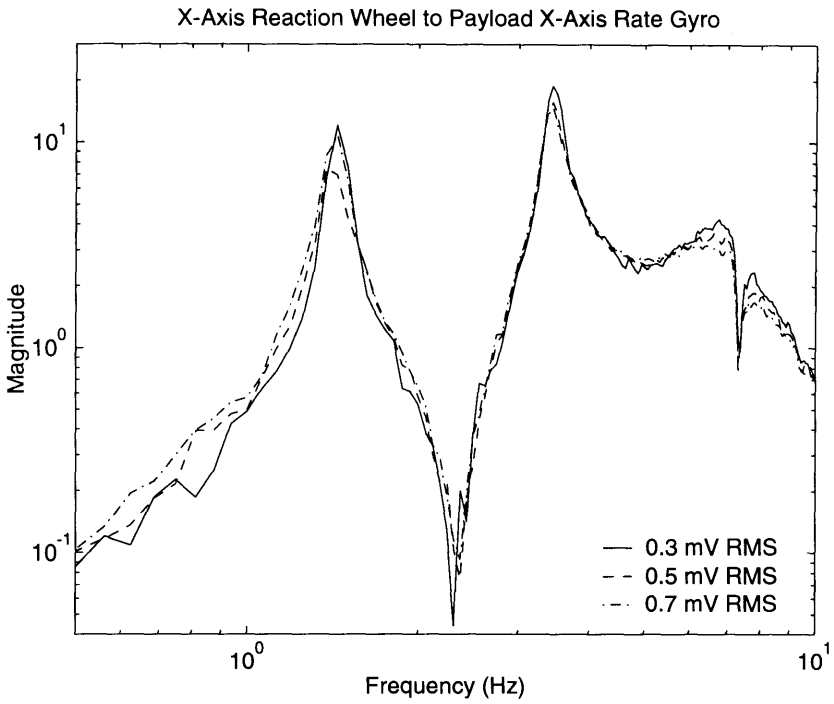


Figure 4.8: Reaction wheel transfer functions

the gimbals, allowing friction to have more effect, the nonlinearity should cause these excitation levels to exhibit lighter damping behavior, which is indeed the case. This gives further credence to the explanation that the nonlinearity is caused by gimbal friction. Another effect of the nonlinearity is seen in the lighter damped modes at 1.5 and 3.3 Hz, where a decrease in frequency is seen along with the increase in damping.

This nonlinearity governs which set of experimental data to use in the automated update. Stated more explicitly, care must be exercised when choosing which actuators and excitation levels to include in the update. The nonlinearity might cause the data to indicate several frequencies and damping ratios. This was seen in the measurement model, where several modes were used to capture the nonlinearity. These conflicting modes might confuse the update procedure, possibly leading to no update at all. To lessen the effects of the nonlinearity on the update procedure, the structural responses to the gimbals should be emphasized. The gimbals should be used because they actuate directly on the friction and hence the nonlinearity should not be as strong.

4.5 Automated Updating of MACE

With the model assumptions and parameters verified by engineering insight, all further open-loop updates to the model are made using automated techniques. Using the results of the previous sections, the automated updating cost used for the MACE model is the logarithmic cost:

$$J = \sum_{\omega} \sum_{actuators} \sum_{sensors} \left\| \log \left(\frac{G_{data}(j\omega)}{G_{model}(j\omega)} \right) \right\|^2 \tag{4.17}$$

With the cost defined, all that is left to complete the setup of the update is to choose how the update parameters should be included in the model and which parameters to use in the update. In order to save computational time and since the cost sensitivities must be computed for parameter selection, the update parameters will be included into the cost using the cost sensitivity method, Equation 4.13. The eligible update parameters in the finite element model include all the parameters that make up the finite element model such as element lengths, masses, and stiffnesses, and parameters of the input-output model, such as damping and time delays.

Since the length and mass properties of the structure have been verified through engineering insight, the physical update parameters are restricted to the stiffness parameters. Some care must be exercised when choosing which parameters to use in the update, as some parameters have identical effects on the stiffness of an element, such as the material and section properties. In order to simplify the choice, the update parameters are further restricted to only the Young's modulus (E) of the element materials and the stiffnesses of the suspension system, giving a total of 13 possible update parameters: $E_{lexan\ end}$, $E_{lexan\ center}$, $E_{stainless\ steel}$, $E_{aluminum}$, $E_{cable\ 1}$, $E_{cable\ 2}$, $E_{cable\ 3}$, $E_{active\ strut\ end}$, $E_{active\ strut\ piezo}$, $E_{active\ strut\ center}$, and the suspension spring stiffnesses, $k_{device\ 1}$, $k_{device\ 2}$, and $k_{device\ 3}$. These parameters are judged using the criteria presented in Section 4.3.2.

In order to use the tests of Section 4.3.2, the cost sensitivities must be determined. To find these sensitivities, the cost of Equation 4.17 is rewritten in vector form according to Equation 4.11. This vector cost can be plotted to show the frequency spectrum of the logarithmic cost. Figure 4.9 shows this plot for the primary X-axis

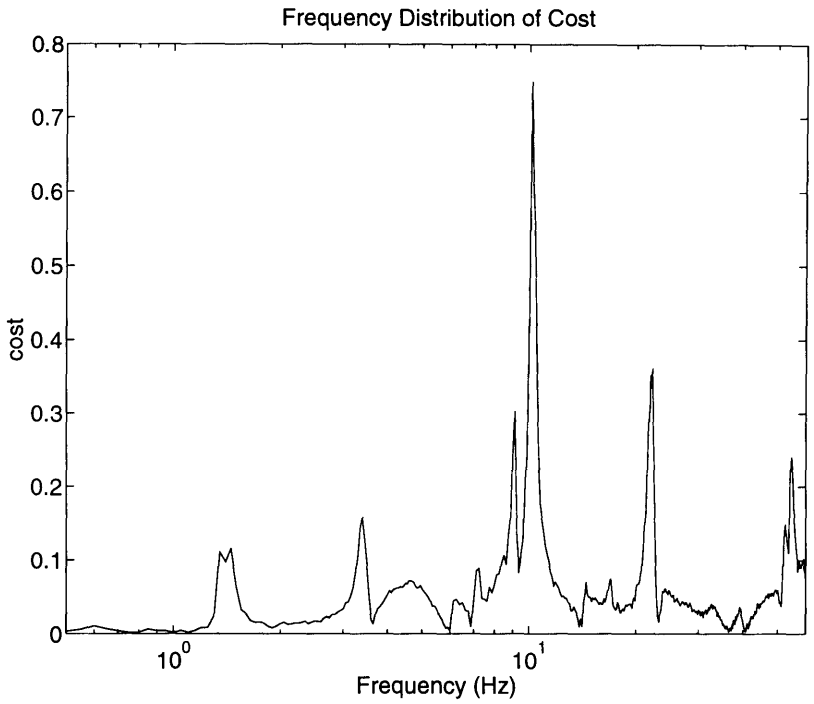


Figure 4.9: Typical cost spectrum

gimbal to payload X-axis rate gyro transfer function. This plot shows explicitly the regions of large cost in the frequency range of interest. These regions may not be very evident in the comparison of the predicted and experimental transfer functions, Figure 4.10. The large spike in the frequency spectrum is due to the frequency error in the zero at 10 Hz. The cost sensitivities can then be found from Equation 4.12.

The first test, the orthogonality test, is shown in Figure 4.11. This test shows that most of the parameters are orthogonal to each other with several exceptions, parameters 1 and 2, parameters 3 and 4, and parameters 8, 9, and 10. These exceptions make sense, as parameters 1 and 2 are the passive strut Lexan^{TM} elasticities at the center and end, which should have similar directions. Parameters 3 and 4 are the aluminum and stainless steel moduli of elasticity, which are much stiffer than the rest of the structure and should also have similar directions. Parameters 8, 9, and 10 are the active strut Lexan^{TM} end, center, and piezo elasticities and again should have similar directions. No parameters are discarded here, until their influence of the on the initial cost has also been determined.

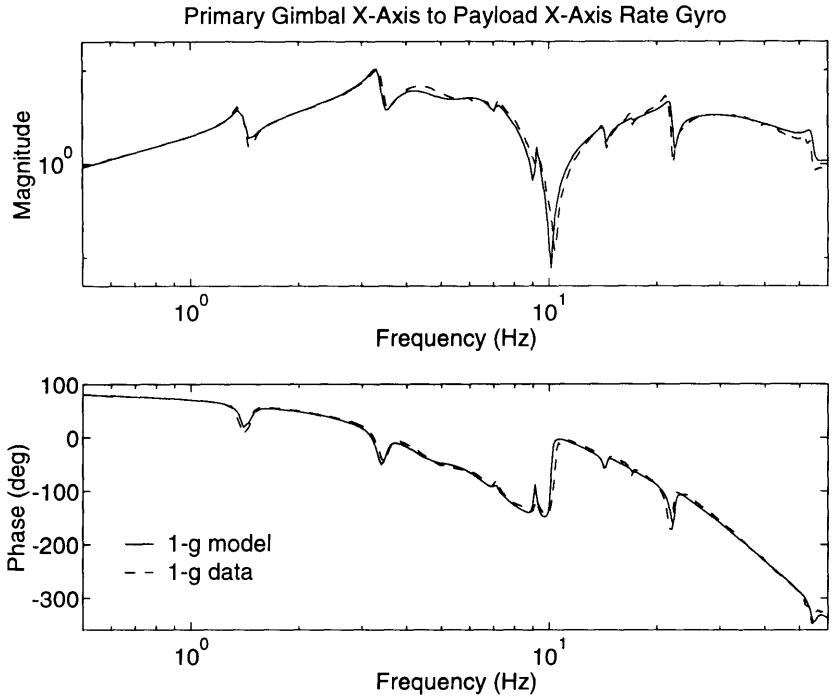


Figure 4.10: Typical cost spectrum

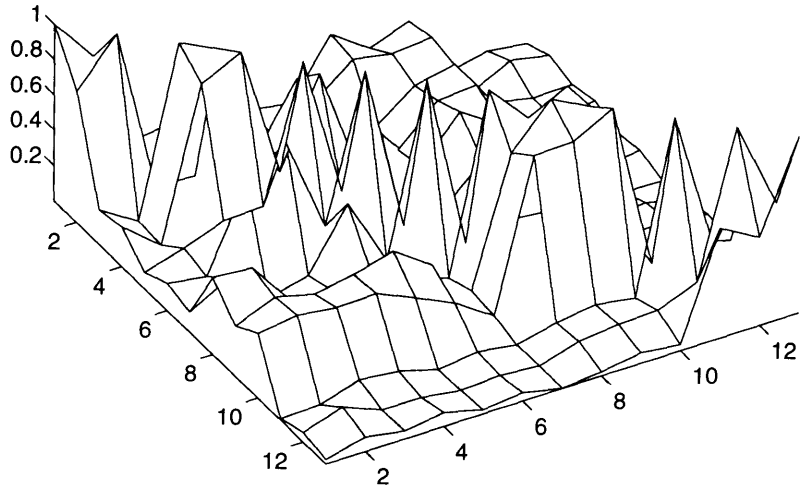


Figure 4.11: Orthogonality test

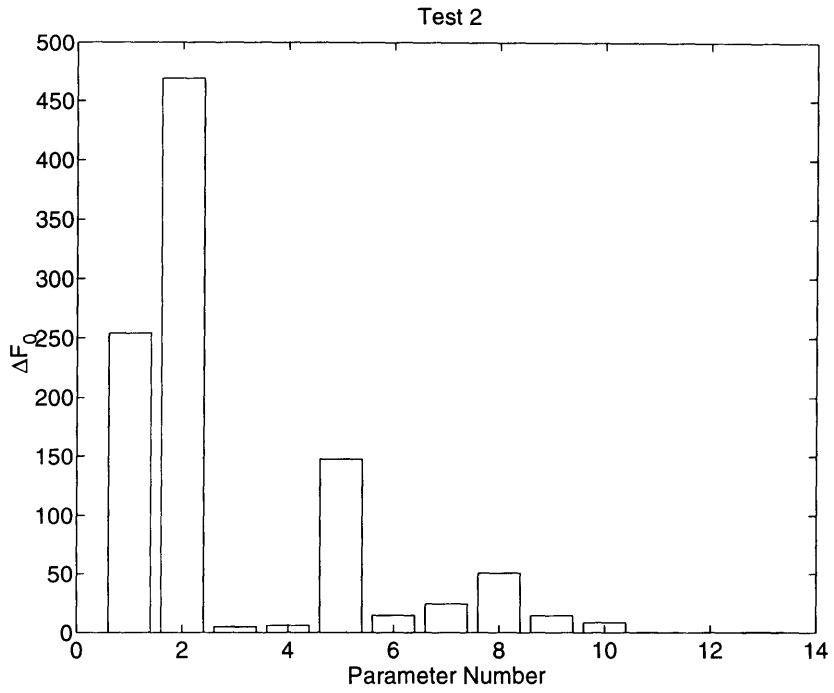


Figure 4.12: Strength test

The second test is the more important test as it relates how the candidate parameters affect the initial error. From Figure 4.12, it is clearly seen that the last three parameters, the suspension stiffnesses, have virtually no effect on the initial error and should definitely be discarded. It is also seen in the figure that parameters 3 and 4 are slightly smaller than parameter 10. These two parameters, aluminum and stainless steel, should also be discarded. Parameter 10 should be retained because it is one of the active strut parameters, which should all be retained or all discarded. As a result of this test, the moduli of aluminum and stainless steel, along with the three suspension device stiffnesses, have been discarded, thereby reducing the number of parameters from 13 to 8.

In examining the retained parameters, it is seen from an engineering insight standpoint that they contribute the most to bus flexibility (lexan and active strut stiffnesses) and suspension flexibility (cable stiffnesses). Since most of the errors are in the flexible behavior of the structure, such parameters would be expected to have the most influence and hence be retained.

The automated updating uses an iterative nonlinear least squares routine found in MATLAB [34, 35]. The transfer functions used for the stiffness update are the principal gimbal to rate gyro transfer functions, meaning the X-axis primary and secondary gimbal inputs to all of the X- and Y-axis rate gyros and the two Z-axis gimbal axes to the Z-axis rate gyros. These transfer functions are used because they are the strongest in magnitude and hence cleanest in noise. The reaction wheel inputs are not used because of the nonlinearity in the test article. This nonlinearity results from friction in the gimbals and the weakness of the reaction wheels as actuators, manifesting itself as a shift upwards in the frequencies and lessening of the damping (relative to the gimbal transfer functions) in some of the modes. Since the gimbals are used to do much of the control, the gimbal transfer functions are used in the update. The frequency range used in the update is from 0.5 Hz to 60 Hz because it is above the scatter in the data at low frequency, but is low enough to capture some of the suspension modes. In addition, the bandwidth is below the effects of the roll-off of the rate gyros, Bessel filters, *etc.* at higher frequency.

One final concern that must be addressed regarding the update is how much a parameter is allowed to vary. If some limits are not placed on the parameters, the optimization will possibly make unrealistic changes, *e.g.* $> \pm 100\%$ changes, which could cause negative stiffness values. A simple solution to this problem is to augment the cost vector with the update parameters:

$$\tilde{j} = \begin{bmatrix} j \\ W\Delta\alpha \end{bmatrix} \tag{4.18}$$

where $\Delta\alpha$ is a vector of the changes in the update parameters and W is a weighting matrix. This forces the optimization to consider the changes it makes to the model. This weighting makes a 1% decrease in the cost worth a γ% parameter change. Although this eliminates the need for closely monitoring the updated parameters, care must still be paid to the choice of weightings.

With the augmentation of the cost vector, all concerns regarding the automated update have been addressed and the physical parameters can be updated. The physical parameter update was run once resulting in a 6% decrease in the cost. The

Table 4.4: Updated material properties

Parameter	Young's Modulus (N/m^2)	change (%)
$E_{lexan\ end}$	2.628×10^9	-1.3
$E_{lexan\ center}$	2.335×10^9	-4.9
$E_{active\ Strut\ end}$	2.767×10^9	4.0
$E_{Active\ Strut\ piezo}$	3.783×10^9	-1.0
$E_{Active\ Strut\ center}$	2.662×10^9	4.7
$E_{Cable\ 1}$	1.982×10^{11}	8.9
$E_{Cable\ 2}$	1.712×10^{11}	-5.9
$E_{Cable\ 3}$	1.842×10^{11}	1.2

updated parameters were substituted back into the finite element model and the sensitivities were recomputed about these new parameters. The physical parameters were updated once more, resulting in a further 5% decrease in the cost. The updated physical parameters along with their net changes are summarized in Table 4.4.

In examining these changes, it is seen that the largest change in the bus parameters is in the Young's modulus of $lexan_{center}$. This makes sense as the bus is composed predominantly of this material and changes in this material should have a large effect on the cost. The active $strut_{end}$ and the active $strut_{center}$ parameters also underwent rather large changes. The active strut parameters are important because they control the stiffness of the active strut. The suspension cable parameters also underwent fairly large changes. These parameters are important because they control the suspension violin behavior, which is difficult to accurately capture in the model.

Although the physical parameters have been presented separately, in the actual update process, they are updated in sequence with time delays and damping ratios. First the physical parameters are updated, followed by the time delays, and then the damping ratios. Then the process is repeated. The time delays used in the update are the time delays used in the servo loop and the time delays associated with the Bessel filters and the rate gyro dynamics. Due to uncertainties in how the servo loop is closed on the model, it was deemed appropriate to update the servo time delays. The sensor time delays are updated just to match the phase characteristics of the updated transfer functions. Once the gimbal time delays are updated, the servo and

Table 4.5: Updated actuator time delays in milliseconds

Prim. X Gimbal (msec)	Prim. Z Gimbal (msec)	Sec. X Gimbal (msec)	Sec. Z Gimbal (msec)	Reaction Wheel X (msec)	Reaction Wheel Y (msec)	Reaction Wheel Z (msec)
4.56	3.97	4.76	4.11	6.74	6.99	5.64

Table 4.6: Updated rate gyro time delays in milliseconds

Payload X	Payload Z	Bus X	Bus Y	Bus Z
6.38	6.12	5.41	5.25	6.43

rate gyro time delays are fixed and the phase of the reaction wheel transfer functions was matched by updating the reaction wheel time delays. Tables 4.5 and 4.6 give the updated time delays for the actuators and the rate gyros, respectively. While these updated time delays do not necessarily represent the correct allocation of the time delays, they do represent the proper net time delay in each transfer function and are the best fit of the actual servo loop using only the servo time delays as the update parameters, other errors such as improper gains notwithstanding. The damping ratios are updated because they can only be obtained from experimental data.

The automated updating steps described in this section have reduced the logarithmic cost from 552.91, for the NASTRAN model resulting from engineering insight, to 399.08, for the updated model. This resulted in a 27.8% reduction. It is likely that further updating could reduce the cost a few more percent, but the majority of the updating has already been accomplished with engineering insight and the automated update.

Two typical updated transfer functions are shown in Figures 4.13 and 4.14, showing the changes in the transfer functions caused by the update. These transfer functions show small changes in the frequencies and damping of some of the poles. The most notable changes are seen in some of the zeros, where they have changed from minimum phase to nonminimum phase and *vice versa*. These changes occur near 60 Hz in Figure 4.13 and near 0.8 and 15 Hz in Figure 4.14. These changes are likely due to the updating of the servo time delays which have a significant effect on the damping and zeros.

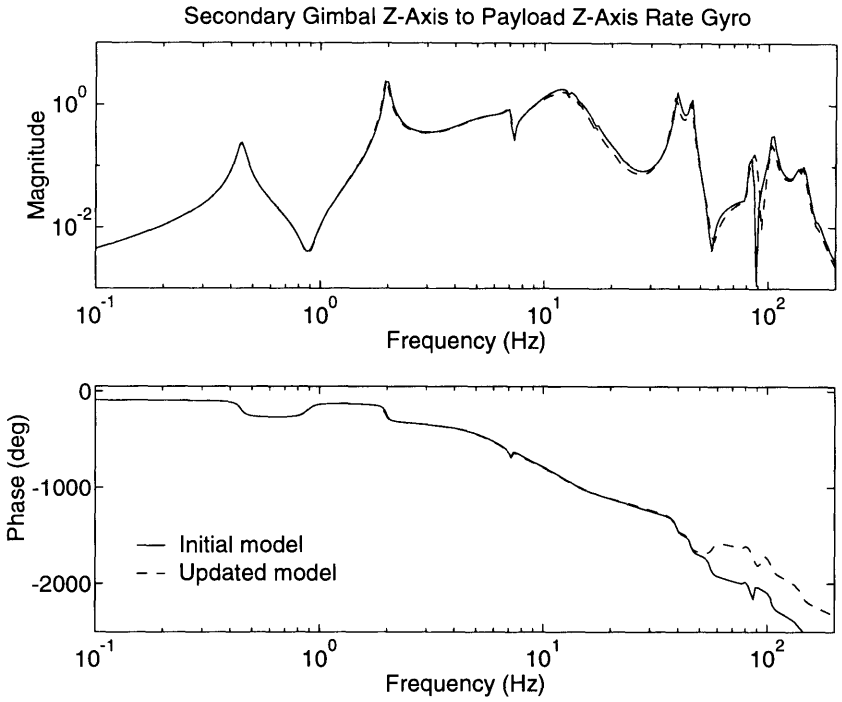


Figure 4.13: Typical updated in-plane transfer function

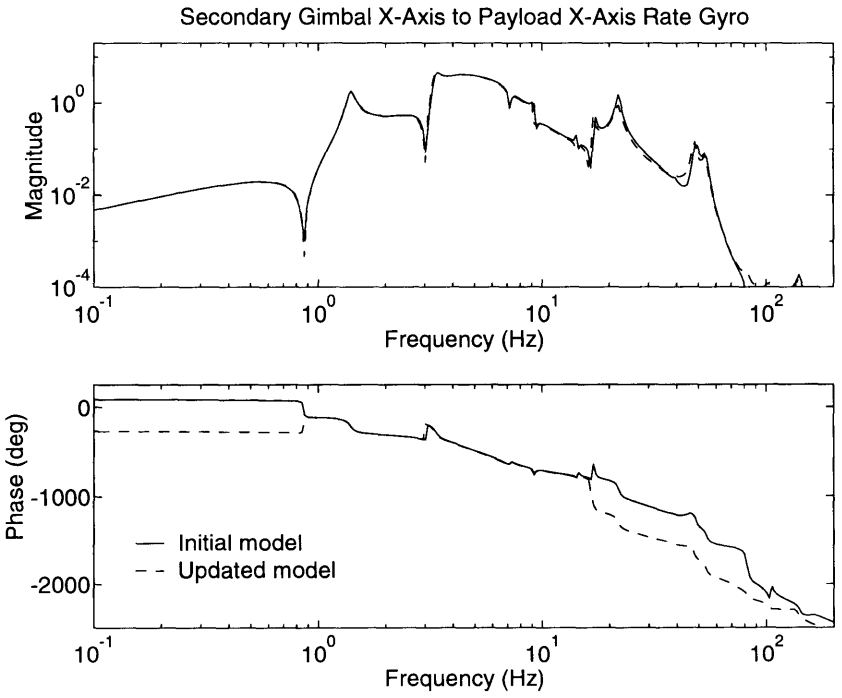


Figure 4.14: Typical updated out-of-plane transfer function

The automated updating described in this section completes the open-loop portion of the model updating procedure. Next, the finite element input-output model must be closed-loop verified by designing controllers based on the 1-g finite element model and implementing them on the test article.

4.6 Closed-Loop Updating

Ideally, all closed-loop experimentation using the finite element model should be done after all automated updating has been completed, as this model represents the best finite element model achievable in an open-loop sense. On the MACE program, though, closed-loop experimentation could not wait for the final updated model to be completed. Instead, control design and implementation was initiated using very early versions of the finite element model. These versions of the finite element model were essentially the finite element input-output models presented at the end of the previous chapter and hence were significantly in error.

Controllers were designed using a variety of robust control techniques and topologies [36, 37]. The different robust control techniques used in the closed-loop experimentation were the Linear Quadratic Gaussian (LQG), Sensitivity Weighted LQG, Maximum Entropy (ME), and the Multiple Model techniques. These techniques (except LQG) are capable of robustifying the control to uncertainties in the design model by including weightings on the uncertain modes. Controllers were designed for the XY- and Z-axis subproblems, as well as the full XYZ-axis problem. These control designs pointed out problem areas in the model (Step C in Figure 2.1).

Most of the problem areas pointed out by the control designs were errors in the frequencies and damping ratios of some of the modes. Typically, the controllers would be designed and implemented on experimental data. One mode would cause instability and thus require more weighting in the control design. Once this instability was prevented, the controllers would be redesigned with higher control authority until another mode (or modes) became unstable. Weightings would be included on those modes as well. In this manner, a list of the more troublesome modes could be compiled

as a function of control authority. For the XY-axis controllers, the order of the modes that caused the problems was one of the 1st violin modes, the 3rd Y-axis bending mode, the 1st Y-axis bending mode, one of the 2nd violin modes, the 4th Y-axis bending mode, the 6th Y-axis bending mode, and the 2nd Y-axis bending mode. For the Z-axis controllers, the order was the 1st Z-axis bending mode, another of the 1st violin modes, another of the 2nd violin modes, the 4th Z-axis bending mode, and several modes above 50 Hz. The modes that the finite element based controllers indicated were going unstable were already known to be in error. These errors were obvious when the predicted transfer functions were compared with the experimental data. Since these modes were already known to be in error, they were already being updated in the open-loop update.

Another aspect that the finite element based control designs pointed out was that the achievable performance of the full XYZ-axis problem was limited mainly by the model predictions of the so-called cross channels. The cross channels are lower magnitude transfer functions that come about because the structure's dynamics do not completely couple into XY- and Z-axis dynamics. The 'cross channel' term describes the crossing of the output along one axis resulting from an input along the other axis. The problem with the finite element model was that the model predicted a different coupling than was seen in the data. These errors were known before the control designs were attempted, but the extent of their effect on the closed-loop results was not known.

Using these robust control techniques, performance improvements approaching 20 dB were achieved using controllers designed using a finite element model that was significantly in error. The performance achieved using the finite element model is close to that achieved using a measurement model (Step D in Figure 2.1). Since controllers designed using the finite element and measurement model achieve similar performances when implemented on the test article, this indicates that the update process can stop and the 1-g model is in its final form.

4.7 Final 1-g Model

The modal frequencies of the final 1-g model, and their errors when compared with a measurement model, are presented in Table 4.7. This table shows that the largest frequency errors occur in modes with damping ratios greater than 10%. Since these modes are so heavily damped, they tend not to cause serious problems during control design. The rest of the flexible modes exhibit frequency errors less than 5%, with most of the structural modes (*i.e.*, non suspension modes) having errors less than 2%. It is important to note that although the suspension modes will not be present in the 0-g model, it is still necessary to capture these modes accurately in order to capture the correct suspension coupling with the structural modes.

The damping ratios are also very accurate, with most of the larger errors also occurring in the heavily damped modes. Since the only source of this much damping is the gimbal servo loops, these errors bring up the possibility of errors in the modeling of these servo loops. Comparison of the modal frequencies is only one method used in evaluating the quality of the model. Another is an examination of the input-output predictions of the model, the most important aspect for control design.

Figure 4.15 shows the comparison between model and data for the Z-axis secondary gimbal command (*i.e.*, the disturbance source) to the inertial angular rate of the Z-axis rate gyro on the primary gimbal (*i.e.*, the performance sensor). The solid line indicates the 1-g model and the dashed line indicates the data. The figure shows that the overall shape of the analytical magnitude transfer function is excellent. The most noticeable error is in the 6-20 Hz range where there appears to be a problem with the general shape of the transfer function. From experience, this shape is due mostly to the gimbal servo loops, giving further indication that there might be a problem with the servo loop closure on the model. Another region with large error is the 80-200 Hz range. This region shows poor correlation with the data because it was not included in the automated update. The only other noticeable error is in the damping ratio of the 45 Hz mode where the model appears to have a damping ratio larger than that indicated by the data. Note that the phase also shows excel-

Table 4.7: Comparison of frequencies and damping ratios for ground experimental data and final 1-g input-output model

Description	Measurement Model		Finite Element Model		Errors	
	Frequency (Hz)	Damping ratio	Frequency (Hz)	Damping ratio	Frequency (%)	Damping (%)
Y twist	0.24	0.100	0.23	0.100	-4.2	0.0
Z tilt	0.44	0.038	0.45	0.038	2.3	0.0
X rotation	1.39	0.036	1.39	0.035	0.0	-2.8
1st Z bending	1.96	0.022	1.96	0.022	0.0	0.0
1st Y bending	3.38	0.038	3.31	0.036	-2.1	-5.3
gimbal pend.	4.57	0.220	4.39	0.236	-3.9	7.3
gimbal pend.	4.79	0.179	5.27	0.431	10.0	140.8
gimbal pend.	5.93	0.414	6.09	0.409	2.7	-1.2
gimbal pend.	6.75	0.271	6.49	0.255	-3.9	-5.9
1st Violin	6.76	0.033	7.11	0.033	5.2	0.0
1st Violin	7.14	0.024	7.12	0.024	-0.3	0.0
1st Violin	7.19	0.022	7.16	0.021	-0.4	-4.5
1st Violin	7.21	0.024	7.18	0.024	-0.4	0.0
2nd Y bending	9.29	0.011	9.12	0.011	-1.8	0.0
2nd Z bending	11.34	0.105	9.85	0.445	-13.1	323.8
3rd Z bending	12.11	0.103	13.00	0.159	7.3	54.4
2nd Violin	12.71	0.010	12.93	0.010	1.7	0.0
2nd Violin	14.29	0.017	14.34	0.015	0.3	-11.8
2nd Violin	14.47	0.010	14.34	0.009	-0.9	-10.0
3rd Y bending	16.88	0.014	16.89	0.009	0.1	-35.7
3rd Violin	21.54	0.017	21.58	0.017	0.2	0.0
3rd Violin	21.71	0.026	21.58	0.026	-0.6	0.0
4th Y bending	22.63	0.070	21.84	0.032	-3.5	-54.3
time delay	29.88	0.458	32.92	0.566	10.2	23.6
time delay	32.24	0.538	34.17	0.506	6.0	-5.9
4th Z bending	39.05	0.021	38.50	0.021	-1.4	0.0
5th Z bending	45.41	0.015	45.19	0.016	-0.5	6.7
5th Y bending	48.55	0.016	47.83	0.016	-1.5	0.0
6th Y bending	53.98	0.022	53.77	0.022	-0.4	0.0

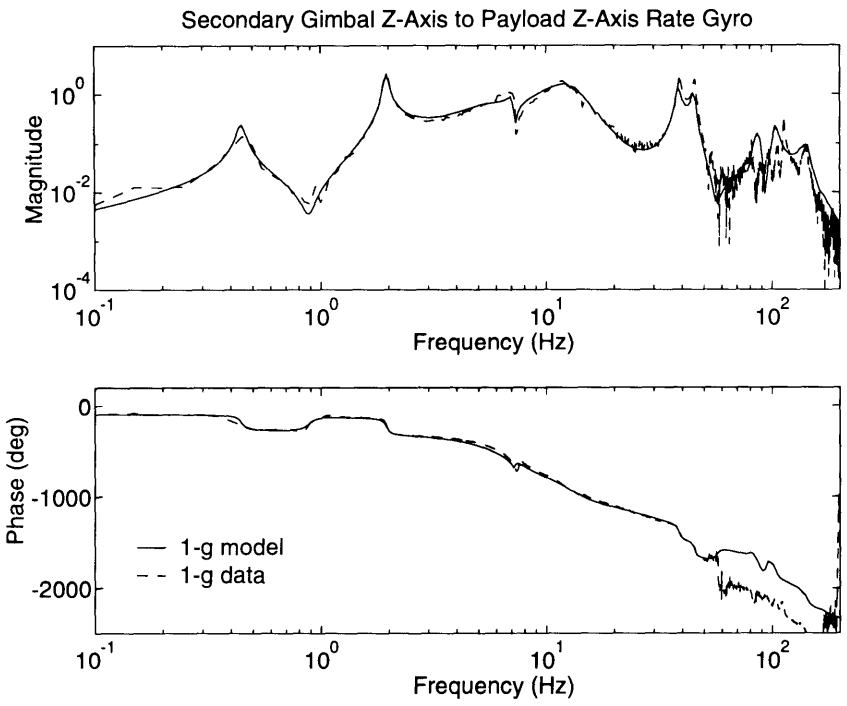


Figure 4.15: Typical in-plane transfer function for final 1-g input-output model

lent agreement with the data. An important point to note is that the finite element model also captures the zeros extremely well. This is important because improperly modeled zeros can cause serious problems in high authority control.

Figure 4.16 shows the comparison between model and data for the X-axis secondary gimbal command to the inertial angular rate of the X-axis rate gyro on the primary gimbal. Again, excellent agreement is seen between modeled and measured transfer functions. One of the most noticeable errors is in the 4-30 Hz range where there again appears to be a problem with the shape of the transfer function, again possibly attributable to an error in the servo loops. Another large error is seen in the modes at 17 Hz and 22 Hz where there appear to be problems with the modal residues. The explanation for these errors is found in the switch to NASTRAN. In order to get the 1-g NASTRAN model to converge, all the gimbals had to be locked during the nonlinear static part of the solution. This caused those modeshapes to not have as much gimbal motion in them as would ordinarily have been present.

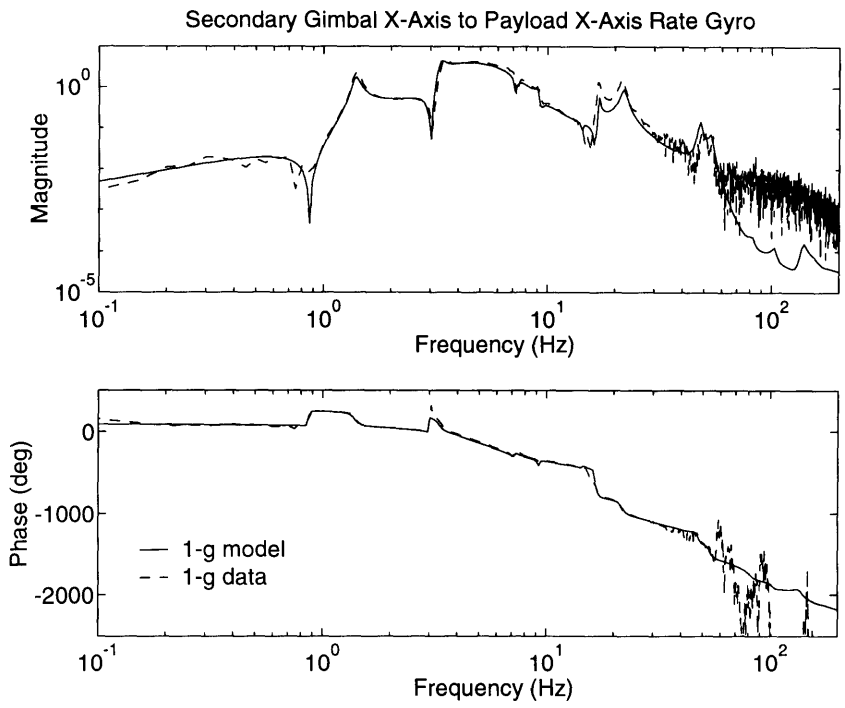


Figure 4.16: Typical out-of-plane transfer function for final 1-g input-output model

Currently, there is no solution to this problem, but this problem should not occur in the 0-g model because the nonlinear static solution is not needed. Again excellent agreement is seen in the phase part of the transfer function. Also note that the zeros are again captured extremely well. (A more complete set of overlays of the final 1-g model with the experimental data is given in Appendix D.)

To illustrate the improvements that the update process has made in the model, Table 4.8 shows a comparison of the frequency errors when compared to a measurement model for the initial model, the model after engineering insight, and the final model after automated updating. For most modes, this table shows a steady improvement in the frequencies. For some modes, engineering insight worsened the error, but automated updating improved it. For other modes, most notably the 1st Y-axis bending mode, the error got progressively worse during the course of the update process, but is still small.

With the completion of the update of the 1-g finite element input model, the next

step in the modeling approach of the MACE program is to remove the suspension system and gravity effects to arrive at a prediction of the 0-g behavior of the MACE test article.

Table 4.8: Comparison of frequency errors for initial, engineering insight updated, and automated updated models

Description	Measurement Model (Hz)	Initial Model (%)	Engineering Insight (%)	Automated Updating (%)
Y twist	0.24	-4.2	-4.2	-4.2
Z tilt	0.44	2.3	2.3	2.3
X rotation	1.39	0.7	0.7	0.0
1st Z bending	1.96	2.0	1.0	0.0
1st Y bending	3.38	0.0	-0.9	-2.1
gimbal pend.	4.57	-6.8	-3.3	-3.9
gimbal pend.	4.79	12.1	17.1	10.0
gimbal pend.	5.93	4.0	2.2	2.7
gimbal pend.	6.75	-3.7	-2.8	-3.9
1st Violin	6.76	1.2	5.2	5.2
1st Violin	7.14	-4.2	-0.4	-0.3
1st Violin	7.19	-4.7	-0.6	-0.4
1st Violin	7.21	-4.7	-0.6	-0.4
2nd Y bending	9.29	3.0	-0.6	-1.8
2nd Z bending	11.34	1.0	-1.8	-13.1
3rd Z bending	12.11	5.5	0.6	7.3
2nd Violin	12.71	3.9	1.8	1.7
2nd Violin	14.29	-3.6	0.1	0.3
2nd Violin	14.47	-4.7	-1.0	-0.9
3rd Y bending	17.06	11.1	1.0	0.1
3rd Violin	21.54	-3.8	0.0	0.2
3rd Violin	21.71	-4.5	-0.7	-0.6
4th Y bending	22.63	21.8	-2.8	-3.5
time delay	29.88	3.2	6.1	10.2
time delay	32.24	1.0	3.1	6.0
4th Z bending	39.05	3.6	0.5	-1.4
5th Z bending	45.41	0.7	0.4	-0.5
5th Y bending	48.55	6.6	0.2	-1.5
6th Y bending	53.98	3.5	0.2	-0.4

Chapter 5

0-g MACE Model Development

Now that the 1-g finite element input-output model has been updated, the last and most important step is the formation of a model of the 0-g behavior. This step also represents a leap of faith in the modeling process. It is the faith in the finite element method to accurately capture the suspension and gravity effects that brings us to this step.

The modeling process up to this point has been to update, with ground experimental data, a finite element model that captures the suspension and gravity effects on the test article. The suspension and gravity effects can then be analytically removed from the finite element model to arrive at a prediction of the 0-g behavior. The removal of the suspension and gravity effects from the finite element model and the derivation of a nominal 0-g input-output model will be discussed in this chapter.

Since the 0-g model derived in this chapter is to be used for control design, some measure of how the model will be in error, when compared with flight experimental data, is needed. Since no experimental data exists before flight, this uncertainty model must be derived based on ground testing and the modeling process. Since the uncertainty model must use ground testing and the analytical power of the finite element method, this uncertainty model must take into account the residual errors in the 1-g model and any errors likely to occur during the transition from the 1-g model to the 0-g model. One such method is to modify the 0-g results based on the 1-g errors and place bounds on remaining errors about this modified model. The

derivation of this uncertainty model will also be discussed in this chapter.

5.1 Removal of suspension and gravity effects

Once the final updated 1-g finite element model has been formed, the 0-g input-output model of the test article can be derived (Figure 2.1) by analytically removing the suspension and gravity effects from the 1-g finite element model. The result of their removal is a prediction of the 0-g natural frequencies and modeshapes of the test article. These frequencies and modeshapes are combined with damping ratios and post-processed to form a 0-g input-output model by adding models of the actuators and sensors, hardware gains, electronic filters and time delays, and servo loops, similar to the 1-g model.

To remove the the suspension system and gravity effects from the finite element model, the 1-g model containing the final updated physical parameters is used as a starting point. The suspension system is removed from this model by eliminating the nodal points, masses, and elements that make up the suspension devices and cables. In addition, any mass due to connections between the suspension system and the test article that may have been placed at nodal points of the base structure must be removed from these locations. For the MACE test article, masses for the connectors were lumped at the end and center nodes, and consequently must be removed.

With the suspension system removed, the resulting finite element model is now a model of the free-free structure that has been refined using ground experimental data. Since the test article in orbit is essentially free-free, there are no initial stresses in the structure due to gravity loading. Consequently, the nonlinear static procedure used in the 1-g finite element model is no longer needed. With the elimination of this procedure, the gravity effects have been removed from the finite element model. Thus, the 0-g natural frequencies and modeshapes are obtained simply by performing an eigensolution on the linear mass and stiffness matrices.

When the 0-g finite element solution is performed, the resulting natural modes will fall into two categories: rigid body modes and flexible modes. Rigid body modes are

characterized by zero natural frequencies and modeshapes that are rigid translations and rotations of the multiple bodies that make up the test article. For the MACE test article, there are 13 rigid body modes: 3 translational modes of the bus; 3 rotational modes of the bus; 1 rotational mode for each gimbal axis (4 total); and 1 rotational mode for each reaction wheel (3 total). The flexible modes are modes that have finite, nonzero frequency and modeshapes which exhibit flexible deformation of the test article.

The 0-g behavior of the test article is much simpler than the 1-g behavior, which is seen in the number of modes necessary to capture the dynamics in the same frequency band. In the 1-g model, nearly 70 modes are necessary to capture the test article behavior in the 0-60 Hz range, while in the 0-g model only 27 modes (which includes the 13 rigid body modes) are necessary to cover the same frequency range. A comparison of the unservoed 0-g flexible modes in this frequency range with the appropriate unservoed 1-g modes, along with the relative frequency change from 1-g to 0-g is presented in Table 5.1. The 1-g and 0-g modes that show similar bus deformations in their modeshapes are paired. This table shows that, except for the 1st Y-axis bending, the 2nd Y-axis bending, and the 4th Z-axis bending modes, all the modes increase in frequency in the transition from 1-g to 0-g. At first glance, this may seem unexpected as the suspension system is supposed to stiffen the structural frequencies. Although this is true, the suspension system also introduces significant mass into the system as well. It is the combination of the extra mass and stiffness that determines whether the mode will increase or decrease in frequency during this transition. For the majority of the structural modes, the mass of the suspension system couples more strongly with the test article than the stiffness of the suspension system, causing the modes to increase in frequency. For the 1st and 2nd Y-axis bending modes, on the other hand, the stiffness of the suspension system couples more strongly than the mass of the suspension system causing the modes to decrease in frequency as expected. The 4th Z-axis bending mode illustrates the case when the coupling of the mass and the stiffness of the suspension system are about equal, causing the mode to change very little in frequency.

Table 5.1: Natural frequencies for 0-g and 1-g finite element models

Description	1-g (Hz)	0-g (Hz)	change (%)
1st Z bending	2.06	2.10	1.9
1st Y bending	3.60	3.07	-14.7
2nd Y bending	8.77	8.64	-1.5
2nd Z bending	8.59	8.67	0.9
torsion, secondary end	10.23	10.33	1.0
3rd Z bending	10.64	10.68	0.4
torsion, primary end	11.87	12.42	4.6
4th Z bending	14.03	14.03	0.0
3rd Y bending	16.89	16.93	0.2
4th Y bending	22.00	22.11	0.5
5th Z bending	38.05	38.18	0.3
6th Z bending	44.69	44.95	0.6
5th Y bending	47.14	48.16	2.2
6th Y bending	53.03	54.27	2.3

In order to complete the 0-g open-loop modal model, damping ratios for the 0-g flexible modes need to be determined. Damping is very difficult to model analytically, and as such, the damping ratios used in the 1-g finite element model have been determined from experimental data. Since experimental data does not exist to determine the 0-g damping ratios, these damping ratios must be determined by some other means. One option is to use the 1-g damping ratios and possibly modify them to account for changes that might occur in the transition from 1-g to 0-g. The difficulty here is in determining how to modify the 1-g damping ratios for 0-g, as there is very little experience in doing this for a MACE type of structure.

In the MODE experiment [38], damping was higher in 0-g than in 1-g. It is believed that this was caused by the joints in the truss structure, which have a deadband. In 0-g, the static shape of the MODE truss places most of these joints in the middle of these bands. Dynamic motion causes one joint surface to collide with the opposing surface resulting in impact damping. In 1-g, the gravity sag causes these surfaces to remain in contact thereby eliminating this damping source.

On the other hand, the MACE suspension system in 1-g provides some electronic damping in the bounce direction (Y direction). It also provides structural transmis-

sion paths which allow energy to escape into the suspension mounting structure. One might surmise that the lack of the suspension system during 0-g tests might cause a decrease in damping. Because of these conflicting effects, it was decided to simply use the unmodified 1-g damping ratios in the 0-g model. This is because, although the directions of the changes are known in the two effects, the magnitudes of these effects are not well known. Thus, it is better to use known quantities for such an important parameter as damping, than to make blind modifications.

Once the 0-g natural frequencies, modeshapes, and damping ratios have been selected, the next step is to form the 0-g input-output model, following the same process as required for the 1-g model. Since the actuators and sensors are identical on the ground and in orbit, their behavior and hardware gains are included in the 0-g model in the same manner as for the 1-g model. It is also assumed that the signal conditioning filters used on the ground are identical to those used in orbit. If this is not the case, the differences between the ground and flight filters need to be known to very high precision so that the effects of these filters can be removed from the update process. Thus, the next step in forming the input-output model is the inclusion of these filters into the model.

The last step in the formation of the input-output model is to include the servos into the model. This is probably the most delicate part of the post-processing step, as some of the parameters that affect the servo have been updated based on ground experimental data. Thus, if the servos used on the ground are not the same as the servos used on orbit, or the hardware used to implement the servos is not the same from ground to flight, the parameters used in the update should be measured and not included in the update. If the flight and ground servos are the same and the flight and ground control computers are the same, the 1-g parameters updated based on ground data can be used for the 0-g model, as there should not be any gravity effects on the computational time delays. With the closure of the servo loops, the formation of the nominal 0-g input-output model is complete.

5.2 Nominal Predictions

The resulting closed-loop modal frequencies of the 1-g and 0-g models, frequency changes from 1-g to 0-g, and frequency errors between the 1-g model and the modal parameters from the 1-g measurement model are given in Table 5.2. From the table, it is also seen that most of the changes in modal frequency are rather small ($\leq 1\%$). The exceptions are the 1st Y-axis bending mode, the gimbal pendular modes, the 2nd and 3rd Z-axis bending modes, and the 5th and 6th Y-axis bending modes. As indicated in the table, most of the modes that have undergone large changes in frequency are also the very heavily damped modes that were the most in error in the final 1-g model, the lightly damped exceptions being the 1st Y-axis bending and 5th and 6th Y-axis bending modes.

The large changes in the heavily damped modes are caused by pendular stiffening which is absent in 0-g. The gimbal pendular modes are expected to undergo large changes because they are changed to servoed rigid body articulation modes in 0-g. The fact that they show up as close in 1-g and in 0-g as they do is attributable to the gimbal servo controllers, which dominate over gravity effects. The 2nd and 3rd Z-axis bending modes exhibit large gimbal motions in their modeshapes and, as such, are heavily influenced by the gimbal servos. Since the pendular stiffening effects of gravity are not present on orbit, the gimbal motions in these modes would be expected to be larger, which, in turn, causes the servos to have a larger effect on these modes causing them to undergo large frequency changes from 1-g to 0-g.

The large changes seen in the lightly damped modes, on the other hand, are caused by the coupling of the structural modes with the suspension system. In examining the 1-g modeshape for the 1st Y-axis bending mode, it is seen that this mode couples very strongly with the suspension system. The suspension cables restrict the range of the structural motion, causing a stiffening of this mode. On orbit, the suspension cables are not present to limit the range of motion of this mode, and the stiffening effect is absent causing the frequency of this mode to drop dramatically. The 5th and 6th Y-axis bending modes, on the other hand, do not couple very strongly with the

Table 5.2: Modal frequencies for 1-g and 0-g input-output models

Description	1-g (Hz)	0-g (Hz)	Frequency changes (%)	Frequency errors (%)
Y twist	0.23			
Z tilt	0.45			
X rotation	1.39			
1st Z bending	1.96	1.94	-1.0	0.0
1st Y bending	3.31	2.96	-10.6	-2.1
gimbal pend.*	4.39	4.00	-8.9	-3.9
gimbal pend.*	5.27	5.09	-3.4	10.0
gimbal pend.*	6.09	5.90	-3.1	2.7
gimbal pend.*	6.49	6.24	-3.9	-3.9
1st Violin	7.11			
1st Violin	7.12			
1st Violin	7.16			
1st Violin	7.18			
2nd Y bending	9.12	9.14	0.2	-1.8
2nd Z bending*	9.85	10.49	6.5	-13.1
3rd Z bending*	13.00	12.69	-2.4	7.3
2nd Violin	12.93			
2nd Violin	14.34			
2nd Violin	14.34			
3rd Y bending	16.89	16.91	0.1	0.1
3rd Violin	21.58			
3rd Violin	21.58			
4th Y bending	21.84	21.78	-0.3	-3.5
4th Z bending	38.50	38.65	0.4	-1.4
5th Z bending	45.19	45.56	0.8	-0.5
5th Y bending	47.83	49.91	4.3	-1.5
6th Y bending	53.77	56.02	4.2	-0.4
* indicates $\zeta \geq 10\%$				

stiffness of the suspension cables. The only coupling that these modes have with the suspension system is through the steel connecting rods at the ends of the test article, which undergo large rotation in these modes. However, these connecting rods are not present on the flight version of the test article, so the masses and rotary inertias decrease, causing the frequencies of the modes to increase. In one mode, one end of the test article is undergoing the large rotation and in the other mode, the opposite end of the test article undergoes large rotation. Thus, since the steel connecting rods at either end of the test article have nearly identical masses and inertias, it would be

expected that the increase in frequency of these modes would also be approximately the same, which is confirmed by both Table 5.1 and Table 5.2.

As was found with the 1-g model, a comparison of modal frequencies is not enough to determine model accuracy. The most important aspect of a model used for control design is the predicted input-output behavior of the model. The same disturbance to performance transfer functions that have been used to illustrate the 1-g input-output model comparisons with experimental data will be used here to illustrate the changes in the input-output behavior that occur from 1-g to 0-g. To examine the changes in the in-plane input-output behavior of the structure, Figure 5.1 shows the comparison between the 1-g and 0-g models for the Z-axis secondary gimbal relative angle command to the inertial angular rate of the Z-axis rate gyro on the primary gimbal. In this transfer function, it is seen that above 8 Hz, the general shape of the transfer function does not change significantly from 1-g to 0-g. Below this frequency, though, the 0-g model exhibits some significant differences. The troublesome 7 Hz suspension mode has disappeared and the 1st Z-axis bending mode has become more pronounced in the 0-g model. The reason that the 1st Z-axis bending mode has become more pronounced is similar to the reason for the large frequency shift in the 1st Y-axis bending mode. The suspension cables limit the range of motion of the test article in 1-g. But since the motion of the test article is in the plane of the suspension system, the amount of coupling with the stiffness of the suspension system is about equal to the coupling with the mass of the suspension system, the frequency of this mode undergoes only small changes. Also note in the transfer function that all the dynamics below the 1st Z-axis bending mode have disappeared in the 0-g model, because all of these dynamics in the 1-g model are caused by the suspension system which is no longer present.

Figure 5.2 can be examined for the differences in the out-of-plane behavior between 1-g and 0-g for the X-axis secondary gimbal relative angle command to the inertial angular rate of the X-axis rate gyro on the primary gimbal. In this transfer function, the same general trends are seen. The general shape of the transfer function does not change significantly above 5 Hz. In this region, though, it is noted that the 0-g

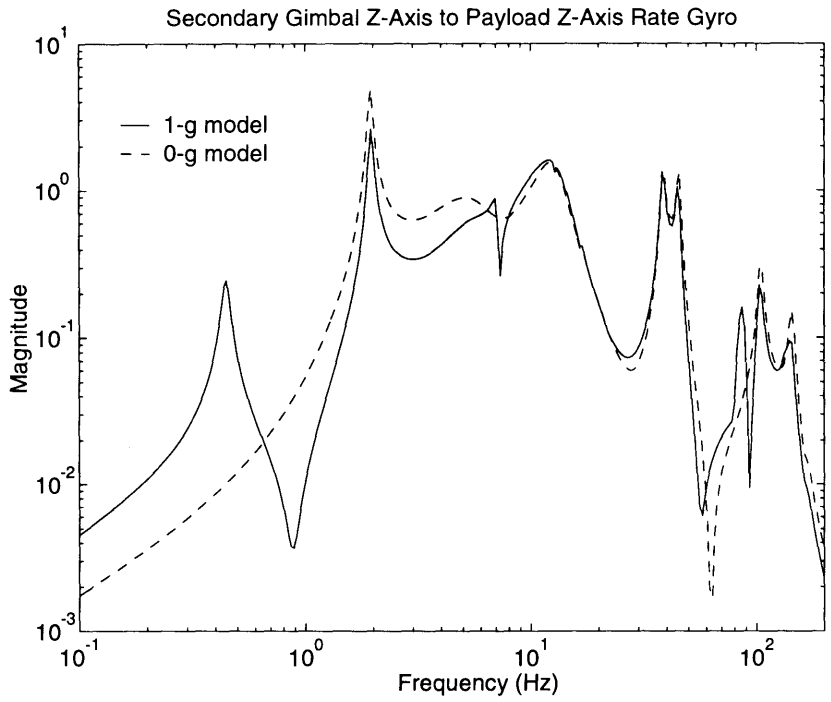


Figure 5.1: Typical in-plane transfer function for nominal 0-g model

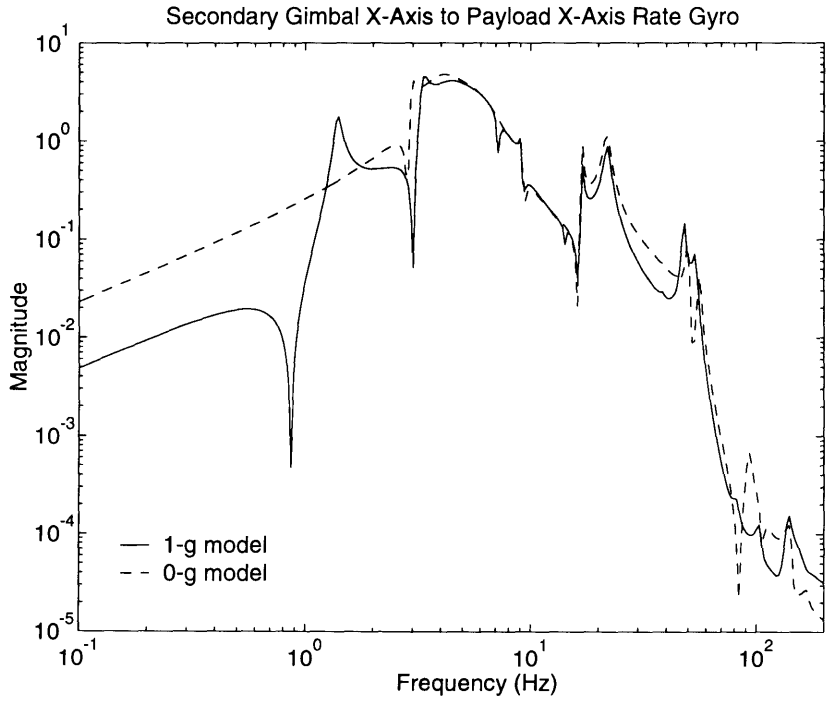


Figure 5.2: Typical out-of-plane transfer function for nominal 0-g model

transfer function is missing the zeros at 7 and 11 Hz that appear in the 1-g transfer function, which is caused by the absence of the violin modes in the 0-g model. In this region, also note that the 17 and 22 Hz modes have become more pronounced in the 0-g model. This confirms the suspicion that their diminished behavior in the 1-g model is caused by the changes made to get the nonlinear static procedure to converge (Chapter 4). The significant drop in the frequency of the 1st Y-axis bending mode is also clearly seen in this transfer function. Below the 1st Y-axis bending mode and its preceding zero, no dynamics are seen in the 0-g model, again due to the absence of the suspension system.

5.3 Uncertainties

Generating a nominal model is only half the problem in deriving a model useful for robust control design. The other half of the problem is determining a model of the uncertainty that can be used in robust control designs. The uncertainty model is important because modern control design techniques, depending on the weighting parameters used in the design, will derive very high authority controllers that try to invert the model and substitute some desired set of dynamics. This is satisfactory when there are no differences between the model and the actual structure. This becomes unsatisfactory when there are errors in the model, because these model errors will cause instability when the controllers are implemented on the actual structure.

At some level, model errors exist in the models of all real systems. In the case of the finite element model, these errors may be easily seen. In the case of a measurement model, the errors may be very difficult to discern. At very high authority, though, even these minute errors will cause instability to result when the controller is implemented on the structure. When model errors are known, robust control techniques such as Sensitivity Weighted LQG, Maximum Entropy, Multiple Model, *etc.* [37], can take these errors into account uncertainty and derive controllers that are stable when implemented on the structure.

Since the model is known to be in error, some measure of this error is necessary.

This is where the uncertainty model comes into play, because it represents how the model is in error when compared to experimental data from the test article in its operating environment. This uncertainty model includes both errors, which have directions, and bounds, which define symmetric regions of uncertainty around the nominal values. The most useful form of the uncertainty model for robust control design is to provide regions of uncertainty for the relevant parameters that make up the model, namely the modal frequencies and damping ratios. When experimental data exists, these regions can simply be found by comparing the model predictions and the experimental data. The difficult part for the MACE program comes about because it is desired to derive controllers based on the 0-g finite element model before any experimental data exists. Thus, the 0-g uncertainty model must be derived without the benefit of on-orbit experimental data. Since ground experimental data and a method of predicting the 0-g behavior of the test article both exist, the best method of determining the 0-g uncertainty model is to extrapolate the residual errors in the 1-g finite element model into bounds on the 0-g modal frequencies, since errors in frequency cause the most severe problems [37].

Two distinct cases need to be considered in determining the 0-g uncertainty model, namely the heavily damped modes and the lightly damped modes, where heavily damped is defined as $\zeta \geq 10\%$. The residual errors in the heavily damped modes of the 1-g model are much larger than the residual errors in the lightly damped modes. This would seem to suggest that the bounds on the heavily damped modes should be much larger than those for the lightly damped modes, but heavily damped modes have a built-in robustness due to the fact that large amounts of damping tend to spread the mode out in frequency and are less likely to lead to instability. It should also be pointed out that all the modes that caused problems in the closed-loop control results presented in the previous chapter were lightly damped modes. Thus, since heavily damped modes do not tend to cause severe problems, the frequency bounds on the heavily damped modes are not really needed. If, during the course of control design, it is discovered that these modes are causing problems, the bounds can be set at some arbitrary value, $\pm$ 2.5% for example.

In the case of the lightly damped modes, one uncertainty model can include both errors and bounds, providing a nonsymmetric uncertainty model. Another uncertainty model can be used for a revised nominal model in which the errors have been removed by adjusting the 0-g open-loop frequencies of the nominal model such that they differ from the unaltered nominal closed-loop model by the residual 1-g errors. The bounds are then are determined about this updated nominal model. This results in symmetric bounds about the nominal value. Changing the open-loop frequencies of the 1-g model was strictly forbidden, because if they were changed, no method would have existed to project the changed frequencies into 0-g. It was for this reason that updating of the 1-g input-output model was performed by updating physical parameters of the finite element model, which could be projected into 0-g. The damping of this new nominal 0-g model can also be adjusted to correct the residual damping errors present in the 1-g model. Since damping is an external parameter to the finite element model, adjusting the damping is a reasonable thing to do in order to correct for the deficiencies of the 1-g model.

With the nominal model having been redefined, the problem still remains to determine the frequency bounds on the lightly damped modes. In determining these bounds, both the residual 1-g error and possible modeling errors in the transition from 1-g to 0-g need to be considered. The residual 1-g error comes mainly from remaining parameter errors. However, it also includes nonlinearities and measurement noise, which may place additional bounds on the uncertainty. The possible 1-g to 0-g modeling errors result from the update process where there is uncertainty in what effects were inadvertently attributed to the suspension and *vice versa*. These sorts of errors cause the nominal 0-g predictions to be uncertain.

Regardless of the rationale used, the bounds that are chosen will always be somewhat arbitrary. In light of this, the bounds that have been chosen for the lightly damped modes are the larger of 1% and one-half of the residual absolute 1-g errors. The 1% bound was chosen as a hedge against modeling errors in the finite element method. For example, certain parameters in the finite element model may have been changed to incorrect values to account for errors in the 1-g model, and to try to ac-

Table 5.3: Updated nominal 0-g model and bounds

Description	Nominal Frequency (Hz)	Nominal Damping Ratio	Frequency Bounds (%)
1st Z bending	1.94	0.022	± 1.0
1st Y bending	3.02	0.024	± 1.1
gimbal pend.	4.02	0.224	± 2.5
gimbal pend.	5.09	0.422	± 2.5
gimbal pend.	5.90	0.417	± 2.5
gimbal pend.	6.24	0.264	± 2.5
2nd Y bending	9.30	0.013	± 1.0
2nd Z bending	10.51	0.503	± 2.5
3rd Z bending	12.69	0.149	± 2.5
3rd Y bending	16.89	0.013	± 1.0
4th Y bending	22.54	0.055	± 1.8
4th Z bending	39.19	0.022	± 1.0
5th Z bending	45.79	0.016	± 1.0
5th Y bending	50.65	0.014	± 1.0
6th Y bending	56.24	0.017	± 1.0

count for some possible nonlinearities that might be different in 1-g and 0-g. One-half of the residual 1-g error was used to account for some of the larger errors in the 1-g model. Since these bounds are rather arbitrary, they may be revised based on the predicted control results. The modal frequencies and damping ratios of the updated nominal model, along with the frequency bounds, are summarized in Table 5.3.

One final aspect that must be considered in how the 0-g model will be in error is the fact that the structure will be assembled and disassembled several times while in orbit. Similar assembly/disassembly tests performed on the ground have shown shifts in the frequencies of some of the modes. Thus, the controllers designed based on the finite element model must not only be robust to the expected modeling errors, but also to these shifts in frequency based on assembly and disassembly. These shifts are usually less than the bounds given in Table 5.3, and as such, no additional uncertainty will be added to the bounds given in the table. If during the course of ground testing, shifts larger than the bounds in the table are seen, the bounds should be adjusted to account for these shifts.

The 0-g model that is used for control design consists of the updated nominal 0-g

input-output model along with the uncertainty bounds. With the definition of this 0-g model, the modeling process presented in Chapter 2 is now complete. This 0-g model is handed over to the control designer, who will design controllers, based on this model, that will be implemented on orbit.

Chapter 6

Conclusions

A modeling approach to obtain models suitable for the design of high authority control of a structure in 0-g has been proposed. This approach combines analytical techniques, in the form of the finite element method, with ground experiments. Since the eventual use of the derived model is control design, the ground experiments also include open- and closed-loop ground testing.

As part of this modeling approach, an input-output model of the behavior of the suspended MACE test article has been created. This model uses as a basis a finite element model which combines a model of the free-free structure with models of the suspension and gravity effects that affect the structure during ground testing. Since the model must match the behavior of the structure as seen by the control computer, the model has been augmented with mechanical and electronic models of the actuators and sensors, as well as noise attenuating filters on the output channels. Since MACE has initial pointing servo loops closed on the gimbals, which are implemented using a digital computer, the associated time delays of the computer and servo loops have been included in the 1-g model as well.

This initial 1-g input-output model was significantly in error when it was compared with experimental data, and was consequently improved through a process called updating. Coarse improvements were made using the engineer's modeling experience, called engineering insight, while finer adjustments were made using automated updating techniques. During the course of the automated updating, a method for judging

the quality of the selected update parameters was presented. In this method, the sensitivity of the cost to the update parameters was used to show the orthogonality between the sensitivities and the magnitude of the influence of the sensitivities on the initial cost. The check on orthogonality between the sensitivities shows if any of the sensitivities have similar directions, which would lead to non-unique solutions in the automated update, possibly invalidating the update. The influence of the sensitivities on the cost is checked in order to tell which parameters have a large influence on the cost. This will eliminate parameters that have little effect on the cost and reduce the size of the update. It was found that the best method for automated updating was to use a frequency domain technique in which the finite element model was compared to the raw complex transfer function data in a logarithmic cost. The best method for incorporating the update parameters into the update process was to use a cost sensitivity method in which the sensitivity of the vectorized cost to the update parameters is used to compute the new cost. Automated updating using a logarithmic cost and the cost sensitivity method resulted in a 28% improvement in the MACE model. As a third step in the update process, the 1-g model was closed-loop verified by designing controllers based on the 1-g finite element input-output model and implementing them on the suspended test article.

Once the update process was complete, the final 1-g model was used to generate a model of the 0-g behavior of the test article by analytically removing the suspension and gravity effects. Because robust control theory requires a measure of the uncertainty present in the model, an uncertainty estimate, which places bounds on the modal parameters of the nominal model, has been derived. Since no flight experimental data exists, this uncertainty model must be derived from the residual errors in the 1-g finite element model and the possible modeling errors in the transition from 1-g to 0-g. This uncertainty model includes errors, which have directions, and bounds, which define symmetric regions of uncertainty around the nominal values. The errors in the 0-g model are reduced by updating the 0-g model derived from the updated 1-g model so that the resulting closed-loop frequencies are shifted by the residual 1-g frequency errors. Bounds may be determined in the 0-g model by con-

sidering the effects of parameter errors, nonlinearities, measurement noise, and errors due to parameter misalignment, which comes about when suspension parameters are inadvertently updated to account for structural effects and *vice versa*.

The modeling approach presented in this thesis will be repeated for the Flight Model of the MACE hardware. Due to the nature of the remaining errors in the EM MACE model, all of which seem to be concentrated in the modes with large damping ratio, the likely culprit for these errors seems to be mismodeling of the gimbal servos. A detailed examination of the implementation of the servos in the FM will hopefully reduce these remaining errors. Other general MACE modeling concerns are the boundary conditions of the suspension cables, and the observability problems of the 17 and 22 Hz modes caused by the convergence of the nonlinear static routine. Finally, the effects of nonlinearity and assembly/repeatability on the observed behavior of the structure, and their impact on the model updating, need to be investigated.

Since 1-g closed-loop results for the full control problem are limited by the prediction of the cross-channels, in the future, these transfer functions will be included in the update algorithm for the finite element model. Future work on parameter selection for automated updating is also recommended.

Much work remains to be done in the area of closed-loop updating. The closed-loop updating presented in this thesis was of an *ad hoc* nature. Future work on closed-loop updating should include means of using the results of high authority control in the update process in a more automated fashion. Also, it is unclear that the proper model to use in control design is the model of the exact system, perhaps a different model would give better results. Thus, future work in the issues associated with modeling for control is recommended.

The usefulness of the proposed modeling approach is driven by how well the 0-g behavior of the structure is predicted. Once MACE flies, the predictions of the modeling approach for the FM MACE hardware can be evaluated. Comparison of the 0-g model with the flight experimental data will either validate the modeling approach or show areas of the approach that need to be revised, the most likely of which is the

modeling of the suspension and gravity effects. The eventual goal of the modeling approach is to be able to derive flight controllers that meet the specifications based on the finite element model and that do not need to be redesigned once in orbit. This would eliminate the need to do expensive and time consuming on-orbit identification and controller redesign.

References

[1] Zienkiewicz, O. and R. Taylor, *The Finite Element Method*, McGraw-Hill Book Company, London, 4th ed., 1992.

[2] Yang, T., *Finite Element Structural Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1986.

[3] Bathe, K.-J., *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.

[4] Ljung, L., *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[5] Miller, D., J. de Luis, and E. Crawley, "Dynamics and Control of Multipayload Platforms; the Middeck Active Control Experiment (MACE)," *Proceedings, 41st Congress of the IAF*, Dresden, GDR, 1990. IAF-90-292.

[6] Crawley, E. F., J. de Luis, and D. W. Miller, "Middeck Active Control Experiment (MACE) Phase A Final Report," Tech. Rep. SERC #16-90-R, Space Engineering Research Center, Massachusetts Institute of Technology, June 1989.

[7] Rey, D., *Gravity and Suspension Effects on Controlled Flexible Spacecraft*, Master's thesis, Massachusetts Institute of Technology, 1993.

[8] SDRC, *CORDS Users Manual.* San Diego, CA, 1992.

[9] Flanigan, C. C., "Test/Analysis Correlation Using Design Sensitivity and Optimization," *Proceedings of SAE*, 1987. Paper 871743.

[10] Balmès, E., *Experimental/Analytical Predictive Models of Damped Structural Systems*, Ph.D. thesis, Department of Aeronautics and Astronautics, M.I.T., Cambridge, MA, June 1993.

[11] Glaese, R. M. and D. W. Miller, "On-Orbit Modelling of the Middeck Active Control Experiment from 1-g Analysis and Experimentation," *Proceedings, 12th International Modal Analysis Conference*, Honolulu, Hawaii, Feb. 1994.

[12] SDRC, *I-DEAS Users Manual.* San Diego, CA, 1992.

[13] Rey, D., E. Crawley, H. Alexander, R. Glaese, and P. Gaudenzi, "Gravity and Suspension Effects on the Dynamics of Controlled Structures," *Proceedings, AIAA Structures, Structural Dynamics, and Materials Conference*, (La Jolla, CA), April 1993. AIAA Paper 93-1662.

[14] Houpis, C. and G. Lamont, *Digital Control Systems-theory, hardware, software*, McGraw-Hill Book Company, London, 1985.

[15] Kienholz, D. A., "A Pneumatic/Electric Suspension System for Simulating On-Orbit Conditions," *Proceedings, Winter Annual Meeting of the ASME*, Dallas, TX, Nov. 1990.

[16] Blevins, R. D., *Formulas for Natural Frequency and Mode Shape*, Robert E. Krieger Publishing Company, Malabar, Florida, 1979.

[17] Hyde, J., 1993. Personal Communication.

[18] Rey, D., H. L. Alexander, and E. F. Crawley, "The Direct Effects of Gravity on the Control and Output Matrices of Controlled Structure Models," *Proceedings, AIAA Dynamics Specialist Conference*, (Dallas, TX), April 1992.

[19] ADINA R&D, *ADINA Users Manual.* 71 Elton Avenue, Watertown, MA 02172, 1987.

[20] de Luis, J., 1992. Personal Communication.

[21] Ewins, D., *Modal Testing: Theory and Practice*, John Wiley and Sons, Inc., New York, NY, 1984.

[22] McCulloch, C., P. Vanhonacker, and E. Dascotte, "Validating and Updating Finite Element Models Using Experimental Modal Analysis," *Environmental Engineering*, Vol. 4, No. 2, June 1991, pp. 18–19,22–2.

[23] Lin, R., H. Du, and J. Ong, "Sensitivity Based Method for Structural Dynamic Model Improvement," *Computational Structures*, Vol. 47, No. 3, 1993, pp. 349–369.

[24] Flanigan, C., "Correlation of Finite Element Models Using Modeshape Design Sensitivity," *Proceedings, 9th International Modal Analysis Conference*, Apr. 1991.

[25] Ting, T., T. Chen, and W. Twomey, "Finite Element Model Refinement with Actual Forced Responses of Structures," *Finite Element Analysis and Design*, Vol. 11, No. 3, July 1992, pp. 213–220.

[26] Visser, W. and M. Imregun, "Technique to Update Finite Element Models Using Frequency Response Data," *Proceedings, 9th International Modal Analysis Conference*, Apr. 1991.

[27] Fritzen, C. and S. Zhu, "Updating of Finite Element Models By Means of Measured Information," *Computational Structures*, Vol. 40, No. 2, 1991.

[28] Balmès, E., "A Finite Element Updating Procedure Using Frequency Response Functions. Applications to the MIT/SERC Interferometer Testbed.," *Proceedings, 11th International Modal Analysis Conference*, Feb. 1993.

[29] Wilke, P., "SPICE Finite Element Model Correlation," *Proceedings, 11th International Modal Analysis Conference*, Feb. 1993.

[30] Saarmaa, E., "Middeck Active Control Experiment (MACE) Development Model (DM) Dynamic Testing and Modelling," Tech. Rep. MACE Document 1-170, Massachusetts Institute of Technology, October 1991.

[31] MacNeal-Schwendler Corporation, *NASTRAN Users Manual Version 67.* Los Angeles, 1993.

[32] Natke, H., "Updating Computational Models in the Frequency Domain Based on Measured Data: a Survey," *Probabilistic Engineering Mechanics*, Vol. 3, No. 1, 1988, pp. 28–35.

[33] Crawley, E. F. and S. R. Hall, "The Dynamics of Controlled Structures," Tech. Rep. SERC #10-91-I, M.I.T. Space Engineering Research Center, July 1991.

[34] The Math Works, Inc., *MATLAB User's Guide*, 1992.

[35] Grace, A., *Optimization Toolbox for Use with MATLAB*, 1992.

[36] How, J., R. Glaese, S. Grocott, and D. Miller, "Finite Element Model Based Robust Controllers for the Middeck Active Control Experiment," *American Control Conference*, 1994.

[37] Grocott, S. C., *Comparison of Control Techniques for Robust Performance on Uncertain Structural Systems*, Master's thesis, Massachusetts Institute of Technology, 1994.

[38] Crawley, E. F., M. S. Barlow, M. C. van Schoor, and A. S. Bicos, "Variation in the Modal Parameters of Space Structures," *Proceedings, AIAA Structures, Structural Dynamics, and Materials Conference*, (Dallas, TX), Apr. 1992.

[39] Maciejowski, J. M., *Multivariable Feedback Design*, Addison-Wesley, Wokingham, England, 1989.

[40] Moore, B., "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Trans. on Automatic Control*, Vol. AC–26, No. 1, Feb. 1981.

Appendix A

Modeling For Control

Since modern control design relies on an accurate model of the system, this model must be derived by either analytical or measurement approaches. These models, no matter how accurate, will have errors. These modeling errors will be of two varieties, namely mismodeled and unmodeled dynamics. Mismodeled dynamics arise when dynamics present in the structure are captured incorrectly in the model. This mismodeling error results in shifts in the frequencies of poles and zeros, as well as shifts in the modal residues. Unmodeled dynamics, on the other hand, result when dynamics appear in the structure, but are not captured in the model, possibly because they are too low or too high in frequency to appear in the experimental data. Often, these modeling errors result in a reduction of achieved performance in the closed-loop system, or in the extreme case, instability. The Linear Quadratic Gaussian (LQG) [39] control technique will be used to investigate the sensitivity of the closed-loop system to mismodeled dynamics, in the form of parameter errors, and unmodeled dynamics.

A.1 Sensitivity to Mismodeled Dynamics

In this section, the LQG cost is used to examine the sensitivity of LQG to parameter errors. The sensitivity is investigated by designing a compensator using a design model containing the uncertain parameters, implementing it on an evaluation model

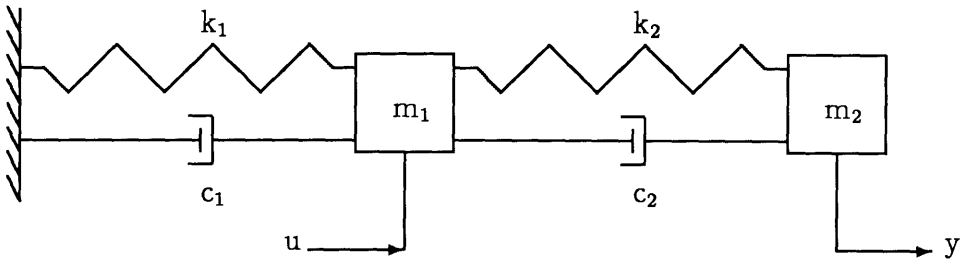


Figure A.1: Mass-Spring System

containing the actual parameters, and computing the closed-loop cost. The system used in this investigation is shown in Figure A.1. This system has known masses m_1=m_2=1 and dampers c_1=c_2=.01, but has uncertain stiffnesses k_1 and k_2 and actual stiffnesses k_1=.75, and k_2=1.2. The system has a force input on the first mass and a displacement output on the second mass. The model for this system is put into state-space form:

$$\begin{aligned} \dot{x} &= A(k)x + B(k)(w+u) \\ y &= C(k)x + v \end{aligned} \tag{A.1}$$

$$J = \int_0^\infty (x^T Q x + u^T R u)dt \tag{A.2}$$

where $A(k)$, $B(k)$, and $C(k)$ are functions of the uncertain parameters and w and v are Gaussian white noise processes with intensities of 1 and θ, respectively. The compensator was designed to minimize the cost in Equation A.2 with state weighting $Q = C(k)^T C(k)$ control weighting $R = \rho$. In order to calculate the closed-loop cost as a function of design model stiffnesses, compensators were designed and the closed-loop costs computed for design models in a grid of uncertain stiffnesses k_1 and k_2. Three levels of control authority (ρ) were examined. These correspond to cases when the bandwidth of the compensator is below the system dynamics (low), in the middle of the system dynamics (moderate), and above the system dynamics (high). The closed-loop costs were normalized by the cost associated with the controller designed using the evaluation model. The normalized costs for the three levels of control authority are shown in contour plots in Figures A.2 through A.4, where the X- and Y-axes represent changes in k_2 and k_1 away from the nominal stiffness values of 1.

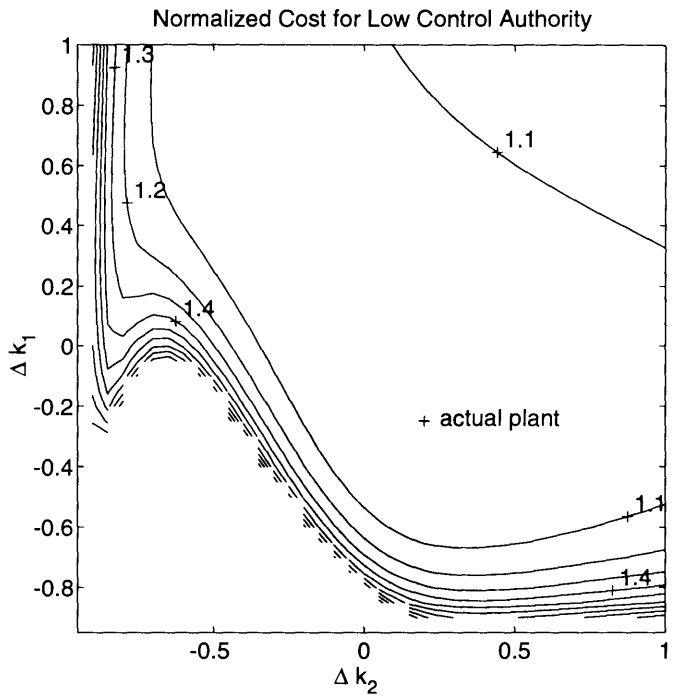


Figure A.2: Sensitivity of LQG to Parameter Errors for Low Authority

A.1.1 Low Authority

For the low authority case, in which the control has rolled off at a frequency below the system dynamics, the contour is relatively flat over most of the considered region. The lower left portion of the contour is a region of instability where the spring stiffnesses have been reduced to the point that the modeled system has dynamics at a low enough frequency that the control unstably interacts with the evaluation system and the low bandwidth assumption has been violated. The region in which the closed-loop cost is within 10% of the minimal cost (boundary marked 1.1) is rather large indicating that the LQG controller is insensitive to the model, so long as the dynamics are above some threshold frequency dependent on the level of control authority and the evaluation model.

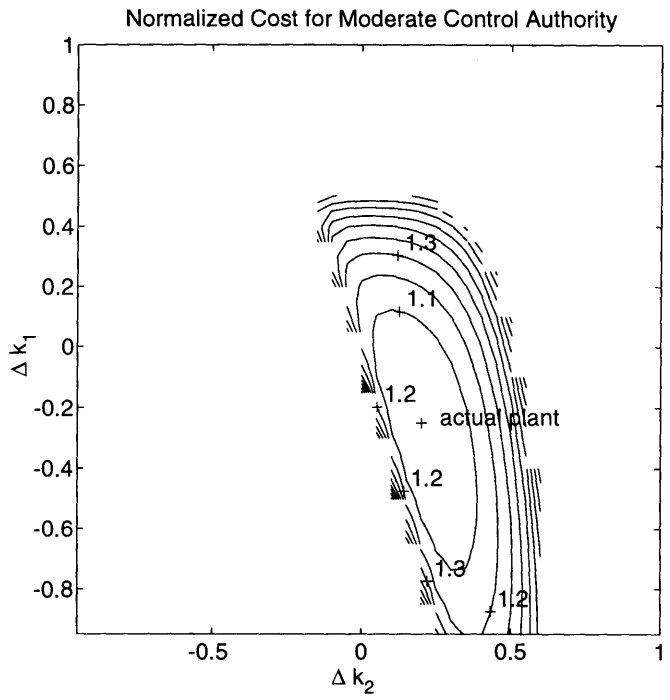


Figure A.3: Sensitivity of LQG to Parameter Errors for Moderate Authority

A.1.2 Moderate Authority

For the moderate control authority case, in which the control is rolling off in the middle of the system dynamics, the contour is bowl-shaped with a relatively small region of closed-loop stability. This contour plot indicates that in order to get tolerable performance (i.e.,< 110% of optimal performance) the region of allowable stiffnesses is rather small, ranging from 1.0 to 1.4 for k_2, indicating that LQG is not tolerant of mismodeling dynamics in the vicinity of the bandwidth. This region also indicates that getting k_2 correct is more important than getting k_1 correct. It is important to note that k_2 is the dominant parameter for the higher frequency mode of the system, indicating that it is important to accurately capture the modes above the bandwidth that could poke above 1 and result in instability. The case of dynamics within the bandwidth of the compensator is typical of controlled structures.

A.1.3 High Authority

For the high authority case, in which the control rolls off beyond all the system dynamics, the contour is valley-shaped. This again indicates that it is more important to model k_2 correctly. It is important to note that k_2 is between the actuator and sensor, which implies that it is most important to model the stiffness between the actuator and sensor correctly. At a closer contour interval, hints of a second, local, minimum are seen, which is indeed found when the search region on k_1 is extended to cover values from 2 to 3. This second minimum, which has a closed-loop cost 60% higher than the global minimum, results from the combination of stiffnesses that combine to give the same frequencies but different mode shapes. This second minimum is also found in the same place for the other control authorities. This second minimum indicates that although it is very important to correctly model the systems modal frequencies, it is also important to model the system mode shapes correctly. This conclusion would be even more apparent in a system with zeros, because the zero locations are not only influenced by the modal spacing, but also by the modal residues, which are due entirely to the mode shapes. The relevance of this control authority to controlled structures is almost nil, as it is impossible to be above all the dynamics of an infinite order system. This aspect of modeling an infinite order plant with a finite order model will be treated in the next section.

A.2 Sensitivity to Unmodeled Dynamics

Since structures are infinite order systems, all structural models are by definition reduced order models. For this reason, it is desirable to examine the implications of this reduced order modeling. To do this, the cantilever beam shown in Figure A.5 was examined. This is an aluminum beam having a rectangular cross-section with a one inch width and 1/8 inch thickness resulting in a fundamental frequency of 27.8 Hz. This model has been truncated to 5 modes to make it more manageable. The damping has been assumed to be 1% in all modes. The input is a tip force and the output is a tip displacement. A compensator was designed on the full order model such that

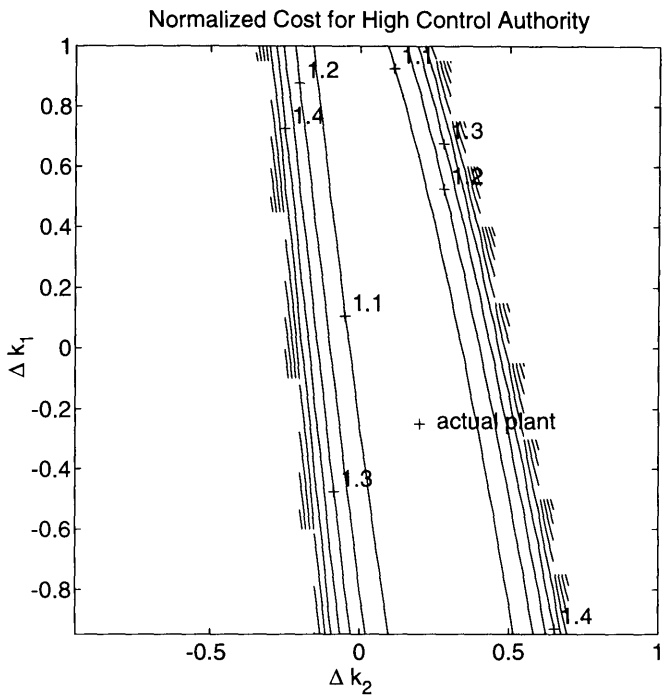


Figure A.4: Sensitivity of LQG to Parameter Errors for High Authority

it had crossover of the dereverberated loop transfer function at approximately the third mode. This is similar to what would happen in a typical controlled structure application. Then this model is reduced to 4 modes, or 8 states, using a balanced reduction [40], which is typically what is done by control designers when designing compensators based on reduced order models. Then, this reduced order model is used to design an LQG compensator using the same weighting as above. This reduced order compensator is implemented on the full order model and the resulting closed-loop cost is computed. The closed-loop LQG cost is used in an optimization routine with the constraint that the closed-loop system be stable to update the pole frequencies and damping ratios, and the B and C matrices in Equation A.1. This optimized model is then examined to determine which dynamics in a reduced order model give the best LQG performance.

The closed-loop costs for the balanced and updated models are shown in Table A.1, along with the closed-loop cost for the full-order model, to show the performance loss associated with reduced order modeling. Figure A.6 shows the open-loop transfer

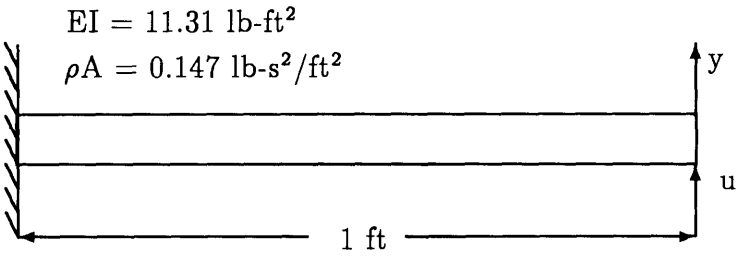


Figure A.5: Cantilever Beam

Table A.1: Closed-Loop Costs

$J_{full-order}$	$J_{balanced}$	$J_{updated}$
7.58x10-6	8.53x10-6	8.27x10-6

functions for the full-order model and balanced and updated models. This plot shows that the updated model has the final two lightly damped complex zeros damped slightly more than the initial guess. In addition, the final pole in the updated model is slightly more damped than in the balanced model. The pole frequencies do not change noticeably between the balanced and updated models, but note that the frequency of the final zero has decreased slightly. The most surprising change is the addition of a real nonminimum phase zero about a decade above the final pole, so that the open-loop transfer function rolls off with a slope of -1 rather than -2. To examine the reason for this nonminimum phase zero, look at the loop transfer function $G(j\omega)K(j\omega)$, where $G(j\omega)$ is the open-loop transfer function and $K(j\omega)$ is the compensator transfer function, shown in Figure A.7. Notice that the extra damping on the poles and zeros has pushed the magnitude of the loop transfer function down in those locations, pushing the magnitude away from 1 and increasing the gain margin there. But, there is still no clear reason for the additional open-loop nonminimum phase zero.

Now examine the two compensators. Figure A.8 shows the compensators designed using the balanced and updated models. It can be seen that the two compensators look very similar, with only the shifts in damping resulting from the increase in damping on the open-loop poles and zeros as discussed above. The compensator based on the updated model does nothing different in the region of the additional nonminimum phase zero than does the compensator based on the balanced model.

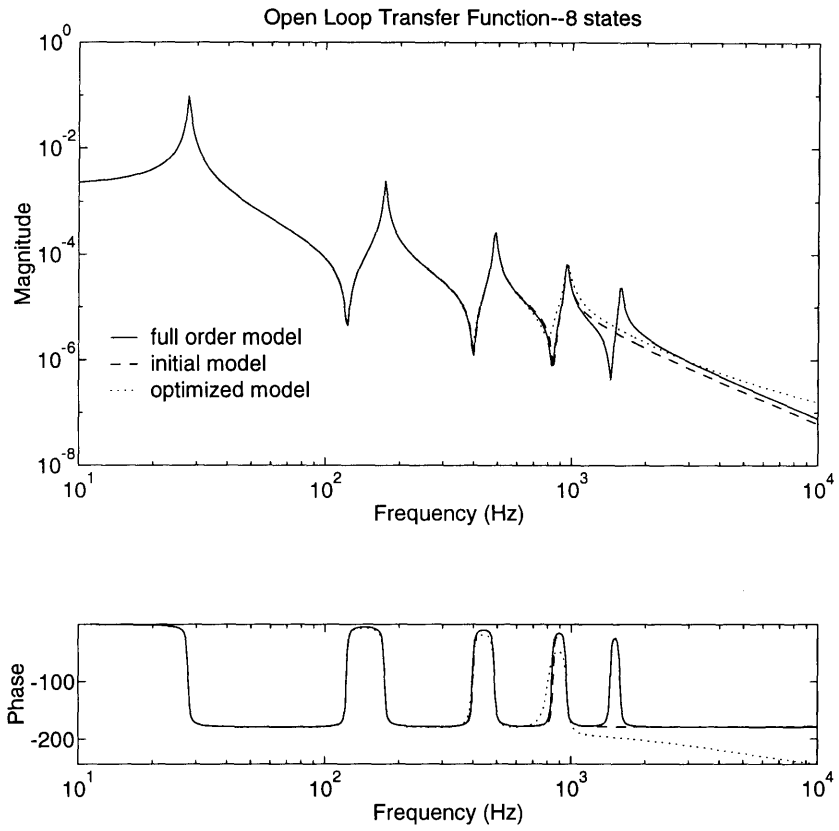


Figure A.6: Open-Loop Transfer Functions

This suggests that the high frequency real nonminimum phase zero is a by-product of the optimization process and probably results from allowing each entry in the B and C matrices to be free, giving the optimization the freedom to add an extra zero.

The implications of this reduced order model updating are to capture the pole frequencies nearly right, but to model the damping as higher than is actually present. The additional real nonminimum phase zero suggests that it is always best to have exactly one more pole than zero, even if this results in an extra zero in the right half plane. This would manifest itself in modeling the system with only resonant poles and zeros and adding a high frequency real zero about a decade above the last dynamics of interest. More investigation is needed to determine where the extra zero should be placed. Also note that the results from the previous section indicate that these conclusions are dependent on the level of control authority used. It is expected that the lower the control authority, the lower in frequency these results become

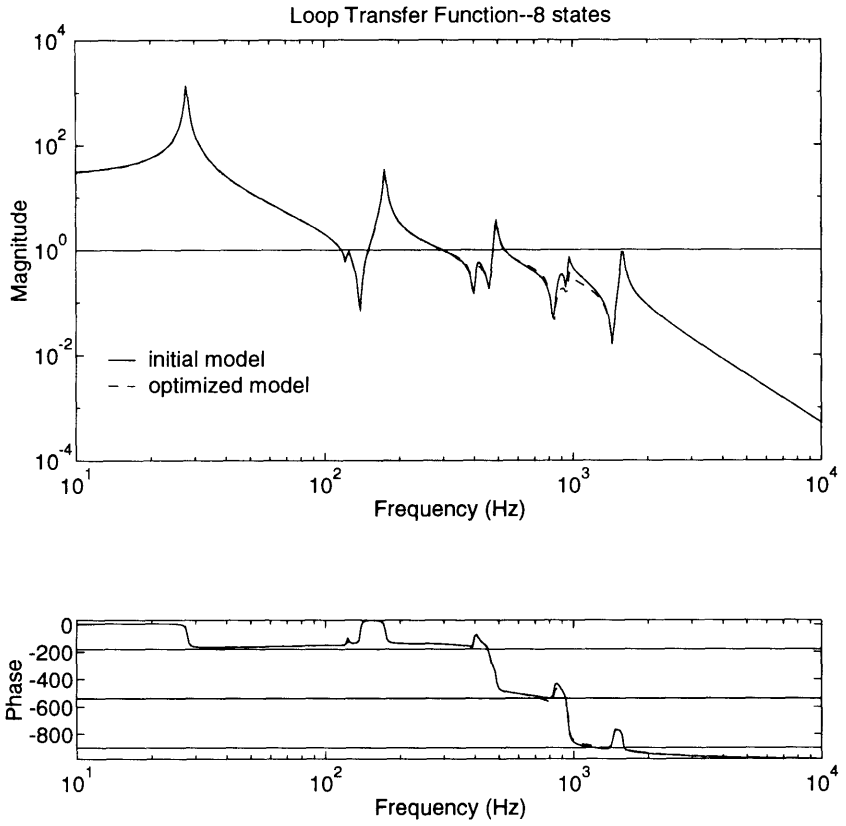


Figure A.7: Loop Transfer Functions

applicable. Investigation of the influence of the level of control authority is needed to verify this assertion. Since the level of control authority needed is seldom precisely known beforehand, the best course of action is to try to correlate the model to the experimental data as exactly as possible and once the control authority is known, iterate between control design and model updating.

Once the level of control authority is known, this procedure could be used to update the models of real systems. This update would result in a model that is optimized for a given set of LQG control weightings. The drawbacks to this would be a different model for each control weighting and control design algorithm. Another drawback is the need to have an exact model upon which the designed compensator can be implemented, which might be circumvented by obtaining the closed-loop cost and stability from the implementation of the compensator on experimental data. This update technique could be enhanced by reducing the number of optimization

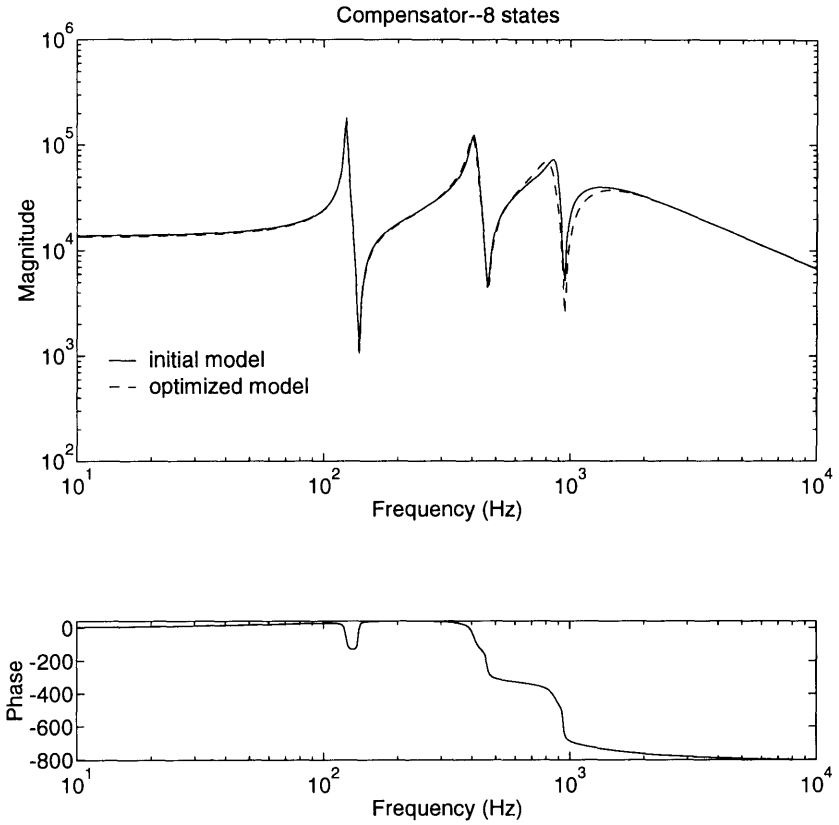


Figure A.8: Compensator Transfer Functions

parameters to only physical parameters and damping.

Appendix B

NASTRAN Input Deck

```
$
$ Need DATABASE solutions to obtain xdb file and SOL >100 must be used.
$
DBSETDEL USROBJ,USRSOU
ASSIGN DBC='runup.xdb'
$
SOL 106
TIME 2000
DIAG 8
$ Use custom DMAP for solution 106
INCLUDE '/home/roger/Mace/Model/Nastran/sol106.dmap'
$
CEND
$
TITLE = MACE Model
SUBTITLE = Gravity preloaded modes
$
PARAM,NMLOOP,24   $ SELECT MODAL STIFFNESS MATRIX
PARAM,NSUBCASE,2  $ SUBCASE WITH MODAL B.C.'S
METHOD=1
$
SUBCASE 1
SPC = 51
MPC = 5
LOAD = 100
NLPARM = 10
DISP=ALL
$
SUBCASE 2
LABEL = Free offload springs for dynamics
PARAM,EIGEN,1  $ JUMP OUT OF STATIC LOOP AND PERFORM EIGEN-ANALYSIS
SPC=53
MPC = 6
NLPARM=1  $ DUMMY NLPARM CARD (NOT USED)
DISP=ALL
$
```

```
BEGIN BULK
PARAM,POST,0 $ TURNS ON DBC
PARAM,DBCCONV,XL $ INSTRUCTS DBC TO GENERATE MSC/XL STYLE DATABASE
$
$*******************************************************************
$*** SOLUTION CONTROL
$*******************************************************************
$ GENERATE MASS PROPERTY INFO
PARAM   GRDPNT  16
$ AUTOMATIC CONSTRAINTS
PARAM   AUTOSPC NO
$ PRINT OUT INTERNAL & EXTERNAL DOF MAP (0=ROW LIST, -1=NONE)
PARAM   USETPRT -1
$ EXTRACT VIBRATION MODES
EIGRL   1       -1.0    250.    200             3
$ Nonlinear solution parameters
NLPARM  1       1
NLPARM  10      20              ITER    1       25      W
                        1.5e-4
PARAM   LGDISP  1
$ Static loading
GRAV    9999    0       9.807   0.0     -1.0    0.0
LOAD    100     1.0     1.0     9999
$
$ Skew coordinate systems aligned with axes of torque wheels
CORD2R          1       0     0.0     0.0     0.0     0.0 0.57729 0.81654
             0.0     1.0     0.0
CORD2R          2       0     0.0     0.0     0.0 0.70715 0.57729-0.40827
             0.0     1.0     0.0
CORD2R          3       0     0.0     0.0     0.0-0.70715 0.57729-0.40827
             0.0     1.0     0.0
$ 345678123456781234567812345678123456781234567812345678123456781234567812345678
GRID            1       0     0.0     0.0     0.0       0       0       0
GRID            2       0 0.03175     0.0     0.0       0       0       0
GRID            3       0 0.06350     0.0     0.0       0       0       0
GRID            4       0 0.10542     0.0     0.0       0       0       0
GRID            5       0 0.23438     0.0     0.0       0       0       0
GRID            6       0 0.23755     0.0     0.0       0       0       0
GRID            7       0 0.36968     0.0     0.0       0       0       0
GRID            8       0 0.41161     0.0     0.0       0       0       0
GRID            9       0 0.44336     0.0     0.0       0       0       0
GRID           10       0 0.47507     0.0     0.0       0       0       0
GRID           11       0 0.51703     0.0     0.0       0       0       0
GRID           12       0 0.64599     0.0     0.0       0       0       0
GRID           13       0 0.64916     0.0     0.0       0       0       0
GRID           14       0 0.78129     0.0     0.0       0       0       0
GRID           15       0 0.82321     0.0     0.0       0       0       0
GRID           16       0 0.85496     0.0     0.0       0       0       0
GRID           17       0 0.88671     0.0     0.0       0       0       0
GRID           18       0 0.92864     0.0     0.0       0       0       0
GRID           19       0 1.05759     0.0     0.0       0       0       0
GRID           20       0 1.06077     0.0     0.0       0       0       0
GRID           21       0 1.19290     0.0     0.0       0       0       0
GRID           22       0 1.23482     0.0     0.0       0       0       0
```

```
GRID          23        0 1.26657      0.0      0.0        0        0        0
GRID          24        0 1.29832      0.0      0.0        0        0        0
GRID          25        0 1.34025      0.0      0.0        0        0        0
GRID          26        0 1.47237      0.0      0.0        0        0        0
GRID          27        0 1.47555      0.0      0.0        0        0        0
GRID          28        0 1.60450      0.0      0.0        0        0        0
GRID          29        0 1.64643      0.0      0.0        0        0        0
GRID          30        0 1.67818      0.0      0.0        0        0        0
GRID          31        0 1.70993      0.0      0.0        0        0        0
$     Suspension Attach Points (bottom set)
GRID          32        0 0.03175 0.17975      0.0        0        0        0
GRID          33        0 0.85496 0.17975      0.0        0        0        0
GRID          34        0 1.67818 0.17975      0.0        0        0        0
$     Bus Rate Gyro
GRID          35        0 0.85496-0.07525      0.0        0        0        0
$     Primary Gimbal points
GRID          36        0 0.02894-0.10432-0.00116        0        0        0
GRID          37        0 0.13683-0.11163      0.0        0        0        0
GRID          38        0 0.13683-0.11163      0.0        0        0        0
GRID          39        0 0.09836-0.10999-0.00437        0        0        0
GRID          40        0 0.13683-0.11163      0.0        0        0        0
GRID          41        0 0.13683-0.11163      0.0        0        0        0
GRID          42        0 0.13768-0.14576-0.00005        0        0        0
GRID          43        0 0.13683-0.29483      0.0        0        0        0
$     Secondary Gimbal points
GRID          44        0 1.68099-0.10432 0.00116        0        0        0
GRID          45        0 1.57310-0.11163      0.0        0        0        0
GRID          46        0 1.57310-0.11163      0.0        0        0        0
GRID          47        0 1.61156-0.10999 0.00437        0        0        0
GRID          48        0 1.57310-0.11163      0.0        0        0        0
GRID          49        0 1.57310-0.11163      0.0        0        0        0
GRID          50        0 1.57226-0.14576 0.00005        0        0        0
GRID          51        0 1.57310-0.27673      0.0        0        0        0
$     Reaction Wheel Assembly points
GRID          52        0 0.85496 0.09454      0.0        0        0        0
GRID          53        0 0.85496 0.11242 0.10629        1        0        0
GRID          54        0 0.85496 0.11242 0.10629        1        0        0
GRID          55        0 0.84226 0.11255 0.10647        0        0        0
GRID          56        0 0.94701 0.11242-0.05314        2        0        0
GRID          57        0 0.94701 0.11242-0.05314        2        0        0
GRID          58        0 0.94082 0.11255-0.06423        0        0        0
GRID          59        0 0.76292 0.11242-0.05314        3        0        0
GRID          60        0 0.76292 0.11242-0.05314        3        0        0
GRID          61        0 0.76911 0.11255-0.06423        0        0        0
$     Reference Node
GRID          62        0     1.0      1.0      0.0        0        0        0
$     Suspension Cables
$     Cable 1
GRID          63        0 0.03175 4.91580      0.0        0        0        0
GRID          64        0 0.03175 4.32379      0.0        0        0        0
GRID          65        0 0.03175 3.73179      0.0        0        0        0
GRID          66        0 0.03175 3.13978      0.0        0        0        0
GRID          67        0 0.03175 2.54778      0.0        0        0        0
GRID          68        0 0.03175 1.95577      0.0        0        0        0
```

```
GRID           69       0 0.03175 1.36376     0.0       0       0       0
GRID           70       0 0.03175 0.77176     0.0       0       0       0
$    Cable 2
GRID           71       0 0.85496 4.91580     0.0       0       0       0
GRID           72       0 0.85496 4.32379     0.0       0       0       0
GRID           73       0 0.85496 3.73179     0.0       0       0       0
GRID           74       0 0.85496 3.13978     0.0       0       0       0
GRID           75       0 0.85496 2.54778     0.0       0       0       0
GRID           76       0 0.85496 1.95577     0.0       0       0       0
GRID           77       0 0.85496 1.36376     0.0       0       0       0
GRID           78       0 0.85496 0.77176     0.0       0       0       0
$    Cable 3
GRID           79       0 1.67818 4.91580     0.0       0       0       0
GRID           80       0 1.67818 4.32379     0.0       0       0       0
GRID           81       0 1.67818 3.73179     0.0       0       0       0
GRID           82       0 1.67818 3.13978     0.0       0       0       0
GRID           83       0 1.67818 2.54778     0.0       0       0       0
GRID           84       0 1.67818 1.95577     0.0       0       0       0
GRID           85       0 1.67818 1.36376     0.0       0       0       0
GRID           86       0 1.67818 0.77176     0.0       0       0       0
$    Ceiling Pneumatic "Spring" Attach Points
GRID           87       0 0.03175 9.49180     0.0       0       0       0
GRID           88       0 0.85496 9.49180     0.0       0       0       0
GRID           89       0 1.67818 9.49180     0.0       0       0       0
$    Suspension Attach Points (top set)
GRID           90       0 0.03175 0.17975     0.0       0       0       0
GRID           91       0 0.85496 0.17975     0.0       0       0       0
GRID           92       0 1.67818 0.17975     0.0       0       0       0
$    Additional strut nodes
GRID           93       0 0.12855     0.0     0.0       0       0       0
GRID           94       0 0.18305     0.0     0.0       0       0       0
GRID           95       0 0.29206     0.0     0.0       0       0       0
GRID           96       0 0.34656     0.0     0.0       0       0       0
GRID           97       0 0.53564     0.0     0.0       0       0       0
GRID           98       0 0.54756     0.0     0.0       0       0       0
GRID           99       0 0.59201     0.0     0.0       0       0       0
GRID          100       0 0.63646     0.0     0.0       0       0       0
GRID          101       0 0.66186     0.0     0.0       0       0       0
GRID          102       0 0.70631     0.0     0.0       0       0       0
GRID          103       0 0.75076     0.0     0.0       0       0       0
GRID          104       0 0.76269     0.0     0.0       0       0       0
GRID          105       0 0.95176     0.0     0.0       0       0       0
GRID          106       0 1.00627     0.0     0.0       0       0       0
GRID          107       0 1.11527     0.0     0.0       0       0       0
GRID          108       0 1.16977     0.0     0.0       0       0       0
GRID          109       0 1.36337     0.0     0.0       0       0       0
GRID          110       0 1.41787     0.0     0.0       0       0       0
GRID          111       0 1.52688     0.0     0.0       0       0       0
GRID          112       0 1.58138     0.0     0.0       0       0       0
$
$ 345678123456781234567812345678123456781234567812345678123456781234567812345678
$ Elements with Nodal Mass Descriptions
$ Nodal Mass Representation of Collar Assemblies
CONM2          1       3         0.17033     0.0     0.0     0.0
```

```
        4.21E-5         4.60E-5                 4.60E-5
CONM2         2       4         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2         3       7         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2         4       8         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2         5      10         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2         6      11         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2         7      14         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2         8      15         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2         9      17         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2        10      18         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2        11      21         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2        12      22         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2        13      24         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2        14      25         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2        15      28         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
CONM2        16      29         0.17033     0.0     0.0     0.0
        4.21E-5         4.60E-5                 4.60E-5
$ Nodal Mass Represenation of Nodes
CONM2        17       2         0.82997     0.0     0.0     0.0
        9.90E-4         1.22E-3                 1.30E-3
CONM2        18       9         0.76916     0.0     0.0     0.0
        9.67E-4         1.12E-3                 5.98E-4
CONM2        19      16         0.73279     0.0     0.0     0.0
        9.45E-4         7.73E-4                 8.67E-4
CONM2        20      23         0.71672     0.0     0.0     0.0
        6.68E-4         8.20E-4                 5.92E-4
CONM2        21      30         0.82977     0.0     0.0     0.0
        9.90E-4         1.22E-3                 1.30E-3
$    Suspension Attach Points
CONM2        22      32         0.07117     0.0     0.0     0.0
        1.31E-4         1.43E-6                 1.31E-4
CONM2        23      33         0.07117     0.0     0.0     0.0
        1.31E-4         1.43E-6                 1.31E-4
CONM2        24      34         0.07117     0.0     0.0     0.0
        1.31E-4         1.43E-6                 1.31E-4
$    Bus Rate Gyro
CONM2        25      35         1.32426     0.0     0.0     0.0
        1.98E-3         1.86E-3                 1.82E-3
$    Mass and Inertia Tensor-Primary Gimbal Stage (Base,Inner,Outer)
CONM2        26      36         1.33696     0.0     0.0     0.0
```

```
        2.58E-3-1.75E-5 3.04E-3 4.84E-5-2.47E-5 3.54E-3
CONM2         27      39         2.07568     0.0     0.0     0.0
        2.95E-3 1.06E-4 7.03E-3-3.21E-4-6.40E-5 6.78E-3
CONM2         28      42         1.23085     0.0     0.0     0.0
        5.05E-3-9.13E-5 1.65E-3 1.40E-7-5.80E-5 4.24E-3
$    Payload Rate Gyro
CONM2         29      43         1.19731     0.0     0.0     0.0
        1.95E-3         1.85E-3                 1.35E-3
$    Mass and Inertia Tensor-Secondary Gimbal Stage (Base,Inner,Outer)
CONM2         30      44         1.33696     0.0     0.0     0.0
        2.58E-3 1.75E-5 3.04E-3 4.84E-5 2.47E-5 3.54E-3
CONM2         31      47         2.07568     0.0     0.0     0.0
        2.95E-3-1.06E-4 7.03E-3-3.21E-4 6.40E-5 6.78E-3
CONM2         32      50         1.23085     0.0     0.0     0.0
        5.05E-3 9.13E-5 1.65E-3 1.40E-7 5.80E-5 4.24E-3
$    Secondary Gimbal Dummy Mass
CONM2         33      51         1.27286     0.0     0.0     0.0
        2.26E-3         2.41E-3                 6.79E-4
$    Reaction Wheel Assembly Base
CONM2         34      52         6.06178     0.0     0.0     0.0
        0.03555         0.05497                 0.03555
$    Reaction Wheel 1
CONM2         35      54         3.41683     0.0     0.0     0.0
        4.56E-3         4.56E-3                 5.69E-3
$    Reaction Wheel 2
CONM2         36      57         3.41683     0.0     0.0     0.0
        4.56E-3         4.56E-3                 5.69E-3
$    Reaction Wheel 3
CONM2         37      60         3.41683     0.0     0.0     0.0
        4.56E-3         4.56E-3                 5.69E-3
$    Carriage Masses
CONM2         38      63         1.31320     0.0     0.0     0.0
            0.0             0.0                     0.0
CONM2         39      71         1.31320     0.0     0.0     0.0
            0.0             0.0                     0.0
CONM2         40      79         1.31320     0.0     0.0     0.0
            0.0             0.0                     0.0
$
$ 345678123456781234567812345678123456781234567812345678123456781234567812345678
$    Element Construction of the Struts--end (Lexan)
CBAR          41       1       4      93     0.0     0.0     1.0
CBAR          42       1      96       7     0.0     0.0     1.0
CBAR          43       1      18     105     0.0     0.0     1.0
CBAR          44       1     108      21     0.0     0.0     1.0
CBAR          45       1      25     109     0.0     0.0     1.0
CBAR          46       1     112      28     0.0     0.0     1.0
$    Element Construction of the Struts--center (Lexan)
CBAR          47       2      93      94     0.0     0.0     1.0
CBAR          48       2      94       5     0.0     0.0     1.0
CBAR          49      11       5       6     0.0     0.0     1.0
CBAR          50       2       6      95     0.0     0.0     1.0
CBAR          51       2      95      96     0.0     0.0     1.0
CBAR          52       2     105     106     0.0     0.0     1.0
CBAR          53       2     106      19     0.0     0.0     1.0
```

```
CBAR          54      11       19       20      0.0      0.0      1.0
CBAR          55       2       20      107      0.0      0.0      1.0
CBAR          56       2      107      108      0.0      0.0      1.0
CBAR          57       2      109      110      0.0      0.0      1.0
CBAR          58       2      110       26      0.0      0.0      1.0
CBAR          59      11       26       27      0.0      0.0      1.0
CBAR          60       2       27      111      0.0      0.0      1.0
CBAR          61       2      111      112      0.0      0.0      1.0
$     Element Construction of the Collar Assemblies (Stainless Steel)
CBAR          62       3        3        4      0.0      0.0      1.0
CBAR          63       3        7        8      0.0      0.0      1.0
CBAR          64       3       10       11      0.0      0.0      1.0
CBAR          65       3       14       15      0.0      0.0      1.0
CBAR          66       3       17       18      0.0      0.0      1.0
CBAR          67       3       21       22      0.0      0.0      1.0
CBAR          68       3       24       25      0.0      0.0      1.0
CBAR          69       3       28       29      0.0      0.0      1.0
$     Element Representation of Nodes (Aluminum)
CBAR          70       4        1        2      0.0      0.0      1.0
CBAR          71       4        2        3      0.0      0.0      1.0
CBAR          72       4        8        9      0.0      0.0      1.0
CBAR          73       4        9       10      0.0      0.0      1.0
CBAR          74       4       15       16      0.0      0.0      1.0
CBAR          75       4       16       17      0.0      0.0      1.0
CBAR          76       4       22       23      0.0      0.0      1.0
CBAR          77       4       23       24      0.0      0.0      1.0
CBAR          78       4       29       30      0.0      0.0      1.0
CBAR          79       4       30       31      0.0      0.0      1.0
$     Element Representation of Attachment Plates (Aluminum)
CBAR          80       4        2       32      0.0      0.0      1.0
CBAR          81       4       16       33      0.0      0.0      1.0
CBAR          82       4       30       34      0.0      0.0      1.0
$     Bus Rate Gyro (Aluminum)
CBAR          83       4       16       35      0.0      0.0      1.0
$     Element Representation of Primary Gimbal (Aluminum)
CBAR          84       4        2       36      0.0      0.0      1.0
CBAR          85       4       36       38      0.0      0.0      1.0
CBAR          86       4       37       39      0.0      0.0      1.0
CBAR          87       4       39       41      0.0      0.0      1.0
CBAR          88       4       40       42      0.0      0.0      1.0
$     Payload Rate Gyro (Aluminum)
CBAR          89       4       42       43      0.0      0.0      1.0
$     Element Representation of Secondary Gimbal (Aluminum)
CBAR          90       4       30       44      0.0      0.0      1.0
CBAR          91       4       44       46      0.0      0.0      1.0
CBAR          92       4       45       47      0.0      0.0      1.0
CBAR          93       4       47       49      0.0      0.0      1.0
CBAR          94       4       48       50      0.0      0.0      1.0
$     Secondary Gimbal Dummy Mass (Aluminum)
CBAR          95       4       50       51      0.0      0.0      1.0
$     Element Representation of Reaction Wheel Assembly (Aluminum)
CBAR          96       4       16       52      0.0      0.0      1.0
CBAR          97       4       52       53      0.0      0.0      1.0
CBAR          98       4       52       56      0.0      0.0      1.0
```

```
CBAR          99       4      52      59     0.0     0.0     1.0
CBAR         100       4      54      55     0.0     0.0     1.0
CBAR         101       4      57      58     0.0     0.0     1.0
CBAR         102       4      60      61     0.0     0.0     1.0
$    Suspension Cables: Carriage to Test Article (GR/EP)
$    Cable 1
CBAR         103       5      90      70     0.0     0.0     1.0
CBAR         104       5      70      69     0.0     0.0     1.0
CBAR         105       5      69      68     0.0     0.0     1.0
CBAR         106       5      68      67     0.0     0.0     1.0
CBAR         107       5      67      66     0.0     0.0     1.0
CBAR         108       5      66      65     0.0     0.0     1.0
CBAR         109       5      65      64     0.0     0.0     1.0
CBAR         110       5      64      63     0.0     0.0     1.0
$    Cable 2
CBAR         111       6      91      78     0.0     0.0     1.0
CBAR         112       6      78      77     0.0     0.0     1.0
CBAR         113       6      77      76     0.0     0.0     1.0
CBAR         114       6      76      75     0.0     0.0     1.0
CBAR         115       6      75      74     0.0     0.0     1.0
CBAR         116       6      74      73     0.0     0.0     1.0
CBAR         117       6      73      72     0.0     0.0     1.0
CBAR         118       6      72      71     0.0     0.0     1.0
$    Cable 3
CBAR         119       7      92      86     0.0     0.0     1.0
CBAR         120       7      86      85     0.0     0.0     1.0
CBAR         121       7      85      84     0.0     0.0     1.0
CBAR         122       7      84      83     0.0     0.0     1.0
CBAR         123       7      83      82     0.0     0.0     1.0
CBAR         124       7      82      81     0.0     0.0     1.0
CBAR         125       7      81      80     0.0     0.0     1.0
CBAR         126       7      80      79     0.0     0.0     1.0
$    Element Construction of the Base Active Strut (Lexan)
CBAR         127       8      11      97     0.0     0.0     1.0
CBAR         128       8     104      14     0.0     0.0     1.0
$    Element Construction of the Active Strut (Piezo)
CBAR         129       9      98      99     0.0     0.0     1.0
CBAR         130       9      99     100     0.0     0.0     1.0
CBAR         131       9     101     102     0.0     0.0     1.0
CBAR         132       9     102     103     0.0     0.0     1.0
$    Element Construction of the Center Active Strut (Lexan)
CBAR         133      10      97      98     0.0     0.0     1.0
CBAR         134      10     100      12     0.0     0.0     1.0
CBAR         135      12      12      13     0.0     0.0     1.0
CBAR         136      10      13     101     0.0     0.0     1.0
CBAR         137      10     103     104     0.0     0.0     1.0
$ Tuned suspension pneumatic springs
CELAS2       138 16.9878      63       2
CELAS2       139 56.2820      71       2
CELAS2       140 17.0381      79       2
$ 345678123456781234567812345678123456781234567812345678123456781234567812345678
$ The Beam Properties are Provided in the following Cards
$     Card Number,ID,A,Iy,Iz,J
PBAR           1       1 2.56E-4 2.14E-8 2.14E-8 4.27E-8
```

```
        0.54921 0.54921
PBAR           2       2 2.56E-4 2.14E-8 2.14E-8 4.27E-8

        0.54921 0.54921
PBAR           3       3 1.55E-3 1.92E-7 1.92E-7 3.83E-7
PBAR           4       4 2.89E-3 1.25E-6 1.25E-6 2.50E-6
PBAR           5       5 7.92E-6 4.99-12 4.99-12 9.98-12
PBAR           6       6 7.92E-6 4.99-12 4.99-12 9.98-12
PBAR           7       7 7.92E-6 4.99-12 4.99-12 9.98-12
PBAR           8       8 2.56E-4 2.14E-8 2.14E-8 4.27E-8

        0.54921 0.54921
PBAR           9       9 4.79E-4 5.14E-8 5.14E-8 1.03E-7

        0.56459 0.56459
PBAR          10      10 4.79E-4 5.14E-8 5.14E-8 1.03E-7

        0.56459 0.56459
PBAR          11       2 2.56E-4 2.14E-8 2.14E-8 4.27E-8
PBAR          12      10 4.79E-4 5.14E-8 5.14E-8 1.03E-7
$ These cards provide the Material Properties
$     ID,Young's Modulus, Poisson's Ratio, Density
MAT1           1 2.628E9            0.37 2021.77
MAT1           2 2.335E9            0.37 2021.77
MAT1           3 127.1E9            0.30     0.0
MAT1           4 68.90E9            0.30     0.0
MAT1           5 198.2E9            0.30 1991.37
MAT1           6 171.2E9            0.30 1991.37
MAT1           7 184.2E9            0.30 1991.37
MAT1           8 2.767E9            0.37 1838.18
MAT1           9 3.783E9            0.37 1968.12
MAT1          10 2.662E9            0.37 1838.18
$
$   Inner Primary Gimbal (mechanism = 4)
MPC            5      37       1     1.0      38       1    -1.0
MPC            5      37       2     1.0      38       2    -1.0
MPC            5      37       3     1.0      38       3    -1.0
MPC            5      37       4     1.0      38       4    -1.0
MPC            5      37       5     1.0      38       5    -1.0
MPC            5      37       6     1.0      38       6    -1.0
$   Outer Primary Gimbal (mechanism = 6)
MPC            5      40       1     1.0      41       1    -1.0
MPC            5      40       2     1.0      41       2    -1.0
MPC            5      40       3     1.0      41       3    -1.0
MPC            5      40       4     1.0      41       4    -1.0
MPC            5      40       5     1.0      41       5    -1.0
MPC            5      40       6     1.0      41       6    -1.0
$   Inner Secondary Gimbal (mechanism = 4)
MPC            5      45       1     1.0      46       1    -1.0
MPC            5      45       2     1.0      46       2    -1.0
MPC            5      45       3     1.0      46       3    -1.0
MPC            5      45       4     1.0      46       4    -1.0
MPC            5      45       5     1.0      46       5    -1.0
```

```
MPC             5       45      6       1.0     46      6       -1.0
$   Outer Secondary Gimbal (mechanism = 6)
MPC             5       48      1       1.0     49      1       -1.0
MPC             5       48      2       1.0     49      2       -1.0
MPC             5       48      3       1.0     49      3       -1.0
MPC             5       48      4       1.0     49      4       -1.0
MPC             5       48      5       1.0     49      5       -1.0
MPC             5       48      6       1.0     49      6       -1.0
$   Reaction Wheel 1 (mechanism = 6)
MPC             5       54      1       1.0     53      1       -1.0
MPC             5       54      2       1.0     53      2       -1.0
MPC             5       54      3       1.0     53      3       -1.0
MPC             5       54      4       1.0     53      4       -1.0
MPC             5       54      5       1.0     53      5       -1.0
MPC             5       54      6       1.0     53      6       -1.0
$   Reaction Wheel 2 (mechanism = 6)
MPC             5       57      1       1.0     56      1       -1.0
MPC             5       57      2       1.0     56      2       -1.0
MPC             5       57      3       1.0     56      3       -1.0
MPC             5       57      4       1.0     56      4       -1.0
MPC             5       57      5       1.0     56      5       -1.0
MPC             5       57      6       1.0     56      6       -1.0
$   Reaction Wheel 3 (mechanism = 6)
MPC             5       60      1       1.0     59      1       -1.0
MPC             5       60      2       1.0     59      2       -1.0
MPC             5       60      3       1.0     59      3       -1.0
MPC             5       60      4       1.0     59      4       -1.0
MPC             5       60      5       1.0     59      5       -1.0
MPC             5       60      6       1.0     59      6       -1.0
$   Suspension Attach Point 1 (mechanisms = 4,6)
MPC             5       90      1       1.0     32      1       -1.0
MPC             5       90      2       1.0     32      2       -1.0
MPC             5       90      3       1.0     32      3       -1.0
MPC             5       90      4       1.0     32      4       -1.0
MPC             5       90      5       1.0     32      5       -1.0
MPC             5       90      6       1.0     32      6       -1.0
$   Suspension Attach Point 2 (mechanisms = 4,6)
MPC             5       91      1       1.0     33      1       -1.0
MPC             5       91      2       1.0     33      2       -1.0
MPC             5       91      3       1.0     33      3       -1.0
MPC             5       91      4       1.0     33      4       -1.0
MPC             5       91      5       1.0     33      5       -1.0
MPC             5       91      6       1.0     33      6       -1.0
$   Suspension Attach Point 3 (mechanisms = 4,6)
MPC             5       92      1       1.0     34      1       -1.0
MPC             5       92      2       1.0     34      2       -1.0
MPC             5       92      3       1.0     34      3       -1.0
MPC             5       92      4       1.0     34      4       -1.0
MPC             5       92      5       1.0     34      5       -1.0
MPC             5       92      6       1.0     34      6       -1.0
$
$   Inner Primary Gimbal (mechanism = 4)
MPC             6       37      1       1.0     38      1       -1.0
MPC             6       37      2       1.0     38      2       -1.0
```

```
MPC             6       37      3       1.0     38      3       -1.0
MPC             6       37      5       1.0     38      5       -1.0
MPC             6       37      6       1.0     38      6       -1.0
$   Outer Primary Gimbal (mechanism = 6)
MPC             6       40      1       1.0     41      1       -1.0
MPC             6       40      2       1.0     41      2       -1.0
MPC             6       40      3       1.0     41      3       -1.0
MPC             6       40      4       1.0     41      4       -1.0
MPC             6       40      5       1.0     41      5       -1.0
$   Inner Secondary Gimbal (mechanism = 4)
MPC             6       45      1       1.0     46      1       -1.0
MPC             6       45      2       1.0     46      2       -1.0
MPC             6       45      3       1.0     46      3       -1.0
MPC             6       45      5       1.0     46      5       -1.0
MPC             6       45      6       1.0     46      6       -1.0
$   Outer Secondary Gimbal (mechanism = 6)
MPC             6       48      1       1.0     49      1       -1.0
MPC             6       48      2       1.0     49      2       -1.0
MPC             6       48      3       1.0     49      3       -1.0
MPC             6       48      4       1.0     49      4       -1.0
MPC             6       48      5       1.0     49      5       -1.0
$   Reaction Wheel 1 (mechanism = 6)
MPC             6       54      1       1.0     53      1       -1.0
MPC             6       54      2       1.0     53      2       -1.0
MPC             6       54      3       1.0     53      3       -1.0
MPC             6       54      4       1.0     53      4       -1.0
MPC             6       54      5       1.0     53      5       -1.0
$   Reaction Wheel 2 (mechanism = 6)
MPC             6       57      1       1.0     56      1       -1.0
MPC             6       57      2       1.0     56      2       -1.0
MPC             6       57      3       1.0     56      3       -1.0
MPC             6       57      4       1.0     56      4       -1.0
MPC             6       57      5       1.0     56      5       -1.0
$   Reaction Wheel 3 (mechanism = 6)
MPC             6       60      1       1.0     59      1       -1.0
MPC             6       60      2       1.0     59      2       -1.0
MPC             6       60      3       1.0     59      3       -1.0
MPC             6       60      4       1.0     59      4       -1.0
MPC             6       60      5       1.0     59      5       -1.0
$   Suspension Attach Point 1 (mechanisms = 4,6)
MPC             6       90      1       1.0     32      1       -1.0
MPC             6       90      2       1.0     32      2       -1.0
MPC             6       90      3       1.0     32      3       -1.0
MPC             6       90      5       1.0     32      5       -1.0
$   Suspension Attach Point 2 (mechanisms = 4,6)
MPC             6       91      1       1.0     33      1       -1.0
MPC             6       91      2       1.0     33      2       -1.0
MPC             6       91      3       1.0     33      3       -1.0
MPC             6       91      5       1.0     33      5       -1.0
$   Suspension Attach Point 3 (mechanisms = 4,6)
MPC             6       92      1       1.0     34      1       -1.0
MPC             6       92      2       1.0     34      2       -1.0
MPC             6       92      3       1.0     34      3       -1.0
MPC             6       92      5       1.0     34      5       -1.0
```

```
$
$ 345678123456781234567812345678123456781234567812345678123456781234567812345678
$ Fix all dofs at top of rods
SPC1    51      123456  63      71      79
SPC1    51      123456  62      87      88      89
$ Fix all dofs at top of rods, except allow vertical spring to move
SPC1    53      13456   63      71      79
SPC1    53      123456  62      87      88      89
$
ENDDATA
```

Appendix C

MATLAB Code to Generate Input-Output Model

```
function [A,B,C,D]=...
abcdmace(evalfem,config,NM,n0,indd,indr,inda,ict,volts,integ,igm,...
isensd,idelay,ibess,NP,dls,cfb,dlk)
%-------------------------------------------------------------------------
%
%  This function evaluates the MIMO MACE model from the FEM model
%
%  The inputs are:  'evalfem'(String variable to load FEM-file)
%                   config   (==1-DM,==2-DM/EM,==3-EM (w/DM))
%                   NM       (Numbers of modes to include in model)
%                   n0       (number of rigid body modes)
%                   For list of DM sensors and actuators see fdm1.m
%                   For list of DM/EM sensors and actuators see fdm2.m
%                   For list of EM sensors and actuators see fdm3.m
%                   volts    (==1 - Volts; ==0 - Physical units)
%                   integ    (==0 - Stabilized integrators are off
%                             ==1 - stabilized integrators are on,
%                                   but dynamics are not accounted for
%                             ==2 - stabilized integrators are on,
%                                   dynamics are included in A,B,C and D)
%                   igm      (==1 - all servos on, ==0 - servos off)
%                   isensd   (==1 - sensor dynamics, ==0 - no sensor dynamics)
%                   idelay   (==1 - time delays, ==0 - no time delays)
%                   ibess    (==1 - Bessel Filters, ==0 - no Bessel Filters)
%                   NP       (Order of Pade approximation)
%                   dls      (Vector of time delays)
%                   cfb      (Vector of Bessel Filter corner frequencies)
%                   dlk      (Servo loop time delay in gimbals)
%
%  The outputs are: A,B,C and D matrices for MIMO MACE model
%
%  Call:     [A,B,C,D]=...
%     abcdmace(evalfem,config,NM,n0,indd,indr,inda,ict,volts,integ,igm,...
%     isensd,idelay,ibess,NP,dls,cfb,dlk)
%-------------------------------------------------------------------------
```

```
if config==1;
  disp('This is for the 1 Payload Case (DM)');
  [w,z,bd,br,ba,bct]=fdm1(evalfem,NM,indd,indr,inda,ict);
elseif config==2;
  disp('This is for the 2 Payload Case (DM/EM)');
  [w,z,bd,br,ba,bct]=fdm2(evalfem,NM,indd,indr,inda,ict);
elseif config==3;
  disp('This is for the 2 Payload Case (EM)');
  [w,z,bd,br,ba,bct]=fdm3(evalfem,NM,indd,indr,inda,ict);
end
disp(' ')
if volts==0;
  disp('Matrices A,B,C and D are in Physical Units');
elseif volts==1;
  disp('Matrices A,B,C and D are in Volts');
end
disp(' ')
if integ==0;
  disp('Stabilized integrators are turned OFF');
elseif integ==1;
  disp('Stabilized integrators are ON');
  disp('Their dynamics are NOT included');
elseif integ==2;
  disp('Stabilized integrators are ON');
  disp('Their dynamics ARE included');
end
disp(' ')
if isensd==0;
  disp('NO sensor dynamics');
else
  disp('Sensor dynamics are included');
end
disp(' ')
if idelay==0;
  disp('NO time delays included');
else
  disp('Time DELAYS included');
end
disp(' ')
if ibess==0;
  disp('Bessel Filters are turned OFF');
else
  disp('Bessel Filters are turned ON');
end
disp(' ')
if igm==0;
  disp('SERVO loop NOT closed');
else
  disp('SERVO loop closed');
end
disp(' ')
end
Nm=length(NM)-n0;
w=w(n0+1:Nm+n0);
```

```
z=z(n0+1:Nm+n0);
[Md,Mr,Ma,Mct]=nums(indd,indr,inda,ict);
[Cdx,Crv,Cax,Cav,Ba]=cm1(length(NM),Md,Mr,Ma,Mct,w,z,bd,br,ba,bct);
% Form system with block diagonal rigid and flexible systems
Ar=[zeros(n0,n0) eye(n0);zeros(n0,2*n0)];
Br=[zeros(n0,Mct);Ba(1:n0,:)];
Af=[zeros(Nm,Nm) eye(Nm);-diag(w.^2) -2*diag(w.*z)];
Bf=[zeros(Nm,Mct); Ba(n0+1:Nm+n0,:)];
A=[Ar zeros(2*n0,2*Nm);zeros(2*Nm,2*n0) Af];
B=[Br;Bf];
if Md>0;
  Cdr=[Cdx(:,1:n0) zeros(Md,n0)];
  Cdf=[Cdx(:,n0+1:n0+Nm) zeros(Md,Nm)];
  Cd=[Cdr Cdf];
  Dd=zeros(Md,Mct);
end
if Mr>0
  if integ==0|integ==2;
    Crr=[zeros(Mr,n0) Crv(:,1:n0)];
    Crf=[zeros(Mr,Nm) Crv(:,n0+1:n0+Nm)];
    Cr=[Crr Crf];
  end
  if integ==1;
    Crr=[Crv(:,1:n0) zeros(Mr,n0)];
    Crf=[Crv(:,n0+1:n0+Nm) zeros(Mr,Nm)];
    Cr=[Crr Crf];
  end
  Dr=zeros(Mr,Mct);
end
if Ma>0;
  Car=[Cax(:,1:n0) Cav(:,1:n0)];
  Caf=[Cax(:,n0+1:n0+Nm) Cav(:,n0+1:n0+Nm)];
  Ca=[Car Caf];
  Da=reshape(ba,Nm,Ma)'*reshape(bct,Nm,Mct);
end
C=assemy(Md,Mr,Ma,Mct,Cd',Cr',Ca')';
D=assemy(Md,Mr,Ma,Mct,Dd',Dr',Da')';
if volts==1;
  [B,C,D]=resabcd(indd,indr,inda,ict,B,C,D,config,2*length(NM));
end
[A,B,C,D]=modext(isensd,idelay,integ,ibess,A,B,C,D,indd,indr,NP,dls,cfb,config);
if igm==1;
  if (config==2)
    [A,B,C,D]=gimcon(A,B,C,D,NM,n0,indd,ict,dlk,NP,evalfem,volts,config);
  elseif (config==3)
    [A,B,C,D]=gimcon3(A,B,C,D,NM,n0,indd,indr,ict,dlk,NP,evalfem,volts,config);
  end
end

function y=assemy(Md,Mr,Ma,Mct,yd,yr,ya)
%-------------------------------------------------------------------------
%
%  This function is used to assemble matrices C, gains G and K
%
```

```
%  The inputs are:   Md   (Number of displacement sensors)
%                    Mr   (Number of rate sensors)
%                    Ma   (Number of acceleration sensors)
%                    Mct  (Number of actuators)
%                    yd   (Displacement output)
%                    yr   (Rate output)
%                    ya   (Acceleration output)
%
%  The outputs are:  y    (assembled vector)
%
%  Call:   y=assemy(Md,Mr,Ma,Mct,yd,yr,ya)
%-------------------------------------------------------------------------

if Md>0;
  y=yd;
end
if Mr>0;
  if Md==0;
    y=yr;
  else;
    y=[y yr];
  end;
end
if Ma>0;
  if Md==0 & Mr==0;
    y=ya;
  else;
    y=[y ya];
  end;
end;

function [ab,bb,cb,db,in]=besfid(iii,cf)
%-------------------------------------------------------------------------
%
%  This function provides state space representations of the Bessel Filters
%
%  The inputs are:  iii   (Numbers of sensors)
%                   cf    (Vector of corner frequencies in Hz)
%                   in    (Numbers of system inputs with nonzero corners)
%
%  The outputs are: ad,bd,cd,dd (Matrices in linear system)
%
%  Call:   [ab,bb,cb,db]=besfid(iii,cf)
%-------------------------------------------------------------------------

N=length(iii);
if N>=1;
j=1;
  for i=1:N
    cfi=cf(i);
    if (cfi~=0.)
      [abi,bbi,cbi,dbi]=bess(8,cfi*2*pi);
      Nab=length(ab);
      ab=[ab zeros(Nab,8);zeros(8,Nab) abi];
```

```
      Ndt=8*(j-1);
      bb=[bb zeros(Ndt,1); zeros(8,j-1) bbi];
      cb=[cb zeros(j-1,8); zeros(1,Ndt) cbi];
      d(j)=dbi;
      db=diag(d);
      in=[in iii(i)];
      j=j+1;
    end
  end
end

function [ab,bb,cb,db]=bess(n,cf);
%-------------------------------------------------------------------------------
%
%  This function computes the Bessel Filter
%
%  The inputs are:  n   (Numbers of poles)
%                   cf  (corner frequencies)
%
%  The outputs are: ad,bd,cd,dd (Matrices in linear system)
%
%  Call:   [ab,bb,cb,db]=bess(n,cf)
%-------------------------------------------------------------------------------

for k=0:n,
 d(k+1)=fact(2*n-k)/(2^(n-k)*fact(k)*fact(n-k));
end;
den=fliplr(flipud(d));
[ab,bb,cb,db]=tf2ss(d(1),den);
cf1=d(1)^(1/n);
scale=cf1/(cf*2);
ab(1:n,:)=ab(1:n,:)/scale;
bb=bb/scale;

function [Cdx,Crv,Cax,Cav,Ba]=cm1(Nm,Md,Mr,Ma,Mct,w,z,bd,br,ba,bct)
%-------------------------------------------------------------------------------
%
%  The inputs are:  Nm      (Number of modes)
%                   Md      (Number of displacement sensors)
%                   Mr      (Number of rate sensors)
%                   Ma      (Number of acceleration sensors)
%                   Mct     (Number of actuators)
%                   w       (Vector of frequencies)
%                   z       (Vector of damping ratios)
%                   bd      (Vector of mode shapes in displacement sensors)
%                   br      (Vector of mode shapes in rate sensors)
%                   ba      (Vector of mode shapes in acceleration sensors)
%                   bct     (Vector of mode shapes in actuators)
%
%  The outputs are: Non-zero blocks in matrix B & C
%
%  Call:  [Cdx,Crv,Cax,Cav,Ba]=cm1(Nm,Md,Mr,Ma,Mct,w,z,bd,br,ba,bct)
%-------------------------------------------------------------------------------
```

```
if Md>0;
  Cdx=reshape(bd,Nm,Md)';
end
if Mr>0;
  Crv=reshape(br,Nm,Mr)';
end
if Ma>0
  ea=ones(Ma,1);
  w2=-w.^2;
  zw=-2*z.*w;
  C=reshape(ba,Nm,Ma)';
  Cax=C.*(ea*w2');
  Cav=C.*(ea*zw');
end
if Mct>0
  Ba=reshape(bct,Nm,Mct);
end

function [ad,bd,cd,dd,in]=delfid(iii,NP,dl)
%-------------------------------------------------------------------------------
%
%  This function provides Pade approximation for N sensors
%
%  The inputs are:  iii   (Numbers of sensors)
%                   NP    (Order of Pade approximation)
%                   dl    (Vector of time delays in each device)
%                   in    (Numbers of system inputs with nonzero time delays)
%
%  The outputs are: ad,bd,cd,dd (Matrices in linear system)
%
%  Call:   [ad,bd,cd,dd]=delfid(iii,NP,dl)
%-------------------------------------------------------------------------------

N=length(iii);
if N>=1;
j=1;
  for i=1:N
    dli=dl(i);
    if (dli~=0.)
      [adi,bdi,cdi,ddi]=pade(dli,NP);
      Nad=length(ad);
      ad=[ad zeros(Nad,NP);zeros(NP,Nad) adi];
      Ndt=NP*(j-1);
      bd=[bd zeros(Ndt,1); zeros(NP,j-1) bdi];
      cd=[cd zeros(j-1,NP); zeros(1,Ndt) cdi];
      d(j)=ddi;
      dd=diag(d);
      in=[in iii(i)];
      j=j+1;
    end
  end
end

function [w,z,bd,br,ba,bct]=fdm1(evalfem,NM,indd,indr,inda,ict)
```

```
%-------------------------------------------------------------------------
%
%  This function selects the proper degrees of freedom for the sensors
%  and actuators from the FEM  modeshapes for the 1 payload case (DM)
%
%  The inputs are: 'evalfem'(String variable to load FEM-file)
%                  NM       (Number of modes, any modes from given ones)
%                  indd     (Numbers of displacement sensors)
%                           1    - Inner encoder on PL (DM) (x)
%                           2    - Outer encoder on PL (DM) (z)
%                           3-10 - Strain Gauges (vertical and horizontal)
%                                  in struts 1-4
%                  indr     (Numbers of rate sensors)
%                           1-3  - Rate gyros on PL (DM) (x, y, z axis)
%                           4-6  - Rate gyros on TWA (x, y, z axis)
%                  inda     (Numbers of acceleration sensors)
%                           1-3  - Accelerometer in the node 2 (x, y, z axis)
%                           4-6  - Accelerometer in the node 4 (x, y, z axis)
%                  ict      (Numbers of actuators)
%                           1    - Inner gimbal on PL (DM) (x)
%                           2    - Outer gimbal on PL (DM) (z)
%                           3-5  - TWA (x, y, z axis)
%
%  The outputs are: w       (Vector of frequencies)
%                   z       (Vector of damping ratios)
%                   bd      (Vector of mode shapes in displacement sensors)
%                   br      (Vector of mode shapes in rate sensors)
%                   ba      (Vector of mode shapes in acceration sensors)
%                   bct     (Vector of mode shapes in actuators)
%
%  Call:            [w,z,bd,br,ba,bct]=fdm1(evalfem,NM,indd,indr,inda,ict)
%-------------------------------------------------------------------------

eval(evalfem)

if NM==0
  Nm=length(freq);
  NM=1:Nm;
end

[Md,Mr,Ma,Mct]=nums(indd,indr,inda,ict);
Nm=length(NM);
Mdm=Md*Nm;
Mrm=Mr*Nm;
Mam=Ma*Nm;
Mctm=Mct*Nm;
rad=.0127;
glength=.00635;
strg=2*rad/glength;
RM=rotmat;            % Rotation matrix in TWA

w=freq(NM);
z=zeta(NM);
```

```
% Displacement mode shapes
bbd(1,:)=modes(238,NM)-modes(316,NM);
bbd(2,:)=-modes(258,NM)+modes(324,NM);
bbd(3,:)=modes(114,NM)-modes(348,NM);
bbd(4,:)=modes(113,NM)-modes(347,NM);
bbd(5,:)=modes(84,NM)-modes(342,NM);
bbd(6,:)=modes(83,NM)-modes(341,NM);
bbd(7,:)=modes(336,NM)-modes(54,NM);
bbd(8,:)=modes(53,NM)-modes(335,NM);
bbd(9,:)=modes(24,NM)-modes(330,NM);
bbd(10,:)=modes(23,NM)-modes(329,NM);
bbd(3:10,:)=strg*bbd(3:10,:);

% Rate mode shapes
bbr(1,:)=-modes(274,NM);
bbr(2,:)=modes(275,NM);
bbr(3,:)=-modes(276,NM);
bbr(4,:)=modes(280,NM);
bbr(5,:)=modes(281,NM);
bbr(6,:)=-modes(282,NM);

% Acceleration mode shapes
bba(1,:)=modes(289,NM);
bba(2,:)=modes(290,NM);
bba(3,:)=modes(291,NM);
bba(4,:)=modes(283,NM);
bba(5,:)=modes(284,NM);
bba(6,:)=modes(285,NM);

% Actuator mode shapes
bbct(1,:)=-modes(238,NM)+modes(316,NM);
bbct(2,:)=-modes(258,NM)+modes(324,NM);
bbct1(3,:)=-modes(209,NM)+modes(299,NM);
bbct1(4,:)=-modes(216,NM)+modes(306,NM);
bbct1(5,:)=-modes(222,NM)+modes(312,NM);
bbct(3:5,:)=RM'*bbct1(3:5,:);

if Md>0;
  bd=zeros(Nm,1);
  for i=1:Md;
    id=indd(i);
    bd=[bd; bbd(id,:)'];
  end
  bd=bd(Nm+1:Mdm+Nm);
end
if Mr>0;
  br=zeros(Nm,1);
  for i=1:Mr;
    ir=indr(i);
    br=[br; bbr(ir,:)'];
  end
  br=br(Nm+1:Mrm+Nm);
end
if Ma>0;
```

```
  ba=zeros(Nm,1);
  for i=1:Ma;
    ia=inda(i);
    ba=[ba; bba(ia,:)'];
  end
  ba=ba(Nm+1:Mam+Nm);
end
bct=zeros(Nm,1);
for i=1:Mct;
  ic=ict(i);
  bct=[bct; bbct(ic,:)'];
end;
bct=bct(Nm+1:Mctm+Nm);

function [w,z,bd,br,ba,bct]=fdm2(evalfem,NM,indd,indr,inda,ict)
%-------------------------------------------------------------------------------
%
%  This function selects the proper degrees of freedom for the sensors
%  and actuators from the FEM  modeshapes for the 2 payload case (DM/EM)
%
%  The inputs are: 'evalfem'(String variable to load FEM-file)
%                  NM       (Number of modes, any modes from given ones)
%                  indd     (Numbers of displacement sensors)
%                           1    - Inner encoder on PL 2 (DM) (x)
%                           2    - Outer encoder on PL 2 (DM) (z)
%                           3-10 - Strain Gauges (vertical and horizontal)
%                                  in struts 1-4
%                           11   - Inner encoder on PL 1 (EM) (x)
%                           12   - Outer encoder on PL 1 (EM) (z)
%                  indr     (Numbers of rate sensors)
%                           1-3  - Rate gyros on PL 1 (EM) (x, y, z axis)
%                           4-6  - Rate gyros on TWA (x, y, z axis)
%                  inda     (Numbers of acceleration sensors)
%                           1-3  - Accelerometer in the node 2 (x, y, z axis)
%                           4-6  - Accelerometer in the node 4 (x, y, z axis)
%                  ict      (Numbers of actuators)
%                           1    - Inner gimbal on PL 2 (DM) (x)
%                           2    - Outer gimbal on PL 2 (DM) (z)
%                           3-5  - TWA (x, y, z axis)
%                           6    - Inner gimbal on PL 1 (EM) (x)
%                           7    - Outer gimbal on PL 1 (EM) (z)
%                           8    - Active strut vertical
%                           9    - Active strut horizontal
%
%  The outputs are: w       (Vector of frequencies)
%                   z       (Vector of damping ratios)
%                   bd      (Vector of mode shapes in displacement sensors)
%                   br      (Vector of mode shapes in rate sensors)
%                   ba      (Vector of mode shapes in acceleration sensors)
%                   bct     (Vector of mode shapes in actuators)
%
%  Call:            [w,z,bd,br,ba,bct]=fdm2(evalfem,NM,indd,indr,inda,ict)
%-------------------------------------------------------------------------------
```

```
eval(evalfem)

if NM==0
  Nm=length(freq);
  NM=1:Nm;
end

[Md,Mr,Ma,Mct]=nums(indd,indr,inda,ict);
Nm=length(NM);
Mdm=Md*Nm;
Mrm=Mr*Nm;
Mam=Ma*Nm;
Mctm=Mct*Nm;
rad=.0127;
glength=.00635;
strg=2*rad/glength;
RM=rotmat;            % Rotation matrix in TWA

w=freq(NM);
z=zeta(NM);

%  Displacement mode shapes
bbd(1,:)=modes(238,NM)-modes(316,NM);
bbd(2,:)=-modes(258,NM)+modes(324,NM);
bbd(3,:)=modes(114,NM)-modes(348,NM);
bbd(4,:)=modes(113,NM)-modes(347,NM);
bbd(5,:)=modes(84,NM)-modes(342,NM);
bbd(6,:)=modes(83,NM)-modes(341,NM);
bbd(7,:)=modes(336,NM)-modes(54,NM);
bbd(8,:)=modes(53,NM)-modes(335,NM);
bbd(9,:)=modes(24,NM)-modes(330,NM);
bbd(10,:)=modes(23,NM)-modes(329,NM);
bbd(11,:)=modes(388,NM)-modes(382,NM);
bbd(12,:)=modes(408,NM)-modes(402,NM);
bbd(3:10,:)=strg*bbd(3:10,:);

%  Rate mode shapes
bbr(1,:)=-modes(418,NM);
bbr(2,:)=modes(419,NM);
bbr(3,:)=-modes(420,NM);
bbr(4,:)=modes(280,NM);
bbr(5,:)=modes(281,NM);
bbr(6,:)=-modes(282,NM);

%  Acceleration mode shapes
bba(1,:)=modes(289,NM);
bba(2,:)=modes(290,NM);
bba(3,:)=modes(291,NM);
bba(4,:)=modes(283,NM);
bba(5,:)=modes(284,NM);
bba(6,:)=modes(285,NM);

%  Actuator mode shapes
bbct(1,:)=-modes(238,NM)+modes(316,NM);
```

```
bbct(2,:)=-modes(258,NM)+modes(324,NM);
bbct1(3,:)=-modes(209,NM)+modes(299,NM);
bbct1(4,:)=-modes(216,NM)+modes(306,NM);
bbct1(5,:)=-modes(222,NM)+modes(312,NM);
bbct(6,:)=modes(382,NM)-modes(388,NM);
bbct(7,:)=modes(402,NM)-modes(408,NM);
bbct(8,:)=8.988*(modes(48,NM)-modes(60,NM))+148.492*(modes(54,NM)-modes(336,NM));
bbct(9,:)=8.988*(modes(59,NM)-modes(47,NM))+148.492*(modes(335,NM)-modes(53,NM));
bbct(3:5,:)=RM'*bbct1(3:5,:);

if Md>0;
  bd=zeros(Nm,1);
  for i=1:Md;
    id=indd(i);
    bd=[bd; bbd(id,:)'];
  end
  bd=bd(Nm+1:Mdm+Nm);
end
if Mr>0;
  br=zeros(Nm,1);
  for i=1:Mr;
    ir=indr(i);
    br=[br; bbr(ir,:)'];
  end
  br=br(Nm+1:Mrm+Nm);
end
if Ma>0;
  ba=zeros(Nm,1);
  for i=1:Ma;
    ia=inda(i);
    ba=[ba; bba(ia,:)'];
  end
  ba=ba(Nm+1:Mam+Nm);
end
bct=zeros(Nm,1);
for i=1:Mct;
  ic=ict(i);
  bct=[bct; bbct(ic,:)'];
end;
bct=bct(Nm+1:Mctm+Nm);

function [w,z,bd,br,ba,bct]=fdm3(evalfem,NM,indd,indr,inda,ict)
%-------------------------------------------------------------------------------
%
%  This function selects the proper degrees of freedom for the sensors
%  and actuators from the FEM  modeshapes for the 2 payload case (EM)
%
%  The inputs are: 'evalfem'(String variable to load FEM-file)
%                  NM       (Number of modes, any modes from given ones)
%                  indd     (Numbers of displacement sensors)
%                           1    - Inner Primary encoder (x)
%                           2    - Outer Primary encoder (z)
%                           3    - Inner Secondary encoder (x)
%                           4    - Outer Secondary encoder (z)
```

```
%                          5    - Strain Gage strut 1 (y)
%                          6    - Strain Gage strut 1 (z)
%                          7    - Strain Gage strut 2 (y)
%                          8    - Strain Gage strut 2 (z)
%                          9    - Strain Gage strut 3 (y)
%                         10    - Strain Gage strut 3 (z)
%                         11    - Strain Gage strut 4 (y)
%                         12    - Strain Gage strut 4 (z)
%                  indr    (Numbers of rate sensors)
%                          1    - Inner Primary encoder (x)
%                          2    - Outer Primary encoder (z)
%                          3    - Inner Secondary encoder (x)
%                          4    - Outer Secondary encoder (z)
%                          5,6  - Payload Rate gyros (x, z axis)
%                          7-9  - Bus Rate gyros (x, y, z axis)
%                          10-12  - RWA Tachometers (x, y, z axis)
%                          13-14 - Secondary Rate gyros (x,z axis)
%                  inda    (There are NO accelerometers)
%                  ict     (Numbers of actuators)
%                          1    - Inner Primary gimbal (x)
%                          2    - Outer Primary gimbal (z)
%                          3    - Inner Secondary gimbal (x)
%                          4    - Outer Secondary gimbal (z)
%                          5-7  - RWA (x, y, z)
%                          8    - Active strut (y)
%                          9    - Active strut (z)
%
%  The outputs are: w      (Vector of frequencies)
%                   z      (Vector of damping ratios)
%                   bd     (Vector of mode shapes in displacement sensors)
%                   br     (Vector of mode shapes in rate sensors)
%                   ba     (Vector of mode shapes in acceleration sensors)
%                   bct    (Vector of mode shapes in actuators)
%
%  Call:             [w,z,bd,br,ba,bct]=fdm3(evalfem,NM,indd,indr,inda,ict)
%-------------------------------------------------------------------------

eval(evalfem)

if NM==0
  Nm=length(freq);
  NM=1:Nm;
end

[Md,Mr,Ma,Mct]=nums(indd,indr,inda,ict);
Nm=length(NM);
Mdm=Md*Nm;
Mrm=Mr*Nm;
Mam=Ma*Nm;
Mctm=Mct*Nm;
rad1=.0144145;
rad2=.0147955;
glength=.003175;
strg1=2*rad1/glength;  % passive strut strain gage gain
```

```
strg2=2*rad2/glength;  % active strut strain gage gain
las=.0889;
as=2*rad2/las;         % active strut gain
RM=rotmat2;            % Rotation matrix in RWA

w=freq(NM);
z=zeta(NM);

%  Displacement mode shapes
bbd(1,:)= modes(220,NM)-modes(226,NM);
bbd(2,:)= modes(240,NM)-modes(246,NM);
bbd(3,:)=-modes(268,NM)+modes(274,NM);
bbd(4,:)=-modes(288,NM)+modes(294,NM);
bbd(5,:)=strg1*(modes(35,NM) -modes(29,NM));
bbd(6,:)=strg1*(-modes(36,NM) +modes(30,NM));
bbd(7,:)=strg2*(-modes(77,NM) +modes(71,NM));
bbd(8,:)=strg2*( modes(78,NM) -modes(72,NM));
bbd(9,:)=strg1*( modes(119,NM)-modes(113,NM));
bbd(10,:)=strg1*( modes(120,NM)-modes(114,NM));
bbd(11,:)=strg1*(-modes(161,NM)+modes(155,NM));
bbd(12,:)=strg1*(-modes(162,NM)+modes(156,NM));

%  Rate mode shapes
bbr(1,:)= modes(220,NM)-modes(226,NM);
bbr(2,:)= modes(240,NM)-modes(246,NM);
bbr(3,:)=-modes(268,NM)+modes(274,NM);
bbr(4,:)=-modes(288,NM)+modes(294,NM);
bbr(5,:)= modes(256,NM);
bbr(6,:)=-modes(258,NM);
bbr(7,:)= modes(208,NM);
bbr(8,:)=-modes(209,NM);
bbr(9,:)=-modes(210,NM);
bbr(10,:)=modes(324,NM)-modes(318,NM);
bbr(11,:)=modes(342,NM)-modes(336,NM);
bbr(12,:)=modes(360,NM)-modes(354,NM);
bbr(10:12,:)=RM'*bbr(10:12,:);
bbr(13,:)=modes(304,NM);
bbr(14,:)=modes(306,NM);

%  Actuator mode shapes
bbct(1,:)=-modes(220,NM)+modes(226,NM);
bbct(2,:)=-modes(240,NM)+modes(246,NM);
bbct(3,:)= modes(268,NM)-modes(274,NM);
bbct(4,:)=-modes(288,NM)+modes(294,NM);
bbct(5,:)=modes(324,NM)-modes(318,NM);
bbct(6,:)=modes(342,NM)-modes(336,NM);
bbct(7,:)=modes(360,NM)-modes(354,NM);
bbct(8,:)=as*((modes(599,NM)-modes(587,NM))+(modes(617,NM)-modes(605,NM)));
bbct(9,:)=as*((modes(600,NM)-modes(588,NM))+(modes(618,NM)-modes(606,NM)));
bbct(5:7,:)=RM'*bbct(5:7,:);

if Md>0;
  bd=zeros(Nm,1);
  for i=1:Md;
```

```
    id=indd(i);
    bd=[bd; bbd(id,:)'];
  end
  bd=bd(Nm+1:Mdm+Nm);
end
if Mr>0;
  br=zeros(Nm,1);
  for i=1:Mr;
    ir=indr(i);
    br=[br; bbr(ir,:)'];
  end
  br=br(Nm+1:Mrm+Nm);
end
bct=zeros(Nm,1);
for i=1:Mct;
  ic=ict(i);
  bct=[bct; bbct(ic,:)'];
end;
bct=bct(Nm+1:Mctm+Nm);

  bd=zeros(Nm,1);
  for i=1:Md;
    id=indd(i);
    bd=[bd; bbd(id,:)'];
  end
  bd=bd(Nm+1:Mdm+Nm);
end
if Mr>0;
  br=zeros(Nm,1);
  for i=1:Mr;
    ir=indr(i);
    br=[br; bbr(ir,:)'];
  end
  br=br(Nm+1:Mrm+Nm);
end
bct=zeros(Nm,1);
for i=1:Mct;
  ic=ict(i);
  bct=[bct; bbct(ic,:)'];
end;
bct=bct(Nm+1:Mctm+Nm);

function [Gd,Gr,Ga,Gct]=gains1(volts,ipur)
%-------------------------------------------------------------------------
%
%  This function stores gains G for outputs (and inputs):
%  [AMPLIFIED OUTPUT] = G * [OUTPUT IN REAL VOLTS]
%
%  This is for the 1 payload case
%
%  The inputs are:  volts (==1 - Volts; ==0 - Physical units)
%                   ipur  (Reflects the purpose: ==0 - for y; ==1 - for B,C,D)
%
%  The outputs are: Gains G for sensors and actuators
```

```
%
%  Call:            [Gd,Gr,Ga,Gct]=gains1(volts,ipur)
%-------------------------------------------------------------------------

if ((volts==0 & ipur==0) | (volts==1 & ipur==1))
% Encoder gains
    Gd(1)=1;             %  Inner on PL (DM) (x)
    Gd(2)=1;             %  Outer on PL (DM) (z)
% Strain gauge gains (Vertical and horizontal in struts 1-4 )
    Gd(3)=1;             %  V (1)
    Gd(4)=1;             %  H (1)
    Gd(5)=1;             %  V (2)
    Gd(6)=1;             %  H (2)
    Gd(7)=1;             %  V (3)
    Gd(8)=1;             %  H (3)
    Gd(9)=1;             %  V (4)
    Gd(10)=1;            %  H (4)
% Rate gyro gains
    Gr(1)=1;             %  On PL (DM) (x)
    Gr(2)=1;             %  On PL (DM) (y)
    Gr(3)=1;             %  On PL (DM) (z)
    Gr(4)=1;             %  On TWA (x)
    Gr(5)=1;             %  On TWA (y)
    Gr(6)=1;             %  On TWA (z)
% Accelerometer gains
    Ga(1)=1;             %  In node 2 (x)
    Ga(2)=1;             %  In node 2 (y)
    Ga(3)=1;             %  In node 2 (z)
    Ga(4)=1;             %  In node 4 (x)
    Ga(5)=1;             %  In node 4 (y)
    Ga(6)=1;             %  In node 4 (z)
% Actuator gains
    Gct(1)=1;            %  Inner gimbal on PL 2 (DM) (x)
    Gct(2)=1;            %  Outer gimbal on PL 2 (DM) (z)
    Gct(3)=1;            %  TWA (x)
    Gct(4)=1;            %  TWA (y)
    Gct(5)=1;            %  TWA (z)
end

function [Gd,Gr,Ga,Gct]=gains2(volts,ipur)
%-------------------------------------------------------------------------
%
%  This function stores gains G for outputs (and inputs):
%  [AMPLIFIED OUTPUT] = G * [OUTPUT IN REAL VOLTS]
%
%  This is for the 2 payload case
%
%  The inputs are:  volts (==1 - Volts; ==0 - Physical units)
%                   ipur  (Reflects the purpose: ==0 - for y; ==1 - for B,C,D)
%
%  The outputs are: Gains G for sensors and actuators
%
%  Call:            [Gd,Gr,Ga,Gct]=gains2(volts,ipur)
%-------------------------------------------------------------------------
```

```
if ((volts==0 & ipur==0) | (volts==1 & ipur==1))
% Encoder gains
    Gd(1)=1;              %  Inner on PL 2 (DM) (x)
    Gd(2)=1;              %  Outer on PL 2 (DM) (z)
    Gd(11)=1;             %  Inner on PL 1 (EM) (x)
    Gd(12)=1;             %  Outer on PL 1 (EM) (z)
% Strain gauge gains (Vertical and horizontal in struts 1-4 )
    Gd(3)=1;              %  V (1)
    Gd(4)=1;              %  H (1)
    Gd(5)=1;              %  V (2)
    Gd(6)=1;              %  H (2)
    Gd(7)=1;              %  V (3)
    Gd(8)=1;              %  H (3)
    Gd(9)=1;              %  V (4)
    Gd(10)=1;             %  H (4)
% Rate gyro gains
    Gr(1)=1;              %  On PL 1 (EM) (x)
    Gr(2)=1;              %  On PL 1 (EM) (y)
    Gr(3)=1;              %  On PL 1 (EM) (z)
    Gr(4)=1;              %  On TWA (x)
    Gr(5)=1;              %  On TWA (y)
    Gr(6)=1;              %  On TWA (z)
% Accelerometer gains
    Ga(1)=1;              %  In node 2 (x)
    Ga(2)=1;              %  In node 2 (y)
    Ga(3)=1;              %  In node 2 (z)
    Ga(4)=1;              %  In node 4 (x)
    Ga(5)=1;              %  In node 4 (y)
    Ga(6)=1;              %  In node 4 (z)
% Actuator gains
    Gct(1)=1;             %  Inner gimbal on PL 2 (DM) (x)
    Gct(2)=1;             %  Outer gimbal on PL 2 (DM) (z)
    Gct(3)=1;             %  TWA (x)
    Gct(4)=1;             %  TWA (y)
    Gct(5)=1;             %  TWA (z)
    Gct(6)=1;             %  Inner gimbal on PL 1 (EM) (x)
    Gct(7)=1;             %  Outer gimbal on PL 1 (EM) (z)
    Gct(8)=1;             %  Active strut vertical
    Gct(9)=1;             %  Active strut horizontal
end

function [Gd,Gr,Ga,Gct]=gains3(volts,ipur)
%------------------------------------------------------------------------
%
%  This function stores gains G for outputs (and inputs):
%  [AMPLIFIED OUTPUT] = G * [OUTPUT IN REAL VOLTS]
%
%  This is for the 2 payload case
%
%  The inputs are:  volts (==1 - Volts; ==0 - Physical units)
%                   ipur  (Reflects the purpose: ==0 - for y; ==1 - for B,C,D)
%
%  The outputs are: Gains G for sensors and actuators
```

```
%
%  Call:            [Gd,Gr,Ga,Gct]=gains3(volts,ipur)
%-------------------------------------------------------------------------

if ((volts==0 & ipur==0) | (volts==1 & ipur==1))
% Encoder gains
    Gd(1)=1;             %  Inner Primary (x)
    Gd(2)=1;             %  Outer Primary (z)
    Gd(3)=1;             %  Inner Secondary (x)
    Gd(4)=1;             %  Outer Secondary (z)
% Strain gauge gains (Vertical and horizontal in struts 1-4 )
    Gd(5)= 1;            %  H (1)
    Gd(6)= 1;            %  V (1)
    Gd(7)= 1;            %  H (2)
    Gd(8)= 1;            %  V (2)
    Gd(9)= 1;            %  H (3)
    Gd(10)=1;            %  V (3)
    Gd(11)=1;            %  H (4)
    Gd(12)=1;            %  V (4)
% Rate gains
    Gr(1)=1;             %  Inner Primary (x)
    Gr(2)=1;             %  Outer Primary (z)
    Gr(3)=1;             %  Inner Secondary (x)
    Gr(4)=1;             %  Outer Secondary (z)
    Gr(5)=1;             %  On Payload (x)
    Gr(6)=1;             %  On Payload (z)
    Gr(7)=1;             %  On Bus (x)
    Gr(8)=1;             %  On Bus (y)
    Gr(9)=1;             %  On Bus (z)
    Gr(10)=1;            %  On RWA Tach x
    Gr(11)=1;            %  On RWA Tach y
    Gr(12)=1;            %  On RWA Tach z
% Accelerometer gains

% Actuator gains
    Gct(1)=1;            %  Inner Primary gimbal (x)
    Gct(2)=1;            %  Outer Primary gimbal (z)
    Gct(3)=1;            %  Inner Secondary gimbal (x)
    Gct(4)=1;            %  Outer Secondary gimbal (z)
    Gct(5)=1;            %  RWA (x)
    Gct(6)=1;            %  RWA (y)
    Gct(7)=1;            %  RWA (z)
    Gct(8)=1;            %  Active strut (y)
    Gct(9)=1;            %  Active strut (z)
end

function [Acl,Bcl,Ccl,Dcl]=...
    gimcon(A,B,C,D,NM,n0,indd,ict,dlk,NP,evalfem,volts,config)
%-------------------------------------------------------------------------
%
%  This function closes the servo loops around the gimbals
%
%  The inputs are:  A       (A matrix containing sensor dynamics & time delays)
%                   B       (B matrix containing sensor dynamics & time delays)
```

```
%                   C       (C matrix containing sensor dynamics & time delays)
%                   D       (D matrix containing sensor dynamics & time delays)
%                   NM      (Numbers of modes)
%                   n0      (number of rigid body modes)
%                   indd    (Numbers of displacement sensors)
%                   ict     (Numbers of actuators)
%                   dlk     (Vector of servo loop time delays in gimbals)
%                   NP      (Order of Pade approximation)
%                   evalfem (String variable to load FEM-file)
%                   volts   (==1 - Volts, ==0 - Physical units)
%                   config  (==2 - DM/EM)
%
%  The outputs are: Acl,Bcl,Ccl,Dcl    (Closed loop system matrices)
%
%  Call:   [Acl,Bcl,Ccl,Dcl]=...
%             gimcon(A,B,C,D,NM,n0,indd,ict,dlk,NP,evalfem,volts,config)
%-------------------------------------------------------------------------

[n,na]=size(B);
[ny,n]=size(C);
Nm=length(NM)-n0;

flagd=0;

if (all(indd~=1))
  indd1=[indd1 1];
  flagd=flagd+1;
  dmie=ny+flagd;
else
  dmie=find(indd==1);
end

if (all(indd~=2))
  indd1=[indd1 2];
  flagd=flagd+1;
  dmoe=ny+flagd;
else
  dmoe=find(indd==2);
end

if (all(indd~=11))
  indd1=[indd1 11];
  flagd=flagd+1;
  emie=ny+flagd;
else
  emie=find(indd==11);
end

if (all(indd~=12))
  indd1=[indd1 12];
  flagd=flagd+1;
  emoe=ny+flagd;
else
  emoe=find(indd==12);
```

```
end

flaga=0;

if (all(ict~=1))
  ict1=[ict1 1];
  flaga=flaga+1;
  dmig=na+flaga;
else
  dmig=find(ict==1);
end

if (all(ict~=2))
  ict1=[ict1 2];
  flaga=flaga+1;
  dmog=na+flaga;
else
  dmog=find(ict==2);
end

if (all(ict~=6))
  ict1=[ict1 6];
  flaga=flaga+1;
  emig=na+flaga;
else
  emig=find(ict==6);
end

if (all(ict~=7))
  ict1=[ict1 7];
  flaga=flaga+1;
  emog=na+flaga;
else
 emog=find(ict==7);
end

if (flagd>0 | flaga>0)
  [w,z,bd,br,ba,bct]=fdm2(evalfem,NM,indd1,[],[],ict1);
  [Md,Mr,Ma,Mct]=nums(indd1,[],[],ict1);
  [Cdx,Crv,Cax,Cav,Ba]=cm1(length(NM),Md,Mr,Ma,Mct,w,z,bd,br,ba,bct);
  Br=[zeros(n0,Mct);Ba(1:n0,:)];
  Bf=[zeros(Nm,Mct); Ba(n0+1:Nm+n0,:)];
  B1=[Br;Bf];
  if Md>0;
    Cdr=[Cdx(:,1:n0) zeros(Md,n0)];
    Cdf=[Cdx(:,n0+1:n0+Nm) zeros(Md,Nm)];
    Cd=[Cdr Cdf];
    Dd=zeros(Md,Mct);
  end
  C1=assemy(Md,0,0,Mct,Cd',[],[])';
  D1=assemy(Md,0,0,Mct,Dd',[],[])';
  if volts==1;
    [B1,C1,D1]=resabcd(indd1,[],[],ict1,B1,C1,D1,config,2*Nm);
  end
```

```
  B1=[B [B1;zeros(n-2*(Nm+n0),flaga)]];
  C1=[C;C1 zeros(flagd,n-2*(Nm+n0))];
  D1=[D zeros(ny,flaga);zeros(flagd,na) D1];
else
  B1=B;
  C1=C;
  D1=D;
end
Cr=C1([dmie dmoe emie emoe],1:n0)
Cf=C1([dmie dmoe emie emoe],n0+1:Nm+n0)
C1(ny+1+flagd:ny+4+flagd,:)=[zeros(4,n0) Cf zeros(4,Nm) Cf zeros(4,n-2*Nm)];
D1(ny+1+flagd:ny+4+flagd,:)=zeros(4,na+flaga);

[K,sig,gain]=gpd(volts,config);

F=zeros(4,8);
F(3,3)=K(1,1);
F(4,4)=K(2,1);
F(1,1)=K(3,1);
F(2,2)=K(4,1);
F(3,7)=K(1,2);
F(4,8)=K(2,2);
F(1,5)=K(3,2);
F(2,6)=K(4,2);
F=F*gain;

[Atd,Btd,Ctd,Dtd,in,out]=delfid([1 2 3 4],NP,dlk);

inp=sig*[dmig dmog emig emog];
out=[dmie dmoe emie emoe (ny+flagd)+[1 2 3 4]];
[Acl,Bcl,Ccl,Dcl]=feedback(A,B1,C1,D1,Atd,Btd*F,Ctd,Dtd*F,inp,out);

Bcl(:,[emig emog dmig dmog])=sig.*Bcl(:,[emig emog dmig dmog])*diag(K(:,1));
Bcl(:,[dmig emig emog])=-Bcl(:,[dmig emig emog]);
Ccl([dmie dmoe emie emoe],:)=Ccl([dmie dmoe emie emoe],:)*gain;
Bcl=Bcl(:,1:na);
Ccl=Ccl(1:ny,:);
Dcl=Dcl(1:ny,1:na);

function [Acl,Bcl,Ccl,Dcl]=...
        gimcon3(A,B,C,D,NM,n0,indd,indr,ict,dlk,NP,evalfem,volts,config)
%-------------------------------------------------------------------------
%
%  This function closes the servo loops around the gimbals and reaction wheels
%
%  The inputs are:  A       (A matrix containing time delays)
%                   B       (B matrix containing time delays)
%                   C       (C matrix containing time delays)
%                   D       (D matrix containing time delays)
%                   NM      (Numbers of modes)
%                   n0      (number of rigid body modes)
%                   indd    (Numbers of displacement sensors)
%                   indr    (Numbers of rate sensors)
%                   ict     (Numbers of actuators)
```

```
%                   dlk     (vector of computer time delays for gimbal servos)
%                   NP      (Order of Pade approximation)
%                   evalfem (String variable to load FEM-file)
%                   volts   (==1 - Volts, ==0 - Physical units)
%                   config  (==3 - EM)
%
%  The outputs are: Acl,Bcl,Ccl,Dcl    (Closed loop system matrices)
%
%  Call:   [Acl,Bcl,Ccl,Dcl]=...
%       gimcon3(A,B,C,D,NM,n0,indd,indr,ict,dlk,NP,evalfem,volts,config)
%-------------------------------------------------------------------------------

[n,na]=size(B);
[ny,n]=size(C);
Nm=length(NM)-n0;

%  Find all the required displacement sensors

flagd=0;

if (all(indd~=1))
  indd1=[indd1 1];
  flagd=flagd+1;
  pied=ny+flagd;
else
  pied=find(indd==1);
end

if (all(indd~=2))
  indd1=[indd1 2];
  flagd=flagd+1;
  poed=ny+flagd;
else
  poed=find(indd==2);
end

if (all(indd~=3))
  indd1=[indd1 3];
  flagd=flagd+1;
  sied=ny+flagd;
else
  sied=find(indd==3);
end

if (all(indd~=4))
  indd1=[indd1 4];
  flagd=flagd+1;
  soed=ny+flagd;
else
  soed=find(indd==4);
end

%  Find all the required rate sensors
```

```
flagr=0;

if (all(indr~=1))
  indr1=[indr1 1];
  flagr=flagr+1;
  pier=ny+flagr+flagd;
else
  pier=find(indr==1)+length(indd);
end

if (all(indr~=2))
  indr1=[indr1 2];
  flagr=flagr+1;
  poer=ny+flagr+flagd;
else
  poer=find(indr==2)+length(indd);
end

if (all(indr~=3))
  indr1=[indr1 3];
  flagr=flagr+1;
  sier=ny+flagr+flagd;
else
  sier=find(indr==3)+length(indd);
end

if (all(indr~=4))
  indr1=[indr1 4];
  flagr=flagr+1;
  soer=ny+flagr+flagd;
else
  soer=find(indr==4)+length(indd);
end

if (all(indr~=10))
  indr1=[indr1 10];
  flagr=flagr+1;
  tachx=ny+flagr+flagd;
else
  tachx=find(indr==10)+length(indd);
end

if (all(indr~=11))
  indr1=[indr1 11];
  flagr=flagr+1;
  tachy=ny+flagr+flagd;
else
  tachy=find(indr==11)+length(indd);
end

if (all(indr~=12))
  indr1=[indr1 12];
  flagr=flagr+1;
  tachz=ny+flagr+flagd;
```

```
else
  tachz=find(indr==12)+length(indd);
end

%  Find all the required actuators

flaga=0;

if (all(ict~=1))
  ict1=[ict1 1];
  flaga=flaga+1;
  pig=na+flaga;
else
  pig=find(ict==1);
end

if (all(ict~=2))
  ict1=[ict1 2];
  flaga=flaga+1;
  pog=na+flaga;
else
  pog=find(ict==2);
end

if (all(ict~=3))
  ict1=[ict1 3];
  flaga=flaga+1;
  sig=na+flaga;
else
  sig=find(ict==3);
end

if (all(ict~=4))
  ict1=[ict1 4];
  flaga=flaga+1;
  sog=na+flaga;
else
  sog=find(ict==4);
end

if (all(ict~=5))
  ict1=[ict1 5];
  flaga=flaga+1;
  rwax=na+flaga;
else
  rwax=find(ict==5);
end

if (all(ict~=6))
  ict1=[ict1 6];
  flaga=flaga+1;
  rway=na+flaga;
else
  rway=find(ict==6);
```

```
end

if (all(ict~=7))
  ict1=[ict1 7];
  flaga=flaga+1;
  rwaz=na+flaga;
else
  rwaz=find(ict==7);
end

%  Form the augmented system

if (flagd>0 | flaga>0 | flagr>0)
  if config==3
    [w,z,bd,br,ba,bct]=fdm3(evalfem,NM,indd1,indr1,[],ict1);
  end
  [Md,Mr,Ma,Mct]=nums(indd1,indr1,[],ict1);
  [Cdx,Crv,Cax,Cav,Ba]=cm1(length(NM),Md,Mr,Ma,Mct,w,z,bd,br,ba,bct);
  if flaga>0
    Br=[zeros(n0,Mct);Ba(1:n0,1:Mct)];
    Bf=[zeros(Nm,Mct); Ba(n0+1:Nm+n0,:)];
    B1=[Br;Bf];
  elseif (flagd>0 | flagr>0);
    if Md>0;
      Cdr=[Cdx(:,1:n0) zeros(Md,n0)];
      Cdf=[Cdx(:,n0+1:n0+Nm) zeros(Md,Nm)];
      Cd=[Cdr Cdf];
      Dd=zeros(Md,Mct);
    end
    if Mr>0
      Crr=[zeros(Mr,n0) Crv(:,1:n0)];
      Crf=[zeros(Mr,Nm) Crv(:,n0+1:n0+Nm)];
      Cr=[Crr Crf];
      Dr=zeros(Mr,Mct);
    end
    C1=assemy(Md,Mr,0,Mct,Cd',Cr',[])';
    D1=assemy(Md,Mr,0,Mct,Dd',Dr',[])';
  end
  if volts==1;
    [B1,C1,D1]=resabcd(indd1,indr1,[],ict1,B1,C1,D1,config,2*length(NM));
  end
  B1=[B [B1;zeros(n-2*length(NM),flaga)]];
  C1=[C;C1 zeros(flagd+flagr,n-2*length(NM))];
  D1=[D zeros(ny,flaga);zeros(flagd+flagr,na) D1];
else
  B1=B;
  C1=C;
  D1=D;
end

%  Remove Uncontrollable Rigid Body Modes and Reaction Wheel angle states

[n,na1]=size(B1);
[ny1,n]=size(C1);
```

```
if (n0==0) %  Do it interactively (Put this in later if necessary)
elseif (n0==3)
  A=A(4:n,4:n);
  B1=B1(4:n,:);
  C1=C1(:,4:n);
elseif (n0==13)
  Ar=[zeros(10,10) eye(10);zeros(10,20)];
  Af=A(27:n,27:n);
  A=[Ar zeros(20,length(Af));zeros(length(Af),20) Af];
  if config==3
    [w,z,bd,br,ba,bct]=fdm3(evalfem,[1:13],[],[1:14],[],[]);
  end
  [Cdx,Crv,Cax,Cav,Ba]=cm1(13,0,14,0,0,zeros(13,1),zeros(13,1),[],br,[],[]);
  [u,s,v]=svd(Crv');
  br=inv(u)*B1(14:26,:);crd=C1(:,1:13)*inv(u');crr=C1(:,14:26)*inv(u');
  B1=[zeros(10,na1);br(1:10,:);B1(27:n,:)];
  C1=[crd(:,1:10) crr(:,1:10) C1(:,27:n)];
  [n,na1]=size(B1);
  [ny1,n]=size(C1);
%  A=A(4:n,4:n);
%  B1=B1(4:n,:);
%  C1=C1(:,4:n);
end

n=length(B1);

gimb=[pig pog sig sog];
rwa=[rwax rway rwaz];
encd=[pied poed sied soed];
encr=[pier poer sier soer];
tach=[tachx tachy tachz];

%  Put encoders in degrees
C1([encd encr],:)=180/pi*C1([encd encr],:);
%  Get gains
[K,sig,gain]=gpd(volts,config);

%  Close Reaction Wheel speed control servos
B1(:,rwa)=B1(:,rwa)*K(5,1);
for i=1:3
  A=A-K(5,2)*(B1(:,rwa(i))*C1(tach(i),:));
end
B1(:,rwa(3))=B1(:,rwa(3))/1.5;

%  Put computational delay (+ zero-order hold) in gimbals
for i=1:4
  [At,Bt,Ct,Dt]=delfid(1,NP,dlk(i));
  A=[A B1(:,gimb(i))*Ct;zeros(NP,n) At];
  B1=[B1;zeros(NP,na+flaga)];
  B1(:,gimb(i))=[B1(1:n,gimb(i))*Dt;Bt];
  C1=[C1 D1(:,gimb(i))*Ct];
  D1(:,gimb(i))=D1(:,gimb(i))*Dt;
  n=n+NP;
end
```

```
%  Put computational delay + zero-order hold in RWA
for i=1:3
  [At,Bt,Ct,Dt]=delfid(1,NP,dlk(i+4));
  A=[A B1(:,rwa(i))*Ct;zeros(NP,n) At];
  B1=[B1;zeros(NP,na+flaga)];
  B1(:,rwa(i))=[B1(1:n,rwa(i))*Dt;Bt];
  C1=[C1 D1(:,rwa(i))*Ct];
  D1(:,rwa(i))=D1(:,rwa(i))*Dt;
  n=n+NP;
end

%  Close gimbal servo loops
g1=K(1:4,1)*gain;
g2=K(1:4,2)*gain;
for i=1:4
  A=A-B1(:,gimb(i))*(g1(i)*C1(encd(i),:)+g2(i)*C1(encr(i),:));
  B1(:,gimb(i))=B1(:,gimb(i))*g1(i)/1.5;
end

Acl=A;
Bcl=B1(:,1:na);
Ccl=C1(1:ny,:);
Dcl=D1(1:ny,1:na);

function [KPD,sig,gain]=gpd(volts,config)
%-------------------------------------------------------------------------
%
%  This function stores gains K for PD-controller:
%  (First column - angles, Second column - rates)
%
%  The inputs are:  volts (==1 - Volts,==0 - Physical units)
%                   config(==2 - DM/EM,==3 - EM)
%
%  The outputs are: KPD   (Gains in PD-controller)
%                   sig   (sign to use in feedback)
%                   gain  (gain to convert encoders to proper units)
%
%  Call:          [KPD,sig,gain]=gpd(volts,config)
%-------------------------------------------------------------------------

KPD=[-10     -0.2                 %  EM inner
     -18     -0.5                 %  EM outer
     -2.5    -0.1                 %  DM inner
      2.5     0.1];               %  DM outer
gain=7.1667;
sig=-1;
if (config==3)
  KPD=[-2.0   -0.1                %  Primary inner
       -2.0   -0.1                %  Primary outer
       -2.0   -0.1                %  Secondary inner
        2.0    0.1                %  Secondary outer
        4.3    1.0                %  RW speed control (forward  backward)
        4.3    1.0
```

```
       4.3    1.0];
  gain=1.0;
  sig=-1;
  end
  if volts==0
    if (config==2)
      [Kd,Kr,Ka,Kct]=units2(0,0);
      Kct=Kct([6 7 1 2]);
    elseif (config==3)
      [Kd,Kr,Ka,Kct]=units3(0,0);
      Kct=Kct([1 2 3 4 5 6 7])
    end
  Kct=[Kct Kct];
  KPD=KPD.*Kct;
end

function [Gs,Gct,Ks,Kct]=guv(volts,ipur,indd,indr,inda,ict,config)
%-------------------------------------------------------------------------------
%
%  This function provides vector of gains (G) and convertion factors (K)
%  for chosen configuration of sensor and actuators
%
%  The inputs are:  volts (==1 - Volts; ==0 - Physical units)
%                   ipur  (Reflects the purpose: ==0 - for y; ==1 - for B,C,D)
%                   indd  (Numbers of displacement sensors)
%                   indr  (Numbers of rate sensors)
%                   inda  (Numbers of acceleration sensors)
%                   ict   (Numbers of actuators)
%                   config(==1 - DM,==2 - DM/EM,==3 - EM)
%
%  The outputs are: G and K for sensors and actuators
%
%  Call:      [Gs,Gct,Ks,Kct]=guv(volts,ipur,indd,indr,inda,ict)
%-------------------------------------------------------------------------------

[Md,Mr,Ma,Mct]=nums(indd,indr,inda,ict);

if config==1
  [Gd,Gr,Ga,Gct]=gains1(volts,ipur);
  [Kd,Kr,Ka,Kct]=units1(volts,ipur);
elseif (config==2)
  [Gd,Gr,Ga,Gct]=gains2(volts,ipur);
  [Kd,Kr,Ka,Kct]=units2(volts,ipur);
elseif (config==3)
  [Gd,Gr,Ga,Gct]=gains3(volts,ipur);
  [Kd,Kr,Ka,Kct]=units3(volts,ipur);
end
if Md>0;
  Gdd=Gd(indd);
  Kdd=Kd(indd);
end;
if Mr>0;
  Grr=Gr(indr);
  Krr=Kr(indr);
```

```
end
if Ma>0;
   Gaa=Ga(inda);
   Kaa=Ka(inda);
 end
Gs=assemy(Md,Mr,Ma,Mct,Gdd,Grr,Gaa)';
Ks=assemy(Md,Mr,Ma,Mct,Kdd',Krr',Kaa')';
Gct=Gct(ict);
Kct=Kct(ict);

function [ain,bin,cin,din]=intfid(indr)
%-------------------------------------------------------------------------------
%
%  This function provides a model of the stabilized integrator
%  for all RG involved
%
%  The inputs are:  indr (Numbers of rate sensors (RS))
%
%  The outputs are: ain,bin,cin,din (Matrices in linear system)
%
%  Call:          [ain,bin,cin,din]=intfid(indr)
%-------------------------------------------------------------------------------

Mr=length(indr);
Mr2=2*Mr;
[a,b,c,d]=stabint;
ain=zeros(Mr2,Mr2);
bin=zeros(Mr2,Mr);
cin=zeros(Mr,Mr2);
din=zeros(Mr,Mr);
for i=1:Mr
  is=(i-1)*2+1;
  it=is+1;
  II=is:it;
  ain(II,II)=a;
  bin(II,i)=b;
  cin(i,II)=c;
  din(i,i)=d;
end

function [A,B,C,D]=...
modext(isensd,idelay,integ,ibess,A,B,C,D,indd,indr,NP,dls,cfb,config)
%-------------------------------------------------------------------------------
%
%  This function expands the linear model if sensor dynamics, time
%  delays, integrators, or Bessel filters are appended
%
%  The inputs are:  isensd   (==1 - sensor dynamics, ==0 - no sensor dynamics)
%                   idlsen   (==1 - time delays, ==0 - no time delays)
%                   integ    (==0 - Stabilized integrators are off
%                             ==1 - stabilized integrators are on,
%                                   but dynamics are not accounted for
%                             ==2 - stabilized integrators are on,
%                                   dynamics are included in A,B,C and D)
```

```
%                   ibess    (==1 - Bessel Filters, ==0 - no Bessel Filters)
%                   A,B,C,D  (linear model of structural dynamics)
%                   indd     (Numbers of displacement sensors)
%                   indr     (Numbers of rate sensors)
%                   NP       (Order of Pade approximation)
%                   dls      (Vector of time delays in each sensor)
%                   cfb      (Vector of Bessel Filter corner frequencies)
%                   config   (==1-DM,==2-DM/EM,==3-EM)
%
%  The outputs are: A,B,C,D  (Expanded linear model)
%
%  Call:  function [A,B,C,D]=...
%      modext(isensd,idelayl,integ,ibess,A,B,C,D,indd,indr,NP,dls,config)
%-------------------------------------------------------------------------

[Md,Mr]=nums(indd,indr,[],[]);
if (config==3)
  k=find(indd>4);sens=k;dis=k;
  k=find(indr>4 & indr<10);sens=[sens k+Md];rate=k;
else
  dis=find(indd>4);Ne=Md-length(dis);
  sens=[sens indr+Md];Ner=Mr-length(indr);rate=indr;
end
if (isensd|idelay|integ|ibess)
  if isensd==1
    [as,bs,cs,ds,in]=sdfid(dis,rate,config,Md);
    [A,B,C,D]=seriesm(A,B,C,D,as,bs,cs,ds,in);
  end
  if idelay==1
    [ads,bds,cds,dds,in]=delfid(sens,NP,dls);
    [A,B,C,D]=seriesm(A,B,C,D,ads,bds,cds,dds,in);
  end
  if integ==2
    [ain,bin,cin,din]=intfid(rate);
    [A,B,C,D]=seriesm(A,B,C,D,ain,bin,cin,din,rate+Md);
  end
  if ibess==1
    [ab,bb,cb,db,in]=besfid(sens,cfb);
    [A,B,C,D]=seriesm(A,B,C,D,ab,bb,cb,db,in);
  end
end

function [Md,Mr,Ma,Mct]=nums(indd,indr,inda,ict)
%-------------------------------------------------------------------------
%
%  This function provides the number of sensors and actuators
%
%  The inputs are:  indd    (Numbers of displacement sensors)
%                   indr    (Numbers of rate sensors)
%                   inda    (Numbers of acceleration sensors)
%                   ict     (Numbers of actuators)
%
%  The outputs are: Md      (Number of displacement sensors)
%                   Mr      (Number of rate sensors)
```

```
%                   Ma      (Number of acceleration sensors)
%                   Mct     (Number of actuators)
%
%  Call:   [Md,Mr,Ma,Mct]=nums(indd,indr,inda,ict)
%-------------------------------------------------------------------------

Md=length(indd);
Mr=length(indr);
Ma=length(inda);
Mct=length(ict);

function [B,C,D]=resabcd(indd,indr,inda,ict,B,C,D,config,N)
%-------------------------------------------------------------------------
%
%  The function rescales matrices A,B,C,D for current units
%
%  The inputs are:  indd    (Numbers of displacement sensors)
%                   indr    (Numbers of rate sensors)
%                   inda    (Numbers of acceleration sensors)
%                   ict     (Numbers of actuators)
%                   B,C,D   (Matrices in linear model)
%                   config  (==1 - DM,==2 - DM/EM,==3 - EM)
%                   N       (Number of states in model)
%
%  The outputs are: B,C,D   (Rescaled matrices in linear model)
%
%  Call:  [B,C,D]=resabcd(indd,indr,inda,ict,B,C,D,config,N)
%-------------------------------------------------------------------------

[Gs,Gct,Ks,Kct]=guv(1,1,indd,indr,inda,ict,config);

C=((Gs.*Ks)*ones(1,N)).*C;
B=(ones(N,1)*(Gct.*Kct')).*B;
D=(Gs.*Ks)*(Gct.*Kct').*D;

function [f,z,bb]=rg6dyn
%-------------------------------------------------------------------------
%
%  This function stores parameters of second order internal dynamics
%  for all 6 rate gyros
%
%  EM or DM Payload:  gyro c, gyro a, gyro b is the order
%  Bus:               gyro b, gyro a, gyro c is the order
%
%
%  The outputs are:  f  (frequencies)
%                    z  (damping ratios)
%      bb (gains)
%
%  Call:  [f z bb]=rg6dyn
%-------------------------------------------------------------------------

fm=[294.0531 304.106 297.823 296.5663 294.0531 296.5663];
zm=[.3 .35 .34 .32 .32 .31];
```

```
bbm=[447.566 458.169 454.982 457.49 453.613 453.062];

OD=[3 1 2 5 4 6];      %  The order of RG

f(OD)=fm;              %  Put in the proper order (PL-B / a-b-c)
z(OD)=zm;
bb(OD)=bbm;

function RM=rotmat
%-------------------------------------------------------------------------------
%
%  This function stores the rotation matrix for DM TWA
%
%  Call:      RM=rotmat
%-------------------------------------------------------------------------------

RM = [        0    0.577286   -0.816539
       0.707146    0.577286    0.408271
      -0.707146    0.577286    0.408271 ];

function RM=rotmat2
%-------------------------------------------------------------------------------
%
%  This function stores the rotation matrix for EM RWA
%
%  Call:      RM=rotmat2
%-------------------------------------------------------------------------------

RM = [        0    0.577286    0.816539
       0.707146    0.577286   -0.408271
      -0.707146    0.577286   -0.408271 ];

function [as,bs,cs,ds,in]=sdfid(dis,rate,config,Md)
%-------------------------------------------------------------------------------
%
%  This function creates a state space representation of any sensor dynamics
%
%  The inputs are:  dis    (Numbers of displacement sensors)
%                   rate   (Numbers of rate sensors)
%                   config(==1-DM,==2-DM/EM,==3-EM)
%                   Md     (Total number of displacement sensors)
%                   in     (Numbers of system inputs with sensor dynamics)
%
%  The outputs are: as,bs,cs,ds (Matrices in linear system)
%
%  Call:          [as,bs,cs,ds,in]=sdfid(dis,rate,config)
%-------------------------------------------------------------------------------

if (config==3)
  return
else
[f6,z6,bb6]=rg6dyn;
f=f6(rate);
z=z6(rate);
```

```
kmat=diag(f.^2);
cmat=diag(2*f.*z);
Nrs=length(f);
as=[zeros(Nrs,Nrs) eye(Nrs); -kmat, -cmat];
b=diag(f);
c=diag(f);
bs=[zeros(Nrs,Nrs); b];
cs=[c zeros(Nrs,Nrs)];
ds=zeros(Nrs,Nrs);
in=rate+Md;
end

function [A,B,C,D]=seriesm(A1,B1,C1,D1,A2,B2,C2,D2,outputs1)
%-------------------------------------------------------------------------
%
%  This function is a modification of MATLAB's series.m
%
%  The inputs are:  Matrices A,B,C and D of 1st and 2nd systems
%                   Numbers of selected outputs of the 1st system
%
%  The outputs are: Matrices A,B,C and D of the extended system
%
%  Call:  [A,B,C,D]=seriesm(A1,B1,C1,D1,A2,B2,C2,D2,outputs1)
%-------------------------------------------------------------------------

[Ny,Nx]=size(C1);
inputs2=1:length(outputs1);
[A,B,Cr,Dr]=series(A1,B1,C1,D1,A2,B2,C2,D2,outputs1,inputs2);
Nxe=length(A);
C=zeros(Ny,Nxe);
C(:,1:Nx)=C1;
D=D1;
C(outputs1,:)=Cr;
D(outputs1,:)=Dr;

function [Ar,Br,Cr,Dr]=stabint
%-------------------------------------------------------------------------
%
%  This function stores A,B,C,D matrices of stabilized integrator
%
%  Call:    [Ar Br Cr Dr]=stabint
%-------------------------------------------------------------------------

Ar=[0 1; -0.03553 -0.2666];
Br=[0; 1];
Cr=[0 1];
Dr=0;

function [Kd,Kr,Ka,Kct]=units1(volts,ipur)
%-------------------------------------------------------------------------
%
%  This function stores coefficients K for the 1 payload case:
%  [VOLTS] = K * [PHYSICAL UNITS] (sensors)
%  [PHYSICAL UNITS] = K * [VOLTS] (actuators)
```

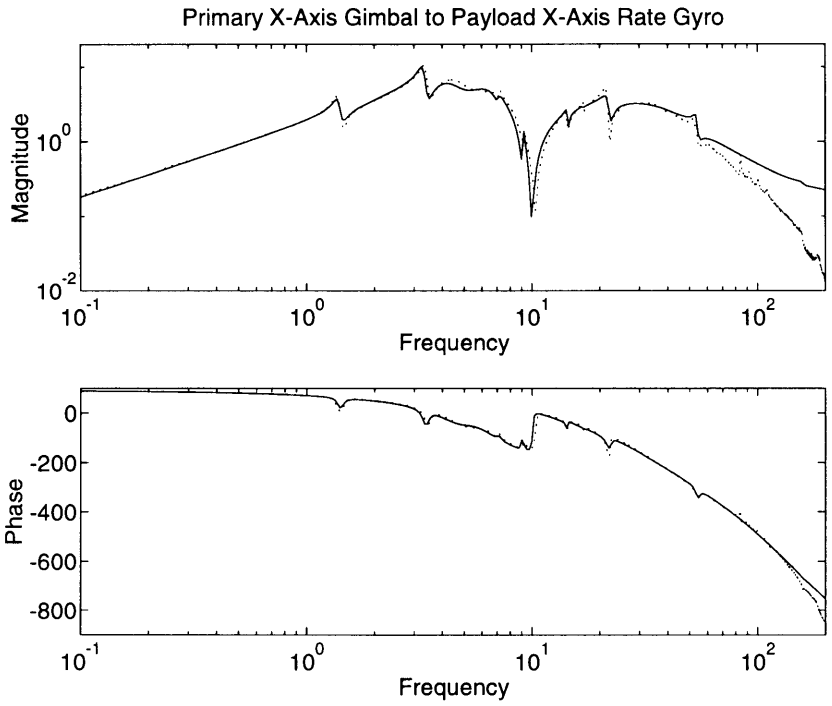
```
%
%  The inputs are:  volts (==1 - Volts; ==0 - Physical units)
%                   ipur  (Reflects the purpose: ==0 - for y; ==1 - for B,C,D)
%
%  The outputs are: Coefficients K for sensors and actuators
%
%  Call:         [Kd,Kr,Ka,Kct]=units1(volts,ipur)
%-------------------------------------------------------------------------

[f,z,bb]=rg6dyn;     %  Internal dynamics of 6 RG

Uen=1.;              %  Encoders
Ustgn=2241.75;       %  Strain gauges  [V/strain]
Urg=(bb./f).^2;      %  6 Rate gyros  [V/(rad/s)]
Uacc=1.3762;         %  Accelerometers  [V/(m/s^2)]
Ugimbdm=.9867*.6581; %  Gimbal (DM) [Nm/V]
Ubus=3*.03;          %  TWA  [Nm/V]

Kd=[Uen*ones(2,1); Ustgn*ones(8,1)];
Kr=Urg';
Ka=Uacc*ones(6,1);
Kct=[Ugimbdm*ones(2,1); Ubus*ones(3,1)];

function [Kd,Kr,Ka,Kct]=units2(volts,ipur)
%-------------------------------------------------------------------------
%
%  This function stores coefficients K for the 2 payload case:
%  [VOLTS] = K * [PHYSICAL UNITS] (sensors)
%  [PHYSICAL UNITS] = K * [VOLTS] (actuators)
%
%  The inputs are:  volts (==1 - Volts; ==0 - Physical units)
%                   ipur  (Reflects the purpose: ==0 - for y; ==1 - for B,C,D)
%
%  The outputs are: Coefficients K for sensors and actuators
%
%  Call:         [Kd,Kr,Ka,Kct]=units2(volts,ipur)
%-------------------------------------------------------------------------

[f,z,bb]=rg6dyn;       %  Internal dynamics of 6 RG

Uen=1.;                %  Encoders
Ustgn=2241.75;         %  Strain gauges  [V/strain]
Urg=(bb./f).^2;        %  6 Rate gyros  [V/(rad/s)]
Uacc=1.3762;           %  Accelerometers  [V/(m/s^2)]
Ugimbdm=.9867*.6581;   %  Gimbal (DM) [Nm/V]
Ugimbem=.2647*.4675*5; %  Gimbal (EM) [Nm/V]
Ubus=3*.03;            %  TWA  [Nm/V]
Uactstr=0.004993;      %  Active strut [Nm/V]

Kd=[Uen*ones(2,1); Ustgn*ones(8,1); Uen*ones(2,1)];
Kr=Urg';
Ka=Uacc*ones(6,1);
Kct=[Ugimbdm*ones(2,1); Ubus*ones(3,1); Ugimbem*ones(2,1); Uactstr*ones(2,1)];
```

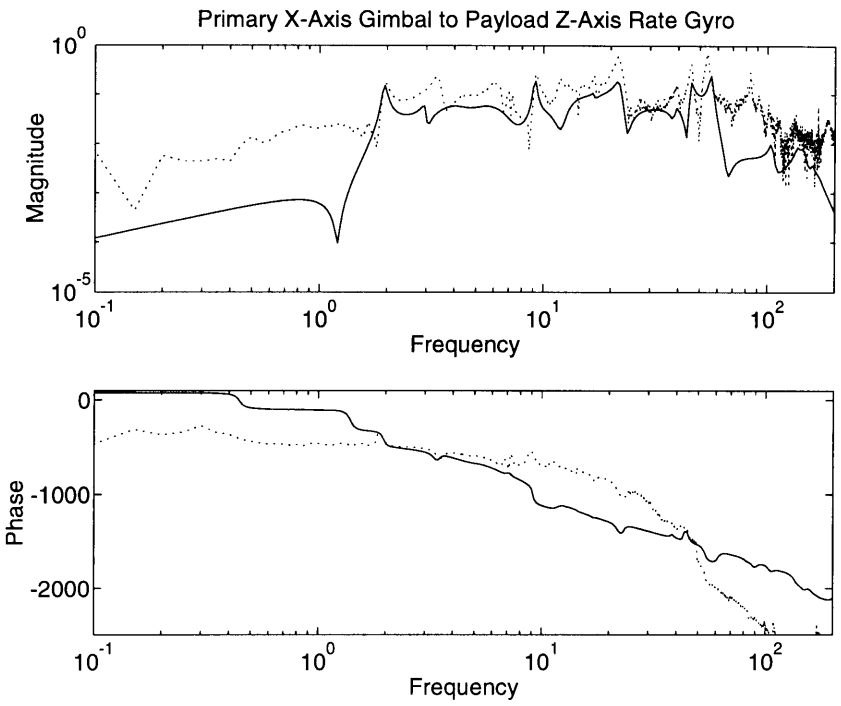
```
function [Kd,Kr,Ka,Kct]=units3(volts,ipur)
%-------------------------------------------------------------------------
%
%  This function stores coefficients K for the EM:
%  [VOLTS] = K * [PHYSICAL UNITS] (sensors)
%  [PHYSICAL UNITS] = K * [VOLTS] (actuators)
%
%  The inputs are:  volts (==1 - Volts; ==0 - Physical units)
%                   ipur  (Reflects the purpose: ==0 - for y; ==1 - for B,C,D)
%
%  The outputs are: Coefficients K for sensors and actuators
%
%  Call:         [Kd,Kr,Ka,Kct]=units3(volts,ipur)
%-------------------------------------------------------------------------

Uen=1.;                 %  Encoders
Ustgn=500;              %  Strain gauges  [V/strain]
Urg=[12.319*2.30        %  5 Rate gyros   [V/(rad/s)]
     12.319*2.70
     12.319*2.59
     12.319*2.32
     12.319*2.39];
Utach=0.093424;         %  Tachometers [V/(rad/s)]
Ugimb=0.2625;           %  Gimbals [Nm/V]
Ubus=0.10306;           %  RWA  [Nm/V]
Uactstr=0.12233;        %  Active strut [Nm/V]

Kd=[Uen*ones(4,1); Ustgn*ones(8,1)];
Kr=[Uen*ones(4,1); Urg; Utach*ones(3,1)];
Kct=[Ugimb*ones(4,1); Ubus*ones(3,1); Uactstr*ones(2,1)];
```

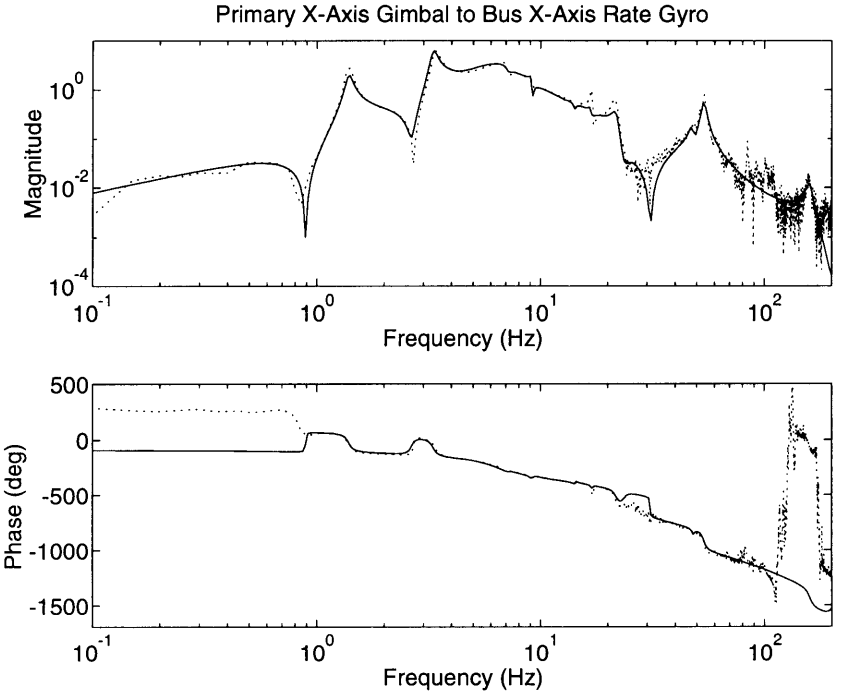
Appendix D

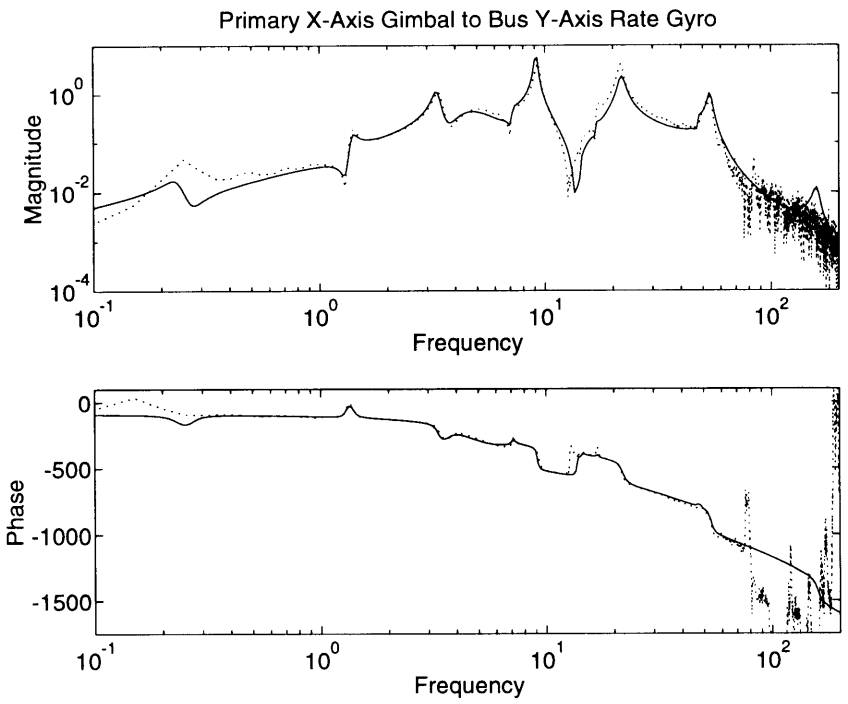
Catalog of Transfer Functions

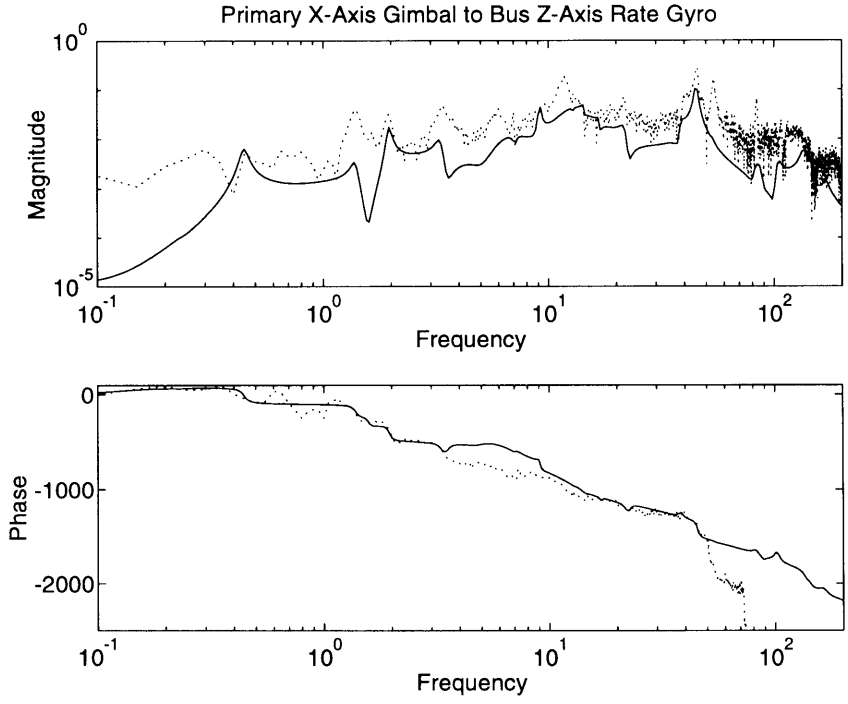
This is a catalog of the MACE Engineering Model transfer functions. In all plots, the updated 1-g finite element model is the solid line and the experimental data is the dotted line. The first group of transfer functions is from the gimbals (primary then secondary) and reaction wheels to the payload and bus rate gyros for both XY- and Z-axis behavior (a total of 35 transfer functions). The second group is the Z-axis actuators to the Z-axis strain gages (12 transfer functions).

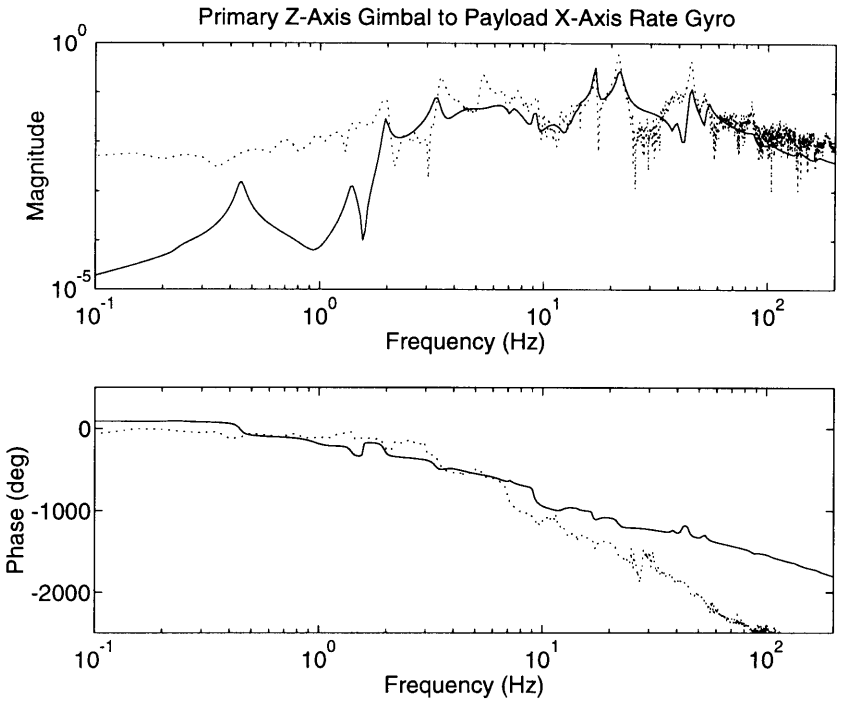


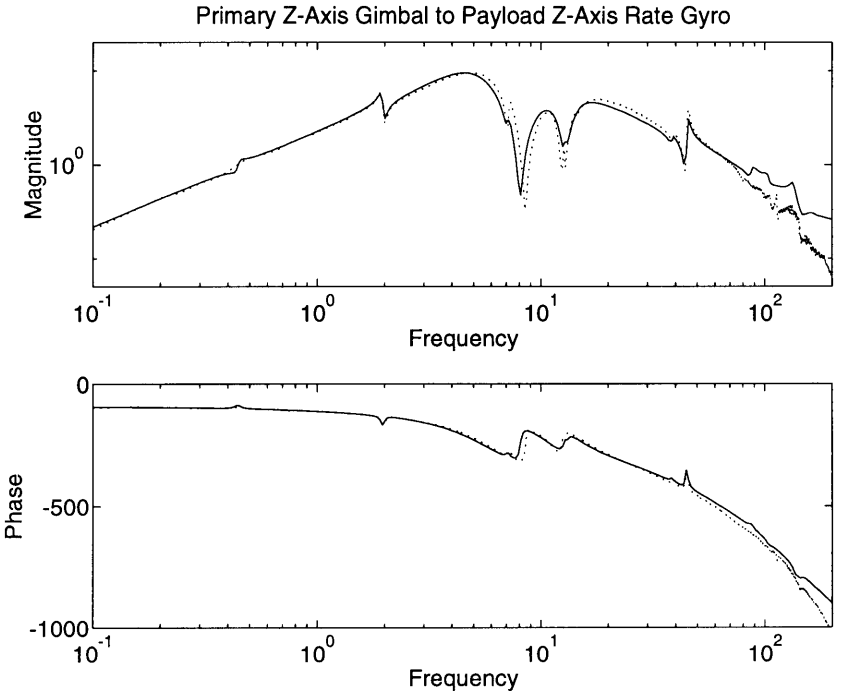

Primary X-Axis Gimbal to Payload Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

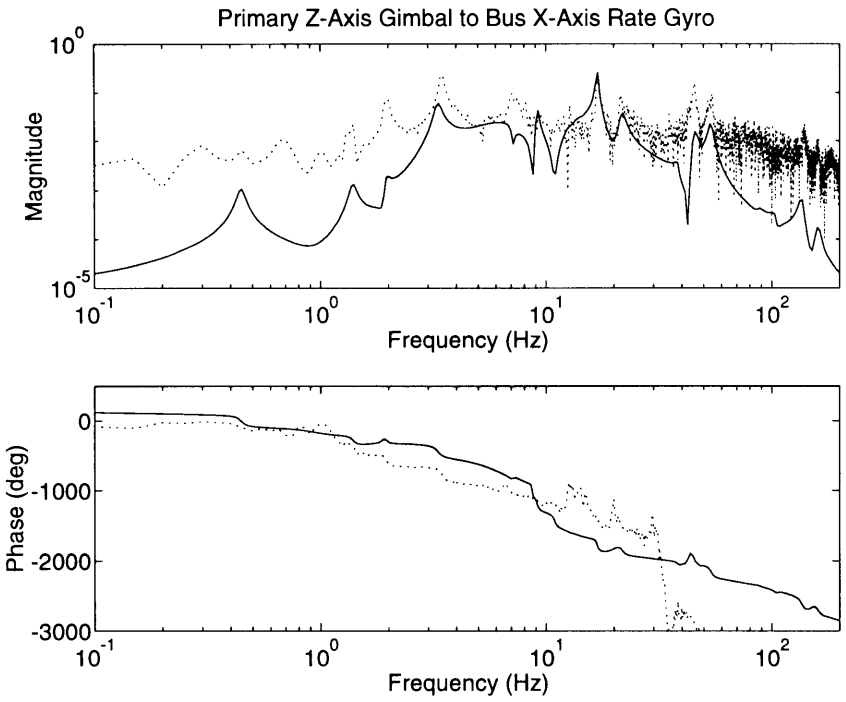

Primary X-Axis Gimbal to Bus X-Axis Rate Gyro
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

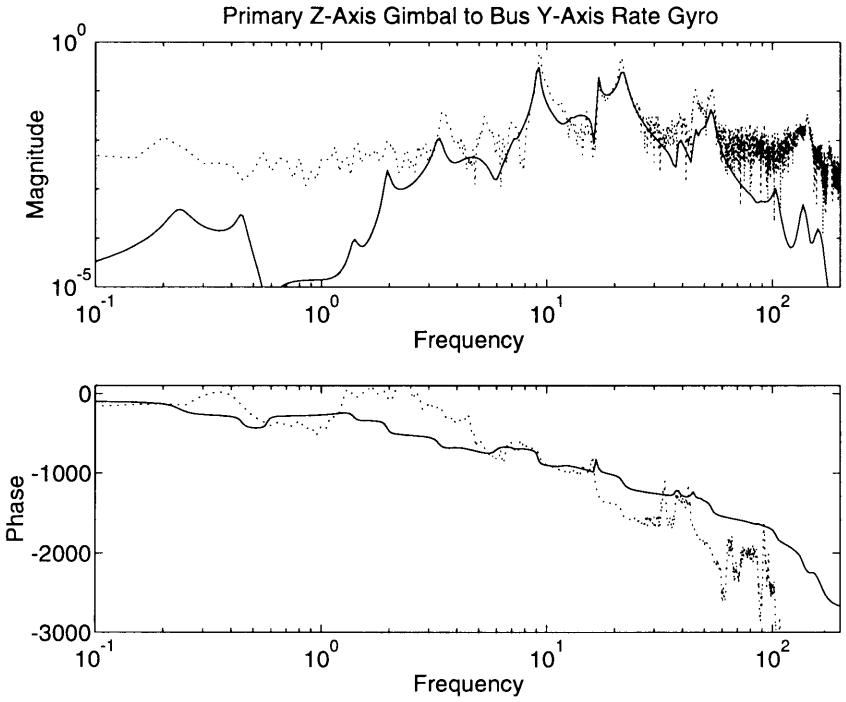

Primary X-Axis Gimbal to Bus Y-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

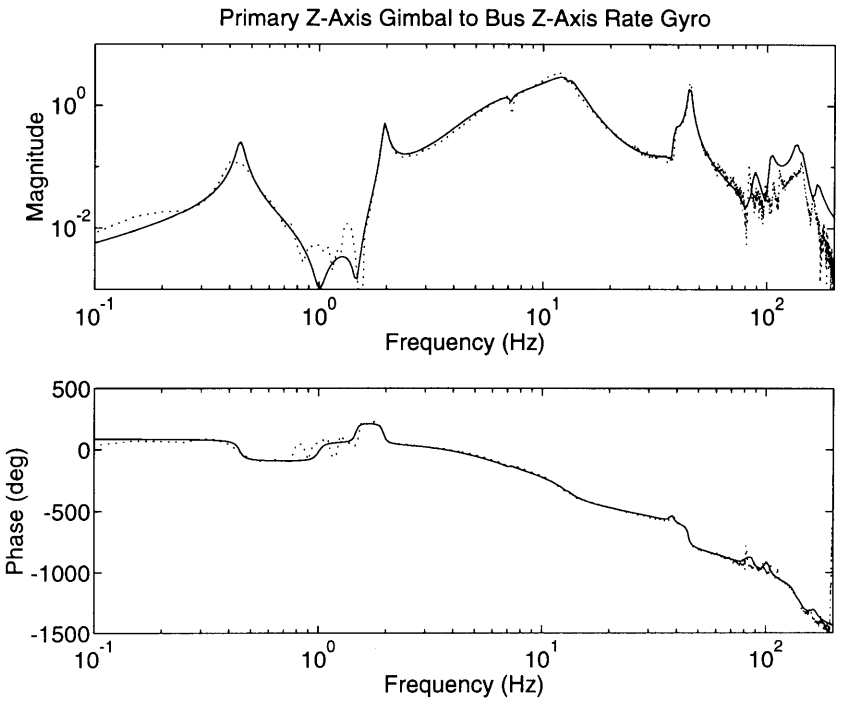

Primary X-Axis Gimbal to Bus Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

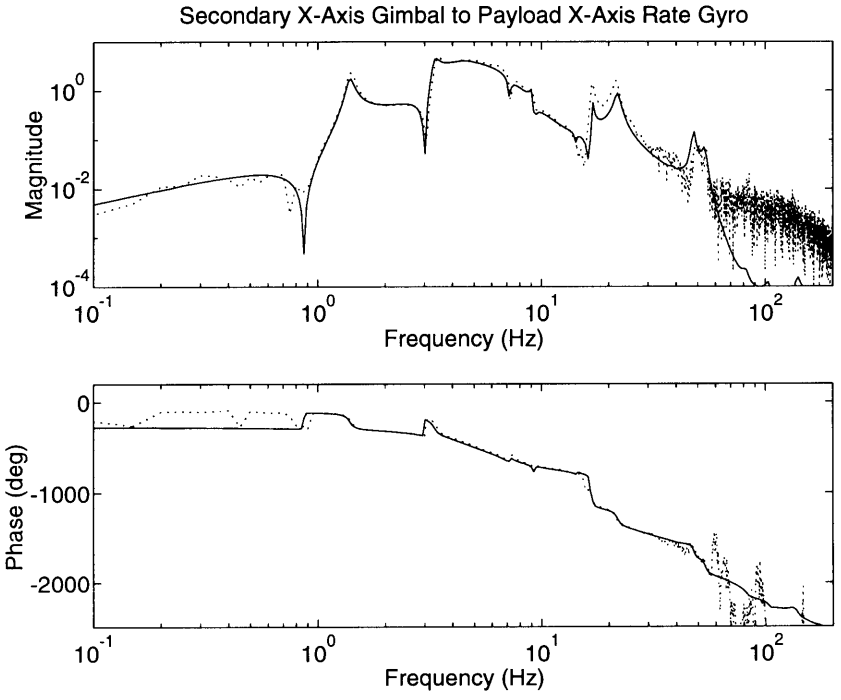

Primary Z-Axis Gimbal to Payload X-Axis Rate Gyro
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

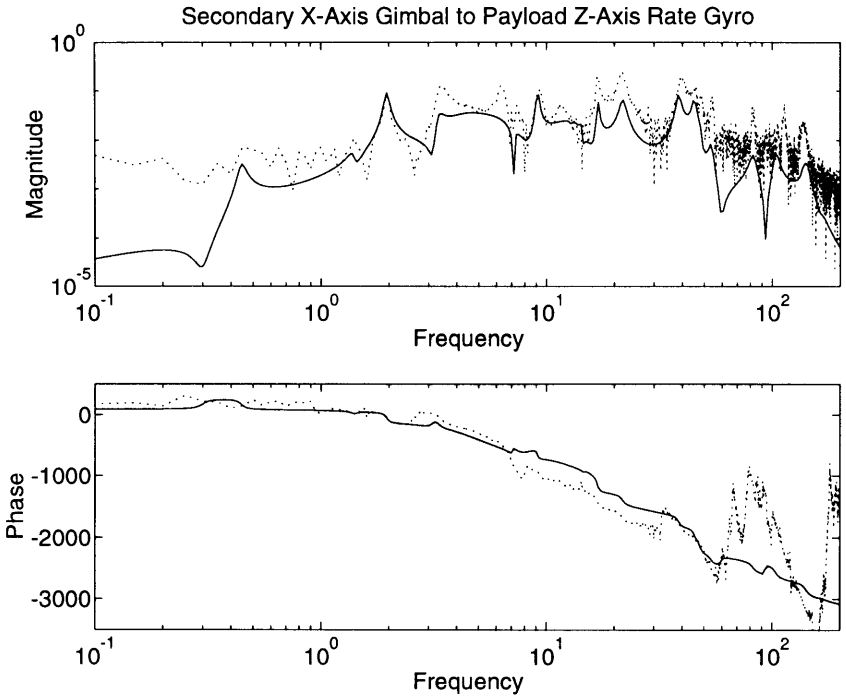

Primary Z-Axis Gimbal to Payload Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

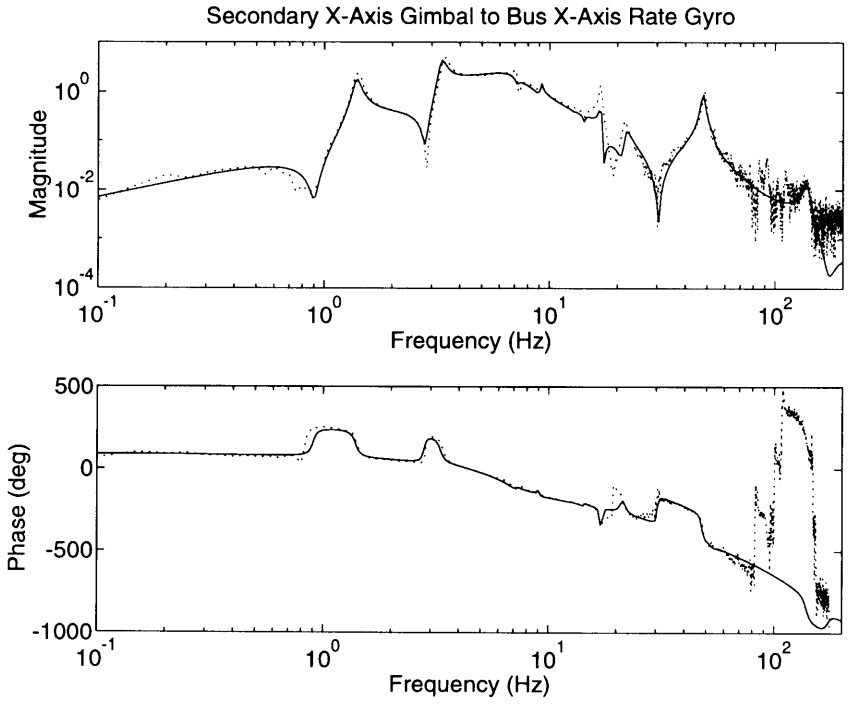

Primary Z-Axis Gimbal to Bus X-Axis Rate Gyro
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

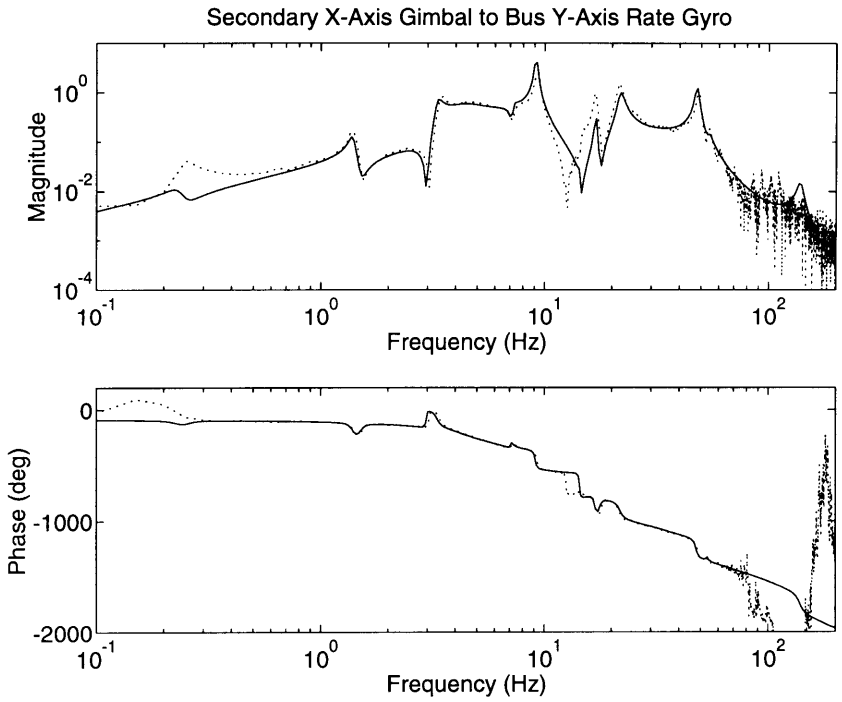

Primary Z-Axis Gimbal to Bus Y-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

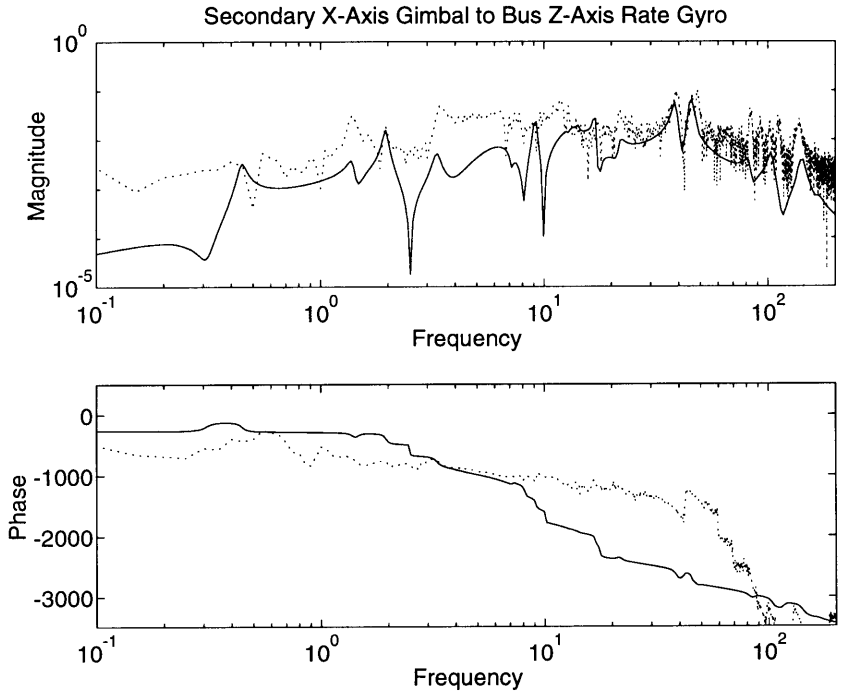

Primary Z-Axis Gimbal to Bus Z-Axis Rate Gyro
Magnitude
10^{0}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)
Phase (deg)
500
0
-500
-1000
-1500
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)

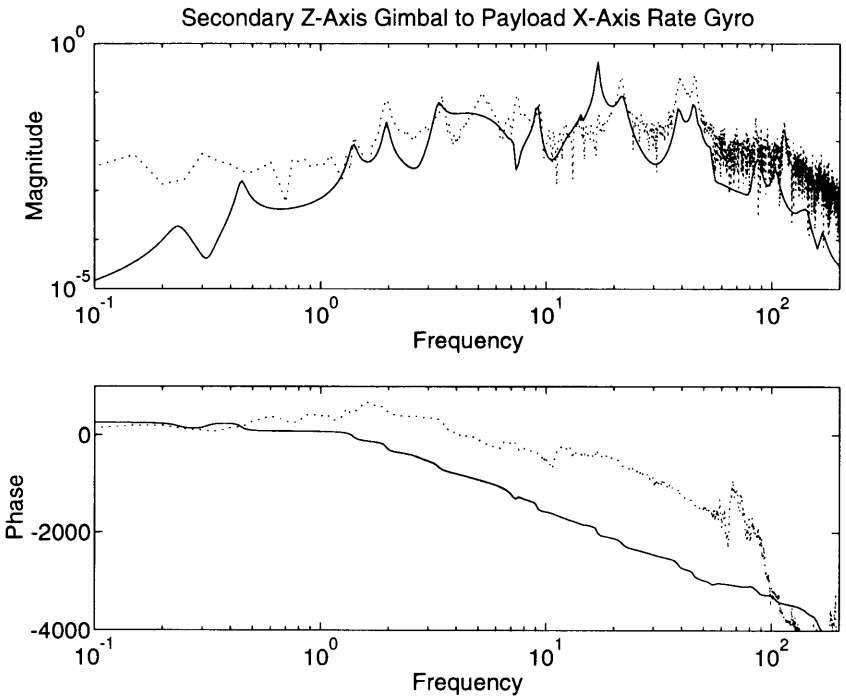

Secondary X-Axis Gimbal to Payload X-Axis Rate Gyro
Magnitude
10^{0}
10^{-2}
10^{-4}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)
Phase (deg)
0
-1000
-2000
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)

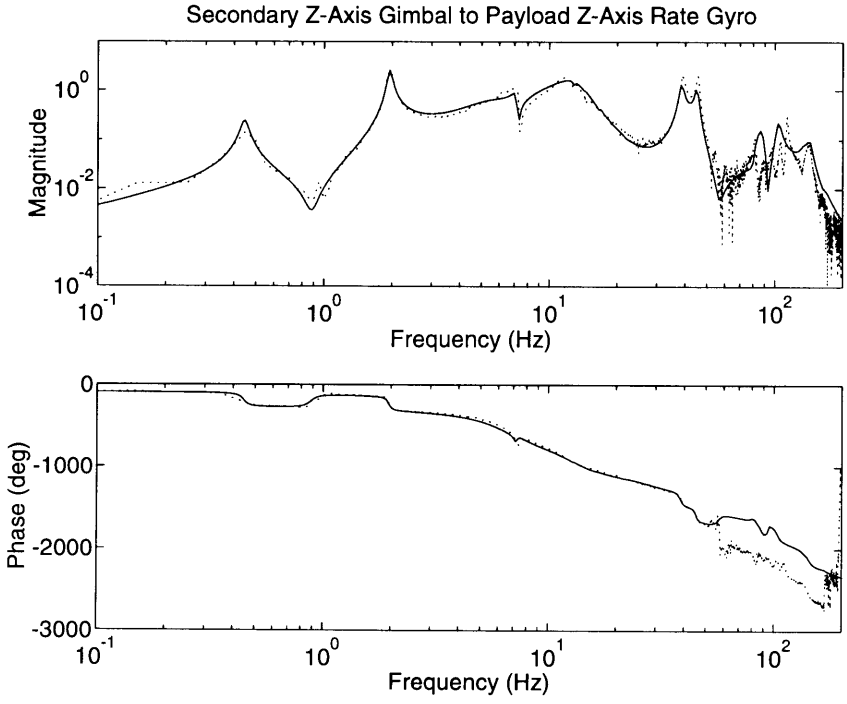

Secondary X-Axis Gimbal to Payload Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

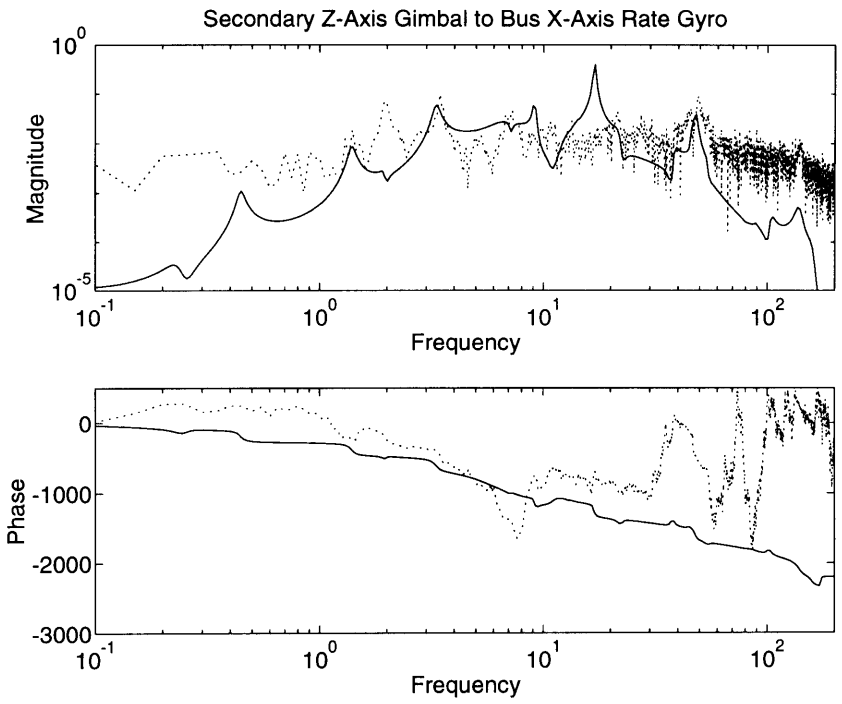

Secondary X-Axis Gimbal to Bus X-Axis Rate Gyro
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

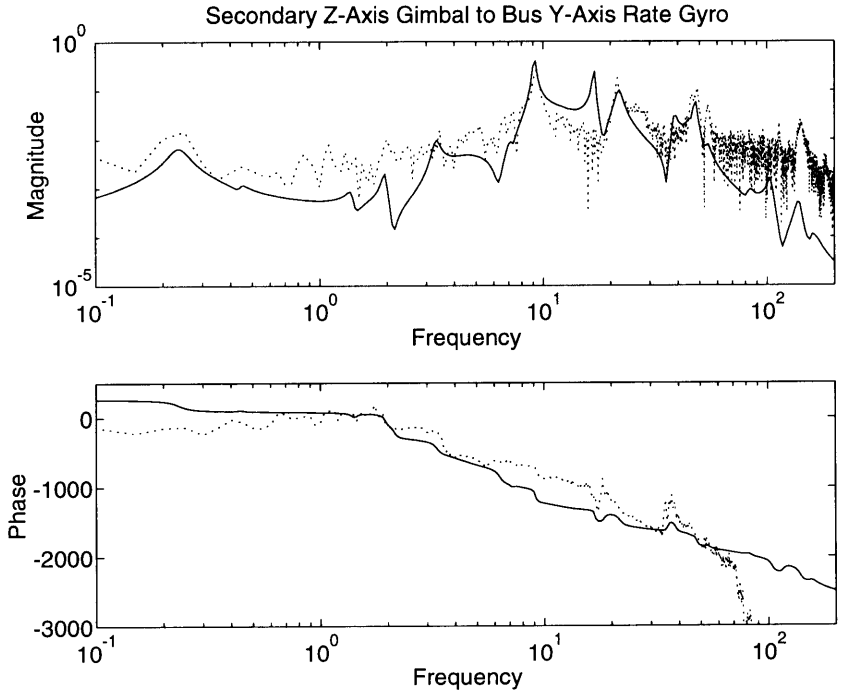

Secondary X-Axis Gimbal to Bus Y-Axis Rate Gyro
Magnitude
10^{0}
10^{-2}
10^{-4}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)
Phase (deg)
0
-1000
-2000
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)

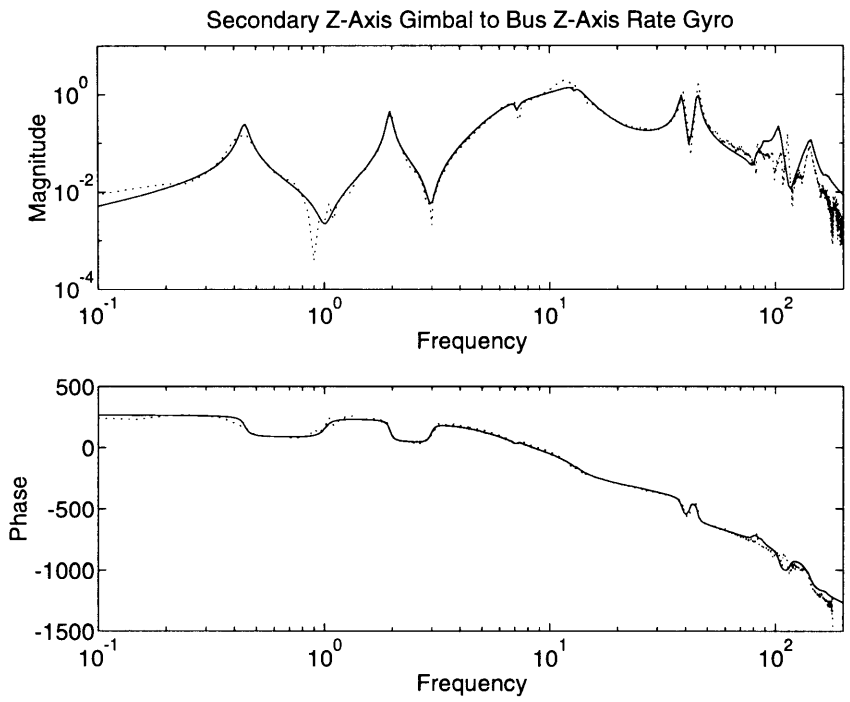

Secondary X-Axis Gimbal to Bus Z-Axis Rate Gyro
Magnitude
10^{0}
10^{-5}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency
Phase
0
-1000
-2000
-3000
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency

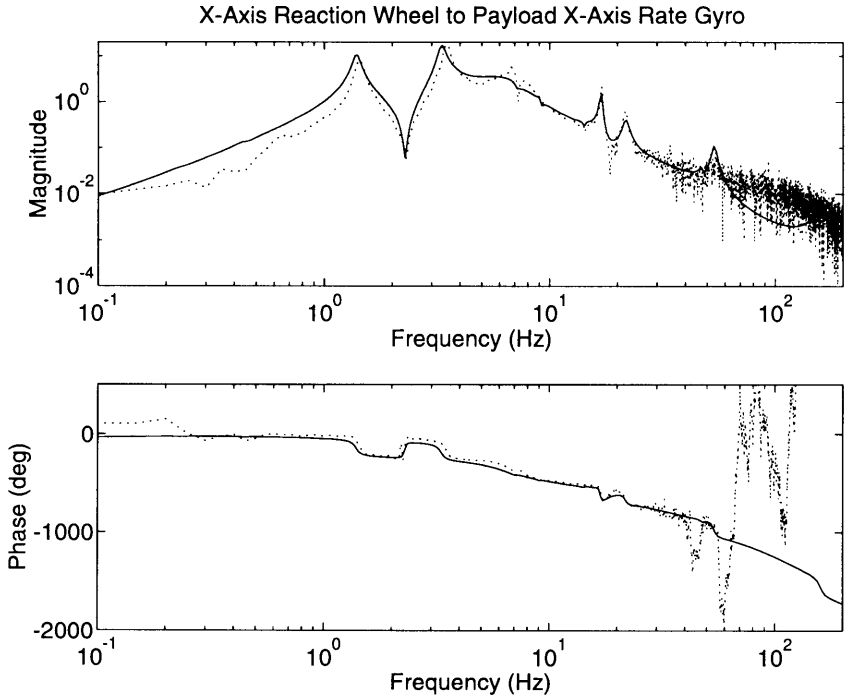

Secondary Z-Axis Gimbal to Payload X-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

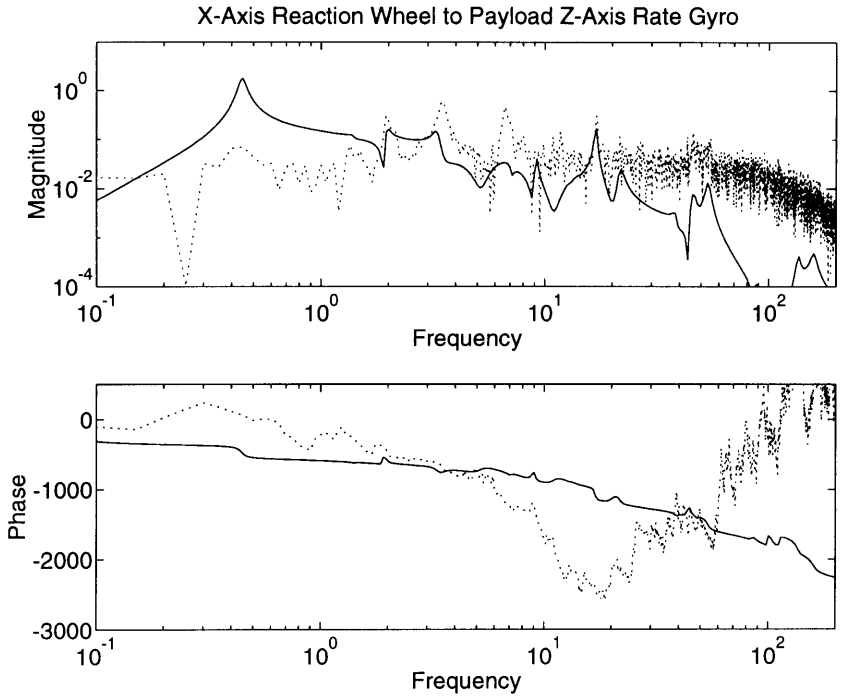

Secondary Z-Axis Gimbal to Payload Z-Axis Rate Gyro
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

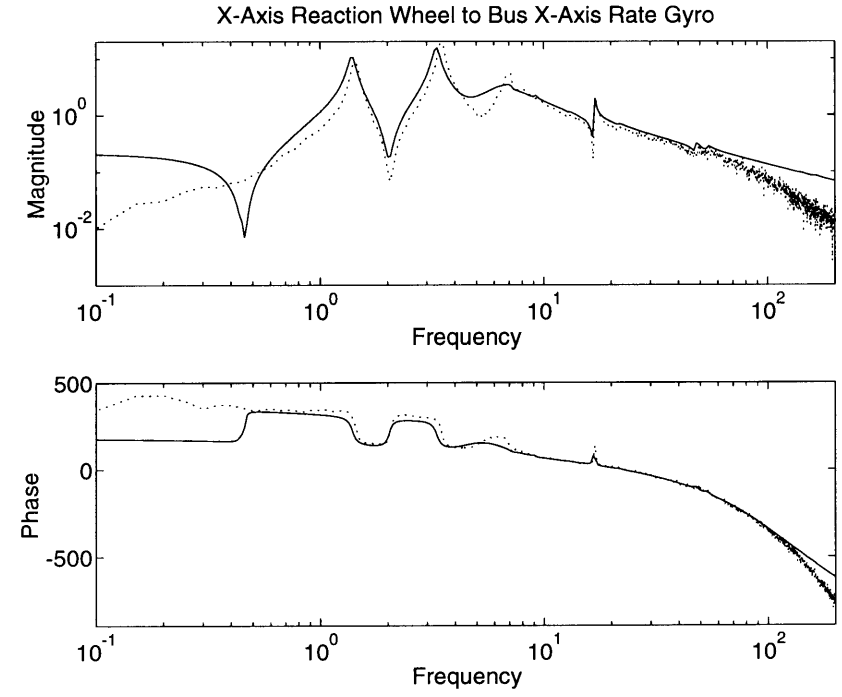

Secondary Z-Axis Gimbal to Bus X-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

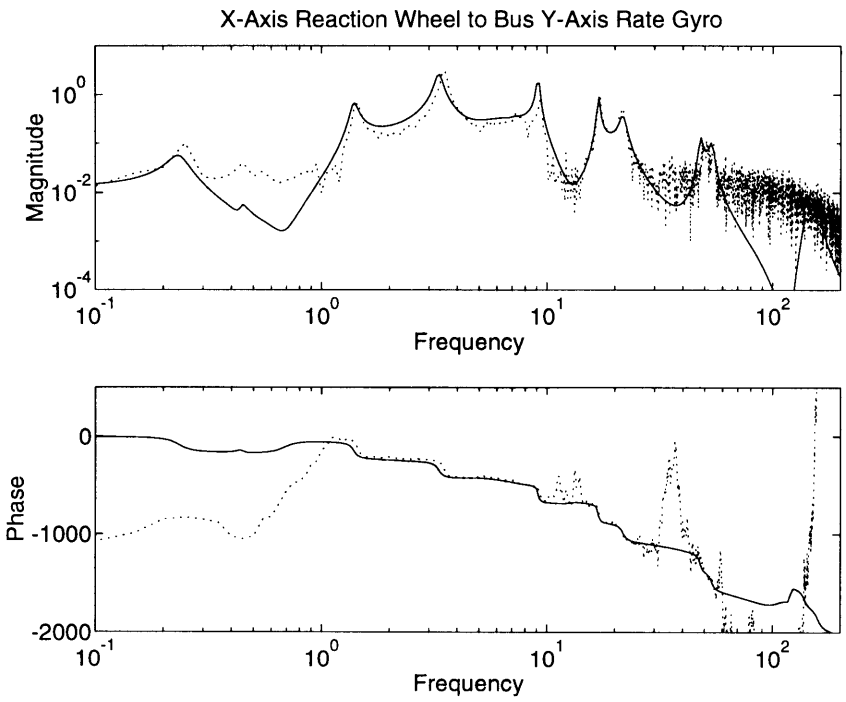

Secondary Z-Axis Gimbal to Bus Y-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

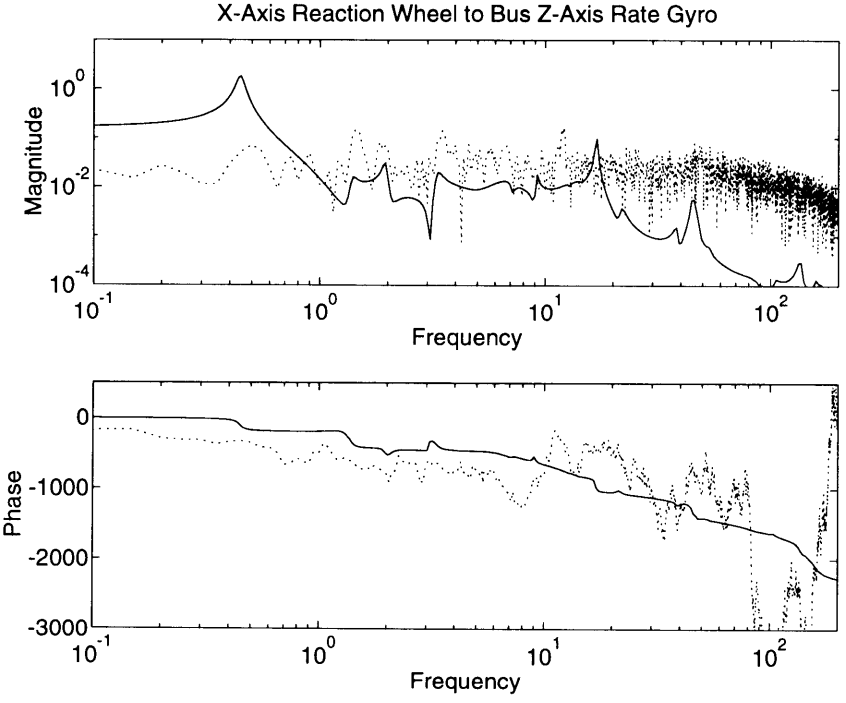

Secondary Z-Axis Gimbal to Bus Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

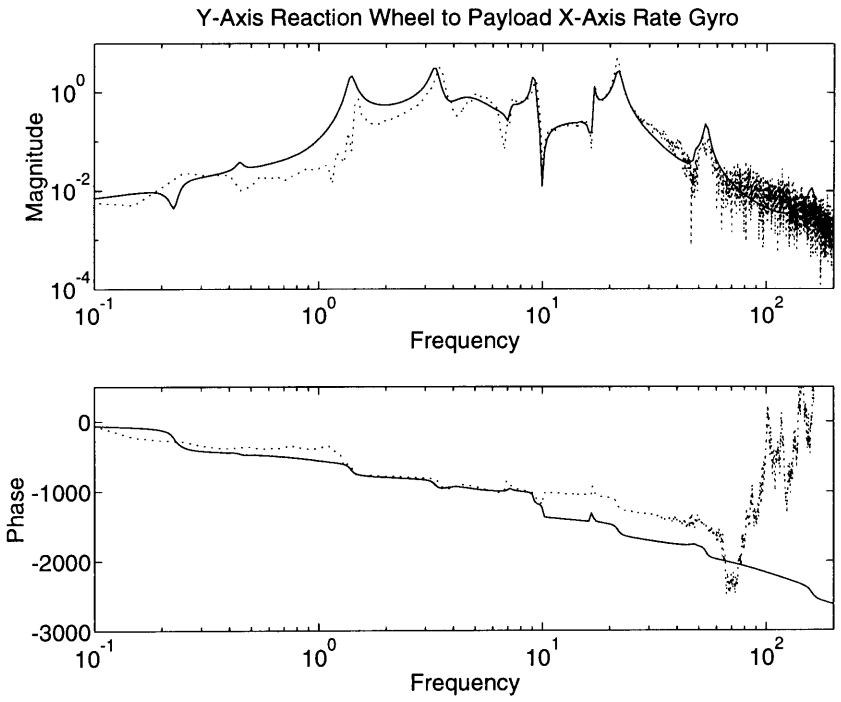

X-Axis Reaction Wheel to Payload X-Axis Rate Gyro
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

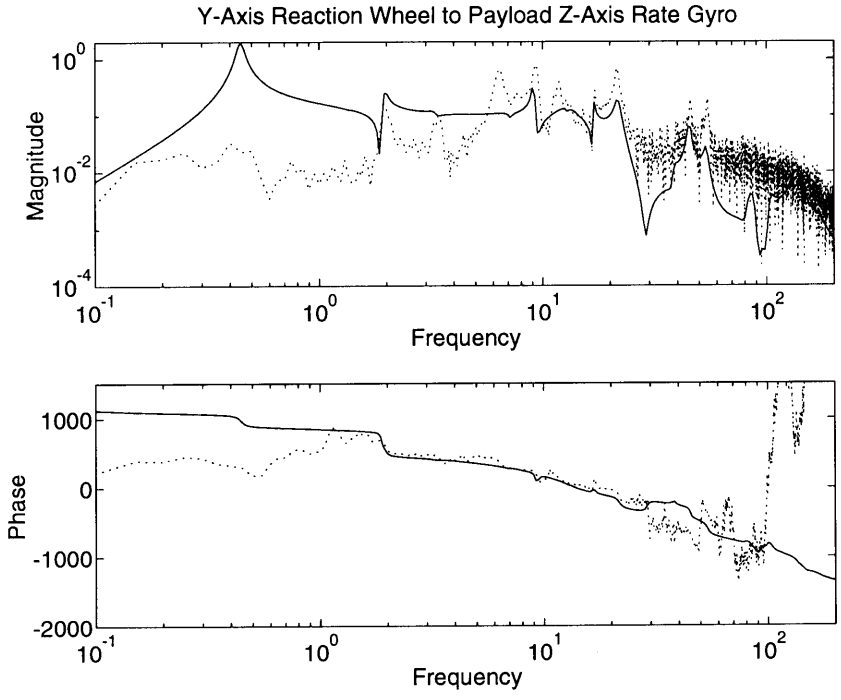

X-Axis Reaction Wheel to Payload Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

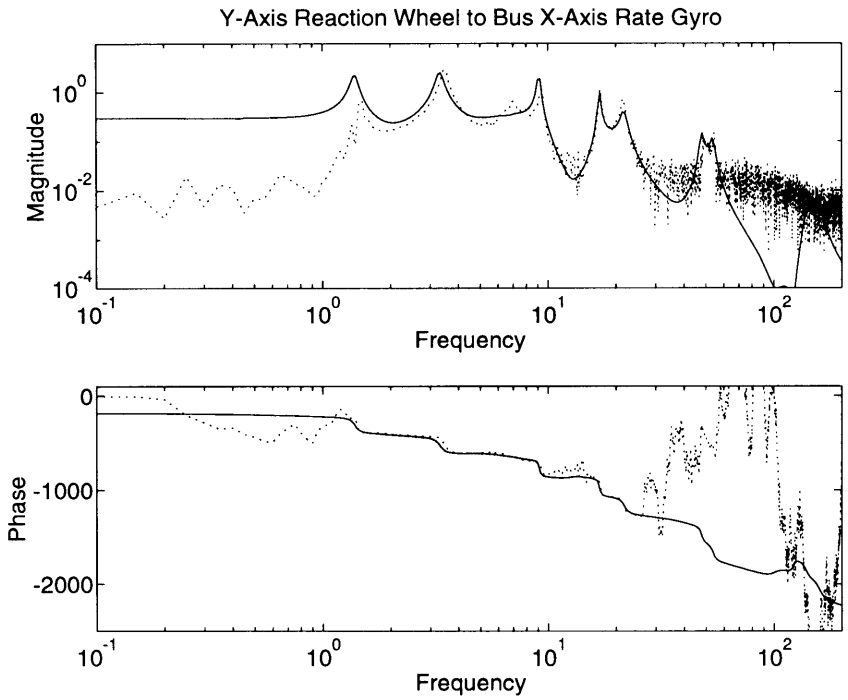

X-Axis Reaction Wheel to Bus X-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

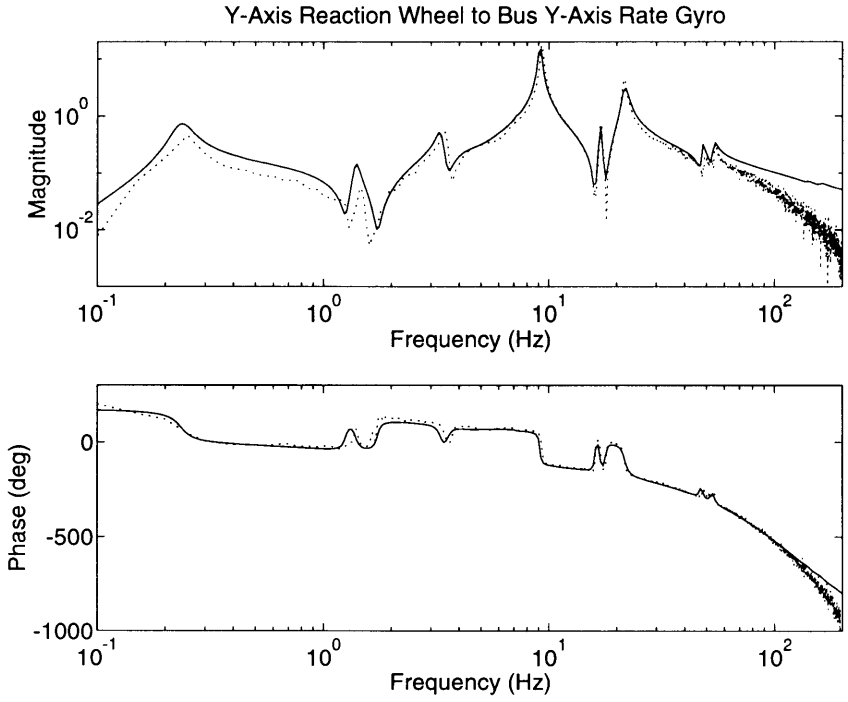

X-Axis Reaction Wheel to Bus Y-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

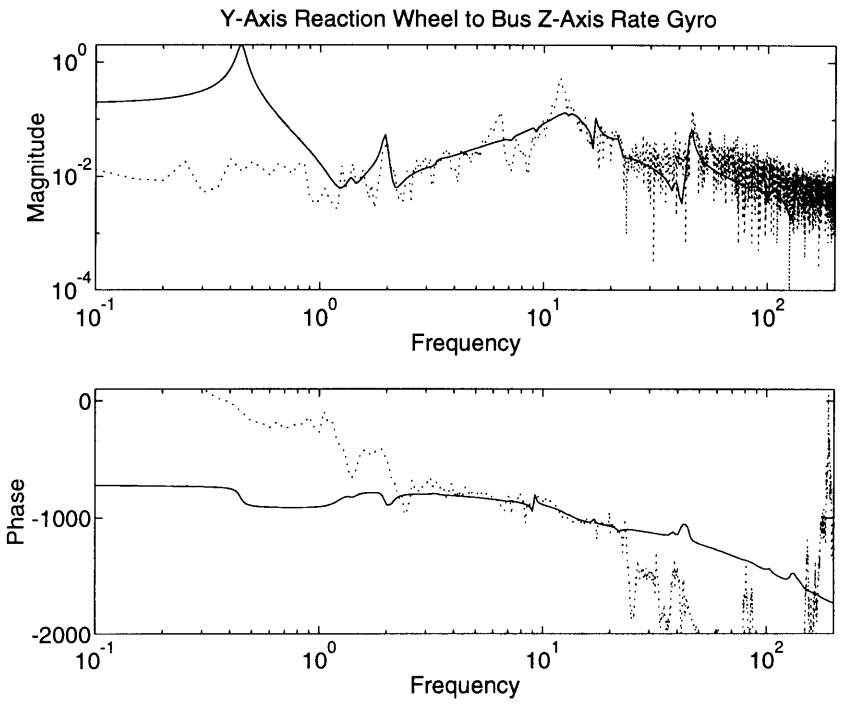

X-Axis Reaction Wheel to Bus Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

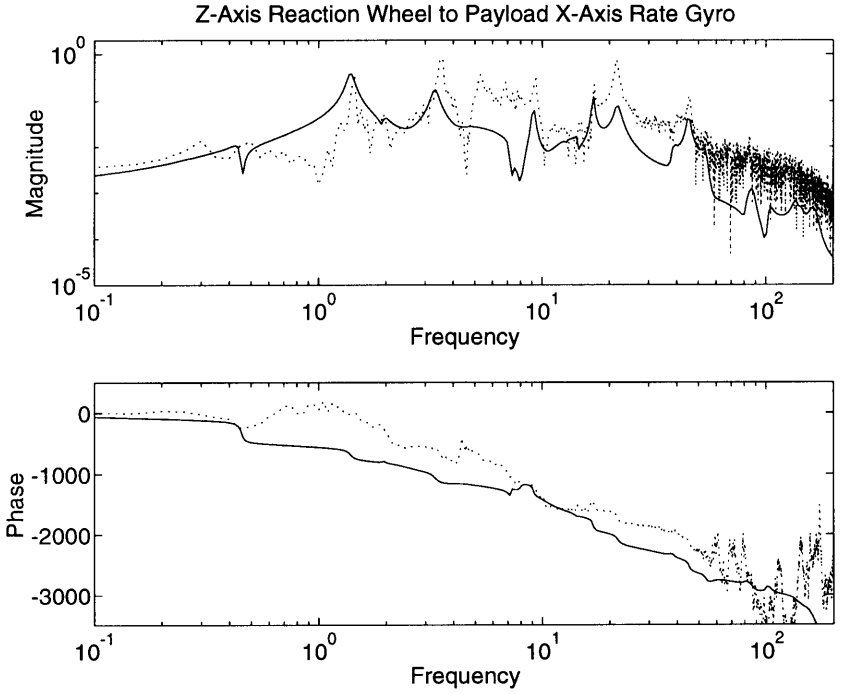

Y-Axis Reaction Wheel to Payload X-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

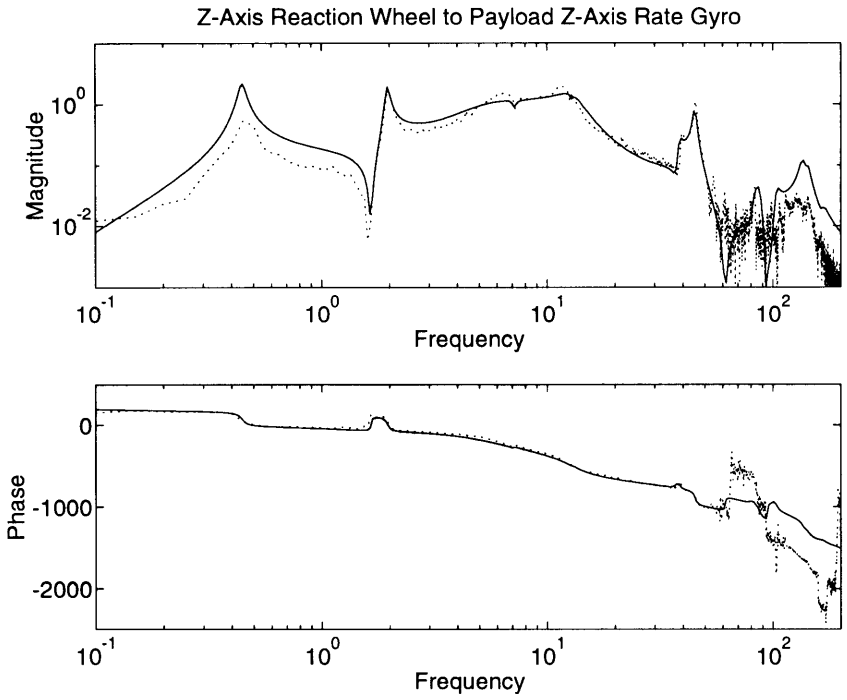

Y-Axis Reaction Wheel to Payload Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

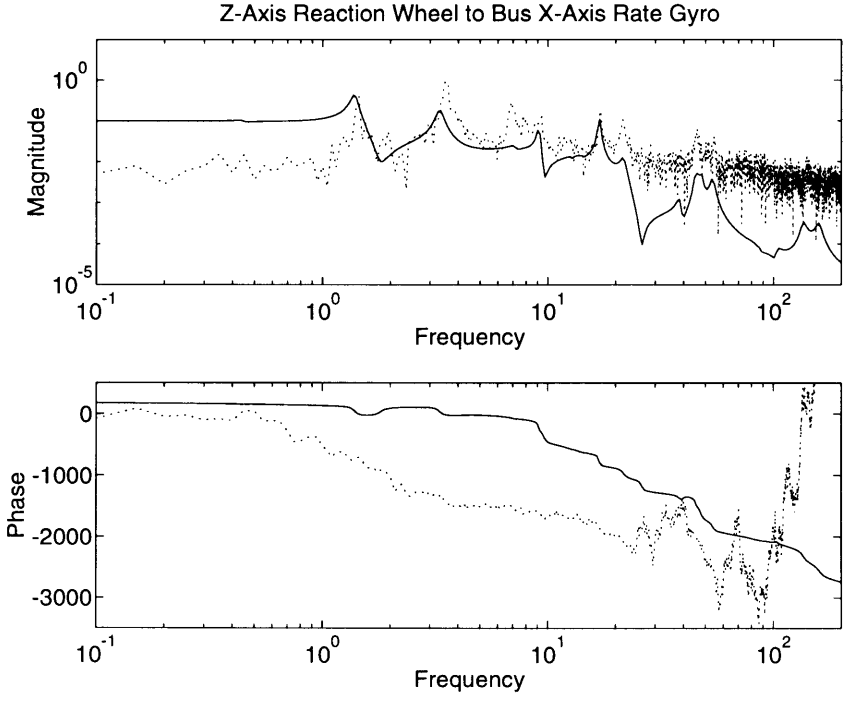

Y-Axis Reaction Wheel to Bus X-Axis Rate Gyro
Magnitude
10^{0}
10^{-2}
10^{-4}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency
Phase
0
-1000
-2000
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency

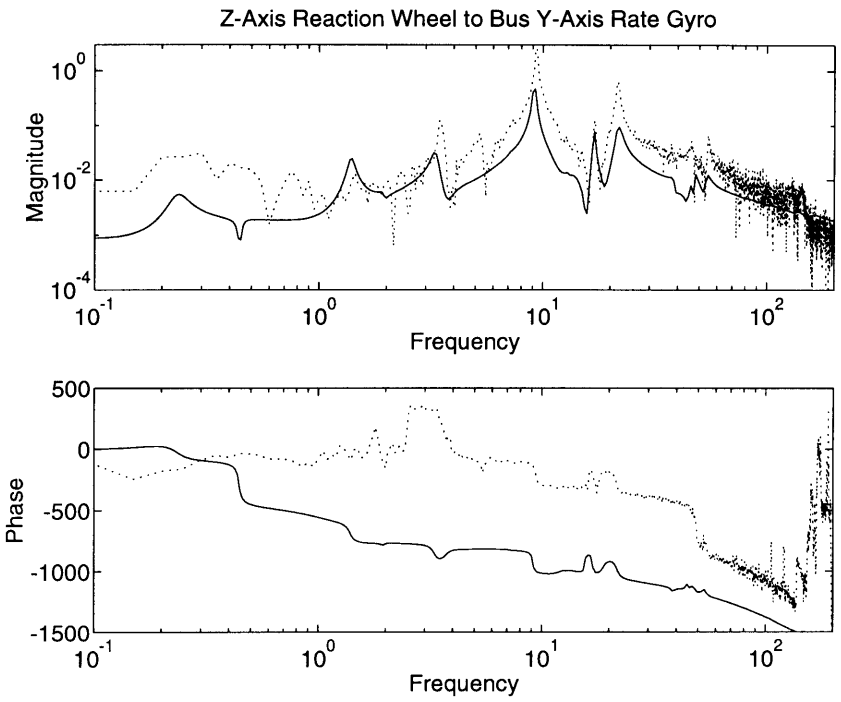

Y-Axis Reaction Wheel to Bus Y-Axis Rate Gyro
Magnitude
10^{0}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)
Phase (deg)
0
-500
-1000
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)

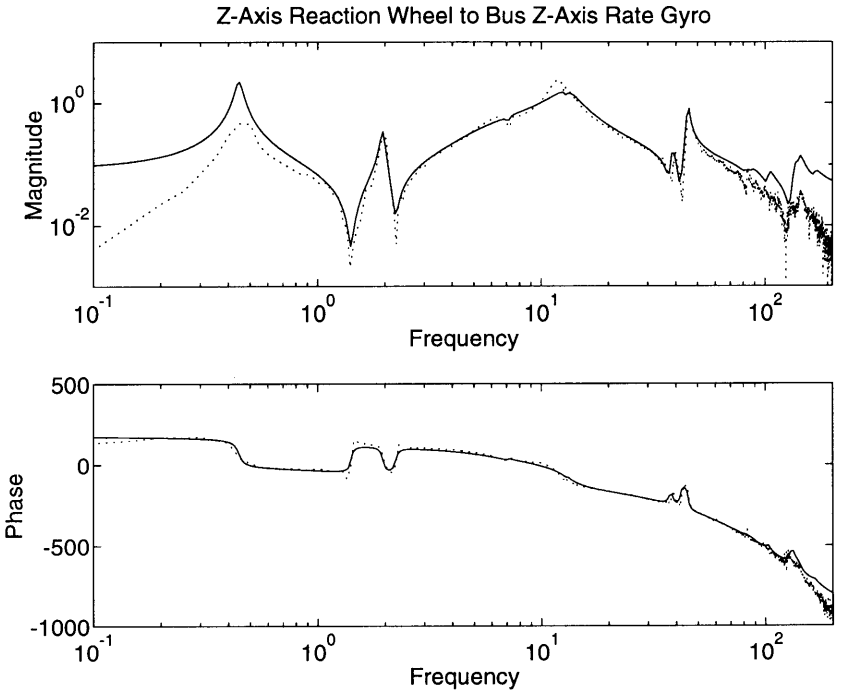

Y-Axis Reaction Wheel to Bus Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

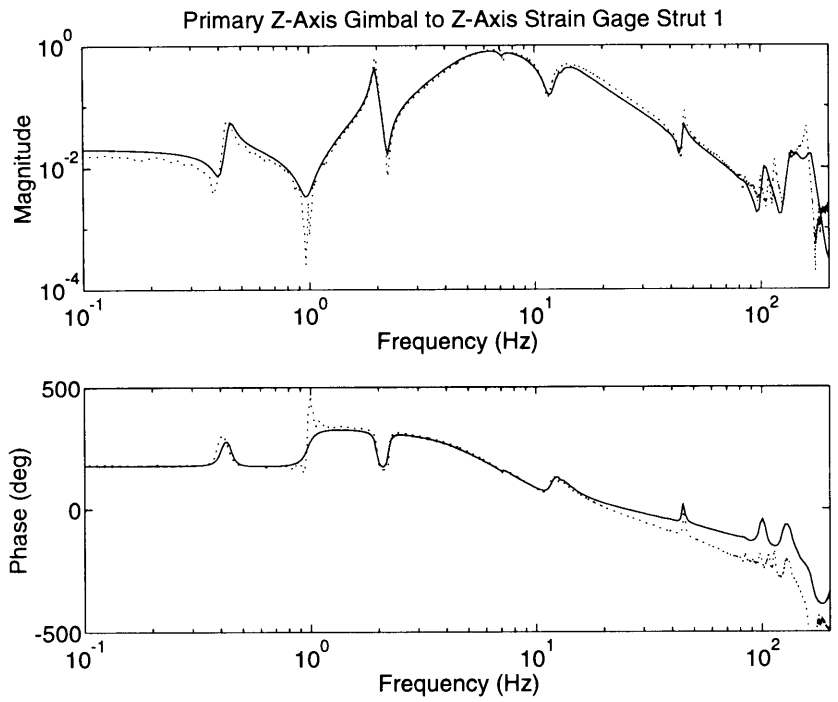

Z-Axis Reaction Wheel to Payload X-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

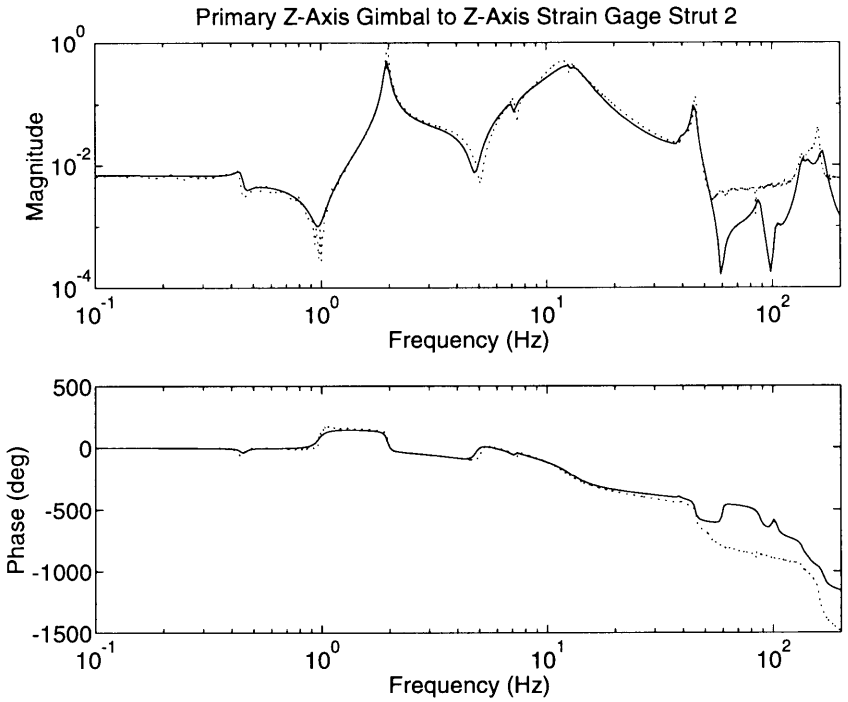

Z-Axis Reaction Wheel to Payload Z-Axis Rate Gyro
Magnitude
10^0
10^{-2}
10^{-1}
10^0
10^1
10^2
Frequency
Phase
0
-1000
-2000
10^{-1}
10^0
10^1
10^2
Frequency

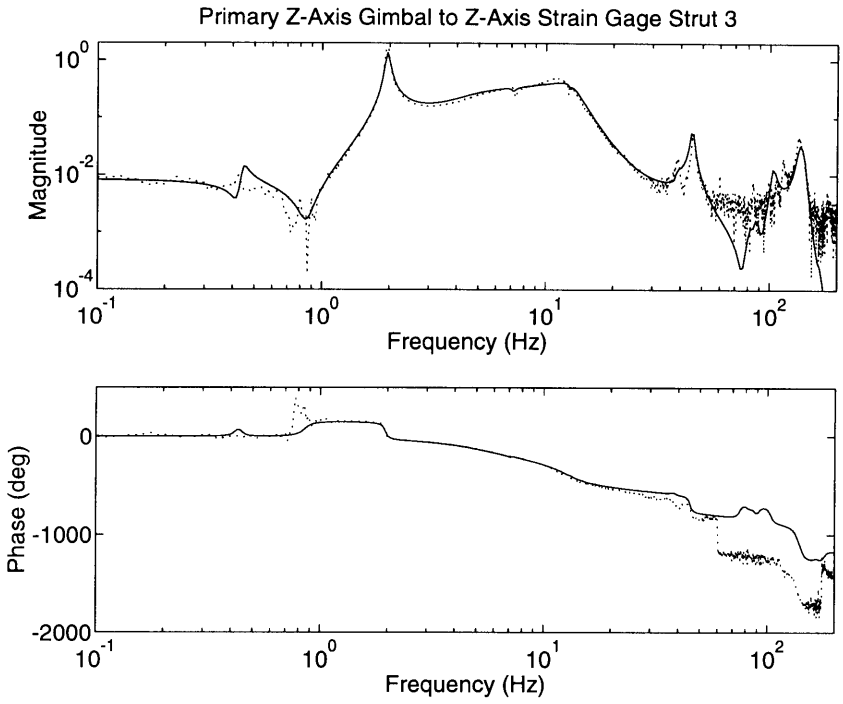

Z-Axis Reaction Wheel to Bus X-Axis Rate Gyro
Magnitude
10^0
10^{-5}
10^{-1}
10^0
10^1
10^2
Frequency
Phase
0
-1000
-2000
-3000
10^{-1}
10^0
10^1
10^2
Frequency

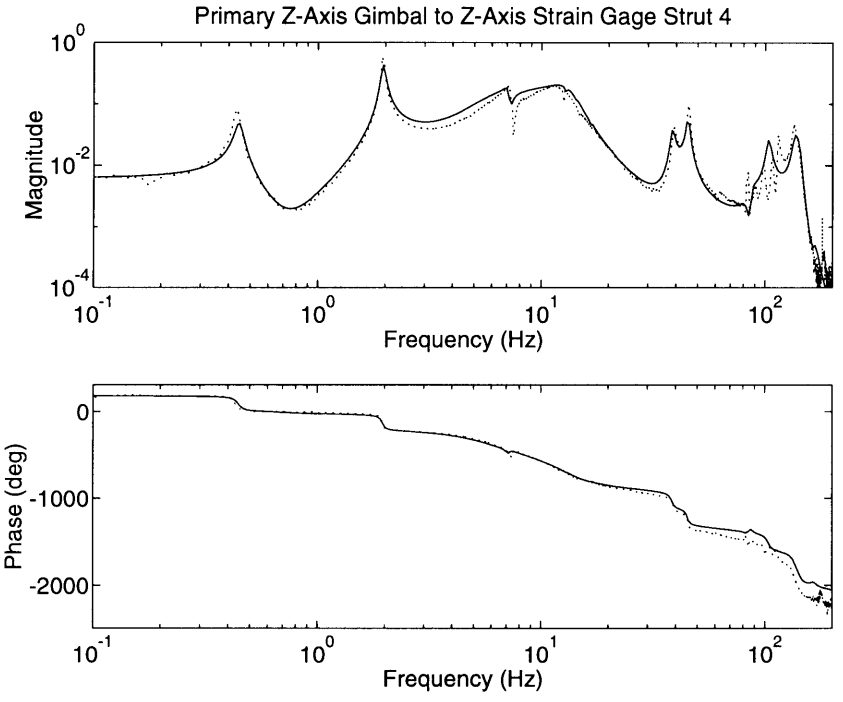

Z-Axis Reaction Wheel to Bus Y-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

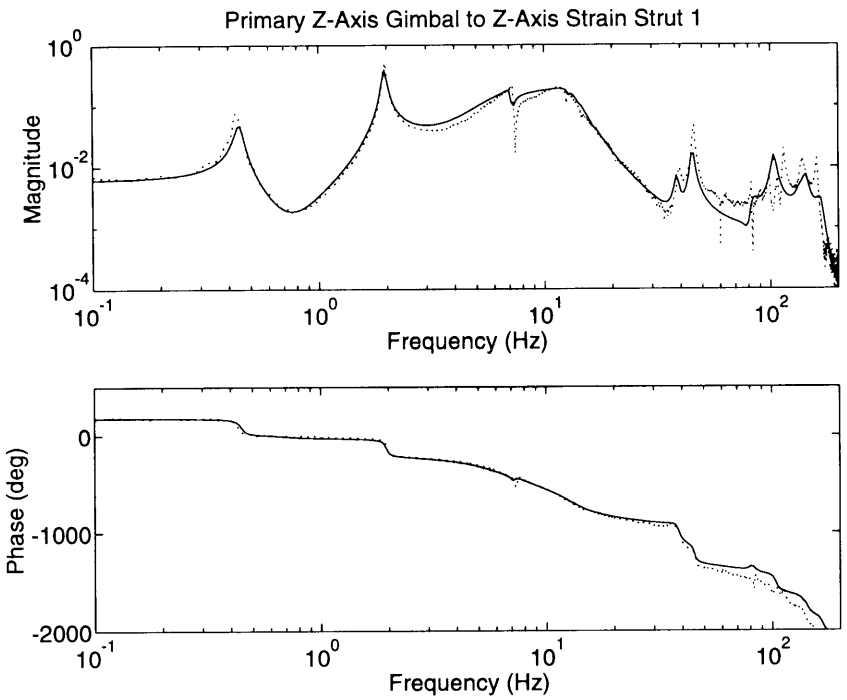

Z-Axis Reaction Wheel to Bus Z-Axis Rate Gyro
Magnitude
Frequency
Phase
Frequency

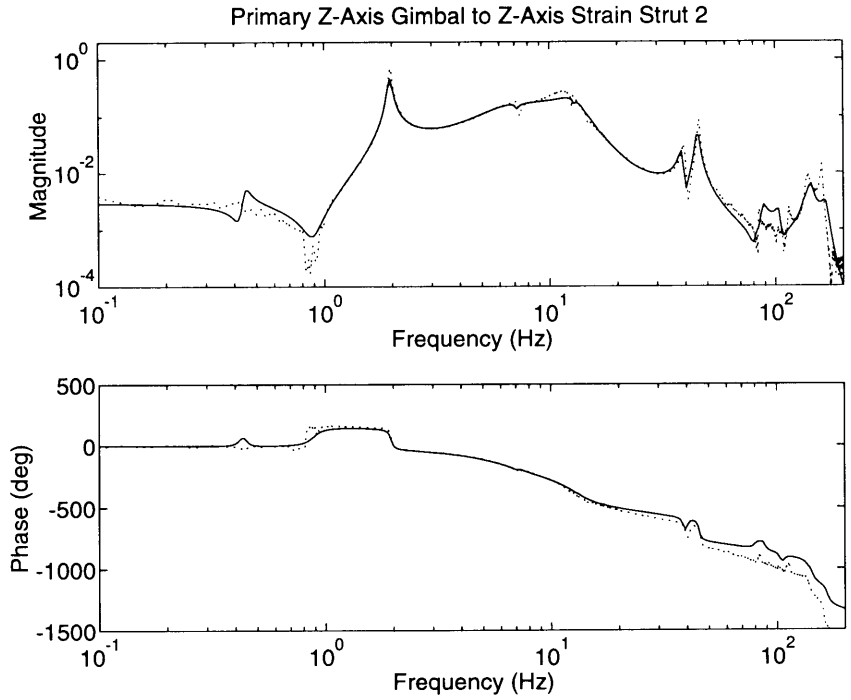

Primary Z-Axis Gimbal to Z-Axis Strain Gage Strut 1
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

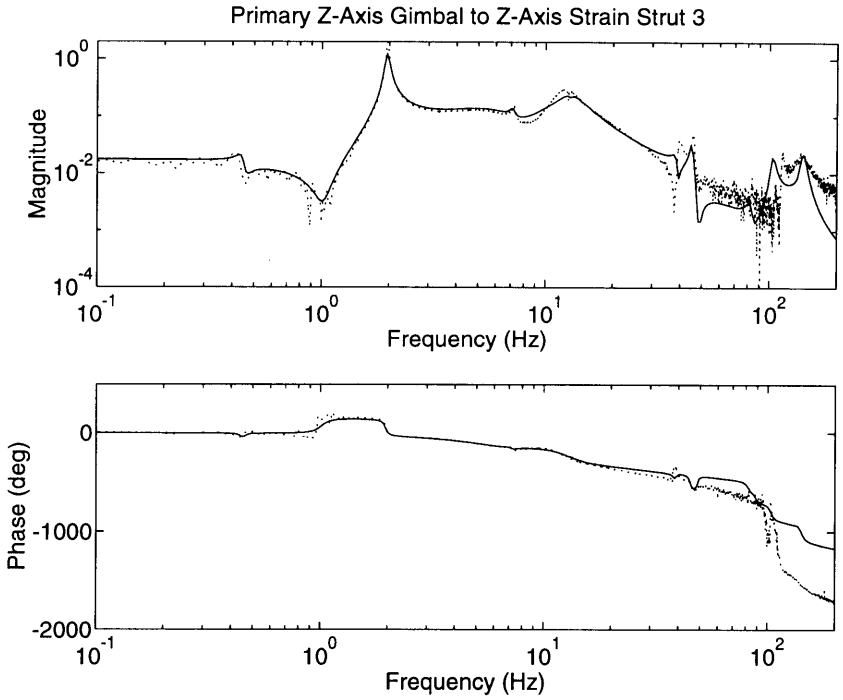

Primary Z-Axis Gimbal to Z-Axis Strain Gage Strut 2
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

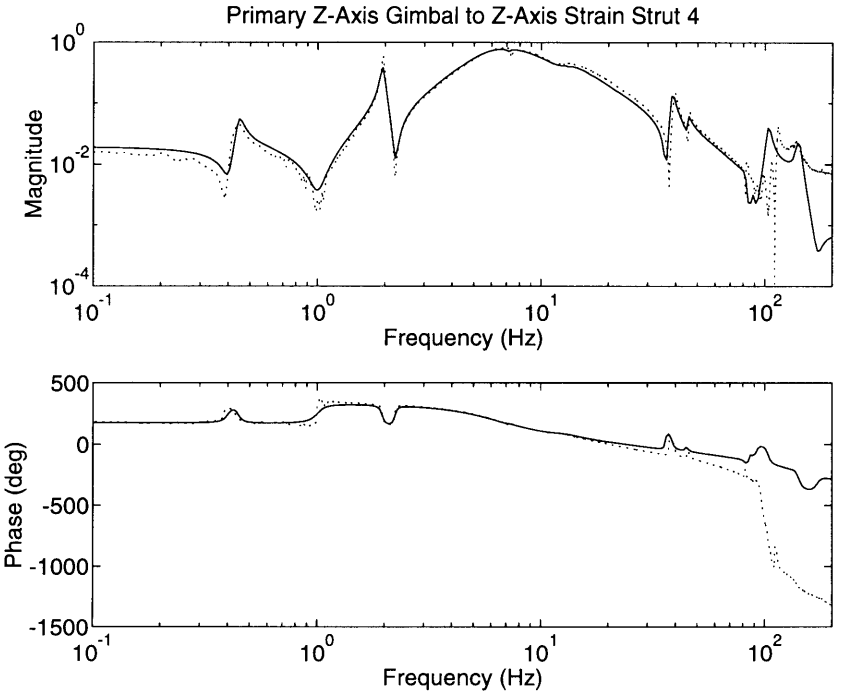

Primary Z-Axis Gimbal to Z-Axis Strain Gage Strut 3
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

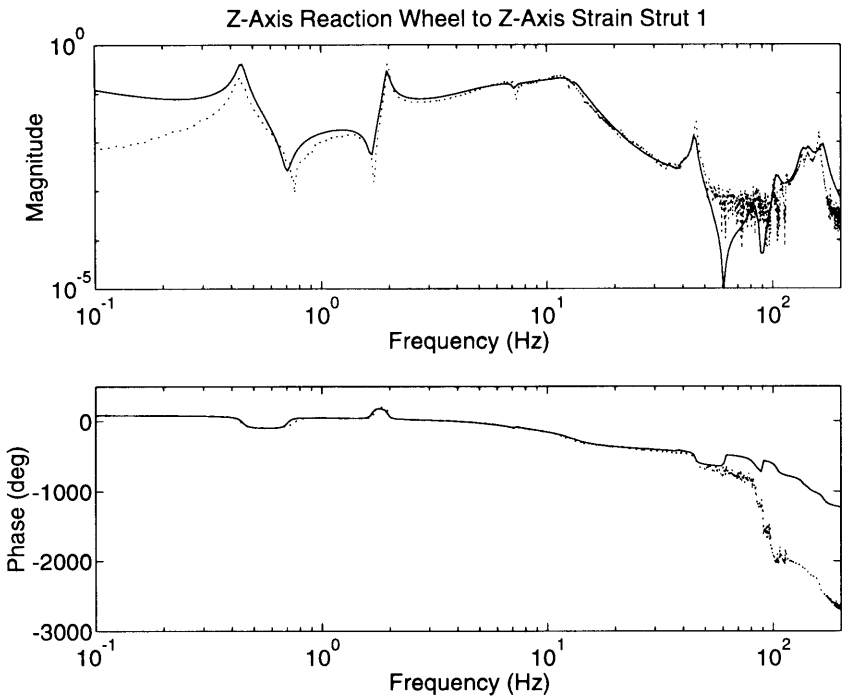

Primary Z-Axis Gimbal to Z-Axis Strain Gage Strut 4
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

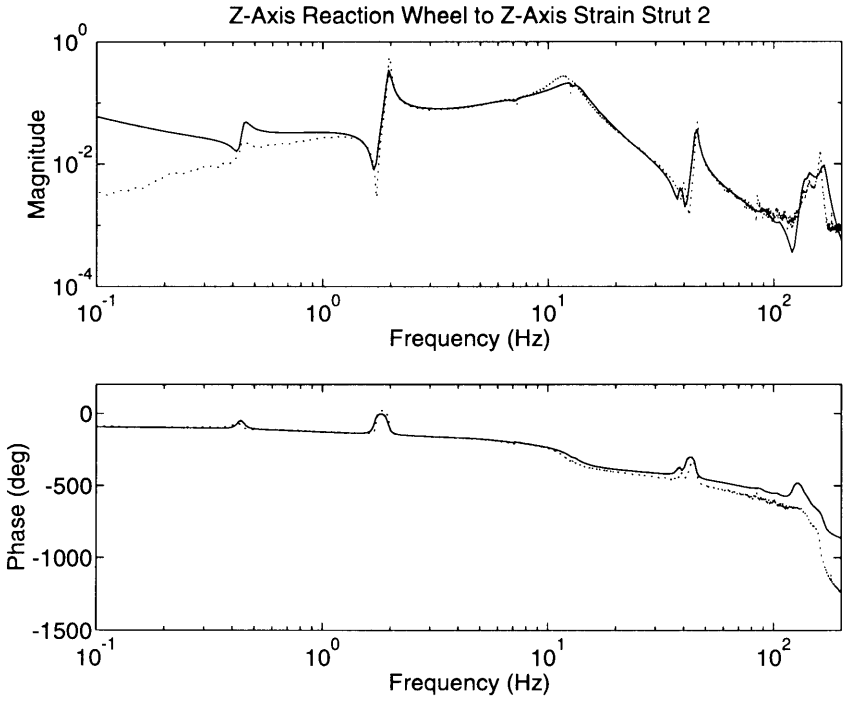

Primary Z-Axis Gimbal to Z-Axis Strain Strut 1
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

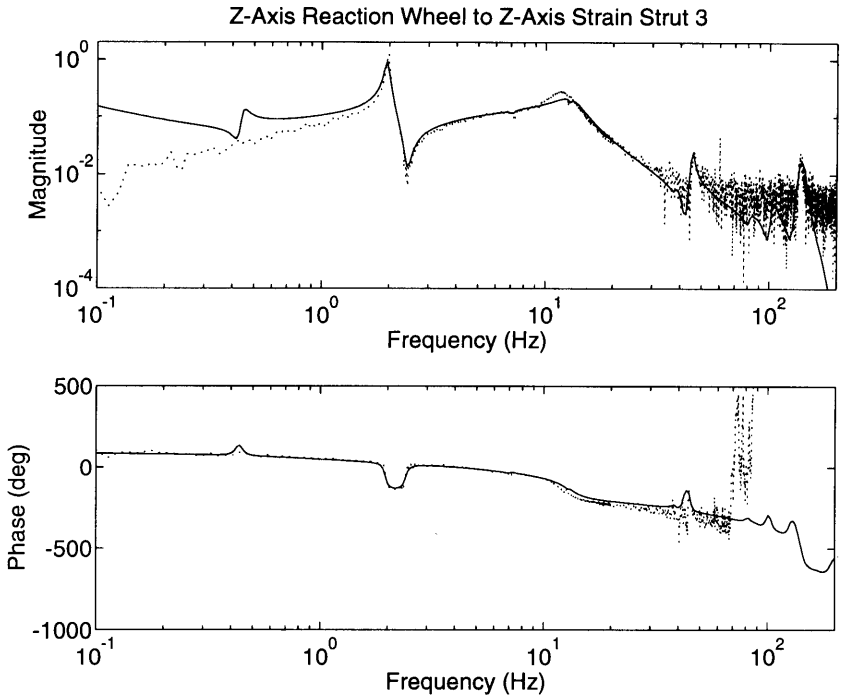

Primary Z-Axis Gimbal to Z-Axis Strain Strut 2
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

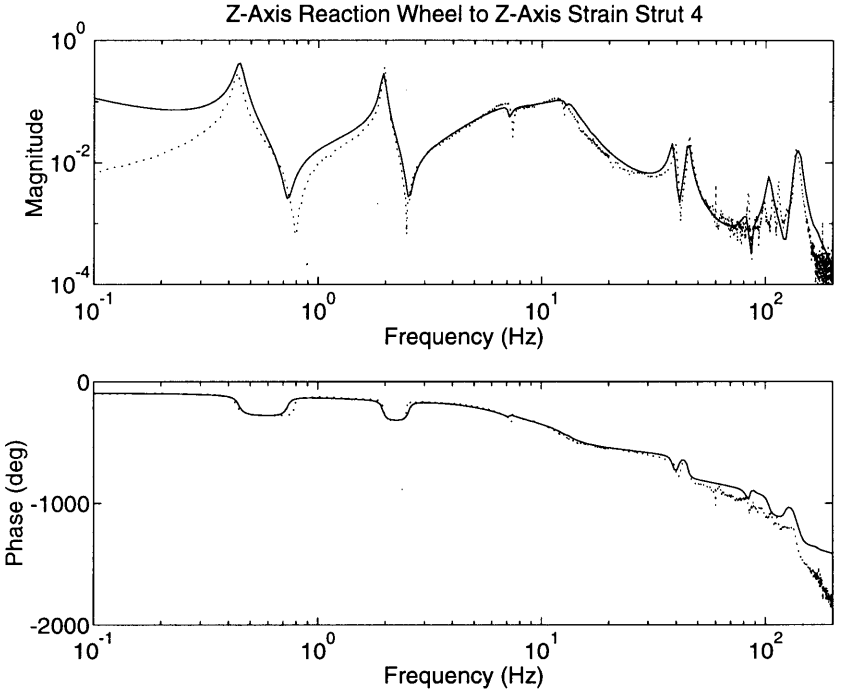

Primary Z-Axis Gimbal to Z-Axis Strain Strut 3
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)


Primary Z-Axis Gimbal to Z-Axis Strain Strut 4
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)


Z-Axis Reaction Wheel to Z-Axis Strain Strut 1
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)


Z-Axis Reaction Wheel to Z-Axis Strain Strut 2
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)


Z-Axis Reaction Wheel to Z-Axis Strain Strut 3
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)


Z-Axis Reaction Wheel to Z-Axis Strain Strut 4
Magnitude
Frequency (Hz)
Phase (deg)
Frequency (Hz)

3524-6