Model Reduction of a Set of Elastic, Nested Gimbals by Component Mode Selection Criteria and Static Correction Modes

by

Phillip James Balucan

B.S., Mechanical Engineering
University of California, Los Angeles, June 1999

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author .....
Department of Mechanical Engineering
May 23, 2001

Certified by .....
Timothy Barrows
Technical Staff
The Charles Stark Draper Laboratory, Inc.
Thesis Supervisor

Certified by .....
Samir Nayfeh
Associate Professor
Department of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Professor Ain A. Sonin
Chairman, Departmental Graduate Committee
Department of Mechanical Engineering

This page intentionally left blank.

Model Reduction of a Set of Elastic, Nested Gimbals by Component Mode Selection Criteria and Static Correction Modes

by

Phillip James Balucan

Submitted to the Department of Mechanical Engineering
on May 23, 2001, in partial fuifillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Model reduction techniques provide for a computationally inexpensive method for solving elastic dynamic problems with complex structures. The elastic nested gimbal problem is a problem which requires model reduction techniques as a means to reduce the dynamic equations. This is done using two methods: one technique employs mode ranking criteria to select modes which influence the dynamics of the problem the most. The second involves the use of static correction modes along with vibration modes to simulate the dynamics of this nested gimbal model.

A model of the structure is described in terms of a lumped-parameter finite element model. This mathematical model of the physical system serves as the basis for developing model reduction techniques for the nested gimbal problem. A truth model based on given initial conditions is used to compare the accuracy of the model reduced problem.

A number of model reduction theories are described and applied to the gimbal simulation. The equations for the mode ranking techniques and the static and vibration mode analysis are developed as well as a quantitative error measure. Comparisons are made with the truth model using the mode ranking criteria base on the momentum coefficients and the frequency cutoff criteria. Test cases are also run using the static correction modes with vibration modes and static correction modes with the ranked vibration modes using momentum coefficients. The use of various static modes is discussed during the implementation of the static correction mode method.

Applying the model reduction theories to a set of elastic, nested gimbals, the mode ranking criteria provides better results based on the error measure than the frequency cutoff criteria when the simulation is run using less than twenty-five modes. Using static modes along with ranked modes to represent the elastic dynamics of the problem does not provide better results than using the unranked vibration modes with the static modes. Modeling the dynamics using static correction modes with the unranked vibration modes provides the best results while using the least number of modes. It is advantageous to take into account the given conditions applied to the system when reducing the model of a complex dynamic problem.

Thesis Supervisor: Timothy Barrows
Title: Technical Staff
The Charles Stark Draper Laboratory, Inc.

Thesis Supervisor: Samir Nayfeh
Title: Associate Professor
Department of Mechanical Engineering

This page intentionally left blank.

Acknowledgments

I would like to thank all those who helped make this thesis possible. To the Charles Stark Draper Laboratory, for giving me the opportunity to work in an exciting and productive work environment. To Bernard Antkowiak, for involving me in this project and for developing the finite element model using MSC/NASTRAN. I would like to especially thank my advisor, Timothy Barrows, for the guidance and inspiration to study dynamics. To my office mate, Gary Hall, who knows everything from non-linear controls to donuts. To Professor Samir Nayfeh, for taking the time to review my thesis. I would also like to thank all my friends from coast to coast and around the world who have influenced me in every way possible. You all know who you are. I would like to give a special thanks to my relatives and family. To Mom, Dad, Joey, Jen, Sparky, and Prince, you all have helped me become what I am today and this thesis is as much yours as it is mine. Lastly, I would like to thank the Lord, Jesus Christ, who makes all things possible.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under contract N00030-99-C-0018 CLIN5 sponsored by SP23.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained therein. It is published for the exchange and stimulation of ideas.

(author's signature)

This page intentionally left blank.

Contents

List of Figures

This page intentionally left blank.

List of Tables

This page intentionally left blank.

Glossary of Terms

$A_{\bar{n}\bar{n}}$	Diagonal matrix of the eigenvalues of the kept modes $\bar{n}$
B	Number of Bodies
D	Number of Modes
E	Error over a time interval
$e(t)$	Error at a specific time t
$F(r,\epsilon)$	Flexibility kernal
f_{con}	Control force vector
$f_{c,r}$	Rigid body control force vector applied to rigid body coordinates
f_{ext}	Externally applied force vector
f_{non}	Nonlinear force vector
G	Inverse of stiffness matrix, K
H_n	Angular momentum coeffecients of body n
I_3	3×3 identity matrix
i	Mode number
$\mathbf{K}$	Global stiffness matrix used in equations of motion of nested gimbals
K	Finite element Stiffness Matrix
k	Node number
$\mathbf{M}$	Global mass matrix used in equations of motion of nested gimbals
M	Finite element Mass Matrix
$\bar{M}_i$	Mass property contribution of i^{th} mode
m_n	Total mass of body n
N	Total number of nodes in a single body
n	Body Number
P_n	Linear momentum coeffecients of body n
q	Elastic generalized coordinates
r_{nk}	Position vector of body n at node k
$r(t)$	Response of truth model
$\tilde{r}_{nk}$	Skew symmetric matrix of 3×1 position vector, r_{nk}
$\mathbf{U}$	Elastic Displacement Field
$\mathbf{u_k}$	Translational and rotational displacements of node k

v	Rigid body generalized coordinates of velocity
X_n	Rigid body mode for body n
y	Vector of generalized coordinates
$\dot{\gamma}_n$	Angular velocity of joint which connects body n to $n+1$
γ_n	Angle of joint which connects body n to $n+1$
$\Lambda_{\bar{n}}$	Diagonal matrix of eigenvalues corresponding to the kept modes, $\bar{n}$
Ω_i	Natural frequency corresponding to the i^{th} mode
$\Psi_n(r_{nk})$	nodal coordinate of the k^{th} node on body n (contains all modes)
$\Phi_{\bar{n}}$	Matrix containing the kept eigenvectors, $\bar{n}$
ϕ_{ni}	i^{th} eigenvector of body n
$\dot{()}$	First time derivative
$\ddot{()}$	Second time derivative

Chapter 1

Introduction

1.1 Overview

Mathematical simulations are useful for studying the dynamics and control of physical systems before the creation of the actual structure. Simulating the dynamics of a flexible body involves the computation of a set of admissible finite-element shape functions or assumed modes associated with the mathematical representation of the structure. The more complex a structure, the larger the number of modes that are used to describe its elastic characteristics. When many of these flexible bodies are linked together in a mechanism the total number of modes for the whole system becomes increasingly large. With the increase in the number of modes used to simulate the multibody structure the number of generalized coordinates associated with elastic deformations increases. The solution time to solve the system of equations becomes quite long. If the number of modes can be reduced by selecting the most influential ones or by using different types of modes then the solution time can be decreased while maintaining reasonable accuracy.

There exist different methods for the model reduction of large complex structures. These include the use of mode rankings to rank the modes according to their significance to the dynamics of the problem. The use of these mode rankings are developed from the modal identities used in Hughes [4]. Another method involves the use of static correction modes. These are special modes which simulate the local deformation at the joints of an articulated structure. An articulated structure is defined by Yoo and Huag [11] 'as one in which a large relative displacements occur between points in the structure, even though small displacement, linear elastic theory is adequate to describe deformation of individual components.' The set of nested gimbals is such a structure and is studied in this paper.

Gimbals are used in Inertial Measurement Units (IMU), which are instruments for the guidance and navigation of flying structures. The purpose of the gimbals is to keep the innermost platform within the IMU stable with respect to an inertial reference frame. A multibody gimbal simulation is used to represent the dynamics of the mathematical model. Both rigid and elastic dynamics are taken into account. It is the elastic component of the dynamics that results in a large system of equations to be solved. A number of different techniques have been developed to decrease the number of generalized coordinates, some of which are described in this thesis.

1.2 Previous Work

Various methods of model reduction have been developed over the years. One of the first is the component-mode synthesis method developed by Hurty [6]. Another involves substructuring techniques by Craig and Bampton in [2]. This method involves the process of idealizing the structure as an assemblage of discrete structural elements, or substructures. This is similar to what is done in the gimbal simulation in which each gimbal is thought of as a separate structure and analyzed as such. Still another method using component cost analysis [9] involves the computation of constraint forces, which for this gimbal simulation would require additional computations. The two methods applied here to the gimbal simulation involve the use of component-mode selection criteria which has its basis in the component mode synthesis method in [6] and the use of static correction modes with vibration modes to model the dynamics of the system.

The use of mode rankings to reduce the system involves the selection of modes based on a measuring criterion to reduce the number of generalized coordinates. Modes are sometimes selected according to frequency cutoff criterion which offers no insight to the dynamics of the problem. This is not always the best method and a mode's influence on the mass properties of a body is applied to the nested gimbal model. The ranking criteria are based on modal identities found in [4].

Vibration and static modes to represent the dynamics of the system have been used by Yoo and Huag in [11] and [13] to represent the dynamics of the system . This process involves the use static modes in addition to vibration modes to represent the dynamics of the system. The static modes are used to represent the local deformation incurred at the joints due to the motion of a linked structure. The high frequency modes of the structure are represented by these static modes. This method was applied to a wiper-blade mechanism in [13] and the same method is applied to the set of elastic, nested gimbals.

1.3 Thesis Scope

The research described in this paper involves three parts. The first is the creation of modal integrals used in the dynamic system of equations of elastic nested gimbals. The gimbal model used in this study is created in a finite-element program and is represented by three rings and one stable platform and an outer case. The equations of motion are shown using nonlinear rigid dynamics and linear-elastic deformation theory. The normal modes of each of the five bodies are found using a finite-element program, the results of which are used in a computer simulation that solves the equations of motion. It is from these modes that the momentum coefficients are computed. From the size of these matrices, it is realized that model reduction of the system is advantageous to this problem. A truth model is formed using normal modes in which the reduced order results will be compared.

The second part involves the theories and equations used in the model reduction techniques. Both the mode selection criteria and the use of static modes will be discussed in chapter 3. The criteria developed in this chapter is then applied to the gimbal model. Finally, results from the reduced order model are compared with the truth model. Error criterion are used to discuss the accuracy of the results and comments are made about the computation time of the truth model versus the reduced order models.

This page intentionally left blank.

Chapter 2

Dynamics of Nested Gimbals

2.1 Overview

To determine the equations of motion from the mathematical model of the elastic nested gimbals, it is necessary to first describe the properties and configuration of the model. From this model the equations of motion can be created. The general set of equations for an arbitrary multibody system have been derived in various papers such as Sincarsin and Hughes [8] [5] and Storch [10]. The detailed derivation of these equations are beyond the scope of this thesis. However, certain aspects of the equations are discussed here to allow for a better understanding of the model reduction techniques of the set of gimbals. The general set of equations for a multibody system from [8], [5] and [10] are applied to the nested gimbals system. It will prove useful to describe these equations which illustrate the complication of the problem using a large number of modes.

The simulation developed from these equations models a base body with a prescribed motion. Attached to the base body is a set of gimbals that have motors attached to each drive axis. Each gimbal has one degree of freedom to rotate about its drive axis. Initial torques, initial angles and angular rates can be applied to each of the bodies within the base body. From these conditions the motion of each of the structures is defined.

This chapter describes a set of elastic, nested gimbals. The equations of motion are then developed and finally the solution is compared to a similar simulation developed by LMS International to determine the validity of the dynamic equations of the mathematical model.

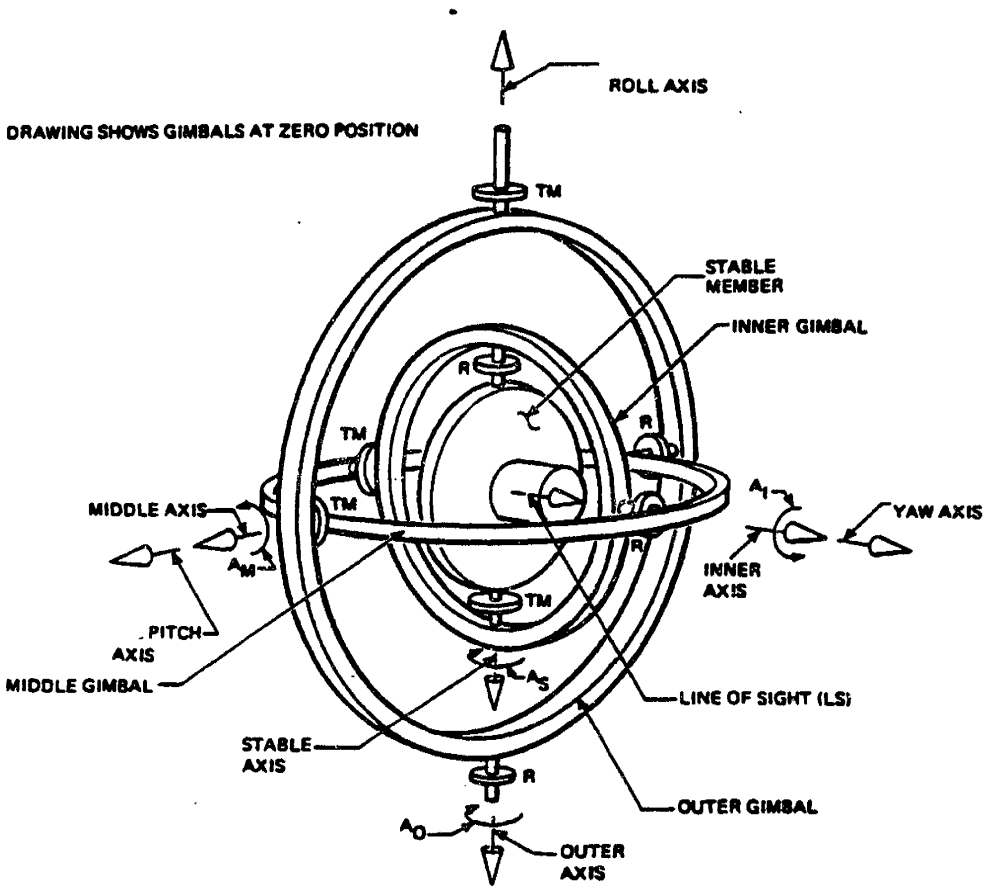


Figure 2-1: Configuration of a Set of Nested Gimbals.

2.2 Model

It is important to first define the configuration of the set of nested gimbals. A set of nested gimbals is shown in Figure 2-1. The assembled configuration of the model at zero position is shown in this figure. The term R defines the location of the resolver, while the location of the term TM shows at which end the torque motor is situated.

Figure 2-1 reveals the orientation of each of the gimbals and stable platform with respect to one other. Each gimbal for this problem is modeled as a thin ring in the finite element analysis. The stable platform is modeled as a circular plate. The finite element mesh for each gimbal and the stable platform is shown in Figures 2-2 through 2-5. It should be noted that the relative size of each body in the figures has been maintained so that the stable platform fits in the inner gimbal

	Diameter [in.]	Distance Between Outer Bearing Points [in.]	Distance Between Inner Bearing Points [in.]
Outer Gimbal	6.75	7.25	6.25
Middle Gimbal	5.75	6.25	5.25
Inner Gimbal	4.75	5.25	3.87
Stable Platform	3.4(inner) 1.4(outer)	3.87	Not Applicable

Table 2.1: Dimensions of Finite Element Model of Gimbals

and so on. The dimensions of these rings in the undeformed state are found in Table 2.1. The thickness of the walls of the rings are .25 inches and the stable platform thickness is 1 inch. The bearing points are ones on the bodies where the gimbals are connected. Each gimbal has four bearings, two outer and two inner. The outer bearings are the points where the current gimbal is connected to the next outside gimbal and the inner bearings are the points where the current gimbal is connected to the next inside gimbal. The stable platform has only two bearing points because it is the innermost body. The outer bearings of the stable platform are points where it is connected to the inner bearing points of the inner gimbal. These bearing points are also associated with the joint locations of each of the bodies.

Also notice the local coordinate system of each of the bodies. The origin of the coordinate system lies in the center of the body and the motor and resolver lie on the x-axis of the local coordinate frame, while the resolver and motor of the next body inward are situated on the z-axis of the current local coordinate system. The origins of the coordinate system in the center of the body is called the gimbal point. For the three gimbals and stable platform, each gimbal point coincides with the center of the nested gimbal structure. In other words, there is no offset between the origins of the local coordinate systems of the gimbals and stable platform. However, the base does not have its coordinate system located at the gimbal point. Its local coordinate system is offset from the gimbal point of the outer gimbal and the others. This offset, defined as the vector ρ in terms of the base local coordinate system, defines the location of the gimbal point with respect to the base coordinate frame. For this gimbal problem, the gimbal point is located -3.3750 inches in the z-direction from the origin of the base coordinate frame. This offset is defined in Figure 2-6.

At this juncture it is beneficial to define terms used throughout this paper: the inside and outside frames and the input and output motors and resolvers. The outside frame is defined as the frame connecting the two outside bearing points and the inside frame is defined as the frame connecting the inside bearing points. Elastic displacements are defined with respect to the outside

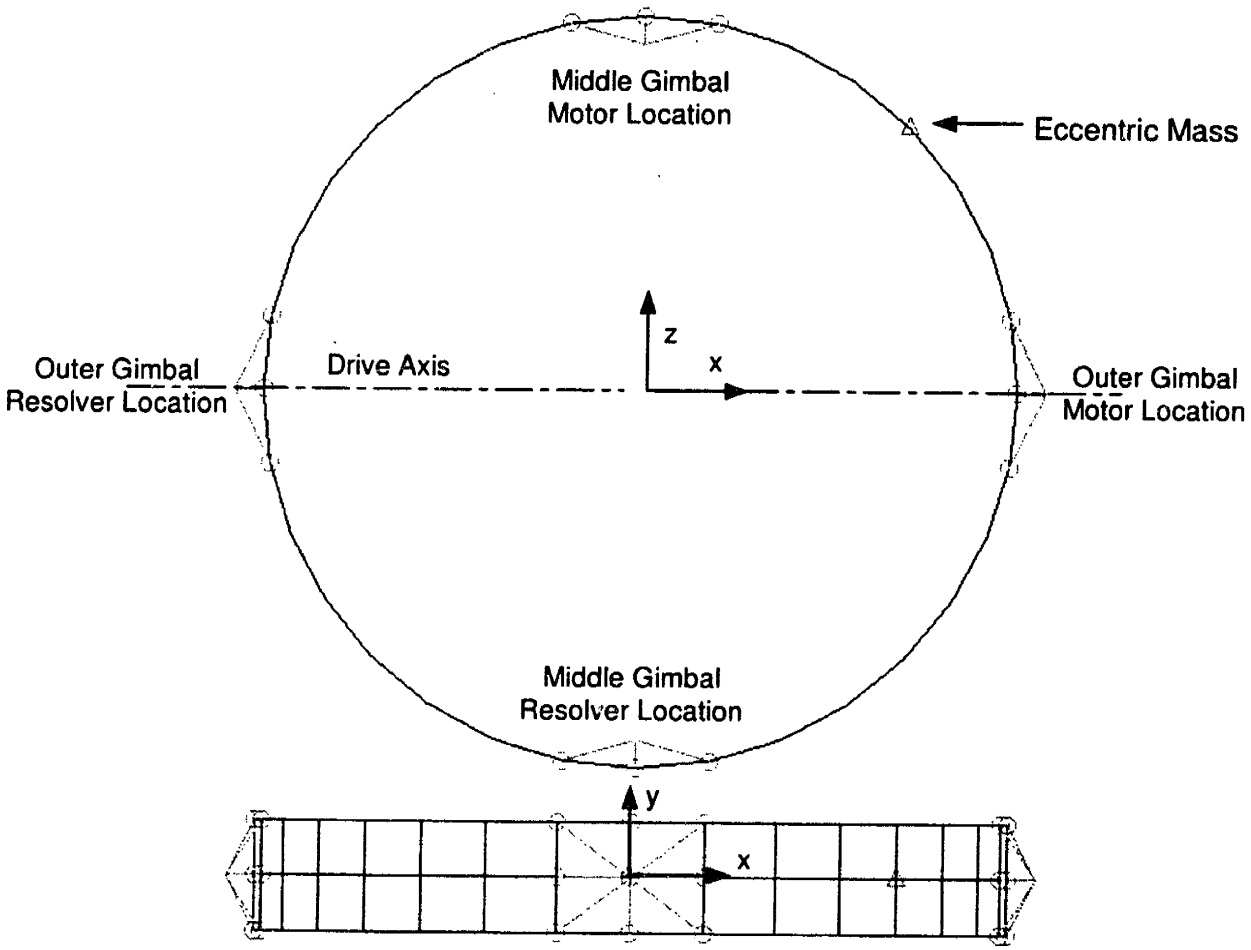


Figure 2-2: Outer Gimbal FEM model

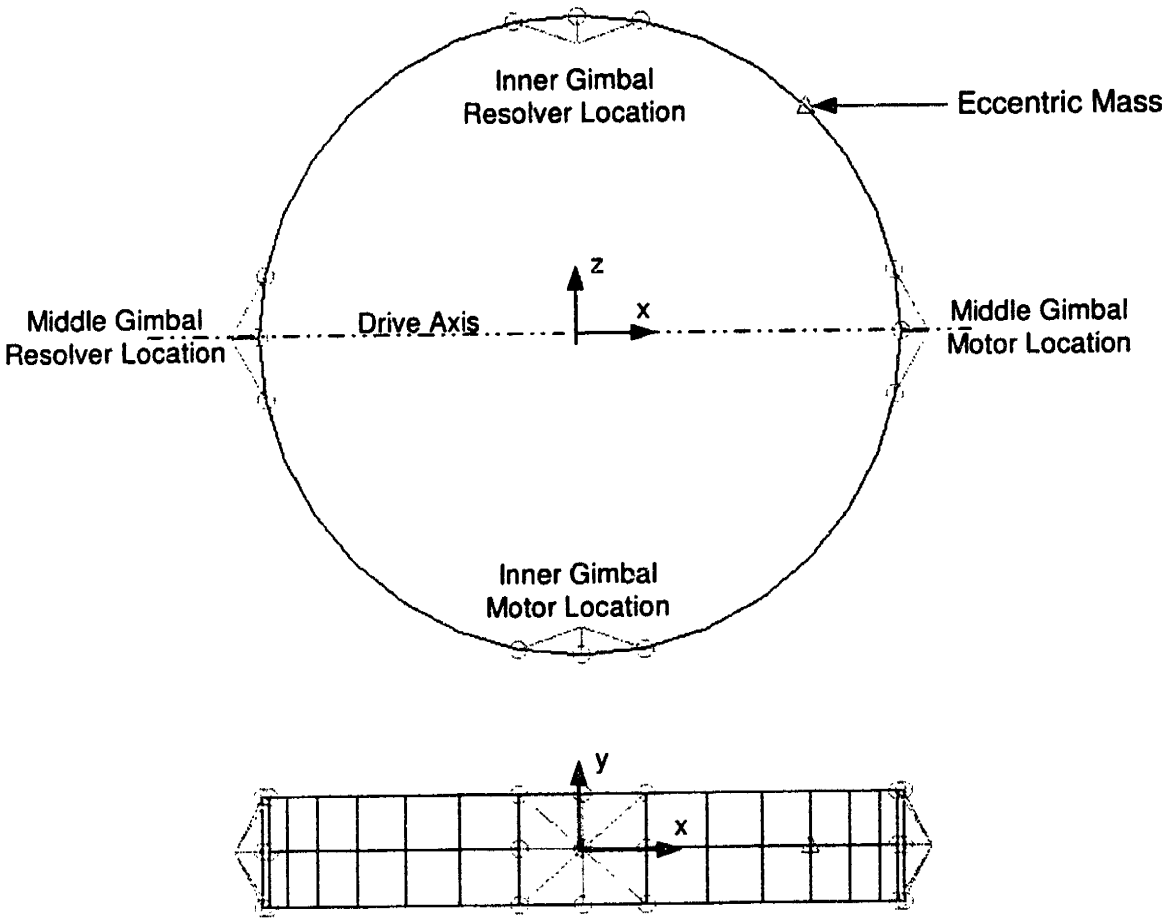


Figure 2-3: Middle Gimbal FEM model

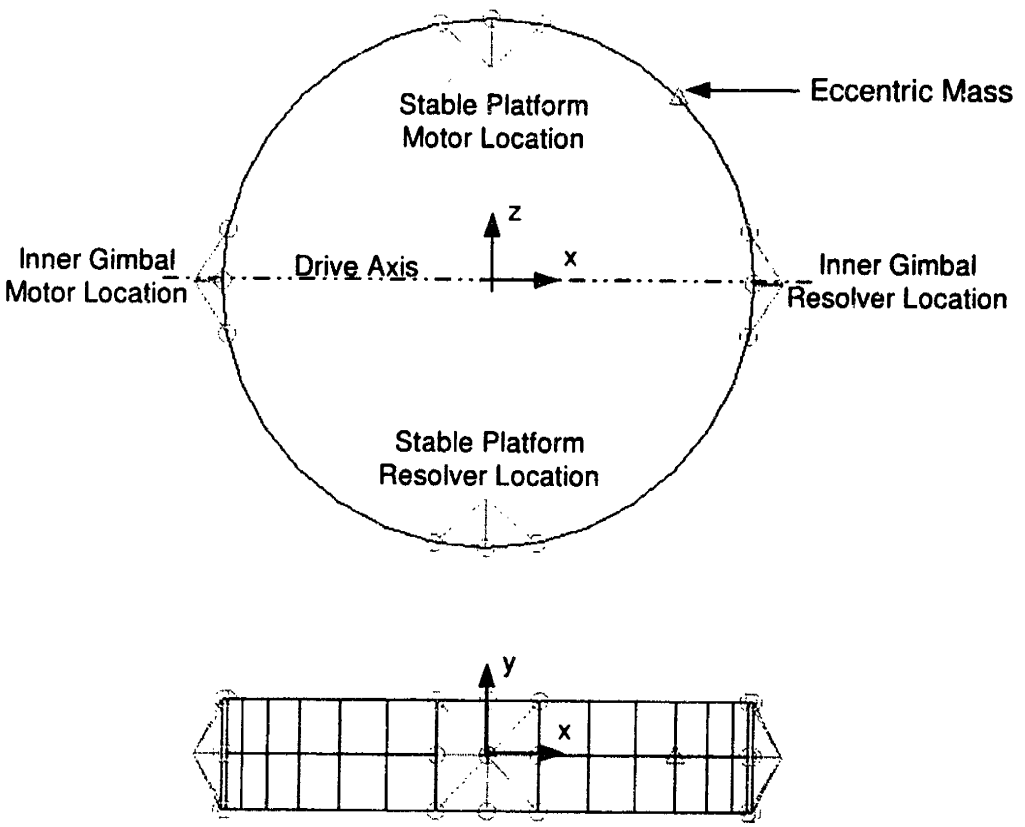


Figure 2-4: Inner Gimbal FEM model

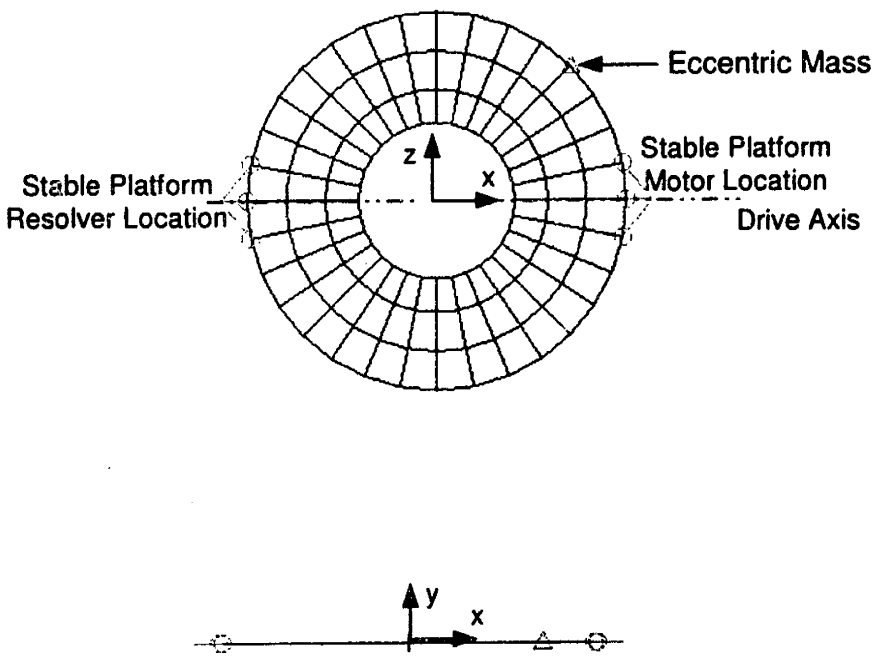


Figure 2-5: Stable Platform FEM model

	Body Number	Motor Location	Resolver Location
Outer Gimbal	2	Positive x-axis	Negative x-axis
Middle Gimbal	3	Positive x-axis	Negative x-axis
Inner Gimbal	4	Negative x-axis	Positive x-axis
Stable Platform	5	Positive x-axis	Negative x-axis

Table 2.2: Motor and Resolver Location of Each Body in Local Coordinate System

frame. The origins of these frames are located at the gimbal point in the undistorted case. For zero deformation these frames coincide with one another, but when the body is deformed the outside frame differs from the inside. The torque from the input motor is applied to the outside frame, while the inside frame reflects the reaction forces due to the next inward body.

The definition of the input and output motor can be seen in Figure 2-7. Although this figure only shows the outer gimbal the terms defined herein are used for the other gimbals as well. The input motor is defined as the motor associated with the outer bearing point. The input resolver is the location of the resolver of the associated input motor. Similarly, the output motor is the motor of the next inner gimbal. For the outer gimbal this motor would be associated with the middle gimbal. The output resolver is the resolver which measures the angle of the middle gimbal with respect to the outer gimbal.

In the 'all angles zero' configuration of the system, the orientation of the local axes with respect to each other are shown in Figure 2-8. The X, Y, and Z frame are the inertial frame. The motor and resolver position are either in the positive x coordinate or negative x coordinate in the body frame. Table 2.2 list the location of the motor and resolver with respect to the local coordinate system. These locations are used when defining the boundary conditions of the finite element model.

The gimbals are modeled as rings composed of quadrilateral plate elements in the finite element model. There are one hundred nodes used to define the structure. For the stable platform the structure is composed of one hundred and thirty nodes.

There are five bodies with the first being the case or base body. The motion of the case is prescribed. All the bodies are labeled from one to five, with the base body being body one and the stable platform being body five. Between these bodies there are the outer gimbal (body 2), the middle gimbal (body 3), and the inner gimbal (body 4). The base body is the outermost structure while the stable platform is the innermost. Each body in the system has the material properties defined in Table 2.3. These properties are chosen to represent the actual physical stiffness properties of the gimbals. There is an eccentric mass equal to ten percent of the remaining body

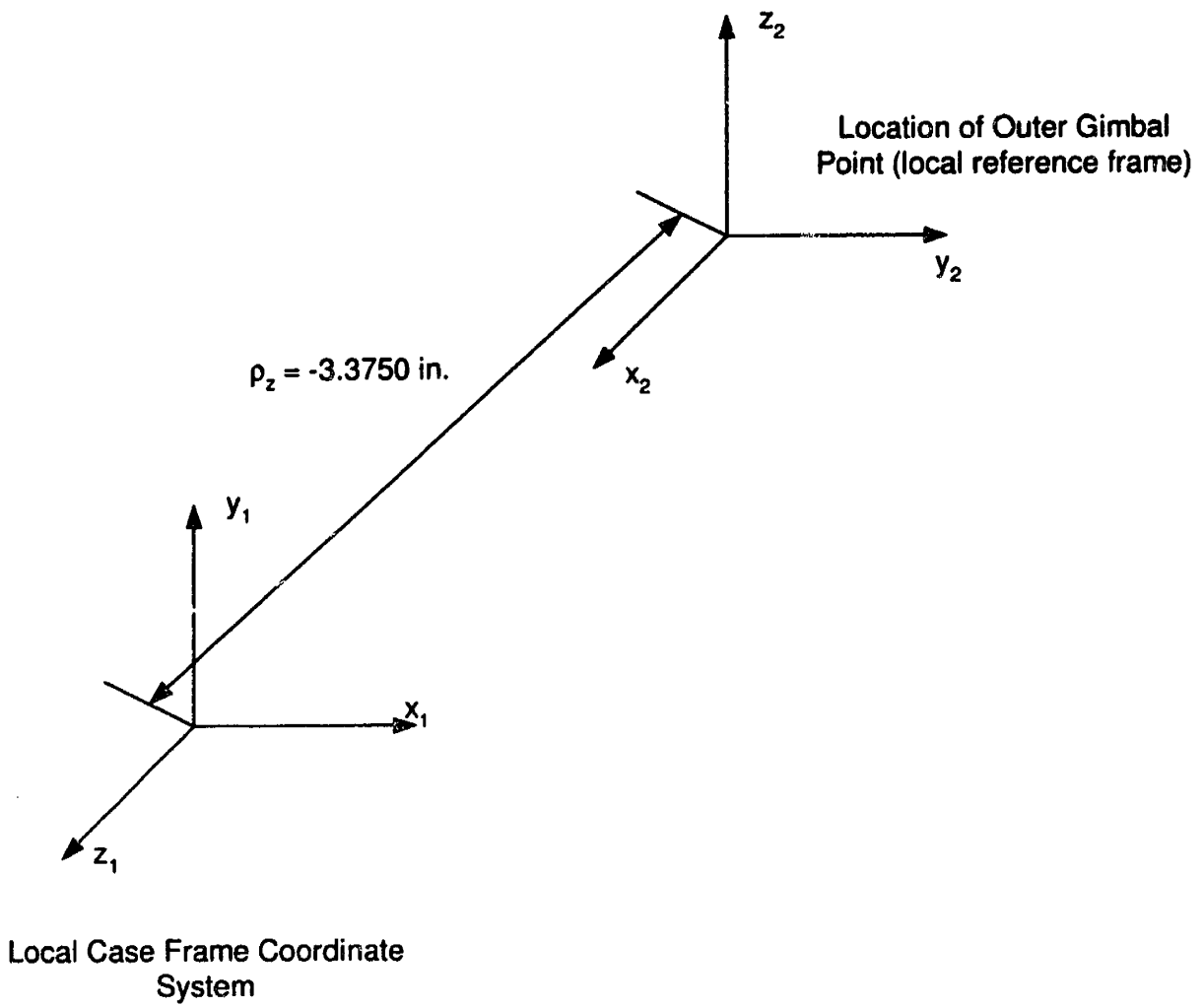


Figure 2-6: Offset of Gimbal Point from Case Frame

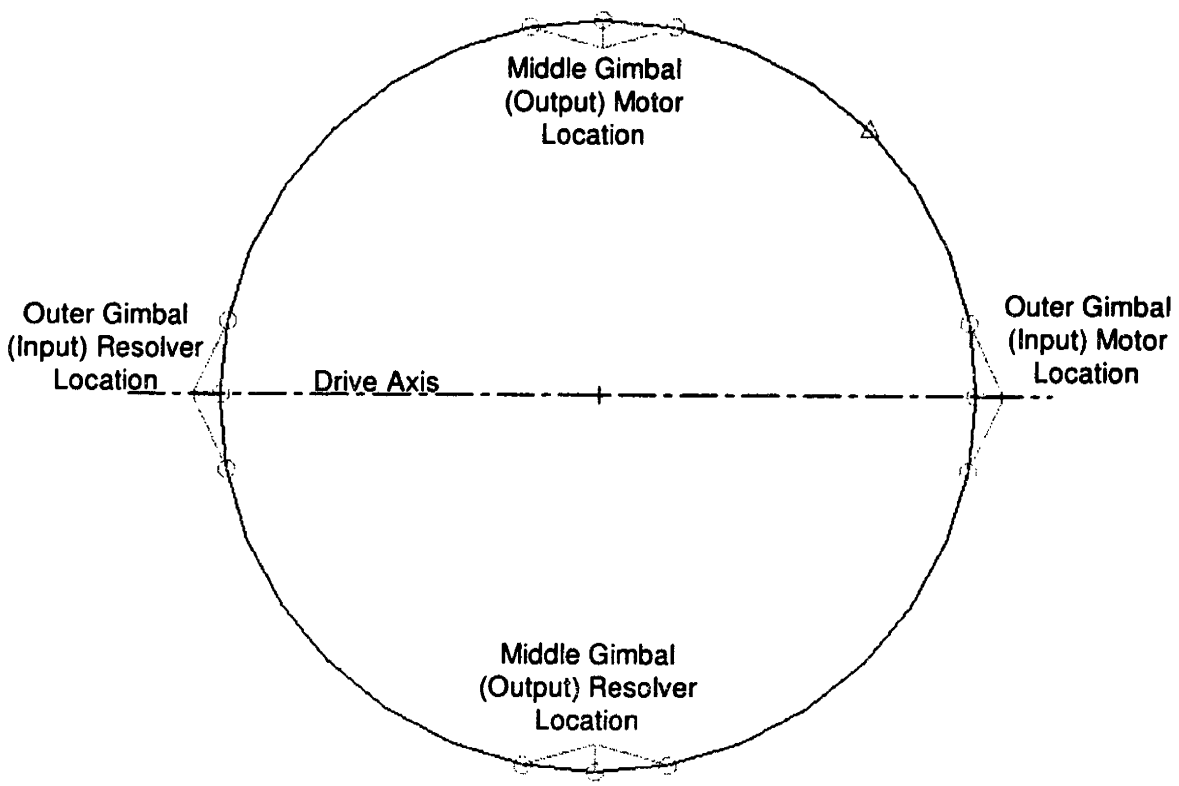


Figure 2-7: Outer Gimbal with Input and Output Motor and Resolver Location

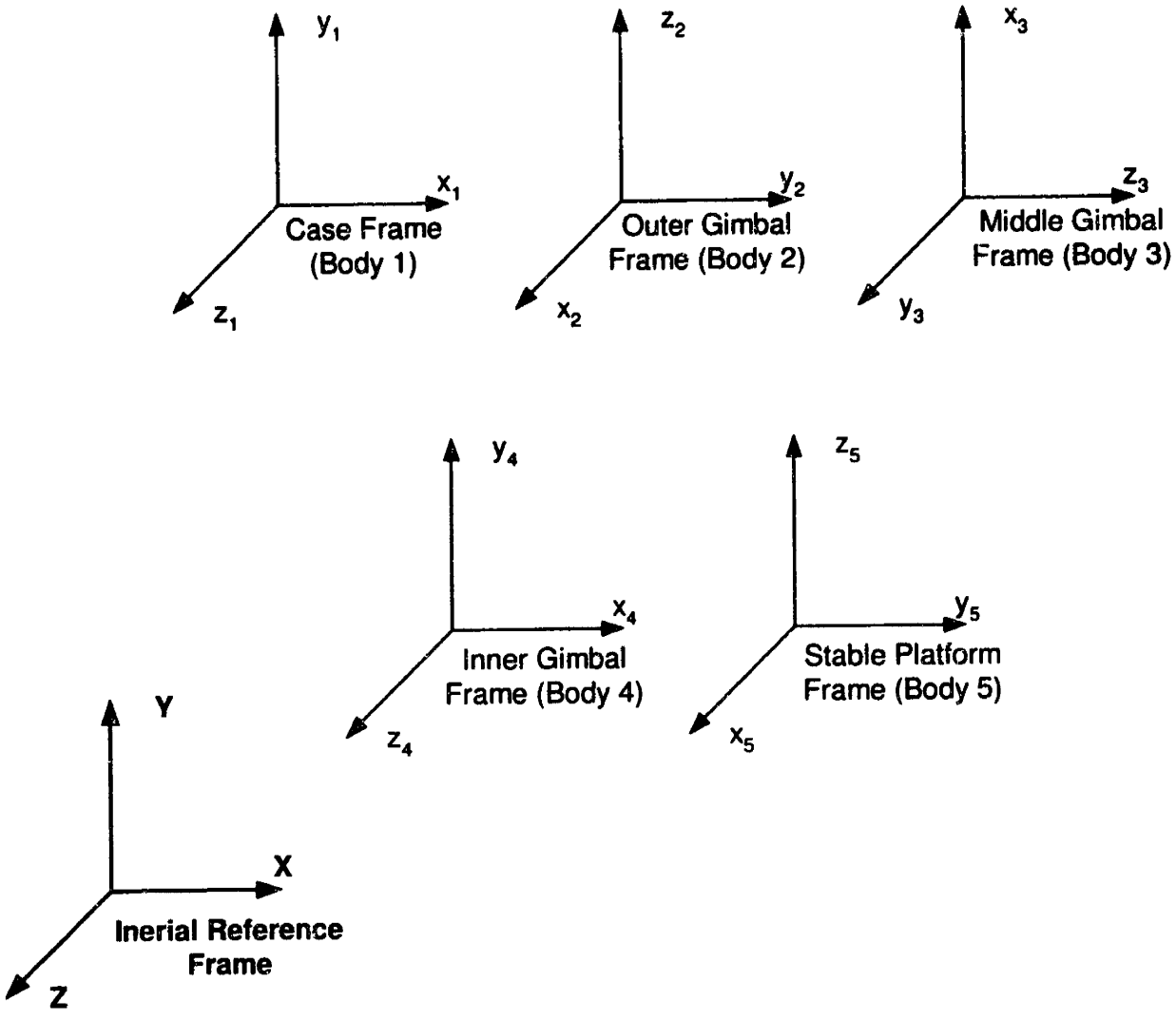


Figure 2-8: Configuration of Local Coordinate Frames with Respect to Inertial Frame.

Young's Modulus, E	$3.00 \times 10^8 \frac{\text{lbs}}{\text{in}^2}$
Poisson's Ratio, ν	.3
Density, ρ	$.0026 \frac{\text{slugs}}{\text{in}^3}$

Table 2.3: Material Properties of the Bodies

mass applied to point 45 degrees from the axis connecting two bearing points located in the positive x and z quadrant in the local coordinate frame as represented in Figures 2-2 through 2-5. The eigenvectors and frequencies of each of the bodies were determined using the finite element program MSC/Nastran.

The gimbal model includes a torque motor on one end of the drive axis. The drive axis is the axis in which the gimbals can rotate and is associated with the outside frame. In all, there are four drive axes. Opposite the motor on each drive axis is a resolver. The boundary conditions used in the finite element code to obtain the eigenvalues and eigenvectors are applied at the outside bearing points. On the motor side, the body is free to translate along the x-axis and rotate about the y-axis and z-axis but constrained to rotate about the x-axis and translate along the y-axis and z-axis. At the resolver, the body is free in all rotational degrees of freedom but constrained in the translational coordinates. From these conditions the body is constrained in six rigid-body degrees of freedom.

Each body has one hundred of the lowest frequency modes calculated from the finite element model. For a full solution using all available modes, there are four hundred elastic coordinates plus four rigid body coordinates of gimbal angles. The equations used to describe this motion are described in the next section.

2.3 Equations of Motion

The derivation of the general full set of equations of motion of an arbitrary chain of structures can be found in [10] and [8]. The full set of matrix equations from these references are defined in equation 2.1. These equations are formulated using a basic Newton-Euler approach supplemented with a finite element treatment of the elasticity [10]. These general equations will be simplified when applied to the nested gimbal model due to the inherent structure of the problem.

$$\mathbf{M}\ddot{y} + \mathbf{K}y = f_{\text{con}} + f_{\text{ext}} + f_{\text{non}} \tag{2.1}$$

$\mathbf{M}$ is the global mass matrix composed of the terms in equation 2.2, where M_{rr} can be thought of as the rigid body element of the mass matrix, M_{re} as the coupling terms between the rigid body coordinates and the elastic body coordinates, and M_{ee} as the elastic mass matrix elements. Likewise, $\mathbf{K}$ is the global stiffness matrix in equation 2.3 applied to the elastic coordinates on the equations. The conservative force vector f_{con}, the external force vector f_{ext}, and the nonlinear force vector f_{non} can be found in equation 2.4. The vector of generalized coordinates is described by equation 2.5. The vector $\dot{v}$ has components representing the rigid body generalized coordinates while $\ddot{q}_{\text{e}}$ are the elastic coordinates. The rigid body coordinates used in the gimbal simulation are the gimbal angles, γ_n, of each of the bodies, thus $\dot{v}$ is a 4×1 vector for a full model describing the angular acceleration about the drive axes (x-axes). In general, γ_n is the angle of the joint which connects body n to n+1. Thus if the case, body 1, is fixed, γ_1 is the angle of body 2.

$$\mathbf{M} = \begin{pmatrix} M_{\text{rr}} & M_{\text{re}} \\ M_{\text{re}}^T & M_{\text{ee}} \end{pmatrix} \tag{2.2}$$

$$\mathbf{K} = \begin{pmatrix} 0 & 0 \\ 0 & K_{\Sigma,\text{ee}} \end{pmatrix} \tag{2.3}$$

$$f_{\text{con}} = \begin{pmatrix} f_{\text{c,r}} \\ 0 \end{pmatrix}, f_{\text{ext}} = \begin{pmatrix} f_{\text{ext,r}} \\ f_{\text{ext,e}} \end{pmatrix}, f_{\text{non}} = \begin{pmatrix} f_{\text{non,r}} \\ f_{\text{non,e}} \end{pmatrix} \tag{2.4}$$

$$\ddot{y} = \begin{pmatrix} \dot{v} \\ \ddot{q}_{\text{e}} \end{pmatrix} \tag{2.5}$$

These equations are for a set of general configuration bodies and general joints. These equations are simplified for the gimbal model. One simplification of equation 2.1 is based on the fact that there are no external forces acting upon the bodies. Thus, the external force vector, f_{ext} is equal to zero. From the generalized coordinate vector described in equation 2.5 it is possible to see that the size of the vector becomes increasingly large as the number of modes used to describe the elastic displacement field increases. Multiplying and taking inverses of the global matrices, $\mathbf{M}$ and $\mathbf{K}$, requires a large amount of computing power.

The individual matrix elements are themselves defined in [10] and applied to the gimbal model. Each of the matrix elements are composed of matrices defined specifically for each body n. At this point it is important to define the complete configuration of the nested gimbal problem as it relates

to the general multibody case using the system idealization in [10]. The gimbal model can also be thought of as a chain with B bodies. For the gimbal case B is equal to 5. Body 1 is the base body with six degrees of freedom (three rotational and three translational) relative to the inertial reference frame. There are two reference frames for each body except for the terminal body, Body B, in this case the stable platform. For the gimbal model, these frames have been described before as the outside frame and the inside frame. In essence, although the gimbals are nested in one another the connection between the bodies can be thought of as a 'chain' where body 1 is the first 'link' which is connected to body 2, which in turn is connected to body 3, which is connected to body 4, and finally the 'chain' is terminated at the stable platform, body 5. Each of these bodies have separate inertia characteristics that can be described independently from the other bodies. The global mass matrix $\mathbf{M}$ has components from each body n. The inertia characteristics of the rigid-elastic coupling term for an arbitrary body n are described next.

An important variable in the assemblage of the global mass matrix is the rigid, elastic coupling mass term, M_{re}. This term is also related to Hughes' first set of modal identities and is composed of momentum coefficients of each body. The n^{th} body's contribution to the global term, M_{re} is

$$M_{\text{n,re}} = \begin{pmatrix} P_{\text{n}} \\ H_{\text{n}} \end{pmatrix} \tag{2.6}$$

The variables P_{n} and H_{n} are the linear and angular momentum coefficients respectively and are computed from equation 2.7. These momentum coefficients play an important role in the mode ranking schemes of section 3.2.4, as they are a part of Hughes modal identities.

$$P_{\text{n}} = \int \Psi_n(r_n)dm, \; H_{\text{n}} = \int \tilde{r}_n \Psi_n(r_n)dm \tag{2.7}$$

Ψ_n is a set of admissible finite shape functions or in our case assumed modes. For a set of D modes, there are D number of momentum coefficients corresponding to each mode in the set. r_n is the position vector locating an element mass dm on the body n. $\tilde{r}_n$ is the skew symmetric matrix defined in equation 2.8 where r_n has the components of r_x, r_y, and r_z. In our case the body in which this equation is applied is a gimbal or the stable platform. To apply this equation to a discrete function, it is necessary to convert the integral into a finite sum. This is possible due to the fact that a lumped-mass parameter model was used to calculate the mode shapes for each of the structures. The following relation from a continuous to discrete case using N nodes for each body,

is found in equations 2.9 and 2.10. These equations were applied to the simulation to calculate these modal integrals.

$$\tilde{r}_n = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{2.8}$$

$$P_n = \int \Psi_n(r_n) dm \;\rightarrow\; P_n = \sum_{k=1}^{N} \Psi_n(r_{nk}) \Delta m_k \tag{2.9}$$

$$H_n = \int \tilde{r}_n \Psi_n(r_n) dm \;\rightarrow\; H_n = \sum_{k=1}^{N} \tilde{r}_{nk} \Psi_n(r_{nk}) \Delta m_k \tag{2.10}$$

The variable r_{nk} is the vector position of the k^{th} node on body n. The term $\Psi_n(r_{nk})$ is the nodal coordinate of the k^{th} node on body n and Δm_k is the mass associated with the position k. $\Psi_n(r_{nk})$ is a $3 \times D$ matrix associated with the shape functions. Once the modes are obtained from the finite element code, a preprocessor is used to compute these modal integrals. The output of this is used in the dynamic simulation of the nested gimbal system. The equations of motion are soived using a fourth order Runge-Kutta integrator. After the simulation is completed, results are compared to the Dynamic Analysis and Design Software (DADS) results from LMS International to verify the results obtained from using equation 2.1.

2.4 Full Model Results

The gimbal simulation model was run through various test scenarios and compared with DADS developed by LMS. The tests cases involve the comparison of both flexible and rigid gimbals. The importance of the comparison lies in the verification of the dynamic equations of the flexible gimbal simulation. One of the cases involves the simulation when the base body motion is prescribed. This comparison with DADS is used for twenty five modes for each body for a total of 100 elastic generalized coordinates. The initial conditions for this case are shown in Table 2.4 and the prescribed motion of the base body is in Table 2.5. Due to the prescribed rotation of the base body about its origin and the offset, ρ, between the base frame and the outer gimbal, the gimbal point of the outer gimbal translates in the z-direction. There is a reaction force applied to the inside bearings of the outer gimbal due to this motion. This force results in a deformation that provides a foundation for choosing a static force when using static correction modes to reduce the model of the set of elastic

	Initial Gimbal Angles (rads)	Initial Gimbal Rates $\left[\frac{rad}{sec}\right]$	Torque (in lbs.)
Outer Gimbal	0.0	0.0	0.0
Middle Gimbal	$-\frac{\pi}{2}$	0.0	0.0
Inner Gimbal	0.0	0.0	0.0
Stable Platform	0.0	0.0	0.0

Table 2.4: Initial Conditions for Truth Model

	Base Body Motion Angular Acceleration $\left[\frac{rad}{sec}\right]$
X	$\cos((2\pi 2)t + 0)$
Y	0.0
Z	0.0

Table 2.5: Base Body Motion for Truth Model

nested gimbals.

The first and second modes and third and fourth modes of the outer gimbal are shown in Figures 2-9 and 2-10, respectively. These modes are elements of the set of the elastic displacement field. There are two general types of modes which are represented in the figures. Each mode in the set can be divided into either one of these two categories. One is a twisting motion in which the nodes on the structure are out of plane. This is represented by the third and fourth modes in Figure 2-10. The second type has plane wave characteristics in which the distance between the nodes and the local x-z plane remain the same. This is shown in Figure 2-9.

The results of the angular rates of the gimbals are found in Figure 2-11 using 25 modes. The solution of this problem using the gimbal simulation was compared with the results from the DADS program. The outputs of the two programs are nearly identical, which builds confidence in the validity of both simulations. A problem arises from the fact that when the gimbals are simulated to be in motion for a small amount of time, the computation time to solve the differential equations is orders of magnitude greater. As a result, it is necessary to study methods to decrease computation time. One of these methods is to use model reduction techniques to reduce the system matrices while still maintaining adequate accuracy. In the next chapter, the theory behind the model reduction techniques is discussed. The truth model used to compare these model reduction techniques contains a total of four hundred elastic coordinates. The output for this model is shown in Figure 2-12.

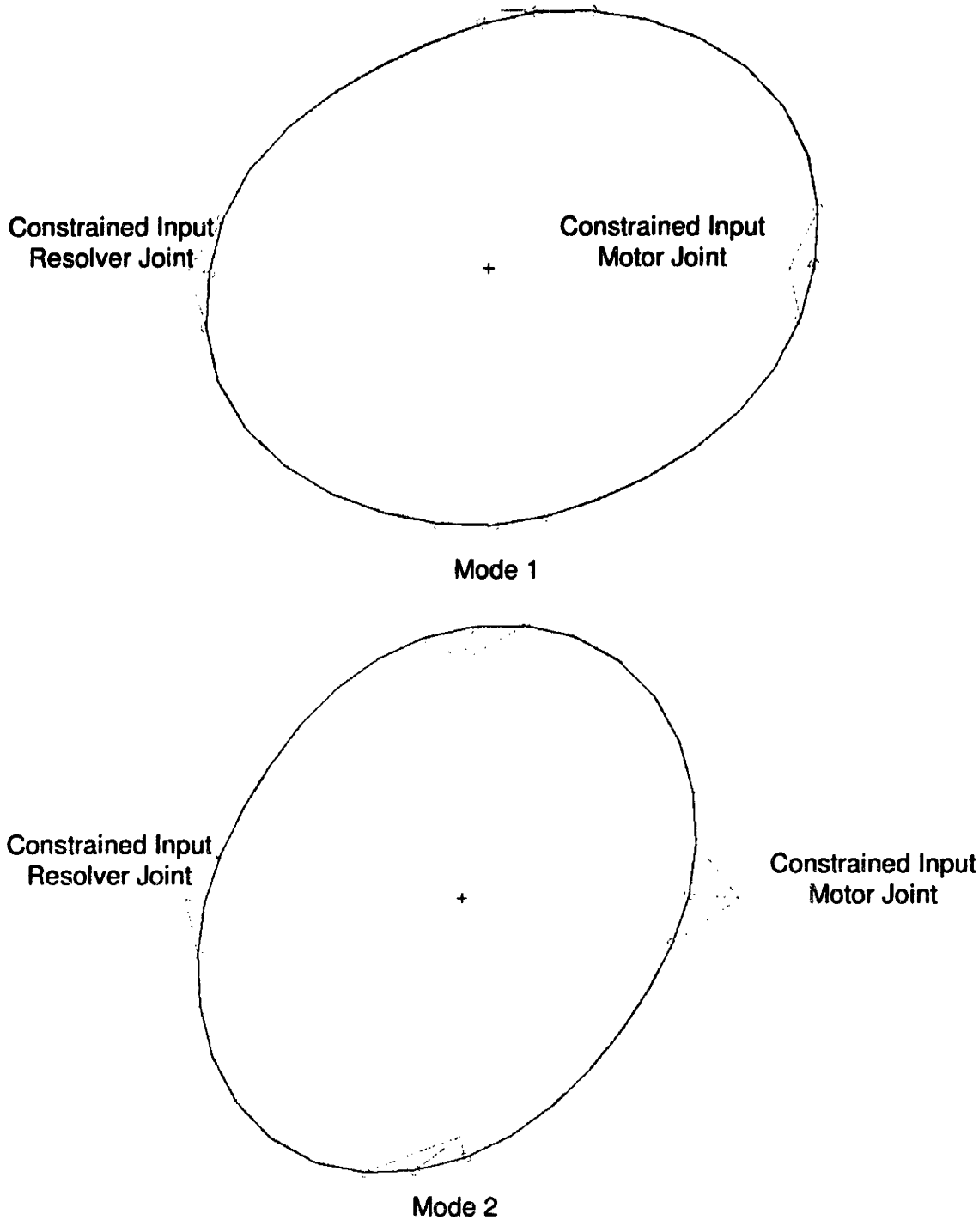
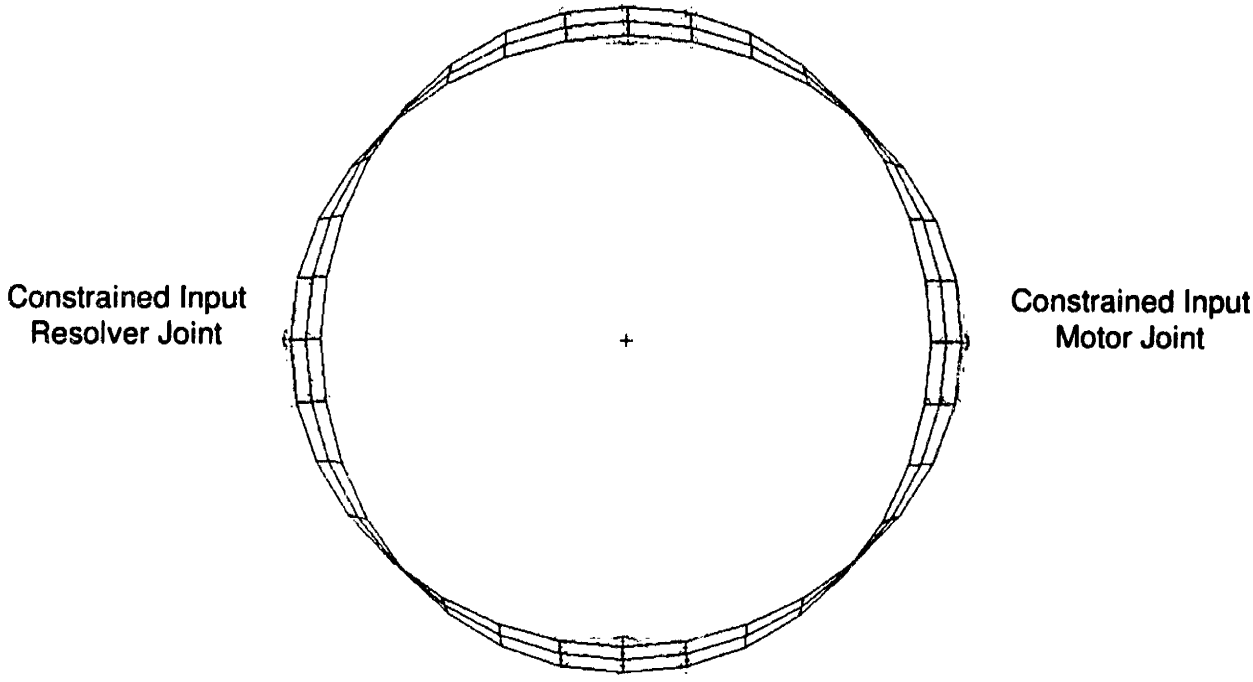


Figure 2-9: First and Second Modes of Outer Gimbal



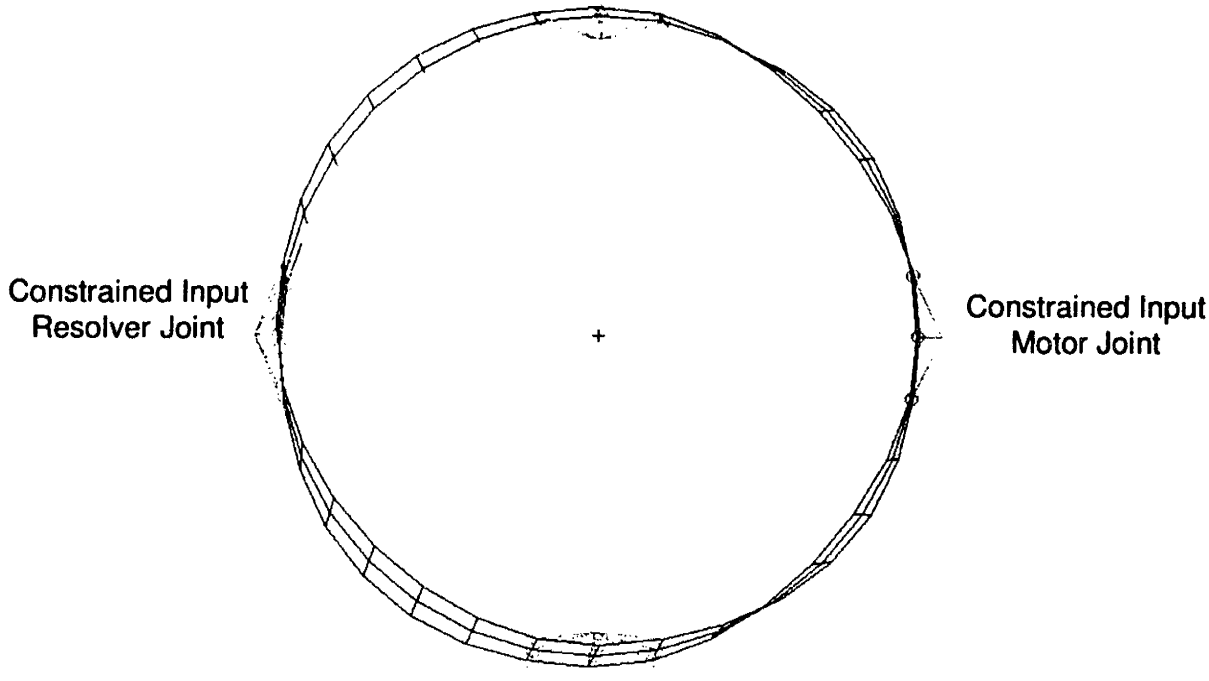


Figure 2-10: Third and Fourth Modes of Outer Gimbal

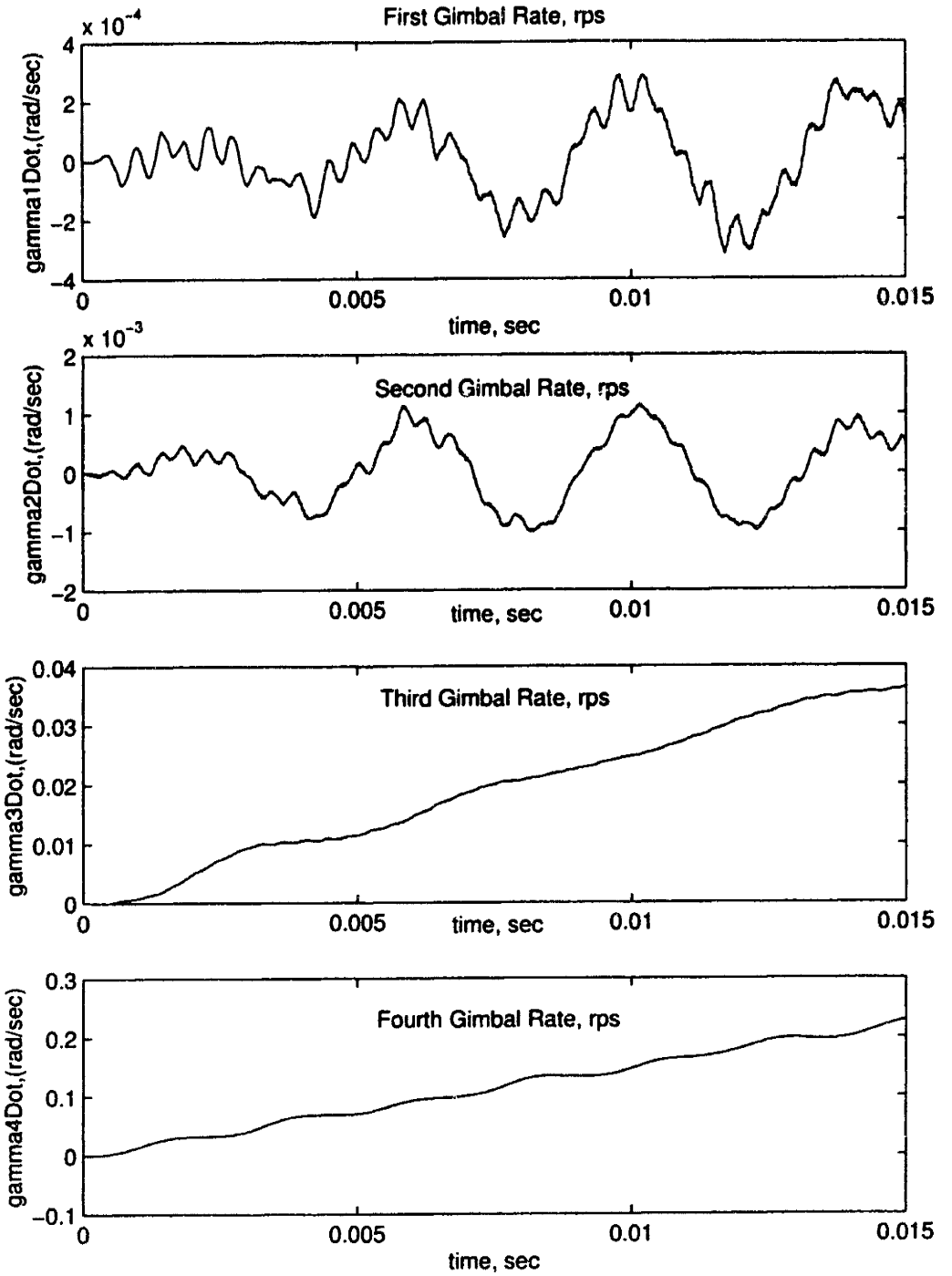


Figure 2-11: Results for Comparison Case with DADS Program, 25 Modes per Body

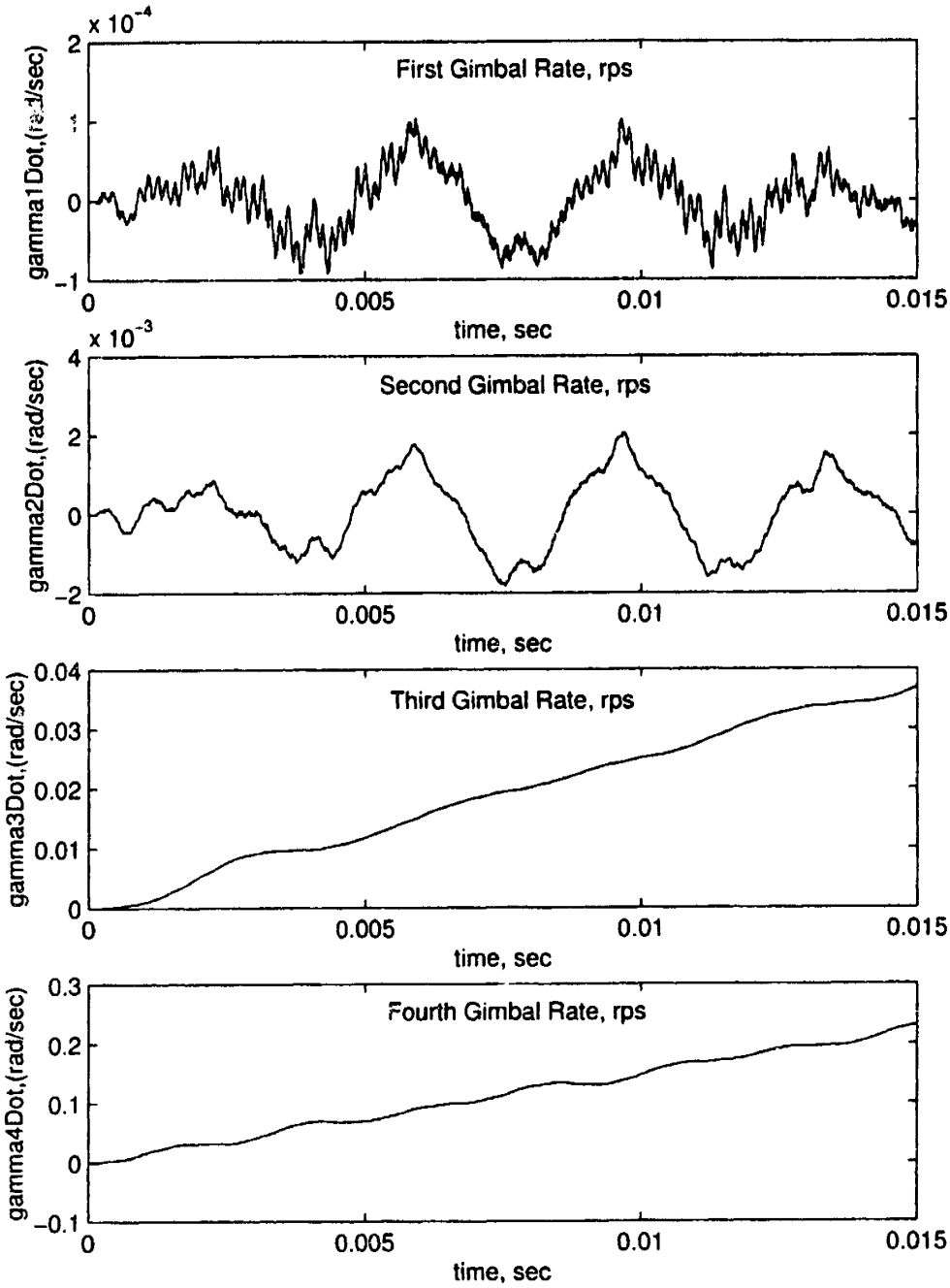


Figure 2-12: Results of Truth Model with 100 Modes for Each Body Using Initial Conditions.

Chapter 3

Model Reduction Theory

3.1 Overview

From the results in the previous chapter, the amount of time to solve the full system of equations with the maximum number of modes is extremely large. To simulate the gimbals using 100 modes for each body for 15 milliseconds requires on the order of hours to complete. To reduce the computation time, model reduction techniques are used to decrease the number of generalized coordinates. These methods include the use of component mode selection criteria and static correction modes. A new set of admissible finite element shape functions are used in the relation for the elastic displacement field $\mathbf{U}$ in equation 3.1 for a single body n, where ϕ_{ni} is an element of the set of shape functions and $q_{ni}(t)$ are the unknown time dependent generalized coordinates associated with the elastic deformation of mode i on body n [7]. $\mathbf{U}$ is a $6N \times 1$ column vector where the terms $\mathbf{u_k}$ correspond to the translational and rotational displacements of node k in the form $\mathbf{u_k} = \begin{bmatrix} U_{xk} & U_{yk} & U_{zk} & \Theta_{xk} & \Theta_{yk} & \Theta_{zk} \end{bmatrix}^T$.

$$\mathbf{U} = col(\mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_N}) = \sum_{i}^{6N} \phi_{ni} q_{ni}(t) \tag{3.1}$$

This chapter covers the formulation of the model reduction methods used in this thesis. The two methods described in this chapter attempt to reduce the model of the system so as to decrease the computation time required by limiting the size of the number of generalized coordinates. Both of these methods are finite-element based, in which a finite element program is used to construct the vibration modes of the system as well as the mass and stiffness matrices of each body. It is assumed that these variables, the eigenvectors, mass matrices, and stiffness matrices are known.

Finally, a basis for determining the error of the model reducing method is also discussed.

3.2 Mode Selection Criteria Using Ranking Methods

Two of the mode selection criteria have their bases from modal identities described by Hughes [4]. The identities that are derived in [4] involve the relation between constrained and unconstrained modal parameters. The unconstrained parameters are the rigid body modes. The basic premise of this method is to rank each mode according to its influence to some given system characteristic. There are two different global mode ranking methods using Hughes modal identities that are used to reduce the gimbal model and each will be described in the following sections. These ranking methods are described for a single body n.

3.2.1 Frequency Ranking

The simplest of the mode ranking criteria involves the use of frequency cutoff criterion in which the modes that are excited above a certain frequency are neglected. This criterion gives no insight into the nature of the modes that affect the system. The neglected high frequency modes may affect the system in the form of local deformation at the joint locations. Thus, this criterion may be valid for systems in which the forcing function does not have any high-frequency components. For problems in which high-frequency modes are excited it is necessary to develop other ranking methods.

3.2.2 XMX Ranking

As stated before, Hughes developed modal identities that relate rigid-body characteristics to those of the flexible bodies. There are two and the first involves only the modal momentum coefficients P_n and H_n defined in section 2.3. First, it is necessary to cast equations 2.9 and 2.10 into matrix form as in equation 3.4 which for a body n shows the calculation of the i^{th} modal integral for that body. ϕ_{ni} is a $6N \times 1$ vector with N being the number of nodes. The first six terms represent the 6 degrees of freedom, translation and rotation, of the first node point and the second six terms represent the 6 degrees of freedom of the second node. In short, each group of six represents the 6 degrees of displacement of that node. This ϕ_{ni} differs slightly from the matrix form in equations 2.9 and 2.10, $\Psi_n(r_{nk})$, in that $\Psi_n(r_{nk})$ contains all the modes of node k, while ϕ_{ni} represents the i^{th} mode and contains all the nodes. These definitions can best be seen through the following matrix

equations. For the k^{th} node

$$\Psi_n(r_{nk}) = \begin{bmatrix} u_{xk,1} & u_{xk,2} & \cdots & u_{xk,D} \\ u_{yk,1} & u_{yk,2} & \cdots & u_{yk,D} \\ u_{zk,1} & u_{zk,2} & \cdots & u_{zk,D} \end{bmatrix}_{3\times D} \tag{3.2}$$

where $u_{xk,i}, u_{yk,i}$, and $u_{zk,i}$ represent the displacement of the x, y, and z, coordinates of the k^{th} node and the i^{th} mode. However, ϕ_{ni} has a different structure. For the i^{th} mode this vector is defined by

$$\phi_{ni} = \begin{bmatrix} u_{x1,i} \\ u_{y1,i} \\ u_{z1,i} \\ \theta_{x1,i} \\ \theta_{y1,i} \\ \theta_{z1,i} \\ \vdots \\ u_{xN,i} \\ u_{yN,i} \\ u_{zN,i} \\ \theta_{xN,i} \\ \theta_{yN,i} \\ \theta_{zN,i} \end{bmatrix}_{6N\times 1} \tag{3.3}$$

In this equation $u_{xk,i}$, $u_{yk,i}$, and $u_{zk,i}$ are again the x, y, and z translation of the k^{th} node. Similarly, $\theta_{xk,i}$, $\theta_{yk,i}$, and $\theta_{zk,i}$ are the x, y, and z, rotation of the k^{th} node. The translational displacement defined in equations 3.2 and 3.3 are the same when referring to node k and i.

The finite element mass matrix, M_n, is also constructed in such a way that every six diagonal terms represent the mass properties of one node. The diagonal terms represent the mass associated with a node, k. The mass matrix is found in equation 3.5, where M_{nk} is the 6×6 mass matrix of the k^{th} node in the finite element analysis. R_n and E_n are defined in equation 3.6. For body n with N nodes, the size of R_n and E_n is $3 \times 6N$ since there is a full six degree of freedom of motion for each node. I is the 3×3 identity matrix, 0 is the 3×3 zero matrix, and $\tilde{r}_{nk}$ is again the skew

symmetric matrix defined in equation 2.8 of the position vector r_k.

$$\begin{aligned} P_{ni} &= E_n M_n \phi_{ni} \\ H_{ni} &= R_n M_n \phi_{ni} \end{aligned} \tag{3.4}$$

$$M_n = \begin{bmatrix} M_{n1} & 0 & \cdots & 0 \\ 0 & M_{n2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & M_{nN} \end{bmatrix} \tag{3.5}$$

$$\begin{aligned} [E_n] &= \begin{bmatrix} I & 0 & I & 0 & \cdots & I & 0 \end{bmatrix} \\ [R_n] &= \begin{bmatrix} \tilde{r}_{n1} & I & \tilde{r}_{n2} & I & \cdots & \tilde{r}_{nN} & I \end{bmatrix} \end{aligned} \tag{3.6}$$

P_{ni} and H_{ni} are both 3×1 column vector representing the momentum coefficient of the i^{th} mode on the n^{th} body. Applying this relation to equation 2.9 and 2.10 results in the following equations.

$$P_n = \begin{bmatrix} P_{n1} & P_{n2} & \cdots & P_{nD} \end{bmatrix}_{3\times D} \tag{3.7}$$

$$H_n = \begin{bmatrix} H_{n1} & H_{n2} & \cdots & H_{nD} \end{bmatrix}_{3\times D} \tag{3.8}$$

Combining the momentum coefficients into a single vector, the relation in equation 3.9 is obtained.

$$\begin{pmatrix} P_{ni} \\ H_{ni} \end{pmatrix}_{6\times 1} = \begin{bmatrix} E_n \\ R_n \end{bmatrix}_{6\times 6N} [M_n]_{6N\times 6N} \, (\phi_{ni})_{6N\times 1} \tag{3.9}$$

The transpose of the rigid body mode, X_n, is represented in equation 3.10 and now it is possible to rewrite equation 3.9 as equation 3.11.

$$\begin{bmatrix} E_n \\ R_n \end{bmatrix}_{6\times 6N} = \begin{bmatrix} I & 0 & I & 0 & \cdots & I & 0 \\ \tilde{r}_{n1} & I & \tilde{r}_{n2} & I & \cdots & \tilde{r}_{nN} & I \end{bmatrix}_{6\times 6N} = X_n^T \tag{3.10}$$

$$\begin{pmatrix} P_{ni} \\ H_{ni} \end{pmatrix}_{6\times 1} = X_n^T M_n \phi_{ni} \tag{3.11}$$

Once the identity for the influence of one mode on the momentum coefficients in equation 3.11 is

obtained, it is important to define the rigid body portion of the global mass matrix used in equation 2.2 as the formula in equation 3.12 and used in [10].

$$M_{n,rr} = \begin{bmatrix} m_n I_3 & -\tilde{c}_n \\ \tilde{c}_n & J_n \end{bmatrix} \tag{3.12}$$

Here, m_n is the total mass of body n and I_3 is again the 3×3 identity matrix. $\tilde{c}_n$ is the skew symmetric matrix of the first mass moment of inertia of body n defined in equation 3.13 and J is the second mass moment of inertia defined in equation 3.14

$$c_n = \int r_n dm \tag{3.13}$$

$$J_n = -\int \tilde{r}_n \tilde{r}_n dm \tag{3.14}$$

Using the discrete form of the first and second mass moments of inertia it is possible to relate the rigid body portion of the global mass matrix for a body n, $\mathbf{M_{n,rr}}$, with the rigid body modes. First the rigid body matrix, X_n is decomposed into its individual nodal components defined by

$$X_n^T = \begin{bmatrix} X_{n1} & X_{n2} & \cdots & X_{nN} \end{bmatrix} \tag{3.15}$$

where X_{nk} is a 6×6 matrix of the k^{th} nodal component of the rigid body mode defined by $X_{nk} = \begin{bmatrix} I & 0 \\ \tilde{r}_{nk} & I \end{bmatrix}$

With this identity and the finite element mass matrix found in equation 3.5, it is possible to form a relation between the rigid body mass properties and the rigid body modes of the n^{th} structure. It begins with the term $X_n^T M_n X_n$ which is calculated in equation 3.16.

$$X_n^T M_n X_n = \begin{bmatrix} X_{n1} & X_{n2} & \cdots & X_{nN} \end{bmatrix} \begin{bmatrix} M_{n1} & 0 & \cdots & 0 \\ 0 & M_{n2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & M_{nN} \end{bmatrix} \begin{bmatrix} X_{n1}^T \\ X_{n2}^T \\ \vdots \\ X_{nN}^T \end{bmatrix} \tag{3.16}$$

Simplifying this equation yields

$$X_n^T M_n X_n = \left[X_{n1} M_{n1} X_{n1}^T + X_{n1} M_{n1} X_{n1}^T + \cdots + X_{nN} M_{nN} X_{nN}^T \right] \tag{3.17}$$

From this equation, an individual term $X_{nk}M_{nk}X_{nk}^T$ is broken down into its components as

$$X_{nk}M_{nk}X_{nk}^T = \begin{bmatrix} I & 0 \\ \tilde{r}_{nk} & I \end{bmatrix} \begin{bmatrix} M_{nk,v} & 0 \\ 0 & M_{nk,w} \end{bmatrix} \begin{bmatrix} I & \tilde{r}_{nk}^T \\ 0 & I \end{bmatrix} \tag{3.18}$$

where $M_{nk,v}$ is the 'translational' degree of freedom of mass, which for this finite element model has diagonal terms all equal to the mass of node k and $M_{nk,w}$ is the 'rotational' degree of freedom mass which is a 3×3 zero matrix. Performing this matrix multiplication, the following relation is obtained in equation 3.19

$$X_{nk}M_{nk}X_{nk}^T = \begin{bmatrix} I_3 M_{nk,v} I_3 & I_3 M_{nk,v} \tilde{r}_{nk}^T \\ M_{nk,v}\tilde{r}_{nk} I_3 & \tilde{r}_{nk} M_{nk,v} \tilde{r}_{nk}^T \end{bmatrix} \tag{3.19}$$

$X^T M X$ is obtained by summing up the components defined in equation 3.19. This results in the following equation 3.20.

$$X_n^T M_n X_n = \sum_{k=1}^{N} \begin{bmatrix} M_{nk,v} & M_{nk,v}\tilde{r}_{nk}^T \\ M_{nk,v}\tilde{r}_{nk} & \tilde{r}_{nk} M_{nk,v} \tilde{r}_{nk}^T \end{bmatrix} \tag{3.20}$$

Noting that $\tilde{r}_{nk}^T = -\tilde{r}_{nk}$ and $M_{nk,v} = I_3 m_{nk,v}$, where $m_{nk,v}$ is the mass of node k, and using equations 3.13 and 3.14, the following relation for each of the elements in the matrix is obtained.

$$\sum_{k=1}^{N} I_3 m_{nk,v} = \sum_{k=1}^{N} m_{nk,v} \approx \int I_3 dm = m_n I_3 \tag{3.21}$$

$$\sum_{k=1}^{N} \tilde{r}_{nk} I_3 m_{nk,v} = \sum_{k=1}^{N} \tilde{r}_{nk} m_{nk,v} \approx \int r_n dm = \tilde{c}_n \tag{3.22}$$

$$\sum_{k=1}^{N} \tilde{r}_{nk}^T I_3 m_{nk,v} = \sum_{k=1}^{N} -\tilde{r}_{nk} m_{nk,v} \approx \int -r_n dm = -\tilde{c}_n \tag{3.23}$$

$$\sum_{k=1}^{N} \tilde{r}_{nk} I_3 m_{nk,v} \tilde{r}_{nk}^T = \sum_{k=1}^{N} -\tilde{r}_{nk} m_{nk,v} \tilde{r}_{nk} \approx \int -r_n r_n dm = J_n \tag{3.24}$$

Using these relations, it is possible to arrive at the following equation 3.25.

$$M_{n,rr} = X_n^T M_n X_n = \begin{bmatrix} m_n I_3 & -\tilde{c}_n \\ \tilde{c}_n & J_n \end{bmatrix} \tag{3.25}$$

Since the momentum coefficients are related to the rigid body modes and the rigid body modes are related to the rigid body portion of the mass matrix it is now possible to associate the momentum coefficients to the rigid body mass matrix and form a basis for the XMX ranking. From [4], the momentum coefficients for a single continuous body are related to the elements of the rigid body mass matrix using the modal identities by equations 3.26 through 3.28.

$$\sum_{i=1}^{\infty} P_{ni} P_{ni}^T = m_n I \tag{3.26}$$

$$\sum_{i=1}^{\infty} H_{ni} P_{ni}^T = \tilde{c}_n \tag{3.27}$$

$$\sum_{i=1}^{\infty} H_{ni} H_{ni}^T = J_n \tag{3.28}$$

However, for a discretized body as in the finite element model of the ring, the limit of the summation becomes the number of degrees of freedom in the system. This is represented for a 6N degree of freedom system in equations 3.29 through 3.31.

$$\sum_{i=1}^{6N} P_{ni} P_{ni}^T = m_n I \tag{3.29}$$

$$\sum_{i=1}^{6N} H_{ni} P_{ni}^T = \tilde{c}_n \tag{3.30}$$

$$\sum_{i=1}^{6N} H_{ni} H_{ni}^T = J_n \tag{3.31}$$

Substituting the equations 3.29 through 3.31 into 3.25 the following equation 3.32 is obtained.

$$M_{n,rr} = X_n^T M_n X_n = \begin{bmatrix} m_n I_3 & -\tilde{c}_n \\ \tilde{c}_n & J_n \end{bmatrix} = \sum_{i=1}^{6N} \begin{bmatrix} P_{ni} P_{ni}^T & P_{ni} H_{ni}^T \\ H_{ni} P_{ni}^T & H_{ni} H_{ni}^T \end{bmatrix} = \sum_{i=1}^{6N} \bar{M}_i \tag{3.32}$$

$\bar{M}_i$ is the contribution of the i^{th} mode to the complete rigid mass matrix. For notational purposes the complete rigid body matrix is defined as $M_\infty = M_{n,rr}$ for the n^{th} body. It should be noted that the mass properties of the structure can be made up of the mass properties of each individual mode. Therefore it is possible to study the contribution of each mode to the total mass matrix M_∞. The contribution of M_i to M_∞ can provide a measure of importance of the i^{th} mode.

It is also possible to prove this identity using equation 3.11 and noting that the matrix of

eigenvectors is mass normalized such that $\sum_{i=1}^{6N} \phi_{ni}\phi_{ni}^T M_n = I$. This proof is shown in the following equations.

$$M_\infty = X_n^T M_n X_n = X_n^T M_n I X_n = \sum_i^{6N} X_n^T M_n \phi_{ni}\phi_{ni}^T M_n X_n = \tag{3.33}$$

$$\sum_i^{6N} (X_n^T M_n \phi_{ni})(X_n^T M_n \phi_{ni})^T = \sum_{i=1}^{6N} \begin{bmatrix} P_i P_i^T & P_i H_i^T \\ H_i P_i^T & H_i H_i^T \end{bmatrix} = \sum_{i=1}^{6N} \bar{M}_i \tag{3.34}$$

This relation is important because $\bar{M}_i$ is computed to measure the influence of node i on the total rigid body mass properties of a body. From this equation, $\bar{M}_i$ can be computed different ways, either using the momentum coefficients or the rigid body matrix, eigenvectors, and mass matrix of body n.

3.2.3 XMGMX Ranking

The second ranking criteria uses another of Hughes' modal identities employing the momentum coefficients along with the natural frequencies of each mode. This identity for a single continuous body, ξ, is found in equations 3.36 to 3.38. Here Ω_i is the natural frequency corresponding to the i^{th} mode. For notational purposes in this section the subscript for a body n is dropped, but the ranking method described here is again for a single body. $F(r, \varepsilon)$ is defined by Hughes as the *flexibility kernel* and along with $\sigma(r)$, the volume mass density of r, describes the distribution of flexibility and inertia respectively. ε is a surrogate for r. The *flexibility kernel* can be defined in terms of the mode shapes as

$$F(r, \varepsilon) = \sum_{i=1}^{\infty} \frac{\phi_i(r)\phi_i^T(\varepsilon)}{\Omega_i^2} \tag{3.35}$$

It is important to note that since the eigenvectors are mass normalized, $\sum_{i=1}^{\infty} \phi_i(r)\phi_i^T(\varepsilon)M = 1$ when r and ε are considered through the same body ξ and that $\int_\xi 1\sigma dr = m$, the mass of the body.

$$\sum_{i=1}^{\infty} \frac{P_i P_i^T}{\Omega_i^2} = \int_\xi \int_\xi F(r, \varepsilon)\sigma(r)\sigma(\varepsilon) dr d\varepsilon \tag{3.36}$$

$$\sum_{i=1}^{\infty} \frac{H_i P_i^T}{\Omega_i^2} = \int_\xi \int_\xi \tilde{r} F(r, \varepsilon)\sigma(r)\sigma(\varepsilon) dr d\varepsilon \tag{3.37}$$

$$\sum_{i=1}^{\infty} \frac{H_i H_i^T}{\Omega_i^2} = -\int_\xi \int_\xi \tilde{r} F(r, \varepsilon)\tilde{\varepsilon}\sigma(r)\sigma(\varepsilon) dr d\varepsilon \tag{3.38}$$

Substituting the *flexibility kernel* into these equations results in

$$\sum_{i=1}^{\infty} \frac{P_i P_i^T}{\Omega_i^2} = \int_\xi \int_\xi \sum_{i=1}^{\infty} \frac{\phi_i(r)\phi_i^T(\varepsilon)}{\Omega_i^2} \sigma(r)\sigma(\varepsilon) dr d\varepsilon \tag{3.39}$$

$$\sum_{i=1}^{\infty} \frac{H_i P_i^T}{\Omega_i^2} = \int_\xi \int_\xi \tilde{r} \sum_{i=1}^{\infty} \frac{\phi_i(r)\phi_i^T(\varepsilon)}{\Omega_i^2} \sigma(r)\sigma(\varepsilon) dr d\varepsilon \tag{3.40}$$

$$\sum_{i=1}^{\infty} \frac{H_i H_i^T}{\Omega_i^2} = -\int_\xi \int_\xi \tilde{r} \sum_{i=1}^{\infty} \frac{\phi_i(r)\phi_i^T(\varepsilon)}{\Omega_i^2} \tilde{\varepsilon} \sigma(r)\sigma(\varepsilon) dr d\varepsilon \tag{3.41}$$

Using the normalization identity, it is possible to arrive at the same form as equations 3.29 through 3.31 except with the Ω_i term in the denominator added. This relation becomes

$$\sum_{i=1}^{\infty} \frac{P_i P_i^T}{\Omega_i^2} = \int_\xi \frac{1}{\Omega_i^2} \sigma(r) dr = \frac{mI}{\Omega_i^2} \tag{3.42}$$

$$\sum_{i=1}^{\infty} \frac{H_i P_i^T}{\Omega_i^2} = \int_\xi \frac{1}{\Omega_i^2} \tilde{r} \sigma(r) dr = \frac{\tilde{c}}{\Omega_i^2} \tag{3.43}$$

$$\sum_{i=1}^{\infty} \frac{H_i H_i^T}{\Omega_i^2} = -\tilde{\xi} \int_\xi \frac{1}{\Omega_i^2} \sigma(r) \tilde{r} dr = \frac{J}{\Omega_i^2} \tag{3.44}$$

Again these equations are for a continuous body that can be described with an infinite number of modes representing the elastic body, but for a finite case the equations can be simplified as in [7]. For a discrete case, the integrals in equations 3.36 to 3.38 are elements of the matrix product MGM. Similar to the XMX ranking, the XMGMX ranking can be composed of these modal identities where G is K^{-1}, the inverse of the global stiffness matrix. Similar to the M_∞ matrix the MGM_∞ matrix is defined by equation 3.45.

$$(MGM)_\infty = X^T MGMX \tag{3.45}$$

For the discrete case, the influence of an individual mode to this matrix can be established by using the modal identity and the momentum coefficients, P_i and H_i, as in equation 3.46 [7].

$$(MGM)_\infty = X^T MGMX = \sum_{i=1}^{6N} \frac{1}{\Omega_i^2} \begin{bmatrix} P_i P_i^T & P_i H_i^T \\ H_i P_i^T & H_i H_i^T \end{bmatrix} = \sum_{i=1}^{6N} \frac{1}{\Omega_i^2} \tilde{M}_i \tag{3.46}$$

Using equation 3.46 we have a second basis for ranking the modes of each structure involving M_i along with the natural frequencies for each mode Ω_i.

Ranking Method	Spectral Radius
XMX	$\rho_i = \rho(M_\infty^{-\frac{1}{2}} M_i M_\infty^{-\frac{1}{2}})$
XMGMX	$\rho_i = \rho((MGM)_\infty^{-\frac{1}{2}} (MGM)_i (MGM)_\infty^{-\frac{1}{2}})$

Table 3.1: Spectral Radii used to Rank Modes

3.2.4 Ranking Method

Now that the modal identities relating the influence of mode i to the complete *infinity* matrices are obtained, it is necessary to form the method of ranking the modes. Since the measure of the influence of the i^{th} mode to the *infinity* matrices is a matrix, it is advantageous to define a scalar term to rank the modes. The ranking criteria involves the use of the spectral radius operator, ρ. This operator computes the largest eigenvalue of the matrix to be studied. It is taken to be a good measure because it corresponds to the matrix norm of a symmetric matrix, which is a measure of the size of the elements of the matrix [7]. The modes are ranked according to the size of the spectral radius, with the largest value having the most influence according to the identity used. Table 3.1 shows the matrix formed from the modal identities that will be operated on by the spectral radius [7].

With the ranking method developed from the modal identities, a criteria for reducing the model of the gimbal simulation can be used. However this method involves the selection of the modes due to their influence on the mass properties of structure and not on the configuration of the structure. It is important to study the use of static modes as a means of reducing the model while maintaining deformations due to local forces at the joints.

3.3 Dynamics Using Vibration and Static Correction Modes

Describing the motion of a body using both vibration and static correction modes reduces the number of modes needed to represent the full system by taking into account the local deformation caused by reaction forces at the joints of the mechanical structure. The static correction modes can be used to represent the high frequency modes of the body because these high frequency modes 'represent local deformation effects' [13]. The local deformation between two adjacent elastic bodies can be represented by these static modes. The use of static modes to replace the high frequency modes not only reduces the size of the global matrices, but also increases the step size that can be used in integration of the equations of motion. The basis for using this method lies in [13] where

'a finite element base numerical method for dynamic analysis of mechanical systems that contain complex-shaped flexible bodies' is used. The subscript n in this section has also been omitted here. The equations here are used on any individual structure.

Before describing how to obtain a set of static modes it is important to understand the finite element equations associated with the model to obtain nodal point displacements due to an applied force. This is represented by the equation 3.47 from [1]. Here, the term U is the nodal point displacements, F_a represents the vector of forces acting on the structure at different nodal points, and K is the finite-element stiffness matrix of the structure. For this problem the matrix K is given in the problem from the finite-element code.

$$KU = F_a \Rightarrow U = K^{-1}F_a \tag{3.47}$$

To determine the static modes it is necessary to solve for the vector U and normalize this vector so that the static modes and kept normal modes are linearly independent. U is a vector of the displacements of the nodes. The length is equal to the number of degrees of freedom of the structure. For the ring problem the vector has six hundred terms, three translational and three rotational degrees of freedom for each of the 100 nodes. To solve for the displacements U it is necessary to take the inverse of K. However if the body is constrained so that rigid body motion is not allowed, then the inverse of K cannot be obtained due to zero rows and columns. To obtain the static modes using the boundary conditions placed on the structure it is important to take into account these constrained nodes.

The basic premise of using static deformation modes is to represent local deformation effects of concentrated loads at kinematic joints. Static deformation modes, also called constraint modes and attachment modes, are defined by imposing a unit force on one of the physical coordinates and zero force on the remainder of the physical coordinates. This force is applied to the constrained structure and the static displacement is obtained. The case in which modes are taken from an unconstrained model is not represented here but can be found in reference [11]. The static modes using the constrained structures can be obtained from equation 3.48. These nodal coordinates for the static modes are found using a similar formulation of the finite-element method.

The following equations are appropriate for 'articulated structures, i.e. structures in which kinematics connections permit large relative displacement between components that undergo small elastic deformation.' [11] The gimbal configuration allows for large rotations between gimbals, but

linear elastic theory is used to describe the gimbals' flexibility. The derivation of the equations used to describe the motion of jointed bodies using static correction modes and vibration modes can be found in [11].

$$\phi = [S^T G S - \Phi_{\bar{n}} \Lambda_{\bar{n}\bar{n}}^{-1} \Phi_{\bar{n}}^T] F_a \tag{3.48}$$

If the boundary conditions of the structure are defined so as to restrict rigid body motion, then S is the identity matrix as in the gimbal model. G is defined as the inverse of the K stiffness matrix without the points constrained by the boundary conditions. Using only the free nodes in the stiffness equation it is possible to take the inverse. G is defined as in equation 3.49 where k_{ww}, k_{wv}, k_{vw}, and k_{vv} are the elements of the stiffness matrix associated with both translation and rotation of the unconstrained nodes of the body. The zeros represent the nodes that are constrained due to the boundary conditions. $\Phi_{\bar{n}}$ is the matrix of kept flexible normal modes which has the form $\Phi_{\bar{n}} = \begin{bmatrix} \phi_1 & \phi_2 & \cdots \end{bmatrix}$ and $\Lambda_{\bar{n}\bar{n}}^{-1}$ is a diagonal matrix that has eigenvalues corresponding to the kept modes. F_a is the force applied to the body. This vector represents the force that statically deforms the structure. The term $\Phi_{\bar{n}} \Lambda_{\bar{n}\bar{n}}^{-1} \Phi_{\bar{n}}^T$ ensures that the modes used in the simulation (both normal and static) are linearly independent. The best results for modeling the dynamics of an articulated structure are obtained using kept normal modes and residual inertia relief attachment modes calculated from equation 3.48 [3].

$$G = \begin{bmatrix} \lceil k_{ww} & k_{wv} \rceil^{-1} & 0 \\ \lfloor k_{vw} & k_{vv} \rfloor & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.49}$$

Once the vibration and static modes are determined, the elastic displacement of a body u and be written as in equation 3.50 Here $\Phi_{\bar{n}}$ and $\Phi_{\bar{m}}$ are mode shapes from vibration and static analysis, respectively, and $q_{\bar{n}}$ and $q_{\bar{m}}$ are the amplitudes of these modes.

$$u = \Phi_{\bar{n}} q_{\bar{n}} + \Phi_{\bar{m}} q_{\bar{m}} \tag{3.50}$$

Now we combine the matrices, $\Phi_{\bar{n}}$ and $\Phi_{\bar{m}}$ to form the complete modal matrix, equation 3.51, which are the set of admissible shape functions defined in section 2.3. The modal mass and stiffness matrices for the set of modes are defined in equation 3.52, where for $\bar{n}$ modes, $I_{\bar{n}\bar{n}}$ is an $\bar{n} \times \bar{n}$ identity matrix, M and K are global mass and stiffness matrices, respectively and $A_{\bar{n}\bar{n}}$ is a $\bar{n} \times \bar{n}$

diagonal matrix of the eigenvalues of the kept vibration modes.

$$\Phi = \begin{bmatrix} \Phi_{\bar{n}} & \Phi_{\bar{m}} \end{bmatrix} \tag{3.51}$$

$$\Phi^T M \Phi = \begin{bmatrix} I_{\bar{n}\bar{n}} & 0 \\ 0 & \Phi_{\bar{m}}^T M \Phi_{\bar{m}} \end{bmatrix}, \quad \Phi^T K \Phi = \begin{bmatrix} A_{\bar{n}\bar{n}} & 0 \\ 0 & \Phi_{\bar{m}}^T K \Phi_{\bar{m}} \end{bmatrix} \tag{3.52}$$

Once the model reduction techniques are developed, it is important to develop a measure error as a source of comparison between the model reduce case and the truth model. This basis is discussed in the next section.

3.4 Error Basis

The development of the methods to reduce the model has been described in the previous sections. The component mode selection criterion and the use of static modes offer different techniques to reduce the model. Now that the equations and theory of the procedure have been presented, it is possible to implement the theory in the simulation of the dynamics of nested gimbals and compare it to the truth model in section 2.4.

A quantitative study is done on the results obtained using the truncation techniques in the previous section. The error measure used in this section is based on the one obtained in [7] that is deemed to be most able to 'distinguish small deviations between responses of the reduced order models and the truth models.' The error, E between the model reduced system and the truth model is defined in equation 3.53, where $r(t)$ is defined as the response of the truth model and $e(t)$ is defined as the response of the reduced order model minus the truth model or the error at some time, t.

$$E = \frac{\int_0^\infty e^2(t)dt}{\int_0^\infty r^2(t)dt} \tag{3.53}$$

This error equation provides a quantitative method for determining the accuracy of the reduced order model with the truth model in terms of the angular velocities of the drive axis of each body. This parameter is computed for a test case and the results are examined in section 4.5

This page intentionally left blank.

Chapter 4

Results

4.1 Overview

This chapter applies the theory from the previous chapter to reduce the full model of a set of elastic, nested gimbals. The implementation of the model reduction theory are added to the original gimbal simulation developed by Timothy Barrows at the Charles Stark Draper Laboratory. The program was appended to allow for the use of the model reduction techniques described in Chapter 3. This chapter discusses the issues in which the method for model reduction was applied to the original program. Next, the results of the model reduction techniques are shown for different combinations of model reduction cases. Lastly, the error for the model reduce programs with respect to the truth model are discussed.

4.2 Computer Implementation of Theory

The development of the model reduction computer program involves three main parts. The first involves the pre-processing of the data from the finite element model. The second involves the solution of the dynamic equations taking data from the pre-processor. Once the data has been used to develop a solution to the differential equations, a post-processor takes the data and creates from it plots that can be understood. A flow chart in Figure 4-1 shows the algorithm of the computer simulation.

The first part of the computer simulation involves the development of the finite element model using MSC/Nastran in the correct local coordinate system and boundary conditions. Once each of the models are created, a frequency analysis is completed in which the eigenvectors, natural

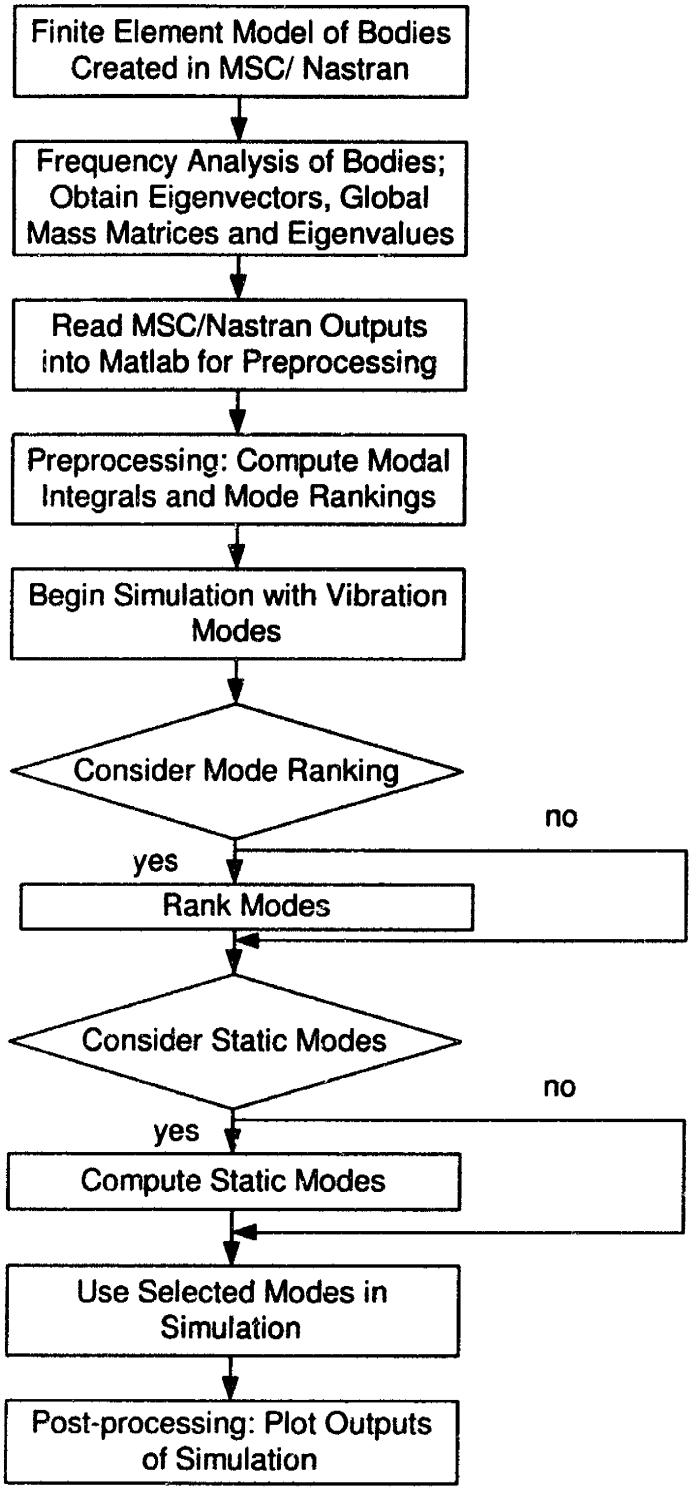


Figure 4-1: Flow Chart of Computer Program Process.

frequencies associated with the eigenvectors, and the global mass and stiffness matrices are output into a text file. This text file is then accessed using MATLAB and the data is imported from the text file into the pre-processor written in MATLAB. At this point the data from the finite element model is used to compute the modal integrals defined in equations 2.9 and 2.10. Using the global mass and stiffness matrices, the modes are ranked according to the equations in section 3.2. The calculations up to this point are one-time calculations which are only dependent on the structure of the each of the gimbals and the stable platform and are independent of time. After the completion of these calculations, the output data is used in the solving the equations of motion in equation 2.1. It is here that the user can determine mode rankings and static modes in the modeling of the elastic displacement field. A data file containing the time vector and the response of the gimbal angles and rates is created. Plots are created using this data. The plots can be used to see the results of a single run or compare the same run using different modes.

Using this gimbal simulation it is possible to compare cases using different types and number of modes. The mode ranking and static-correction mode software written in MATLAB are in Appendix A. The next section shows the results of these runs.

4.3 Comparisons

This section compares the results from the truth model with the reduced order model. First models using only the global selection criteria are used. These mode rankings are used by including the most influential modes according to the criteria developed in section 3.2. Next static modes are used along with the unranked vibration modes to compute the results of the reduced order model. Finally static modes are used along with the ranked modes to determine if this combination produces more accurate results. The same initial conditions as described in section 2.4, Tables 2.4 and 2.5 are used in this comparison.

4.3.1 Frequency Cutoff Case

The most basic case of model reduction involves using the lowest frequency modes of each of the bodies. This method described in the previous chapter uses the modes under a certain frequency. As stated before, this method uses no information about the configuration of the system. The comparisons are done using the lowest frequency modes, ranging from 5 to 25 modes. The system is put through the same initial conditions as the truth model in section 2.4. The qualitative

comparison of the gimbal rates are shown in Figures 4-2 to 4-6.

Using only five of the lowest frequency modes for each of the bodies does not produce accurate results as seen from Figure 4-2. There seem to be two prevailing frequencies for the first gimbal rate, a low frequency response coupled with a high frequency contribution. The low frequency response of the system matches closely with that of the truth model but the high frequency amplitude is far from representing the results from truth model. The low frequency response of the other gimbal rates match the truth model quite well for an eighty percent reduction in the number of the generalized coordinates.

The results from using 10, 15, and 20 modes (Figures 4-3, 4-4, and 4-5 respectively) have similar results. The frequency of the response appears to correspond to that of the truth model, but again the amplitude of the response varies significantly. The other gimbal rates match the truth model relatively well although there are significant differences in the second gimbal rate. The final comparison is done using 25 modes. In this case, the amplitudes of the response match the truth model well. With these results, comparisons are made using the XMX ranking and XGMX ranking, to see if the same type of results can be obtained by using less number of modes.

4.3.2 Mode Ranking Case

The first model reduction technique that is applied to the gimbal simulation is the mode ranking techniques. The ranking is done after the mode shapes are calculated but before the actual time dependent simulation is run. The ranking of the modes for the XMX and XMGMX method are found in Table 4.1. The first 40 ranked modes are present in the table where 1 represent the first mode with the lowest frequency and 100 represents the 100th mode with the highest frequency among the calculated modes.

The ranking of the first forty modes are the same in the XMX ranking method and the XMGMX ranking method. The rankings however for the other modes are different for the each of the ranking methods. The same ranking of the modes is due of the high stiffness rods that are used to mathematically represent the physical structure. An actual gimbal is extremely stiff because it is made so as to be rigid. From the mode ranking equations if there is high frequency the term in the XMGMX ranking is small and its influence from the XMX ranking is not noticeable.

Using the ranking from Table 4.1 the gimbal simulation was run using modes varying from 5 to 25 for each body. The total number of elastic coordinates used in the system varies from 20 to 100. The same conditions as the truth model are applied to the system. The low frequency response of

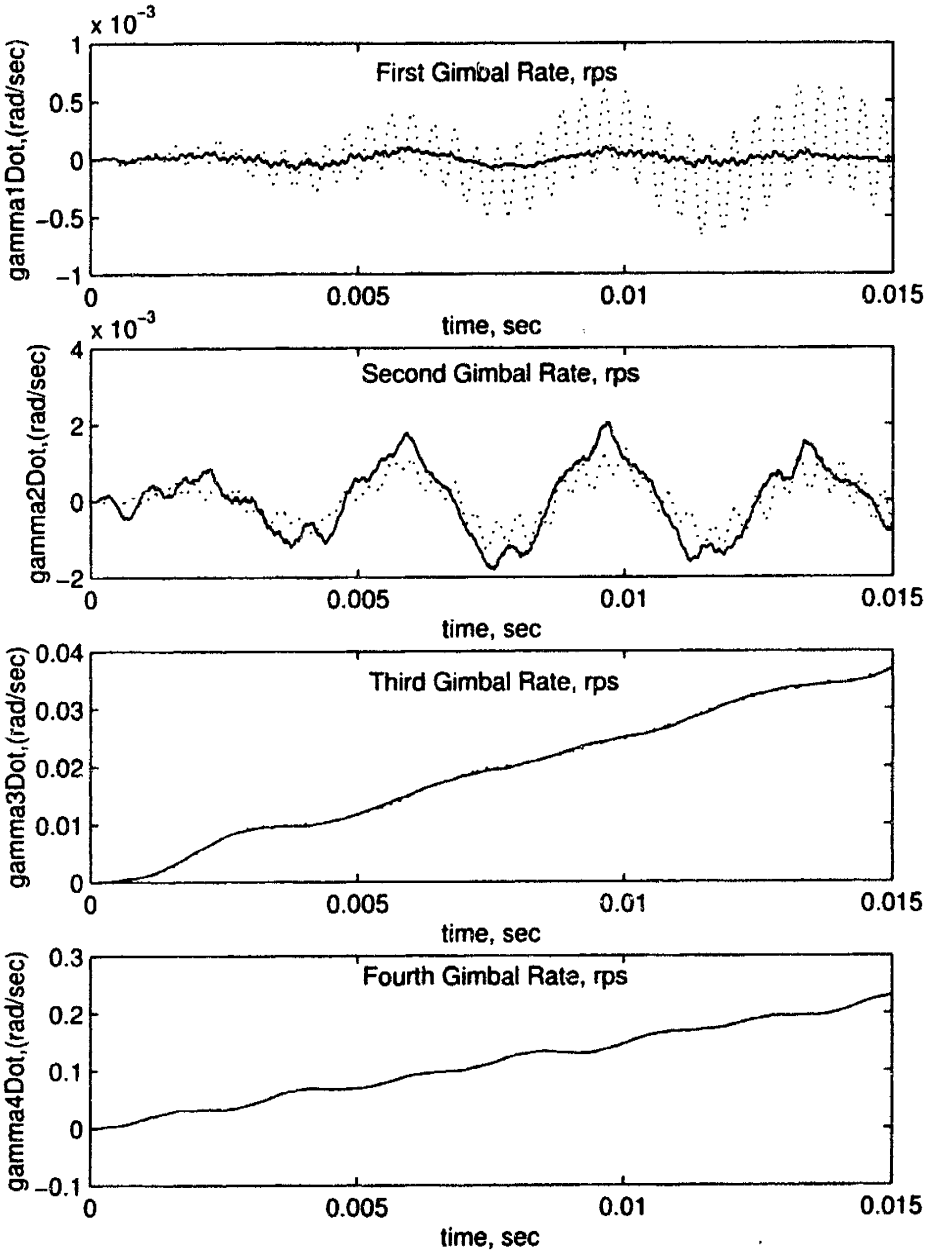


Figure 4-2: Comparison of the Truth Model with 5 Lowest Frequency Modes for Each Body.

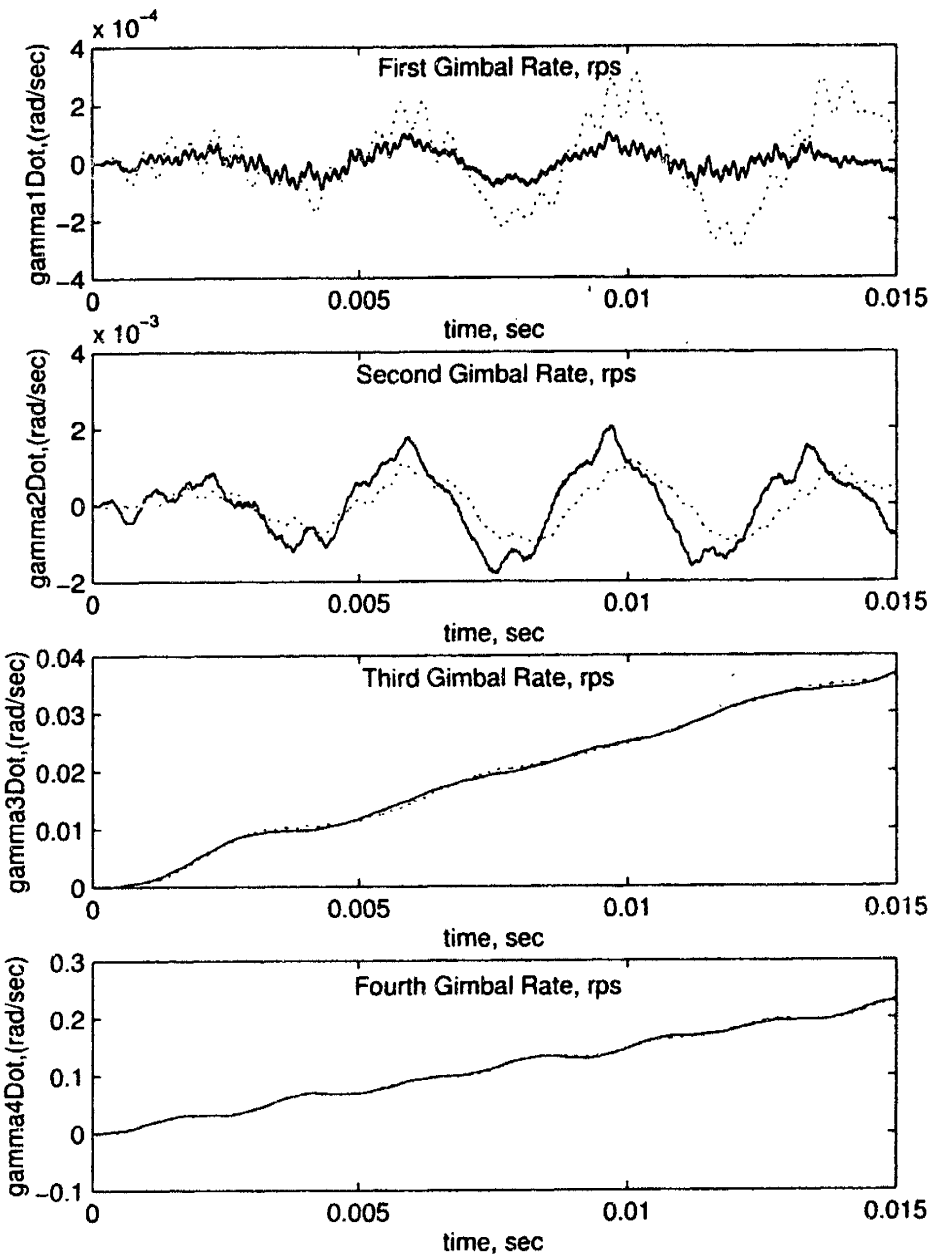


Figure 4-3: Comparison of the Truth Model with 10 Lowest Frequency Modes for Each Body.

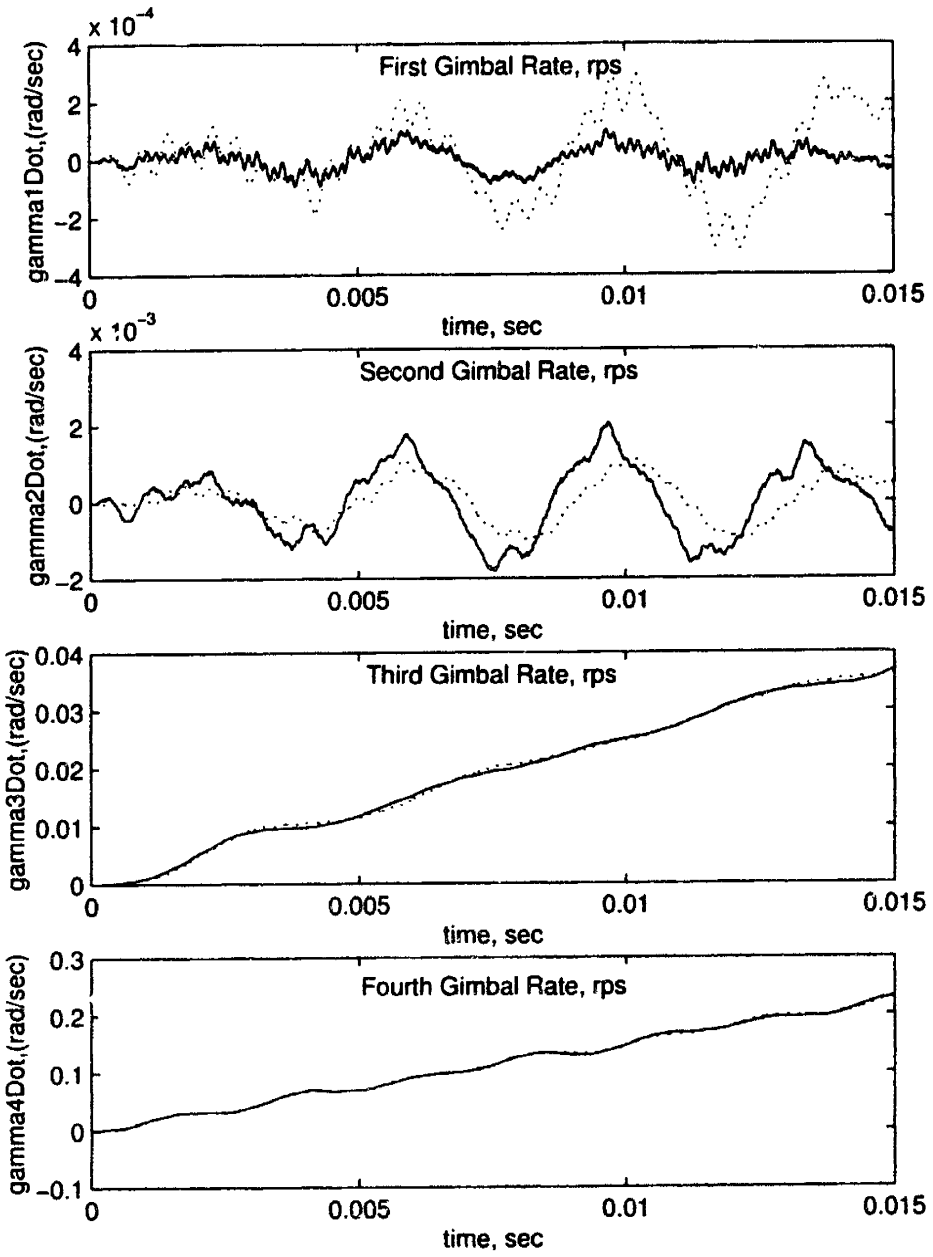


Figure 4-4: Comparison of the Truth Model with 15 Lowest Frequency Modes for Each Body.

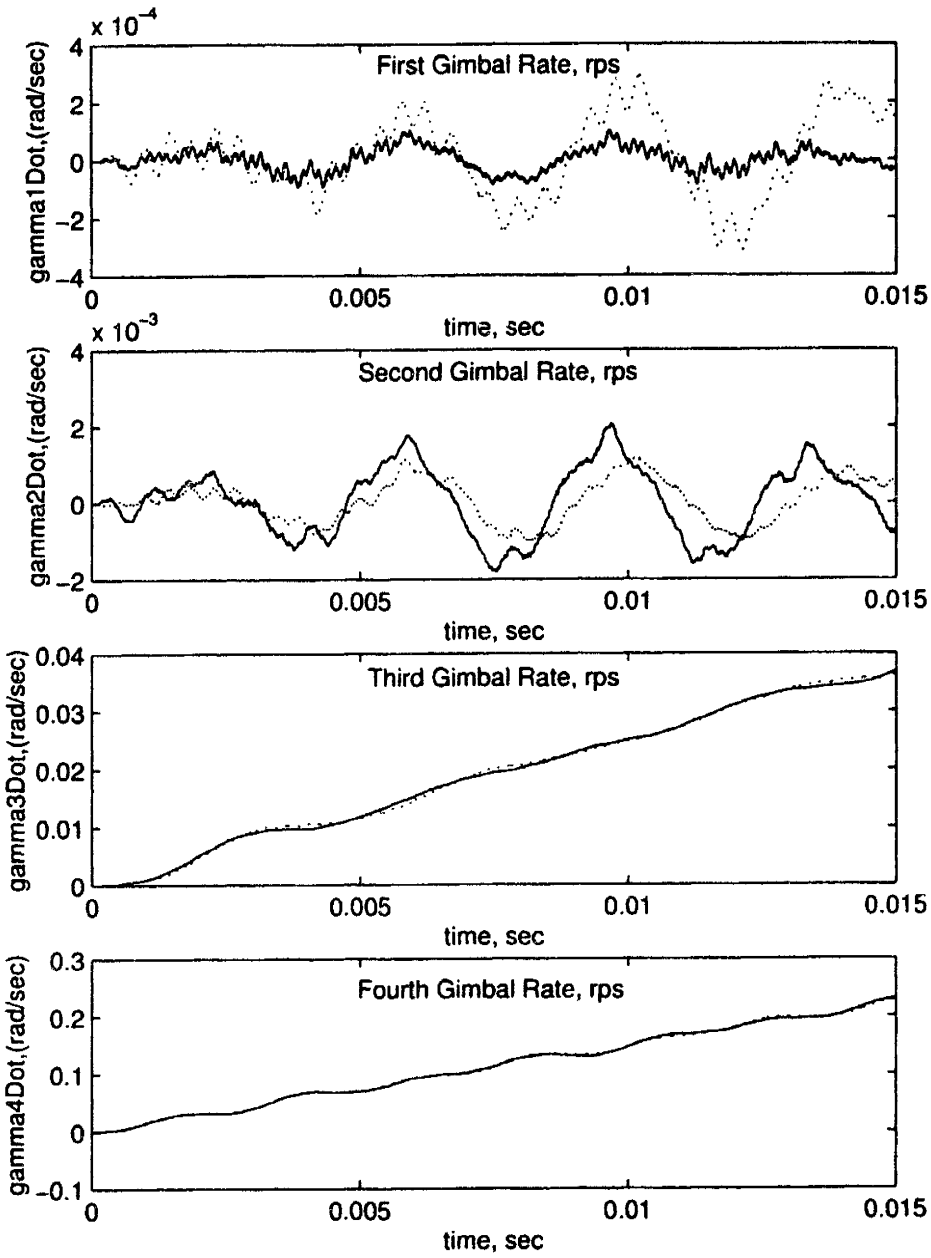


Figure 4-5: Comparison of the Truth Model with 20 Lowest Frequency Modes for Each Body.

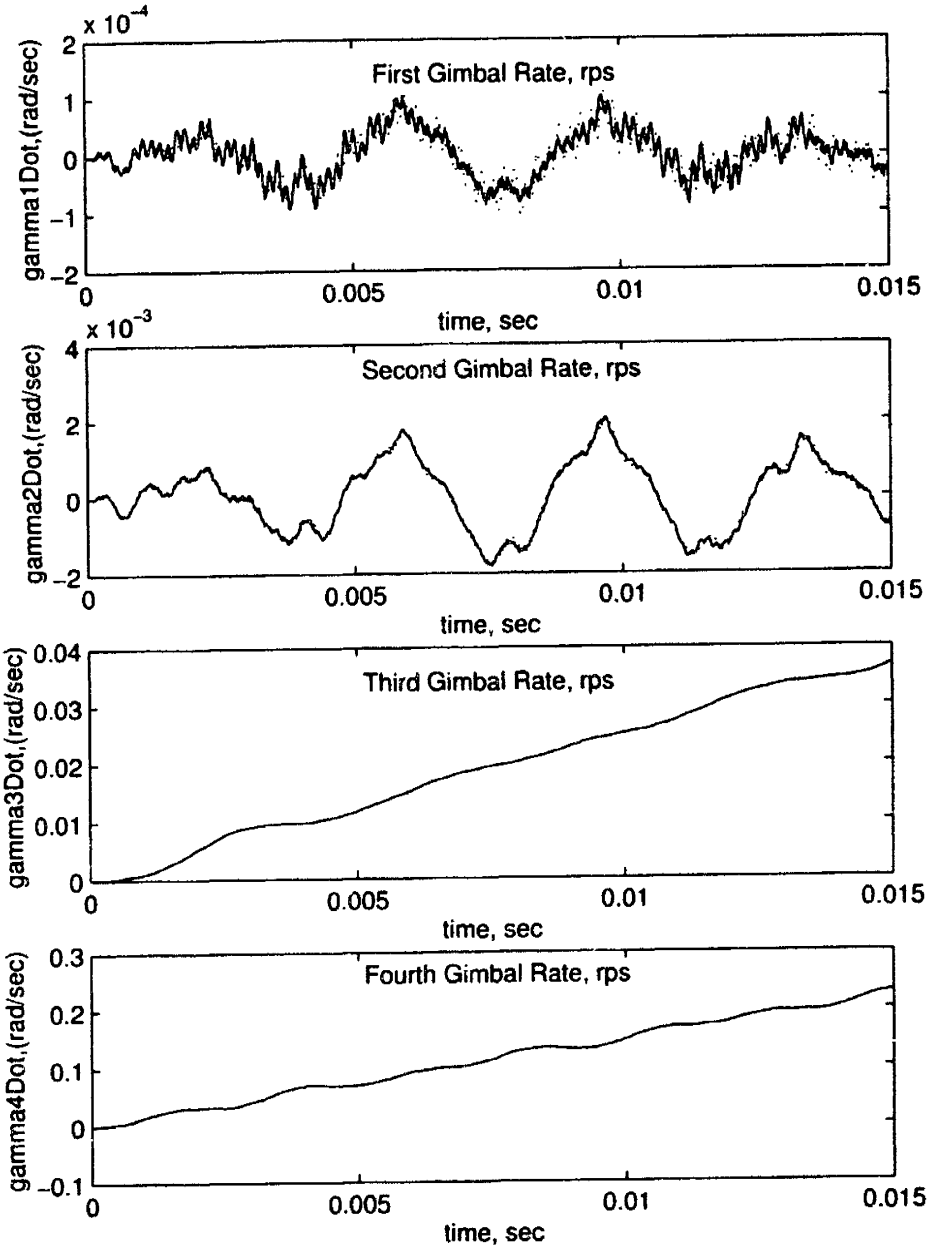


Figure 4-6: Comparison of the Truth Model with 25 Lowest Frequency Modes for Each Body.

Outer Gimbal	Middle Gimbal	Inner Gimbal	Stable Platform
4	4	4	1
3	3	3	2
16	7	13	4
7	1	7	8
1	15	1	3
2	16	19	5
26	2	2	7
27	26	25	10
8	24	20	6
5	8	8	13
11	14	5	11
15	5	11	21
10	11	14	23
17	10	26	12
13	27	10	9
34	13	12	31
12	12	24	24
19	20	15	25
14	17	29	20
6	23	17	33
52	49	38	17
24	40	59	29
22	70	74	45
36	36	66	16
44	6	48	27
81	41	40	28
41	98	28	32
97	92	60	40
57	39	47	38
93	25	75	41
25	71	31	14
58	55	73	39
61	45	6	15
29	29	9	19
39	33	63	47
33	64	62	18
37	28	42	37
51	42	43	22
18	72	32	42
42	83	30	48

Table 4.1: XMX/XMGMX Ranking of the Four Bodies (Most Important to Least Important)

the reduced order model (dashed line), varies greatly depending on the number of ranked modes used. This frequency matches the truth model (solid line) best when the number of modes reaches 25. The results from the frequency ranking are found in Figures 4-7 to 4-11. From the plots, it can be seen that the number of modes increases as the solution approaches that of the truth model. The results differ from that of the frequency analysis in that the reduced order model approaches the truth model when there are less that 25 modes. A more quantitative analysis of this comparison is found in section 4.5.

Using five ranked modes in Figure 4-7, the first gimbal rate of the reduced order model matches the amplitude of the truth model. The low frequency response of the model does not match the truth model well. The frequency of the second gimbal rate also does not correspond to the truth model well. There is significant improvement in the response from using 10 ranked modes. The low frequency of the first gimbal rate approaches that of the 100 mode model. After the improved response from using 15 modes, the 20 ranked mode response matches the truth model with almost the same response as the one using 25 low frequency modes. With 25 ranked modes being used the solution closely resemble the truth model. In all, using this ranking method with a small number of modes produces much better results than the frequency cutoff criteria. After the completion of these comparisons, static modes are used to see if even more accurate results can be obtained when a smaller number of modes are used.

4.3.3 Static and Unranked Vibration Mode Case

The next set of cases use static modes as well as vibration modes. In this section unranked vibration modes are used meaning that the lower frequency modes are used. The static modes are defined by 'imposing a unit forces in the direction of reaction or spring forces that are expected to cause significant deformation of the body.' [13]. The unit force applied to the outer gimbal is found in Figure 4-12. This force is caused by the rotation of the case frame from the initial conditions in section 2.4. Similar static forces are applied to the other gimbals. It is this force that causes deformation at the output joint. This unit force is applied to the resolver location of the next inner body. From this unit force, the static mode was computed using equation 3.48.

The choice of the location of the force is based on the configuration of the problem. For the general multibody equation, cantilever modes are used to describe the flexible motion. For the gimbal system, however, pinned-pinned modes are a more natural way to describe the flexible motion. Since the outside joints and inside joints are related to the single outside frame and inside

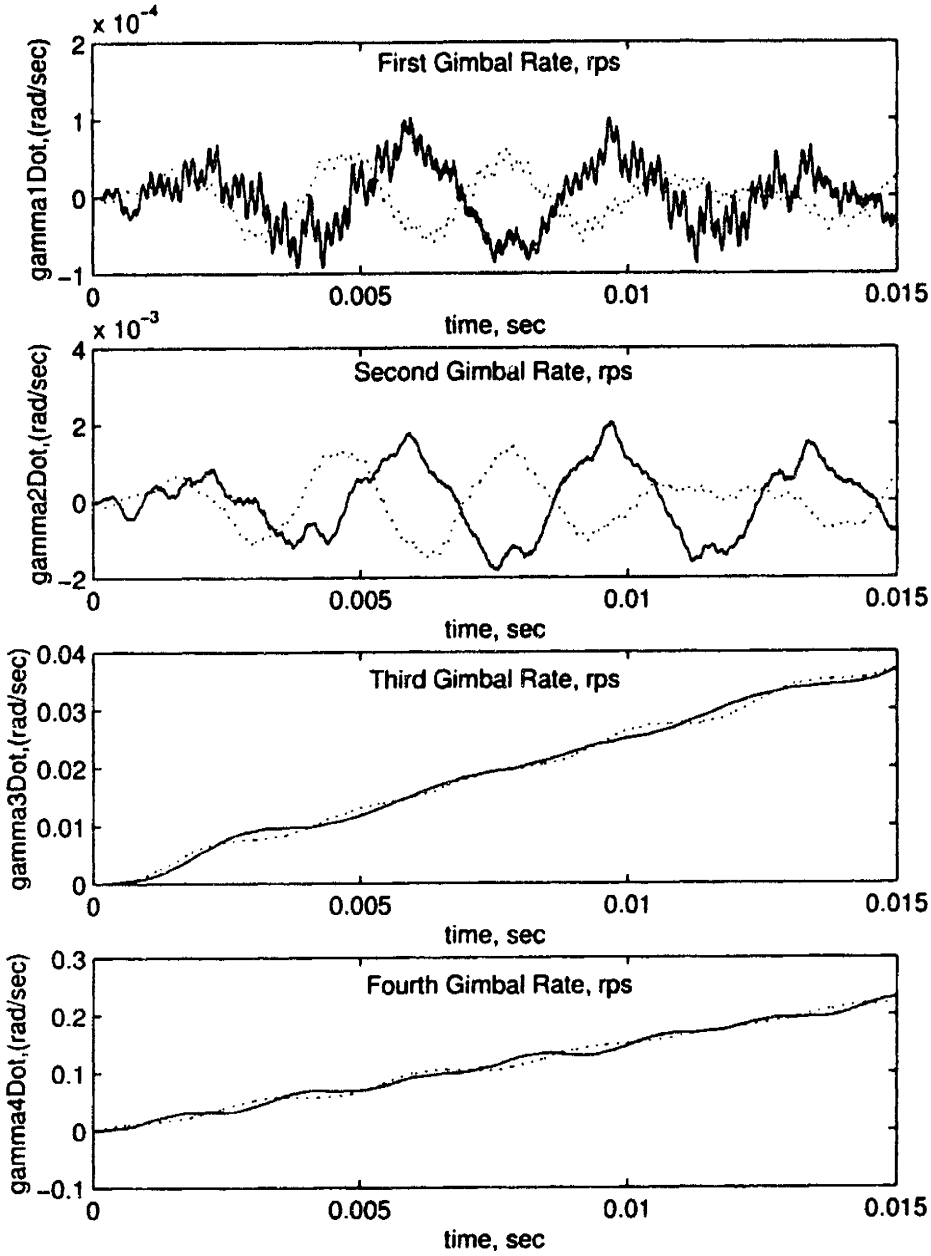


Figure 4-7: Comparison of the Truth Model with 5 Ranked Modes for Each Body.

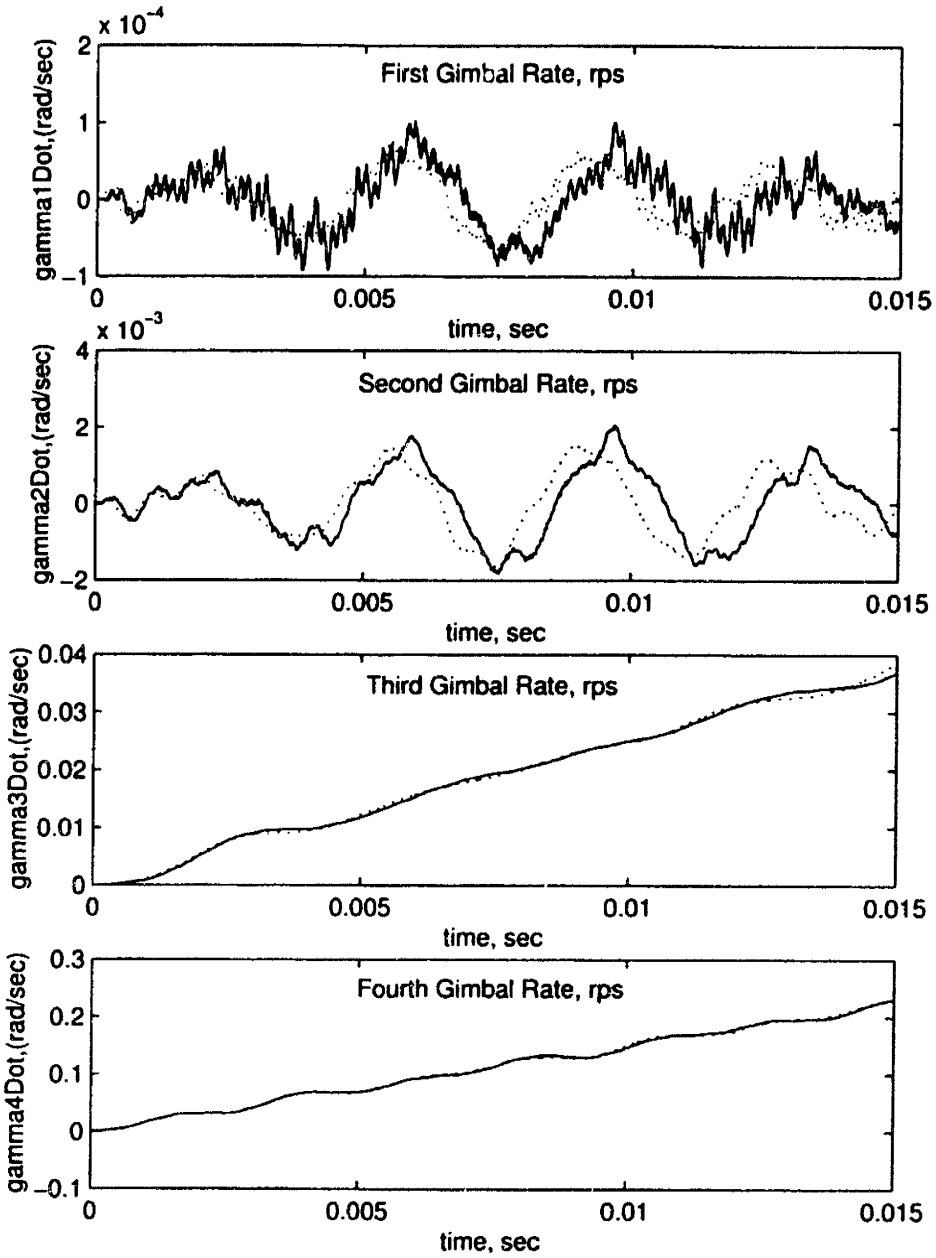


Figure 4-8: Comparison of the Truth Model with 10 Ranked Modes for Each Body.

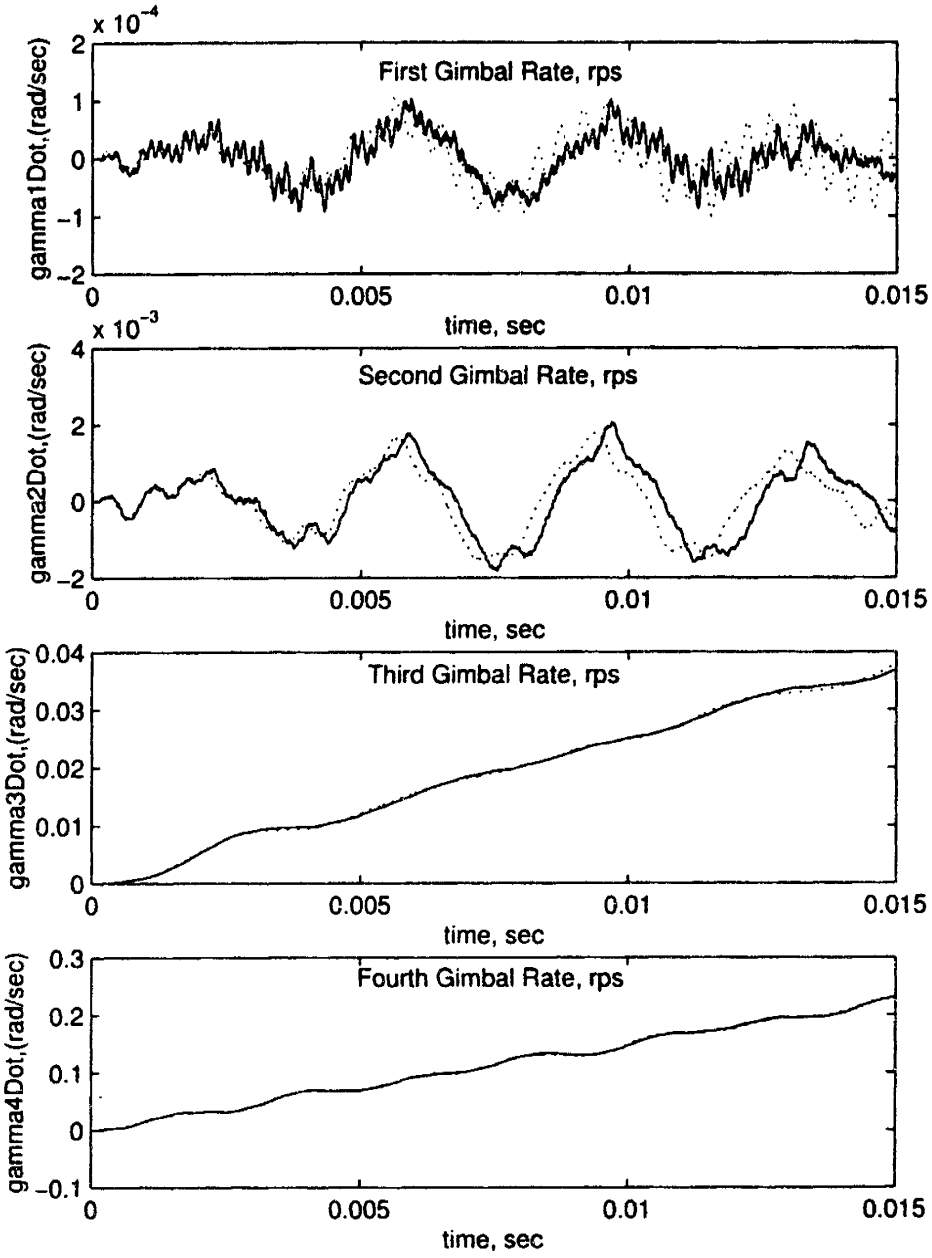


Figure 4-9: Comparison of the Truth Model with 15 Ranked Modes for Each Body.

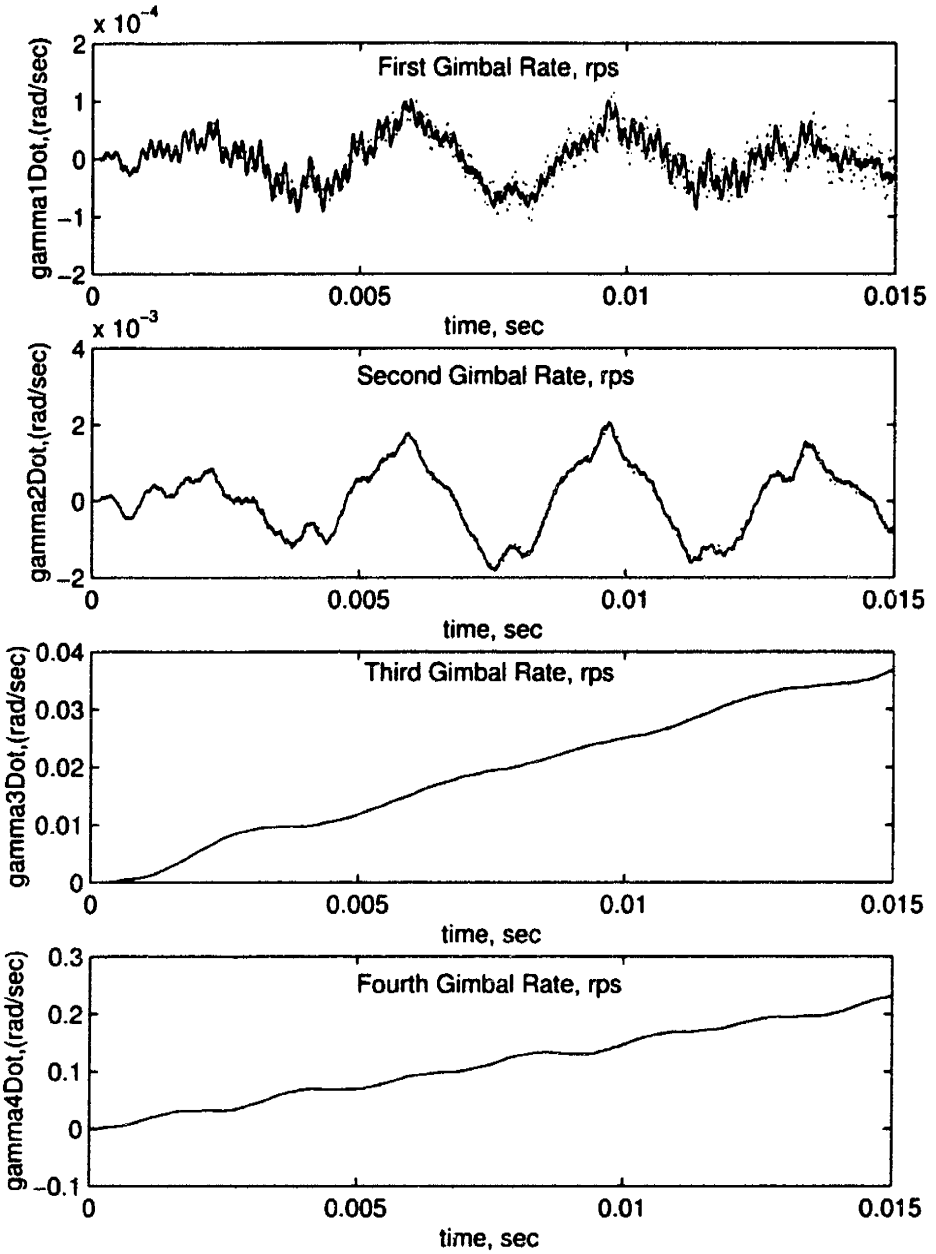


Figure 4-10: Comparison of the Truth Model with 20 Ranked Modes for Each Body.

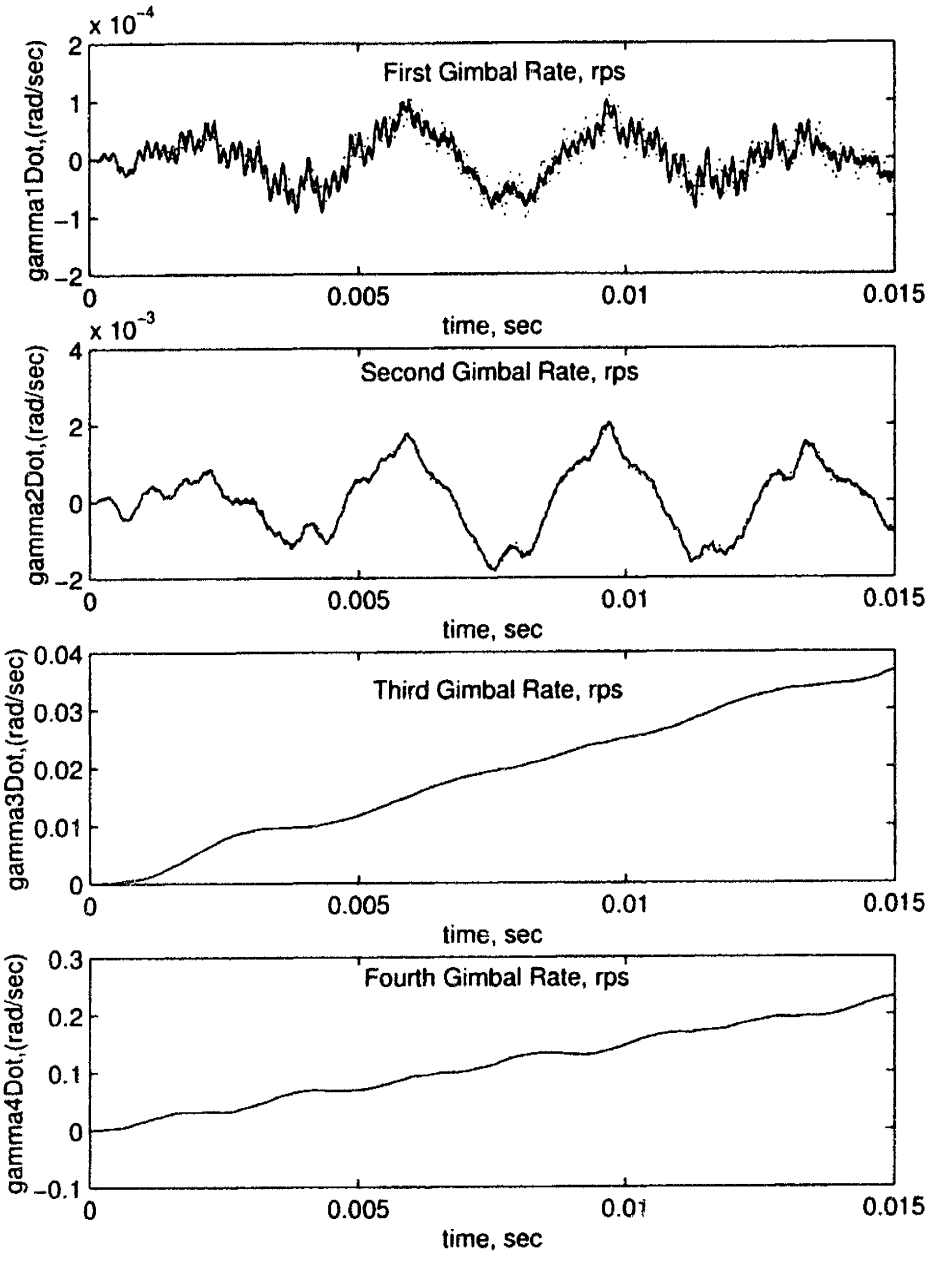


Figure 4-11: Comparison of the Truth Model with 25 Ranked Modes for Each Body.

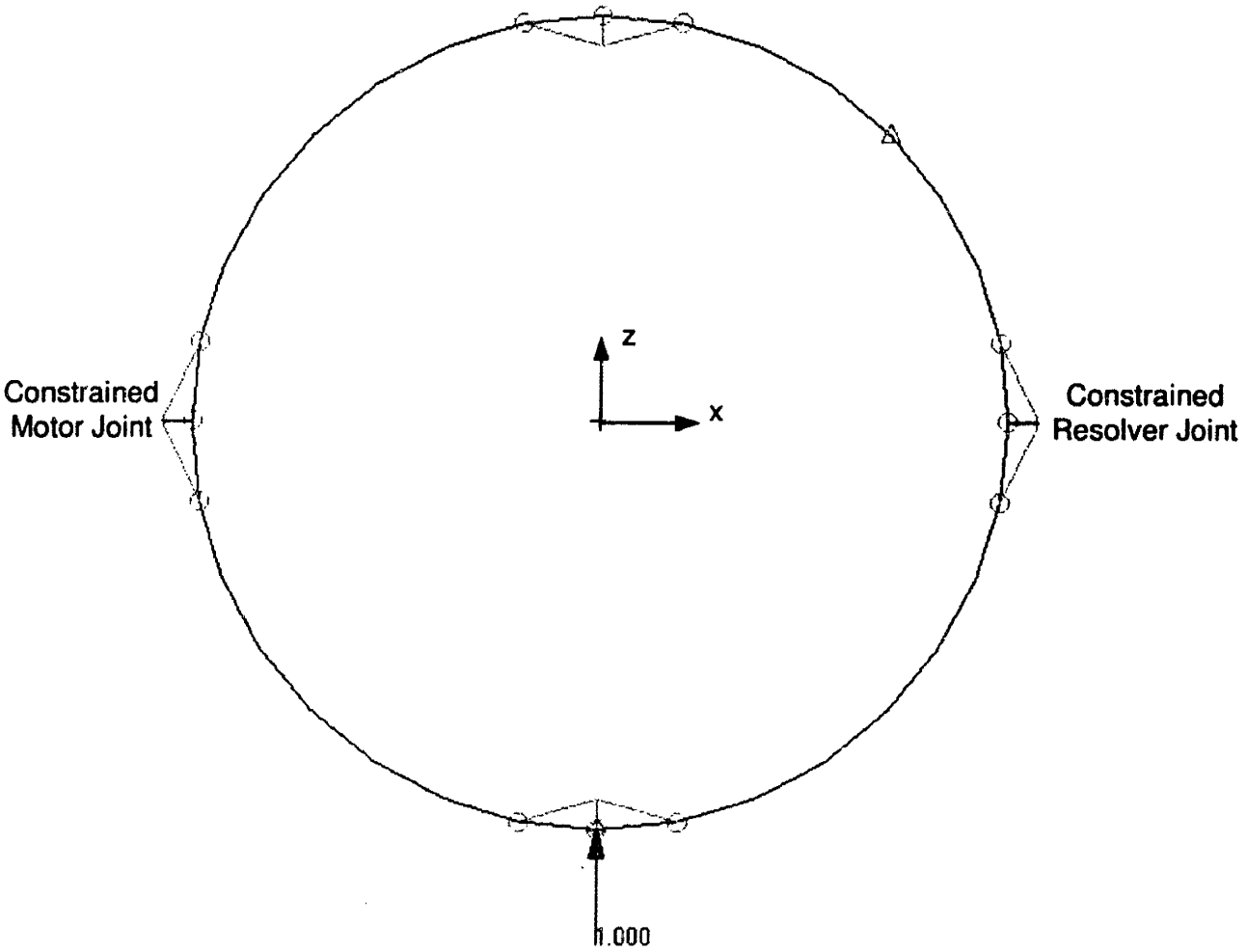


Figure 4-12: Application of Force to the inside bearing of the Outer Gimbal to Determine Static Correction Mode

frame, it is only necessary to apply a force at either one outside joint or one inside joint to obtain the deflection of the body. The use of different static modes are explained in section 4.4.

The results of using 2 through 8 low frequency modes and 1 static correction or attachment mode are found in Figures 4-13 through 4-16. The results for this case show good comparisons with the truth model despite the use of low a low number of modes. This method is useful for cases in which the simulation is run for a large amount of time.

When using a low number of modes, the use of a static mode produces more accurate results than if one is not used. The number of generalized coordinates is reduced to half of the number used in the frequency criteria and XMX/XMGMX ranking criteria. The results from using 2, 4 and six modes, are better than using up to 10 modes from the previous section. The last comparison using 8 lowest frequency modes and a static mode compares favorably to the truth model and to the model reduction methods using 25 modes. It is advantageous to use this combination of modes because the number of generalize coordinates can be reduced by 10 and the results can be near the truth model.

4.3.4 Static and Ranked Vibration Mode Case

To obtain a more efficient solution method for solving such a large system of equations, the model reduction techniques of section 4.3.2 and section 4.3.3 are applied together with static correction modes. The plots for these cases are found in Figures 4-17 to 4-20.

Using both ranked modes and the static correction mode does not produce better results than when unranked modes are used. From the rankings in Table 4.1 some of the high frequency modes are used. Using the static mode makes using the higher frequency modes redundant. Thus, the results from using this method do not improve as much as when the simulation is run using a static mode and the lower frequency modes.

4.4 Model Reduction with Different Static Modes

From the previous section the use of static modes provides good results when using half of the number of modes used in the ranking methods. This section describes the effects of using different types of static correction or attachment modes. Using one static mode for the outer gimbal and ten normal modes for all the bodies produced accurate results. Only a static mode for the outer gimbal is used because that is where most of the deformation takes place. Using the ten lowest

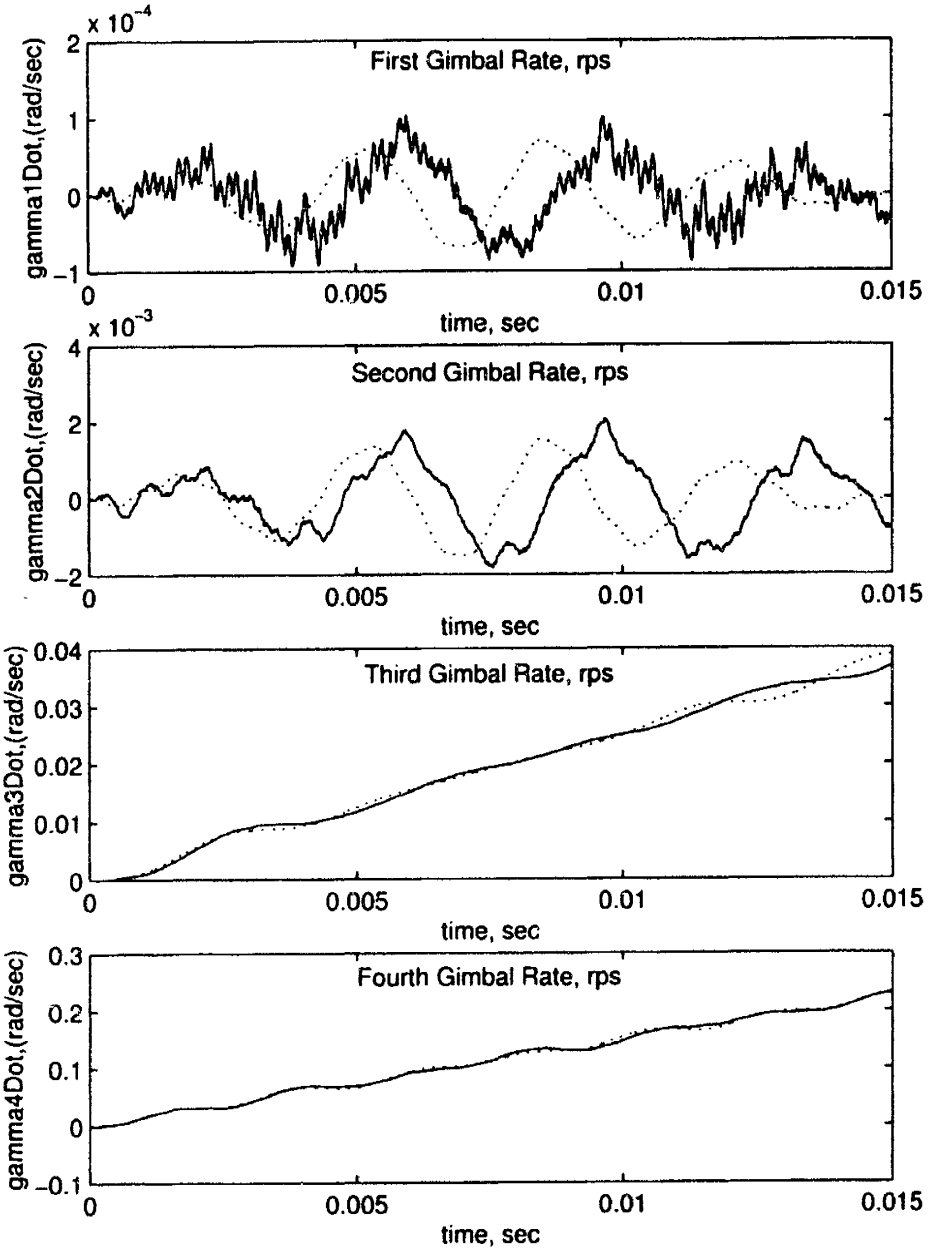


Figure 4-13: Comparison of the Truth Model with 2 Normal Modes and One Static Mode for Each Body.

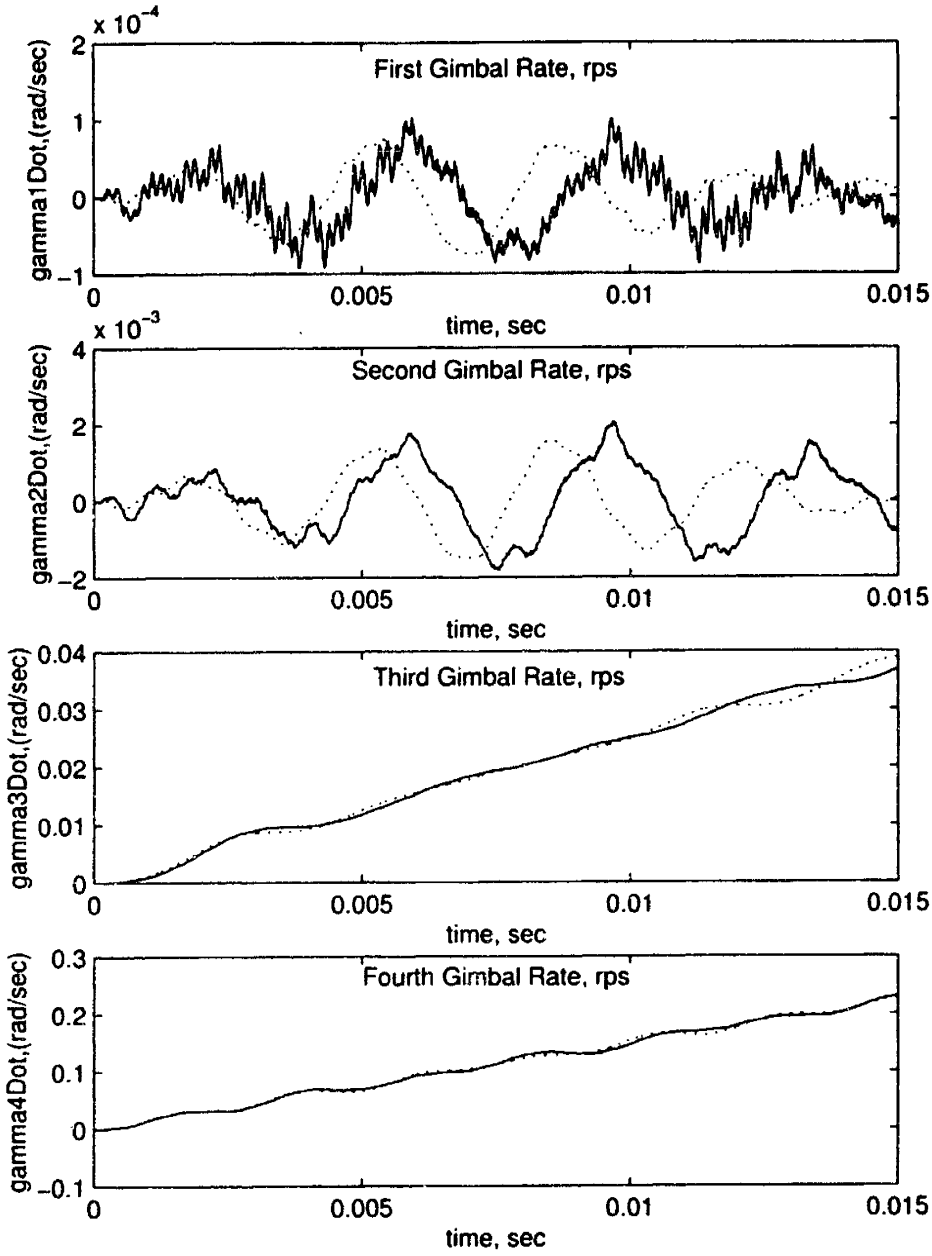


Figure 4-14: Comparison of the Truth Model with 4 Normal Modes and One Static Mode for Each Body.

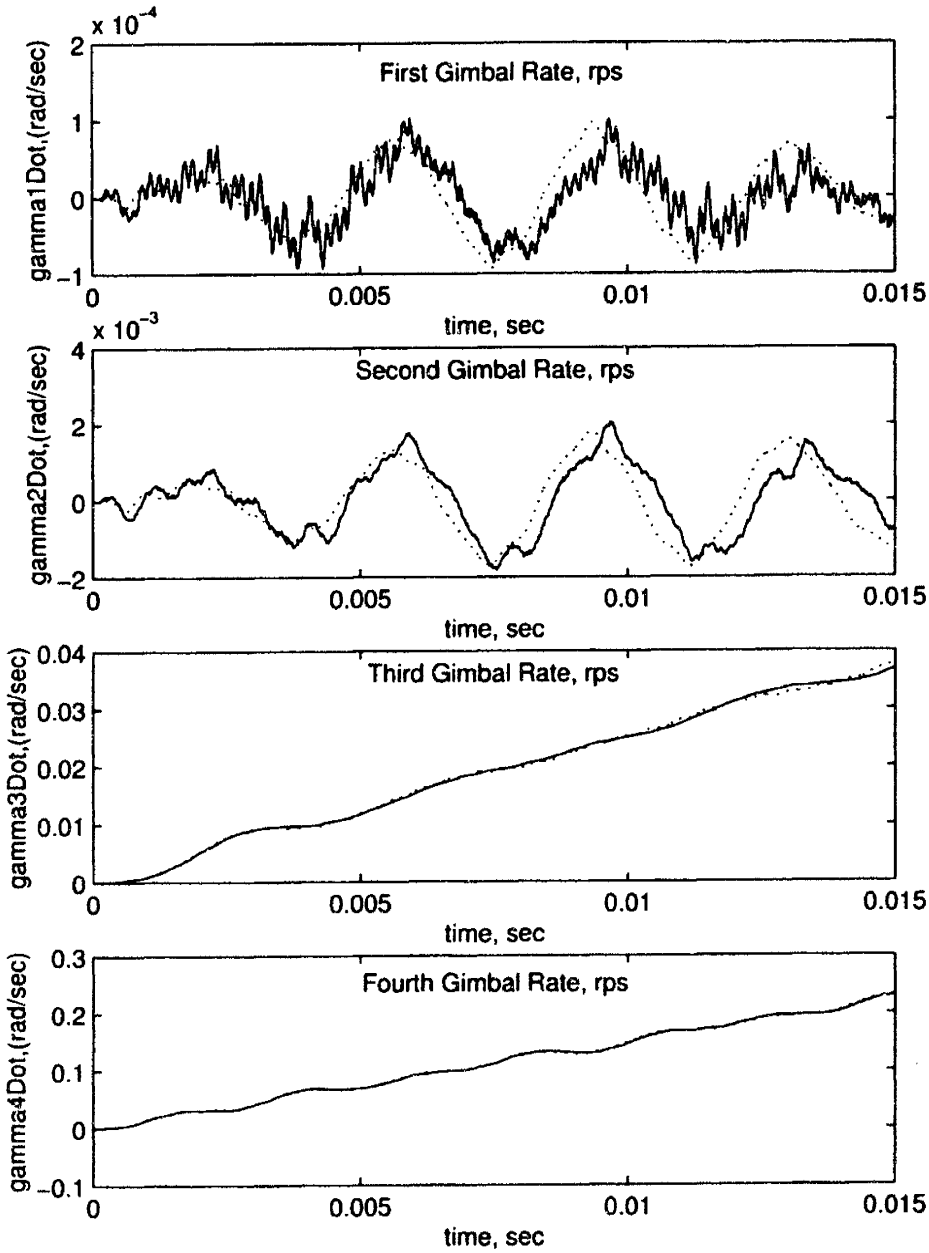


Figure 4-15: Comparison of the Truth Model with 6 Normal Modes and One Static Mode for Each Body.

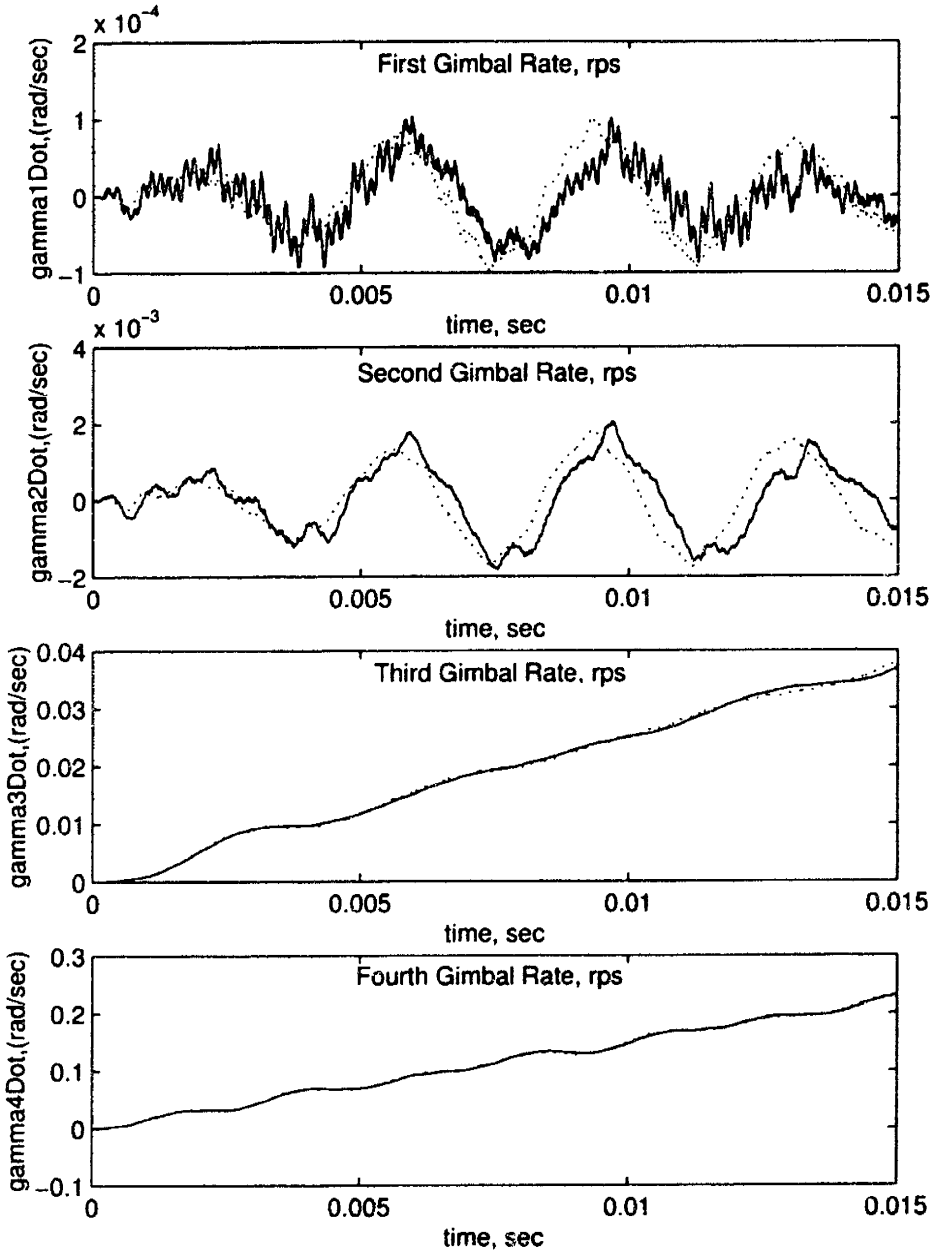


Figure 4-16: Comparison of the Truth Model with 8 Normal Modes and One Static Mode for Each Body.

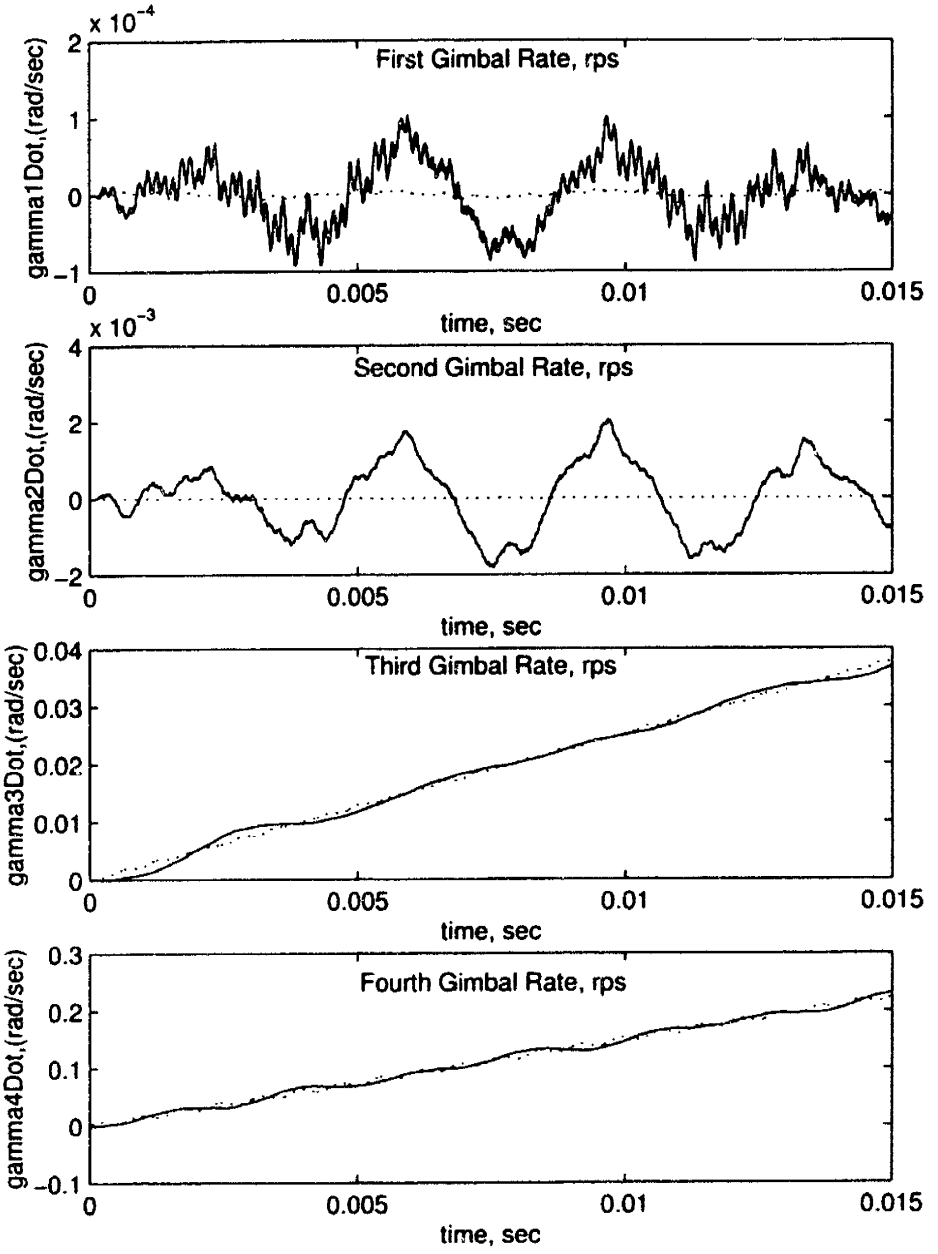


Figure 4-17: Comparison of the Truth Model with 2 Ranked Modes and One Static Mode for Each Body.

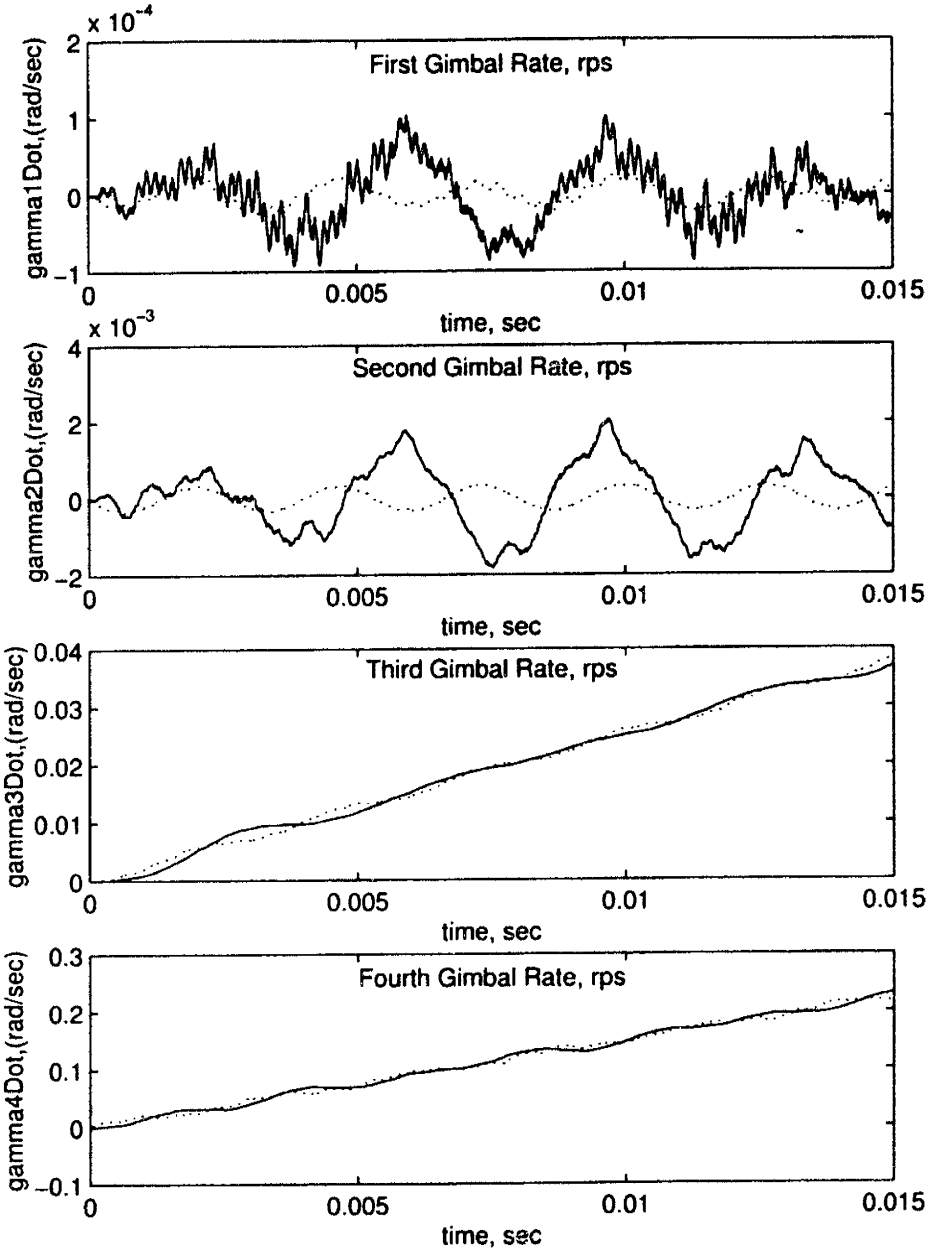


Figure 4-18: Comparison of the Truth Model with 4 Ranked Modes and One Static Mode for Each Body.

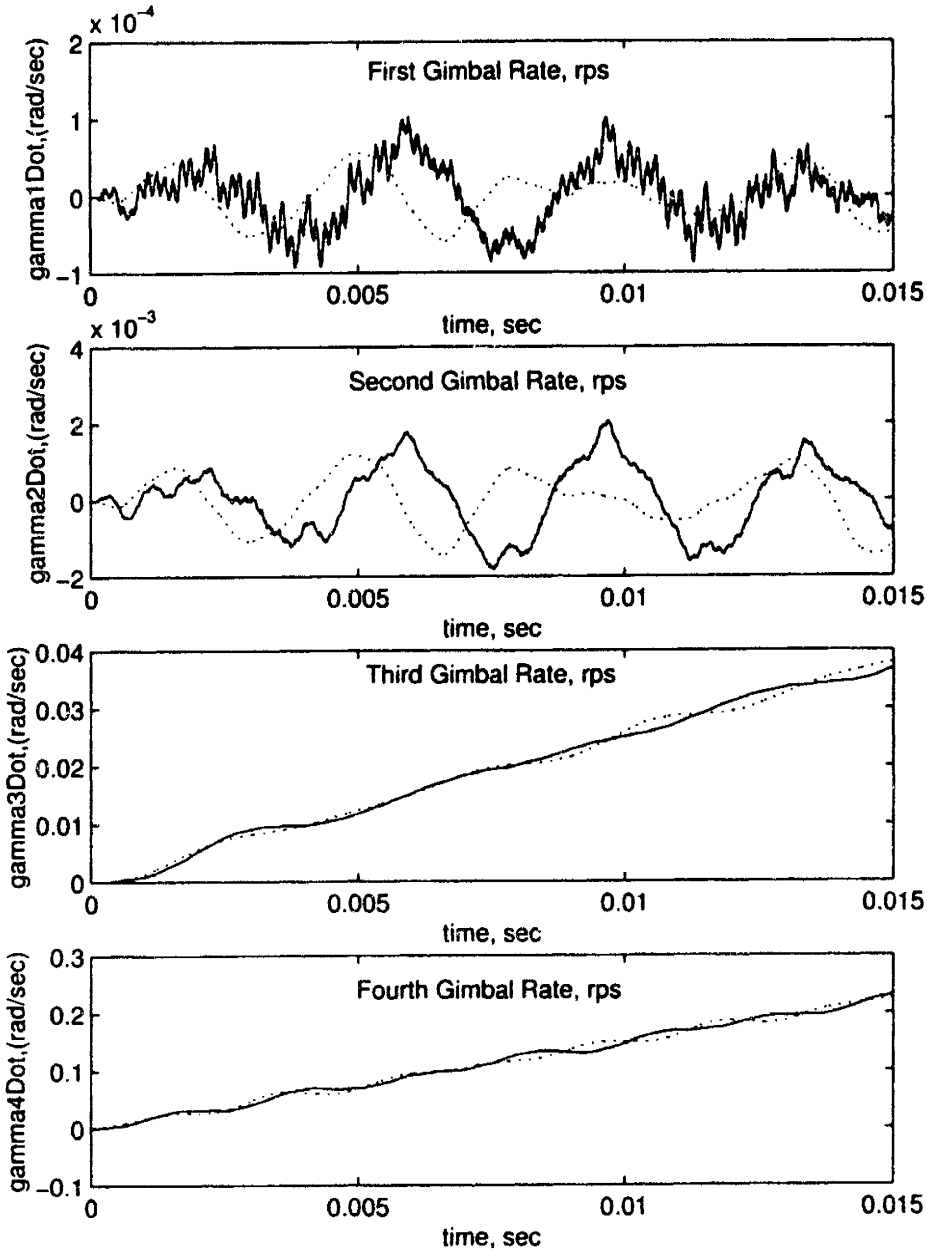


Figure 4-19: Comparison of the Truth Model with 6 Ranked Modes and One Static Mode for Each Body.

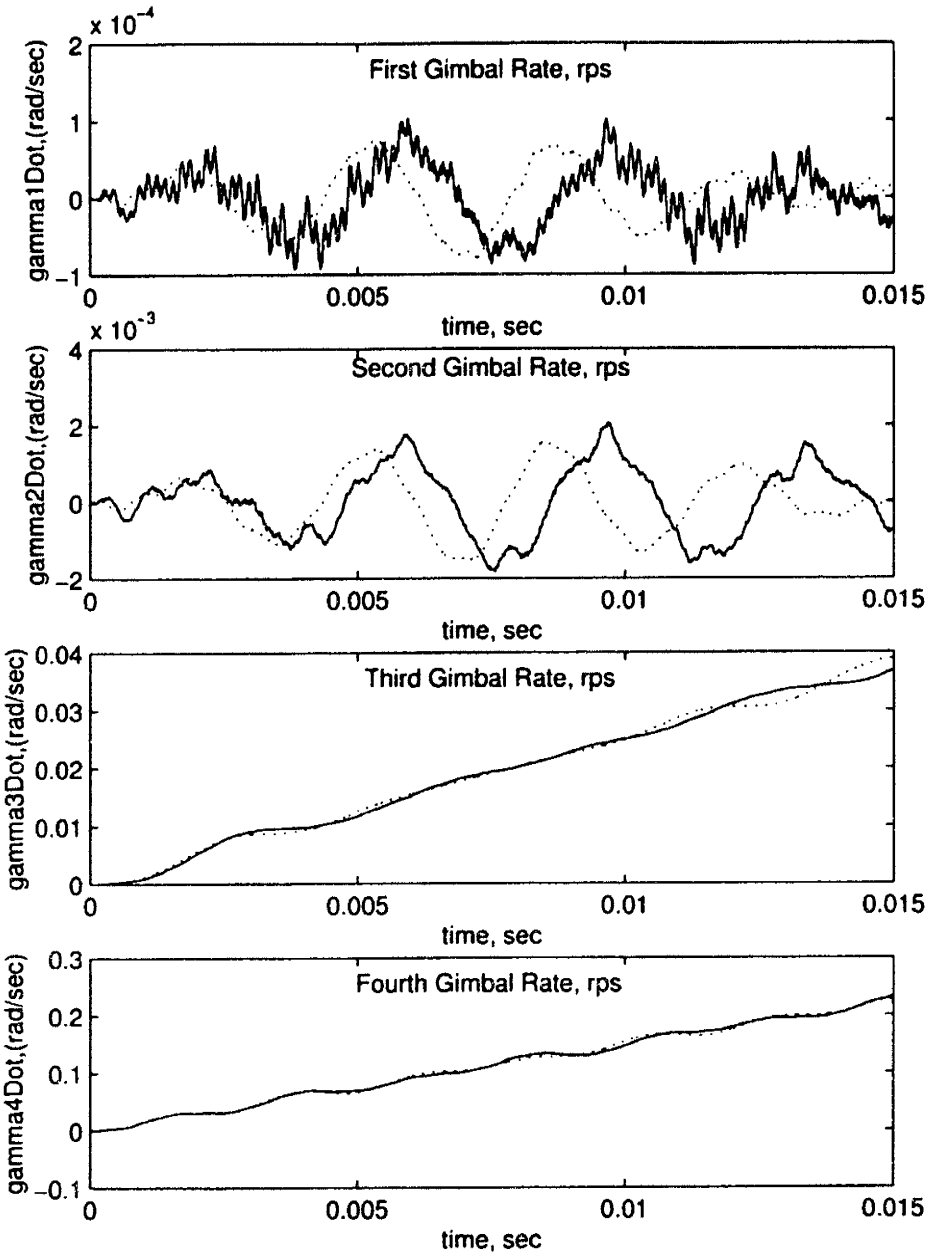


Figure 4-20: Comparison of the Truth Model with 8 Ranked Modes and One Static Mode for Each Body.

frequency modes, Figure 4-3, the low frequency response tends to match the truth model, but the high frequency response deviates from that of the truth model. Using one static mode in addition to the 10 normal modes, produces results of $\dot{\gamma}$ that are within twenty percent of the truth model. Figures 4-21 through 4-27 show the results of these test cases. These test cases were chosen due to the fact that they result in the largest deformation because of the boundary conditions.

The case in Figures 4-21 and 4-22 involves a static mode in which a torque is applied at the output motor joint in the y-direction and z-direction, respectively. Figure 4-23 uses a static mode in which a force is applied in the x-direction. The next figure is the same case as discussed in the previous section in which a force in the z direction is applied to the output motor bearing. Figure 4-25 and 4-26 is similar to the first two cases except that the torques are applied at the input motor bearing. The last figure, Figure 4-27, uses a static mode in which a force is applied along the drive axis of the outer gimbal. It should be noted that using these static modes produce similar results for each case. The reason for this might be explained through further research of this gimbal problem.

Once the cases are run it is important to quantify the error. The next section explains the error in each of the model reduction methods.

4.5 Error of Model Reduction Methods

From the results from the previous section, qualitative comparisons are made with the truth model. From the plots, it is only possible to see qualitatively which combinations of modes produce adequate results. Another method for determining the accuracy of the results is needed. This is done with the use of equation 3.53 in section 3.4. The error of the ranking methods is found in Figure 4-28 while the error using a static mode is found in Figure 4-29. The data used to make these plots is found in Table 4.2 and 4.3.

Using the error measure, the XMX/XMGMX ranking method produces better results using a smaller number of modes than using the unranked lower frequencies in the previous section. At 25 modes however, the solutions appear to be the same. This is a result of the fact that the modes used in the ranking method and the frequency cutoff method are almost the same. The error measure using a static mode and unranked modes produced more consistent results than when a static mode and ranked modes were used.

From the results in this section, the model reduction of the set of nested gimbals can be accom-

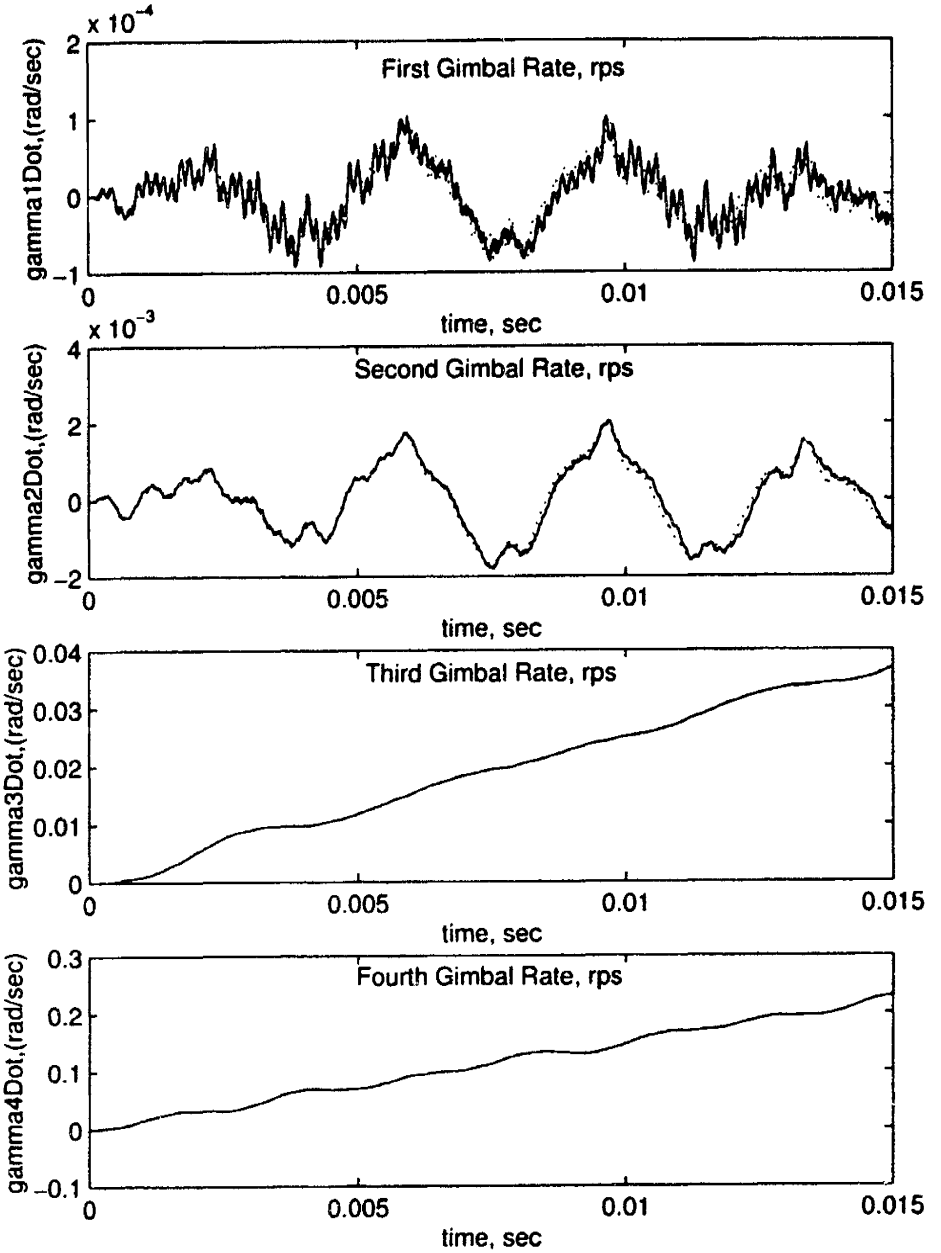


Figure 4-21: Comparison of the Truth Model with Static Mode Using y-Moment Applied to Output Motor Joint.

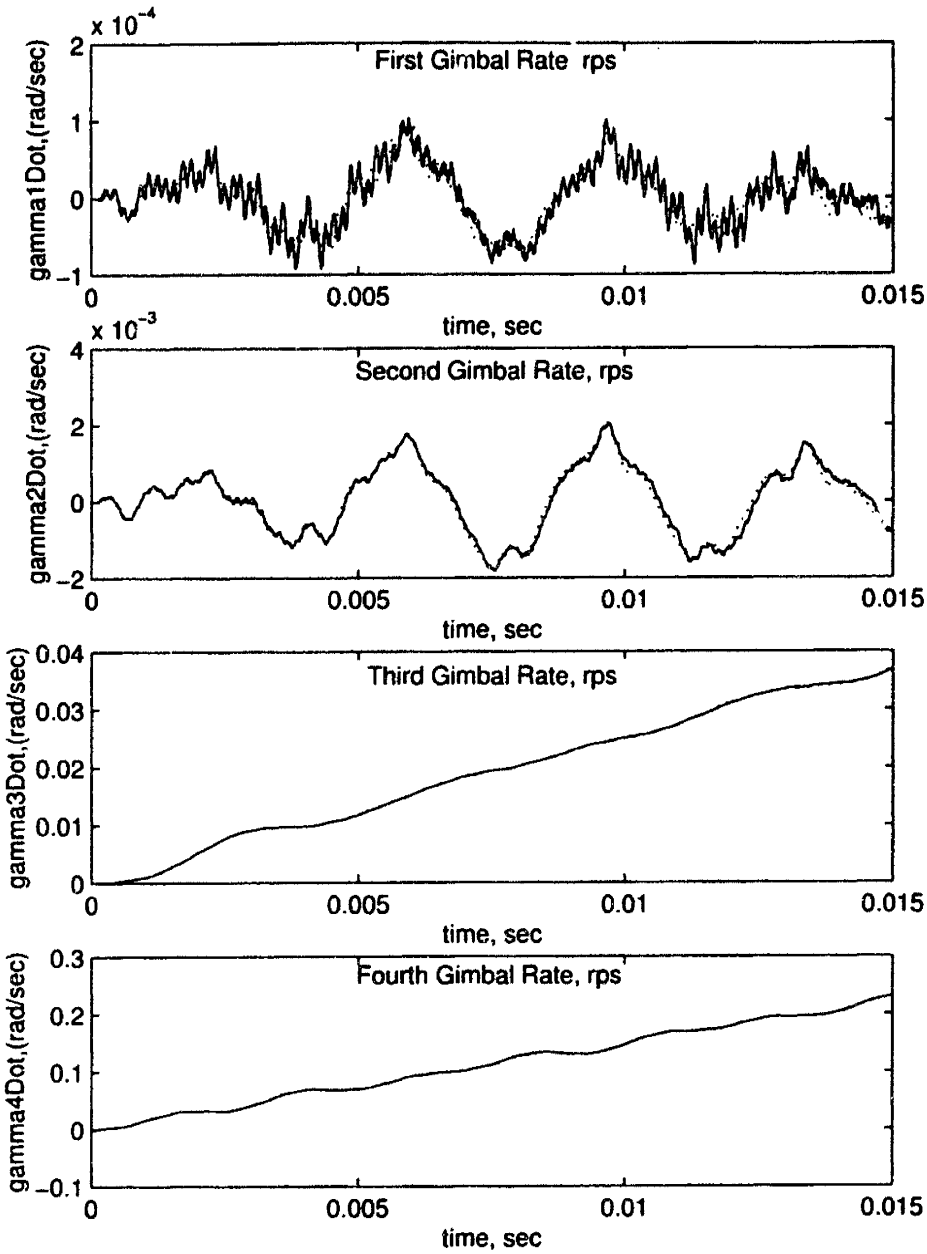


Figure 4-22: Comparison of the Truth Model with Static Mode Using z-Moment Applied to Output Motor Joint.

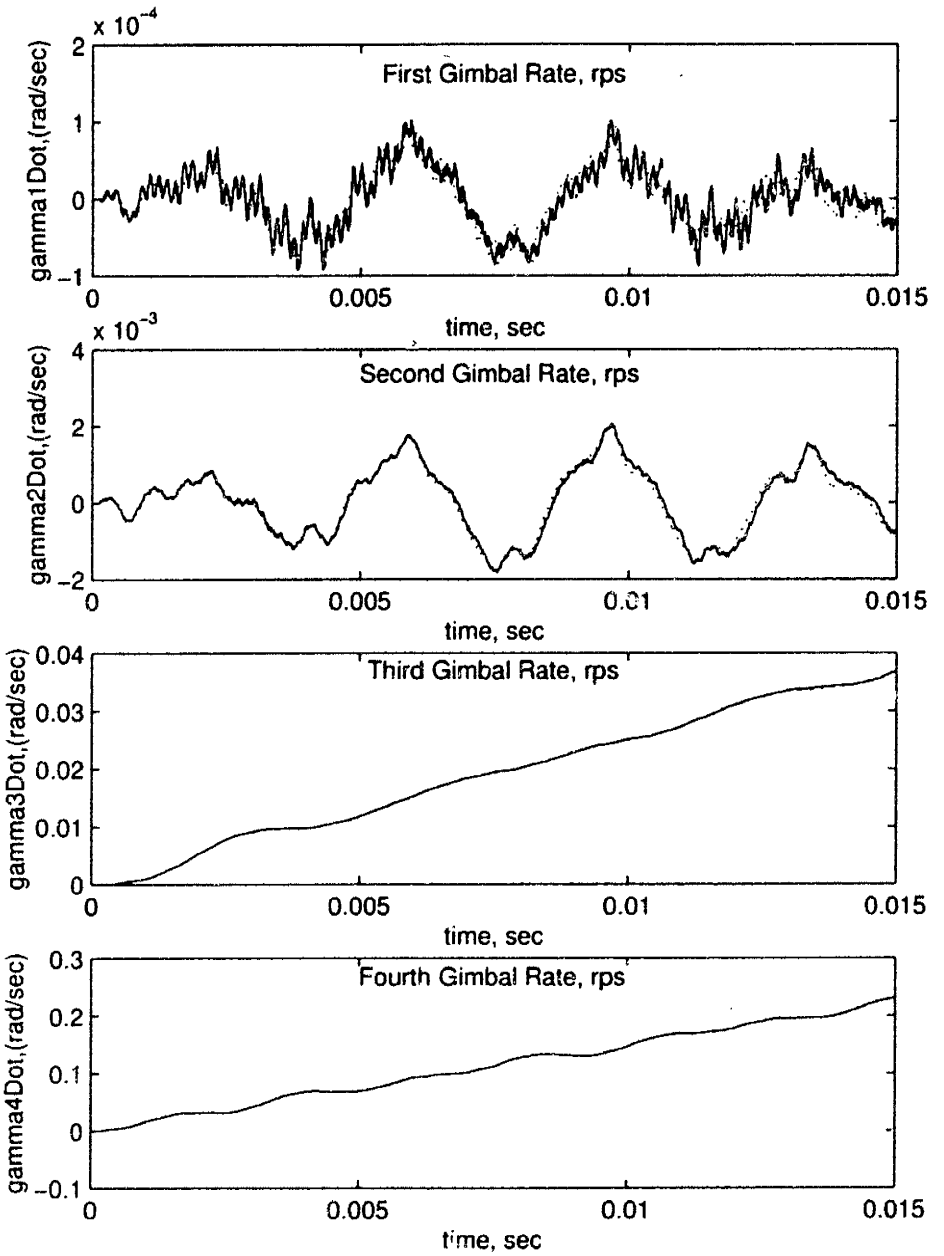


Figure 4-23: Comparison of the Truth Model with Static Mode Using x-Force Applied to Output Motor Joint.

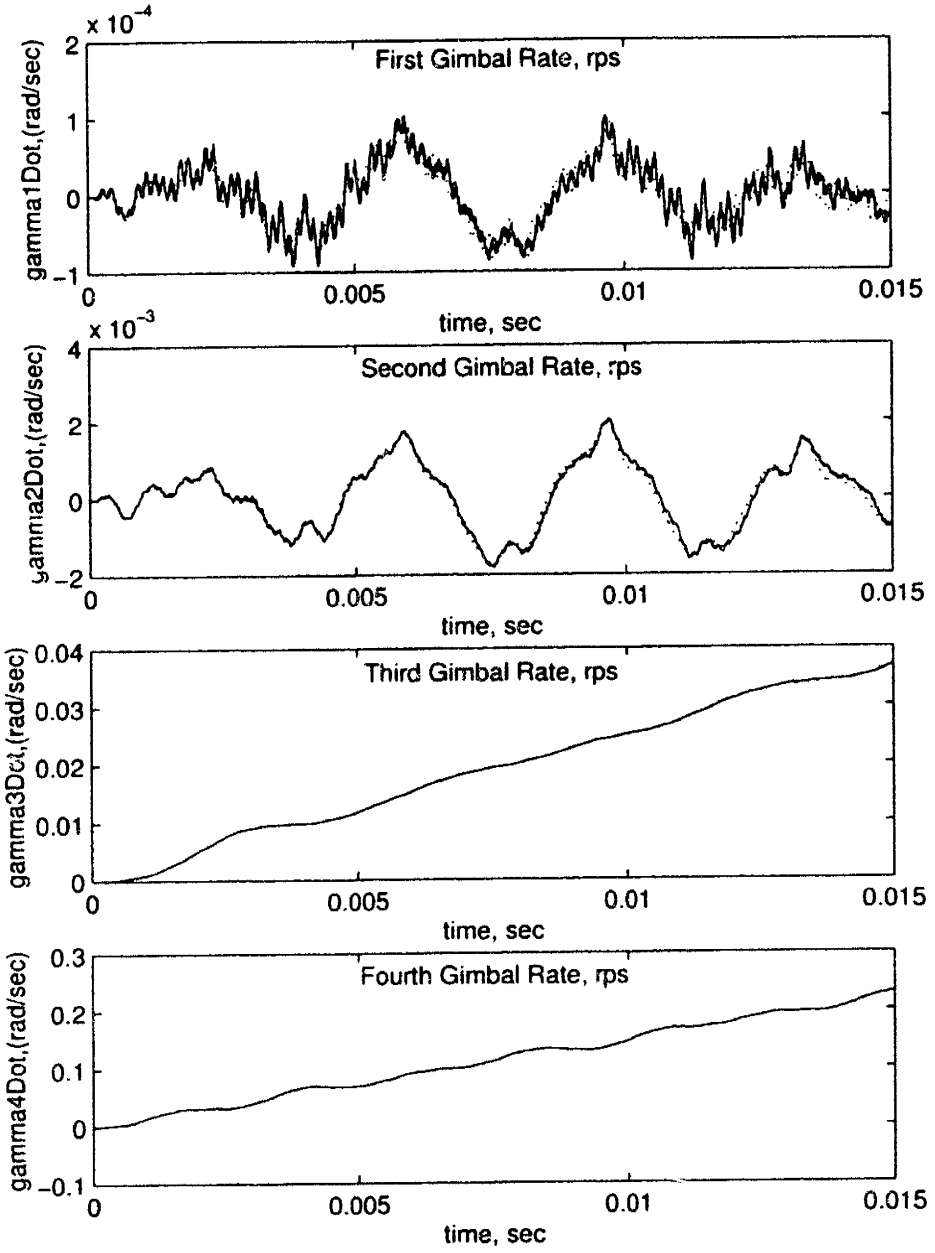


Figure 4-24: Comparison of the Truth Model with Static Mode Using z-Force Applied to Output Motor Joint.

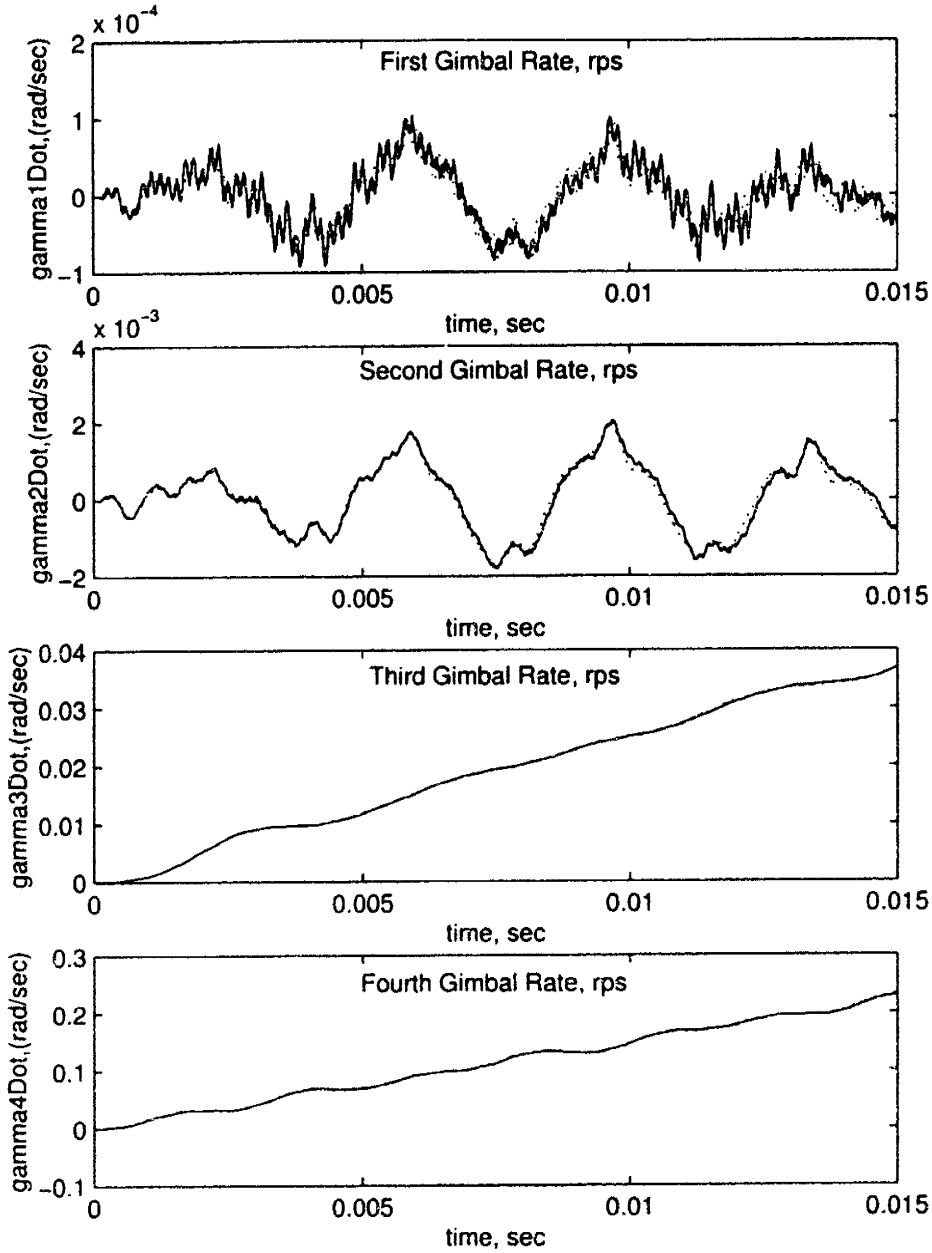


Figure 4-25: Comparison of the Truth Model with Static Mode Using y-Moment Applied to Input Motor Joint.

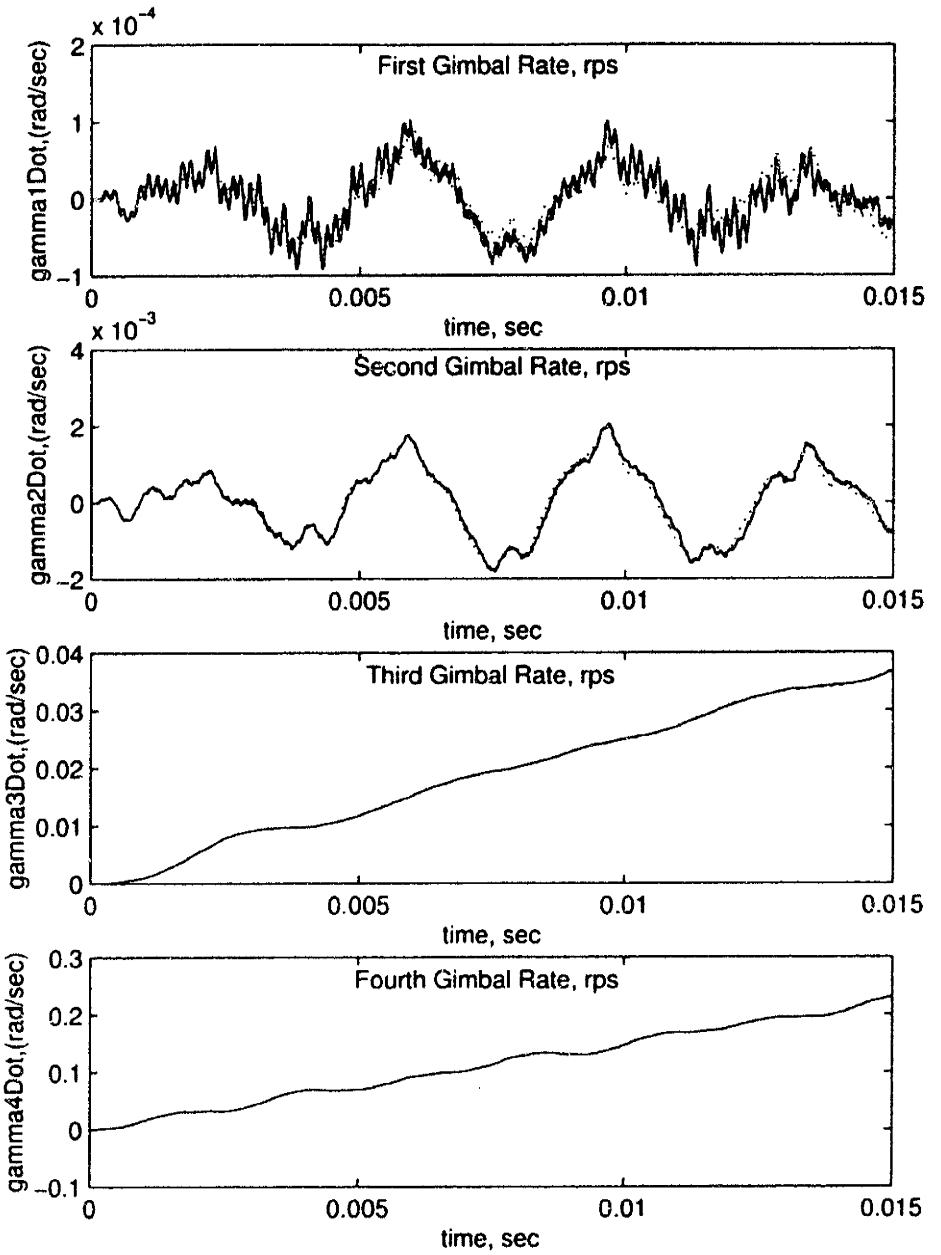


Figure 4-26: Comparison of the Truth Model with Static Mode Using z-Moment Applied to Input Motor Joint.

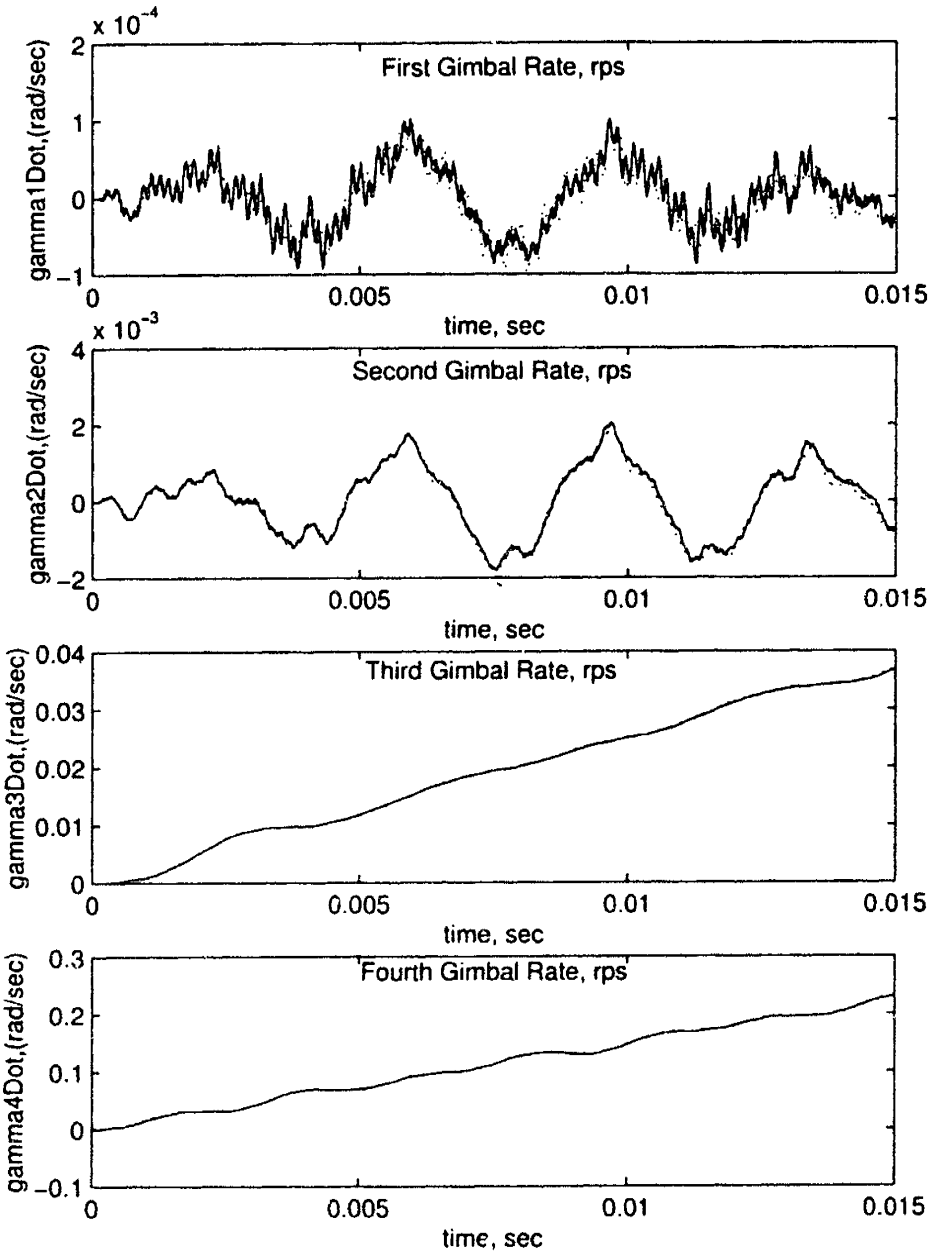


Figure 4-27: Comparison of the Truth Model with Static Mode Using x-Force Applied to Input Motor Joint.

Error Using Frequency Cutoff Criteria

	5 Modes	10 modes	15 modes	20 modes	25 modes
Outer Gimbal	36.6137	8.4256	8.5771	8.5861	.2935
Middle Gimbal	.2906	.4333	.4819	.4890	.0077
Inner Gimbal	9.4628×10^{-5}	3.2343×10^{-4}	3.693×10^{-4}	3.7412×10^{-4}	3.5410×10^{-6}
Stable Platform	4.047×10^{-3}	1.5972×10^{-4}	1.8865×10^{-4}	1.9149×10^{-6}	7.3453×10^{-6}

Error Using XMX/XMGMX Ranking Criteria

	5 Modes	10 modes	15 modes	20 modes	25 modes
Outer Gimbal	2.0725	.6494	.9382	.4348	.2935
Middle Gimbal	2.3429	.5355	.2886	.0078	.0077
Inner Gimbal	.0015	4.86×10^{-4}	1.9353×10^{-4}	5.713×10^{-6}	3.5414×10^{-6}
Stable Platform	.0032	2.0409×10^{-4}	1.0435×10^{-4}	9.897×10^{-6}	7.3482×10^{-4}

Table 4.2: Error Using Frequency Cutoff Criteria and Mode Ranking

Error Using 1 Static Mode and Unranked Modes

	2 Modes	4 modes	6 modes	8 modes
Outer Gimbal	1.8254	1.6109	.6264	.6383
Middle Gimbal	1.6275	1.6218	.3740	.3731
Inner Gimbal	.0020	.0021	3.5112×10^{-4}	3.4677×10^{-4}
Stable Platform	5.84×10^{-4}	6.4667×10^{-4}	1.146×10^{-4}	1.1199×10^{-6}

Error Using 1 Static Mode and Ranked Modes

	2 Modes	4 modes	6 modes	8 modes
Outer Gimbal	.9226	.9684	1.3165	1.6020
Middle Gimbal	.9684	1.0471	1.4725	1.5996
Inner Gimbal	.0014	.0012	.0081	.0020
Stable Platform	.0018	.0021	.0027	6.3717×10^{-4}

Table 4.3: Error Using 1 Static Mode with Ranked and Unranked Modes

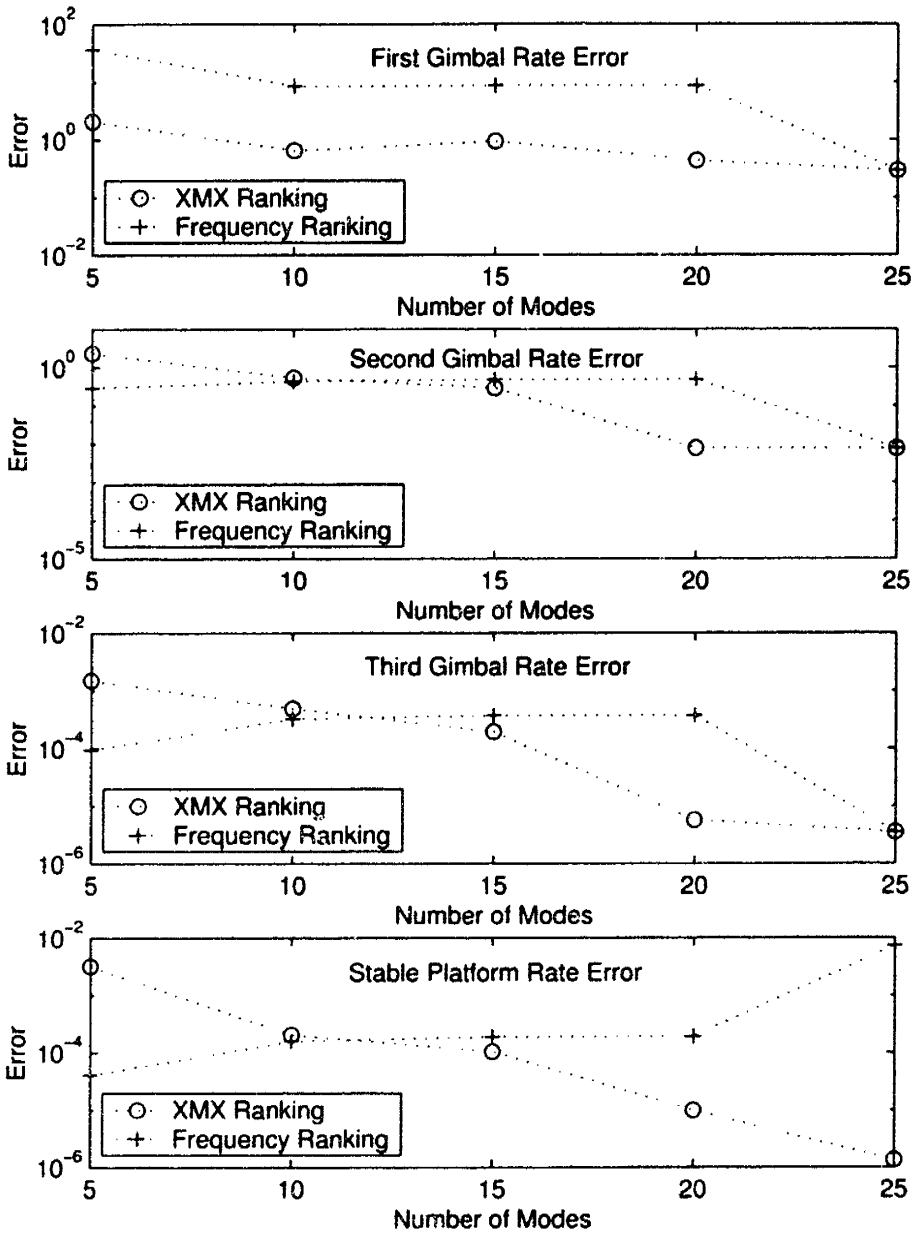


Figure 4-28: GammaDot Error of the XMX/XMGMX Ranking and the Frequency Ranking

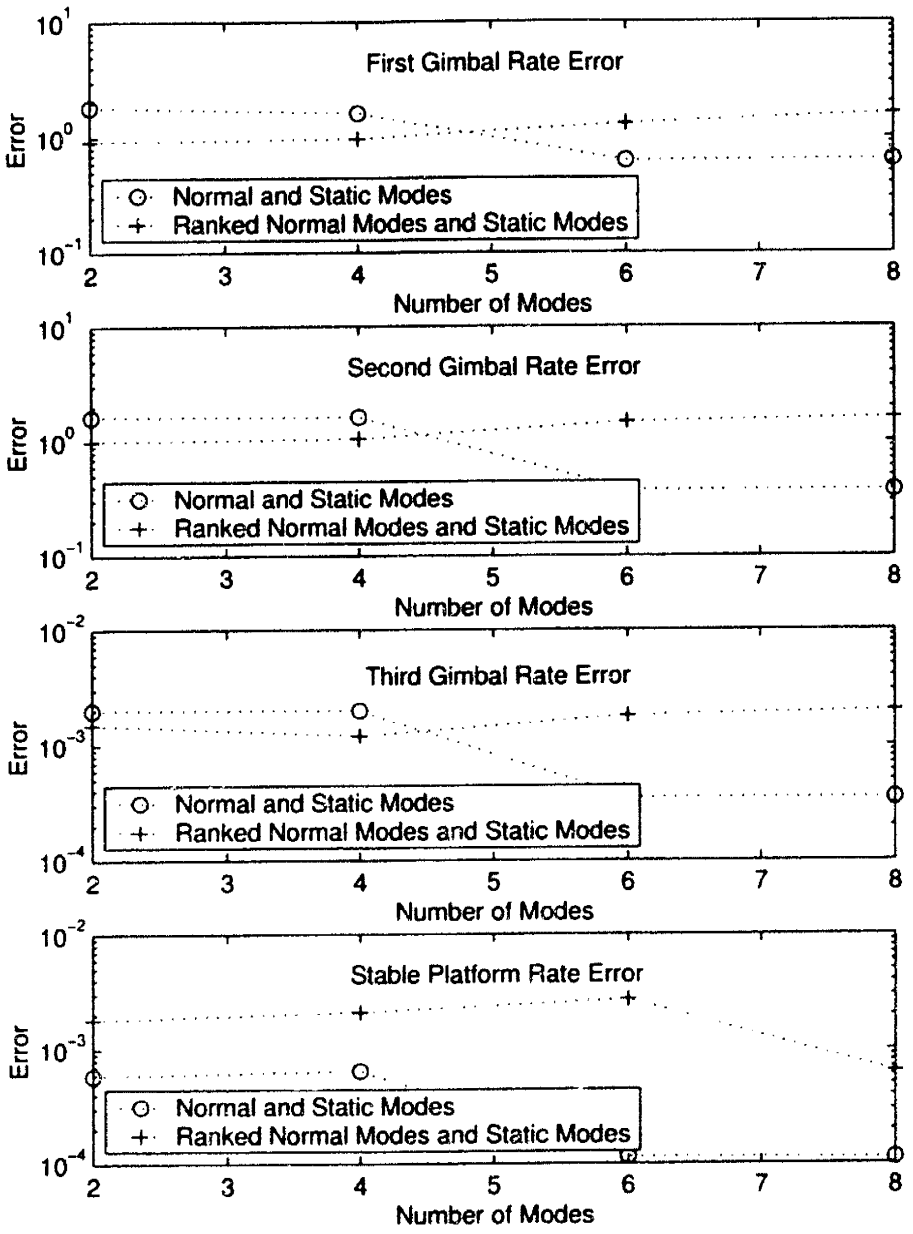


Figure 4-29: GammaDot Error of the Normal with Static Modes and Ranked Normal with Static Modes

plished while still maintaining results that are close to the truth model. Using the XMX/XMGMX ranking provided good results if the number of modes was at least 20. Taking advantage of the static deformation of the structure at high frequencies, using a static mode in conjuncture with the unranked modes produces very good results with only 10 modes. In short, when using the simulation without taking into account any type of forcing conditions it is advantageous to use the XMX/XMGMX ranking since only structural mass properties are taken into account. If the type of static deformation can be inferred from the type of force being applied to the structure then, using a static mode in addition to the low frequency modes provide adequate results with a low number of modes.

Chapter 5

Conclusion

5.1 Future Work

With the completion of the use of static correction modes and mode selection criteria for model reduction analysis of the set of elastic nested gimbals, it is possible to further refine the model reduction method as it applies to the nested gimbals. For instance, static correction modal analysis based on other sets of initial conditions can be developed. From these conditions, it can be possible to obtain an idea as to which locations and forces will provide the most deformation due to local joint connections. It would also be advantageous to develop a global system in which all the modes for each body are analyzed together instead of separately as in this model. For example, the modes to be ranked can be taken from a pool of all the modes associated with each body. This may prove useful in identifying the mode contribution to the whole global system. Another area that can be analyzed is the improvement of computation time that occurs with larger time steps when using these model reduction methods. These studies can help the model reduction techniques of not only this nested gimbal problem, but others in which large systems of equations are to be solved.

5.2 Summary

Applying model reduction theories to a set of elastic, nested gimbals provides adequate results while reducing the computing power necessary to solve the system of equations. The model reduction is possible through the use of mode selection criteria or static correction modes. The combination of both methods, however, does not result in any improvement. The use of mode selection methods allow of the ranking of the modes based solely on mass properties. The configuration of the bodies

does not have an affect on the ranking of these modes. Thus, this method can be applied to any type of body regardless of how it is attached to each others at the joints.

Static correction modes are useful when an idea about the reaction forces at the joints are known. These reaction forces can be used to create static correction modes from the deformation at the joints. It is here where insight to the problem proves useful in obtaining an accurate solution. The model reduction techniques proved useful for the nested gimbal model. The advantage of using these methods is not only restricted to this model. It can be applied to dynamic problems in which large complex structures need to be modeled. In cases where solving these full problems is not practical, the use of model reduction techniques can prove to be a valuable tool to the insight of the dynamic problems with reasonable computation time. However, although it seems that all model reduction techniques produce the same results, it is interesting to see that different techniques work better for a certain type of problem. There is no one method that will work well for every large dynamic problem thus, as long as different these large dynamic models exist, different methods can be created to reduce the model.

Appendix A

Matlab Source Code

A.1 Matlab Code For Mode Ranking

The mode ranking code here is used for the outer gimbal only. It can be applied to the other bodies.

```
% This m-file ranks the modes of the outer gimbal

clear all

load fullMassFlex2 % Contains location of nodes in local coordinate
                   % frame, eigenvectors and eigenvalues of outer gimbal.
load GlobalOuterMat % Contains sparse mass and stiffness matrix of outer
                    % gimbal.

eigvals = eigvals2 % Eigenvalues                                          10
location = location2; % Node Location
eigvect = eigvect2; % Eigenvectors
MassMatrix = full(sparseMouterGimb); % Mass matrix
StiffMatrix = full(sparseKouterGimb); % Stiffness matrix

[row, column, depth] = size(eigvect)
depth = 100'
for i = 1: depth
   for j = 1:row
      eigMat(6*j-5:6*j,i) = eigvect(j,1:6,i)';                            20
   end
end

I3 = eye(3);
zero3 = zeros(3);
for jj = 1: row
   rigidX(6*jj-5: 6*jj ,1:6) = [I3 skew(location(jj,1:3))
      zero3 I3];
```

```
end
                                                                    30
% compute XMX ranking
MInfinity = rigidX' * MassMatrix * rigidX;
MInfinityHalf = MInfinity^(-1/2);
% spectral radius computation

for iii = 1: depth
   term = MInfinityHalf * rigidX' * MassMatrix * eigMat(:,iii) * eigMat(:,iii)' ...
      * MassMatrix * rigidX* MInfinityHalf;
   rho(iii,1) = max(eig(term));
end                                                                 40

% This sorts the matrices in terms of the smalles to the largest. However the
% more important modes are the ones that are largest
[Y,Index] = sort(rho); % Index = XMX rank

%--------------- XMGMX Ranking --------------------
MreduceMatrix = MassMatrix(1:576,1:576);
KreduceMatrix = StiffMatrix(1:576,1:576);
rigidXreduce = rigidX(1:576,:);
GMatrix = inv(KreduceMatrix);                                       50

MGMInfinity = rigidXreduce' * MreduceMatrix * GMatrix * MreduceMatrix * ...
   rigidXreduce;
MGMInfinityHalf = MGMInfinity^(-1/2);
MGM = 0;
for iiii = depth
   term = 1/eigvals(iiii) * rigidX' * MassMatrix * eigMat(:,iiii) *...
      eigMat(:,iiii)' * MassMatrix * rigidX;
   MGM = MGM + term;
end                                                                 60

for iiii = depth
   term = MGMInfinityHalf * 1/eigvals(iiii)^2 * rigidX' * MassMatrix *...
      eigMat(:,iiii) * eigMat(:,iiii)' * MassMatrix * rigidX * MGMInfinityHalf;
   rho(iiii,1) = max(eig(term));
end

[Y,Index2] =sort(rho); % Index = XMGM rank

save OuterRank Index Index2                                         70
% End of File
```

A.2 Matlab Code For Static Modes

```
% This m-file appends the mode matrices if the use of static modes is
% selected by the user to solve the dynamic equations. It is necessary
% for the user to determine the node location of the force to be applied.
% The number of kept modes has already been established and it is
% necessary to append the time independent calculations of the kept
% normal nodes with static modes.
% ---------------------------------------
% Initialize Modal Mass matrices
statModalMass1 =[];
statModalMass2 = [];                                                    10
statModalMass3 =[];
statModalMass4 = [];
statModalMass5 =[];
% ------------ Outer Gimbal -------------

if staticModes2 > 0                    %Check if outer gimbal static modes are used
   load globalOuterMat        %Load Mass and Stiffness Matrices.
   % Truncate eigenvectors and eigenvalues
   eigvect2 = eigvect2(:,:,1:nModes2);
   eigvals2 = eigvals2(1:nModes2)                                       20
   [row, column, depth] = size(eigvect2)
   % Convert Eigenvectors from 3-D to 2_D array.
   for i = 1: depth
      for j = 1:row
         eigMat(6*j-5:6*j,i) = eigvect2(j,1:6,i)';
      end
   end
   % Create D Matrix and Lambda Matrix
   Kmat = full(sparseKouterGimb);
   KreduceMatrix = Kmat(1:576,1:576);                                   30
   invKmat = inv(KreduceMatrix);
   zero24x576 = zeros(24,576);
   zero24x24 = zeros(24,24);
   Lambda = diag(eigvals2);
   location = location2;
   nNodes = row;
   Dmat = [invKmat  zero24x576';
     zero24x576    zero24x24];
   % Start Loop to Create Static Correction Modes.
   for kk = 1: staticModes2                                             40
      % ---------------------
       force = zeros(600,1);
      % force placement coordinate.
      if kk ==1
         force(105,1) = 1;
```

```
elseif kk == 2
  force(156,1) = 1;
        elseif kk == 3
   force(391,1) = 1;
elseif kk == 4                                                    50
   force(392,1) = 1;
elseif kk ==5
        force(104,1) = 1;
elseif kk == 6
   force(105,1) = 1;
end
% ----------------------
% Check to see if normal modes were used
% and Create static mode
if nModes2 ~= 0                                                   60
   statMode = (Dmat - eigMat*inv(Lambda)*eigMat') * force;
else
   statMode = Dmat * force;
end
% 'unpack' static mode to create mode matrices
for jj = 1: row
   statModeMat(jj,1:6) = statMode(6*jj-5:6*jj)';
end
staticMode = statModeMat;
%Modal integral P and H                                           70
alpha = zeros(3,3);
for jj = 1 : nNodes
   psiXTimesDm(jj) = staticMode(jj,1)* sparseMouterGimb(6*jj-5,6*jj-5);
   psiYTimesDm(jj) = staticMode(jj,2)* sparseMouterGimb(6*jj-4,6*jj-4);
   psiZTimesDm(jj) = staticMode(jj,3)* sparseMouterGimb(6*jj-3,6*jj-3);
   rskewTimesPsi(1:3,jj) = skew(location(jj,1:3))* staticMode(jj,1:3)' .* ...
      [sparseMouterGimb(6*jj-5,6*jj-5); ...
      sparseMouterGimb(6*jj-4,6*jj-4); sparseMouterGimb(6*jj-3,6*jj-3)];

   xx = staticMode(jj,1)* location(jj,1)*sparseMouterGimb(6*jj-5,6*jj-5);   80
   xy = staticMode(jj,1)* location(jj,2)*sparseMouterGimb(6*jj-4,6*jj-4);
   xz = staticMode(jj,1)* location(jj,3)*sparseMouterGimb(6*jj-3,6*jj-3);

   yx = staticMode(jj,2)*location(jj,1)*sparseMouterGimb(6*jj-5,6*jj-5);
   yy = staticMode(jj,2)*location(jj,2)*sparseMouterGimb(6*jj-4,6*jj-4);
   yz = staticMode(jj,2)*location(jj,3)*sparseMouterGimb(6*jj-3,6*jj-3);

   zx = staticMode(jj,3)*location(jj,1)*sparseMouterGimb(6*jj-5,6*jj-5);
   zy = staticMode(jj,3)*location(jj,2)*sparseMouterGimb(6*jj-4,6*jj-4);
   zz = staticMode(jj,3)*location(jj,3)*sparseMouterGimb(6*jj-3,6*jj-3);   90
   alpha = alpha + [xx xy xz;
      yx yy yz;
      zx zy zz];
```

```
end
pSumX = sum(psiXTimesDm);
pSumY = sum(psiYTimesDm);
pSumZ = sum(psiZTimesDm);

hSumX = sum(rskewTimesPsi(1,:));
hSumY = sum(rskewTimesPsi(2,:));                                    100
hSumZ = sum(rskewTimesPsi(3,:));
% place summation in 3x1 matrix
PmatStatic = [pSumX; pSumY; pSumZ];
HmatStatic = [hSumX; hSumY; hSumZ];
Hmatrix = [alpha(3,2) - alpha(2,3); alpha(1,3) - alpha(3,1); ...
      alpha(2,1) - alpha(1,2)];
ALPHAMATRIX = alpha;
% Rotations and Translations
inMotXTrans = staticMode(nNodes,1);
inMotYTrans = staticMode(nNodes,2);                                 110
inMotZTrans = staticMode(nNodes,3);

inMotXRot = staticMode(nNodes,4);
inMotYRot = staticMode(nNodes,5);
inMotZRot = staticMode(nNodes,6);

inResXTrans = staticMode(nNodes-1,1);
inResYTrans = staticMode(nNodes-1,2);
inResZTrans = staticMode(nNodes-1,3);
                                                                    120
inResXRot = staticMode(nNodes-1,4);
inResYRot = staticMode(nNodes-1,5);
inResZRot = staticMode(nNodes-1,6);

outMotXTrans = staticMode(nNodes-3,1);
outMotYTrans = staticMode(nNodes-3,2);
outMotZTrans = staticMode(nNodes-3,3);

outMotXRot = staticMode(nNodes-3,4);
outMotYRot = staticMode(nNodes-3,5);                                130
outMotZRot = staticMode(nNodes-3,6);

outResXTrans = staticMode(nNodes-2,1);
outResYTrans = staticMode(nNodes-2,2);
outResZTrans = staticMode(nNodes-2,3);

outResXRot = staticMode(nNodes-2,4);
outResYRot = staticMode(nNodes-2,5);
outResZRot = staticMode(nNodes-2,6);
                                                                    140
% Calculation of PSI
```

```
    outMotTrans = [outMotXTrans outMotYTrans outMotZTrans];
    outResTrans = [outResXTrans outResYTrans outResZTrans];
    PSI = (outMotTrans + outResTrans)'/2;
    PSI(3,:) = outResZTrans';
    % Calculation of Theta
    thetaX = (outResYTrans - outMotYTrans)/(location(97,3) - location(98,3));
    thetaY = (outMotXTrans - outResXTrans)/(location(97,3) - location(98,3));
    thetaZ = outMotZRot;
    THT = [thetaX thetaY thetaZ]';                                          150

    outDiam = (location(97,3) + outMotZTrans) - (location(98,3) + outResZTrans);
    inMotxDisp = inMotXTrans;
    inMotRot = [inMotXRot; inMotYRot; inMotZRot];
    inResRot = [inResXRot; inResYRot; inResZRot];
    outMotRot = [outMotXRot; outMotYRot; outMotZRot];
    outResRot = [outResXRot; outResYRot; outResZRot];

    [nRows nCols] = size(eigvals2);
    if nCols > nRows                                                        160
      eigvals2 = eigvals2';
    end

    eigenvalue = statMode' * Kmat * statMode
    statModalMass = statMode' * sparseMouterGimb * statMode;

    Hmat2 = [Hmat2 HmatStatic];
    Pmat2 = [Pmat2 PmatStatic];
    THT2 = [THT2 THT];
    PSI2 = [PSI2 PSI];                                                      170
    inMotxDisp2 = [inMotxDisp2 inMotxDisp];
    inMotRot2 = [inMotRot2 inMotRot];
    inResRot2 = [inResRot2 inResRot];
    outMotRot2 = [outMotRot2 outMotRot];
    outResRot2 = [outResRot2 outResRot];
    outDiam2 = [outDiam2 outDiam];
    [aRow aColumn aDepth] = size(ALPHAMATRIX2);
    ALPHAMATRIX2(:,:,aDepth+1) = ALPHAMATRIX;
    eigvals2 = [eigvals2; eigenvalue];
    statModalMass2 = [statModalMass2; statModalMass];                       180
  end
end
nModes2 = nModes2 + staticModes2
% ------------------------------------------
% ---------------- Middle Gimbal ----------------
if staticModes3 > 0
      load globalMiddleMat
  eigvect3 = eigvect3(:,:,1:nModes3);
  eigvals3 = eigvals3(1:nModes3)
```

```
[row, column, depth] = size(eigvect3)                                    190
for i = 1: depth
   for j = 1:row
      eigMat(6*j-5:6*j,i) = eigvect3(j,1:6,i)';
   end
end
Kmat = full(sparseKmiddleGimb);
KreduceMatrix = Kmat(1:576,1:576);
invKmat = inv(KreduceMatrix);
zero24x576 = zeros(24,576);
zero24x24 = zeros(24,24);                                                200
Lambda = diag(eigvals3);
location = location3;
nNodes = row;
Dmat = [invKmat   zero24x576';
        zero24x576    zero24x24];
for kk = 1: staticModes3
     % ----------------------
     force = zeros(600,1);
   % force placement coordinate.
   if kk ==1                                                             210
                    force(129,1) = 1;
   elseif kk == 2
      force(84,1) = 1;
   elseif kk == 3
      force(415,1) = 1;
   elseif kk == 4
      force(416,1) = 1;
   elseif kk ==5
      force(128,1) = 1;
   elseif kk == 6                                                        220
      force(129,1) = 1;
   end
     % ----------------------
   if nModes3 ~= 0
     statMode = (Dmat - eigMat*inv(Lambda)*eigMat') * force;
   else
      statMode = Dmat * force;
   end
   % unpack
   for jj = 1: row                                                       230
      statModeMat(jj,1:6) = statMode(6*jj-5:6*jj)';
   end
   staticMode = statModeMat;
   %Modal integral P and H
   alpha = zeros(3,3);
   for jj = 1 : nNodes
      psiXTimesDm(jj) = staticMode(jj,1)* sparseMmiddleGimb(6*jj-5,6*jj-5);
```

```
    psiYTimesDm(jj) = staticMode(jj,2)* sparseMmiddleGimb(6*jj-4,6*jj-4);
    psiZTimesDm(jj) = staticMode(jj,3)* sparseMmiddleGimb(6*jj-3,6*jj-3);
    rskewTimesPsi(1:3,jj) = skew(location(jj,1:3))* staticMode(jj,1:3)' .*...
    [sparseMmiddleGimb(6*jj-5,6*jj-5); ...
    sparseMmiddleGimb(6*jj-4,6*jj-4); sparseMmiddleGimb(6*jj-3,6*jj-3)];

    xx = staticMode(jj,1)* location(jj,1)*sparseMmiddleGimb(6*jj-5,6*jj-5);
    xy = staticMode(jj,1)* location(jj,2)*sparseMmiddleGimb(6*jj-4,6*jj-4);
    xz = staticMode(jj,1)* location(jj,3)*sparseMmiddleGimb(6*jj-3,6*jj-3);

    yx = staticMode(jj,2)*location(jj,1)*sparseMmiddleGimb(6*jj-5,6*jj-5);
    yy = staticMode(jj,2)*location(jj,2)*sparseMmiddleGimb(6*jj-4,6*jj-4);
    yz = staticMode(jj,2)*location(jj,3)*sparseMmiddleGimb(6*jj-3,6*jj-3);

    zx = staticMode(jj,3)*location(jj,1)*sparseMmiddleGimb(6*jj-5,6*jj-5);
    zy = staticMode(jj,3)*location(jj,2)*sparseMmiddleGimb(6*jj-4,6*jj-4);
    zz = staticMode(jj,3)*location(jj,3)*sparseMmiddleGimb(6*jj-3,6*jj-3);
    alpha = alpha + [xx xy xz;
                     yx yy yz;
                     zx zy zz];
end
pSumX = sum(psiXTimesDm);
pSumY = sum(psiYTimesDm);
pSumZ = sum(psiZTimesDm);

hSumX = sum(rskewTimesPsi(1,:));
hSumY = sum(rskewTimesPsi(2,:));
hSumZ = sum(rskewTimesPsi(3,:));
% place summation in 3x1 matrix
PmatStatic = [pSumX; pSumY; pSumZ];
HmatStatic = [hSumX; hSumY; hSumZ];

Hmatrix = [alpha(3,2) - alpha(2,3); alpha(1,3) - alpha(3,1);...
      alpha(2,1) - alpha(1,2)];
ALPHAMATRIX = alpha;

% Rotations and Translations
inMotXTrans = staticMode(nNodes,1);
inMotYTrans = staticMode(nNodes,2);
inMotZTrans = staticMode(nNodes,3);

inMotXRot = staticMode(nNodes,4);
inMotYRot = staticMode(nNodes,5);
inMotZRot = staticMode(nNodes,6);

inResXTrans = staticMode(nNodes-1,1);
inResYTrans = staticMode(nNodes-1,2);
inResZTrans = staticMode(nNodes-1,3);
```

```
inResXRot = staticMode(nNodes-1,4);
inResYRot = staticMode(nNodes-1,5);
inResZRot = staticMode(nNodes-1,6);

outMotXTrans = staticMode(nNodes-3,1);
outMotYTrans = staticMode(nNodes-3,2);
outMotZTrans = staticMode(nNodes-3,3);

outMotXRot = staticMode(nNodes-3,4);
outMotYRot = staticMode(nNodes-3,5);
outMotZRot = staticMode(nNodes-3,6);

outResXTrans = staticMode(nNodes-2,1);
outResYTrans = staticMode(nNodes-2,2);
outResZTrans = staticMode(nNodes-2,3);

outResXRot = staticMode(nNodes-2,4);
outResYRot = staticMode(nNodes-2,5);
outResZRot = staticMode(nNodes-2,6);

% Calculation of PSI
outMotTrans = [outMotXTrans outMotYTrans outMotZTrans];
outResTrans = [outResXTrans outResYTrans outResZTrans];
PSI = (outMotTrans + outResTrans)'/2;
PSI(3,:) = outResZTrans';
% Calculation of Theta
thetaX = (outResYTrans - outMotYTrans)/(location(97,3) - location(98,3));
thetaY = (outMotXTrans - outResXTrans)/(location(97,3) - location(98,3));
thetaZ = outMotZRot;
THT = [thetaX thetaY thetaZ]';

outDiam = (location(97,3) + outMotZTrans) - (location(98,3) + outResZTrans);
inMotxDisp = inMotXTrans;
inMotRot = [inMotXRot; inMotYRot; inMotZRot];
inResRot = [inResXRot; inResYRot; inResZRot];
outMotRot = [outMotXRot; outMotYRot; outMotZRot];
outResRot = [outResXRot; outResYRot; outResZRot];

[nRows nCols] = size(eigvals3);
if nCols > nRows
   eigvals3 = eigvals3';
end
eigenvalue = statMode' * Kmat * statMode
statModalMass = statMode' * sparseMmiddleGimb * statMode;

Hmat3 = [Hmat3 HmatStatic];
Pmat3 = [Pmat3 PmatStatic];
```

```
      THT3 = [THT3 THT];
      PSI3 = [PSI3 PSI];
      inMotxDisp3 = [inMotxDisp3 inMotxDisp];
      inMotRot3 = [inMotRot3 inMotRot];
      inResRot3 = [inResRot3 inResRot];
      outMotRot3 = [outMotRot3 outMotRot];
      outResRot3 = [outResRot3 outResRot];                              340
      outDiam3 = [outDiam3 outDiam];
      [aRow aColumn aDepth] = size(ALPHAMATRIX3);
      ALPHAMATRIX3(:,:,aDepth+1) = ALPHAMATRIX;
      eigvals3 = [eigvals3; eigenvalue];
      statModalMass3 = [statModalMass3; statModalMass];

   end
end
nModes3 = nModes3 + staticModes3
% ------------------------------------------                          350
% ------------ Inner Gimbal ----------------

if staticModes4 > 0
        load globalInnerMat
   eigvect4 = eigvect4(:,:,1:nModes4);
   eigvals4 = eigvals4(1:nModes4)
   [row, column, depth] = size(eigvect4)
   for i = 1: depth
              for j = 1:row
      eigMat(6*j-5:6*j,i) = eigvect4(j,1:6,i)';                         360
     end
   end
   statModalMass4 = [];
   Kmat = full(sparseKinnerGimb);
   KreduceMatrix = Kmat(1:576,1:576);
   invKmat = inv(KreduceMatrix);
   zero24x576 = zeros(24,576);
   zero24x24 = zeros(24,24);
   Lambda = diag(eigvals4);
   location = location4;                                              370
   nNodes = row;
   Dmat = [invKmat  zero24x576';
           zero24x576   zero24x24];
   for kk = 1: staticModes4
      % ----------------------
      force = zeros(600,1);
      % force placement coordinate.
      if kk ==1
         force(441,1) = 1;
      elseif kk == 2                                                  380
         force(204,1) = 1;
```

```
elseif kk == 3
   force(151,1) = 1;
elseif kk == 4
   force(152,1) = 1;
elseif kk == 5
   force(440,1) = 1;
elseif kk == 6
   force(441,1) = 1;
end                                                                    390
% ------------------------
if nModes4 ~= 0
   statMode = (Dmat - eigMat*inv(Lambda)*eigMat') * force;
else
   statMode = Dmat * force;
end
% unpack
for jj = 1: row
   statModeMat(jj,1:6) = statMode(6*jj-5:6*jj)';
end                                                                    400
staticMode = statModeMat;

%Modal integral P and H
alpha = zeros(3,3);
for jj = 1 : nNodes
   psiXTimesDm(jj) = staticMode(jj,1)* sparseMinnerGimb(6*jj-5,6*jj-5);
   psiYTimesDm(jj) = staticMode(jj,2)* sparseMinnerGimb(6*jj-4,6*jj-4);
   psiZTimesDm(jj) = staticMode(jj,3)* sparseMinnerGimb(6*jj-3,6*jj-3);
   rskewTimesPsi(1:3,jj) = skew(location(jj,1:3))* staticMode(jj,1:3)' .*...
      [sparseMinnerGimb(6*jj-5,6*jj-5); ...                            410
      sparseMinnerGimb(6*jj-4,6*jj-4); sparseMinnerGimb(6*jj-3,6*jj-3)];
  xx = staticMode(jj,1)* location(jj,1)*sparseMinnerGimb(6*jj-5,6*jj-5);
        xy = staticMode(jj,1)* location(jj,2)*sparseMinnerGimb(6*jj-4,6*jj-4);
   xz = staticMode(jj,1)* location(jj,3)*sparseMinnerGimb(6*jj-3,6*jj-3);

   yx = staticMode(jj,2)*location(jj,1)*sparseMinnerGimb(6*jj-5,6*jj-5);
   yy = staticMode(jj,2)*location(jj,2)*sparseMinnerGimb(6*jj-4,6*jj-4);
   yz = staticMode(jj,2)*location(jj,3)*sparseMinnerGimb(6*jj-3,6*jj-3);

   zx = staticMode(jj,3)*location(jj,1)*sparseMinnerGimb(6*jj-5,6*jj-5);  420
   zy = staticMode(jj,3)*location(jj,2)*sparseMinnerGimb(6*jj-4,6*jj-4);
   zz = staticMode(jj,3)*location(jj,3)*sparseMinnerGimb(6*jj-3,6*jj-3);
   alpha = alpha + [xx xy xz;
                    yx yy yz;
                    zx zy zz];
 end
pSumX = sum(psiXTimesDm);
pSumY = sum(psiYTimesDm);
pSumZ = sum(psiZTimesDm);
```

```
hSumX = sum(rskewTimesPsi(1,:));
hSumY = sum(rskewTimesPsi(2,:));
hSumZ = sum(rskewTimesPsi(3,:));
% place summation in 3x1 matrix
PmatStatic = [pSumX; pSumY; pSumZ];
HmatStatic = [hSumX; hSumY; hSumZ];

Hmatrix = [alpha(3,2) - alpha(2,3); alpha(1,3) - alpha(3,1); ...
      alpha(2,1) - alpha(1,2)];
ALPHAMATRIX = alpha;

% Rotations and Translations
inMotXTrans = staticMode(nNodes,1);
inMotYTrans = staticMode(nNodes,2);
inMotZTrans = staticMode(nNodes,3);

inMotXRot = staticMode(nNodes,4);
inMotYRot = staticMode(nNodes,5);
inMotZRot = staticMode(nNodes,6);

inResXTrans = staticMode(nNodes-1,1);
inResYTrans = staticMode(nNodes-1,2);
inResZTrans = staticMode(nNodes-1,3);

inResXRot = staticMode(nNodes-1,4);
inResYRot = staticMode(nNodes-1,5);
inResZRot = staticMode(nNodes-1,6);

outMotXTrans = staticMode(nNodes-3,1);
outMotYTrans = staticMode(nNodes-3,2);
outMotZTrans = staticMode(nNodes-3,3);

outMotXRot = staticMode(nNodes-3,4);
outMotYRot = staticMode(nNodes-3,5);
outMotZRot = staticMode(nNodes-3,6);

outResXTrans = staticMode(nNodes-2,1);
outResYTrans = staticMode(nNodes-2,2);
outResZTrans = staticMode(nNodes-2,3);

outResXRot = staticMode(nNodes-2,4);
outResYRot = staticMode(nNodes-2,5);
outResZRot = staticMode(nNodes-2,6);

% Calculation of PSI
outMotTrans = [outMotXTrans outMotYTrans outMotZTrans];
outResTrans = [outResXTrans outResYTrans outResZTrans];
PSI = (outMotTrans + outResTrans)'/2;
```

```
      PSI(3,:) = outResZTrans';
      % Calculation of Theta
      thetaX = (outResYTrans - outMotYTrans)/(location(100,3) - location(99,3));
      thetaY = (outMotXTrans - outResXTrans)/(location(100,3) - location(99,3));
      thetaZ = outMotZRot;
      THT = [thetaX thetaY thetaZ]';

      outDiam = (location(97,3) + outMotZTrans) - (location(98,3) + outResZTrans);
      inMotxDisp = inMotXTrans;
      inMotRot = [inMotXRot; inMotYRot; inMotZRot];
      inResRot = [inResXRot; inResYRot; inResZRot];
      outMotRot = [outMotXRot; outMotYRot; outMotZRot];
      outResRot = [outResXRot; outResYRot; outResZRot];

      [nRows nCols] = size(eigvals4);
      if nCols > nRows
        eigvals4 = eigvals4';
      end
      eigenvalue = statMode' * Kmat * statMode
      statModalMass = statMode' * sparseMinnerGimb * statMode;

      Hmat4 = [Hmat4 HmatStatic];
      Pmat4 = [Pmat4 PmatStatic];
      THT4 = [THT4 THT];
      PSI4 = [PSI4 PSI];
      inMotxDisp4 = [inMotxDisp4 inMotxDisp];
      inResRot4 = [inResRot4 inResRot];
      outMotRot4 = [outMotRot4 outMotRot];
      outResRot4 = [outResRot4 outResRot];
      outDiam4 = [outDiam4 outDiam];
      [aRow aColumn aDepth] = size(ALPHAMATRIX4);
      ALPHAMATRIX4(:,:,aDepth+1) = ALPHAMATRIX;
      eigvals4 = [eigvals4; eigenvalue];
      statModalMass4 = [statModalMass4; statModalMass];

   end
end
nModes4 = nModes4 + staticModes4
% ------------------------------------
% ----------Stable Platform ----------------
if staticModes5 > 0
      load GlobalStabMat
   counter = 1;
   eigvect5 = eigvect5(:,:,1:nModes5);
   eigvals5 = eigvals5(1:nModes5)
   [row, column, depth] = size(eigvect5)
   for i = 1: depth
      for j = 1:row
```

```
    eigMat(6*j-5:6*j,i) = eigvect5(j,1:6,i)';
  end
end
Mmat = full(sparseMstabPlat);
Kmat = full(sparseKstabPlat);
MreduceMatrix = Mmat(1:768,1:768);
KreduceMatrix = Kmat(1:768,1:768);
eigTempJoints = eigMat(769:780,1:depth);
eigMat = eigMat(1:768,1:depth);
eigTemp = zeros(128,depth);
for i=768:-1:1
      if mod((i+1),6) == 0
          MreduceMatrix(i,:)= [];
     MreduceMatrix(:,i)= [];
     KreduceMatrix(i,:)= [];
     KreduceMatrix(:,i)= [];
     eigTemp(counter,1:depth) = eigMat(i,1:depth);
     counter = counter +1;
     eigMat(i,:) = [];
  end
end
eigMat = [eigMat; eigTemp; eigTempJoints];
[reduceRows reduceCols] = size(KreduceMatrix)
invKmat = inv(KreduceMatrix);
zero24x576 = zeros(780- reduceRows, reduceRows);
zero24x24 = zeros(780 - reduceRows, 780 - reduceRows);
Lambda = diag(eigvals5);
location = location5;
nNodes = row;
Dmat = [invKmat  zero24x576';
       zero24x576   zero24x24];
fullKreduceMatrix = [KreduceMatrix zero24x576'
                     zero24x576   zero24x24];
fullMreduceMatrix = [MreduceMatrix zero24x576'
                     zero24x576   zero24x24];
for kk = 1: staticModes5
% ----------------------
    counter2 = 1;
  counter3 = 1;
  force = zeros(780,1);
  % force placement coordinate.
  if kk ==1
    force(628,1) = 1;
  elseif kk == 2
    force(70,1) = 1;
  elseif kk == 3
  end
  % ----------------------
```

```
if nModes4 ~= 0
   statMode = (Dmat - eigMat*inv(Lambda)*eigMat') * force;
else
   statMode = Dmat * force;
end
statModeTemp = statMode;
% ----------unpack----------                                          580
for jjjj = 1:768
   if mod((jjjj+1),6) == 0
      statModeTemp2(jjjj,1) = statModeTemp(640+counter2,1);
      counter2 = counter2+1;
   else
      statModTemp2(jjjj,1) = statModeTemp(counter3,1);
      counter3 = counter3+1;
   end
end
statMode = [statModTemp2; statMode(769:780,1)];                        590
for jj = 1: row
   statModeMat(jj,1:6) = statMode(6*jj-5:6*jj)';
end
staticMode = statModeMat;
%Modal integral P and H
alpha = zeros(3,3);
for jj = 1 : nNodes
   psiXTimesDm(jj) = staticMode(jj,1)* sparseMstabPlat(6*jj-5,6*jj-5);
   psiYTimesDm(jj) = staticMode(jj,2)* sparseMstabPlat(6*jj-4,6*jj-4);
   psiZTimesDm(jj) = staticMode(jj,3)* sparseMstabPlat(6*jj-3,6*jj-3);  600
   rskewTimesPsi(1:3,jj) = skew(location(jj,1:3))* staticMode(jj,1:3)' .*. . .
      [sparseMstabPlat(6*jj-5,6*jj-5); ...
      sparseMstabPlat(6*jj-4,6*jj-4); sparseMstabPlat(6*jj-3,6*jj-3)];

   xx = staticMode(jj,1)* location(jj,1)*sparseMstabPlat(6*jj-5,6*jj-5);
   xy = staticMode(jj,1)* location(jj,2)*sparseMstabPlat(6*jj-4,6*jj-4);
   xz = staticMode(jj,1)* location(jj,3)*sparseMstabPlat(6*jj-3,6*jj-3);

   yx = staticMode(jj,2)*location(jj,1)*sparseMstabPlat(6*jj-5,6*jj-5);
   yy = staticMode(jj,2)*location(jj,2)*sparseMstabPlat(6*jj-4,6*jj-4);  610
   yz = staticMode(jj,2)*location(jj,3)*sparseMstabPlat(6*jj-3,6*jj-3);

   zx = staticMode(jj,3)*location(jj,1)*sparseMstabPlat(6*jj-5,6*jj-5);
   zy = staticMode(jj,3)*location(jj,2)*sparseMstabPlat(6*jj-4,6*jj-4);
   zz = staticMode(jj,3)*location(jj,3)*sparseMstabPlat(6*jj-3,6*jj-3);
   alpha = alpha + [xx xy xz;
          yx yy yz;
          zx zy zz];
  end
  pSumX = sum(psiXTimesDm);                                            620
pSumY = sum(psiYTimesDm);
```

```
pSumZ = sum(psiZTimesDm);

hSumX = sum(rskewTimesPsi(1,:));
hSumY = sum(rskewTimesPsi(2,:));
hSumZ = sum(rskewTimesPsi(3,:));
% place summation in 3x1 matrix
PmatStatic = [pSumX; pSumY; pSumZ];
HmatStatic = [hSumX; hSumY; hSumZ];

Hmatrix = [alpha(3,2) - alpha(2,3); alpha(1,3) - alpha(3,1); ...
     alpha(2,1) - alpha(1,2)];
ALPHAMATRIX = alpha;

% Rotations and Translations
inMotXTrans = staticMode(nNodes,1);
inMotYTrans = staticMode(nNodes,2);
inMotZTrans = staticMode(nNodes,3);

inMotXRot = staticMode(nNodes,4);
inMotYRot = staticMode(nNodes,5);
inMotZRot = staticMode(nNodes,6);

inResXTrans = staticMode(nNodes-1,1);
inResYTrans = staticMode(nNodes-1,2);
inResZTrans = staticMode(nNodes-1,3);

inResXRot = staticMode(nNodes-1,4);
inResYRot = staticMode(nNodes-1,5);
inResZRot = staticMode(nNodes-1,6);

inMotxDisp = inMotXTrans;
inMotRot = [inMotXRot; inMotYRot; inMotZRot];
inResRot = [inResXRot; inResYRot; inResZRot];

[nRows nCols] = size(eigvals5);
if nCols > nRows
   eigvals5 = eigvals5';
end
eigenvalue = statMode' * fullKreduceMatrix * statMode
statModalMass = statMode' * fullMreduceMatrix * statMode
Hmat5 = [Hmat5 HmatStatic];
Pmat5 = [Pmat5 PmatStatic];
inMotxDisp5 = [inMotxDisp5 inMotxDisp];
inMotRot5 = [inMotRot5 inMotRot];
inResRot5 = [inResRot5 inResRot];
[aRow aColumn aDepth] = size(ALPHAMATRIX5);
ALPHAMATRIX5(:,:,aDepth+1) = ALPHAMATRIX;
eigvals5 = [eigvals5; eigenvalue];
```

```
      statModalMass5 = [statModalMass5; statModalMass];

   end
end

%-------------------------------------------------------------------
nModes5 = nModes5 + staticModes5

Pmat = [Pmat1 Pmat2 Pmat3 Pmat4 Pmat5];
Hmat = [Hmat1 Hmat2 Hmat3 Hmat4 Hmat5];
PSI = [PSI1 PSI2 PSI3 PSI4];
THT = [THT1 THT2 THT3 THT4];
inResRot = [inResRot1 inResRot2 inResRot3 inResRot4 inResRot5];
outResRot = [outResRot1 outResRot2 outResRot3 outResRot4];
outMotRot = [outMotRot1 outMotRot2 outMotRot3 outMotRot4];
statModalMass = [statModalMass2; statModalMass3; statModalMass4;  statModalMass5];
statIndex2 = nModes2-staticModes2+1:nModes2;
if staticModes2 == 0
   statIndex3 = [];
else
   statIndex3 = statIndex2(staticModes2) + nModes3-staticModes3+ ...
      1:nModes3+ statIndex2(staticModes2);
end

if staticModes3 == 0
   statIndex4 = [];
else
   statIndex4 = statIndex3(staticModes3) + nModes4-staticModes4+...
      1:nModes4+ statIndex3(staticModes3);
end

if staticModes4 == 0
   statIndex5 = [];
else
   statIndex5 = statIndex4(staticModes4) + nModes5-staticModes5+...
      1:nModes5+ statIndex4(staticModes4);
end
% Stat Index keeps the index of location of the static modal masses in the
% elastic mass matrix Mee.
statIndex = [statIndex2'; statIndex3'; statIndex4'; statIndex5'];
% End of File
```

Bibliography

[1] Bathe, K.J. *Finite Element Procedures*, Prentice Hall, New Jersey, 1996.

[2] Craig, R. R. and Bampton, C.C., "Coupling of Subtructures for Dynamic Analysis," *AIAA journal*, Vol. 6, 1968, pp. 1313- 1319.

[3] Craig, R. R. and Chang, C.J., "On the Use of Attachment Modes in Substructure Coupling for Dynamic Analysis," *Paper presented at Dynamics and Structural Dynamics, AIAA/ASME 18th Structures, Structural Dynamics and Material Conference*, 1977.

[4] Hughes, P. C., "Modal Indentities for Elastic Bodies, With Application to Vehicle Dynamics and Control," *Journal of Applied Mechanics*, ASME Vol. 47, March 1980, pp 177-184.

[5] Hughes, P.C. and Sincarsin, G.B., "Dynamics of an Elastic Multibody Chain: Part B - Global Dynamics," *Dynamics and Stability of Systems*, Vol.4, Nos. 3&4, 1989, pp. 227-244.

[6] Hurty, Walter C., "Dynamic Analysis of Structural Systems by Component Mode Synthesis," *AIAA journal*, Vol.3, No. 4, 1965, pp 678-684.

[7] Patel, Ashok B., "Component Mode Selection Criteria for Model Reduction of Large Flexible Structures," Massachusetts Institute of Technology Thesis, 1992.

[8] Sincarsin, G.B. and Hughes, P.C., "Dynamics of an Elastic Multibody Chain: Part A - Body Motion Equations," *Dynamics and Stability of Systems*, Vol.4, Nos. 3&4, 1989, pp. 209-226.

[9] Skelton, R.E., Singh, R. and Ramakrishnan, J., "Component Model Reduction by Component Cost Analysis," AIAA Guidance and Control Conference, Minneapolis, MN, Aug. 1988.

[10] Storch, J., "Dynamics of Flexible Manipulator with General Joint Displacements Rept. CSDL-R 2365, Oct. 1991.

[11] Yoo, W.S., and Haug, E.J., "Dynamics of Articulated Structures, Part 1: Theory," *Journal of Structural Mechanics*, Vol. 14, No. 1, 1986.

[12] Yoo, W.S., and Haug, E.J., "Dynamics of Articulated Structures, Part 2: Computer Implementation and Applications," *Journal of Structural Mechanics*, Vol. 14, No. 2, 1986.

[13] Yoo, W.S., and Haug, E.J., "Dynamics of Flexible Mechanical Systems Using Vibration and Static Correction Modes," *Journal Mechanisms, Transmissions, and Automation in Design*, Vol. 108, 1986.

THESIS PROCESSING SLIP

FIXED FIELD: ill. _____ name _____

index _____ biblio _____

► **COPIES:** (Archives) Aero Dewey (Barker) Hum

Lindgren Music Rotch Science Sche-Plough

TITLE VARIES: ► ☐ _____

_____

_____

NAME VARIES: ► ☐ _____

_____

IMPRINT: **(COPYRIGHT)** _____

_____

► **COLLATION:** _____

_____

► **ADD: DEGREE:** _____ ► **DEPT.:** _____

► **ADD: DEGREE:** _____ ► **DEPT.:** _____

SUPERVISORS: _____

_____

NOTES:

cat'r: date:

page: J130

► **DEPT:** M.E.

► **YEAR:** 2001 ► **DEGREE:** S.M.

► **NAME:** BALUCAN, Phillip James

_____