INPUT-OUTPUT APPROXIMATION FOR

NONLINEAR STRUCTURAL DYNAMICS

A Thesis

by

STEFANIE RENE' BEAVER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2007

Major Subject: Aerospace Engineering

INPUT-OUTPUT APPROXIMATION FOR

NONLINEAR STRUCTURAL DYNAMICS

A Thesis

by

STEFANIE RENE' BEAVER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,   John E. Hurtado
Committee Members,   John L. Junkins
                      Reza Langari
Head of Department,   Helen Reed

August 2007

Major Subject: Aerospace Engineering

ABSTRACT

Input-Output Approximation for

Nonlinear Structural Dynamics. (August 2007)

Stefanie Rene' Beaver, B.S., Texas A&M University

Chair of Advisory Committee: Dr. John E. Hurtado

Input-output approximation of spacecraft motion is convenient and necessary in many situations. For a rigid-body spacecraft, this process is simple because the system is governed by a set of equations that is linear in the system parameters. However, the combination of a flexible appendage and a rigid hub adds complexity by increasing the degrees of freedom and by introducing nonlinear coupling between the hub and appendage.

Assumed Modes is one technique for modeling flexible body motion. Traditional Assumed Modes uses a set of linear assumed modes, but when dealing with rotating flexible systems, a modification of this method allows for the use of quadratic assumed modes. The quadratic assumed model provides an increased level of sophistication, but the derivation still neglects a set of higher-order terms. This work develops the nonlinear equations of motion that arise from retaining these nonlinear, higher-order terms. Simulation results reveal that the inclusion of these terms noticeably changes the motion of the system.

Once the equations of motion have been developed, focus turns to the input-output mapping of a system that is simulated using this set of equations. Approximating linear systems is straightforward, and many methods exist that can successfully perform this function. On the other hand, few approximation methods exist for nonlinear systems. Researchers at Texas A&M University recently developed one

such method that obtains a linear estimation and then uses an adaptive polynomial estimation method to compensate for the disparity between that estimate and the true measurements. This research includes an in-depth investigation of this nonlinear approximation technique.

Finally, these two major research thrusts are combined, and input-output approximation is performed on the nonlinear rotational spacecraft model developed herein. The results of this simulation show that the nonlinear method holds a significant advantage over a traditional linear method in certain situations. Specifically, the nonlinear algorithm provides superior approximation for systems with nonzero natural frequencies. For the algorithm to be successful when rigid-body modes are present, the system motion must be persistently exciting. This research is an important first step toward developing a nonlinear parameter identification algorithm.

To Paige

test your limits, and keep going . . .

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Dr. John E. Hurtado, for his continued mentoring and guidance throughout my undergraduate and graduate careers. I have had the opportunity to work with Dr. Hurtado off and on for the past five years, and he has been a source of encouragement from my first research experience through the culmination of this work.

Thank you also to Dr. John L. Junkins and Dr. Reza Langari for their support during this process. Both of these individuals also helped shape my technical skills during both my undergraduate and graduate experiences at Texas A&M. Dr. Junkins was kind enough to sponsor my summer research in 2004. I learned a great deal about research methods and methodology during those few months. Dr. Langari was my instructor and mentor during the 2004-2005 Boeing Rocket Design project. Dr. Puneet Singla, Troy Henderson, and Manoranjan Majji have also provided assistance with GLOMAP, ERA, and adaptive estimation, respectively. Thanks are also due to the staff of the Aerospace Engineering department who have been helpful in giving advice and ensuring that I was able to graduate on time and without complications.

My parents, other family, and friends have been invaluable in maintaining my sanity. I could not have come this far without all of you at my side to support me and at times, in front of me to pull me along. Thank you to Mom and Chip who were always there for me, to Eyal who stayed beside me during the successes and the tears, to Paige for all of the pictures and the phone calls, to Carolina for the hours we spent together in various coffee shops, and to the rest of my friends whom I could not enumerate here, thank you all. Special thanks are due to the following individuals who read this thesis and provided insightful feedback: Julie Jones, Carolina Restrepo, and Jeremy Davis.

Last, but certainly not least, I would like to express my gratitude to my high school mathematics and physics teachers, Mr. Neil Gander and Mr. David Carroll. Without them, I would not have pursued Aerospace Engineering so fervently in the first place. Mr. Gander passed away May 24, 2005, but he will always be fondly remembered as "Commander Gander."

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE   Page

CHAPTER I

INTRODUCTION

A number of operational satellites have flexible body appendages such as solar panels. Approximating the motion of these spacecraft is essential for path planning and determining control inputs. The Assumed Modes method is one method for modeling flexible body motion. Coupling this method with the rigid body motion for a single-axis rotation produces a set of linear equations that are valid for small deflections [1]. However, for large slewing maneuvers, these equations are no longer accurate. The Assumed Modes method with quadratic assumed modes is one solution that provides a more realistic representation of the system motion for these large maneuvers [2][3]. Quadratic assumed modes produce a set of linear equations only after nonlinear terms have been neglected. One focus of this research is to develop the equations of motion using the quadratic assumed modes but to retain the higher-order terms that will lead to a set of nonlinear equations.

The other main research focus is encompassed by system identification, also known as parameter identification. System identification is a topic that receives, and will continue to receive, significant attention in structural dynamics research. The goal of parameter identification is to use input-output data to identify an accurate model of the system so that the model can be used to predict future system behavior. This problem is straightforward for linear systems, and numerous solution methods exist for the linear class of problems. The Eigen Realization Algorithm and Observer/Kalman Filter Identification are two examples of methods that have been thoroughly investigated [4]. Regretfully, these algorithms are only valid for linear

---

This thesis follows the style of *IEEE Transactions on Automatic Control.*

systems which are very rarely present in practice.

True systems are most often governed by a set of nonlinear equations. The degree or severity of the nonlinearity varies from one system to the next, and for mildly nonlinear systems, linear parameter identification is often sufficient. Many nonlinear systems that only undergo small deflections can be easily approximated by a set of linear equations, but when these systems undergo moderate to large deflections, the estimate can be severely degraded. Finding a parameter identification method for nonlinear systems is one primary motivation for this work. Neural networks have been proposed as one possible identification method [5], and although they are typically very accurate after the learning process, little insight can be gained from the resulting model. The Global-Local Orthogonal Mapping algorithm (GLO-MAP) that was developed at Texas A&M University has been proposed as a neural network alternative that gives the user more insight into the approximation [6]. Recently at Texas A&M, GLO-MAP was integrated into a robust nonlinear parameter identification algorithm [7]. A thorough investigation and several applications of this algorithm will be the focus of the second half of this thesis. Although parameter identification has not been achieved in this work, nonlinear input-output approximation is an important first step in the nonlinear system identification process.

This research sits at the intersection of dynamics and estimation. Both of these research areas are sophisticated, and when combined, the true beauty of each can be better appreciated. In this work, a complex, nonlinear technique input-output approximation technique is investigated for use on the class of rotating flexible spacecraft. An overview of the traditional Assumed Modes method is given in Chapter II. Then, Chapter III will use variations of the Assumed Modes method to derive the equations of motion for rotating flexible systems as well as provide simulation results from these equations. At that point, the discussion will transition to input-output

mapping. The Eigen Realization Algorithm presented in Chapter IV is the chosen linear input-output approximation method, and Chapter V contains an extensive discussion of the nonlinear input-output approximation method recently developed at Texas A&M. Finally, these two research thrusts are combined when the nonlinear mapping is applied to a rotating spacecraft with a flexible appendage in Chapter VI. Recommendations are the focus of Chapter VII, and conclusions are given in Chapter VIII. To aid the reader, a nomenclature section has been included in Appendix A which begins on pg. 104.

CHAPTER II

LINEAR STRUCTURAL DYNAMICS OVERVIEW

The equations of motion for flexible, continuous systems are partial differential equations (PDEs) over space and time. Analytical solutions to these equations are frequently difficult to find or simply cannot be found using current PDE solution methods. Fortunately, many approximate solution methods exist. The Finite Element Method (FEM) and the Assumed Modes method are two such approximate solution methods. FEM produces reliable results for systems with arbitrary geometry, but this reliability comes at the expense of high dimensionality [1]. On the other hand, the Assumed Modes method has a manageable number of degrees-of-freedom but is only useful for systems with relatively simple geometry [1]. This work will focus on the Assumed Modes method because of its simplicity and limited computational requirements.

In the Assumed Modes method, the transverse deflection of a deformable body, y, is assumed to be a function of time and space, and these functions are assumed to be independent of one another. This leads to an approximation by a finite set of space-dependent functions, or assumed modes, linearly combined with a set of time-dependent functions, or modal amplitudes.

$$y(x,t) = \sum_{j=1}^{n} \phi_j(x) q_j(t) \tag{2.1}$$

Here, n is the number of modes included in the approximation, x is the distance along the longitudinal axis, ϕ_j is the jth assumed mode, and q_j is the jth generalized coordinate.

For computational purposes, this infinite sum is typically truncated to the sum of less than ten modes and amplitudes. Spatial functions, or assumed modes, are usually

chosen to be a linearly independent set of functions that satisfy various constraints, and the temporal functions, or generalized coordinates, satisfy a set of differential equations derived from the equations of motion for the system.

The simplest and most common set of spatial functions only satisfies the geometric boundary conditions. These functions are called admissible functions. Admissible functions are simple to compute, differentiate, and integrate, all of which will be required to obtain an approximate solution.

A subset of the admissible functions is the slightly more complicated set of comparison functions. The comparison functions satisfy the natural boundary conditions in addition to satisfying the geometric boundary conditions. As one would expect, the increased accuracy requires a more complicated set of assumed modes.

Occasionally, an even higher level of accuracy may be required. For these cases, eigenfunctions are chosen as the assumed modes. Eigenfunctions are a subset of comparison functions. These functions satisfy the geometric and natural boundary conditions as well as the partial differential equations that govern the system. This produces the highest level of accuracy in the assumed modes method. Therefore, for some simple geometries, the eigenfunctions actually form the exact solution to the differential model. All in all, each set of assumed modes has its advantages and disadvantages.

A. Cantilevered Beam Example

The assumed modes method will now be applied to a simple problem. The chosen system is a beam of length L with x as the longitudinal coordinate and y as the transverse displacement. The beam is fixed at the left end, or $x = 0$, and free at the right end, or $x = L$. Figure 1 on the following page is an illustration of this system.
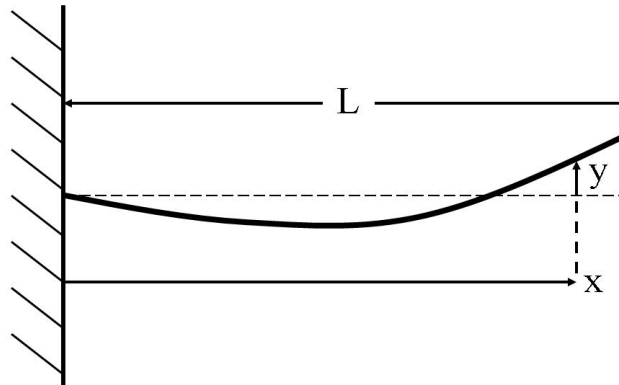
Fig. 1. Simple Cantilevered Beam

For this simple system, the geometric boundary conditions require that the displacement and the slope of the beam at the fixed end are both zero. In addition, the shear stress and moment at the free end are constrained to be zero; these constraints are called the natural boundary conditions.

One derivation of the equations of motion begins by forming the kinetic and potential energies of the system, which are well-known and given by the following two equations.

$$T \ = \ \frac{1}{2} \int_0^L \rho \dot{y}^2 \ dx \tag{2.2}$$

$$V \ = \ \frac{1}{2} \int_0^L EI y''^2 \ dx \tag{2.3}$$

In these equations and those to follow, the dot notation represents differentiation with respect to time, and the prime notation represents differentiation with respect to the longitudinal coordinate.

The assumed modes method presumes that the transverse deflection can be approximated by a linear combination of time- and space-dependent variables as seen

in Equation 2.1 which has been rewritten below.

$$y(x,t) = \sum_{j=1}^{n} \phi_j(x) q_j(t)$$

This equation can be easily substituted into the kinetic and potential energy equations to produce the following results.

$$T = \frac{1}{2} \int_0^L \rho \left(\sum_{j=1}^{n} \phi_j(x) \dot{q}_j(t) \right)^2 dx \qquad (2.4)$$

$$V = \frac{1}{2} \int_0^L \rho \left(\sum_{j=1}^{n} \phi_j''(x) q_j(t) \right)^2 dx \qquad (2.5)$$

From these equations, the differential equations of motion for the generalized coordinates of the beam can be obtained using Lagrange's equations. If there are no externally applied forces, the equations are simply

$$M\ddot{q} + Kq = 0 \qquad (2.6)$$

where the matrices M and K are constant, positive-definite matrices that are functions of the beam parameters and the chosen class of assumed modes. These matrices are defined as in the following equations.

$$M_{ij} \equiv \int_0^L \rho \phi_i(x) \phi_j(x) \, dx \qquad (2.7)$$

$$K_{ij} \equiv \int_0^L EI \phi_i''(x) \phi_j'' \, dx \qquad (2.8)$$

$$\text{for } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, n$$

The chosen set of assumed modes will dictate the numerical values for these matrices and therefore will impact the resulting generalized coordinate time histories. Regardless of the choice of assumed modes, Equation 2.6 will remain a linear matrix equation.

When a solution to the equation of motion for the generalized coordinates is combined with the modal amplitudes, it forms a mathematical model for the motion of a flexible beam. Looking at Equation 2.6 on the preceding page more closely reveals that there is no damping in this system. The lack of damping is a consequence of the model and is rarely valid in reality. Indeed, even simple electrical cables and sensors that are attached to a beam for measurement purposes add damping. To make this model more realistic, the equation will be modified to include viscous damping.

B.   Rayleigh Damping

The exact damping present in structural systems is generally unknown. As a result, the damping is typically approximated as viscous damping [8], which can be implemented by using the following generic equation of motion.

$$M\ddot{q} + C\dot{q} + Kq = 0 \tag{2.9}$$

It is easy to see the similarity between this equation and Equation 2.6 on the page before. There are several ways to generate the damping matrix C, but this research will focus on Rayleigh Damping which is sometimes called proportional damping [8].

The theory behind Rayleigh damping is simple: let the damping be proportional to the mass and stiffness of the system. This is convenient in terms of modeling, but data suggest that this approximation has some experimental validity. The approximation will take the following form

$$C = \alpha M + \beta K \tag{2.10}$$

where α and β are constants that determine the level of viscous damping [1] [8]. To simplify the process of choosing these constants, the equation of motion is diagonalized

and therefore uncoupled.

Uncoupling a system of equations of the form given in Equation 2.6 on page 7 can be accomplished through a simple transformation by the eigenvectors. The eigenvectors come from solving the following equation

$$K\Theta = M\Theta\Lambda \tag{2.11}$$

where $\Lambda = \text{diag}(\lambda_j)$ is the eigenvalue matrix, and $\Theta = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_n]$ is the eigenvector matrix or modal matrix. It can easily be shown that this modal matrix is orthogonal with respect to the mass and stiffness matrices [8], so Θ can be used to diagonalize. If the modal matrix is scaled such that Θ is orthonormal to the mass matrix, then the diagonalization takes the following form

$$\Theta^T M\Theta = [\mathbf{1}] \tag{2.12}$$

$$\Theta^T K\Theta = \text{diag}(\omega_1^2, \omega_2^2, \ldots, \omega_n^2) \tag{2.13}$$

where ω_j is the jth natural frequency of the system. Proportional damping guarantees that the modal matrix is orthogonal to the damping matrix as well, so the diagonal form of the viscous damping can be assumed to be the following

$$\Theta^T C\Theta = \text{diag}(2\zeta_1\omega_1, 2\zeta_2\omega_2, \ldots, 2\zeta_n\omega_n) \tag{2.14}$$

where ζ_j is the jth damping coefficient. This diagonalization will lead to a system of uncoupled equations where the rth equation has the form shown below.

$$\ddot{\eta}_r + 2\zeta_r\omega_r\dot{\eta}_r + \omega_r^2\eta_r = 0 \tag{2.15}$$

Values for ζ_r between zero and one correspond to an underdamped mode, a value equal to one corresponds to a critically damped mode, and values greater than one

correspond to an overdamped mode.

Recalling the equation for the proportional damping coefficient, Equation 2.10 on page 8, the following relationship can be realized.

$$2\zeta_r\omega_r = C_r = \alpha(1) + \beta\omega_r^2 \tag{2.16}$$

This equation can be solved for the rth damping coefficient.

$$\zeta_r = \frac{\alpha}{\omega_r} + \frac{\beta\omega_r}{2} \tag{2.17}$$

From this equation, it is clear that choosing two damping coefficients will provide unique values for α and β. Further examination of this equation reveals that at high frequencies, the damping is dominated by β, and at low frequencies, the damping is dominated by α. If the designer wishes to choose all of the damping coefficients as opposed to only choosing two, modal damping maybe be used in place of Rayleigh damping [8].

Including proportional damping in the system equation of motion yields the following equation.

$$M\ddot{q} + C\dot{q} + Kq = 0 \tag{2.18}$$

C. Implementation

As previously stated, there are three main choices for assumed modes. These choices along with candidate functions are listed here:

- Admissible functions: satisfy only the geometric boundary conditions.

$$\phi_j(x) = \left(\frac{x}{L}\right)^{j+1} \tag{2.19}$$

- Comparison functions: a subset of admissible functions that also satisfy the

natural boundary conditions.

$$\phi_j(x) = 1 - \cos\left(\frac{j\pi x}{L}\right) + \frac{1}{2}\left(\frac{j\pi x}{L}\right)^2 (-1)^{j+1} \qquad (2.20)$$

- Eigenfunctions: a subset of comparison functions that satisfy the partial differential equation of the system. For a detailed derivation and explanation of the eigenfunction solution, the reader is referred to Junkins and Kim, pg. 175-184 [1].

After choosing a set of assumed modes, the mass and stiffness matrices are calculated according to Equations 2.7 to 2.8 on page 7, respectively. The differential equation of motion for the generalized coordinates, $M\ddot{q} + C\dot{q} + Kq = 0$, is then solved using an numerical solver. For the simulation in this section, three assumed modes were used. To obtain the estimated deflection, linearly combine the generalized coordinates with the assumed modes calculated at the desired measurement points along the span.

Table I. Beam Parameters

Length	45.52 in
Modulus of Elasticity	161.6e6 oz/in^2
Second Moment of Inertia of Bending Section	0.000813 in^4
Density	0.003007 oz s^2/in

The beam deflection plots result from applying the assumed modes method to a beam with the characteristics given in Table I. There are three measurement points: at the quarter-span, the midspan, and the tip. For simulation purposes, the initial conditions for these measurement points were defined as two, four, and eight inches respectively. This corresponds to initial deflections of approximately twenty percent of the distance from the wall to the measurement location. Figures 2 and 3 on page
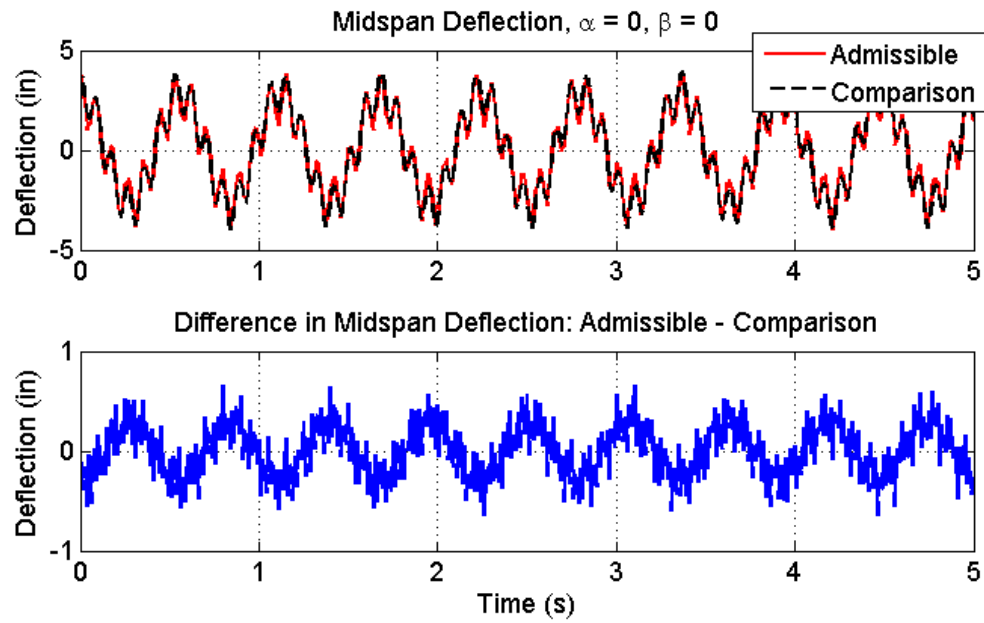
Fig. 2. Midspan Deflection without Damping
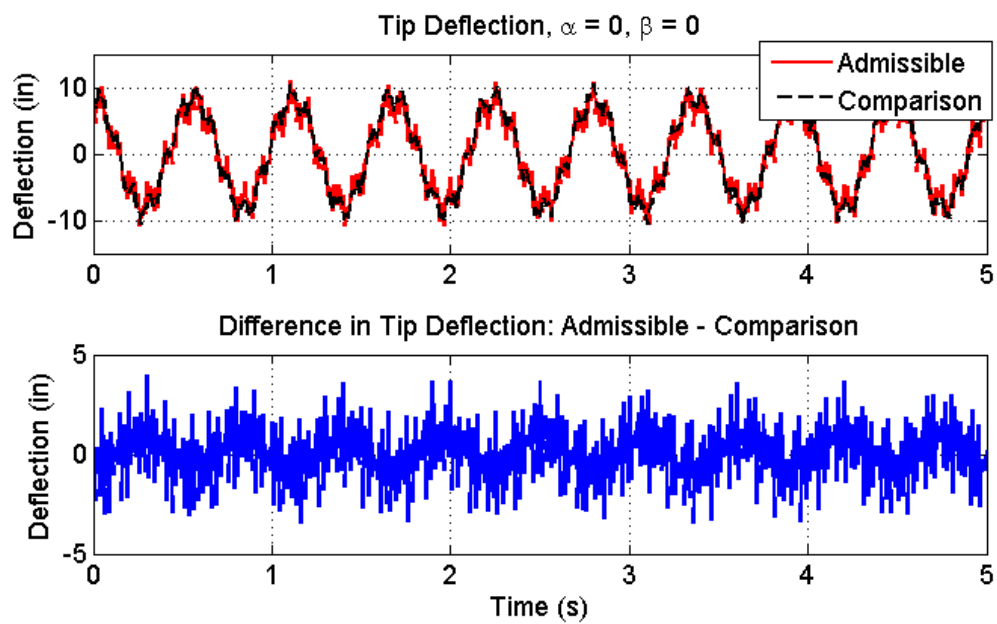
Fig. 3. Tip Deflection without Damping

12 show the simulation results for the midspan and tip deflections simulated without proportional damping.

From these figures, it is easy to see that the difference in deflection between the two simulation models is on the order of ten percent. The second figure also shows that the error between the Admissible and Comparison solutions at the tip appears to oscillate. This corresponds to the differences in natural frequencies for the two systems. The differences become more apparent as the number of modes increases. Table II is a comparison between the natural frequencies of the two simulation models.

Table II. Comparison of Natural Frequencies (Hz)

	Admissible	Comparison
ω_1	1.785	1.786
ω_2	11.29	11.28
ω_3	59.98	31.79
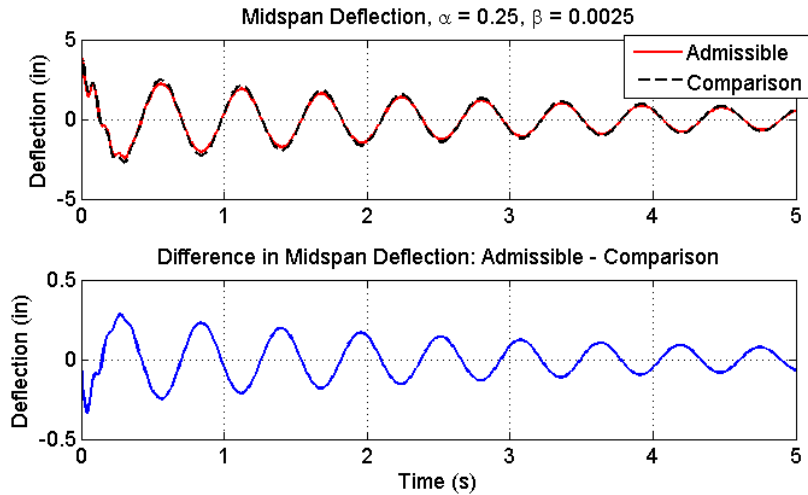
Fig. 4. Midspan Deflection of Beam with Proportional Damping

The same system simulations were run with proportional damping. For Figures 4 and 5 on pages 13-14, the proportional damping constants were $\alpha = 0.25$ and $\beta = 0.0025$. Comparing the figures with proportional damping to those without damping reveals that the error between the two types of assumed modes is significantly less for the simulation with damping. With damping, the error in the tip deflection is on the order of ten percent whereas the error calculated without damping is closer to twenty percent.
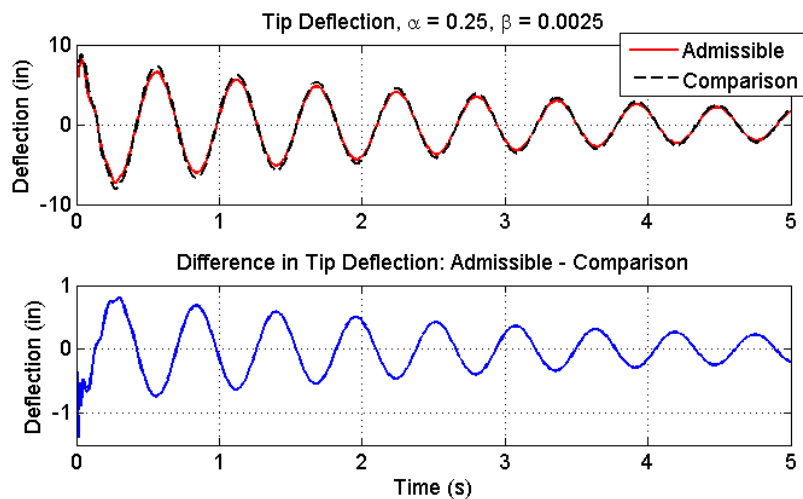
Fig. 5. Tip Deflection of Beam with Proportional Damping

It appears that as long as damping is included in the system then the Assumed Modes method calculated using admissible versus comparison functions yields similar results. This coupled with the simplicity of the admissible functions led the author to prefer admissible functions for the simulations included in this work.

CHAPTER III

STRUCTURAL DYNAMICS OF ROTATING SYSTEMS

As seen in the previous chapter, the assumed modes method produces linear systems
for some simple flexible body configurations. For rotating systems, the assumed
modes method yields results that are slightly more complex. One such example
is spacecraft, flexible appendage combination performing a single-axis rotation. A
derivation of the equations of motion for this system is straightforward and follows
the pattern for the simple cantilevered beam in the previous chapter.

The system is defined to be a spacecraft with cylindrical cross-section and a
flexible appendage modeled as a cantilevered beam. It is also assumed that the
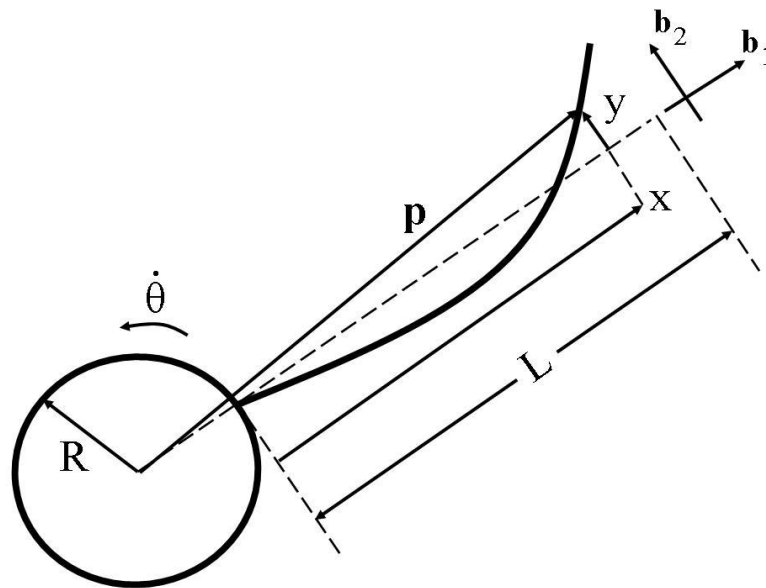
Fig. 6. Rotating Spacecraft with Flexible Appendage

longitudinal and transverse axes of each beam lie in a plane and there is no motion
out of this plane. These assumptions are not necessary, but they simplify the problem
without interfering with the analysis. Figure 6 is a diagram of this system.

A.   Linear Assumed Modes Model

As before, the derivation begins by writing the kinetic and potential energies. However, the kinetic energy now has two parts: one for the hub and one for the flexible appendage.

$$T \; = \; T_{\text{hub}} + T_{\text{flex}} \tag{3.1}$$

$$= \; \frac{1}{2} J_{\text{hub}} \dot{\theta}^2 + \frac{1}{2} \int_0^L \rho\, \dot{\boldsymbol{p}} \cdot \dot{\boldsymbol{p}}\; dx \tag{3.2}$$

The potential energy remains unchanged.

$$V = \frac{1}{2} \int_0^L EIy'' \; dx \tag{3.3}$$

Variables $\boldsymbol{p}$, x, y, L, and $\dot{\theta}$ are defined as in Figure 6 on the page before, J_{hub} is the inertia of the spacecraft hub, and variables ρ and EI are parameters that depend on the size and material of the beam.

Before proceeding, it is necessary to investigate the kinematics for a generic point on the beam, denoted by $\boldsymbol{p}$ in the previous figure. The radius to a differential mass element can be defined from the center of mass of the system in a body-fixed coordinate frame.

$$\boldsymbol{p} = (R + x)\; \hat{\boldsymbol{b}}_1 + y\; \hat{\boldsymbol{b}}_2 \tag{3.4}$$

If the spacecraft is spinning, the first derivative of the position can be easily calculated.

$$\dot{\boldsymbol{p}} = (-y\, \dot{\theta})\; \hat{\boldsymbol{b}}_1 + (\dot{y} + (R + x)\, \dot{\theta})\; \hat{\boldsymbol{b}}_2 \tag{3.5}$$

This expression can now be substituted, and after simplification, the kinetic energy is the following:

$$T = \frac{1}{2} \hat{J} \dot{\theta}^2 + \int_0^L \rho \left(y^2 \dot{\theta}^2 + \dot{y}^2 + 2 \dot{y} \dot{\theta}(R + x) \right)\; dx \tag{3.6}$$

where $\hat{J} \equiv J_{\mathrm{hub}} + \int_0^L \rho\,(R+x)^2\,dx = J_{\mathrm{hub}} + \rho(R^2L + RL^2 + 1/3L^3)$.

At this point, the transverse displacement can be replaced by its assumed modes approximation which was introduced in the previous chapter and is rewritten below.

$$y(x,t) = \sum_{j=1}^{n} \phi_j(x)q_j(t) \tag{3.7}$$

Substituting this expression into the kinetic energy equation yields the following equation.

$$T = \frac{1}{2}\hat{J}\dot{\theta}^2 + \frac{1}{2}\int_0^L \rho\left[\left(\phi_j q_j \dot{\theta}\right)^2 + (\phi_j \dot{q}_j)^2 + 2\dot{\theta}(R+x)\phi_j \dot{q}_j\right]dx \tag{3.8}$$

For simplicity, the summation symbols have been dropped, and index notation will be employed in its place. From this point forward, it will also be implied that q is a function of time and ϕ is a function of the longitudinal coordinate.

The q_j terms are not functions of x, so these can be moved outside the integral. This will allow for simplification by defining two constant matrices that are functions of the assumed modes.

$$M_{ij} \equiv \int_0^L \rho\,\phi_i\phi_j\,dx \tag{3.9}$$

$$N_j \equiv \int_0^L \rho\,(R+x)\,\phi_j\,dx \tag{3.10}$$

Here, M is the mass matrix, and N contains the coupling between the rotational and deformational motion. The kinetic energy equation can now be written in the following simplified form.

$$T = \frac{1}{2}\hat{J}\dot{\theta}^2 + N_j\dot{\theta}\dot{q}_j + \frac{1}{2}M_{ij}\dot{q}_i\dot{q}_j + \frac{1}{2}M_{ij}q_iq_j\dot{\theta}^2 \tag{3.11}$$

This is the final form of the kinetic energy equation.

Now, the focus returns to the potential energy from Equation 3.3 on the preceding page. This equation can also be simplified by defining a constant matrix called the

stiffness matrix.

$$K_{ij} \equiv \int_0^L EI\phi_i''\phi_j''dx \tag{3.12}$$

Using this definition, the potential energy can be rewritten as

$$V = \frac{1}{2}K_{ij}q_iq_j \tag{3.13}$$

where the summation symbols have again been dropped in favor of index notation. This equation is the final form of the potential energy equation.

The kinetic and potential energies can now be combined to form the Lagrangian. After taking the necessary derivatives, one can obtain the equations of motion for this system, written below in matrix form

$$\begin{cases} M\ddot{q} + N\ddot{\theta} + \left(K - M\dot{\theta}^2\right)q &= 0 \\ \left(\hat{J} + q^TMq\right)\ddot{\theta} + N^T\ddot{q} + 2q^TM\dot{q}\dot{\theta} &= v \end{cases} \tag{3.14}$$

where v is the control torque applied to the spacecraft. Because this development mirrors that of the flexible beam in the previous chapter, it is not surprising that there is no damping in this system either, and as stated in that chapter, the equations of motion without damping do not provide a realistic model of the system. Therefore, damping will be added to both the flexible and rotational equations. Applying Rayleigh damping in the flexible body equation and adding proportional damping in the rotational equation yields the following set of equations that will be implemented for the Linear Assumed Model in this work.

$$\begin{cases} M\ddot{q} + N\ddot{\theta} + C\dot{q} + \left(K - M\dot{\theta}^2\right)q &= 0 \\ \left(\hat{J} + q^TMq\right)\ddot{\theta} + N^T\ddot{q} + C_\theta\dot{\theta} + 2q^TM\dot{q}\dot{\theta} &= v \end{cases} \tag{3.15}$$

Figure 7 on page 20 depicts the motion of this system in response to an initial deflection of two, four, and eight inches at the quarter-span, midspan, and tip as well as an

initial rotation rate of 0.3 rad/s. The beam parameters for the spacecraft simulations are the same as for the flexible beam simulations in the previous chapter. These parameters along with the hub dimensions can be seen in Table III and will be used throughout this work.

Table III. Rotational Spacecraft Parameters

Beam Length	45.52 in
Beam Modulus of Elasticity	161.6e6 oz/in^2
Second Moment of Inertia of Bending Section	0.000813 in^4
Beam Density	0.003007 oz s^2/in
Hub Radius	5.5470 in
Hub Mass	31.59 oz
Rayleigh Damping Coefficient, α	0.25
Rayleigh Damping Coefficient, β	0.0025

There are several items of interest in this set of equations. First, analyze the case of zero rotational acceleration. For this case, the the angular velocity is a constant, and the equations of motion for the system are just the equations for the generalized coordinates. Also note that the first set of equations by themselves represent a linear system for this case.

The addition of a control torque requires the addition of the rotational equation which is linear by itself but nonlinear when coupled with the generalized coordinates. These equations together govern the overall motion of the system.

The fourth term in the first equation also shows that as the rotation rate increases, the beam appears to soften. This is actually counterintuitive because one would expect the beam to stiffen as the spacecraft spins faster. There are several
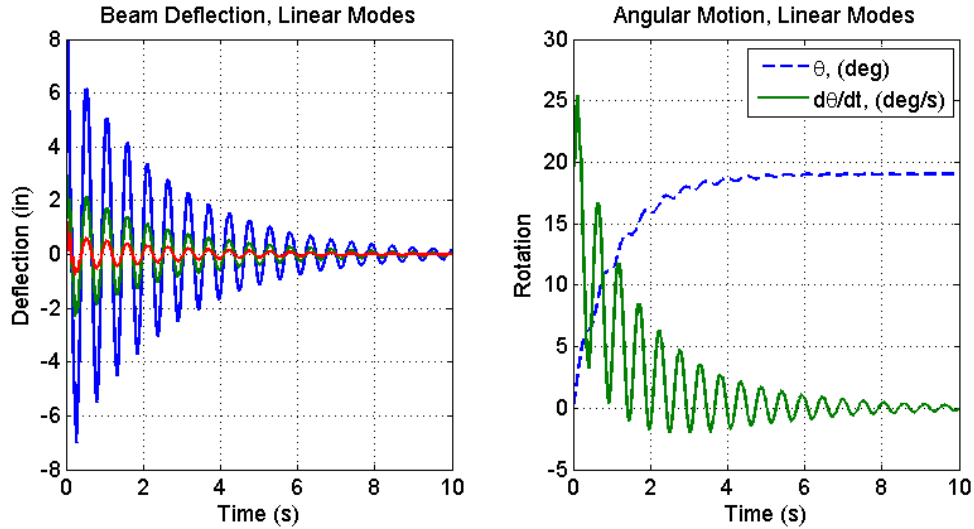
Fig. 7. Spacecraft Motion, Linear Modes

solutions to this apparent error, and this study will focus on the quadratic assumed model to solve this problem [2].

B.   Quadratic Assumed Modes Model

The previous section highlighted a derivation of the equations of motion for a rotating spacecraft with a flexible appendage. The resulting equations showed that as the spacecraft spins faster, the flexible structure will soften. However, intuition leads one to believe that there should be a stiffening instead.

Quadratic assumed modes were created as a solution to this problem [2][3]. The quadratic assumed mode allow for larger transverse deflections as well as for motion along the longitudinal axis. A derivation of the equations of motion for this method is outlined below.

This method begins by assuming that there is generic deformable body attached at a point to a cylindrical rigid hub as seen in Figure 8 on the next page. The $\hat{\boldsymbol{b}}$ frame is a body-fixed frame. A vector from the center of the hub to a generic mass element
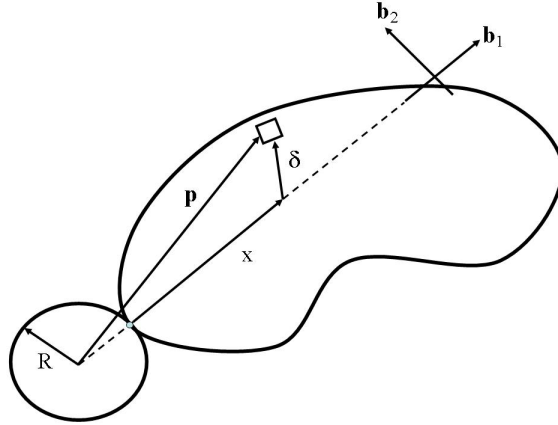
Fig. 8. Rotating Spacecraft with Deformable Body

can be defined by

$$\boldsymbol{p} = (R + x)\ \hat{\boldsymbol{b}}_1 + \boldsymbol{\delta} \tag{3.16}$$

where R is the radius of the hub, x is the distance from the point of attachment to the original location of the mass element along the $\hat{\boldsymbol{b}}_1$ axis, and $\boldsymbol{\delta}$ is the vector from the end of the length x to the generic mass element. The vector $\boldsymbol{\delta}$ can be split into components in the $\hat{\boldsymbol{b}}$ frame.

$$\boldsymbol{\delta}(x, t) = u\ \hat{\boldsymbol{b}}_1 + y\ \hat{\boldsymbol{b}}_2 \tag{3.17}$$

This equation can then be substituted into the previous one to produce the following equation.

$$\boldsymbol{p} = (R + x + u)\ \hat{\boldsymbol{b}}_1 + y\ \hat{\boldsymbol{b}}_2 \tag{3.18}$$

At this point, it is necessary to impose the constraint that each segment is inextensible, which means that a differential section retains its length. From Figure 9 on the following page, the corresponding constraint equation is,

$$\|P_2 - P_1\| = \|P_2' - P_1'\| \tag{3.19}$$
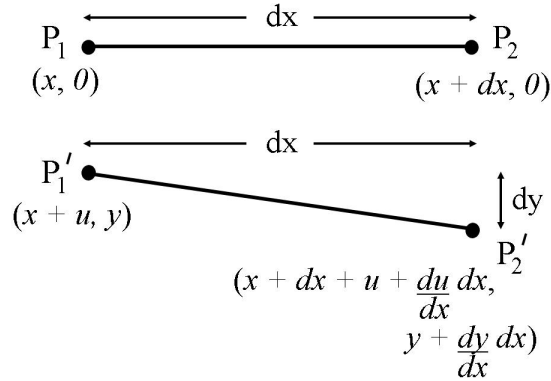
Fig. 9. Inextensible Section

and each side can be handled independently.

$$\|P_2 - P_1\| \;=\; (x + dx - x)^2 + (0 - 0)^2 \tag{3.20}$$

$$\|P_2' - P_1'\| \;=\; (dx + \frac{du}{dx}dx)^2 + (\frac{dy}{dx}dx)^2 \tag{3.21}$$

Equating both sides produces the following result after dividing by dx^2.

$$1 = 1 + 2\,\frac{du}{dx} + \underbrace{\left(\frac{du}{dx}\right)^2}_{assume\approx 0} + \left(\frac{dy}{dx}\right)^2 \tag{3.22}$$

It is important to notice that the third term on the right side of the previous equation has been assumed to be approximately equal to zero. This is the only assumption that has been made thus far.

Integrating this equation with respect to x produces an equation for the longitudinal displacement.

$$u(x,t) = -\frac{1}{2}\int_0^x \left(\frac{dy}{dx}\right)^2 \, dx \tag{3.23}$$

The assumed modes simplification can then be substituted into Equation 3.23 to

obtain the following equation.

$$u(x,t) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_i q_j \int_0^x -\frac{1}{2}\phi_i'\phi_j' \tag{3.24}$$

This leads to the definition of the quadratic assumed modes [2] which is given by Equation 3.25.

$$\psi_{ij}(x) = -\frac{1}{2}\int_0^x \phi_i'\phi_j' dx \tag{3.25}$$

Now, the longitudinal displacement can be written as seen below.

$$u(x,t) = \sum_{i=1}^{n}\sum_{j=1}^{n} \psi_{ij} q_i q_j \tag{3.26}$$

Finally, an equation for the motion along both the axial and transverse directions is formed.

$$\boldsymbol{\delta}(x,t) = \sum_{i=1}^{n}\sum_{j=1}^{n} \psi_{ij} q_i q \ \hat{\boldsymbol{b}}_1 + \sum_{i=1}^{n} q_i \phi_i \ \hat{\boldsymbol{b}}_2 \tag{3.27}$$

This is the final form of the equation for displacement of a generic mass element.

With this modified definition of the distance to a differential mass element, the derivation of the flexible spacecraft equations of motion can continue with the quadratic assumed modes. In this derivation as in the one prior, the summation notation has been dropped in favor of index notation. From above, one can substitute the expression for $\boldsymbol{\delta}$ into Equation 3.18 on page 21 and take the time derivative to obtain an expression for the velocity of an arbitrary point on the beam.

$$\dot{\boldsymbol{p}} = \left(2\psi_{ij}\dot{q}_i q_j - \phi_i q_i \dot{\theta}\right) \ \hat{\boldsymbol{b}}_1 + \left((R + x + \psi_{ij}q_i q_j)\dot{\theta} + \phi_i \dot{q}_i\right) \ \hat{\boldsymbol{b}}_2 \tag{3.28}$$

This leads to the kinetic energy of the flexible appendage.

$$T_{flex} = \frac{1}{2}\int_0^L \rho[\underbrace{4\psi_{ij}\psi_{lk}\dot{q}_i q_j \dot{q}_l q_k}_{\text{H.O.T.}} - \underbrace{4\phi_i\psi_{lj}q_i\dot{q}_l q_j \dot{\theta}}_{\text{H.O.T.}} + \dot{\theta}^2\phi_i\phi_j q_i q_j$$

$$+\dot{\theta}^2((R+x)^2 + 2(R+x)\psi_{ij}q_i q_j + \underbrace{\psi_{ij}\psi_{lk}q_i q_j q_l q_k}_{\text{H.O.T.}}) \qquad (3.29)$$

$$+\phi_i\dot{q}_i\dot{\theta}(2(R+x) + \underbrace{2\psi_{lj}q_l q_j}_{\text{H.O.T.}}) + \phi_i\phi_j\dot{q}_i\dot{q}_j]\ dx$$

The higher-order terms (H.O.T.) in this expression consist of products of more than two assumed modes which will lead to nonlinear equations of motion if they are included. To obtain linear equations, these higher-order terms will be neglected. Before writing the final form of the kinetic energy, another matrix needs to be defined.

$$H_{ij} \equiv \int_0^L \rho\ (R+x)\psi_{ij}\ dx \qquad (3.30)$$

This term is the quadratic modes analogy to $N_i = \int_0^L \rho\ (R+x)\phi_i\ dx$ for the linear assumed modes. Now, the kinetic energy equation can be written in its final form as

$$T = \frac{1}{2}\hat{J}\dot{\theta}^2 + \frac{1}{2}M_{ij}q_i q_j\dot{\theta}^2 + \frac{1}{2}M_{ij}\dot{q}_i\dot{q}_j + H_{ij}q_i q_j\dot{\theta}^2 + N_j\dot{q}_j\dot{\theta} \qquad (3.31)$$

where the remaining variables are defined as before. For the quadratic assumed modes derivation, the potential energy equation is the same as that used in the linear assumed modes derivation.

Applying Lagrange's equations yields the following equations of motion, which were developed using quadratic assumed modes. Again, damping is added to this set of equations, as with the linear modes, to maintain consistency. Figure 10 on the next page is a plot of the beam deflection and associated rotation of the spacecraft

with respect to time.

$$
\begin{cases}
M\ddot{q} + N\ddot{\theta} + C\dot{q} + \left(K - \dot{\theta}^2(M + 2H)\right)q &= 0 \\
\left(\hat{J} + q^T(M + 2H)q\right)\ddot{\theta} + N^T\ddot{q} + C_\theta\dot{\theta} + 2q^T(M + 2H)\dot{q}\dot{\theta} &= v
\end{cases}
\tag{3.32}
$$
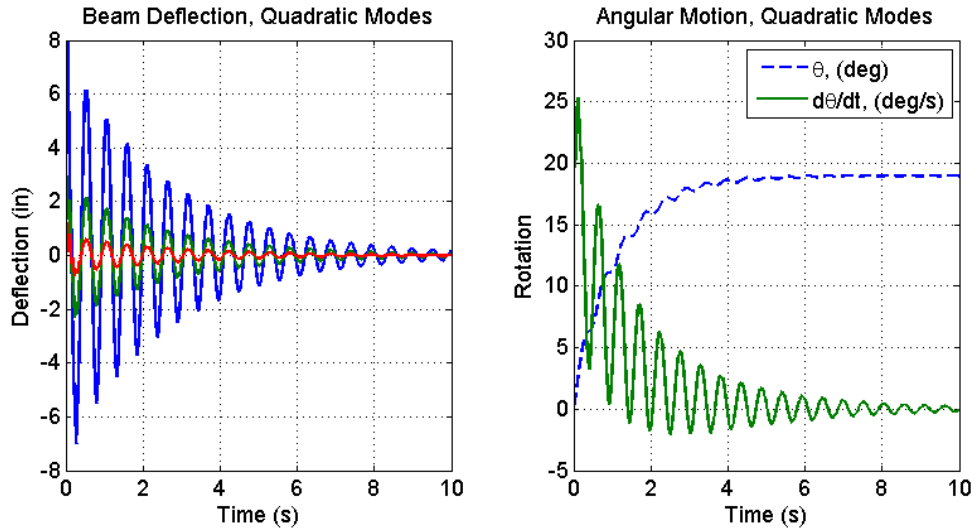
Fig. 10. Spacecraft Motion, Quadratic Modes

This set of equations yields a few interesting results. For the case of constant rotation rate, the beam appears to exhibit the same softening with increased rotation as seen with the linear modes. However, the term $(M + 2H)$ can be shown to be negative definite [9], which implies that the beam actually experiences a stiffening as the spacecraft rotation rate increases. The special case of constant or prescribed rotation rate also produces a linear system. When the rotation rate is variable, the system becomes nonlinear due to the coupling between the flexible and rotational motion.

C.   Quadratic Assumed Modes Model with Nonlinear Terms

The quadratic assumed modes model provides a more realistic representation of the motion of the given rotational system than does the linear assumed modes model. However, the observant reader will recall that a number of higher-order terms were neglected in the derivation of the quadratic model. The fact that the quadratic assumed modes provide a more realistic picture without the higher-order terms begs the question of how the equations would differ if those nonlinear terms were not neglected. A derivation follows that develops the equations of motion for the system using the quadratic modes while keeping the higher-order terms.

From the previous section, the equation for kinetic energy with H.O.T.'s has been rewritten below.

$$
\begin{aligned}
T_{flex} = \ & \tfrac{1}{2} \int_0^L \rho [\underbrace{4\psi_{ij}\psi_{lk}\dot{q}_i q_j \dot{q}_l q_k}_{\text{H.O.T.}} - \underbrace{4\phi_i \psi_{lj} q_i \dot{q}_l q_j \dot{\theta}}_{\text{H.O.T.}} + \dot{\theta}^2 \phi_i \phi_j q_i q_j \\
& + \dot{\theta}^2 ((R+x)^2 + 2(R+x)\psi_{ij} q_i q_j + \underbrace{\psi_{ij}\psi_{lk} q_i q_j q_l q_k}_{\text{H.O.T.}}) \\
& + \phi_i \dot{q}_i \dot{\theta}(2(R+x) + \underbrace{2\psi_{lj} q_l q_j}_{\text{H.O.T.}}) + \phi_i \phi_j \dot{q}_i \dot{q}_j]\ dx
\end{aligned}
$$

This equation can be split into a linear and a nonlinear portion: $T = T_L + T_{NL}$. The linear portion, T_L, was thoroughly examined in the previous section, so the focus now turns to the nonlinear portion.

$$
T_{NL} = \frac{1}{2} \int_0^L \rho \left[4\psi_{ij}\psi_{lk}\dot{q}_i q_j \dot{q}_l q_k - 4\dot{\theta}\phi_i \psi_{lj} q_i \dot{q}_l q_j + \dot{\theta}^2 \psi_{ij}\psi_{lk} q_i q_j q_l q_k + 2\dot{\theta}\phi_i \psi_{lj}\dot{q}_i q_l q_j \right]
$$

$$(3.33)$$

Close examination of the nonlinear terms reveals that there are two types of nonlin-

earity. To simplify the subsequent derivation, the following tensors are defined.

$$F_{ilj} \equiv \int_0^L \rho\ \phi_i \psi_{lj}\ dx \tag{3.34}$$

$$G_{ijlk} \equiv \int_0^L \rho\ \psi_{ij} \psi_{lk}\ dx \tag{3.35}$$

Substituting these tensors into the kinetic energy, the nonlinear terms simplify to the following.

$$T_{NL} = G_{ijlk} \left(2\dot{q}_i q_j \dot{q}_l q_k + \frac{1}{2}\dot{\theta}^2 q_i q_j q_l q_k \right) + F_{ilj}\dot{\theta}\left(-2q_i \dot{q}_l q_j + \dot{q}_i q_l q_j \right) \tag{3.36}$$

This result can be used to derive the nonlinear portion of the equations of motion for the rotating spacecraft with a flexible appendage.

The linear portion of the equations of motion is the same as that obtained in the previous section. Therefore, the remaining terms can be obtained by applying taking appropriate partials of the nonlinear portion. The potential energy still leads to a set of linear equations, so $L_{NL} = T_{NL} - 0$. Application of Lagrange's equations will continue in index notation.

The derivation proceeds by taking partial derivatives with respect to the individual generalized coordinates and with respect to the rotational coordinate. Taking the partial derivative with respect to the first derivative of the sth generalized coordinate yields

$$\frac{\partial T_{NL}}{\partial \dot{q}_s} = 2G_{ijlk}q_j q_k \left(\frac{\partial \dot{q}_i}{\partial \dot{q}_s}\dot{q}_l + \dot{q}_i \frac{\partial \dot{q}_l}{\partial \dot{q}_s} \right) - 2\dot{\theta}F_{ilj}q_i q_j \frac{\partial \dot{q}_l}{\partial \dot{q}_s} + \dot{\theta}F_{ilj}q_l q_j \frac{\partial \dot{q}_i}{\partial \dot{q}_s} \tag{3.37}$$

In index notation, the term $\frac{\partial \dot{q}_i}{\partial \dot{q}_s}$ reduces to δ_{is}, where δ is the substitution operator [10]. Application of this property and simple manipulation yields the partial with respect to $\dot{q}_s$.

$$\frac{\partial T_{NL}}{\partial \dot{q}_s} = 2q_j q_k \left(G_{sjlk}\dot{q}_l + G_{ijsk}\dot{q}_i \right) + \dot{\theta}q_l q_j \left(F_{slj} - 2F_{lsj} \right) \tag{3.38}$$

This equation must be differentiated with respect to time. Foresight dictates that the tensor symmetry will be explored before proceeding with the differentiation.

The definitions of the F and G tensors are rewritten here.

$$F_{ilj} \equiv \int_0^L \rho \, \phi_i \psi_{lj} \, dx$$

$$G_{ijlk} \equiv \int_0^L \rho \, \psi_{ij} \psi_{lk} \, dx$$

The definition of the ψ term is also repeated.

$$\psi_{ij}(x) = -\frac{1}{2} \int_0^x \phi_i' \phi_j' dx$$

From this equation, it is obvious that $\psi_{ij} = \psi_{ji}$. This observation easily leads to the following

$$G_{ijlk} = G_{jilk} \tag{3.39}$$

$$G_{ijlk} = G_{ijkl} \tag{3.40}$$

$$G_{ijlk} = G_{lkij} \tag{3.41}$$

$$F_{ilj} = F_{ijl} \tag{3.42}$$

which will be defined as symmetry properties of the higher-order tensors. These properties are easily applied in the partial derivative of Equation 3.38 on the preceding page. Simplifying and collecting terms yields the following equation.

$$\frac{d}{dt}\left(\frac{\partial T_{NL}}{\partial \dot{q}_s}\right) = 2\left(\dot{q}_j q_k \dot{q}_l + q_j \dot{q}_k \dot{q}_l + q_j q_k \ddot{q}_l\right)\left(G_{sjlk} + G_{ljsk}\right) \tag{3.43}$$
$$+ \left(\ddot{\theta} q_l q_j + \dot{\theta} \dot{q}_l q_j + \dot{\theta} q_l \dot{q}_j\right)\left(-2F_{lsj} + F_{slj}\right)$$

Next, the partial derivative of the nonlinear kinetic energy terms with respect to

the sth generalized coordinate is taken to produce the following, after simplification.

$$
\begin{aligned}
\frac{\partial T_{NL}}{\partial q_s} \;=\;& G_{islk}\left(2\dot{q}_i\dot{q}_l q_k + \dot{\theta}^2 q_i q_l q_k\right) + G_{ijls}\left(2\dot{q}_i\dot{q}_l q_j + \dot{\theta}^2 q_i q_j q_l\right) \\
& -2\dot{\theta}\dot{q}_l q_j F_{slj} + F_{ils}\left(-2\dot{\theta}q_i\dot{q}_l + 2\dot{\theta}\dot{q}_i q_l\right)
\end{aligned}
\tag{3.44}
$$

This and the previous equation are then combined along with the terms in Equation 3.32 on page 25 which were obtained by neglecting H.O.T. in the quadratic modes derivation. The F and G terms are collected, rearranged, and regrouped to obtain the simplified final form of the differential equation of motion for the sth generalized coordinate.

A similar procedure is used to obtain the rotational differential equation of motion. From the nonlinear kinetic energy, partial derivatives are taken, and the results after simplification are shown below.

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial T_{NL}}{\partial \dot{\theta}}\right) \;=\;& \ddot{\theta}\, G_{ijlk} q_i q_j q_l q_k + 2\dot{\theta}\, G_{ijlk} q_j q_k (\dot{q}_i q_l + q_i \dot{q}_l) \\
& + F_{ilj}\left(\ddot{q}_i q_l q_j - 2 q_i \ddot{q}_l q_j - 2 q_i \dot{q}_l \dot{q}_j\right)
\end{aligned}
\tag{3.45}
$$

$$
\frac{\partial T_{NL}}{\partial \theta} \;=\; 0
\tag{3.46}
$$

These terms are combined with the linear terms obtained from the quadratic modes derivation. For the forcing function v, the final form of the equations of motion for the system is the following.

$$
\begin{cases}
(M + \mathcal{M}_{NL})\ddot{q} + (N + \mathcal{N}_{NL})\ddot{\theta} + C\dot{q} + \mathcal{K}_L q + \mathcal{K}_{NL} \;=\; 0 \\
(N + \mathcal{N}_{NL})^T \ddot{q} + (\mathcal{D}_L + \mathcal{D}_{NL})\ddot{\theta} + C_\theta \dot{\theta} + 2\dot{\theta}q^T(M + 2H + \mathcal{E}_G)q - \mathcal{E}_F \;=\; v
\end{cases}
\tag{3.47}
$$

where the newly introduced coefficients are defined as follows.

$$\mathcal{M}_{NL} = 2q_jq_k(G_{sjlk} + G_{ljsk}) \tag{3.48}$$

$$\mathcal{N}_{NL} = q_lq_j(F_{slj} - 2F_{jls}) \tag{3.49}$$

$$\mathcal{K}_L = K - \dot{\theta}^2(M + 2H) \tag{3.50}$$

$$\mathcal{K}_{NL} = \mathcal{K}_F + \mathcal{K}_G \tag{3.51}$$

$$\mathcal{K}_F = 2\dot{\theta}\dot{q}_j(F_{slj}(q_l + \dot{q}_l) - 2F_{jls}q_l) \tag{3.52}$$

$$\mathcal{K}_G = (2\dot{q}_jq_k\dot{q}_l - \dot{\theta}^2q_jq_kq_l)(G_{sjlk} + G_{ljsk}) \tag{3.53}$$

$$\mathcal{D}_L = \hat{J} + q^T(M + 2H)q \tag{3.54}$$

$$\mathcal{D}_{NL} = G_{sjlk}q_sq_jq_lq_k \tag{3.55}$$

$$\mathcal{E}_F = 2F_{sjl}q_s\dot{q}_l\dot{q}_j \tag{3.56}$$

$$\mathcal{E}_G = G_{sjlk}(\dot{q}_sq_l + q_s\dot{q}_l) \tag{3.57}$$

The equations of motion for the quadratic assumed model with nonlinear terms, Equation 3.47 on the preceding page, is a set of nonlinear equations even for the case of prescribed rotation rate. In contrast, the reader may recall that the linear and quadratic assumed modes models produced sets of linear equations.

Figure 11 on the next page is a plot of the rotational spacecraft motion with respect to time. The same spacecraft and beam parameters were used for this set of equations as for the previous two, so the following section will compare the motion simulated using these three derivations.

D. Comparison of Results

The linear and quadratic assumed modes models produce beam deflections that are quite similar. A comparison of the output from these two sets of equations can be
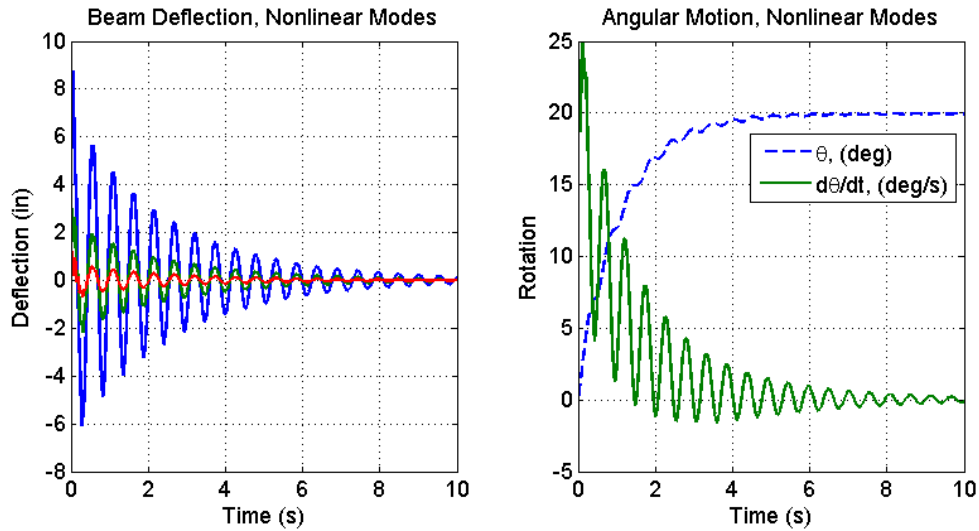
Fig. 11. Spacecraft Motion, Nonlinear Modes

seen in Figure 12. The initial rotation rate was 0.3 rad/s or slightly less than twenty degrees per second. This is not a large angular velocity, so the differences between the beam motion for these two methods are quite small in comparison to the magnitude of the motion. The figures plot true difference instead of percent difference due to the large number of zero crossings. Measurement differences for the quarter-span, midspan, and tip are the three lines plotted in this figure.

From this figure, one might wonder if the differences between the linear and quadratic assumed models are significant. It was hypothesized that these two derivations would produce different output due to the rotation rate. To test this hypothesis, the equations were simulated with various initial rotation rates. Figure 13 on the following page shows the quadratic minus linear tip deflection for three different initial rotation rates given in radians per second, and this figure reveals that as the rotation rate increases, the two simulations begin to produce significantly different results. It was previously stated that as the angular velocity increased, the quadratic assumed models would experience a stiffening whereas the linear assumed models would expe-
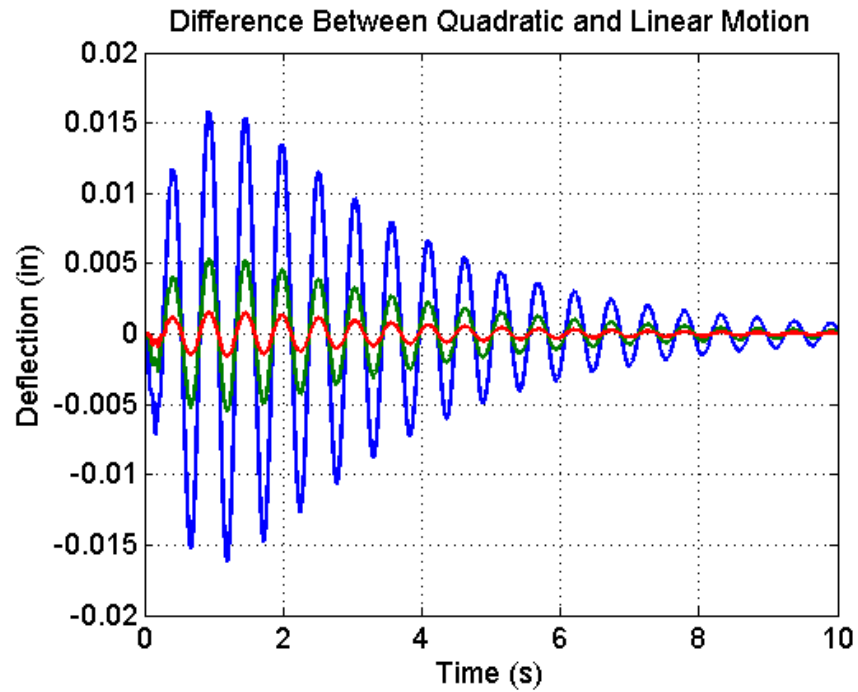
Fig. 12. Difference Between Quadratic and Linear Motion
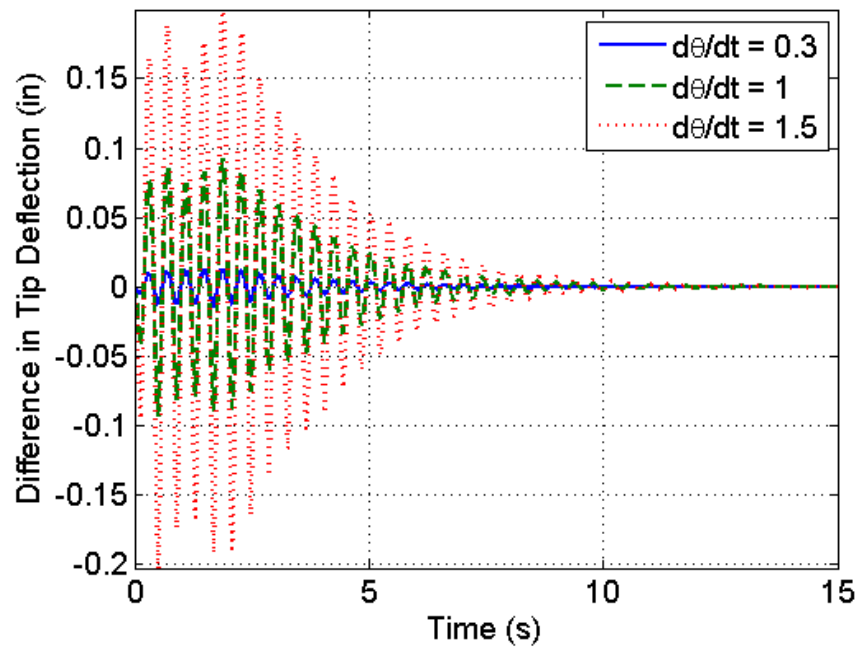
Fig. 13. Difference in Tip Deflection for Various Initial Rotation Rates

rience a softening. This can be easily shown by assuming the rotation rate is fixed, calculating the effective mass and stiffness matrices for both methods, and then calculating the natural frequencies. The effective stiffness matrices for the linear and quadratic assumed modes with constant angular velocity are shown in the following expressions.

$$\text{Linear:} \quad K_{\text{eff}} \;=\; K - \dot{\theta}^2 M \tag{3.58}$$

$$\text{Quadratic:} \quad K_{\text{eff}} \;=\; K - \dot{\theta}^2 (M + 2H) \tag{3.59}$$

Calculating the first natural frequency for a variety of rotation rates shows that the
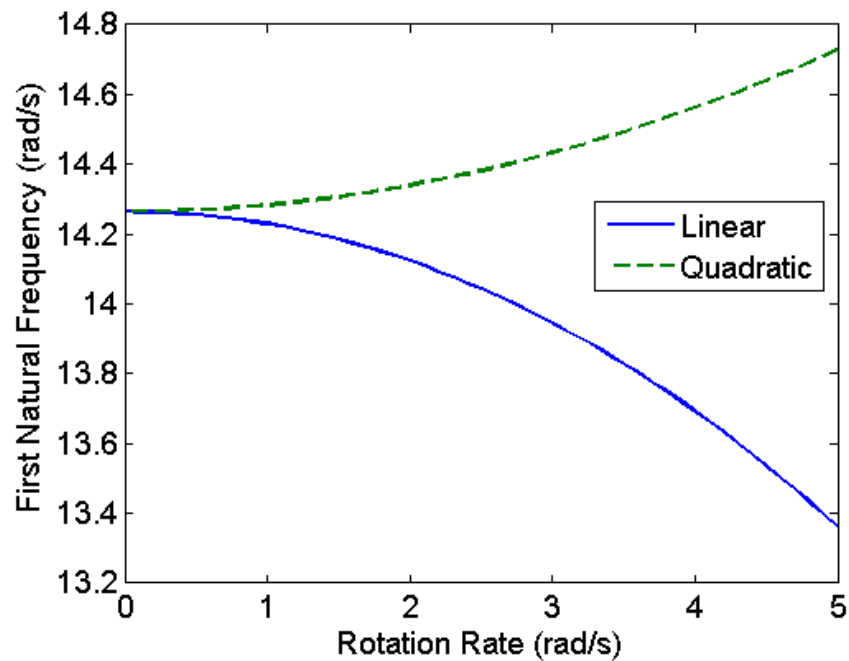
Fig. 14. First Natural Frequency versus Angular Velocity

quadratic assumed modes natural frequency increases, or stiffens, as expected. A comparison of the natural frequencies as a function of angular velocity can be seen in Figure 14. Higher natural frequencies exhibit this same trend.

Just as the quadratic and linear assumed modes developments produced different results, the quadratic and quadratic nonlinear assumed models also produce very different output. The nonlinear derivation is an extension of the quadratic modes derivation obtained by retaining the higher-order terms which were previously neglected, so it is interesting to examine the differences in the two sets of output given in Figure 15.
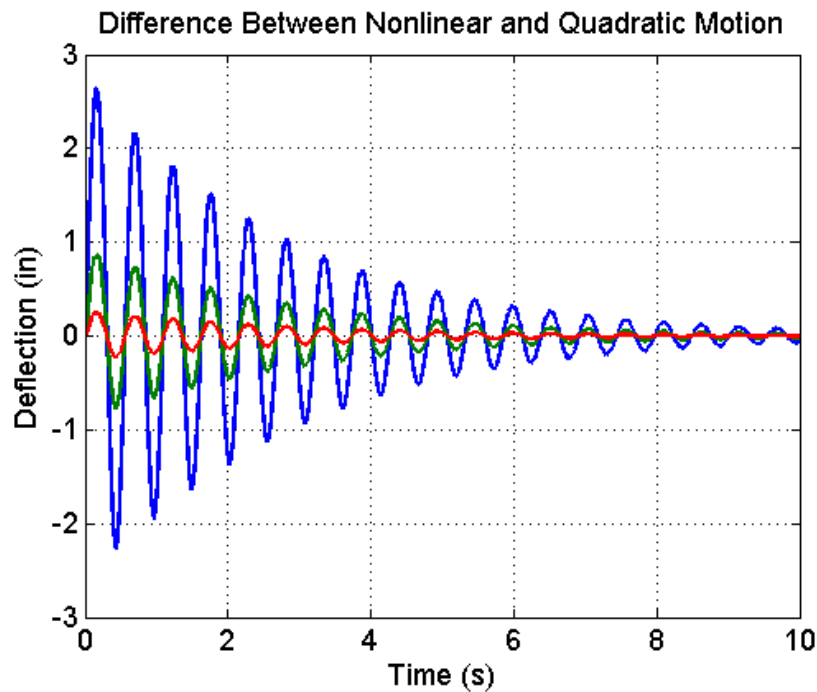
Fig. 15. Difference Between Nonlinear and Quadratic Motion

The difference between these two simulations is significant, and although this difference is sizable, the result should not discourage the use of the nonlinear set of equations. Figure 16 on the next page is a plot of one second of tip deflection calculated using the quadratic and nonlinear assumed modes. The plot reveals that there is a phase shift between the two sets of measurements. This could easily account for the large disparity between the two methods. The natural frequency evaluation
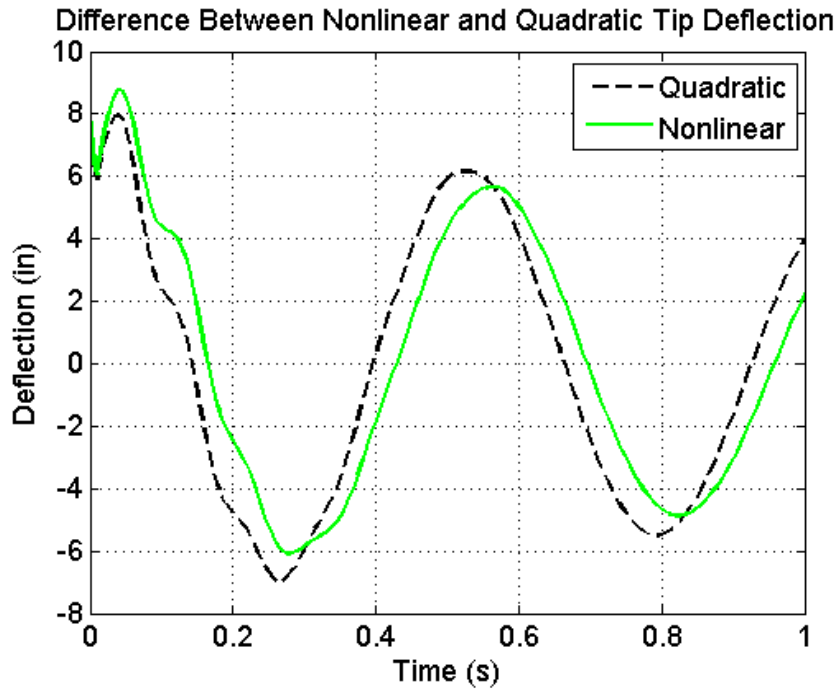
Fig. 16. Difference Between Nonlinear and Quadratic Motion, 1 sec

could not be performed for the nonlinear modes because they do not form a set of linear equations even for the case of constant angular velocity.

The simulations from this chapter revealed that the motion of a rotating space-craft with flexible appendage can be calculated using several variations of the Assumed Modes method. Linear, quadratic, and nonlinear assumed models each produce a different set of equations, and these equations yield results that are close enough to be believable but different enough to be significant. In the chapters that follow, input-output approximation techniques will be explored for simple systems as well as for the three rotational flexible systems developed in this chapter.

CHAPTER IV

LINEAR INPUT-OUTPUT APPROXIMATION

There are numerous occasions when identification of a system from input-output behavior is desirable or necessary. Identification of linear systems can be accomplished using a number of different techniques. The Eigen Realization Algorithm (ERA) is one parameter identification method that can provide the desired input-output mapping. This work will focus on the Eigen Realization Algorithm for because of its ability to provide a minimal realization and return only the states that participate in the measured motion. These advantages as well as the method itself will be developed further in the following sections.

A. Eigen Realization Algorithm

The Eigen Realization Algorithm is a linear input-output mapping that evaluates the sensor data time history to generate a state space model that could have produced the data. ERA is particularly useful in determining the order of a given system. For example, the motion of flexible structures is a function of an infinite number of modes, but it is not practical to use an infinite number of modes in modeling. ERA can determine the number of modes that are actually contributing to the measured motion. If the higher frequency modes cannot be measured by the sensors, then they do not need to be included in the model. The form of the identification is the standard state space model which is included below.

$$\begin{cases} \dot{\boldsymbol{x}} &= \mathrm{A}_L\boldsymbol{x} + \mathrm{B}_L\boldsymbol{u} \\ \boldsymbol{y} &= \mathrm{C}_L\boldsymbol{x} + \mathrm{D}_L\boldsymbol{u} \end{cases} \tag{4.1}$$

The subscript L signifies that the realization is linear, and the variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{u}$ represent the state, output, and input vectors respectively. These vectors are printed in boldface and should not be confused with x, y, and u which represented the longitudinal coordinate, the transverse displacement, and the longitudinal displacement in the previous chapters.

State space models such as this one are not unique. In fact, there are an infinite number of possible state space models for a given system, and all of the models are related to one another through linear transformations. With this in mind, it is not important whether ERA returns the true model that was used to generate the data. Rather, any state space model is acceptable as long as it is a linear transformation of the truth. For example, let the subscript t denote the true linear system as seen in the following equation.

$$\begin{cases} \dot{\boldsymbol{x}} &= A_t\boldsymbol{x} + B_t\boldsymbol{u} \\ \boldsymbol{y} &= C_t\boldsymbol{x} + D_t\boldsymbol{u} \end{cases} \tag{4.2}$$

A linear transformation of this system by the transformation matrix T where $\boldsymbol{z} = T\boldsymbol{x}$ would yield this equation.

$$\begin{cases} T\dot{\boldsymbol{x}} &= A_tT\boldsymbol{x} + B_t\boldsymbol{u} \\ \boldsymbol{y} &= C_tT\boldsymbol{x} + D_t\boldsymbol{u} \end{cases} \tag{4.3}$$

After rearranging, the new system becomes

$$\begin{cases} \dot{\boldsymbol{z}} &= A_z\boldsymbol{z} + B_z\boldsymbol{u} \\ \boldsymbol{y} &= C_z\boldsymbol{z} + D_z\boldsymbol{u} \end{cases} \tag{4.4}$$

where $A_z = T^{-1}A_tT$, $B_z = T^{-1}B_t$, $C_z = C_tT$, and $D_z = D_t$. It is well known that a linear transformation does not affect the frequency response of a system, so the eigenvalues of A_t and A_z will be the same [1].

The Eigen Realization Algorithm identifies the most likely state space system for the measurements given. The measurements should be the result of either an impulse input or an initial condition input. These are the only two conditions for which ERA is designed, but this is not as prohibitive as one might imagine. The initial condition response can be a time history of the measurements without control. The algorithm will be outlined here, but further details and explanations can be found in [4] and [11].

Let the measured output vector, denoted by $\boldsymbol{y}_{\text{meas}}$, contain a time history of the discrete time outputs. These outputs are combined as follows to form the Hankel matrix.

$$\boldsymbol{H}(1) = \begin{bmatrix} \boldsymbol{y}_{\text{meas}}(2) & \boldsymbol{y}_{\text{meas}}(3) & \cdots & \boldsymbol{y}_{\text{meas}}(d_1 + 1) \\ \boldsymbol{y}_{\text{meas}}(3) & \boldsymbol{y}_{\text{meas}}(4) & \cdots & \boldsymbol{y}_{\text{meas}}(d_1 + 2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{\text{meas}}(d_2 + 1) & \boldsymbol{y}_{\text{meas}}(d_2 + 2) & \cdots & \boldsymbol{y}_{\text{meas}}(d_1 + d_2 + 1) \end{bmatrix} \tag{4.5}$$

In this notation, $\boldsymbol{y}_{\text{meas}}(i)$ denotes the vector of output measurements at the ith discrete time. The constants d_1 and d_2 determine the size of the matrix, and if both are greater than the number of states, n, then the Hankel matrix will have rank n.

The next step in the Eigen Realization Algorithm is to decompose the Hankel matrix using Singular Value Decomposition (SVD). More information on SVD can be found in [12] and [13].

$$\boldsymbol{H} = \boldsymbol{U\Sigma V}^T \tag{4.6}$$

In this equation, the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal to each other, and $\boldsymbol{\Sigma}$ is

the diagonal matrix of singular values.

$$\mathbf{\Sigma} \ = \ \begin{bmatrix} \mathbf{\Sigma}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.7}$$

$$\mathbf{\Sigma}_n \ \equiv \ \mathrm{diag}(\sigma_1, \ \sigma_2, \ldots, \ \sigma_\mathrm{n}) \tag{4.8}$$

Only the n singular values that are above a certain threshold are included in the estimation. The rest are ignored or assumed to be approximately zero. The other matrices in the singular value decomposition are truncated accordingly to $\boldsymbol{U}_n$ and $\boldsymbol{V}_n$.

After this decomposition, the minimal realization is the following

$$\mathrm{A}_L \ = \ \mathbf{\Sigma}_n^{-1/2} \boldsymbol{V}_n^T \boldsymbol{H}(1) \boldsymbol{U}_n \mathbf{\Sigma}_n^{-1/2} \tag{4.9}$$

$$\mathrm{B}_L \text{ or } \mathrm{X}_{0\mathrm{L}} \ = \ \mathbf{\Sigma}_n^{1/2} \boldsymbol{V}_n^T \boldsymbol{E}_r \tag{4.10}$$

$$\mathrm{C}_L \ = \ \boldsymbol{E}_m^T \boldsymbol{U}_n \mathbf{\Sigma}_n^{1/2} \tag{4.11}$$

$$\mathrm{D}_L \ = \ \boldsymbol{y}_{\mathrm{meas}}(0) \tag{4.12}$$

$$\text{and } \boldsymbol{E}_\mathrm{j} \ \equiv \ [\ \mathbf{1}_j \ \ \mathbf{0}_j \ \ \cdots \ \ \mathbf{0}_j \ ]^T \tag{4.13}$$

where the subscript n is the number of states to be included in the realization, the subscript r is the number of inputs, and the subscript m is the number of outputs. The term $\mathbf{1}_j$ signifies an identity matrix of size $j \times j$, and $\mathbf{0}_j$ signifies a matrix of size $j \times j$ fully populated with zeros. Equation 4.10 yields the state space B_L matrix if the measured values were excited by an impulse input, and it yields the initial condition vector $\mathrm{X}_{0\mathrm{L}}$ if the measured values were the consequence of an initial condition response. This state space representation is assumed to capture the modal frequencies of interest.

B. Spring-Mass-Damper Example

A simple problem to which ERA can easily be applied is the spring-mass-damper problem. This is a classic problem where a block of mass m is translating in one degree of freedom. The mass is attached to a wall by a spring with spring constant k and by a damper with damping coefficient c. The motion of the block is governed by the second-order differential equation shown below, and there is no forcing function on the system.

$$m\ddot{x} + c\dot{x} + kx = 0 \tag{4.14}$$

If the output is the position and velocity of the block, then the state space model for this system is simply the following.

$$
\begin{cases}
\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \\[2em]
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}
\end{cases}
\tag{4.15}
$$

It is obvious that this is a linear system, so the Eigen Realization Algorithm will have no problems identifying this system. The system is simulated with the following constants: $m = 1$, $c = 0.25$, and $k = 1$. Figure 17 on the next page shows the output of this system where y_1 is the position and y_2 is the velocity of the block. From this figure, one can see that the system was simulated with a non-zero initial condition vector as opposed to an impulse input.

Taking this measurement time history and applying the Eigen Realization Algorithm yields the following state space realization. It is important to note that ERA outputs the system in discrete time, so the matrices must be converted into a continu-
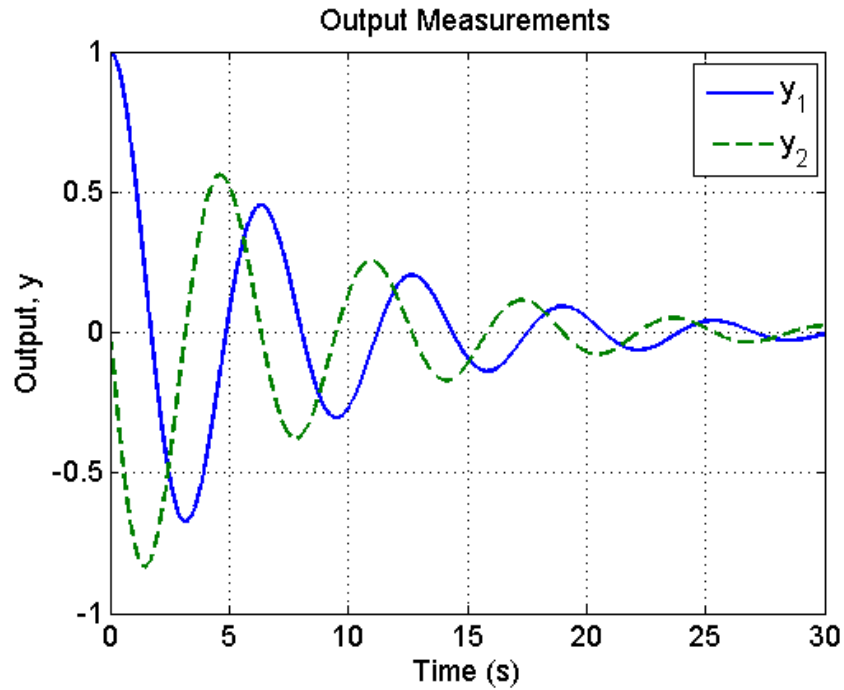
Fig. 17. Spring-Mass-Damper Output Measurements

ous time state space representation. This can be easily accomplished using a built-in function like `d2c` in Matlab for example.

$$
\begin{cases}
\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} -0.0093 & 0.3343 \\ -2.9842 & -0.2399 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, & X_{0L} = \begin{bmatrix} -0.5991 \\ -0.3293 \end{bmatrix} \\
\\
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -1.6194 & -0.0907 \\ 0.2856 & -0.5197 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}
\end{cases}
\tag{4.16}
$$

There is no control input to the system which is why ERA cannot estimate B_L and D_L.

A plot of the error between the measurements and the ERA-identified system is shown in Figure 18 on the following page. This plot reveals that the Eigen Realization Algorithm is a very accurate approximation of the system. Comparing the eigenvalues
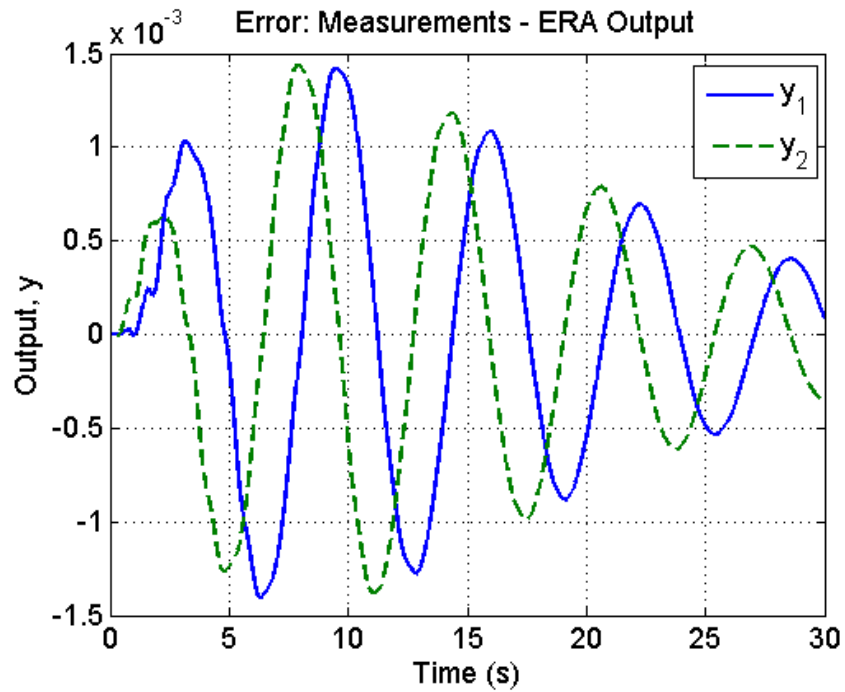
Fig. 18. Error: Measurement - ERA Simulation Output

Table IV. Comparison of Truth and Estimated A Matrices

	Eigenvalues	Frequency (rad/s)	Damping
Estimated System	$-0.1246 \pm 0.9922i$	1.00	0.125
True System	$-0.1250 \pm 0.9922i$	1.00	0.125

of the true versus the estimated state space A matrices shows close correlation between the two. Table IV shows that the ERA representation accurately estimates the system characteristics.

1. Duffing Oscillator

ERA was shown to be successful on the simple spring-mass-damper example shown above. However, that was a linear system. The duffing oscillator is also a spring-

mass-damper system except the spring has linear and nonlinear components. This section will show that applying ERA to this system will result in significant modeling errors. The equation of motion for this system involves a minor modification from that of the spring-mass-damper.

$$m\ddot{x} + c\dot{x} + kx + \varepsilon x^3 = 0 \tag{4.17}$$

With the same coefficients as in the previous example, and letting $\varepsilon = 3$, this system exhibits visibly different motion as seen in Figure 19.
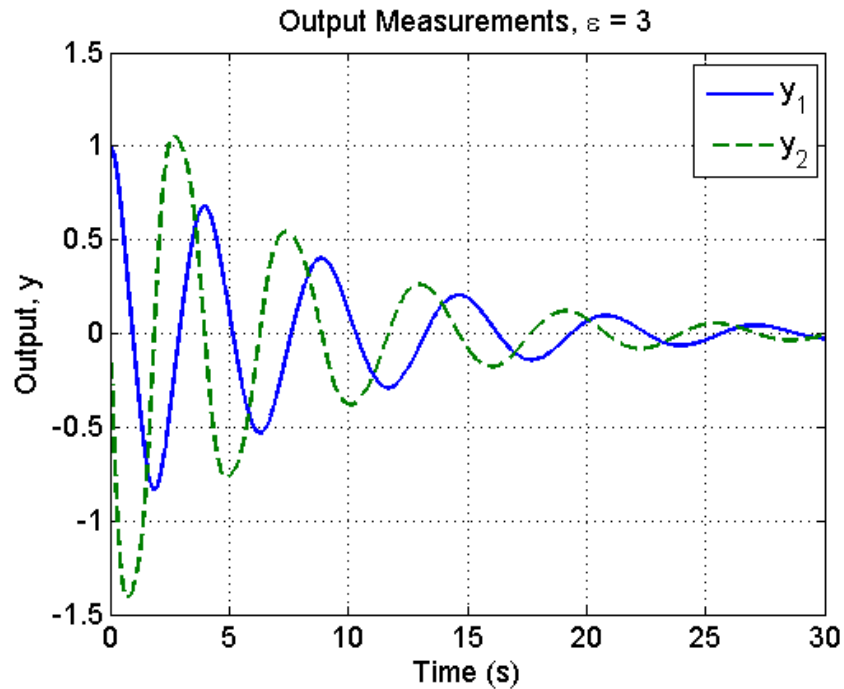
Fig. 19. Measured Values for Duffing Oscillator

The Eigen Realization Algorithm is applied to this output. For moderate to large nonlinearities, ERA may try to realize the system with an inflated number of states to compensate for the nonlinearity. With prior knowledge of the system kinematics and dynamics, the ERA algorithm applied to this problem is instructed to return a

system with exactly two states. The resulting system, transformed into continuous time, is the following state space realization.

$$
\begin{cases}
\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 1.1050 & 1.4724 \\ -4.5862 & -2.2793 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, & X_{0L} = \begin{bmatrix} -0.4661 \\ -0.63030 \end{bmatrix} \\[3em]
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -1.3890 & -0.5593 \\ 1.0304 & -0.7667 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}
\end{cases}
\tag{4.18}
$$

This realization is decidedly different from the one obtained in the previous example. Indeed, the nonlinearity cannot be captured by the linear state space model.
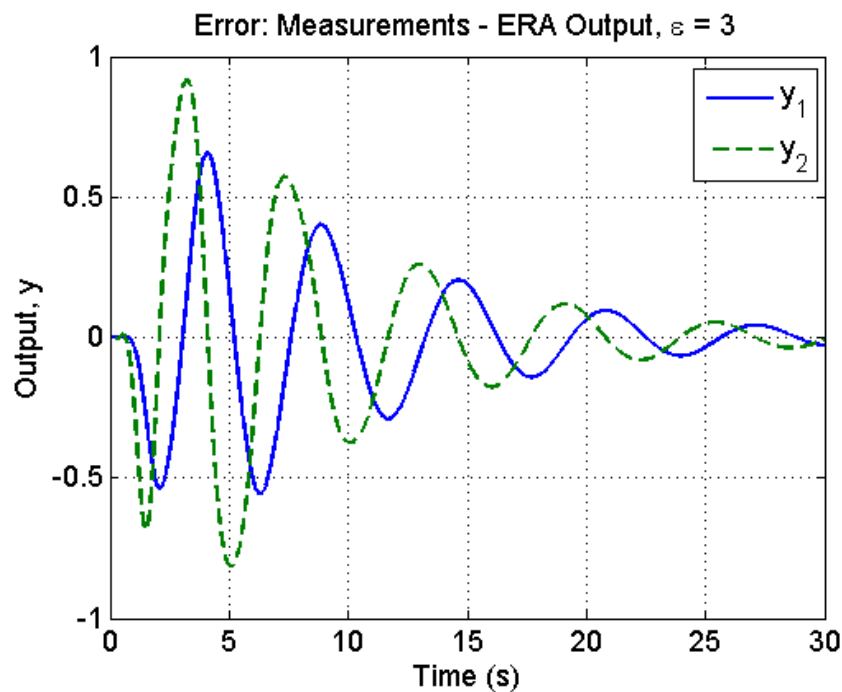
Fig. 20. Measurements versus ERA Output, $\varepsilon = 3$

A comparison of the ERA-identified model output and the true duffing oscillator measurements can be seen in Figure 20. Comparing this figure with Figure 18 on page 42 shows that the errors are 1000 times larger for the system with the nonlinear

spring. The percent error for the duffing oscillator ERA model output is approximately 100%. Moreover, these errors resulted from an estimation based on perfect measurements. Noisy data would have decreased the accuracy of the estimate even further. From these observations, it is obvious that this linear approximation is unacceptable for the nonlinear system.

C.   Rotating Spacecraft Application

Another linear system to which ERA can be applied is the case of the rotating spacecraft with flexible appendage. The equations of motion for this system were developed for three different assumed mode types in Chapter III. In that chapter, it was noted that for a known angular profile, the equations of motion for the linear and quadratic assumed mode shapes were linear. The same was true for a constant angular velocity. On the other hand, the nonlinear assumed modes model produced nonlinear equations whether the angular profile was known or unknown. Therefore, the Eigen Realization Algorithm should be able to accurately model the systems based on output from the linear and quadratic assumed mode equations with fixed angular velocity.

1.   Linear Assumed Model with Fixed Rotation Rate

Equations of motion for the spacecraft calculated using the linear assumed modes was given in Equation 3.15 on page 18 and is repeated below for convenience.

$$M\ddot{q} + C\dot{q} + \left(K - M\dot{\theta}^2\right)q = 0$$

This set of equations was simulated using the same spacecraft constants seen in the previous chapter. The beam was initially deflected 1.5 in, 3 in, and 6 in at the quarter-span, midspan, and tip. In addition, the rotation rate was set at 1.5 rad/s. Figure 21

shows the results of this simulation using three modes and Admissible functions. The deflections at the quarter-span, the midspan, and the tip are plotted on this figure.
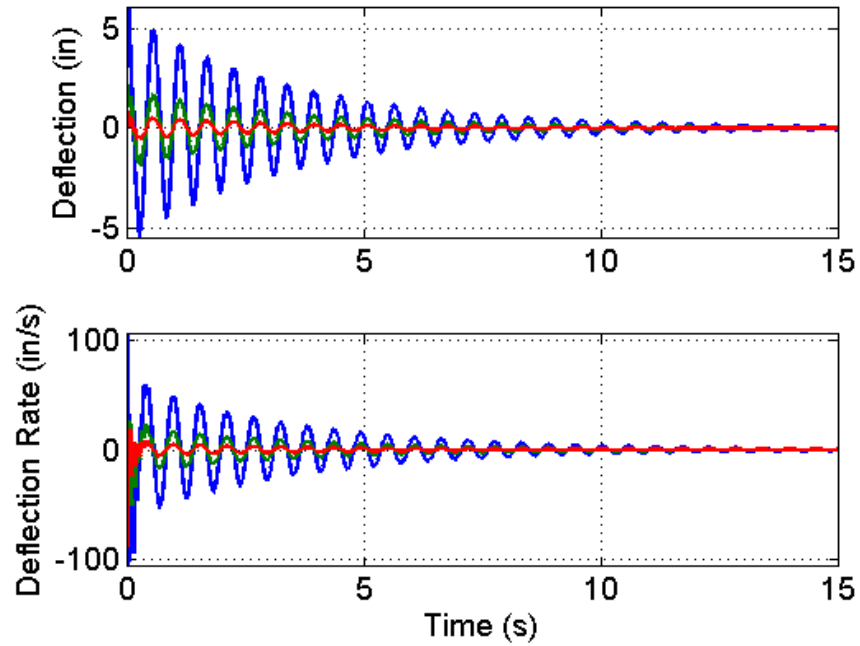
Fig. 21. Rotational Spacecraft Beam Measurements, Linear Modes

Measurements from this system were inputs for the Eigen Realization Algorithm, and the linear state space realization converted to continuous time is given in the following equations.

$$
\dot{x} = \begin{bmatrix}
0.651 & 3.331 & -18.86 & -123.7 & -57.52 & -30.59 \\
4.305 & 8.044 & 90.25 & -119.1 & 3.409 & -28.54 \\
15.98 & -32.28 & 46.9 & -296.8 & -96.51 & -11.39 \\
13.32 & 97.34 & 233 & -511.8 & -374 & -113.5 \\
2.094 & -41.28 & -54.25 & 363.7 & 84.36 & 11.11 \\
1.685 & 2.444 & -6.438 & 27.25 & 20.67 & 3.185
\end{bmatrix} x, \quad (4.19)
$$

$$
X_{0L} = \begin{bmatrix} -0.7885 & -5.854 & -8.287 & 12.82 & 4.84 & 1.289 \end{bmatrix}^T
$$

$$\boldsymbol{y} = \begin{bmatrix} -0.004384 & -0.08377 & -0.01627 & 0.03503 & -0.01325 & 0.3773 \\ -0.03186 & -0.1876 & 0.04637 & 0.06282 & -0.03814 & 1.273 \\ -0.1755 & -0.1918 & 0.4062 & 0.1794 & 0.2119 & 3.709 \\ 0.4511 & 4.716 & -1.789 & 2.889 & -4.884 & -0.2098 \\ 2.814 & 7.688 & -5.643 & 1.197 & -3.69 & 2.303 \\ 14.64 & 2.538 & 11.48 & 10.35 & -2.334 & 0.03119 \end{bmatrix} \boldsymbol{x} \quad (4.20)$$
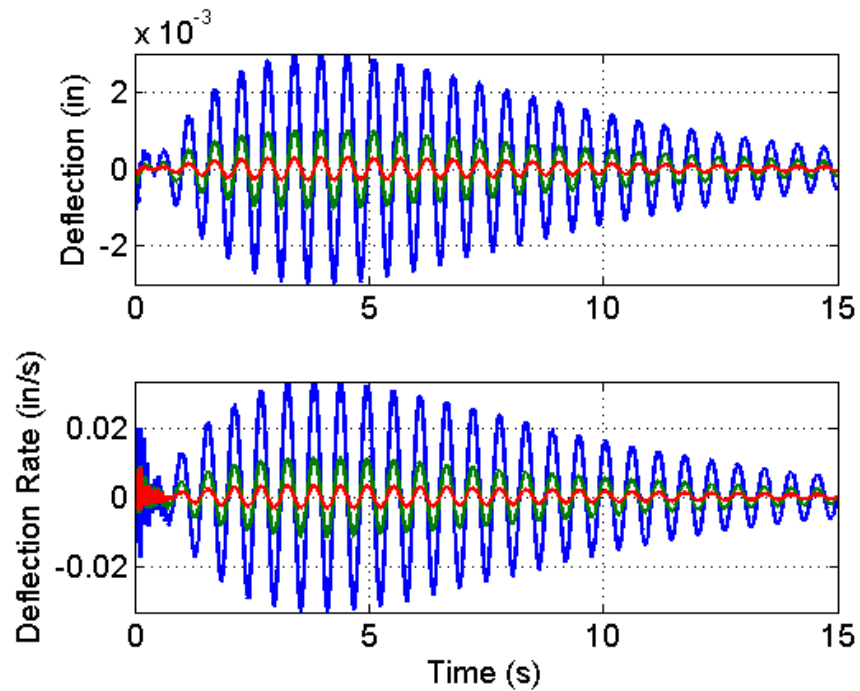
Fig. 22. Error: Measurements - ERA Output, Linear Modes

A comparison of the system characteristics reveals that the ERA-identified system closely matches the true system. Figure 22 shows the agreement between the two models.

2.   Quadratic Assumed Model with Fixed Rotation Rate

The same rotational system motion calculated using quadratic assumed modes also yields a linear system. The equations of motion were developed in Chapter III, and the linear equation for constant angular velocity is repeated below for reference.

$$M\ddot{q} + C\dot{q} + \left(K - \dot{\theta}^2(M + 2H)\right)q = 0$$

Simulation results for this set of equations are shown in Figure 23. The spacecraft constants for this simulation were given in the previous chapter.
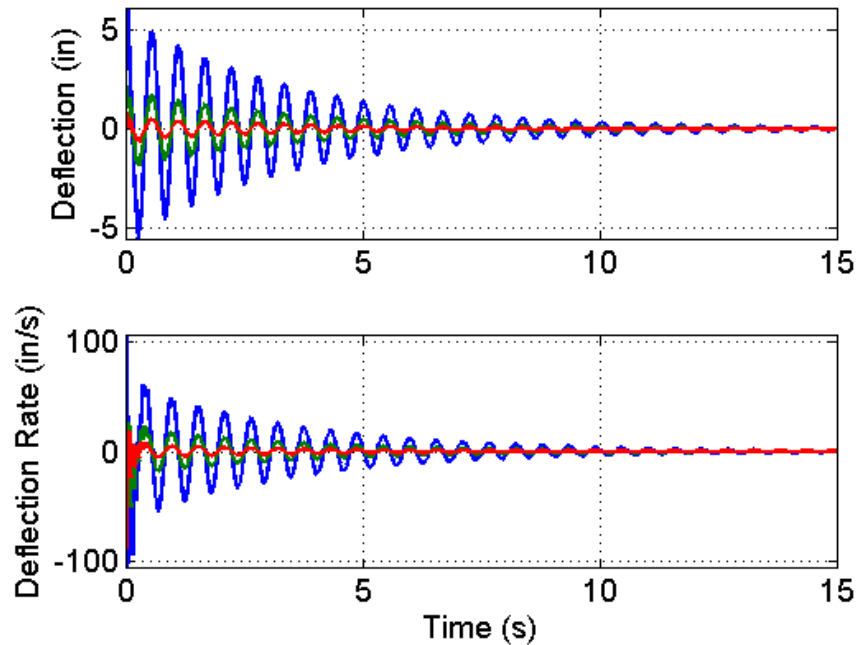
Fig. 23. Rotational Spacecraft Beam Measurements, Quadratic Modes

When this data is entered into ERA, the resulting state space realization is the

following.

$$\dot{x} = \begin{bmatrix} 0.6313 & 3.36 & -18.93 & -123.5 & -57.11 & -31.64 \\ 4.247 & 7.969 & 90.17 & -118.5 & 3.825 & -29.11 \\ 15.89 & -32.45 & 46.76 & -296.9 & -96.65 & -12.34 \\ 13.3 & 97.11 & 232.1 & -511.9 & -373.4 & -118.6 \\ 2.018 & -41.29 & -53.9 & 363.7 & 84.24 & 11.87 \\ 1.673 & 2.129 & -7.19 & 31.64 & 21.93 & 3.665 \end{bmatrix} x, \quad (4.21)$$

$$X_{0L} = \begin{bmatrix} -0.802 & -5.848 & -8.269 & 12.83 & 4.833 & 1.346 \end{bmatrix}^T$$

$$y = \begin{bmatrix} -0.004313 & -0.08342 & -0.01511 & 0.03533 & -0.0162 & 0.3779 \\ -0.03133 & -0.1864 & 0.04995 & 0.06374 & -0.04839 & 1.273 \\ -0.1728 & -0.1883 & 0.4156 & 0.1817 & 0.1809 & 3.708 \\ 0.4724 & 4.715 & -1.79 & 2.891 & -4.883 & -0.2452 \\ 2.862 & 7.685 & -5.631 & 1.204 & -3.707 & 2.336 \\ 14.68 & 2.503 & 11.5 & 10.35 & -2.337 & 0.0114 \end{bmatrix} x \quad (4.22)$$

The estimated-system characteristics match closely with the true characteristics. This agreement can be seen in a plot of the error between the output of the ERA-identified model and the true measurements in Figure 24 on the next page.
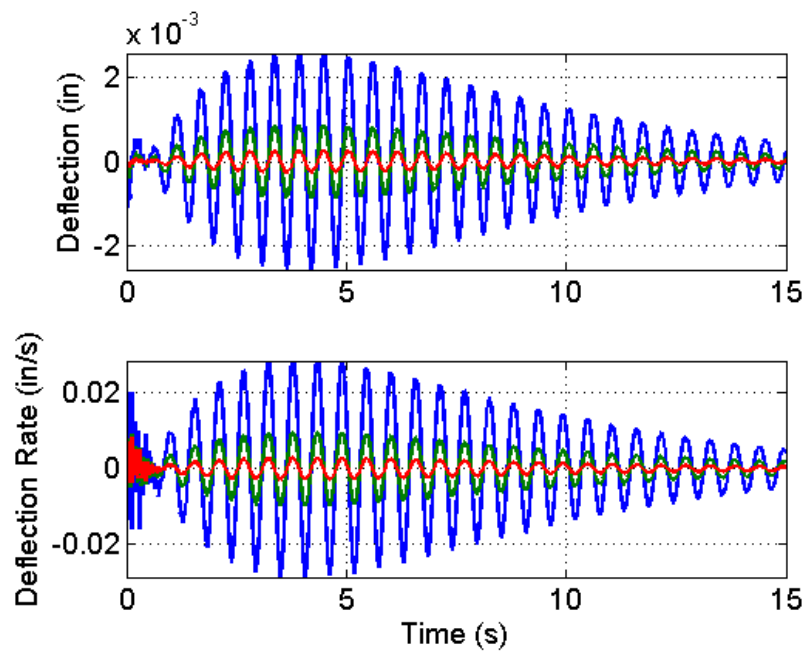
Fig. 24. Error: Measurements - ERA Output, Quadratic Modes

identified model.

$$
\dot{\boldsymbol{x}} \;=\;
\begin{bmatrix}
0.6217 & 2.898 & -18.18 & -123.2 & -57.05 & -30.95 \\
4.497 & 8.201 & 90.49 & -120.4 & 4.142 & -28.97 \\
15.82 & -32.16 & 47.28 & -296 & -94.18 & -10.85 \\
13.37 & 97.69 & 236 & -505.8 & -368.5 & -114 \\
2.25 & -40.79 & -53.74 & 355.3 & 80.97 & 10.81 \\
1.694 & 2.612 & -6.393 & 25.85 & 20.04 & 3.044
\end{bmatrix}
\boldsymbol{x}, \quad (4.23)
$$

$$
X_{0L} \;=\; \begin{bmatrix} -0.7757 & -5.883 & -8.396 & 12.85 & 4.823 & 1.302 \end{bmatrix}^{T}
$$

$$
\boldsymbol{y} =
\begin{bmatrix}
-0.004535 & -0.08377 & -0.01591 & 0.03457 & -0.01346 & 0.3771 \\
-0.03235 & -0.1871 & 0.04778 & 0.06202 & -0.03653 & 1.271 \\
-0.1769 & -0.1894 & 0.4081 & 0.1774 & 0.2239 & 3.702 \\
0.4409 & 4.731 & -1.763 & 2.898 & -4.899 & -0.2063 \\
2.797 & 7.684 & -5.646 & 1.17 & -3.757 & 2.333 \\
14.65 & 2.636 & 11.47 & 10.43 & -2.275 & 0.05039
\end{bmatrix}
\boldsymbol{x} \quad (4.24)
$$

Figure 26 on the next page shows that the error between the measurements and the estimated output are three orders of magnitude larger than for the quadratic or linear modes estimations. Although the errors are still relatively small compared to the motion, it is important to note that all of the identification in this work is performed on perfect measurements. Noisy measurements would significantly degrade the estimates at which point a linear estimation of the nonlinear system would be inadequate. The equations of motion for the rotational spacecraft with unknown rotation rate are nonlinear for all three sets of assumed mode shapes, so ERA will not provide an adequate approximation for these systems.

From these examples, it is obvious that the Eigen Realization Algorithm is a
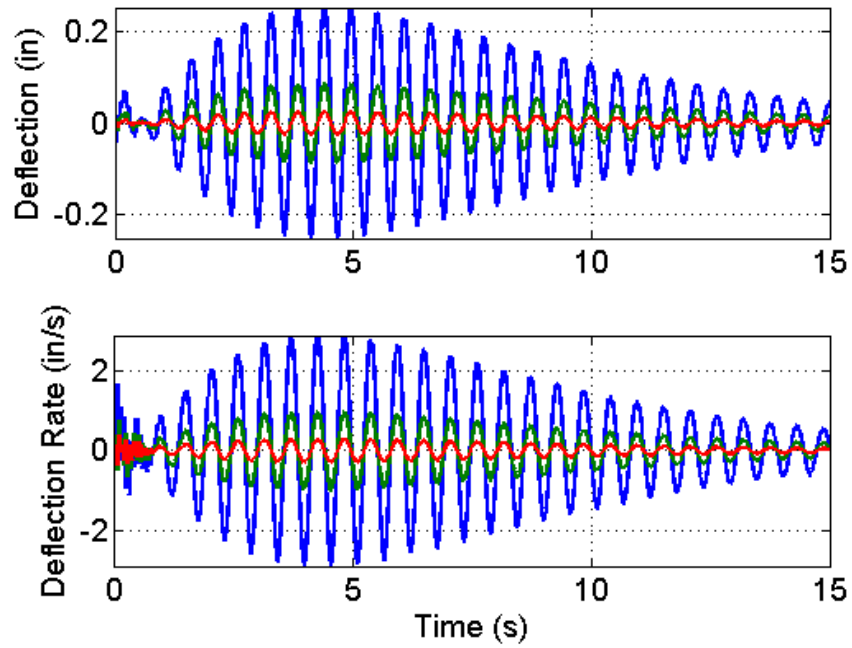
Fig. 26. Error: Measurements - ERA Output, Nonlinear Modes

suitable linear parameter identification method. ERA can determine an appropriate order for the system based on the output measurements. It can also match the system parameters with surprising accuracy. This method will be employed in future chapters as the chosen linear approximation method.

CHAPTER V

NONLINEAR INPUT-OUTPUT APPROXIMATION

The linear approximation technique outlined in the previous chapter is useful on a wide class of systems, but it is only valid for linear systems. Many systems are nonlinear, so ERA and similar parameter identification algorithms will not accurately approximate the motion. Nonlinear systems range in complexity from the simple pendulum undergoing moderate or large deflections, to the rotating spacecraft problem described earlier, and to systems that are even more complex. For each of these systems, input-output approximation is complicated but often necessary. Singla, et al, have developed a nonlinear input-output mapping technique [7] [14]. This method involves combining linear estimation techniques with a nonlinear function estimator. A rigorous description of this method is the focus of this chapter.

Nonlinear Input-Output Approximation Goal: Given the output measurements, $\boldsymbol{y}_{\mathrm{meas}}$, and the measured control input, $\boldsymbol{u}_{\mathrm{meas}}$, as functions of time, find an accurate representation of the system that generates these measurements.

A.   Algorithm Overview

The goal of this method is to identify a linear model that captures the linear motion of the system and then determine an estimate for a nonlinear function that captures the difference between the linear motion and the measured values. The estimated linear model is a linear state space representation as seen in the following set of equations

$$\begin{cases} \dot{\boldsymbol{x}}_L &= \mathrm{A}_L \boldsymbol{x}_L + \mathrm{B}_L \boldsymbol{u} \\ \boldsymbol{y}_L &= \mathrm{C_L} \boldsymbol{x}_L + \mathrm{D}_L \boldsymbol{u} \end{cases} \qquad (5.1)$$

where the subscript L identifies the best linear model of the system. This model may be obtained by any number of linear identification algorithms, but the Eigen Realization Algorithm (ERA) will be used in this work for reasons expressed in the previous chapter.

Once the state space representation has been found, the output equation can be combined with the input measurements, $\boldsymbol{y}_{\mathrm{meas}}$, and the control input, $\boldsymbol{u}_{\mathrm{meas}}$, to obtain an estimate for the hidden state vector $\boldsymbol{x}$. The output equation with the measurements is simply the following.

$$\boldsymbol{y}_{\mathrm{meas}} = \mathrm{C}_L \boldsymbol{x} + \mathrm{D}_L \boldsymbol{u}_{\mathrm{meas}} \tag{5.2}$$

Because the only unknown in this equation is $\boldsymbol{x}$, a linear least squares routine will reveal the best guess for this hidden state vector.

$$\hat{\boldsymbol{x}} = \left(\mathrm{C}_L^{\mathrm{T}} \mathrm{C}_L \right)^{-1} \mathrm{C}_L^{\mathrm{T}} \left(\boldsymbol{y}_{\mathrm{meas}} - \mathrm{D}_L \boldsymbol{u}_{\mathrm{meas}} \right) \tag{5.3}$$

For nonlinear systems, simulating the ERA-identified linear state equation, $\dot{\boldsymbol{x}}_L = \mathrm{A}_L \boldsymbol{x}_L + \mathrm{B}_L \boldsymbol{u}_{\mathrm{meas}}$, will produce a state vector that will not match the least squares estimate. From this and the previous equations, one can see that if the nonlinearly estimated state vector can track $\hat{\boldsymbol{x}}$, then the estimated output will equal $\boldsymbol{y}_{\mathrm{meas}}$.

To match the hidden state vector and the true output, the estimated system has to capture the nonlinear motion as well as the linear motion. The goal of the nonlinear function estimator is to estimate $g(\boldsymbol{x}_g)$ such that the estimated state vector, $\boldsymbol{x}_g$ given by

$$\dot{\boldsymbol{x}}_g = \mathrm{A}_L \boldsymbol{x}_g + \mathrm{B}_L \boldsymbol{u}_{\mathrm{meas}} + g(\boldsymbol{x}_g) \tag{5.4}$$

tracks the best estimate of the hidden state vector $\hat{\boldsymbol{x}}$. The nonlinear function estimator compensates for the difference between the estimated hidden state vector and

the state vector calculated using the linear model. Overall, the identified model is of the form

$$\begin{cases} \dot{\boldsymbol{x}}_g & = & \mathrm{A}_L\boldsymbol{x}_g + \mathrm{B}_L\boldsymbol{u}_{\mathrm{meas}} + g(\boldsymbol{x}_g) \\ \boldsymbol{y}_g & = & \mathrm{C}_L\boldsymbol{x}_g + \mathrm{D}_L\boldsymbol{u}_{\mathrm{meas}} \end{cases} \tag{5.5}$$

where A_L, B_L, C_L, and D_L are the linear state space matrices, and $g(\boldsymbol{x})$ is the nonlinear function to modify the linear motion.

Duffing Oscillator Example: In Chapter IV, it was shown that the linear estimation from the Eigen Realization Algorithm could not capture the nonlinear motion of a duffing oscillator. The nonlinear input-output approximation algorithm from this chapter will therefore be applied to this simple problem to aid in the explanation of this technique. Although the problem was introduced in the previous chapter, it will be reviewed here for convenience. The equation of motion for this system is rewritten below.

$$m\ddot{x} + c\dot{x} + kx + \varepsilon x^3 = 0 \tag{5.6}$$

As in the previous chapter, the measured output will be the position and velocity of the moving block.

The duffing oscillator will be simulated with no external control, so the equation of motion can be rewritten in state space form.

$$\begin{cases} \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} & = & \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \epsilon x^3 \end{bmatrix} \\ \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} & = & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \end{cases} \tag{5.7}$$

For this example, the numerical values for the constants are as follows: $m = 1$,
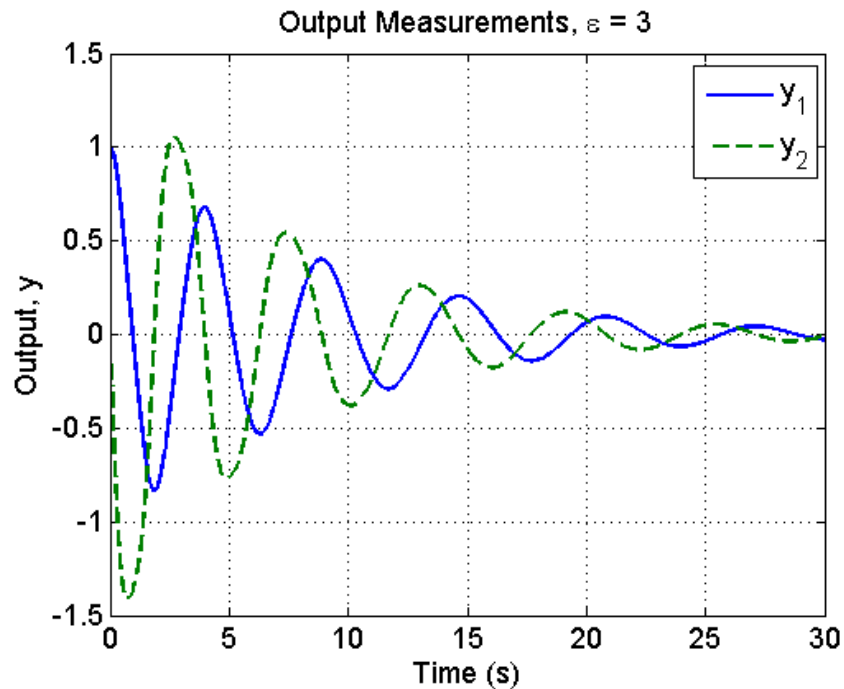
Fig. 27. Measured Output of Duffing Oscillator

$c = 0.25$, and $k = 1$. This system is simulated with no control and initial conditions $x_1 = 1$ and $x_2 = 0$ which corresponds to an initial deflection with no initial velocity. The resulting motion can be seen in Figure 27.

B. Linear Input-Output Approximation

As previously mentioned, this method begins by obtaining a linear function estimate. The Eigen Realization Algorithm is a preferred method, but any linear identification technique could be used in its place. This algorithm was covered in depth in the previous chapter, so the reader is referred there for more information.

The goal of ERA is to identify the most likely linear state space realization from a time history of system outputs and inputs. Only the state space matrices are estimated. The state vector is not estimated, but it can be easily calculated from the

identified system. This system takes the form

$$\dot{\boldsymbol{x}}_L \;=\; \mathrm{A}_L\boldsymbol{x}_L + \mathrm{B}_L\boldsymbol{u}_{\text{meas}} \tag{5.8}$$

$$\boldsymbol{y}_L \;=\; \mathrm{C}_L\boldsymbol{x}_L + \mathrm{D}_L\boldsymbol{u}_{\text{meas}} \tag{5.9}$$

where the output vector $\boldsymbol{y}_L$ is not guaranteed to be equal to the measured output vector $\boldsymbol{y}_{\text{meas}}$. This potential problem arises because the algorithm is only guaranteed for linear systems.

Application to a Duffing Oscillator: In the case of the duffing oscillator, the system is not linear, so the ERA-identified realization cannot reproduce the measured output. The realization for this system was given in the previous chapter, but it is included again here for easy reference.

$$\left\{ \begin{array}{l} \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 1.1050 & 1.4724 \\ -4.5862 & -2.2793 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad x_{0L} = \begin{bmatrix} -0.4661 \\ -0.63030 \end{bmatrix} \\[2em] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -1.3890 & -0.5593 \\ 1.0304 & -0.7667 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \end{array} \right.$$

As described previously, simulating this ERA-estimated linear system produces linear output estimates that do not match the true measurements. This is due to the nonlinearity in the true system. The discrepancy can be seen in Figure 28 on the next page. This difference comes from the fact that the ERA-identified model can only capture the linear motion. To capture the nonlinear motion as well, a nonlinear function estimation algorithm like the one described in this chapter must be used.
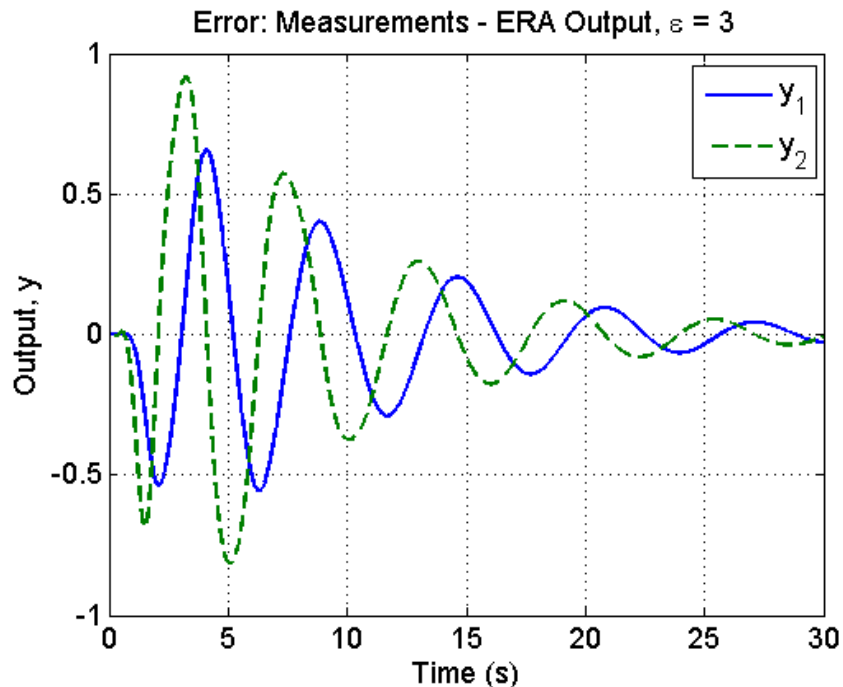
Fig. 28. Measurements vs. ERA-Estimated Output, $\varepsilon = 3$

C. Least Squares Estimation

The output equation for the linear system, Equation 5.9 on the preceding page, is a linear equation. If the output vector, $\boldsymbol{y}_{\mathrm{meas}}$, and the input vector, $\boldsymbol{u}_{\mathrm{meas}}$, are known, and if the linear state space realization is also known, then the only unknown in this equation is the hidden state vector. It is referred to as the hidden state vector because the output and input of the system are known, but the true state vector is not.

With an estimate for the linear system, the hidden state vector can be estimated using a linear least squares algorithm. The measurements and the estimated linear state space matrices are combined as follows to obtain the estimate $\hat{\boldsymbol{x}}$.

$$\hat{\boldsymbol{x}} = (\mathrm{C}_L^{\mathrm{T}}\mathrm{C}_L)^{-1}\mathrm{C}_L^{\mathrm{T}}(\boldsymbol{y}_{\mathrm{meas}} - \mathrm{D}_L\boldsymbol{u}_{\mathrm{meas}}) \tag{5.10}$$

The least squares algorithm returns the best solution for the unknown parameter [15].

Therefore, $\hat{\boldsymbol{x}}$ is the best estimate of the hidden state vector. It is important to note that the measurements must be a function of all the states of the system. Otherwise, the product $(\mathrm{C}_L^{\mathrm{T}}\mathrm{C}_L)$ will become singular. This will result in poor estimates for the hidden state vector, and in turn, poor estimates for the entire nonlinear input-output approximation process.

Once this estimate exists, the final step in the process is to estimate the nonlinear functions that are required to have the calculated state vector, $\boldsymbol{x}_g$, match the estimated state vector, $\hat{\boldsymbol{x}}$.

Application to a Duffing Oscillator: The least squares estimation is straightforward to apply to the duffing oscillator. There are only two states, so the inverse is quite easy. Similarly, there is no control, so the equation is simply the following.

$$\hat{\boldsymbol{x}} = (\mathrm{C}_L^{\mathrm{T}}\mathrm{C}_L)^{-1}\mathrm{C}_L^{\mathrm{T}}\boldsymbol{y}_{\mathrm{meas}} \tag{5.11}$$

Evaluating this equation with the output measurements and the ERA-identified C_L matrix yields an estimated state vector that is significantly different than the state vector obtained by simulating the linear model. The magnitude of these differences can be seen in Figure 29 on the next page. From this figure, one can see that $\boldsymbol{x}_L$ follows $\hat{\boldsymbol{x}}$ closely for about two seconds. After this point, the ERA-estimated linear state vector motion damps out significantly faster than the least squares estimated state vector. The nonlinear function estimator will need to capture this nonlinear motion so that the calculated state vector can match the LSE-estimated state vector.
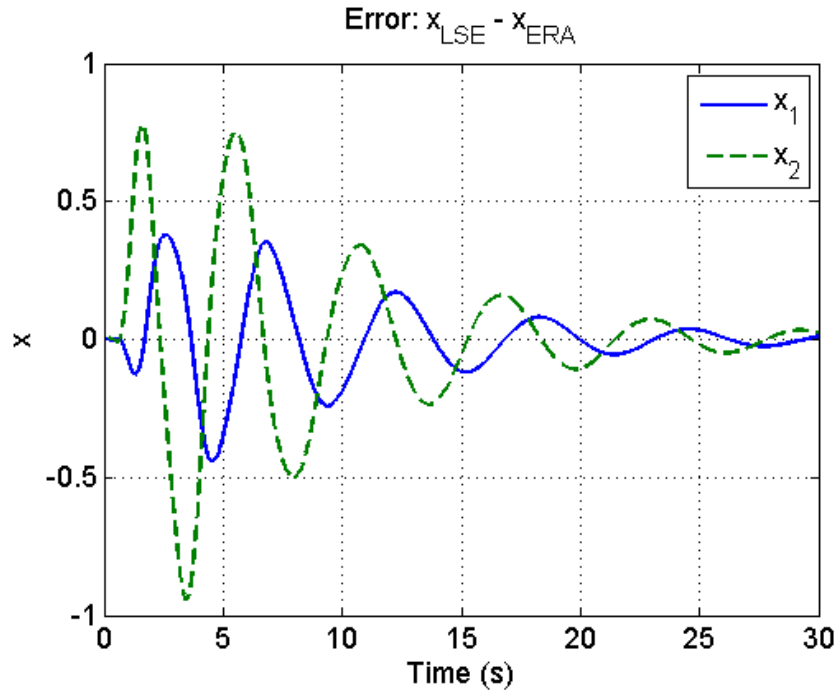
Fig. 29. Error: LSE Estimated $\hat{\boldsymbol{x}}$ - ERA Estimated $\boldsymbol{x}_L$

D.   Nonlinear Function Estimation

In the parameter identification process presented here, the nonlinear function estimator should estimate $g(\boldsymbol{x}_g)$ such that the output of the following system

$$\begin{cases} \dot{\boldsymbol{x}}_g & = \ \text{A}_L \boldsymbol{x}_g + \text{B}_L \boldsymbol{u}_{\text{meas}} + g(\boldsymbol{x}_g) \\ \boldsymbol{y}_g & = \ \text{C}_L \boldsymbol{x}_g + \text{D}_L \boldsymbol{u}_{\text{meas}} \end{cases} \tag{5.12}$$

tracks the measured output $\boldsymbol{y}_{\text{meas}}$. It has already been shown that if $\boldsymbol{x}_g$ can track $\hat{\boldsymbol{x}}$ then $\boldsymbol{y}_g$ will track $\boldsymbol{y}_{\text{meas}}$. This comes straight from the least squares estimate.

Assume that the nonlinear function can be matched arbitrarily closely by a linear combination of polynomial functions. Using an infinite number of polynomial functions would provide an exact match but is computationally prohibitive. On the other hand, too few polynomials make the estimation simple but will result in poor

matching. One solution is to use a finite number of *orthogonal* polynomials to better match the truth while keeping the number of functions required to a minimum. Unique properties of orthogonal polynomials allow them to capture a wider array of motion with a smaller number of functions.

This reduction can be taken one step further. Rather than estimating the nonlinear function with a set of polynomials valid over the entire domain, or global polynomials, one can use a smaller set of polynomials to fit the function in local regions and then combine these local estimates to form the global solution. This type of estimation scheme is the Global-Local Orthogonal Mapping algorithm (GLO-MAP) developed by Junkins et al [6]. GLO-MAP is an innovative algorithm that can perform nonlinear function estimation in N-dimensions. The method is more translucent than an Artificial Neural Networks (ANN), which typically operate as a black box algorithm [7].

1. GLO-MAP Explanation

The GLO-MAP estimation routine operates on the principle that any function can easily be approximated on a small interval by a finite number of orthogonal polynomials. Therefore, to approximate a function on a large interval, one can split the domain into a series of smaller intervals over which the function approximation is straightforward. Then, combine these local approximations to obtain the global estimate.

The basic GLO-MAP algorithm is implemented according to the following process. To approximate a function $\mathcal{L}(\mathcal{X})$:

1. Obtain function measurements in global coordinates $\mathcal{X}$.

2. Discretize the domain and choose grid points accordingly.

3. For the Jth grid point, $^J\boldsymbol{X}$:

   (a) Nondimensionalize the global coordinates using the following equation for each measurement location.

$$^J\boldsymbol{\xi} \equiv (\boldsymbol{\mathcal{X}} - {}^J\boldsymbol{X})/h \tag{5.13}$$

   (b) Find all $^J\boldsymbol{X}$ for which each state of $^J\boldsymbol{\xi}$ lies within the interval $[-1,\ 1]$.

   (c) Calculate the weight function value for each local coordinate

$$w(\boldsymbol{\xi}) = \Pi_{i=1}^n 1 - \xi_i^2 (3 - 2|\xi_i|) \tag{5.14}$$

   where n is the number of states.

   (d) Calculate the set of basis function values $\boldsymbol{\Phi}(\boldsymbol{\xi})$ for each local coordinate. The total basis function matrix is the set of all possible combinations of these functions.

   (e) Find the Fourier coefficient for each basis function and each local coordinate. These are based on a ratio of inner products.

   (f) Combine the Fourier coefficients with the basis functions to obtain the local approximation.

   (g) Combine the weight functions with the local approximation values to obtain the Jth portion of the global approximation.

4. Sum the global approximations over all J to obtain the overall function approximation.

2. Time-Varying Function Estimation Using GLO-MAP

In the input-output approximation presented here, GLO-MAP needs to estimate $g(\boldsymbol{x}_g)$ so that the error, $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}_g \rightarrow 0$. The assumed form of this nonlinear function is

$$g(\boldsymbol{x}_g, t) = \mathrm{K}_g \underbrace{\boldsymbol{\Phi}(\boldsymbol{x}_g) w(\boldsymbol{x}_g)}_{\equiv \boldsymbol{\psi}(\boldsymbol{x}_g)} \tag{5.15}$$

where $\boldsymbol{x} = \boldsymbol{x}(t)$, K_g is a constant matrix of Fourier coefficients that multiplies the matrix of basis function combinations, $\boldsymbol{\Phi}$, and the vector of weight function combinations, w. Here, the subscript g denotes the nonlinear estimate of the term obtained with the help of GLO-MAP. The adaptive estimation of the Fourier coefficient matrix and the control influence matrix are the final steps in the nonlinear parameter identification algorithm. The coefficient matrix, K_g, has dimensions: (number of states) $\times$ (number of acceptable basis function combinations). This will be explained in greater detail in the following discussion. Substituting this approximation into the true state equation yields

$$\dot{\boldsymbol{x}}_t(t) = \mathrm{A}_L \boldsymbol{x}_\mathrm{t}(\mathrm{t}) + \mathrm{B}_\mathrm{t} \boldsymbol{u}_\mathrm{t}(\mathrm{t}) + \mathrm{K}_\mathrm{t} \boldsymbol{\psi}(\boldsymbol{x}_\mathrm{t}) + \epsilon \tag{5.16}$$

where $\boldsymbol{x}_t$ is the true hidden state vector, B_t is the true control influence matrix not necessarily equal to its estimated value B_L, and ϵ is the difference between the nonlinear function approximation and the true nonlinear function. It is important to note that the state space A_L matrix is assumed to be estimated exactly by the Eigen Realization Algorithm. This is valid because in the ERA development it was assumed that A_L captured all of the modal frequencies of interest.

The identified system will not exactly match the true system given by Equation 5.16. Instead, the nonlinearly identified system will contain an estimate for the Fourier coefficient matrix K_g as well as for the control influence matrix B_g. These

estimates will attempt to capture the difference between the least squares estimated state vector, $\hat{\boldsymbol{x}}$, and the state vector estimated with the nonlinear function estimation routine, $\boldsymbol{x}_g$.

The estimated control influence matrix, $\mathrm{B_g}$, will not capture a nonlinear relationship between the state vector and the control, but this term will capture the any difference in this matrix between the linearly-estimated system and the nonlinearly-estimated one. On the other hand, the Fourier coefficient matrix combined with the GLO-MAP polynomial generation will model any nonlinear relationship in the equations of motion for the state vector. The adaptive estimation equations for these terms is developed in Appendix B as well as in Reference [7].

$$\dot{\mathrm{B}}_g \;=\; -P\boldsymbol{e}\boldsymbol{u}^T\Gamma_1^T \tag{5.17}$$

$$\dot{\mathrm{K}}_g \;=\; -P\boldsymbol{e}\boldsymbol{\psi}^T(\boldsymbol{x}_g)\Gamma_2^T \tag{5.18}$$

Here, P is a positive definite symmetric matrix, and Γ_1 and Γ_2 are convergence parameter matrices with full rank. All of these matrices determine the convergence speed of the algorithm, and they must be tuned accordingly. A formal proof of convergence is in Reference [7].

a.   Grid the Domain

The method begins by gridding the domain of the system. The grid is set up such that the state vector will never leave the grid domain. This may seem unrealistic since the true state vector is unknown, but the least squares estimation process provides estimates for the best guess of the hidden state vector. These estimates can be used as a starting point for the grid.

Once the domain for the grid has been chosen, the grid itself can be generated. It is recommended that a reasonably small number of centroids or grid points are

initially chosen for each dimension. Otherwise the number of grid points will become an unnecessary computational burden when the individual grids are combined. If the system exhibits highly nonlinear behaviors that cannot be estimated well, then the grid size may be decreased as needed.

The adaptive estimation algorithm only estimates the K_g elements. The polynomial combination $\boldsymbol{\psi}(\boldsymbol{x}_g)$ is determined by the current state of the system. Initially, all coefficients are zero, the state vector is initialized according to the least squares estimate $\hat{\boldsymbol{x}}$ at time zero, and $\boldsymbol{y}_g(t_0) = C_L \boldsymbol{x}_g(t_0) + D_L \boldsymbol{u}_{\mathrm{meas}}(t_0)$.

Application to a Duffing Oscillator: The nonlinear spring-mass-damper has two degrees-of-freedom, and the grid domains have been specified as $[-1, \ 2]$ in both directions. These values were chosen after examining the least squares estimated state vector. Choosing four intervals in each domain would give centroids at { -1, -0.25, 0.5, 1.25, 2} in each direction. It is only convenient, not necessary, that the interval length is the same in both directions.

Now, one must obtain the coordinates for all possible grid points in the domain. Rather than writing these combinations, they are displayed in graphical format in Figure 30 on the next page. There are twenty-five grid points total.

It is clear from this simple example that as the number of degrees-of-freedom increases, the total number of grid points quickly becomes very large. The number of degrees-of-freedom also partially determines the number of coefficients that must be adaptively estimated.

b.   Integration and Adaptation

The estimation algorithm is completed inside a numerical integration routine. It is important that the integration routine has a fixed time step equal to the time step of
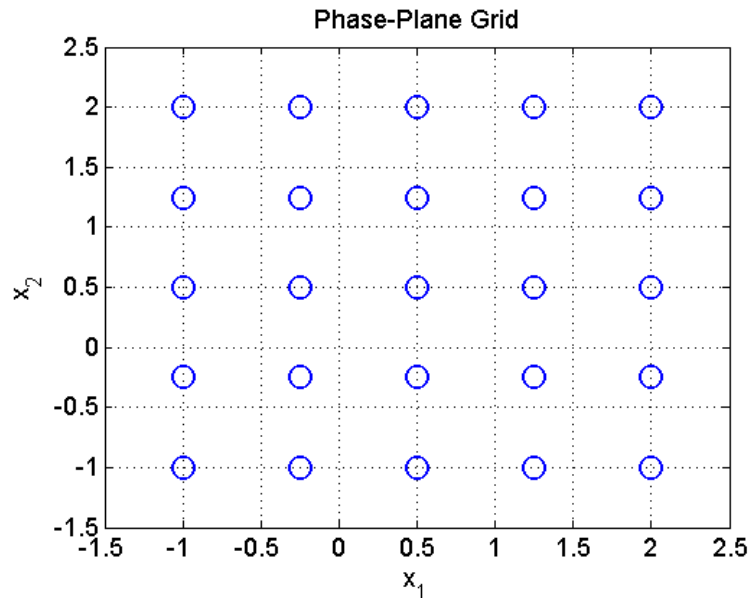
Fig. 30. Grid Point Locations for Spring-Mass-Damper

the system measurements. This is due to the fact that the error $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}_g$ is based on $\hat{\boldsymbol{x}}$, and values for $\hat{\boldsymbol{x}}$ are only known at the measurement times. Comparing these values to $\boldsymbol{x}_g$ at multiple intermediate times could prevent convergence or make the system unstable.

The following steps are completed inside the numerical integration solver for a given time step, t_i. Let the estimated state vector at this time be denoted by x_{gi}, and let the estimated state vector be $\hat{x}_i$. The matrix of current coefficient estimates, K_g^i, and the current estimate for the control influence matrix, B_g^i, are also inputted.

1. Begin by finding the nearest grid points for the state vector.

Application to a Duffing Oscillator: In the example, $x_{g0} = [-0.4661, \ -0.6303]^T$. This point is denoted by the red x, so it would be nearest to the grid points labeled 1, 2, 6, and 7 in Figure 31 on the following page. The solid line in this figure is the time history of the $\boldsymbol{x}_g$ vector as it travels through the phase plane.

In this figure, the grid points have been numbered for easy reference although this is not required for the estimation. It is important, however, that the order of the grid points remains constant throughout the estimation process. This order will determine the placement of values calculated in later steps.
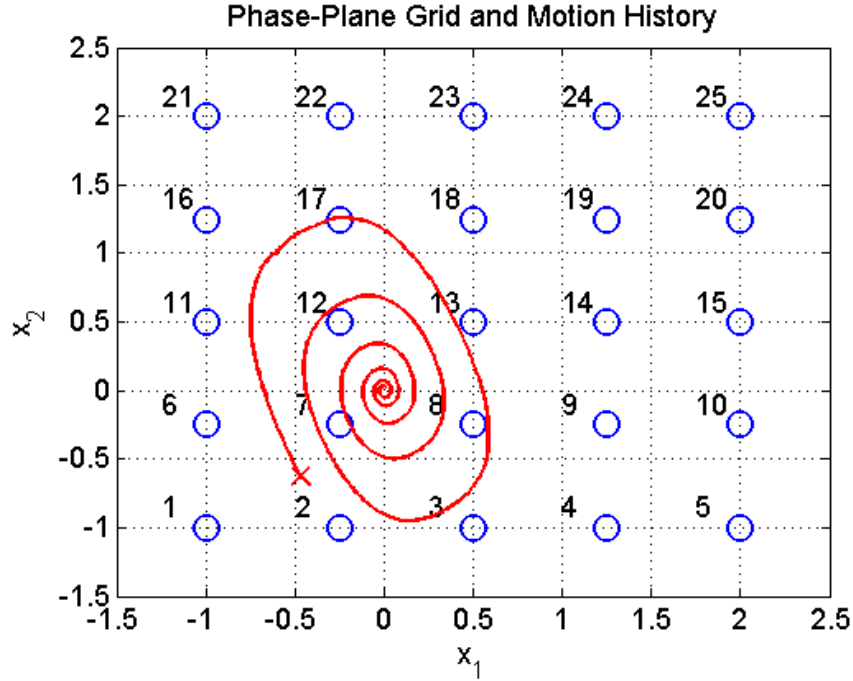
Fig. 31. Phase-Plane Grid Points and State Vector x_g Time History

2. The state vector is then converted into nondimensional coordinates with respect to each of the nearest grid points. Choosing a uniform grid size for all states will simplify this process.

$$^{J}\boldsymbol{\xi} \equiv (\boldsymbol{x}_{gi} - {}^{J}\mathcal{X})/h \qquad (5.19)$$

Here, the J superscript denotes the Jth centroid, $\boldsymbol{\xi}$ is the nondimensional coordinate vector, and h is the grid size. These local coordinates are used to calculate the weight and basis function values in the following steps.

Application to a Duffing Oscillator: Table V shows the values for the local coordinates of $\boldsymbol{x}_{g0}$ with respect to the four nearest grid points.

Table V. Nondimensional Coordinates for x_{g0}

Grid point	ξ_1	ξ_2
1	0.7119	0.4930
2	−0.2881	0.4930
6	0.7119	−0.5070
7	−0.2881	−0.5070

3. Next, the weight function values are calculated. The total weight function value for a state vector with respect to each centroid is simply the product of the one dimensional weight functions for all states. The generic form for the single state weight function is given by the following.

$$w(\xi_i) = 1 - \xi_i^2(3 - 2|\xi_i|) \tag{5.20}$$

This function was generated to have zero slope at $\xi_i = 0$ and at $\xi_i = \pm 1$. In addition, the maximum magnitude of the weight function equals one [7] [14]. It should also be noted that the total weight function vector has as many entries as the number of centroids. Only two, raised to the power of the number of states, entries of the weight function vector are nonzero.

Application to a Duffing Oscillator: The spring-mass-damper example has two degrees-of-freedom, so its weight function value is simply the product of each individual weight function.

$$w(^J\boldsymbol{\xi}) = \left(1 - {}^J\xi_1^2(3 - 2|\xi_1|)\right)\left(1 - {}^J\xi_2^2(3 - 2|\xi_2|)\right) \tag{5.21}$$

These values are aligned in a vector in order of grid point number. The total weight vector for this example is a column of twenty-five values where the nonzero values are listed below.

$$
\begin{bmatrix} w(^1\boldsymbol{\xi}) \\ w(^2\boldsymbol{\xi}) \\ w(^6\boldsymbol{\xi}) \\ w(^7\boldsymbol{\xi}) \end{bmatrix} = \begin{bmatrix} 0.1027 \\ 0.4078 \\ 0.0985 \\ 0.3910 \end{bmatrix} \tag{5.22}
$$

The observant reader may notice that the sum of the weight functions with respect to these four centroids is equal to one. This will always be true. It is a consequence of the choice of weight function and the product combination of multiple weight functions for multiple states.

4. Similar to weight functions, basis functions are also calculated as products of one another for multi-dimensional cases. However, the calculation is somewhat more complicated because there are many choices for the basis function combinations. The basis functions presented in Table VI were generated to be orthogonal using the Gram-Schmidt process [7]. To start, calculate this set of basis functions for

Table VI. Basis Functions [7]

Degree	Basis Function
0	1
1	ξ
2	$(-2 + 15\xi^2)/13$
3	$(-9\xi + 28\xi^3)/19$
$\vdots$	$\vdots$

each local coordinate. Then the total set of basis functions for each grid point is just the product of all possible combinations of the basis functions.

$$\mathbf{\Phi} = \Pi_{i=1}^{n} \varphi_\kappa(\xi_i) \tag{5.23}$$

In this equation, $\varphi_\kappa(\xi_i)$ is the basis function for the ith state, and κ denotes the order of the ith basis function which will vary between different states.

Application to a Duffing Oscillator: Looking back to the nondimensionalized coordinates from Table V on page 69, there are only four distinct local coordinates. The basis functions of order zero through three for these coordinates are listed in Table VII.

Table VII. Individual Basis Functions for Local Coordinates

	ξ_i			
	0.7119	0.493	-0.2881	-0.507
$\varphi_0(\xi_i)$	1	1	1	1
$\varphi_1(\xi_i)$	0.7119	0.493	-0.2881	-0.507
$\varphi_2(\xi_i)$	0.431	0.1266	-0.0581	0.1428
$\varphi_3(\xi_i)$	0.1945	-0.0569	0.1012	0.0481

The order of magnitude of the total basis function combination is just the sum of the individual basis function orders. Some of these basis function combinations produce functions with high orders of magnitude, but one of the beauties of GLO-MAP is that only low-order polynomials are required to fit the function in the local interval. Therefore, the higher-order basis function combinations are not necessary and can be omitted. The $\mathbf{\Phi}$ matrix has the dimensions: (number

of acceptable basis function combinations) × (number of grid points).

Application to a Duffing Oscillator: The spring-mass-damper example uses four basis functions: degrees zero through three. For two states, that makes sixteen possible basis function combinations. Table VIII lists all of these combinations. For the duffing oscillator, the maximum order of the basis functions

Table VIII. Possible Basis Function for the SMD

Combination	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\varphi(\xi_1)$	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$\varphi(\xi_2)$	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3

was restricted to four. This leads to combinations 12, 15, and 16 being dropped. The combined basis function values for the remaining combinations can be seen in Table IX on the following page. Once again, these are only the nonzero entries of the basis function matrix. In the example given here, $\boldsymbol{\Phi}$ has dimensions 13×25. The first, second, sixth, and seventh columns are shown in Table IX on the next page, and the remaining columns are filled with zeros.

5. Combine the basis and weight functions to obtain the linear combinations of polynomials.

$$\boldsymbol{\psi}(\boldsymbol{x}_{gi}) = \boldsymbol{\Phi}(\boldsymbol{x}_{gi})w(\boldsymbol{x}_{gi}) \qquad (5.24)$$

This vector is premultiplied by the coefficient matrix to obtain the nonlinear polynomial estimate as a function of that specific state vector at that specific time $\boldsymbol{x}_{gi}$.

$$g(\boldsymbol{x}_{gi}) = \mathrm{K}_g^{\mathrm{i}}\boldsymbol{\psi}(\boldsymbol{x}_{gi}) \qquad (5.25)$$

Table IX. Basis Function Combination Values for x_{g0}

	$^1\boldsymbol{\xi}$	$^2\boldsymbol{\xi}$	$^6\boldsymbol{\xi}$	$^7\boldsymbol{\xi}$
φ_{00}	1	1	1	1
φ_{10}	0.7119	-0.2881	0.7119	-0.2881
φ_{20}	0.431	-0.0581	0.431	-0.0581
φ_{30}	0.1945	0.1012	0.1945	0.1012
φ_{01}	0.493	0.493	-0.507	-0.507
φ_{11}	0.351	-0.142	-0.3609	0.1461
φ_{21}	0.2125	-0.0286	-0.2185	0.0294
φ_{31}	0.0959	0.0499	-0.0986	-0.0513
φ_{02}	0.1266	0.1266	0.1428	0.1428
φ_{12}	0.0901	-0.0365	0.1016	-0.0411
φ_{22}	0.0546	-0.0074	0.0615	-0.0083
φ_{03}	-0.0569	-0.0569	0.0481	0.0481
φ_{13}	-0.0405	0.0164	0.0342	-0.0139

Then the derivative of the GLO-MAP-estimated state vector can be calculated.

$$\dot{\boldsymbol{x}}_{gi} = A_L \boldsymbol{x}_{gi} + B_g^i \boldsymbol{u}_{\text{meas}} + g(\boldsymbol{x}_{gi}) \qquad (5.26)$$

Application to a Duffing Oscillator: The values of $\boldsymbol{\Phi}$ and w given previously for the first time step of the spring-mass-damper example combine to produce

$$\boldsymbol{\psi}(\boldsymbol{x}_{g0}) = \begin{bmatrix} 1 & -0.0869 & 0.0403 & 0.12 & 0.0035 & -0.0003 & 0.0001 & \cdots \\[2mm] \cdots & 0.0004 & 0.1345 & -0.0117 & 0.0054 & -0.0055 & 0.0005 \end{bmatrix}^T \quad (5.27)$$

It is trivial to perform the $K_g^0 \boldsymbol{\psi}(\boldsymbol{x}_{g0})$ multiplication for the spring-mass-damper system because at t_0 the coefficient matrix is initialized to zero. Again, the dimensions of this coefficient matrix are: (number of states $= 2$) $\times$ (number of acceptable basis function combinations $= 13$).

6. Finally, it is necessary to calculate the estimated coefficient derivative, also known as the adaptation of the coefficient matrix, and to calculate the control influence matrix derivative. This adaptation is given by Equations 5.18 to 5.18 on page 65 which are rewritten below for convenience.

$$\dot{B}_g = -P e \boldsymbol{u}^T \Gamma_1^T$$
$$\dot{K}_g = -P e \boldsymbol{\psi}^T(\boldsymbol{x}_g) \Gamma_2^T$$

Reference [7] defines the variables in a slightly different manner, so the reader should use caution when comparing the two algorithm developments.

This process is repeated for the entire time history. If the error $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}_g$ converges, then the coefficients, and therefore the estimated output $\boldsymbol{y}_g$, will also converge.

Application to a Duffing Oscillator: Returning to the duffing oscillator example, the final estimation can be seen in the following figures. Figure 32 shows the measured output and the output from the nonlinear input-output approximation on the left. The two sets of data appear to coincide. A plot of the error between the measured and the estimated output, shown on the right, shows that the two cases do match quite closely. Finally, the convergence of the coefficient matrix terms can be seen in Figure 33 on the following page. From this plot, one can see that the coefficients converge to constant values within the simulation time. This system was not excited by a control input, so the control influence matrix was not estimated.
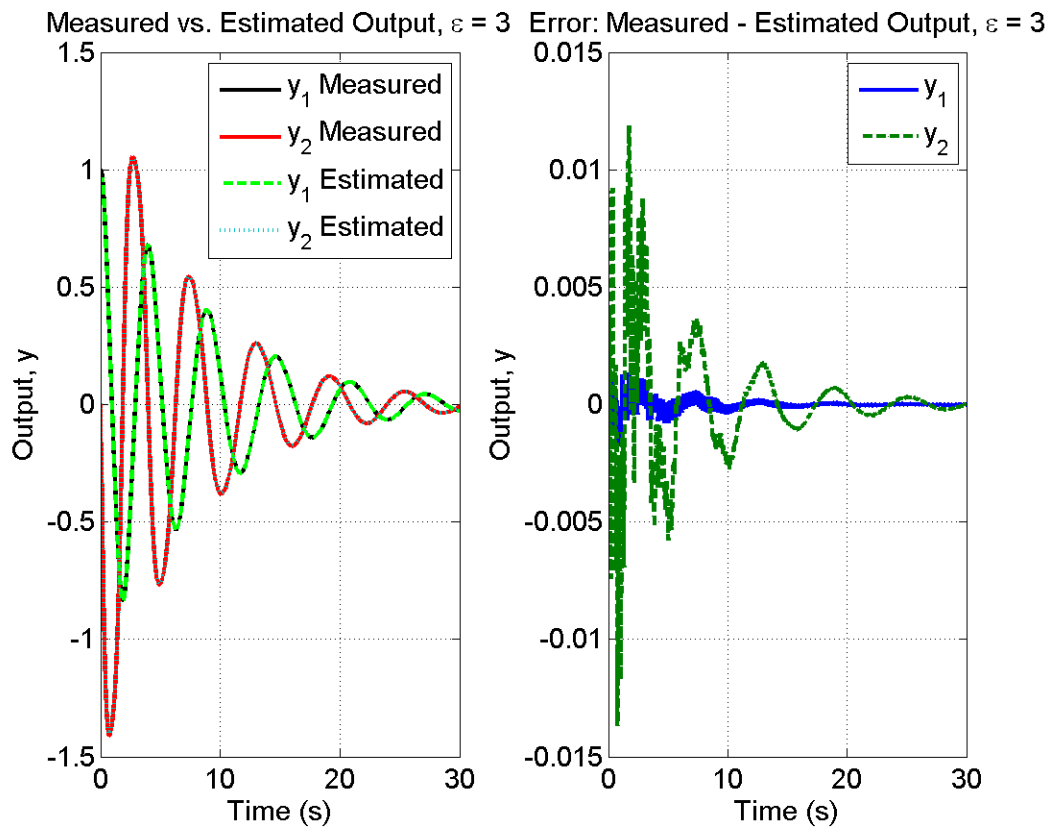
Fig. 32. Error between Measurements and Nonlinear Estimated Output

The system simulated using the nonlinearly identified model yielded significantly
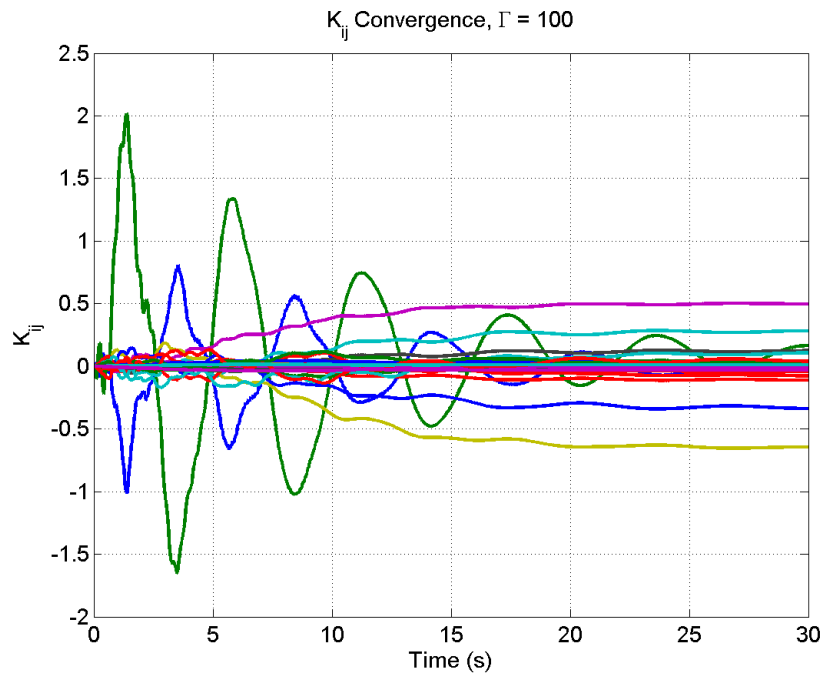
Fig. 33. Coefficient Matrix Convergence

lower error than that of the linearly identified model. This reveals that the combination of ERA and GLO-MAP function estimation for input-output approximation of nonlinear systems easily outperforms ERA by itself. In the next chapter, this algorithm will be applied to more complicated problems than the one presented here.

CHAPTER VI

INPUT-OUTPUT APPROXIMATION FOR ROTATING FLEXIBLE SYSTEMS

The previous chapter presented a robust nonlinear input-output approximation method that was recently developed at Texas A&M University. This method combines linear estimation with a unique nonlinear function estimator to approximate nonlinear systems. Now the algorithm will be applied to the class of rotating flexible systems the equations for which were developed in Chapter III.

A.   Prescribed Rotation Rate

Linear parameter identification in the form of the Eigen Realization Algorithm (ERA) was applied to the rotating spacecraft with constant rotation rate for the linear and quadratic assumed modes in Chapter IV. The method was shown to be successful as expected because the equations of motion for these cases are linear. However, simulating the rotational spacecraft with constant angular velocity using the nonlinear quadratic assumed modes yields a set of nonlinear equations which ERA is not equipped to handle. This is the first rotating flexible system to which the nonlinear approximation algorithm will be applied.

1.   Quadratic Assumed Modes with Higher-Order Terms

The simplest case for this set of equations is, of course, the case of one mode and a constant rotation rate. For this case, the set of equations given in Equation 3.47 on page 29 simplifies to the following

$$(M + \mathcal{M}_{NL})\ddot{q} + C\dot{q} + \mathcal{K}_L q + \mathcal{K}_{NL} = 0 \qquad (6.1)$$

where the coefficients, defined earlier, have been simplified for one mode.

$$M_{11} = \int_0^L \rho \, \phi_1^2 \, dx \tag{6.2}$$

$$K_{11} = \int_0^L EI\phi_1''^2 dx \tag{6.3}$$

$$N_1 = \int_0^L \rho \, (R + x) \, \phi_1 \, dx \tag{6.4}$$

$$H_{11} = \int_0^L \rho \, (R + x)\psi_{11} \, dx \tag{6.5}$$

$$F_{111} = \int_0^L \rho \, \phi_1\psi_1 \, dx \tag{6.6}$$

$$G_{1111} = \int_0^L \rho \, \psi_{11}^2 \, dx \tag{6.7}$$

$$\mathcal{M}_{NL} = 4q^2 \, G_{1111} \tag{6.8}$$

$$\mathcal{K}_L = K_{11} - \dot{\theta}^2(M_{11} + 2H_{11}) \tag{6.9}$$

$$\mathcal{K}_{NL} = \mathcal{K}_F + \mathcal{K}_G \tag{6.10}$$

$$\mathcal{K}_F = 2\dot{\theta} \, F_{111}(\dot{q}^2 - q\dot{q}) \tag{6.11}$$

$$\mathcal{K}_G = 4q \, G_{1111}(\dot{q}^2 - \dot{\theta}^2 q^2) \tag{6.12}$$

In these equations, the variable x denotes the transverse distance from the hub to a generic point on the flexible beam. It should not be confused with the state vector $\boldsymbol{x}$ which is printed in bold.

From this point forward, the subscript **1** will be substituted for repeated 1 subscripts. Values for these coefficients will be calculated using a admissible functions. The first and only mode shape is given below.

$$\phi_1 = \left(\frac{x}{L}\right)^2 \tag{6.13}$$

Calculating the first and second derivatives of this function with respect to x gives

the following.

$$\phi_1' = \frac{2x}{L^2} \tag{6.14}$$

$$\phi_1'' = \frac{2}{L^2} \tag{6.15}$$

Similarly, the equation for the quadratic mode yields this expression.

$$\psi_1 = \frac{-2x^3}{3L^4} \tag{6.16}$$

These can be used to calculate the constants given in Equations 6.2 to 6.7 on the preceding page. After integration and simplification, expressions for these coefficients can be found.

$$M_1 = \frac{\rho L}{5} \tag{6.17}$$

$$K_1 = 4\frac{EI}{L^2} \tag{6.18}$$

$$N_1 = \frac{\rho L^2}{4} + \frac{\rho RL}{3} \tag{6.19}$$

$$H_1 = -\frac{2\rho L}{15} - \frac{\rho R}{6} \tag{6.20}$$

$$F_1 = \frac{\rho}{18} \tag{6.21}$$

$$G_1 = \frac{2}{63}\frac{\rho}{L} \tag{6.22}$$

The same spacecraft configuration simulated in previous chapters is implemented here. Table X on the next page gives values for the system parameters. It is trivial to substitute these parameters and obtain numerical values for the system constants. These values can be seen in Table XI on the following page. The reader may notice that the terms F_1 and G_1 are significantly smaller than the other terms. However, they still contribute to the motion of the system as seen in Chapter III and in this chapter as well. Once the generalized coordinate and its derivative have been calculated as functions of time, the data can be translated into deflection and deflection rate at

Table X. Rotational Spacecraft Parameters

Beam Length	45.52 in
Beam Modulus of Elasticity	161.6e6 oz/in^2
Second Moment of Inertia of Bending Section	0.000813 in^4
Beam Density	0.003007 oz s^2/in
Hub Radius	5.5470 in
Hub Mass	31.59 oz
Rayleigh Damping Coefficient, α	0.25
Rayleigh Damping Coefficient, β	0.0002

Table XI. Rotational Spacecraft Constants

$M_1 = 0.0274$ oz s^2	$K_1 = 55.71$ oz in^2
$N_1 = 1.811$ oz s^2 in	$H_1 = -0.0210$ oz s^2
$F_1 = 1.671e{-}4$ oz s^2/in	$G_1 = 2.0970e{-}6$ oz s^2/in^2

different measurement locations on the beam. The deflection at the jth measurement location is given by the Assumed Modes assumption

$$\begin{aligned} y(x_j) &= \phi_1(x_j)q \\ &= \left(\frac{x_j}{L}\right)^2 q \end{aligned}$$

and similarly, the deflection rate at the jth measurement location is given by the following equation.

$$\begin{aligned} \dot{y}(x_j) &= \phi_1(x_j)\dot{q} \\ &= \left(\frac{x_j}{L}\right)^2 \dot{q} \end{aligned}$$

In these equations, the variable y is the transverse deflection of the beam and should not be confused with the state space output vector $\boldsymbol{y}$.
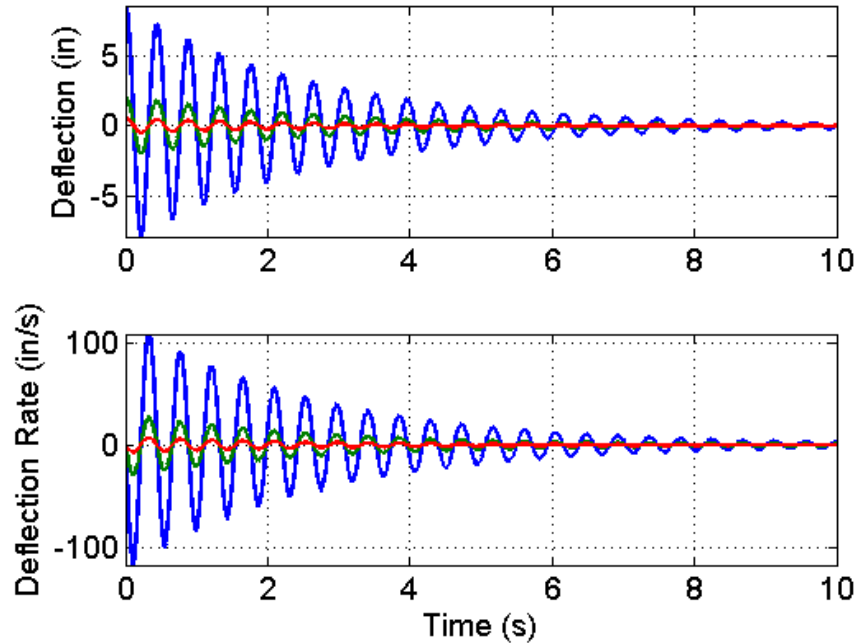
Fig. 34. Beam Deflection: 1 Nonlinear Mode, $\dot{\theta}$ Constant

The system was simulated with initial beam deflections of two, four, and eight inches at the quarter-span, midspan, and tip measurement points respectively. The constant rotation rate was set at 1.5 rad/s which is slightly less than ninety degrees per second. Figure 34 is a plot of the transverse beam deflection at the measurement locations.

2. Application of Nonlinear Input-Output Approximation Algorithm

Once measurements of the beam displacement and displacement rate have been gathered for the beam, the nonlinear input-output approximation algorithm can be applied. Both the generalized coordinates and the first derivative of the generalized

coordinates must be directly or indirectly included in the measurements because the measurements must contain information about all of the states of the system. As mentioned in Chapter V, if only some of the states are included in the output, the product $C_L^T C_L$ becomes singular, and the least squares state estimation will not be successful.

For this work, the measurement data, $\boldsymbol{y}_{\text{meas}}$, is arranged as follows: deflection at the quarter-span, midspan, tip, and deflection rate at the quarter-span, midspan, and tip. The order is only important for comparing numerical results, such as the specific values given in the linear state space realization for example, but it is not important for algorithm convergence. It is important to note that the beam deflections and deflection rates are not states of the system. Instead, they are linear combinations of the states of the system. This is a small but important distinction. In the duffing oscillator example, the measured outputs were the states themselves, but from this example, it is clear that this is not necessary.

a. Linear Input-Output Approximation

The first step is to apply the Eigen Realization Algorithm to this data set. The resulting continuous-time state space realization is given in the following equations.

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1.441 & 40.13 \\ -5.128 & -2.151 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} -0.9104 \\ -2.197 \end{bmatrix} \tag{6.23}
$$

$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} -0.02454 & -0.2312 \\ -0.09818 & -0.925 \\ -0.3927 & -3.700 \\ 1.150 & -0.4788 \\ 4.602 & -1.915 \\ 18.41 & -7.66 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{6.24}
$$

In these equations, x_1 and x_2 represent generic states that do not necessarily equal the generalized coordinate and its derivative. It is important to recall from Chapter IV that ERA does not necessarily return the true system. Instead, it returns a linear transformation of the true system, so the generic states x_1 and x_2 could be related to q and $\dot{q}$ through linear transformations.
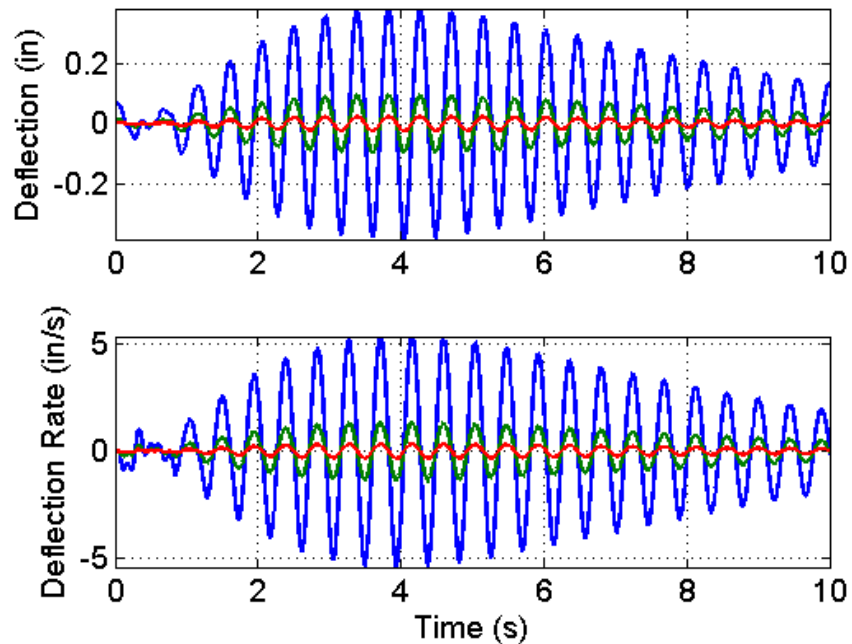

Fig. 35. Comparison of Beam Deflection: Measurements - ERA Estimates

Simulating this linear system and comparing it to the true, measured output

reveals significant errors. The error in the deflection is shown in Figure 35 on the preceding page. By comparing this figure with Figure 34 on page 81, one can see that the errors reach a maximum of approximately five percent between four and five seconds. Errors of this magnitude would be unacceptable in cases where high precision is required.

b. Least Squares Estimation

Since the linear state space realization is known, the best estimate of the hidden state vector can now be calculated using linear least squares. This system was excited by an initial deflection and no control, so the least squares estimation (LSE) is given by the following.

$$\hat{\boldsymbol{x}} = (C_L^T C_L)^{-1} C_L^T \boldsymbol{y}_{\text{meas}} \tag{6.25}$$

The error between the LSE-estimated state vector and the ERA-identified state vector is proportional to the error between the measured output and the ERA output. Similarly, the motion of the LSE-estimated state vector is proportional to the measured output. Therefore, plots of these variables will not be included because they provide the reader with no new information.

c. Nonlinear Function Estimation

Before proceeding, it will be advantageous to take another look at the equation of motion for this one degree-of-freedom case. Unlike the duffing oscillator example given in the previous chapter, the equation of motion for the rotating spacecraft has several forms of nonlinearity.

$$(M_{\mathbf{1}} + \underbrace{4\,G_{\mathbf{1}}q^2}_{\mathcal{M}_{NL}})\ddot{q} + C\dot{q} + \left(K_{\mathbf{1}} - \dot{\theta}^2(M_{\mathbf{1}} + 2H_{\mathbf{1}})\right)q + \underbrace{2\dot{\theta}\,F_{\mathbf{1}}(\dot{q}^2 - q\dot{q})}_{\mathcal{K}_F} + \underbrace{2q\,G_{\mathbf{1}}(2\dot{q}^2 - \dot{\theta}^2 q^2)}_{\mathcal{K}_G} = 0 \tag{6.26}$$

The nonlinear terms are identified above by their abbreviations. It is easy to see that the nonlinearities in this equation come from the following types of terms: $\dot{q}^2\ddot{q}$, $\dot{q}^2$, $\dot{q}^2 q$, $\dot{q}q$, and q^3, and this is a significant set of nonlinear terms.

The final step in the input-output approximation process for this system is the nonlinear function estimation. The estimation is straightforward and follows the adaptive estimation algorithm with GLO-MAP, which was described in the previous chapter. One drawback of the technique is that the convergence parameter matrix, Γ, must be tuned in order to achieve satisfactory performance. This is not a complicated process, but it is a delicate one. For simplicity, the gain matrix was assumed to be an identity matrix with a constant multiple, γ. The nonlinear parameter estimation was performed for a wide range of values, and $\gamma = 2.5$ was finally chosen as the gain that produces the best coefficient convergence. A time history of the Fourier coefficient estimates can be seen in Figure 36 on the following page. This figure shows that the values converge to constants within the allotted time.

The error between the measured output and the estimated output for the two methods can be seen in Figure 37 on page 87. The dark line is the error with respect to ERA, and the light line is the error with respect to the nonlinear input-output mapping. For all of the measurement locations, the nonlinear approximation outperformed the linear method. This provides enough confidence to perform the same investigation with an increased number of degrees of freedom.

3. Several Quadratic Assumed Modes with Higher-Order Terms

The same analysis was performed for the rotational spacecraft with three quadratic nonlinear modes. The total number of degrees-of-freedom was limited to three for all simulation cases because of memory constraints. Having more than three degrees-of-freedom produces more than six states. At this point, the author encountered
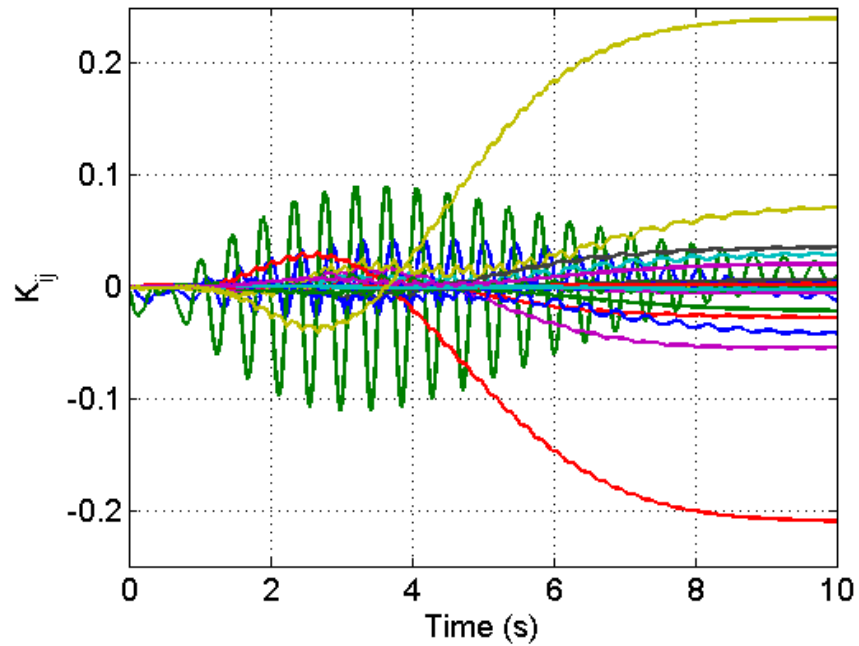
Fig. 36. Fourier Coefficient Convergence, 1 Nonlinear Mode, θ Prescribed, $\gamma = 2.5$

`Out of Memory` errors in Matlab due to the number of grid points required for the nonlinear function estimation. Recall that for the same number of grid points in each dimension, the total number of grid points is equal to the number of points in each dimension raised to the power of the number of dimensions. For these simulations, five grid points in each of eight dimensions produces 390625 total grid point combinations whereas five grid points in six dimensions produces only 15625 combinations.

The results of the previous estimation were employed as a starting point for this estimation, so the adaptive gain matrix, Γ, was initially chosen to be an identity matrix with a constant multiple, γ. To tune the gain for this test case, the estimation was run for a wide range of values. Ultimately, a gain of two was chosen. Gains that are to either too small or too large do not allow the Fourier coefficients to converge. The chosen value yields the convergence history seen in Figure 38 on page 88. This
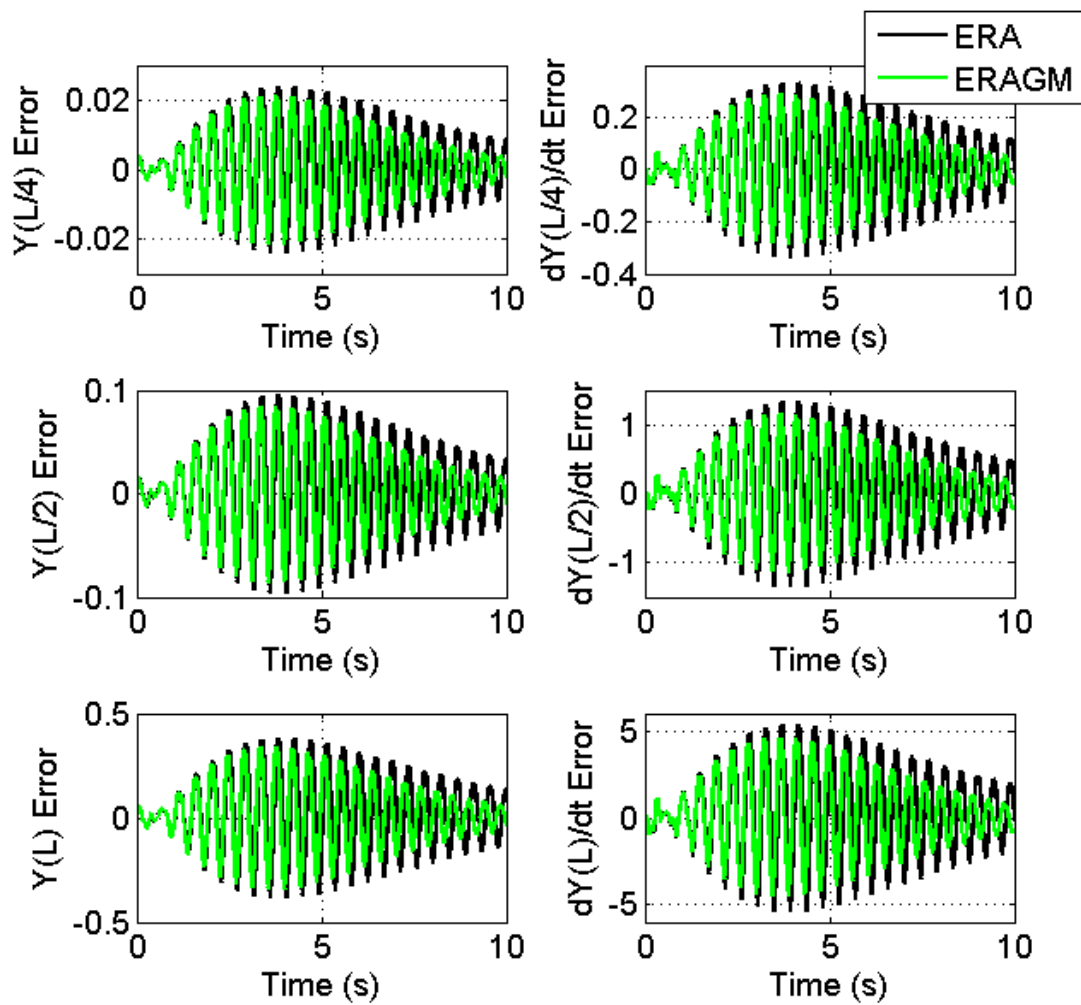
Fig. 37. Output Error Comparison, 1 Nonlinear Modes, θ Prescribed, $\gamma = 2.5$
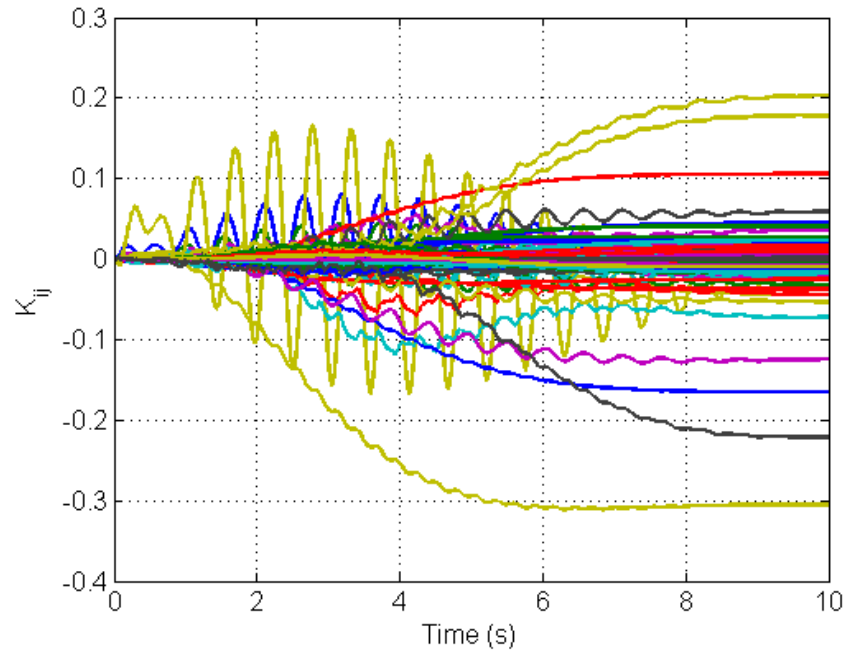
Fig. 38. Fourier Coefficient Convergence, 3 Nonlinear Modes, θ Prescribed, $\gamma = 2$

figure shows the individual elements of the Fourier coefficient matrix as a function of time. From this plot, one can easily see that the coefficients may oscillate, but the oscillation damps out before the end of the adaptation.

In addition, the estimate calculated using these coefficients in the nonlinear parameter identification is superior to the estimate using ERA by itself. Figure 39 shows the error between the true output and the identified models. The results displayed in this figure are even more compelling than that for the single degree-of-freedom case. ERA by itself is not nearly as reliable an estimator as the combination of ERA and the adaptive function estimator using GLO-MAP.

These results show that the nonlinear parameter identification method developed at Texas A&M University is successful in identifying the rotating spacecraft with flexible appendage. In fact, the simulations also suggest that as the number of
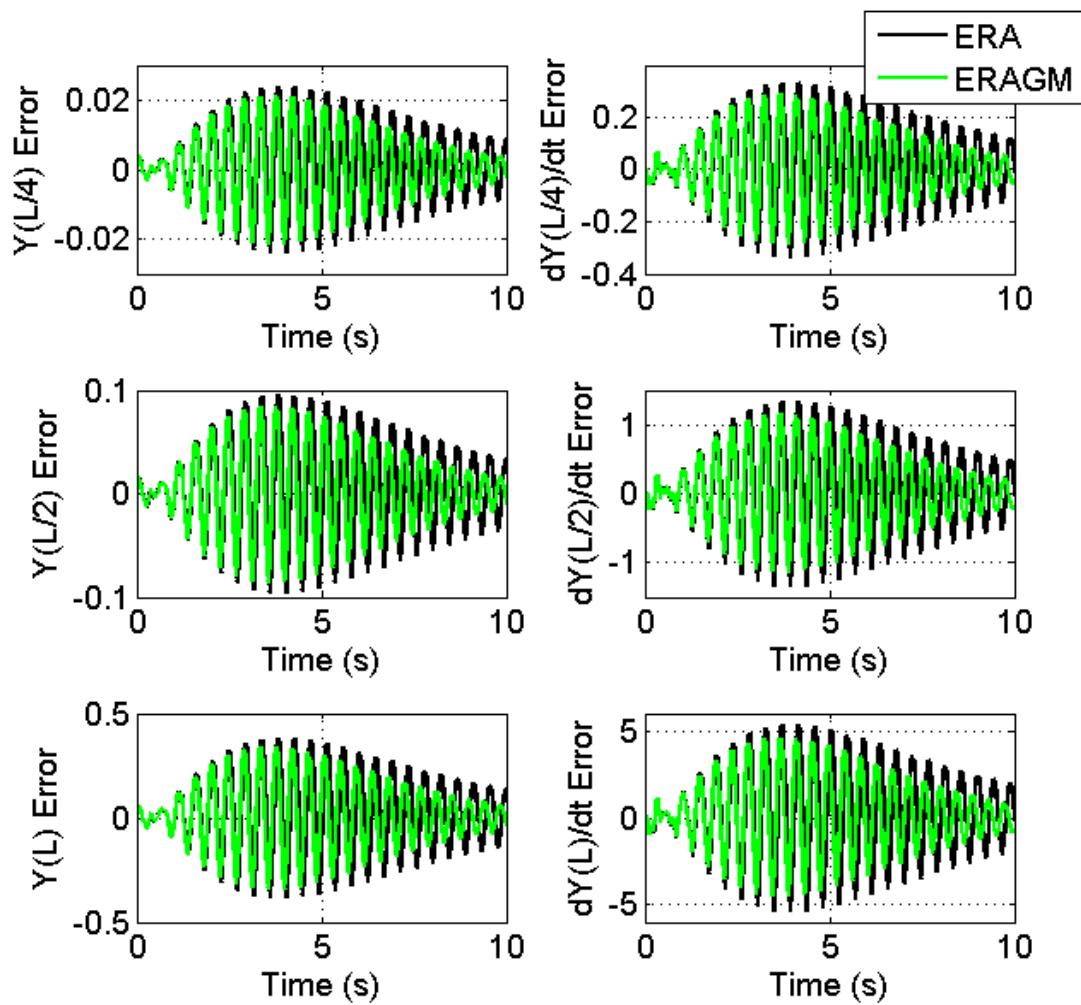
Fig. 39. Output Error Comparison, 3 Nonlinear Modes, θ Prescribed, $\gamma = 2$

degrees-of-freedom in the simulation increases, the combination of ERA and GLO-MAP becomes even more effective compared to only using ERA. This is reassuring because as the number of modes used in the simulation model increases, the measurements provide a more realistic picture of the true system output.

B. Unknown Rotation Rate

For a spacecraft with a flexible appendage rotating at unprescribed rate, Chapter III showed that the equations of motion are nonlinear regardless of the choice of assumed modes. Before proceeding to this level of complexity, it was decided that the nonlinear input-output approximation method should first be applied to a simple problem with analogous characteristics. In Chapter V, the nonlinear approximation algorithm was first applied to a nonlinear spring-mass-damper before it was applied to flexible motion earlier in this chapter. Similarly, the approximation will now be performed on a nonlinear spring-mass-damper system with rigid-body mode. The rigid-body mode will mimic the coupling between the flexible and rotational motion for the spacecraft with unprescribed rotation rate.

1. Duffing Oscillator Simplification

A rigid-body mode is present in the spacecraft with flexible appendage example. This mode can be simulated in a simple spring-mass-damper system by allowing the system to move relative to an inertially-fixed reference frame. Figure 40 on the next page is a visual representation of this system. The duffing oscillator is connected to a block of mass M that is not constrained to the reference frame. Therefore, when the small mass is initially displaced, the result is to produce motion in the entire system. The

equations of motion for this system are given below.

$$M\ddot{x}_1 + c(\dot{x}_1 - \dot{x}_2) + k(x_1 - x_2) + \varepsilon(x_1 - x_2)^3 = 0 \qquad (6.27)$$

$$m\ddot{x}_2 + c(\dot{x}_2 - \dot{x}_1) + k(x_2 - x_1) + \varepsilon(x_2 - x_1)^3 = 0 \qquad (6.28)$$

This set of equations was simulated with the following constants: $M = 3$, $m = 1$, $c = 0.25$, $k = 1$, and $\varepsilon = 3$. With the exception of M, these values are the same as for the regular duffing oscillator. Figure 41 on the following page is a plot of the system motion for the following initial state vector, $\boldsymbol{x}_0 = [2, 1, 0.1, 0, 5]^T$. Applying
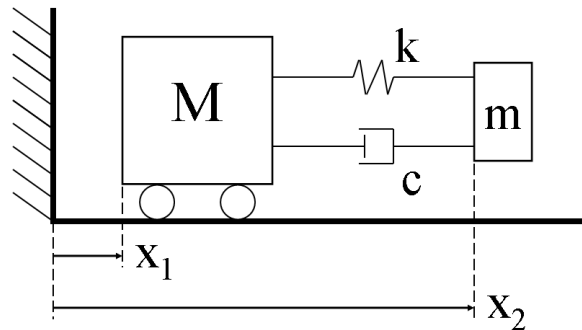
Fig. 40. Duffing Oscillator with Rigid-Body Mode

the Eigen Realization Algorithm to this system results in moderate approximation errors. For the velocity of the small mass, dx_2/dt, the percent error at four seconds is on the order of fifty percent. This is clearly significant, so ERA does not effectively approximate this system.

The nonlinear input-output approximation method was then applied to this output data set. As previously stated, one challenge of this method is tuning the adaptive gain parameters. Extensive efforts to tune the gain matrix could not produce a suitable combination of gains even if the grid size was significantly reduced. Either the gain was too small and the method would not converge, or the gain was too large and the coefficient estimates would oscillate. Figure 43 on page 94 shows two test
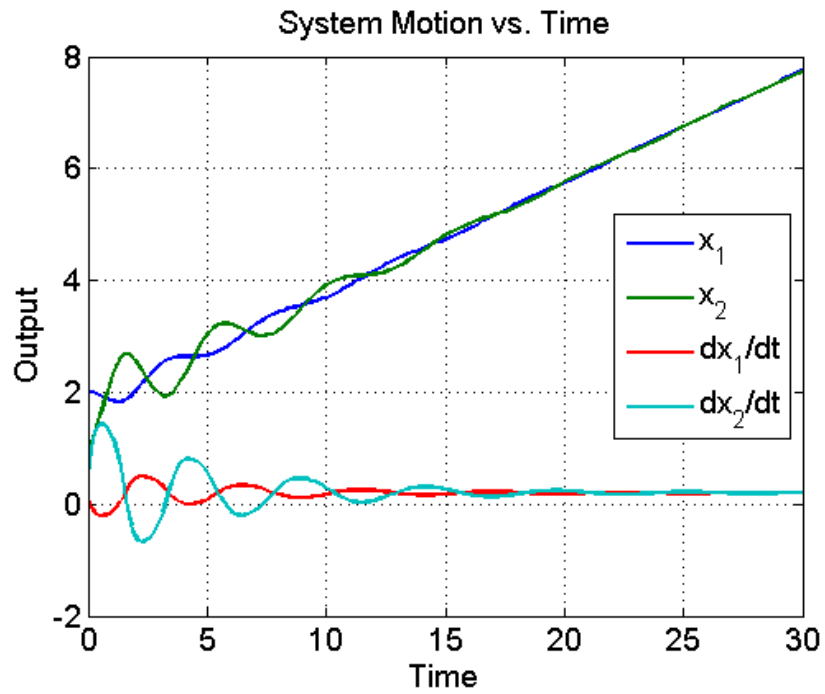
Fig. 41. Output for Duffing Oscillator with Rigid Body Mode
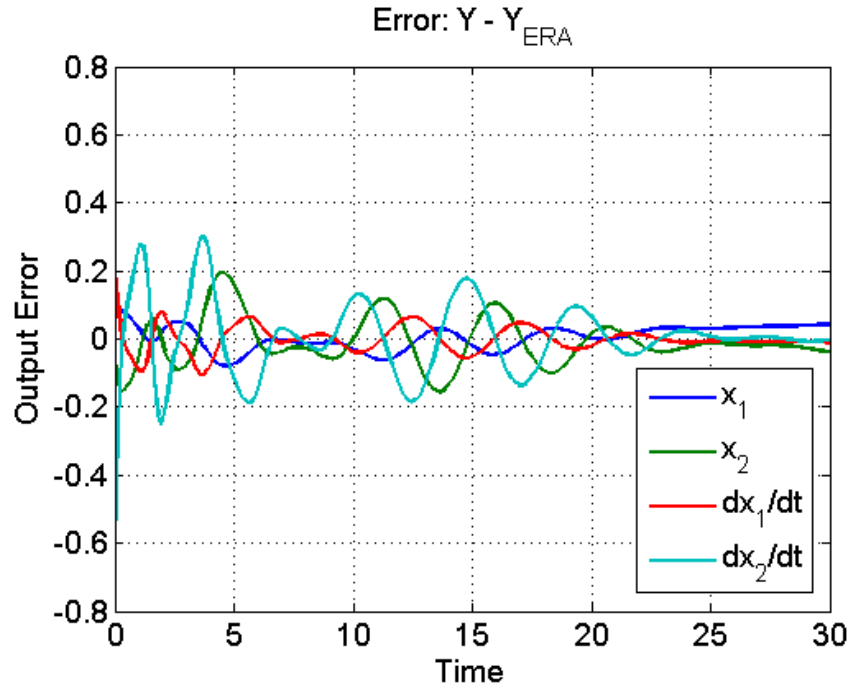
Fig. 42. Error: Measurements - ERA Output. Duffing Oscillator with Rigid Body Mode

cases. For simplicity, the gain matrix in these examples was defined to be an identity matrix multiplied by a constant value, γ; this simplification was not applied during the tuning process, but it is included here to more concisely display the results. With $\gamma = 0.1$, most of the coefficients converge, but they converge to values that do not noticeably change the approximation. On the other hand, a value of 1 for this gain produces coefficient estimates that are significantly larger, but the estimates oscillate without showing signs that they will ever converge. It is important to mention that the maximum basis function order for this simulation was decreased to three which decreased the number of coefficients to estimate without significantly degrading the performance.

This result is discouraging because it appears that the nonlinear input-output approximation algorithm fails for this case. However, there is an underlying cause for this apparent failure. In adaptive control, it is well known that a controller may adequately control a system without converging to the true parameters of the system. The analogy for adaptive estimation is that the estimator may track the output even if it does not converge to the true system parameters. To converge on the true parameters, the input signal must be persistently exciting. Persistence of excitation is a term that describes the richness of the input signal, which usually translates to the number of distinct frequencies in the signal. For an adaptive controller to estimate s parameters, the input signal must have at least $s/2$ distinct frequencies [16].

The duffing oscillator with rigid body mode has to estimate all of the K_g coefficients. In Chapter V, the dimensions of this matrix were given as the number of basis function combinations by the number of states. For this example, there are thirty-five basis function combinations and four states which leads to 140 different parameters that must be estimated. The initial condition input signal clearly does not contain a sufficient number of frequencies to estimate the coefficients.
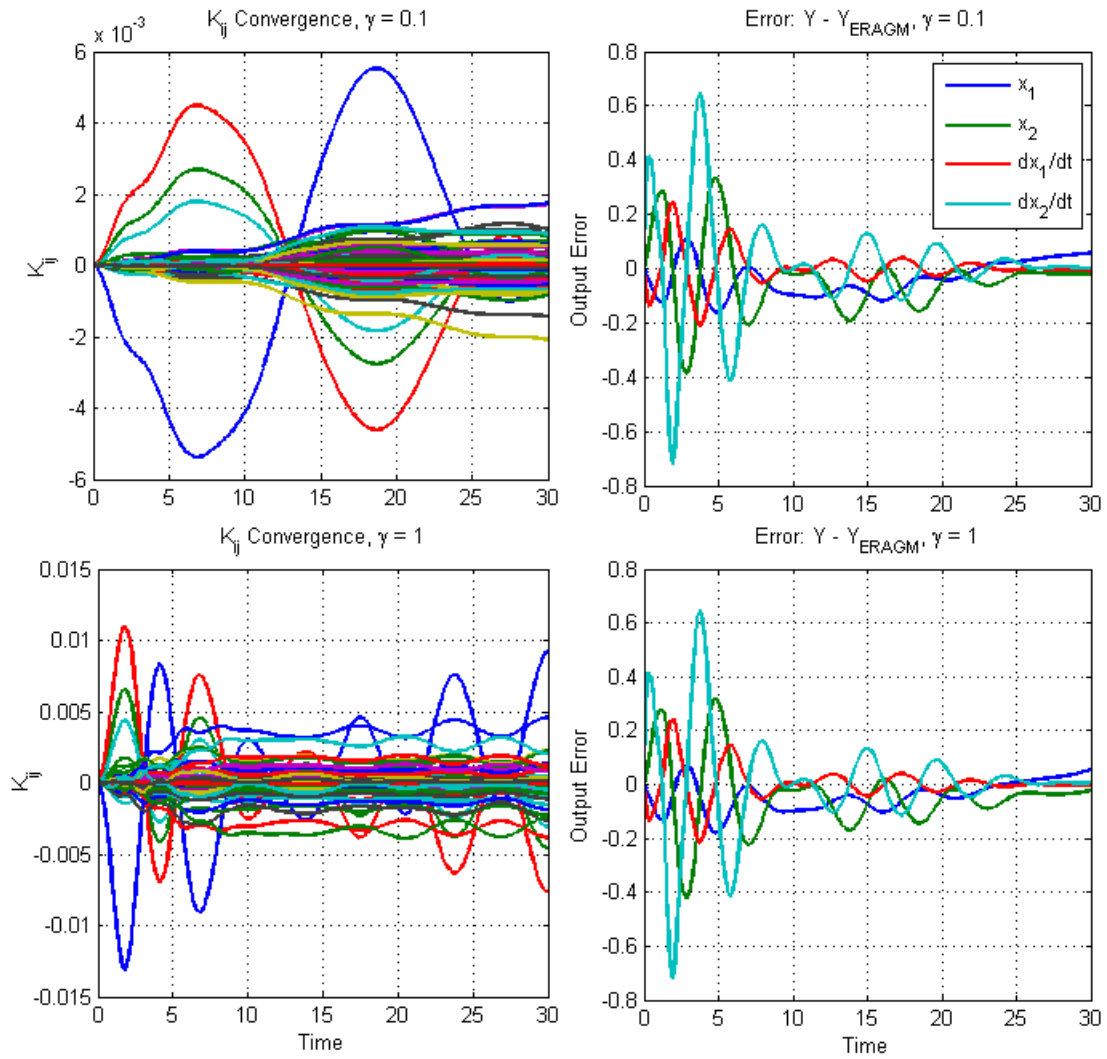
Fig. 43. Error: Measurements - ERA + GLO-MAP Output. Duffing Oscillator with Rigid Body Mode

In fact, this may lead an individual to question the results for the simple duffing oscillator. There were thirteen basis function combinations and two states, so the initial condition input was not persistently exciting for that example either. The duffing oscillator example obeyed the phenomenon outlined in the previous paragraph wherein an estimator may track the output even if the input is not persistently exciting. Unfortunately, it has been shown that the nonlinear input-output approximation for the duffing oscillator with rigid body mode cannot track the output when given an input that is not persistently exciting.

It is clear that if the nonlinear approximation algorithm could not estimate this simple problem then it would not be able to successfully approximate the more complicated problem of the spacecraft with flexible appendage rotating at an unprescribed rate. Further investigation of this method needs to be performed before the algorithm can be applied to a problem of this complexity. Suggestions on future research in this area are given in the following chapter.

CHAPTER VII

RECOMMENDATIONS

The results from this research were extremely encouraging, so there are several areas that can be expanded. It has been shown that in order to guarantee coefficient convergence in the nonlinear input-output approximation method, the input signal should be persistently exciting. Typical adaptive control literature suggests that to estimate a set of s linear parameters, the input signal must contain at least $s/2$ natural frequencies [16]. To test this hypothesis, the method should be applied to a simple system, like the duffing oscillator, with persistently exciting input. However, the Eigen Realization Algorithm, which is the first step of the nonlinear input-output approximation process, is only valid for systems excited by impulse or initial condition inputs, so other linear input-output approximation methods would need to be investigated.

The Observer/Kalman Filter Identification algorithm (OKID) mentioned in the introduction is a linear mapping that could be substituted for ERA. Both methods were developed for linear systems and depend on the Markov parameters. However, OKID contains an observer in addition to a state estimator. The observer gains can be tuned by the user to create desired observer convergence rates. This algorithm also takes an arbitrary input time history, which is the most attractive feature for this research. More information on this identification algorithm can be found in Chapter VI of Reference [4].

Another available estimation method is the Auto-Regressive Moving-Average with eXogenous inputs (ARMAX) model. This is an estimation scheme that identifies the system parameters as a function of both past inputs and outputs where the inputs come in the form of state inputs and exogenous control inputs [5]. The estimated

parameters of the system must appear linearly because they are approximated using a simple least squares routine. The Nonlinear ARMAX, or NARMAX model, is a modification of this routine that appears to estimate the parameters of a nonlinear input-output set in a manner similar to the nonlinear input-output approximation method developed in this work [17]. The author feels that the approximation method included in this research is significantly more flexible than the NARMAX model, but combining NARMAX with GLO-MAP may be a worthwhile endeavor.

As previously mentioned, the adaptive estimation algorithm in the nonlinear approximation requires the estimation of a relatively large parameter set. The obvious solution to this problem is to excite the system with a known, time-varying controller that is sufficiently exciting. One less obvious, and perhaps more creative, solution is to generate the input as a function of artificial states that are augmented to the true system. These states would be simulated with the true system, and the control on the true states would be linear combinations of these augmented states. The entire system, including these fictitious states, would be estimated by the linear parameter estimation algorithm, so the nonlinear input-output approximation would have full knowledge of the control structure. In addition, the controls would be linear functions that appear linearly in the state space model. One final advantage of this scheme is that the input frequencies can be controlled by the designer. All in all, this promises to be a clever means of producing persistently exciting motion which would then lead to successful system identification.

The nonlinear approximation algorithm could also be updated to handle constraints. It is well known that a system is governed by kinematic equations, which are known exactly, and dynamic equations, which contain the uncertainty and nonlinearity of the system. If one could separate the kinematic and dynamic motion of the linearly estimated system, then the adaptive function estimator would only need to

be employed for the system dynamics. The function estimator could be constrained to produce zero nonlinear motion in the exact equations, so all of the nonlinearity would be contained in the dynamic equations which should produce a more realistic approximation.

One final aspect that can still be explored in this research area is the extraction of system parameters from the identified model. Parameter identification can mimic the true system motion quite well, so the estimated system must contain accurate representations of the true system parameters, like inertia, stiffness, etc. Being able to extract these constants from the model would open a gauntlet of applications ranging from parasite detection, to determining flutter characteristics, and even to nondestructive testing.

Finally, with regard to the rotating flexible structures work, it would be interesting to perform hardware experiments to determine the correlation between the true motion of a rotational flexible system and the three simulation methods. Specifically, the differences between the quadratic and nonlinear assumed modes could be evaluated against measurement data. All in all, this research was very rewarding, and it would be worthwhile to expand this work in the future.

CHAPTER VIII

CONCLUSIONS

This research began with three main goals: develop a set of nonlinear equations of motion for the class of rotating spacecraft with flexible appendages, rigorously investigate the nonlinear input-output approximation method recently developed at Texas A&M, and apply this approximation technique to the spacecraft model developed herein. Each of these was extensively evaluated and the results were encouraging.

The higher-order model for the rotational structural dynamics was insightful. Traditional Assumed Modes uses linear assumed modes which produce a softening with increased rotation rates. This result is counter-intuitive and indeed incorrect. Previously developed quadratic assumed modes produce beam stiffening that is more realistic. The development of the nonlinear assumed modes in this work yields an even higher fidelity model from which rotational flexible motion can be calculated.

A comparison of linear and nonlinear input-output approximation techniques revealed that the nonlinear approximation significantly outperforms linear parameter identification for the class of nonlinear systems. Most real-world systems are governed by a set of nonlinear equations, so it is reasonable to conclude that the nonlinear algorithm would be useful in a wide variety of situations.

Finally, the combination of these major research thrusts was successful. The combined ERA/GLO-MAP approximation estimated the output significantly better than the ERA-identified model. Results also revealed that the nonlinear estimation becomes even more superior as the number of degrees-of-freedom in the motion increases. Flexible systems have an infinite number of modes, a large number of which can be measured, so the new approximation method would be extremely successful in real-world situations.

This work has applications in input-output approximation for spacecraft, and this was the major case to which the research was applied. However, the work could also be applied to the field of aeroelasticity. Flutter dynamics, especially for fighter aircraft, could be modeled and estimated using the techniques developed in this research.

REFERENCES

[1] J. L. Junkins and Y. Kim, *Introduction to Dynamics and Control of Flexible Structures.* Washington, D.C.: American Institute of Aeronautics and Astronautics, Inc., 1993.

[2] D. J. Segalman and C. R. Dohrmann, "A method for calculating the dynamics of rotating flexible structures. 1. derivation," in *Transactions of the ASME Journal of Vibration and Acoustics*, 1996, vol. 3, pp. 313–317.

[3] ——, "A method for calculating the dynamics of rotating flexible structures. 2. example calculations," in *Transactions of the ASME Journal of Vibration and Acoustics*, 1996, vol. 3, pp. 318–322.

[4] J. Juang, *Applied System Identification.* Englewood Cliffs, New Jersey: Pearson Education, 1994.

[5] K. Worden and G. Tomlinson, *Nonlinearity in Structural Dynamics: Detection, Identification, and Modeling.* Philadelphia, Pennsylvania: Institute of Physics Publishers, Inc., 2001.

[6] J. L. Junkins, P. Singla, T. D. Griffith, and T. Henderson, "Orthogonal global/local approximation in n-dimensions: Applications to i/o approximation," in *6th International Conference on Dynamics and Control of Systems and Structures in Space*, Cinque Terre, Italy, July 2004.

[7] P. Singla, T. Henderson, J. L. Junkins, and J. E. Hurtado, "A robust nonlinear system identification algorithm using orthogonal polynomial network," in *Proceedings of 15th AAS/AIAA Spaceflight Mechanics Conference*, American As-

tronautical Society. Copper Mountain, Colorado: AAS Publications, January 2005.

[8] R. R. Craig and A. J. Kurdila, *Fundamentals of Structural Dynamics*, 2nd ed. Hoboken, New Jersey: John Wiley and Sons, Inc., 2006.

[9] H. Baruh, *Analytical Dynamics*. New York, New York: McGraw-Hill Companies, 1999.

[10] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Inc., 2003.

[11] J.-N. Juang and R. S. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[12] G. Strang, *Linear Algebra and Its Applications*, 3rd ed. San Diego, California: Thompson Learning, Inc., 1988.

[13] S. Leon, *Linear Algebra with Applications*, 6th ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc., 2002.

[14] P. Singla, "Multi-resolution methods for high fidelity modeling and control allocation in large-scale dynamical systems," Ph.D. dissertation, Texas A&M University, College Station, Texas, May 2006.

[15] J. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*. Boca Raton, Florida: CRC Press LLC, 2004.

[16] P. A. Iannou and J. Sun, *Stable and Robust Adaptive Control*. Upper Saddle River, New Jersey: Prentice-Hall, Inc., 1995.

[17] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, "Past, present and future of nonlinear system identification structural dynamics," *Mechanical Systems and Signal Processing*, vol. 20, pp. 505–592, 2006.

APPENDIX A

VARIABLE DEFINITIONS

Table XII.: Variable Definitions

Variable	Definition	First Introduced
Roman Letters		
A_L	Linear State Space Matrix	Equation 4.9, 5.1
B_L	Linear State Space Matrix	Equation 4.10, 5.1
$\boldsymbol{b}$	Body axes	Figure 6
C	Rayleigh Damping Matrix	Equation 2.10
C_θ	Damping coefficient for Rotational eom	Equation 3.15
C_L	Linear State Space Matrix	Equation 4.11, 5.1
c	Damping Coefficient for Spring-Mass-Damper	Equation 4.14
$\mathcal{D}_{NL}$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.55
$\mathcal{D}_L$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.54
D_L	Linear State Space Matrix	Equation 4.12, 5.1
d_1, d_2	Integer constants, determine the size of the Hankel matrix	Equation 4.5
E	Modulus of Elasticity	Equation 2.2
$\boldsymbol{E}_j$	Matrix used in ERA Formulation	Equation 4.13

Table XII.: *continued*

Variable	Definition	First Introduced
$\mathcal{E}_F$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.56
$\mathcal{E}_G$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.57
$\boldsymbol{e}$	Error between state vector estimated using Least Squares and using the GLO-MAP function estimator	
F	Third-Order Quadratic Tensor of Nonlinear Terms	Equation 3.34
G	Fourth-Order Quadratic Tensor of Nonlinear Terms	Equation 3.35
g	Estimated nonlinear function	Equation 5.5
H	Quadratic Mass Matrix	Equation 3.30
$\boldsymbol{H}$	Hankel Matrix	Equation 4.5
h	Grid Size, distance between grid points in a given dimension	
I	Second Moment of Inertia of Bending Section	Equation 2.2
i	Generic looping variable or subscript	
$\hat{J}$	Moment of Inertia of Spacecraft Hub/Appendage Combination	Equation 3.2
J_{hub}	Moment of Inertia of Spacecraft Hub	Equation 3.2

Table XII.: *continued*

Variable	Definition	First Introduced
j	Generic looping variable or subscript	
K	Stiffness Matrix	Equation 2.6, 2.8, 3.12
K_g	Fourier Coefficient Matrix for Nonlinear Parameter Identification	Equation 5.18
$\mathcal{K}_{NL}$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.51
$\mathcal{K}_F$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.52
$\mathcal{K}_G$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.53
$\mathcal{K}_L$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.50
k	Generic looping variable or subscript	
k	Spring Stiffness for Spring-Mass-Damper	Equation 4.14
L	Lagrangean	
l	Generic looping variable or subscript	
M	Mass Matrix	Equation 2.6, 2.7, 3.9
$\mathcal{M}_{NL}$	Coefficient Matrix for Nonlinear Rotational Spacecraft eom	Equation 3.48
m	Mass of block in Spring-Mass-Damper	Equation 4.14
N	Rotation and Flexion Coupling	Equation 3.10

Variable	Definition	First Introduced
$\mathcal{N}_{NL}$	Coefficient Vector for Nonlinear Rotational Spacecraft eom	Equation 3.49
n	Number of Assumed Modes included in the Approximation or Number of states in estimation	Equation 2.1, 4.8
P	Positive definite solution to the Algebraic Ricatti Equation	Equation 5.18, B.6
$\boldsymbol{p}$	Distance from spacecraft central axis to differential beam length	Figure 6
Q	Matrix for Algebraic Ricatti Equation	Equation B.6
q	Generalized coordinates for Assumed Modes method	Equation 2.1
R	Radius of Spacecraft Hub	Figure 6
r	Generic subscript	
T	Kinetic Energy	Equation 2.2
T_{flex}	Kinetic Energy of the Spacecraft Flexible Appendage	Equation 3.1
T_{hub}	Kinetic Energy of the Spacecraft Hub	Equation 3.1
T_{NL}	Nonlinear Terms of Kinetic Energy, Quadratic Modes	
T_L	Linear Terms of Kinetic Energy, Quadratic Modes	

Table XII.: *continued*

Variable	Definition	First Introduced
T	Transformation Matrix	Equation 4.3
t	Time	
$\boldsymbol{U}$	Singular Value Decomposition, Orthonormal Matrix	Equation 4.6
$\boldsymbol{U}_n$	Singular Value Decomposition, Truncated Orthonormal Matrix	
u	Longitudinal displacement of generic mass element	Equation 3.17, 3.23
$\boldsymbol{u}$	Control vector for State Space Realization	Equation 4.1
$\boldsymbol{V}$	Singular Value Decomposition, Orthonormal Matrix	Equation 4.6
$\boldsymbol{V}_n$	Singular Value Decomposition, Truncated Orthonormal Matrix	
V	Potential Energy	Equation 2.2
v	Rotational Control Torque	Equation 3.14
w	Weight function vector for GLO-MAP Estimation	Equation 5.15, 5.20
X_{0L}	Linear State Space Initial Condition Vector	Equation 4.10
x	Distance along transverse axis of flexible beam	Equation 2.1
$\boldsymbol{x}$	State space state vector	Equation 4.1

Table XII.: *continued*

Variable	Definition	First Introduced
$\hat{\boldsymbol{x}}$	Best Estimate of Hidden State Vector	Equation 5.3
$\boldsymbol{x}_g$	GLO-MAP estimated state space state vector	Equation 5.5
y	Transverse deflection of flexible beam	Equation 2.1
$\boldsymbol{y}$	Output vector for State Space Realization	Equation 4.1
$\boldsymbol{y}_g$	GLO-MAP estimated state space output vector	Equation 5.5
z	Transformed state vector	Equation 4.4

Greek Letters

α	Rayleigh damping coefficient	Equation 2.10
β	Rayleigh damping coefficient	Equation 2.10
Γ	Convergence Parameter Matrix for Nonlinear Function Estimation	Equation 5.18
γ	Constant multiple of Convergence Parameter Matrix	
δ	Substitution operator for index notation	
$\boldsymbol{\delta}$	Deflection of generic mass element	Equation 3.17, 3.27
ε	Nonlinear spring constant	Equation 4.17
ζ	Damping coefficient	Equation 2.14
η	Transformed generalized coordinates	Equation 2.15
Θ	Modal or eigenvector matrix	Equation 2.11
$\boldsymbol{\theta}$	Eigenvector	Equation 2.10

Table XII.: *continued*

Variable	Definition	First Introduced
$\dot{\theta}$	Rotation Rate	Figure 6
κ	Order of a generic basis function	Equation 5.23
Λ	Eigenvalue matrix	Equation 2.11
λ	Eigenvalue	Equation 2.10
ξ	Local Coordinate for Adaptive Function Estimation	Equation 5.13
ρ	Beam Density	Equation 2.2
Σ	Singular Value Decomposition, Singular Value Matrix	Equation 4.7
Σ_n	Singular Value Decomposition, Truncated Singular Value Matrix	Equation 4.8
σ	Singular Values	Equation 4.8
Φ	Basis Function Matrix for GLO-MAP Estimation	Equation 5.15, 5.23
ϕ	Space-dependent function for Assumed Modes method	Equation 2.1
φ	Basis function for GLO-MAP Estimation	Equation 5.23
ψ	Quadratic assumed mode for Assumed Modes method	Equation 3.25
$\boldsymbol{\psi}$	Vector of polynomial combinations from GLO-MAP polynomial estimation	Equation 5.15
ω_n	Natural frequency	Equation 2.13

APPENDIX B

DERIVATION OF COEFFICIENT ADAPTATION LAWS FOR NONLINEAR

FUNCTION ESTIMATION

In Chapter V, the nonlinear parameter identification algorithm estimated the nonlinear function $g(\boldsymbol{x}_g)$ using the product of a Fourier coefficient matrix and a polynomial function estimation. Adaptive estimation was employed to obtain final values for these coefficients. The estimated control influence matrix B_g was also adaptively estimated. Adaptation laws were presented in that chapter, and they will be formally derived here according to Reference [7]. This derivation uses slightly different notation than the reference, so the reader is urged to use caution when comparing the two derivations.

First, define the error between the best estimate $\hat{\boldsymbol{x}}$ and the GLO-MAP estimate $\boldsymbol{x}_g$ of the hidden state vector.

$$\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}_g \tag{B.1}$$

Taking a derivative and substituting accordingly gives the following.

$$\dot{\boldsymbol{e}} = A_L\boldsymbol{e} + (B_t - B_g)\boldsymbol{u} + (K_t - K_g)\boldsymbol{\psi}(\boldsymbol{x}_g) + \epsilon \tag{B.2}$$

In this equation, B_g is the GLO-MAP estimate of the true control influence matrix, B_t. The unknowns will be consolidated as is typical in adaptive control.

$$\dot{\boldsymbol{e}} = A_L\boldsymbol{e} + \tilde{B}\boldsymbol{u}(t) + \tilde{K}\boldsymbol{\psi}(\boldsymbol{x}_g) + \epsilon \tag{B.3}$$

To derive the adaptive laws for B_g and K_g, evaluate the following candidate Lyapunov function

$$\mathcal{V} = \frac{1}{2}\boldsymbol{e}^T P\boldsymbol{e} + \frac{1}{2}\mathrm{Tr}(\tilde{B}\Gamma_1\tilde{B}^T) + \frac{1}{2}\mathrm{Tr}(\tilde{K}\Gamma_2\tilde{K}^T) \tag{B.4}$$

where P is a positive definite symmetric matrix, and Γ_1 and Γ_2 are constant matrices with full rank. After taking the time derivative, the following equation is revealed.

$$\dot{\mathcal{V}} = \frac{1}{2}\boldsymbol{e}^T \underbrace{\left(P\mathrm{A}_L + \mathrm{A}_L^{\mathsf{T}}\mathrm{P}\right)}_{\equiv -Q}\boldsymbol{e} + \boldsymbol{e}^T P \left(\tilde{\mathrm{B}}\boldsymbol{u}(t) + \tilde{\mathrm{K}}\boldsymbol{\psi}(\boldsymbol{x}_g) + \epsilon\right) + \tag{B.5}$$
$$+ \mathrm{Tr}\left(\tilde{\mathrm{B}}\Gamma_1\dot{\tilde{\mathrm{B}}}\right) + \mathrm{Tr}\left(\tilde{\mathrm{K}}\Gamma_2\dot{\tilde{\mathrm{K}}}^{\mathsf{T}}\right)$$

The matrix Q is defined to satisfy the algebraic Ricatti equation.

$$P\mathrm{A}_L + \mathrm{A}_L^{\mathsf{T}}\mathrm{P} = -Q \tag{B.6}$$

Substituting this matrix and gathering terms leads to the following equation.

$$\dot{\mathcal{V}} = -\frac{1}{2}\boldsymbol{e}^T Q\boldsymbol{e} + \mathrm{Tr}\left(\tilde{\mathrm{B}}\left[\Gamma_1\dot{\tilde{\mathrm{B}}}^{\mathsf{T}} + \boldsymbol{u}\boldsymbol{e}^{\mathsf{T}}\mathrm{P}\right]\right) + \mathrm{Tr}\left(\tilde{\mathrm{K}}\left[\Gamma_2\dot{\tilde{\mathrm{K}}}^{\mathsf{T}} + \boldsymbol{\psi}(\boldsymbol{x}_g)\boldsymbol{e}^{\mathsf{T}}\mathrm{P}\right]\right) + \boldsymbol{e}^{\mathsf{T}}\mathrm{P}\epsilon \tag{B.7}$$

Set the terms in square braces equal to zero and solve for the adaptation laws. This combined with the knowledge that B_t and K_t are constants yields adaptation laws for the unknown constants.

$$\dot{\tilde{\mathrm{B}}}^{\mathsf{T}} = \dot{\mathrm{B}}_g^{\mathsf{T}} = -\Gamma_1\boldsymbol{u}\boldsymbol{e}^T P \tag{B.8}$$
$$\dot{\tilde{\mathrm{K}}}^{\mathsf{T}} = \dot{\mathrm{K}}_g^{\mathsf{T}} = -\Gamma_2\boldsymbol{\psi}(\boldsymbol{x}_g)\boldsymbol{e}^T P \tag{B.9}$$

Substituting these equations in to the derivative of the Lyapunov equation produces a negative definite $\dot{\mathcal{V}}$.

$$\dot{\mathcal{V}} = -\frac{1}{2}\boldsymbol{e}^T Q\boldsymbol{e} + \epsilon^T P\boldsymbol{e} \tag{B.10}$$

This function is negative definite if the norm of the error bounds the remaining terms. For a formal proof of convergence for the adaptation law, see Reference [7].

VITA

Stefanie Rene' Beaver was born and raised in Texas. She began her undergraduate studies at Texas A&M University in August 2001 and received her Bachelor of Science degree in Aerospace Engineering from Texas A&M University on May 13, 2005. In August 2005, she began her graduate studies under the advisement of Dr. John E. Hurtado at Texas A&M. Stefanie graduated with a Master of Science in Aerospace Engineering from Texas A&M University on August 10, 2007.

Ms. Beaver may be reached at sbeaver@gmail.com or by contacting Dr. John E. Hurtado, Department of Aerospace Engineering, TAMU 3141, Texas A&M University, College Station, TX- 77843.