A Formulation for Updating Finite Element Models Through Consistent Use of Laser Vibrometer Data

by

Eric ten Siethoff

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Robert L. West, Chair
Charles E. Knight
William R. Saunders

April 28, 1998

Blacksburg, Virginia

Keywords: finite element, updating, laser Doppler vibrometer

A Formulation for Updating Finite Element Models Through Consistent Use of Laser Vibrometer Data

Eric ten Siethoff

(ABSTRACT)

This thesis suggests a formulation for updating physically meaningful parameters in analytical finite element(FE) models using scanning laser Doppler vibrometer(SLDV) dynamic response data. The update formulation is demonstrated in several computer simulations.

The formulation is the result of incorporating an analytical FE model into an experimental model. The experimental model efficiently utilizes SLDV data to fully exploit the instrument's capability to automatically make measurements at many locations. The data in the experimental model is posed in a manner consistent with an analytical FE model's representation for harmonic response, simplifying comparison between the two. The experimental model, which uses finite element shape functions as a basis for a least squares fit to the data, can be solved to give a velocity field based only on that data. The function resulting from inserting the analytical model into the experimental model is an expression of the prediction error of the FE model as compared to the test data. This function is minimized using a quasi-Newton optimization routine, reducing the error and resulting in an updated model.

Computer simulations of the update algorithm indicate that:

1. Analytically supplied derivatives and variable scaling are required by the optimization routine to consistently converge,

2. The percentage error of updated parameters falls within two standard deviations of the data's percentage error,

3. Error in the position of the laser results in the update algorithm's failure, and,

4. Error in the parameters not included in the update will appear as error in the updated parameters' solution.

To my family

Acknowledgments

First and foremost I'd like to thank Dr. Robert L. West for his many patient hours serving as my advisor. He provided direction throughout my research and, as I wrote my thesis, ensured that I maintained the correct focus. He helped me better understand my work by providing his vision of my contribution to the field of model updating. Most importantly, he provided encouragement and direction at times when I was unsure of myself.

I'd also like to thank Dr. Charles E. Knight and Dr. William R. Saunders for serving on my committee. Along with Dr. Larry D. Mitchell, who attended my defense, they provided important input on the implications of my research. The different points of view they brought lead me to insights I could not have made on my own.

I wish to thank David Coe and Mike Stafne for providing a wealth of information on the operation of the scanning laser Doppler vibrometer and on the details of Experimental Spatial Dynamics Modeling. Finally, I'd like to thank all the graduate students who let me bounce ideas off of them and who gave their candid evaluation of those ideas.

Eric ten Siethoff

Virginia Polytechnic Institute and State University
April 28, 1998

Contents

List of Tables

List of Figures

Nomenclature

A	cross-section area
$\mathbf{A}$	matrix of regressor variable values
$a_0,\ a_1,\ a_2$	linear system properties
B	boundary operator
b	beam cross-section width
$\mathbf{b}$	vector of regression parameters
$\mathbf{C}$	damping matrix
$\mathbf{c}$	experimental model data vector
D	problem domain
d_i	finite element degree of freedom i
d_k	search direction k
$\mathbf{d}$	finite element node displacements
E	elastic modulus
ESDM	Experimental Spatial Dynamics Modeling
$\mathbf{e}$	vector of error values
e_i	error between model prediction and sample i
$\mathbf{e_l}$	unit vector in direction of incident laser beam
F	frequency domain representation of the force
FE	Finite Element
FFT	Fast Fourier Transform
FRF	Frequency Response Function

f	excitation
	objective function
$\mathbf{f}$	vector of sampled force values
	finite element force vector
g	inequality constraint equation
h	beam cross-section height
	equality constraint equation
h_e	element length
IMAC	Inverse Modal Assurance Criteria
i	sample indice
	$\sqrt{-1}$
$\mathbf{K}$	stiffness matrix
k	optimization routine iteration step indice
L	linear differential operator
	beam length
$\mathbf{L}$	matrix of test data
$\mathbf{l}$	vector of test data
$\mathbf{M}$	mass matrix
N_i	shape function
$\mathbf{N}$	shape function matrix
$\mathbf{N_R}$	transformed shape function matrix
$\mathbf{Q}$	experimental model data matrix
$\mathbf{q}$	generalized force vector
$\mathbf{R}$	shape function transformation matrix
S	domain boundary
SLDV	Scanning Laser Doppler Vibrometer
S/N	signal-to-noise ratio
t	time
$\mathbf{u}$	displacement field vector
$u,\ v,\ w$	displacements in the x-, y-, and z-coordinate directions, respectively

$\dot{\mathbf{u}}_s$	surface velocity vector
V	frequency domain representation of v_L
v_L	velocity magnitude measured by the laser
x	response
	coordinate direction
$\mathbf{x}$	design variable vector
x_l	assumed laser x-position
$\mathbf{Y}$	finite element mobility matrix
$\mathbf{y}$	vector of direction cosines
$\mathbf{Z}$	finite element impedance matrix
z_l	assumed laser z-position

Greek Symbols

α	mass proportional damping constant
α_k	step size k
β	stiffness proportional damping constant
$\beta_1,\ \beta_2$	regression parameters for force
$\gamma_1,\ \gamma_2$	regression parameters for measured velocity magnitude
$\Delta\mathbf{x_k}$	change in design variable vector for step k
ε	random error
θ	beam cross-section rotation
ξ	general element coordinate
Π	update formulation model
σ	standard deviation
ϕ	phase angle
	field variable for a differential equation
	response vector
χ^2	experimental model
$\Psi_x,\ \Psi_y,\ \Psi_z$	direction cosines
ψ	basis function

ω	circular frequency

Subscripts

ae	added error
exp	experimental
Im	imaginary part of the frequency domain representation
$global$	global matrix/vector
na	neutral axis
r	relative to the reference input force
Re	real part of the frequency domain representation
ref	relative to the reference angle, ωt
s	surface

Superscripts

., ..	first and second derivatives with respect to time
$\wedge$	estimate or predicted value
T	transpose
$-$	amplitude
-1	inverse
$*$	optimum value

Chapter 1

Introduction

1.1 Statement of Need

In engineering, there is a need for accurate numerical models. Computational fluid dynamics for new aircraft, the finite element method for structural analysis, and finite difference programs for heat transfer are examples of numerical methods that have become essential to the design process. In these applications, numerical models provide predictions of design performance which facilitate making decisions early in the design process, saving money and ultimately producing a better product. Numerical models allow the exploration of the design space that is prohibitively expensive to conduct using manufactured prototypes.

The users of numerical models may require response predictions that are quantitatively accurate, or they may only need a qualitative assessment of trends. The degree of complexity of the system will limit what numerical model users can expect. For simple systems, numerical models can illuminate even subtle aspects of the response, giving quantitatively accurate response estimates. For more complex systems, a qualitative assessment of the response may only be possible. Whether looking for trends or accurate numbers in the response, the users will always want some confidence that the model predicts how the actual system will respond.

The goal of numerical models is to allow the examination of different configurations and different design points. It is necessary for the models to include parameters which are directly related to the changes in the design, so that changes in the design can be reflected by changes in the parameters. The values of these parameters will be based on assumptions made about the system. Sometimes these assumptions are appropriate and there is no significant error introduced by making them. Other times, these assumptions are made entirely because there is great uncertainty in the true value of the parameter. If the parameter's value is incorrectly assumed, the level of error may be unacceptably high, rendering the numerical model useless for qualitative or quantitative examination of the system's response.

Gathering test data from an existing system makes information available which may be able to clear up some uncertainties in the assumptions. Updating is defined as using experimental data(inputs and outputs) from an actual system to *improve* estimates for parameters in a numerical model by comparing the experimental result with the model's predictions. *Improvement* is defined as bringing the model's prediction closer to how the actual system will respond. Assuming the experimental data is accurate, this involves comparing the data and the model predictions. The measure of how much the model is *improved* depends on a criteria involving the user's needs and the system's complexity, factors that will change from situation to situation.

The primary numerical modeling tool for structural analysis is the Finite Element Method. A level of detail is demanded of some of today's finite element(FE) models for dynamic response analysis that can only be achieved through updating. Boundary conditions, geometry, and material properties are examples of parameters that can have a large effect on the response predicted by the FE model. These parameters are subject to uncertainties, which leads to errors in the model. There is a need for FE models developed specifically to be better predictors of dynamic structural response. It is important to develop these models within a consistent framework which brings together experimental vibration data and FE models. This thesis proposes a consistent method to update parameters in a finite element model used for vibration analysis using experimental data gathered with a Scanning Laser Doppler Vibrometer(SLDV).

1.2 Research Hypothesis and Goals

The hypothesis of the research presented in this thesis is that a general framework can be developed for updating a FE model for dynamic response analysis using experimental data gathered by a SLDV. Furthermore, the updated model will be an *improvement* in the sense that it will predict the actual system response more closely, quantitatively or qualitatively, depending on the system complexity and the user's needs.

The goals of this research are:

- To develop an algorithm to update finite element models that utilizes experimental data,
- To develop the formulation so that the data and analytical model are posed in a consistent fashion,
- To make the development general enough that the formulation and the algorithm can be extended and applied to many structures, with different types of finite elements, and different update parameters, and,
- To demonstrate the feasibility of the algorithm through simulations.

The research presented in this thesis meets these goals by developing an experimental model to represent the test data, an analytical model, and a method to compare them. In this algorithm, the experimental data is contained in an expression of the squared error between the measurements of structural velocities made by the laser and a function fit to these values. Included in the equation's development is a transformation from the outer surface of the structure, where the laser measurements are made, to the midplane, where the FE nodes are located [1]. This equation is an experimental model which can be solved independently to give a purely experimental solution of the structure's operating shape during a vibration test. A finite element model of the structure serves as the analytical model and is inserted into the experimental model

[1] The assumption is that finite elements which model response at the midplane or the neutral axis will be used. If not, the transformation may not be needed.

to update the parameters in the FE model. An unconstrained optimization procedure is used on the resulting equation to minimize the sum of the squared error and arrive at the *best* values for the parameters. Thus, the *best* values for the parameters are defined as those values that minimize the error between the experimental data and a candidate model's predictions in a least squares sense. This minimum is different from what would be attained if the FE model was not included. The presence of the FE model constrains the optimization to only find a solution that is consistent with the deformations allowed by the mechanics of the model. By updating the parameters directly with experimental data, the analytical model and data are consistent with a minimum amount of data processing, and the updated model will be an *improvement* relative to the initial model's prediction.

1.3 Key points and Thesis Scope

The key points in the development of this algorithm are:

1. Understanding the relationship between a structure's motion and what the laser measures,

2. Processing this measurement into a form useful for the update algorithm,

3. Developing an experimental model to hold the information from the vibration test,

4. Developing an analytical FE model that is consistent with the conditions of the test and the experimental model,

5. Choosing parameters to update in the FE model,

6. Formulating a consistent method to bring the FE model and the experimental model together,

7. Calculating the updated model through optimization, and,

8. Validating the updated model.

This thesis will cover all the objectives listed here except for numbers five and eight. Choosing the parameters to update is very difficult and requires physical insight into the specific problem. This thesis will not attempt to develop a general framework to choose parameters that are applicable to all problems, but instead covers some of the issues involved. Validating the updated model depends heavily on what the user defines as *improvement*. A general rule on how to perform model validation is not given here. Instead, some validation techniques are presented. The subject of accurately finding the laser's position with respect to the structure before a test is not covered in this thesis; however, it is important and can adversely affect the experimental results, which is demonstrated in this thesis. No attempts to update a model of an actual structure with test data were made. The research presented in this thesis constructs the framework needed to perform this next step.

1.4 Thesis Structure

Model updating is a subset of system identification. System identification is defined as recording the input and output signals from a system in order to infer a model. System identification is often used in control applications where the model is produced on-line and used in the control algorithm. While model updating can be used in this capacity, more often it is used in design applications where an updated model is used as the starting point for exploration of the design space. Figure 1.1, which is based on a figure by Ljung (1987), shows the parts of the system identification process and their relationship to one another. Since system identification and model updating are so similar, the model updating process has the same steps as those illustrated in the figure. This thesis is organized in a similar manner as Figure 1.1, with each chapter developed around an element or elements.

Prior knowledge is the starting point for the entire process. It is the sum of the experience and theoretical knowledge brought to the process and is a prerequisite to several of the steps. The impact of prior knowledge is large and, as Figure 1.1 shows, it will appear in multiple chapters.

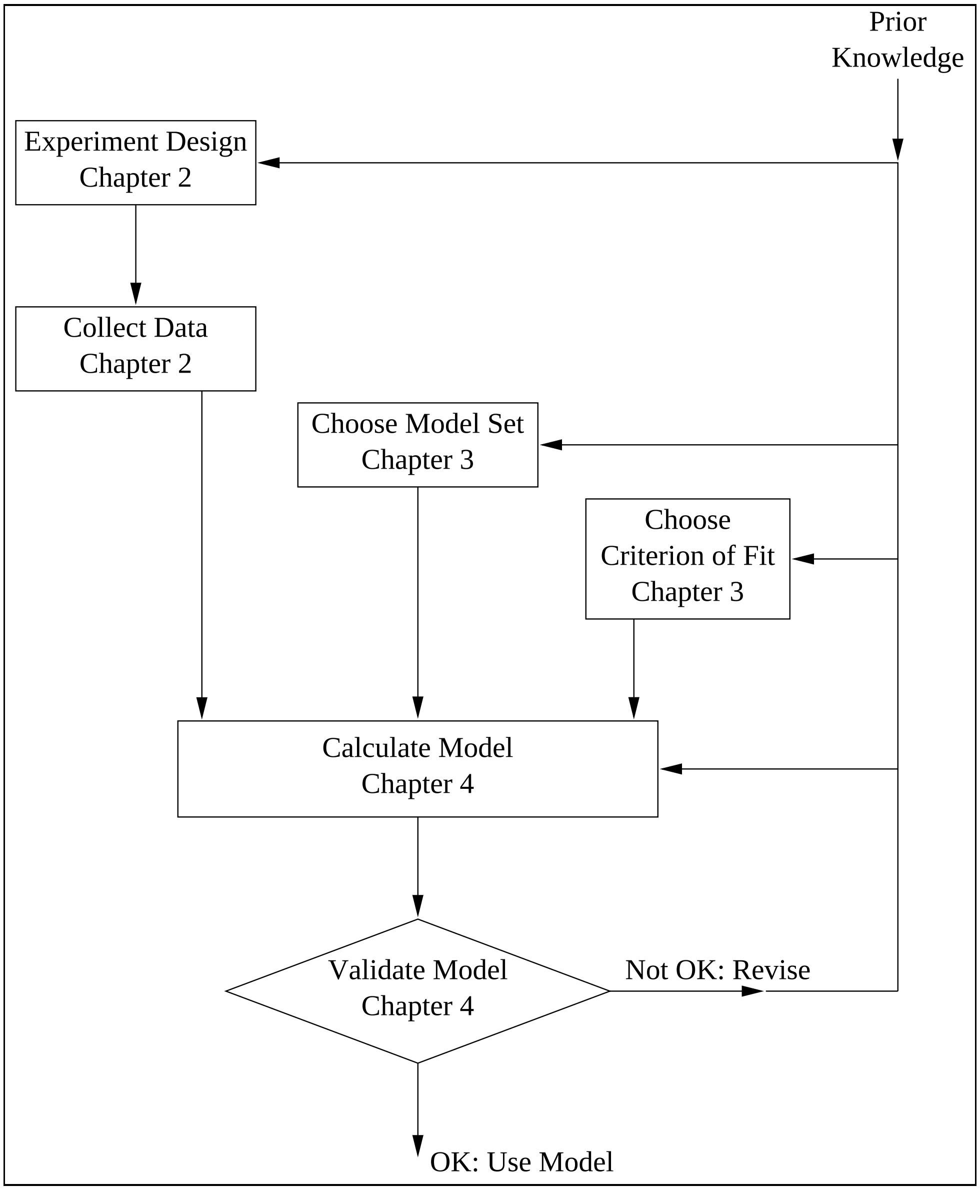

Figure 1.1: The Elements of the System Identification Process

Experimentation can be costly to repeat, so it is worthwhile to design an experiment so that the correct information important to updating the model is gathered the first time. Experiment design consists of deciding which signals to measure(input and output), choosing the inputs(if possible), and exercising control over an experiment so that the data gathered is maximally informative. Prior knowledge helps decide what data will be the most useful to gather and how to gather it. If the experiment is going to be performed in a lab, the inputs can be chosen and the environment can be controlled so that unwanted inputs are minimized. If the test occurs outside a lab, the inputs usually cannot be controlled, and sometimes not even measured. In either case, time spent planning before the experiment is performed will ensure the data gathered is the most useful. After the experiment is designed, the next step is to perform the test and to gather the data. In this step, it is important to gather the data and store it correctly. Chapter 2 will cover the experiment design and some aspects of gathering data using the scanning laser Doppler vibrometer. Only those aspects of experiment design essential to the development of this thesis are covered. This chapter will show what the laser measures and how the measurement is processed into a form useful for the update algorithm.

In choosing the model set to use with the data, the first choice is whether to use a parametric or nonparametric model. While both types of models have quantities that can be referred to as parameters, here parametric models refer to those models whose parameters are based on a physical relationship or other well-established law. Nonparametric models refer to models whose parameters are there solely to fit data. Nonparametric models have the advantage that they are not necessarily constrained by any rules and that increasing the order of the model involves just adding more parameters. A linear regression to fit experimental data is an example of a nonparametric model. While nonparametric models like linear regressions benefit from prior knowledge by influencing the order of model chosen, parametric models benefit from prior knowledge of plausible values for the actual parameters. The first estimates for the parameters can be based on what was known before and the final answers can be compared to these initial estimates. The final result benefits from the confidence that it is based on accepted theories.

In this thesis, the experimental model is a nonparametric model. The analytical model is produced using the finite element method, hence it is a parametric model. Since this algorithm will always use the finite element method to produce analytic models, the model set choice is to decide what parameters of the model to update: plate thickness, boundary conditions, damping, etc. Choosing the parameters requires physical insight and the use of engineering judgement. This choice requires great care because the wrong choice can lead to a model that is reconciled with the experimental data but does not capture the qualities that were misrepresented in the analytic model in the first place. Chapter 3 will briefly discuss the issues involved with choosing parameters.

Along with choosing the model set, a method to compare the analytic model to the experimental data must be decided on. There must be some method to ensure that the analytic model can compare to experimental data taken at specific spatial locations and reflect other conditions of the experiment. The formulation must be developed to decide what is the *best* model. Chapter 3 covers the issues involved in choosing the criterion of fit between the FE model and the experimental data. The experimental data is held in a nonparametric model which could be solved without the update procedure. A finite element mechanics model for harmonic response is developed which is used in the update procedure. The relationship between the FE model and the experimental model is explained. The different optimization formulations considered for the algorithm and the philosophies behind each are discussed. Finally, the justification for the formulation chosen in this thesis is given.

As Figure 1.1 shows, once the test data has been collected, the model set has been chosen, and the criterion of fit decided upon, the next step is to bring these elements together to calculate the updated model. In this algorithm, calculating the updated model involves running an optimization routine to find the *best* model. Once the *best* model has been selected, it must be validated. This entails assessing how the updated model relates to the observed data, to prior knowledge about the system, and to the model's intended use. Numerical models are never the perfect description of the system. The goal is to develop the model for the purpose of getting the job done. This is key in deciding if the model is *good enough.* Chapter 4 gives an overview of the optimization routine and the issues involved in making it work. This chapter also shows some of the techniques to validate the model. If the updated model is shown to adequately represent the actual system, the model can then be put to its intended use. If not, the knowledge gained in the first attempt to update the model will guide decisions made in subsequent attempts. For this algorithm, these decisions will be made mainly in the areas of experiment design and in the choice of parameters in the FE model to update.

The final two chapters do not include steps of the system identification process, illustrated in Figure 1.1. Chapter 5 demonstrates the algorithm through three simulations. A cantilever beam is used as a test case to show how the algorithm performs in the face of several types of error: random error in the experimental data, a systematic error made while performing the experiment, and the effect of choosing the wrong parameter to update in the FE model. The Matlab code for these simulations is located in the Appendix. Chapter 6 lists the conclusions drawn from the work in this thesis. The chapter also includes recommendations for work to extend or improve the algorithm presented here.

1.5 Literature Review

There is a large body of literature available on the subject of finite element model updating. Friswell and Mottershead (1995) provide a comprehensive overview that illustrates many of the different techniques and issues involved in updating a finite element model. For each of the topics covered in this book, the original papers are cited so that further investigation is possible.

The authors divide the various model update techniques into two groups based on the form of the experimental data that they utilize: those that use modal data and those that use Frequency Response Function(FRF) data. Friswell and Mottershead further divide the model updating techniques into those that update parameters and those that update the elements of the stiffness and mass matrices directly. Directly updating the matrices results in models that match the test data very closely, but the mass and stiffness matrices lose any physical meaning they initially had. Updating models by changing parameters allows the use of physically based parameters, which will still have meaning following the update.

Modal analysis distills the information from a vibration test into the system's natural frequencies, the mode shapes associated with those frequencies, and a damping value for each mode. Only the natural frequencies and mode shapes are used to compare model predictions and measurements. Damping is usually not used in comparisons since modeling damping presents difficulties for finite element models. Proportional damping is often used, but mostly because it is easy to implement and because it results in a symmetric eigenvalue problem(with real valued eigenvalues and eigenvectors). In this respect, the updating techniques using modal data benefit from the fact that damping does not have to be included in the finite element model. In contrast, model updating techniques that utilize FRF data require damping to be included in the model since its effects are always present in the response. Simulations of FRF data can be accomplished by solving the model directly or by solving for the mode vectors and natural frequencies and performing mode superposition. The techniques simulating FRF data by direct solution benefit from the fact that it is much easier to calculate the sensitivity of the modeled response to a change in a physical parameter than to calculate the sensitivity of the natural frequencies and mode shapes the model predicts. This makes it much easier to use physical parameters in model updating techniques that simulate FRF data directly as compared to methods using modal data.

When measuring dynamic response with accelerometers, a relatively small number of locations will be sampled compared to the number of nodes in a typical finite element model of a structure. Friswell and Mottershead (1995) address the incompleteness of the measured data due to the limited number of measurement locations. They show techniques to reduce the analytic mass and stiffness matrices or to expand the test mode shapes. In any of the methods, the goal is to have consistent degrees of freedom— locations where experimentally measured motions can be directly compared to analytic predictions.

The book also discusses the selection of physical parameters and the authors make several recommendations. One recommendation is to keep the number of parameters small since this should provide better results. The parameters should be chosen to correct a recognized uncertainty in the model, they shouldn't be there simply because they can be. Lastly they recommend that the data should be sensitive to the parameters chosen so that it carries the information necessary to effectively update the parameters.

Friswell and Mottershead (1995) show the importance of several areas in model updating. The first is the choice between using modal data or FRF data. Secondly is whether to use direct updates or to use parameters. The authors lean toward the use of parameters since they allow quantities with physical meaning to be used. The majority of the book is dedicated to modal methods. This reflects the fact that modal analysis is so popular as a test analysis technique and most of the research in model updating has utilized modal data. However, model update techniques using FRF data lend themselves to the use of physical parameters more easily than techniques using modal data. Partly for this reason, the model update method presented in this thesis uses frequency domain data with physical parameters. Friswell and Mottershead also illustrate the need to expand the measured data or contract the model's degrees of freedom so that there is a consistent basis to compare model predictions to test measurements. The algorithm derived in this thesis provides a consistent and automatic method to bring experimental and analytical degrees of freedom together for comparison that does not rely on artificially expanding the test model or contracting the FE model.

Montgomery (1994) laid the groundwork for modeling dynamic response using finite element shape functions, coined Experimental Spatial Dynamic Modeling(ESDM). He showed the method used to gather velocity data using the laser Doppler vibrometer and the linear regressions to process the time series data. He showed the registration process used to find the laser beam direction and location for each sample location so that each sample's position relative to the structure's coordinate system is known. Montgomery's work demonstrated the validity of the ESDM method. This algorithm is based on that work, but it differs on two counts. First, in this thesis the mesh is not located at the surface, but at the midplane. Secondly, Montgomery's algorithm is nonparametric. The nodal velocities are parameters whose values are chosen solely to fit the data. In the algorithm developed in this thesis, a mechanics model for the structure imposes constraints on the nodal velocities. The velocities can only take values allowed by the physical relationships prescribed by the mechanics model.

Venter (1995) presented a study of the sensitivity of the error function, the nonparametric experimental model used in ESDM, with respect to structure parameters. Expressions were also developed for the sensitivity of the data to error in the laser position and orientation. Errors in laser position and orientation are also explored in this thesis; however, only a systematic error in the laser's position is simulated.

Lindholm (1996) demonstrated a model updating technique that used FRF data. He updated physical parameters in finite element models within a Bayesian statistics framework. After the models were updated, statistics about the updated parameters provided information about the quality of the new values. This dissertation showed how to rigorously incorporate prior knowledge into the model updating process. Lindholm also presented some procedures to validate that the updated model is an improvement. He demonstrated the importance of parameter selection and how accurate prior information about the parameters will improve the resulting updated model. While parameter selection techniques are not presented in this thesis, it is recognized that this algorithm will not produce good results if the parameters are not chosen carefully. Also, the process presented in this thesis emphasizes the importance of prior knowledge to several of its elements. There is a short discussion in this thesis on methods to validate updated models, but there is still a large amount of work to be done to develop a methodical way of testing the performance of updated models.

The research presented in this thesis brings together the frameworks presented by Lindholm and Montgomery, respectively. A technique to update finite element models utilizing prior knowledge, like Lindholm advocates, is presented. This model updating technique utilizes the experimental method developed by Montgomery. The work in both dissertations shows how the formulation presented here could be fully implemented.

Chapter 2

Processing Test Data

The algorithm presented in this thesis to update finite element models requires experimental data taken while the structure is tested. The primary results of this chapter are structural velocity components measured by the laser vibrometer and force components measured by a force transducer during a test. These quantities are used by the model update algorithm in a frequency domain form that represents the time domain signals that were originally measured. This chapter describes the process involved in taking velocity and force measurements and translating these time domain samples into the frequency domain representations needed by the algorithm. The steps in the signal processing development to get a frequency domain signal representation for force and velocity that can be directly inserted into the FE formulation are to:

1. Design the experiment so that the data is the most useful. This includes choosing the method of exciting the structure. The development here shows the processing involved for periodic excitation; however, this type of excitation is not the only option. A Fourier transform of a random or transient signal produces the same information that the update algorithm needs. It will be shown that the periodic excitation is the most practical choice for this application. Other decisions about the test must be made for each situation to ensure the data is useful.

2. Derive the relationship between the quantity that the laser measures, a velocity magnitude, and the motion of a point on the vibrating structure. This is essential when measuring motion which has more than one directional component.

3. Show how raw force and velocity signals can be sampled in time and then linear regressions can be used to fit functions to represent these signals in the time domain. For both the force and laser measurements, an analog-to-digital signal converter controlled by data acquisition software is used to take the time series data needed for the regressions. Linear regressions are used instead of Fourier transforms for two reasons. The first is that, for periodic excitation, the excitation frequencies are known, so there is no need to process the data to find Fourier coefficients for frequencies where there is no activity. Secondly, for multiple frequency response, the frequencies of interest are not limited to integer multiples of the fundamental frequency as they would be with a Fourier transform.

4. Show how the time domain functions are related to the frequency domain representations and how the phase angle of the velocity function for a location on the structure is measured relative to a reference force function. Measuring the phase angle in this fashion arises automatically in the theoretical mathematics that describes dynamic systems, but the data must be manipulated to produce this result for the experimental data. It is essential that the velocity measurements at each of the sample locations on the structure all have the same reference for phase so that their phases relative to one another have meaning. The signal used for an input force to the test structure is used as this reference.

5. Generalize the regressions for the force and velocity data to the case of multiple input frequencies and multiple response frequencies. This is important since one multiple frequency test can provide the same information in less time compared to several single frequency tests, and also since factors beyond the control of the test engineer may result in multiple frequencies.

2.1 Experimental Design

Proper experiment design is necessary if the data is to be useful for updating. This includes deciding how to measure the vibrations and how to excite the structure. McConnell (1995) provides a thorough treatment of vibration testing that is a good source for information when designing a vibration test.

While there are several transducers that can be used to measure vibrations, this update method uses a scanning laser Doppler vibrometer. The unique characteristics of this instrument should be considered in designing the experiment. It does not physically contact the system like accelerometers must, so the structure being tested is not slightly different due to transducer mass. The scanning mirrors in the laser allow many locations on a structure to be sampled in one test. These samples mean that a very accurate picture of the operating shape of the vibrating structure can be inferred. It is necessary to have a method in place to automatically specify the sample locations in the structure coordinate system to practically make measurements at many points. Since the laser only reads velocities in the direction it is pointed, it is important to place the laser with respect to the structure so that it can *see* the components of a structure's velocity at every measurement location.

The experiment design also includes choosing the method of exciting the structure. An electromagnetic or hydraulic shaker in conjunction with a signal generation source can be used for several types of harmonic testing: stepped-sinusoidal, slow sine sweep, and periodic. A shaker can also be used for random excitation. An impact hammer or some method to apply and release a displacement can be used for transient excitations. If information at many frequencies is desired, the harmonic testing techniques take more time than the random and transient techniques, but they allow energy to be input to the structure at specific frequencies and produce a high signal-to-noise ratio. The harmonic techniques also allow more information to be gathered at frequencies of interest, for example near a resonance point. Table 2.1 compares the advantages and disadvantages of each excitation method for application in a test using a SLDV to gather data.

Table 2.1: Comparison of Vibration Testing Techniques for Use With the SLDV

Excitation Method	**Advantage**	**Disadvantage**
Stepped Sinusoidal and Slow Sine Sweep	Take measurements only at frequencies of interest	Need to be sure to measure only the steady state response, not any transients after a frequency shift
Periodic	Can measure at several frequencies at once	Only get information at chosen frequencies
Random and Transient	Get information across many frequencies quickly	Averaging for FRF's would take too long for number of locations SLDV samples

Periodic excitation using a shaker will be used for this update algorithm since it allows the data to include only those frequencies deemed useful for the update. Also, the high signal-to-noise ratio of the response signal due to periodic excitation should allow the laser to discern small components of a structure's velocity in the presence of a dominant component. Testing with a random signal would provide information at many more frequencies, but the time required is prohibitive. Performing the Fast Fourier Transform(FFT) with the required averages at the number of locations the SLDV is capable of sampling on a structure would take more time than is practical. While an accelerometer used with the FFT can provide information across many frequencies at a few locations, the SLDV used with periodic excitation will provide information at a few frequencies at many locations. Since, in the application presented here, the SLDV is only gathering data at a few frequencies, it is important that the frequencies are chosen carefully. Prior knowledge is useful in deciding what frequencies are important for the structure being tested. Energy should not be wasted in frequencies that aren't useful. Testing with signals that include multiple desired frequencies should result in a better updated model since more information is available. Also, the structure may encounter several known frequencies in operation, so information about the response at those frequencies is especially important.

It is also important to decide where to place the shaker attachment point to excite the structure as this will control the operating shape during the test. A force transducer will be placed between the shaker and the structure so that the actual force input to the structure can be recorded. It is important that the magnitude and location of the input force match what the structure encounters or will encounter in its operating environment. This ensures the usefulness of the test data, and ultimately the usefulness of the updated model of the structure.

2.2 Linear System Properties

For the purposes of the algorithm presented in this thesis, the vibrating structure being tested is assumed to behave linearly. This assumption is necessary since two characteristics of linear systems are used to full advantage by the algorithm. These two characteristics can be illustrated mathematically if the linear system is represented as

$$a_0\ddot{x}(t) + a_1\dot{x}(t) + a_2x(t) = f(t) \tag{2.1}$$

where $x(t)$ is the response and $f(t)$ is the excitation. The dots over the response represent derivatives with respect to time. The constants a_0, a_1, and a_2 are parameters of the system. One characteristic of a linear system is that the steady-state response to a harmonic excitation at the frequency ω will be at the same frequency. As shown by Meirovitch (1997), if $f(t) = \bar{f}\cos(\omega t + \phi_f)$, where $\bar{f}$ is the amplitude of the force and ϕ_f is the phase, then the particular solution(steady state) to Equation 2.1 is $x(t) = \bar{x}\cos(\omega t + \phi_x)$. This solution shows that the response is also harmonic and at the same frequency as the forcing function. The values for the amplitude and the phase of the response depend on the system(a_0, a_1, a_2) and the forcing function.

Meirovitch (1997) also shows another characteristic of linear systems— the response due to several forcing functions is simply the sum of the responses to each individual forcing function. If

$$f(t) = f_1(t) + f_2(t) + \cdots + f_n(t)$$

then the solution to Equation 2.1 is

$$x(t) = x_1(t) + x_2(t) + \cdots + x_n(t)$$

where $x_i(t)$ is the solution to

$$a_0\ddot{x}(t) + a_1\dot{x}(t) + a_2x(t) = f_i(t)$$

If the forcing function is $f_i(t) = \bar{f}_i \cos(\omega_i t + \phi_{f_i})$ then $x_i(t) = \bar{x}_i \cos(\omega_i t + \phi_{x_i})$. Consequently, the steady-state response of a linear system to a linear combination of harmonic functions at different frequencies(periodic excitation) is the sum of the responses at each of the frequencies. For the remainder of this document, the term *response* refers to a steady-state harmonic response.

These properties of linear systems are important to the update algorithm because they allow important assumptions to be made about the test structure's dynamic response. Assuming the structure is linear, its response to a periodic excitation can be assumed to also be periodic at the frequencies of excitation. The assumed form of the response is part of the prior knowledge used in gathering the data and modeling the response analytically. Before taking data with the SLDV, the validity of the assumptions about the linearity of the structure can be checked. Amplitude linearity can be tested by incrementally increasing the amplitude of the input, which should result in proportionally increasing response. Also, the response's frequency content can be examined to ensure that the primary components correspond to the excitation frequencies. Failure of either of these tests may indicate nonlinearities in the structure which will present difficulties.

2.3 Velocity Magnitude Measured By the SLDV

Drain (1980) supplies details of how velocity is measured by a laser Doppler vibrometer. For the development in this thesis, it is important to understand that the measured quantity is the magnitude of a vector whose direction is coincident with the ray that strikes the surface. When the laser beam strikes a moving structure at a particular point on the structure's surface, the laser *sees* the resultant of all three components of the velocity at that point. At any particular instant in time, t, the surface velocity vector, denoted $\dot{\mathbf{u}}_s$, may appear as shown in Figure 2.1. In this figure, the structure's surface lies in the x-y plane, but in general this does not have to be true and the relationships derived here are valid regardless of the orientation of the structure's surface. The velocity vector at this location on the surface is the sum of the components of the structure velocities at that point:

$$\dot{\mathbf{u}}_s = \dot{u}\mathbf{i} + \dot{v}\mathbf{j} + \dot{w}\mathbf{k} \tag{2.2}$$

A unit vector in the direction of the incident laser beam is given by

$$\mathbf{e_l} = \cos\alpha_x\mathbf{i} + \cos\alpha_y\mathbf{j} + \cos\alpha_z\mathbf{k} \tag{2.3}$$

where α_x, α_y, α_z are the angles between the positive x, y, and z coordinate directions and the laser beam. The magnitude of the projection of the structure's velocity vector onto the laser beam direction is what the laser *sees* as the structure's velocity. The magnitude, v_L, is shown at the bottom of Figure 2.1. Geometrically, this magnitude is what is seen when looking perpendicular to the laser beam. The direction does not appear perpendicular in the figure due to the view point in three-dimensional space. This projection is equivalent to dotting the surface velocity vector with the unit vector in the laser's direction. In other words, the measured velocity magnitude is

$$\mathrm{v}_L = \dot{\mathbf{u}}_s \cdot \mathbf{e_l}$$

Substituting the definitions in Equations 2.2 and 2.3,

$$\mathrm{v}_L = \dot{u}\cos\alpha_x + \dot{v}\cos\alpha_y + \dot{w}\cos\alpha_z$$

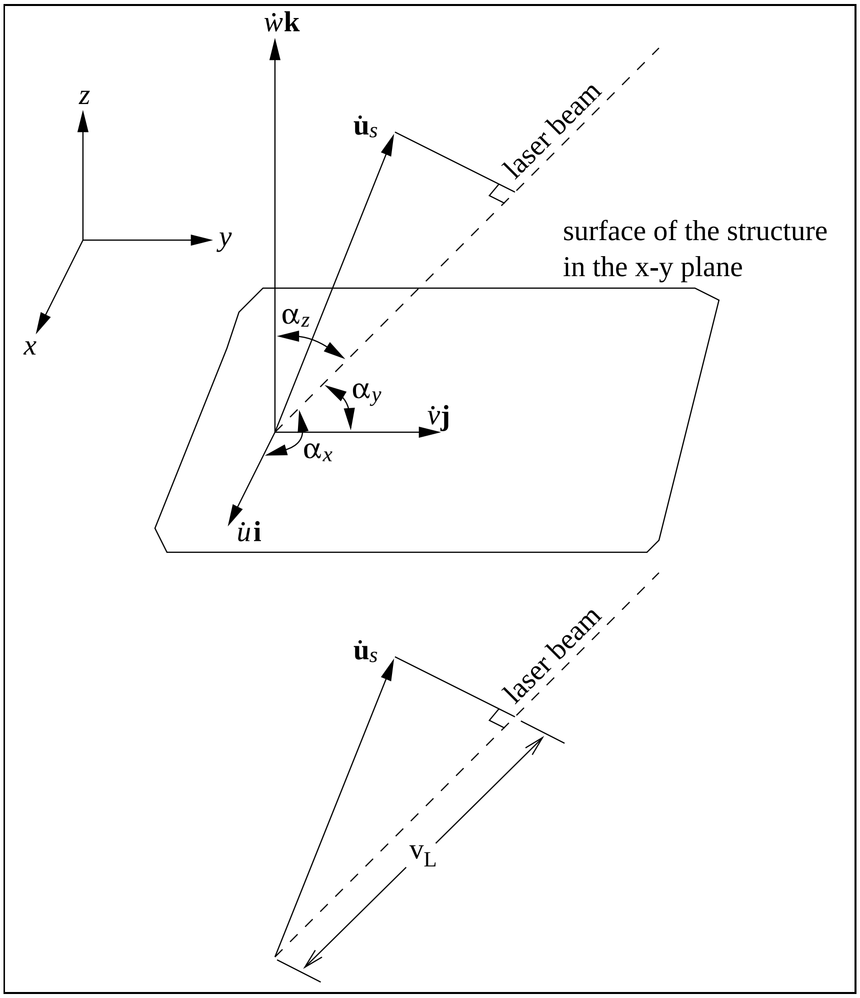

Figure 2.1: Components of Velocity Projected on to the Laser Beam

Introducing variables to represent the direction cosines

$$\Psi_x = \cos\alpha_x$$

$$\Psi_y = \cos\alpha_y$$

$$\Psi_z = \cos\alpha_z$$

the equation may be written as

$$\mathrm{v}_L = \dot{u}\Psi_x + \dot{v}\Psi_y + \dot{w}\Psi_z \tag{2.4}$$

This equation defines the relationship between the structure's surface velocity vector, $\dot{\mathbf{u}}_s$, and the velocity magnitude measured by the laser, v_L. Examination of Equation 2.4 reveals an important property of the quantity that the laser measures.

For each component of a structural velocity vector, two factors control the size of the contribution to v_L: the size of the velocity component and the value of the direction cosine. Regardless of the size of $\dot{u}$, if the laser is placed such that Ψ_x is very small, the velocity component in the x-direction will hardly be *seen* by the laser. This condition also provides the test engineer the opportunity to place the laser so that relatively small velocity components make significant contributions to v_L. This allows the laser to discern a small velocity(within reason) component in the face of a larger one; information that may be useful in updating a model. Prior knowledge about the expected motion of a structure during a test should guide the decision of where to place the SLDV.

The relationship in Equation 2.4 holds regardless of how the structural velocity components vary with time. Whether $\dot{u}$, $\dot{v}$, $\dot{w}$ are steady, vary randomly in time, decay in time, or vary periodically, Equation 2.4 defines the velocity magnitude that the laser will measure at an instant in time. A derivation in the next section shows how a function describes how v_L varies in time given that $\dot{u}$, $\dot{v}$, and $\dot{w}$ vary harmonically.

2.4 Velocities Measured When the Structure Vibrates Harmonically

As stated in the Section 2.2, a linear structure excited by a harmonic function of the form $f = \cos(\omega t + \phi_f)$ will respond by vibrating harmonically at the same frequency. This information, along with Equation 2.4, will be used to derive a function describing how the quantity the laser measures varies in time when the structure is responding to harmonic excitation.

In general, the vibration at a given point on a structure excited harmonically at the single frequency ω will take the form

$$\dot{u} = \dot{\bar{u}} \cos(\omega t + \phi_{\dot{u}}) \tag{2.5}$$

$$\dot{v} = \dot{\bar{v}} \cos(\omega t + \phi_{\dot{v}}) \tag{2.6}$$

$$\dot{w} = \dot{\bar{w}} \cos(\omega t + \phi_{\dot{w}}) \tag{2.7}$$

where $\dot{u}$, $\dot{v}$, and $\dot{w}$ are velocities in the three coordinate directions, $\dot{\bar{u}}$, $\dot{\bar{v}}$, and $\dot{\bar{w}}$ are the velocity amplitudes, and $\phi_{\dot{u}}$, $\phi_{\dot{v}}$, and $\phi_{\dot{w}}$ are the phase angles of the responses. The three components, $\dot{u}$, $\dot{v}$, and $\dot{w}$, fully define the motion of the point at the frequency, ω, in three dimensions for all time t that the forcing function persists. Although the vibrations of the point in each of the three coordinate directions are at the frequency of the excitation, the phases of the motions($\phi_{\dot{u}}$, $\phi_{\dot{v}}$, $\phi_{\dot{w}}$) are in general different. The phase angles and the velocity amplitudes are functions of the position on the structure so that the components of the velocity at a point are a function of position, frequency, and time

$$\begin{aligned}
\dot{u} &= \dot{u}(x,y,z,\omega,t) = \dot{\bar{u}}(x,y,z)\cos[\omega t + \phi_{\dot{u}}(x,y,z)] \\
\dot{v} &= \dot{v}(x,y,z,\omega,t) = \dot{\bar{v}}(x,y,z)\cos[\omega t + \phi_{\dot{v}}(x,y,z)] \\
\dot{w} &= \dot{w}(x,y,z,\omega,t) = \dot{\bar{w}}(x,y,z)\cos[\omega t + \phi_{\dot{w}}(x,y,z)]
\end{aligned}$$

By inserting Equations 2.5, 2.6, and 2.7 into Equation 2.4, the velocity magnitude at a location measured by the laser of a harmonically excited structure is shown to be

$$\mathrm{v}_L = \dot{\bar{u}}\cos(\omega t + \phi_{\dot{u}})\Psi_x + \dot{\bar{v}}\cos(\omega t + \phi_{\dot{v}})\Psi_y + \dot{\bar{w}}\cos(\omega t + \phi_{\dot{w}})\Psi_z \qquad (2.8)$$

Inspection of this equation shows that when the laser is used to measure a velocity magnitude of a harmonically excited structure, the measurement is the value of a harmonic signal at the instant of measurement. This signal is the projection of the structure velocity onto the laser beam. In this section, a harmonic(single frequency) response is examined as opposed to a periodic(multiple frequencies) response; however, all the results here can easily be extended to periodic response, as will be shown in Section 2.8.

If an oscilloscope were placed so that v_L could be traced as the laser views a location on a structure during a single frequency harmonic test, the trace would appear as shown in Figure 2.2. Actually this trace is an idealization since there would be noise present in the signal; however, it does illustrate that none of the information Equation 2.8 shows goes into creating v_L is readily apparent in the trace. However, the circular frequency and direction cosines are known, and the amplitudes

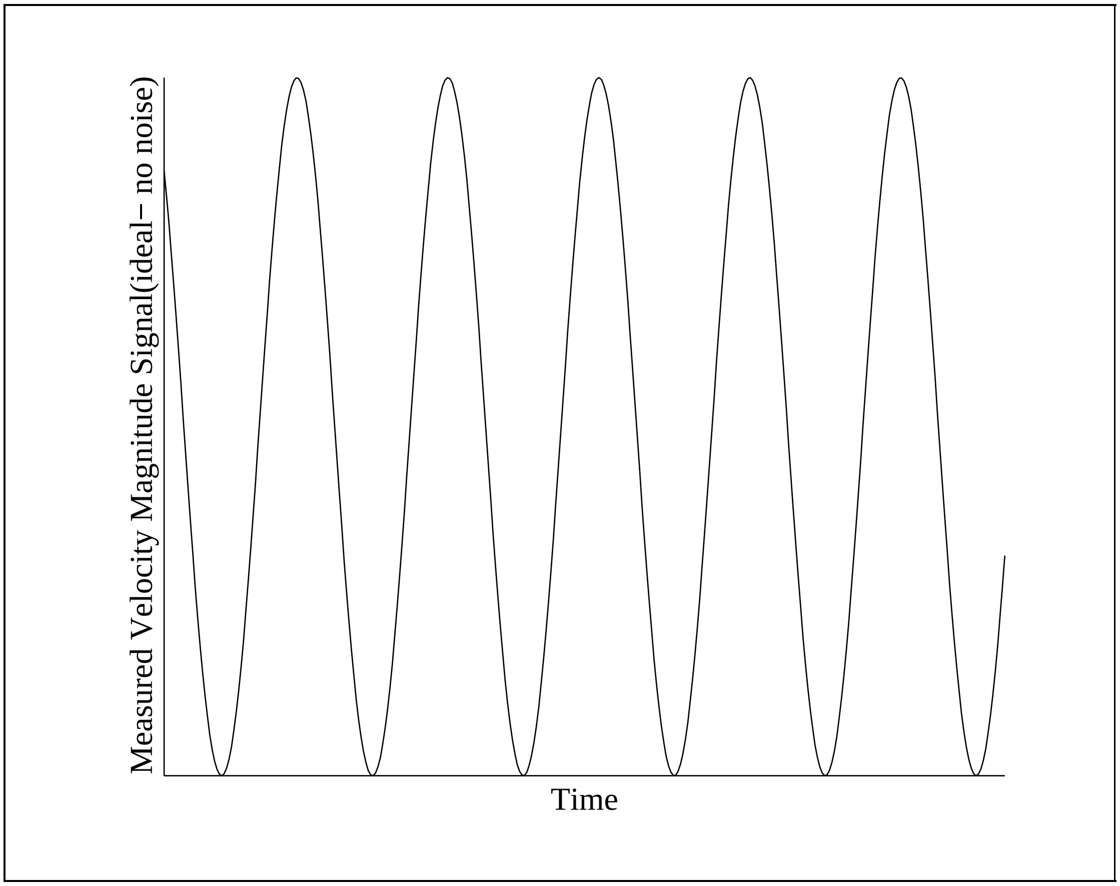

Figure 2.2: Measured Velocity Magnitude Signal

and phases of the structural velocity components can be found. In fact, the challenge for the algorithm is to extract the three structural velocity components' amplitudes and phases at every location the SLDV samples. The next section shows how signals like that shown in Figure 2.2 can be sampled in time and processed into a form useful for the model updating algorithm.

2.5 Regressions for a Time Domain Representation of the Input Force

Suppose a structure is being forced harmonically at a single frequency, ω. An equation that models the force being input at any time t is

$$f = \bar{f}cos(\omega t + \phi_f) \tag{2.9}$$

where $\bar{f}$ is the amplitude of the force being input and ϕ_f is the phase angle. The actual force input at the frequency ω during an experiment will never be exactly given by Equation 2.9. A model that allows for force values that do not lie exactly on the signal of Equation 2.9 is given by

$$f = \bar{f} cos(\omega t + \phi_f) + \varepsilon$$

where ε represents an error that accounts for any unpredictable or unknown factors not included in the function. The error is assumed to be a random variable that is normally distributed with a mean value of zero. The expected value of the force is given by Equation 2.9. By taking advantage of the trigonometric identity

$$\cos(A + B) = \cos A \cos B - \sin A \sin B \tag{2.10}$$

Equation 2.9 can be rewritten as

$$f = \bar{f} \cos \phi_f \cos \omega t - \bar{f} \sin \phi_f \sin \omega t \tag{2.11}$$

If new variables β_1 and β_2 are introduced such that

$$\beta_1 = \bar{f} \cos \phi_f$$
$$\beta_2 = \bar{f} \sin \phi_f$$

then Equation 2.11 can be written as

$$f = \beta_1 \cos \omega t - \beta_2 \sin \omega t \tag{2.12}$$

The true values of the constants β_1 and β_2 can never be known exactly, but they can be estimated using the method of least squares. If s samples are taken of the actual value of the input force($f_1, f_2, \ldots, f_s$) during the experiment at times $t_1, t_2, \ldots, t_s$, these samples can be used to produce an estimated model of the force input. This estimate is represented as

$$\hat{f} = \hat{\beta}_1 \cos \omega t - \hat{\beta}_2 \sin \omega t \tag{2.13}$$

where the hat symbols are used to denote estimates. The goal is to choose the parameters $\hat{\beta}_1$ and $\hat{\beta}_2$ so that the error of the prediction at any given time is small. The error between the prediction and sample taken at time t_i is given by

$$e_i = f_i - \hat{f}_i$$

where f_i is the i^{th} sample in the time series data, and $\hat{f}_i$ is the value of the function estimate at time t_i. Inserting Equation 2.13

$$e_i = f_i - (\hat{\beta}_1 \cos \omega t_i - \hat{\beta}_2 \sin \omega t_i)$$

The error values for all the samples($i = 1, 2, \ldots, s$) can be put into one vector as

$$\left\{ \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_s \end{array} \right\} = \left\{ \begin{array}{c} f_1 \\ f_2 \\ \vdots \\ f_s \end{array} \right\} - \left\{ \begin{array}{c} \hat{\beta}_1 \cos \omega t_1 - \hat{\beta}_2 \sin \omega t_1 \\ \hat{\beta}_1 \cos \omega t_2 - \hat{\beta}_2 \sin \omega t_2 \\ \vdots \\ \hat{\beta}_1 \cos \omega t_s - \hat{\beta}_2 \sin \omega t_s \end{array} \right\}$$

Factoring out the parameter estimates

$$\left\{ \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_s \end{array} \right\} = \left\{ \begin{array}{c} f_1 \\ f_2 \\ \vdots \\ f_s \end{array} \right\} - \left[\begin{array}{cc} \cos \omega t_1 & -\sin \omega t_1 \\ \cos \omega t_2 & -\sin \omega t_2 \\ \vdots & \vdots \\ \cos \omega t_s & -\sin \omega t_s \end{array} \right] \left\{ \begin{array}{c} \hat{\beta}_1 \\ \hat{\beta}_2 \end{array} \right\}$$

The error can be represented compactly as

$$\mathbf{e} = \mathbf{f} - \mathbf{A}\hat{\mathbf{b}}$$

where

$$\mathbf{e} = \left\{ \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_s \end{array} \right\} \qquad \mathbf{f} = \left\{ \begin{array}{c} f_1 \\ f_2 \\ \vdots \\ f_s \end{array} \right\} \qquad \mathbf{A} = \left[\begin{array}{cc} \cos \omega t_1 & -\sin \omega t_1 \\ \cos \omega t_2 & -\sin \omega t_2 \\ \vdots & \vdots \\ \cos \omega t_s & -\sin \omega t_s \end{array} \right] \qquad \hat{\mathbf{b}} = \left\{ \begin{array}{c} \hat{\beta}_1 \\ \hat{\beta}_2 \end{array} \right\}$$

The sum of the squared error is

$$\mathbf{e}^{\mathrm{T}}\mathbf{e} = \mathbf{f}^{\mathrm{T}}\mathbf{f} - 2\hat{\mathbf{b}}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{f} + \hat{\mathbf{b}}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\hat{\mathbf{b}}$$

The best estimates for the parameters minimize the sum of squared error. The minimum of this equation occurs where its derivative with respect to the parameter estimates is zero. This derivative is

$$\frac{\partial(\mathbf{e}^{\mathrm{T}}\mathbf{e})}{\partial\hat{\mathbf{b}}} = -2\mathbf{A}^{\mathrm{T}}\mathbf{f} + 2\mathbf{A}^{\mathrm{T}}\mathbf{A}\hat{\mathbf{b}}$$

Setting this derivative equal to zero and solving for $\hat{\mathbf{b}}$ results in

$$\hat{\mathbf{b}} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{f}$$

These values for the parameter estimates minimize the errors between the predicted force values and the sampled values. Thus, the estimate for the force input at a single frequency found by sampling the values of the actual input force is

$$\hat{f} = \hat{\beta}_1 \cos\omega t - \hat{\beta}_2 \sin\omega t$$

The derivation presented earlier in this section(Equations 2.9 through 2.12) shows that this equation is equivalent to $\hat{f} = \widehat{\overline{f}}\cos(\omega t + \hat{\phi}_f)$.

Combining the identities

$$\mathrm{Re}[r(\cos\theta + i\sin\theta)] = r\cos\theta$$

and

$$r(\cos\theta + i\sin\theta) = re^{i\theta}$$

results in $\mathrm{Re}\left[re^{i\theta}\right] = r\cos\theta$. It is clear that the forcing function can be written as $\hat{f} = \mathrm{Re}\left[\widehat{\overline{f}}e^{i(\omega t+\hat{\phi}_f)}\right]$. This mathematical expression can be represented geometrically as shown in Figure 2.3. In this representation, $\hat{f}$ is the projection of a vector in the complex plane onto the Real axis, where $\widehat{\overline{f}}$ is the vector's magnitude and $\hat{\phi}_f$ is the angle indicating the orientation of the vector measured from the reference angle, ωt. The phase angle $\hat{\phi}_f$ is a result of when in time the force signal was first sampled for the regression. Had the samples been acquired at a different starting time, the phase angle would have been different.

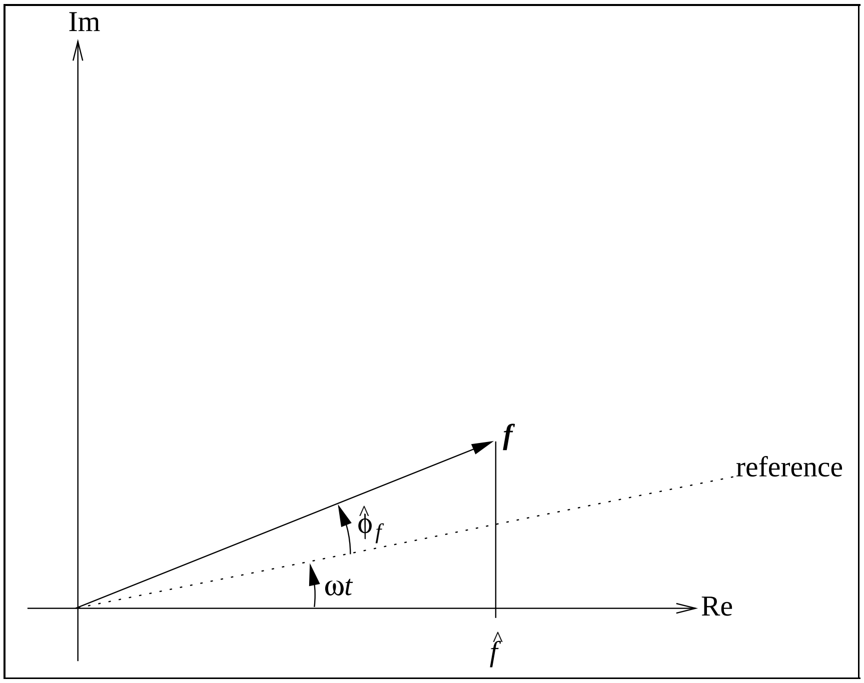

Figure 2.3: Harmonic Forcing Function as a Complex Vector

2.6 Regressions for a Time Domain Representation of v_L

The velocity magnitude measured by the laser at a location on the structure can be treated in a similar manner to the input force function to produce an estimate of a function describing the measured velocity magnitude. The value of velocity magnitude that the laser *sees* at any given instant in time, v_L, was given in Equation 2.4. As shown in Section 2.3, the velocity magnitude of a harmonically excited structure is given by(restating Equation 2.8)

$$\mathrm{v}_L = \dot{u}\cos(\omega t + \phi_{\dot{u}})\Psi_x + \dot{v}\cos(\omega t + \phi_{\dot{v}})\Psi_y + \dot{w}\cos(\omega t + \phi_{\dot{w}})\Psi_z \tag{2.8}$$

so that a laser measurement at an instant in time is a sample of the harmonically varying signal. This signal is a projection onto the laser of the combination of the components of the velocity at the measurement point on the structure. The exact form of the wave is unknown, but by assuming that its expected value is given by Equation 2.8, a function can be fit that gives an estimate of how the measured velocity varies with time at the sample location. Using the same trigonometric identity as used earlier(Equation 2.10) to expand the cosine terms and then grouping terms by $\cos\omega t$ and $\sin\omega t$ results in:

$$\begin{aligned}\mathrm{v}_L = (\dot{u}\cos\phi_{\dot{u}}\Psi_x + \dot{v}\cos\phi_{\dot{v}}\Psi_y + \dot{w}\cos\phi_{\dot{w}}\Psi_z)\cos\omega t \\ - (\dot{u}\sin\phi_{\dot{u}}\Psi_x + \dot{v}\sin\phi_{\dot{v}}\Psi_y + \dot{w}\sin\phi_{\dot{w}}\Psi_z)\sin\omega t\end{aligned}$$

If new variables γ_1 and γ_2 are introduced such that

$$\gamma_1 = \dot{u}\cos\phi_{\dot{u}}\Psi_x + \dot{v}\cos\phi_{\dot{v}}\Psi_y + \dot{w}\cos\phi_{\dot{w}}\Psi_z \tag{2.14}$$

$$\gamma_2 = \dot{u}\sin\phi_{\dot{u}}\Psi_x + \dot{v}\sin\phi_{\dot{v}}\Psi_y + \dot{w}\sin\phi_{\dot{w}}\Psi_z \tag{2.15}$$

then the equation may be rewritten as

$$\mathrm{v}_L = \gamma_1\cos\omega t - \gamma_2\sin\omega t$$

This form of the equation shows more clearly that the measured velocity magnitude is simply a harmonic signal with circular frequency ω.

Just like the force model, this equation is the expected value of the measured velocity magnitude at the location the laser strikes. The actual value at any given time will vary randomly about the expected value. Using a regression similar to the one used for the force input, the estimate of the function for the velocity magnitude measured at a location on the structure is

$$\hat{\mathrm{v}}_L = \hat{\gamma}_1\cos\omega t - \hat{\gamma}_2\sin\omega t \tag{2.16}$$

The parameters $\hat{\gamma}_1$ and $\hat{\gamma}_2$ are given by the equation $\hat{\mathbf{c}} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{v}$ where

$$\mathbf{A} = \begin{bmatrix} \cos\omega t_1 & -\sin\omega t_1 \\ \cos\omega t_2 & -\sin\omega t_2 \\ \vdots & \vdots \\ \cos\omega t_s & -\sin\omega t_s \end{bmatrix} \qquad \mathbf{v} = \begin{Bmatrix} \mathrm{v}_{L_1} \\ \mathrm{v}_{L_2} \\ \vdots \\ \mathrm{v}_{L_s} \end{Bmatrix} \qquad \hat{\mathbf{c}} = \begin{Bmatrix} \hat{\gamma}_1 \\ \hat{\gamma}_2 \end{Bmatrix}$$

Assuming that Equation 2.8 truly represents how the motion of the structure translates into what the laser measures, $\hat{\gamma}_1$ and $\hat{\gamma}_2$ carry the information about the magnitude and phase of all three of the structural velocity components at the sample location on the structure.

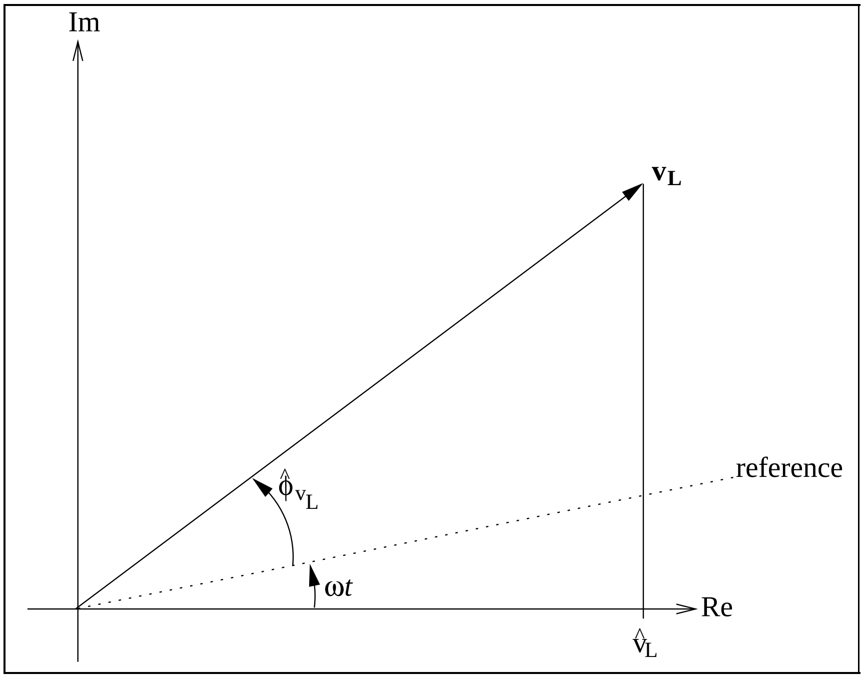

Figure 2.4: Measured Velocity Magnitude as a Complex Vector

Similar to the force, Equation 2.16 can be represented as $\hat{\mathrm{v}}_L = \mathrm{Re}\left[\hat{\hat{\mathrm{v}}}_L e^{i(\omega t+\hat{\phi}_{\mathrm{v}_L})}\right]$. As with the force, this can be thought of as the projection of a vector in the complex plane onto the Real axis, as shown in Figure 2.4. The magnitude of the vector is $\hat{\hat{\mathrm{v}}}_L$ and its phase angle with respect to the reference, ωt, is $\hat{\phi}_{\mathrm{v}_L}$. This phase angle is the result of the particular instant in time that samples began being acquired of the velocity magnitude that the laser measures.

2.7 Setting Phases Relative to a Reference Force

As was shown in the previous sections, the regressions for force and velocity result in time domain estimates for the harmonic functions that have phase angles that are dependent on when the time series data samples began. In the application presented in this thesis, the SLDV will be used to measure velocities at many points on a structure. Each velocity function estimate resulting from the regression has a phase angle measured from a different reference, making the phase information useless. Even though the motion is assumed harmonic, it would be impractical, if not impossible, to guarantee that each velocity would be sampled beginning at the same instant in the period so that the phase for each velocity is measured with respect to the same reference. A better solution is to measure the phase for the velocity function at each measurement location with respect to a reference force chosen before the test begins. In order to use a force as the reference, the force and velocity must be sampled simultaneously for each laser measurement location on the test structure. Also, the velocities and force must be assumed to be stationary, so that the phase used at each velocity measurement location is made with respect to the same reference.

For a test where there are multiple force inputs, one would be chosen as the reference and the other forces would measure their phases relative to it. For example, with two force inputs at one frequency, ω, sampled simultaneously so that their phase is measured with respect to the same reference, the function estimates are

$$\hat{f}_1 = \mathrm{Re}\left[\widehat{\bar{f}}_1 e^{i(\omega t+\hat{\phi}_{f_1})}\right] \tag{2.17}$$

$$\hat{f}_2 = \mathrm{Re}\left[\widehat{\bar{f}}_2 e^{i(\omega t+\hat{\phi}_{f_2})}\right] \tag{2.18}$$

The complex vector representation for these functions is shown in Figure 2.5. Following the regression, both forces measure phase relative to the reference angle ωt by when the time series data was first taken.

In the process of setting a reference for phase, a frequency domain representation will be introduced that carries only the information necessary for the model updating algorithm. A frequency domain representation of these functions will show only their magnitude and phase. The frequency domain representation takes the form $\mathrm{F} = \mathrm{f}_{Re} + i\mathrm{f}_{Im}$, where the magnitude is $\bar{f} = \sqrt{\mathrm{f}_{Re}^2 + \mathrm{f}_{Im}^2}$ and the phase relative to the reference is $\hat{\phi}_f = \arctan\left(\mathrm{f}_{Im}/\mathrm{f}_{Re}\right)$. The harmonic nature of the signals, held in $e^{i\omega t}$, is understood to be there, but is not explicitly presented in F. Choosing $\hat{f}_1$ as the reference force,

$$\hat{f}_1 = \mathrm{Re}\left[\left(\widehat{\bar{f}}_1\right)\left(e^{i(\omega t+\hat{\phi}_{f_1})}\right)\right]$$

so that its frequency domain representation is $\mathrm{F}_r = \hat{\bar{f}}_1$. This makes its compnents $\mathrm{f}_{Re_r} = \widehat{\bar{f}}_1$ and $\mathrm{f}_{Im_r} = 0$. The second force's frequency domain representation will have a phase angle measured relative to the first force, which is now the reference. The

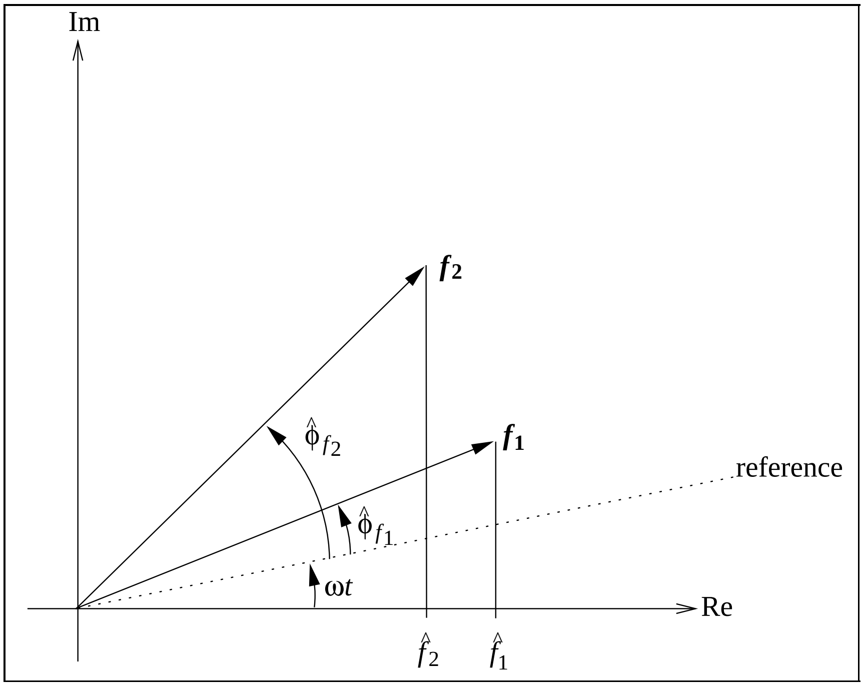

Figure 2.5: Two Force Inputs in the Complex Plane

time domain representation for the second force is

$$\hat{f}_2 = \mathrm{Re}\left[\left(\hat{\overline{f}}_2 e^{i\phi_r}\right)\left(e^{i(\omega t+\phi_{ref})}\right)\right] \tag{2.19}$$

where $\phi_{ref} = \hat{\phi}_{f_1}$ since f_1 is the reference force for this frequency. As shown in Figure 2.6, the phase angle relative to the reference force is $\phi_r = \hat{\phi}_{f_2} - \hat{\phi}_{f_1}$. The function in Equation 2.19 is necessary to mathematically recreate the signal for $\hat{f}_2$, but is not useful for updating. The frequency domain representation, $\mathrm{F}_2 = \hat{\overline{f}}_2 e^{i\phi_r}$ is more useful. Using Euler's identity,

$$\mathrm{f}_{Re_2} = \hat{\overline{f}}_2 \cos\phi_r$$

$$\mathrm{f}_{Im_2} = \hat{\overline{f}}_2 \sin\phi_r$$

For each velocity sample location, the reference force must be sampled simultaneously so that the relative phase can be calculated. The frequency domain representation for the velocity at a sample location is $\mathrm{V} = \mathrm{v}_{L_{Re}} + i\mathrm{v}_{L_{Im}}$, where

$$\mathrm{v}_{L_{Re}} = \hat{\bar{\mathrm{v}}}_L \cos\phi_r$$

$$\mathrm{v}_{L_{Im}} = \hat{\bar{\mathrm{v}}}_L \sin\phi_r$$

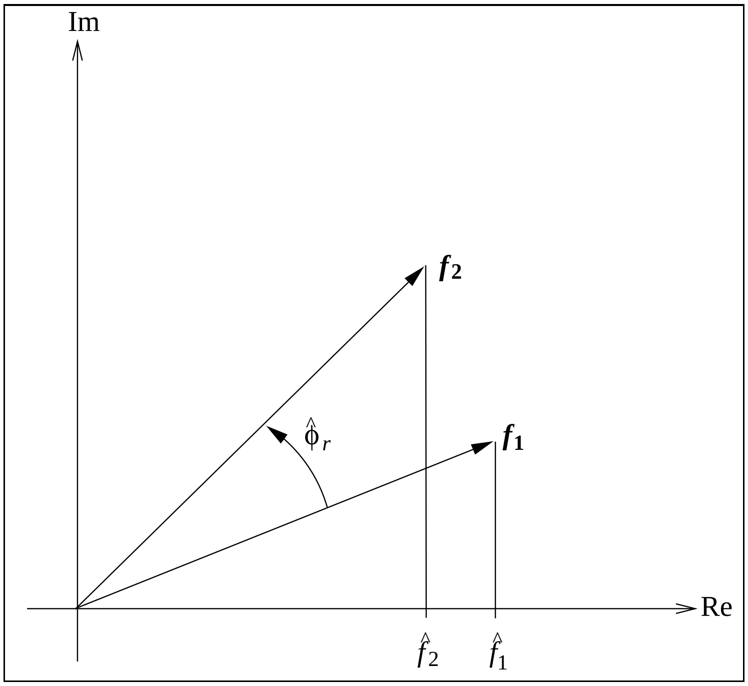

Figure 2.6: Force Inputs With the Phase Measured Relative to the Reference

The equivalent time domain representation for sample point i is

$$
\begin{aligned}
\mathrm{v}_{L_i} &= \mathrm{Re}\left[(V)\left(e^{i(\omega t+\phi_{ref})}\right)\right] \\
\mathrm{v}_{L_i} &= \mathrm{Re}\left[\left(\hat{\mathrm{v}}_{L_i} e^{i\phi_r}\right)\left(e^{i(\omega t+\phi_{ref})}\right)\right] \\
\mathrm{v}_{L_i} &= \mathrm{Re}\left[(\mathrm{v}_{L_{Re_i}} + i\mathrm{v}_{L_{Im_i}})\left(e^{i(\omega t+\phi_{ref})}\right)\right] \qquad (2.20)
\end{aligned}
$$

Each sample location will have a different phase, ϕ_r. In terms of the regression coefficients, the parts of the frequency domain representation, V, are

$$\mathrm{v}_{L_{Re}} = (\hat{\beta}_1\hat{\gamma}_1 + \hat{\beta}_2\hat{\gamma}_2) \Big/ \sqrt{\hat{\beta}_1^2 + \hat{\beta}_2^2} \qquad (2.21)$$

$$\mathrm{v}_{L_{Im}} = (\hat{\beta}_1\hat{\gamma}_2 - \hat{\beta}_2\hat{\gamma}_1) \Big/ \sqrt{\hat{\beta}_1^2 + \hat{\beta}_2^2} \qquad (2.22)$$

where $\hat{\beta}_1$ and $\hat{\beta}_2$ are the coefficents found for the reference force and $\hat{\gamma}_1$ and $\hat{\gamma}_2$ are the coefficents for the measured velocity magnitude. The definitions given by Equations 2.21 and 2.22 can be used to calculate the components of V directly from the regression coefficients.

The velocity components, $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$, and force components, f_{Re} and f_{Im}, are used by the update algorithm which also uses a frequency domain representation.

The velocities are used in the experimental model, while the forces are used in the finite element model equations. In the next chapter, it will be shown that the form of this data(F and V) is exactly the same as that used in a finite element model for harmonic response.

2.8 Multiple Frequency Response

Section 2.7's results were based on a single frequency input, and single frequency response. This section shows how the previous results can be generalized to a periodic force input— one that has more than one frequency component.

Suppose a forcing function has the form

$$f = f_0 + \bar{f}_1 \cos(\omega_1 t + \phi_{f_1}) + \bar{f}_2 \cos(\omega_2 t + \phi_{f_2}) + \cdots + \bar{f}_n \cos(\omega_n t + \phi_{f_n}) \tag{2.23}$$

Using the identity of Equation 2.10, this function can be written as

$$\begin{aligned} f = f_0 &+ \bar{f}_1 \cos\phi_{f_1} \cos\omega_1 t - \bar{f}_1 \sin\phi_{f_1} \sin\omega_1 t \\ &+ \bar{f}_2 \cos\phi_{f_2} \cos\omega_2 t - \bar{f}_2 \sin\phi_{f_2} \sin\omega_2 t + \cdots \\ &+ \bar{f}_n \cos\phi_{f_n} \cos\omega_n t - \bar{f}_n \sin\phi_{f_n} \sin\omega_n t \end{aligned} \tag{2.24}$$

Introducing the variables

$$\begin{array}{rlrl} \beta_0 &= f_0 & & \\ \beta_1 &= \bar{f}_1 \cos\phi_{f_1} & \beta_2 &= \bar{f}_1 \sin\phi_{f_1} \\ \beta_3 &= \bar{f}_2 \cos\phi_{f_2} & \beta_4 &= \bar{f}_2 \sin\phi_{f_2} \\ &\vdots & &\vdots \\ \beta_{2n-1} &= \bar{f}_n \cos\phi_{f_n} & \beta_{2n} &= \bar{f}_n \sin\phi_{f_n} \end{array} \tag{2.25}$$

the function can be written as

$$\begin{aligned} f = \beta_0 &+ \beta_1 \cos\omega_1 t - \beta_2 \sin\omega_1 t \\ &+ \beta_3 \cos\omega_2 t - \beta_4 \sin\omega_2 t + \cdots \\ &+ \beta_{2n-1} \cos\omega_n t - \beta_{2n} \sin\omega_n t \end{aligned} \tag{2.26}$$

As with the single frequency force, the parameters β_i can never be known exactly; however, they can be estimated using the method of least squares. For the i^{th} sample in time of the force, the error between the actual value and estimated model is

$$e_i = f_i - \hat{f}_i \tag{2.27}$$

Placing the error for each sample into a vector and then factoring out the parameter estimates

$$\left\{\begin{matrix} e_1 \\ e_2 \\ \vdots \\ e_s \end{matrix}\right\} = \left\{\begin{matrix} f_1 \\ f_2 \\ \vdots \\ f_s \end{matrix}\right\} - \begin{bmatrix} 1 & \cos\omega_1 t_1 & -\sin\omega_1 t_1 & \cdots & \cos\omega_n t_1 & -\sin\omega_n t_1 \\ 1 & \cos\omega_1 t_2 & -\sin\omega_1 t_2 & \cdots & \cos\omega_n t_2 & -\sin\omega_n t_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos\omega_1 t_s & -\sin\omega_1 t_s & \cdots & \cos\omega_n t_s & -\sin\omega_n t_s \end{bmatrix} \left\{\begin{matrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_{2n} \end{matrix}\right\} \tag{2.28}$$

As before this equation can be written compactly as

$$\mathbf{e} = \mathbf{f} - \mathbf{A}\hat{\mathbf{b}} \tag{2.29}$$

Upon squaring the error, $\mathbf{e}$, and setting the derivative with respect to the parameters equal to zero, $\hat{\mathbf{b}}$ can be solved for, giving the estimates for the parameters. The zero frequency component, $\hat{\mathrm{f}}_0$, is equal to $\hat{\beta}_0$. For the p^{th} frequency component, the time domain representation for the forcing function is

$$\hat{\mathrm{f}}_p = \mathrm{Re}\left[\mathrm{F}_p e^{(i\omega_p t + \hat{\phi}_p)}\right]$$

where $\mathrm{F}_p = \mathrm{f}_{Re_p} + i\mathrm{f}_{Im_p}$ is the frequency domain representation. Assuming that there is only one force at the frequency ω_p, $\mathrm{f}_{Re_p} = \hat{\overline{f}}_p$ and $\mathrm{f}_{Im_p} = 0$. If there is more than one force acting at frequency ω_p, then one would be chosen as the reference and the others would have phase angles relative to the reference(they could have nonzero f_{Im_p}).

As was discussed in Section 2.2, the response of a linear system to a function comprised of several harmonic functions of differing frequencies is the sum of harmonic responses at each individual frequency. Since it was assumed that the structure being tested is linear, the response measured by the laser as velocity magnitude when the forcing function takes the form of Equation 2.23 is

$$\mathrm{v}_L = \mathrm{v}_{L_0} + \bar{\mathrm{v}}_{L_1}\cos(\omega_1 t + \phi_{\mathrm{v}_{L_1}}) + \bar{\mathrm{v}}_{L_2}\cos(\omega_2 t + \phi_{\mathrm{v}_{L_2}}) + \cdots + \bar{\mathrm{v}}_{L_n}\cos(\omega_n t + \phi_{\mathrm{v}_{L_n}})$$

As with the force, the method of least squares can be used to estimate the parameters that define this response. Setting the velocity phase angle relative to the reference force for the frequency, ω_p, the frequency domain representation for the measured velocity is $\mathrm{V}_p = \mathrm{v}_{L_{Re_p}} + i\mathrm{v}_{L_{Im_p}}$. The magnitude is $\overline{\mathrm{V}}_p = \sqrt{\mathrm{v}^2_{L_{Re_p}} + \mathrm{v}^2_{L_{Im_p}}}$ and the phase with respect to the reference force is $\phi_{r_p} = \arctan(\mathrm{v}_{L_{Im_p}}/\mathrm{v}_{L_{Re_p}})$. In terms of the regression coefficients defined earlier in the section, the components of the frequeny domain representation are

$$\begin{aligned}\mathrm{v}_{L_{Re_p}} &= (\hat{\beta}_{2p-1}\hat{\gamma}_{2p-1} + \hat{\beta}_{2p}\hat{\gamma}_{2p}) \Big/ \sqrt{\hat{\beta}^2_{2p-1} + \hat{\beta}^2_{2p}} \\ \mathrm{v}_{L_{Im_p}} &= (\hat{\beta}_{2p-1}\hat{\gamma}_{2p} + \hat{\beta}_{2p}\hat{\gamma}_{2p-1}) \Big/ \sqrt{\hat{\beta}^2_{2p-1} + \hat{\beta}^2_{2p}}\end{aligned} \tag{2.30}$$

where $\hat{\beta}_i$ are the coefficients for the reference force, and $\hat{\gamma}_i$ are the coefficients for the velocity at the measurement location and frequency.

As was stated in the previous section, the velocity and force components will be used in the update algorithm in their frequency domain forms. For a multiple frequency test, more than one frequency will be used in the model update, hopefully resulting in a better model than one that is the product of a single frequency update.

2.9 Summary

This chapter shows the steps to process experimental data from a vibration test into a form that is useful for the update algorithm. Force values measured by a transducer at the input point are processed into a frequency domain representation, $\mathrm{F} = \mathrm{f}_{Re} + i\mathrm{f}_{Im}$. For the frequency of interest, this representation contains all the information the update algorithm needs: the magnitude and phase of the signal. Not coincidentally, this representation is also used by the FE model for harmonic response. The values found for the force in the experiment will be used in the FE model so that there is no need to assume a value for the force— it is known to the accuracy of the transducer. This chapter also shows how the frequency domain representation for the velocity magnitude at a point measured by the laser is found. This representation, $\mathrm{V} = \mathrm{v}_{L_{Re}} + i\mathrm{v}_{L_{Im}}$, has a phase angle that is measured relative to a reference force. It is important to remember that $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$ will be found for many points on the test structure since the SLDV has the capability to automatically scan and measure velocity at many locations. The velocity components may also be found at several frequencies since the linear regressions can process multiple frequencies. The next chapter will show how the $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$ components for many points on a structure will be integrated into an experimental model for each frequency. The relationship between the structure's motion and what the laser measures, shown in Equation 2.4, is important since it forms part of the experimental model.

Chapter 3

Optimization Formulation

Updating requires experimental data, an analytical model, and some criteria to compare them. In this chapter, an equation which is an expression of the squared error between the measurements of structural velocities made by the SLDV and the modeled response is developed. Included in the experimental error equation's development is a transformation from the outer surface of the structure, where the laser measurements are made, to the midplane, where the nodes for a finite element model are typically located. The equation for the experimental error is developed specifically so that it shares a method consistent with finite element models to model displacement. This consistency appears as the common component $\dot{\mathbf{d}}$, a vector of nodal velocities. Through this common component, an objective function is formulated so that parameters in the finite element model can be updated. Minimizing this objective function should cause the *best* values for the parameters to be found for a given candidate finite element model.

3.1 Finite Element Shape Functions to Model Displacement

Cook (1995) states that the finite element method was first introduced for practical use by engineers who needed to solve complex structural problems. Later, mathematicians proved the validity of the technique and generalized the formulation for use in many areas of analysis. The following discussion follows a mathematical development for the finite element method since shape functions are more easily presented in this type development. Shape functions are discussed in detail since they can be used to construct a piecewise continuous function representing displacement for structures with any geometry. In this algorithm, they will be used, in conjunction with measurements of velocity at a limited number of locations during a vibration test, to represent the velocity over the whole test structure.

Consider the governing differential equation for the analysis of a steady state problem

$$L(\phi) = f \quad \text{in } D \tag{3.1}$$

where L is a linear differential operator, ϕ is the field variable to be calculated, and f is the forcing function. The solution for ϕ must satisfy Equation 3.1 over the problem domain, D. The solution for ϕ must also satisfy the boundary conditions

$$B(\phi) = q \quad \text{on } S \tag{3.2}$$

on the boundary S. This boundary may be made up of several sections that are not necessarily connected. Together, Equations 3.1 and 3.2 define a boundary value problem. For simple problems, a closed form solution for the boundary value problem can be found. For complex problems, a closed form solution may not be possible.

An approximate solution of the form $\hat{\phi} = \sum_{i=1}^{n} a_i \psi_i$ can be used for the cases where the boundary value problem is too complex to be solved in closed form. This approximate solution is a linear combination of linearly independent basis functions, ψ_i. The multipliers, a_i, are determined in the solution. Finite element analysis produces a solution of this form. As shown by Bathe (1982), the formulation for the finite element method can be derived by applying Galerkin's method, a weighted residual method, to a boundary value problem like that defined in Equations 3.1 and 3.2. The important step that distinguishes the finite element method is to define the basis functions, ψ_i, only over small subdomains(elements) of the problem's domain,

D. The basis functions are defined over small subdomains since some aspect of the problem makes it very difficult(or impossible) to define them over all of D. Since the basis functions are only defined over small domains, they may be defined by simple functions. This is due to the fact that it is assumed that, for a small part of D, it is reasonable to approximate the exact solution in this fashion. One important part of any finite element solution is to ensure that the subdomains are small enough that this assumption is valid.

In the jargon that has arisen for the finite element method, locally defined basis functions are called shape functions and are typically symbolized N_i. Defining the functions over small subdomains and not requiring basis functions defined over all of D gives the FE method the capability to find solutions for problems with arbitrary geometry and loads. These are the situations that classical analytical methods generally can not handle. By connecting all of the elements together, the field variable that is being solved for may be interpolated over the entire problem domain.

Generally, more than one element is used over the domain for a problem, but the remainder of this discussion limits the formulation to one element for simplicity. The process of modeling an entire structure with multiple elements will be discussed later.

For the boundary value problems arising in structural analysis, the field variable to be calculated is the displacement. When using the finite element method to define these problems, the displacement is modeled as the linear combination of the shape functions. In general, this linear combination takes the form

$$\mathbf{u} = \mathbf{N}\mathbf{d} \tag{3.3}$$

where $\mathbf{u}$ is the vector of field variables, displacements, $\mathbf{N}$ is the matrix of shape functions for the element, and $\mathbf{d}$ is the vector of nodal displacements for that element. This equation is also called the interpolation equation since the displacements can be found at any point in the element given the nodal displacements.

For a linear rod element, shown in Cook (1995), the only displacement is axial, u, and, as shown in Figure 3.1, each node has only one degree of freedom. The axial

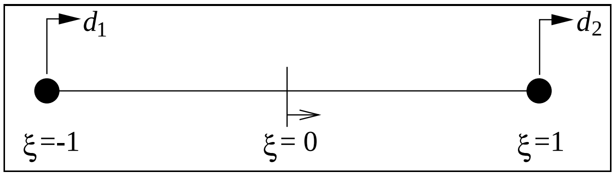

Figure 3.1: Rod Element

displacements of the nodes, d_1 and d_2, are referred to as the degrees of freedom for the element. The interpolation equation(Equation 3.3) for a rod takes the form

$$u = \begin{bmatrix} N_{1r} & N_{2r} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}$$

where N_{1r} and N_{2r} are the shape functions associated with the first and second degrees of freedom. The shape functions are defined as

$$N_{1r} = \frac{1}{2}(1 - \xi)$$
$$N_{2r} = \frac{1}{2}(1 + \xi)$$

Based on the form of these shape functions, the displacement within this element varies linearly with ξ on the interval (-1, 1). The shape functions are initially defined in a general element coordinate system, here the ξ coordinate. When a structure is meshed, a transformation is introduced to define the relationship between the coordinate system for the structure being modeled and the general element coordinate system. Along with establishing the relationship between coordinate systems for elements arbitrarily oriented in the structure coordinate system, the transformation must also allow for arbitrarily sized elements. Cook shows how this relationship is established using isoparametric elements and the Jacobian matrix.

A beam element based on the mechanics of the Bernoulli-Euler beam theory, discussed in Donaldson (1993), has the degrees of freedom shown in Figure 3.2. The first and third degrees of freedom are the transverse displacements at the first and second nodes, respectively. The second and fourth degrees of freedom are rotations at the first and second nodes, respectively. The interpolation equation for a beam

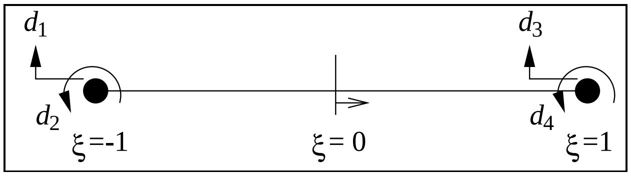

Figure 3.2: Beam Element

element, $\mathbf{u} = \mathbf{N}\mathbf{d}$, has the components

$$w = \begin{bmatrix} N_{1b} & N_{2b} & N_{3b} & N_{4b} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix}$$

As shown, the field variable for a beam is the transverse displacement, w and the field vector $\mathbf{u}$ reduces to this one component. The shape functions associated with the transverse displacements for a beam are

$$\begin{aligned} N_{1b} &= \frac{1}{4}(2 - 3\xi + \xi^3) \\ N_{2b} &= \frac{h_e}{8}(1 - \xi - \xi^2 + \xi^3) \\ N_{3b} &= \frac{1}{4}(2 + 3\xi + \xi^3) \\ N_{4b} &= \frac{h_e}{8}(-1 - \xi + \xi^2 + \xi^3) \end{aligned}$$

These shape functions are defined for the general element with the coordinate ξ spanning the interval (-1, 1). The transverse displacement field as modeled by these shape functions can be a cubic function. This field is the exact solution for any beam loaded only at its ends(no span load).

A beam element can be developed that represents both the axial displacements of a rod and the transverse displacements associated with a beam. This rod/beam element can be developed if the displacements, axial and transverse, are assumed to be small enough that rod and beam models can be linearly superposed. The

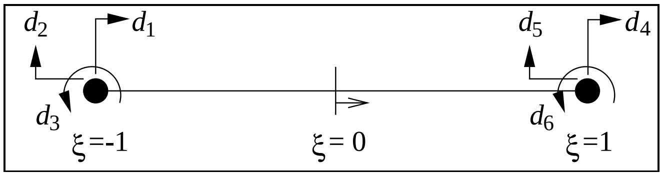

Figure 3.3: Rod/Beam Element

interpolation equation takes the form

$$\left\{ \begin{array}{c} u \\ w \end{array} \right\} = \left[\begin{array}{cccccc} N_{1r} & 0 & 0 & N_{2r} & 0 & 0 \\ 0 & N_{1b} & N_{2b} & 0 & N_{3b} & N_{4b} \end{array} \right] \left\{ \begin{array}{c} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{array} \right\} \tag{3.4}$$

where u is the axial displacement and w is the transverse displacement. The shape functions for the rod and beam are the same as those defined earlier. The components of $\mathbf{d}$ are defined as shown in Figure 3.3. The first and fourth degrees of freedom are the axial displacements for the rod element. The second and fifth degrees of freedom are transverse displacements of the beam at the first and second nodes, respectively. The third and sixth degrees of freedom are beam rotations at the left and right nodes, respectively. The interpolation for the superposed rod/beam given by Equation 3.4 is in the same form as Equation 3.3 where the shape function matrix is

$$\mathbf{N} = \left[\begin{array}{cccccc} N_{1r} & 0 & 0 & N_{2r} & 0 & 0 \\ 0 & N_{1b} & N_{2b} & 0 & N_{3b} & N_{4b} \end{array} \right] \tag{3.5}$$

The interpolation equations for the rod, beam, and rod/beam elements all took the form of Equation 3.3, $\mathbf{u} = \mathbf{N}\dot{\mathbf{d}}$. In fact, the interpolation equation for any finite element, based on any structural or continuum mechanics theory, will have this form. Through this equation, the displacement is modeled as a linear combination of the shape functions. When the problem domain is divided into multiple elements, the elements are connected by the nodes. The connections at the nodes ensure that the displacement is at least piecewise continuous across elements. In some cases, higher derivatives of the displacement are also continuous across elements. For example, the slope(first derivative of the transverse displacement) between beam elements is continuous. By connecting all the elements together(direct stiffness method), and thus piecing together the shape functions, the displacement is modeled over the entire problem domain.

A traditional structural finite element model gives the displacement over the problem domain due to some load. The model is based on a structural or continuum mechanics theory. The finite element framework will be used in a nontraditional fashion for the purposes of this research to develop an experimental model. In this experimental model, shape functions will interpolate a velocity field based on values measured experimentally by the SLDV, not due to a load and not based on a particular mechanics theory. Shape functions will be used in the experimental model for the same reasons that they are used in the finite element method— so that the model can be applied to any shape structure being tested. In addition, it will be shown that this formulation results in a convenient method to relate test measurements to a traditionally developed analytical model through the displacement vector, $\dot{\mathbf{d}}$.

3.2 Transformation From Midplane to Surface

The node displacements and shape functions map the displacement over the whole element. Structural elements(rods, beams, plates, shells) are conventionally developed from the neutral axis(for beams), midplane(for plates), or midsurface(for shells) of the modeled structure, so the displacements are for this surface of the structure. For structures modeled with these types of elements, a constraint equation which is consistent with the deformation mechanics of the model is needed to relate these midplane displacements to the displacements at the surface of the structure, where the laser makes its measurements. The following discussion illustrates how the transformation is developed by going through the process for a beam.

Figure 3.4 shows how displacements at the surface of a beam relate to the displacements at the neutral axis, where beams are modeled. Point A is at the

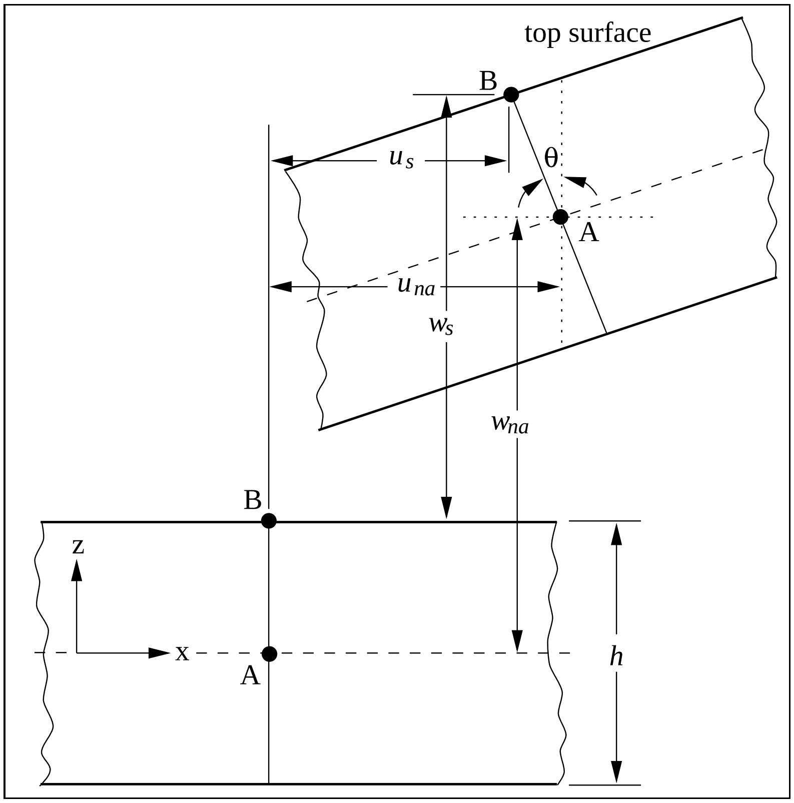

Figure 3.4: Transformation of Displacement from the Neutral Axis to the Surface for a Beam

neutral axis of the beam, which typically lies midway between the top and bottom surfaces, while Point B is the corresponding point on the top surface of the beam. The transformation developed here is for a test where laser measurements are made on this top surface. The figure shows a typical displaced configuration with the displacements highly exaggerated to make the illustration clearer. The points A and B are shown in the lower left corner of the figure in their undisplaced positions. The displacements at the neutral axis are marked by u_{na} and w_{na}, while u_s and w_s are the displacements at the top surface. Figure 3.5 shows a close-up of the top half of the cross-section of the beam so the relationship between the surface and neutral axis displacements is clear. As the points on the neutral axis displace in the z-direction, the cross-sections of the beam rotate but remain planar in the displaced configuration. The depth of the beam is h, so the distance from the neutral axis to the top surface is $\frac{h}{2}$. The angle of rotation of the cross-section AB is θ. The relationships between the neutral axis and surface displacements are

$$\begin{aligned} u_s &= u_{na} - \frac{h}{2}\sin\theta \\ w_s &= w_{na} - (\frac{h}{2} - \frac{h}{2}\cos\theta) \qquad (3.6) \end{aligned}$$

Since the algorithm is based on structural velocities, the time derivative needs to be taken of these equations, which results in

$$\dot{u}_s = \dot{u}_{na} - \frac{h}{2}\dot{\theta}\cos\theta \qquad (3.7)$$

$$\dot{w}_s = \dot{w}_{na} - \frac{h}{2}\dot{\theta}\sin\theta \qquad (3.8)$$

Assuming that the angle θ is small so that $\cos\theta \approx 1$ and $\sin\theta \approx \theta$ gives the relationship

$$\dot{u}_s = \dot{u}_{na} - \frac{h}{2}\dot{\theta} \qquad (3.9)$$

$$\dot{w}_s = \dot{w}_{na} - \frac{h}{2}\dot{\theta}\theta \qquad (3.10)$$

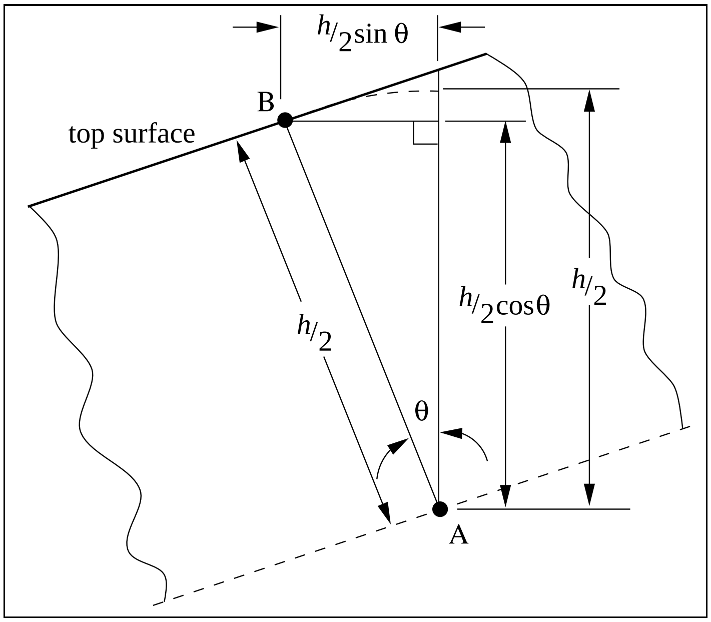

Figure 3.5: Close-up of Top Half of Beam Cross-Section

For most problems the displacements will be small, so the term $\frac{h}{2}\dot{\theta}\theta$ will be very small in comparison to the other terms so that it can be neglected. Neglecting this term, the transformation of the two components of velocity from the neutral axis to the surface for a beam is

$$\dot{u}_s = \dot{u}_{na} - \frac{h}{2}\dot{\theta} \tag{3.11}$$

$$\dot{w}_s = \dot{w}_{na} \tag{3.12}$$

The finite element node velocities can be related to the velocity at any point on the surface by introducing the transformed shape function matrix $\mathbf{N_R}$ which is defined as

$$\mathbf{N_R} = \mathbf{N} + \mathbf{R} \tag{3.13}$$

where $\mathbf{N}$ is the shape function matrix for the element and $\mathbf{R}$ is a transformation matrix. The shape functions model the velocities at the neutral axis and the addition of $\mathbf{R}$ will allow surface velocities to be modeled. The surface velocities can then be related to the node velocities as

$$\dot{\mathbf{u}}_s = \mathbf{N_R}\dot{\mathbf{d}} \tag{3.14}$$

For the beam element, the transformations are given by Equations 3.11 and 3.12. Writing these equations in matrix form

$$\left\{ \begin{array}{c} \dot{u}_s \\ \dot{w}_s \end{array} \right\} = \left\{ \begin{array}{c} \dot{u}_{na} \\ \dot{w}_{na} \end{array} \right\} + \left\{ \begin{array}{c} -\frac{h}{2}\dot{\theta} \\ 0 \end{array} \right\} \tag{3.15}$$

$$\dot{\mathbf{u}}_s = \dot{\mathbf{u}}_{na} + \left\{ \begin{array}{c} -\frac{h}{2}\dot{\theta} \\ 0 \end{array} \right\} \tag{3.16}$$

Remembering that the neutral axis velocities are given by $\dot{\mathbf{u}}_{na} = \mathbf{N}\dot{\mathbf{d}}$,

$$\dot{\mathbf{u}}_s = \mathbf{N}\dot{\mathbf{d}} + \left\{ \begin{array}{c} -\frac{h}{2}\dot{\theta} \\ 0 \end{array} \right\} \tag{3.17}$$

where $\dot{\mathbf{u}}_{na}$and $\dot{\mathbf{d}}$ are time derivatives of the quantities defined in Equation 3.4 and $\mathbf{N}$ is the same since the shape functions are time invariant.

A relationship for the angular velocity, $\dot{\theta}$, is required in terms of the nodal velocities at the neutral axis. The derivation requires the assumption that the derivative with respect to the axial coordinate of the displacement(the slope) of the beam's neutral axis is equal to the angle of deflection for small deflection angles. The transverse velocity is

$$\dot{w} = N_{1b}\dot{d}_2 + N_{2b}\dot{d}_3 + N_{3b}\dot{d}_5 + N_{4b}\dot{d}_6 \tag{3.18}$$

so the time derivative of the angle of deflection is

$$\begin{aligned} \dot{\theta} &= \frac{\partial}{\partial x}\dot{w} \\ \dot{\theta} &= \frac{\partial}{\partial x}N_{1b}\dot{d}_2 + \frac{\partial}{\partial x}N_{2b}\dot{d}_3 + \frac{\partial}{\partial x}N_{3b}\dot{d}_5 + \frac{\partial}{\partial x}N_{4b}\dot{d}_6 \end{aligned} \tag{3.19}$$

In matrix form

$$\dot{\theta} = \left\{ \begin{array}{cccccc} 0 & \frac{\partial}{\partial x}N_{1b} & \frac{\partial}{\partial x}N_{2b} & 0 & \frac{\partial}{\partial x}N_{3b} & \frac{\partial}{\partial x}N_{4b} \end{array} \right\} \left\{ \begin{array}{c} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \\ \dot{d}_4 \\ \dot{d}_5 \\ \dot{d}_6 \end{array} \right\}$$

$$\dot{\theta} = \left\{ \begin{array}{cccccc} 0 & \frac{\partial}{\partial x}N_{1b} & \frac{\partial}{\partial x}N_{2b} & 0 & \frac{\partial}{\partial x}N_{3b} & \frac{\partial}{\partial x}N_{4b} \end{array} \right\} \dot{\mathbf{d}} \tag{3.20}$$

where $\dot{\mathbf{d}}$ is the vector of node velocities at the neutral axis for a beam. From this equation, the transformation matrix is defined for a beam as

$$\mathbf{R} = -\frac{h}{2} \begin{bmatrix} 0 & \frac{\partial}{\partial x}N_{1b} & \frac{\partial}{\partial x}N_{2b} & 0 & \frac{\partial}{\partial x}N_{3b} & \frac{\partial}{\partial x}N_{4b} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.21}$$

Inserting the transformation matrix, $\mathbf{R}$, into Equation 3.17 results in

$$\begin{aligned} \dot{\mathbf{u}}_s = & \begin{bmatrix} N_{1r} & 0 & 0 & N_{2r} & 0 & 0 \\ 0 & N_{1b} & N_{2b} & 0 & N_{3b} & N_{4b} \end{bmatrix} \dot{\mathbf{d}} \\ & -\frac{h}{2} \begin{bmatrix} 0 & \frac{\partial}{\partial x}N_{1b} & \frac{\partial}{\partial x}N_{2b} & 0 & \frac{\partial}{\partial x}N_{3b} & \frac{\partial}{\partial x}N_{4b} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \dot{\mathbf{d}} \end{aligned} \tag{3.22}$$

or

$$\begin{aligned} \dot{\mathbf{u}}_s &= \mathbf{N}\dot{\mathbf{d}} + \mathbf{R}\dot{\mathbf{d}} \\ \dot{\mathbf{u}}_s &= (\mathbf{N} + \mathbf{R})\dot{\mathbf{d}} \end{aligned} \tag{3.23}$$

and finally, restating Equation 3.14

$$\dot{\mathbf{u}}_s = \mathbf{N_R}\dot{\mathbf{d}} \tag{3.24}$$

Hence the transformed shape function matrix, $\mathbf{N_R}$, for a beam is

$$\mathbf{N_R} = \begin{bmatrix} N_{1r} & -\frac{h}{2}\frac{\partial}{\partial x}N_{1b} & -\frac{h}{2}\frac{\partial}{\partial x}N_{2b} & N_{2r} & -\frac{h}{2}\frac{\partial}{\partial x}N_{3b} & -\frac{h}{2}\frac{\partial}{\partial x}N_{4b} \\ 0 & N_{1b} & N_{2b} & 0 & N_{3b} & N_{4b} \end{bmatrix} \tag{3.25}$$

The process used here to derive a transformation matrix for a beam element can be performed for any structural finite element so that the relation between the neutral axis(or midplane) velocities and surface velocities can always be found in the form of Equation 3.14. This transformation is not needed for all cases. For example, if the structure is modeled with 3-D brick elements, then no transformation is needed. The surface velocities can be found directly using $\dot{\mathbf{u}} = \mathbf{N}\dot{\mathbf{d}}$ for elements that border the surface.

The shape functions discussed in the last section and the transformation derived in this section have laid the foundation for modeling the experimental data with a piecewise continuous function made up of the shape functions. In the next section, the experimental model will be developed where $\dot{\mathbf{d}}$ is the vector of parameters whose best solution will minimize the error between the modeled response function and values measured by the SLDV.

3.3 Formulation of the Experimental Model

Sections 3.1 and 3.2 established how the velocity at the surface of the structure can be related to the velocity of nodes for a finite element mesh using shape functions. The velocities of the nodes of the finite element mesh can be used to model the velocity at the structure surface for every point that the laser samples using the transformed interpolation equation of Equation 3.14. This allows the use of the finite element shape functions as the basis for a piecewise continuous surface fit to the experimental velocity data. The node velocities can be thought of as parameters in a least squares fit whose values should minimize the error between the modeled response field, defined by the shape functions, and the velocity measured at each sample point by the SLDV. The measured velocity at each sample point is represented by the frequency domain components, $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$, which were developed in Chapter 2.

Equation 2.4, restated here shows how the velocity at a point on the surface of a structure is *seen* by the laser.

$$\mathrm{v}_L = \dot{u}\Psi_x + \dot{v}\Psi_y + \dot{w}\Psi_z \tag{2.4}$$

This equation includes the actual structural velocity components, $\dot{u}, \dot{v}$, and $\dot{w}$— quantities that are not known. The values for the components can be guessed, but there will be an error between the projection of the guesses onto the laser beam direction and what the instrument actually *saw* in the form of v_L. The error for any location i the laser samples is given by

$$e_i = \mathrm{v}_{L_i} - (\dot{u}_i \Psi_{x_i} + \dot{v}_i \Psi_{y_i} + \dot{w}_i \Psi_{z_i}) \tag{3.26}$$

If the structure is vibrating harmonically, and time series data at location i has been processed as shown in Chapter 2, the estimated function that describes how v_L varies in time is given by(restating Equation 2.20)

$$\hat{\mathrm{v}}_{L_i} = \mathrm{Re}\left[(\mathrm{v}_{L_{Re_i}} + i\mathrm{v}_{L_{Im_i}}) e^{i(\omega t + \phi_{ref})} \right] \tag{2.20}$$

where ϕ_{ref} is the phase angle for the reference input force for that frequency. This phase angle is not important to the formulation of the experimental model, but is included for completeness. The important information is carried by the components $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$which represent the magnitude and phase(relative to the reference) of $\hat{\mathrm{v}}_L$. Based on the function in Equation 2.4, estimates must be made for the functions that describe how $\dot{u}$, $\dot{v}$, and $\dot{w}$ vary with time.

Expanding Equation 2.20 and leaving out ϕ_{ref}, since the relevant information is carried in $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$, the velocity at a location sampled by the laser varies harmonically in time as

$$\hat{\mathrm{v}}_L = \mathrm{v}_{L_{Re}} \cos \omega t - \mathrm{v}_{L_{Im}} \sin \omega t$$

Assuming the structure surface velocities have the same form, the three components are given by

$$\begin{aligned} \dot{u}_s &= \dot{u}_{s_{Re}} \cos \omega t - \dot{u}_{s_{Im}} \sin \omega t \\ \dot{v}_s &= \dot{v}_{s_{Re}} \cos \omega t - \dot{v}_{s_{Im}} \sin \omega t \\ \dot{w}_s &= \dot{w}_{s_{Re}} \cos \omega t - \dot{w}_{s_{Im}} \sin \omega t \end{aligned}$$

Inserting these equations into Equation 3.26, the error for sample point i is

$$\begin{aligned} e_i = \left[\mathrm{v}_{L_{Re}} - (\dot{u}_{s_{Re}}\Psi_x + \dot{v}_{s_{Re}}\Psi_y + \dot{w}_{s_{Re}}\Psi_z)\right]_i \cos\omega t \\ - \left[\mathrm{v}_{L_{Im}} - (\dot{u}_{s_{Im}}\Psi_x + \dot{v}_{s_{Im}}\Psi_y + \dot{w}_{s_{Im}}\Psi_z)\right]_i \sin\omega t \end{aligned} \quad (3.27)$$

Thus the error between the guessed velocity values and the measured values for a sample location varies harmonically. The best guesses will minimize the amplitude of the error, which is

$$\bar{e}_i = \sqrt{e^2_{Re_i} + e^2_{Im_i}}$$

where

$$\begin{aligned} e_{Re_i} &= \left[\mathrm{v}_{L_{Re}} - (\dot{u}_{s_{Re}}\Psi_x + \dot{v}_{s_{Re}}\Psi_y + \dot{w}_{s_{Re}}\Psi_z)\right]_i \\ e_{Im_i} &= \left[\mathrm{v}_{L_{Im}} - (\dot{u}_{s_{Im}}\Psi_x + \dot{v}_{s_{Im}}\Psi_y + \dot{w}_{s_{Im}}\Psi_z)\right]_i \end{aligned}$$

The subscript i following the closing parenthesis in these equations is used to indicate that each term inside the parenthesis is the value for sample location i.

Minimizing the squared error amplitude will achieve the same goal as minimizing the error amplitude while providing an easier expression with which to work. The squared error amplitude for sample location i is

$$\bar{e}_i^2 = \left[\mathrm{v}_{L_{Re}} - (\dot{u}_{s_{Re}}\Psi_x + \dot{v}_{s_{Re}}\Psi_y + \dot{w}_{s_{Re}}\Psi_z)\right]_i^2 + \left[\mathrm{v}_{L_{Im}} - (\dot{u}_{s_{Im}}\Psi_x + \dot{v}_{s_{Im}}\Psi_y + \dot{w}_{s_{Im}}\Psi_z)\right]_i^2 \quad (3.28)$$

Minimizing the squared error will result in the best estimate for the magnitude and phase of each directional component($\dot{u}$, $\dot{v}$, $\dot{w}$). Introducing vectors to simplify Equation 3.28

$$\mathbf{y} = \begin{Bmatrix} \Psi_x \\ \Psi_y \\ \Psi_z \end{Bmatrix} \qquad \dot{\mathbf{u}}_{s_{Re}} = \begin{Bmatrix} \dot{u}_{s_{Re}} \\ \dot{v}_{s_{Re}} \\ \dot{w}_{s_{Re}} \end{Bmatrix} \qquad \dot{\mathbf{u}}_{s_{Im}} = \begin{Bmatrix} \dot{u}_{s_{Im}} \\ \dot{v}_{s_{Im}} \\ \dot{w}_{s_{Im}} \end{Bmatrix}$$

the squared error amplitude for sample point i can be written as the multiplication of two vectors

$$\bar{e}_i^2 = \begin{Bmatrix} \mathrm{v}_{L_{Re}} - \dot{\mathbf{u}}^{\mathrm{T}}_{s_{Re}}\mathbf{y} & \mathrm{v}_{L_{Im}} - \dot{\mathbf{u}}^{\mathrm{T}}_{s_{Im}}\mathbf{y} \end{Bmatrix}_i \begin{Bmatrix} \mathrm{v}_{L_{Re}} - \mathbf{y}^{\mathrm{T}}\dot{\mathbf{u}}_{s_{Re}} \\ \mathrm{v}_{L_{Im}} - \mathbf{y}^{\mathrm{T}}\dot{\mathbf{u}}_{s_{Im}} \end{Bmatrix}_i \quad (3.29)$$

Using the relationship between node velocities and surface velocities from Equation 3.14, $\dot{\mathbf{u}}_s = \mathbf{N_R}\dot{\mathbf{d}}$, the squared error amplitude is

$$\bar{e}_i^2 = \left\{ \begin{array}{cc} \mathrm{v}_{L_{Re}} - \dot{\mathbf{d}}_{Re}^{\mathrm{T}}\mathbf{N_R}^{\mathrm{T}}\mathbf{y} & \mathrm{v}_{L_{Im}} - \dot{\mathbf{d}}_{Im}^{\mathrm{T}}\mathbf{N_R}^{\mathrm{T}}\mathbf{y} \end{array} \right\}_i \left\{ \begin{array}{c} \mathrm{v}_{L_{Re}} - \mathbf{y}^{\mathrm{T}}\mathbf{N_R}\dot{\mathbf{d}}_{Re} \\ \mathrm{v}_{L_{Im}} - \mathbf{y}^{\mathrm{T}}\mathbf{N_R}\dot{\mathbf{d}}_{Im} \end{array} \right\} \tag{3.30}$$

where $\dot{\mathbf{d}}_{Re}$ and $\dot{\mathbf{d}}_{Im}$ are vectors of nodal velocity. Thus, the first degree of freedom for an element is

$$\dot{d}_1 = \mathrm{Re}[(\dot{d}_{1_{Re}} + i\dot{d}_{1_{Im}})e^{i(\omega t + \phi_{ref})}]$$

where $\dot{d}_{1_{Re}}$ is the first component of $\dot{\mathbf{d}}_{Re}$ and $\dot{d}_{1_{Im}}$ is the first component of $\dot{\mathbf{d}}_{Im}$. Together, these components make up the frequency domain representation for the motion of the first degree of freedom. The components, $\dot{d}_{1_{Re}}$ and $\dot{d}_{1_{Im}}$ carry the amplitude and phase relative to the reference input. Expanding Equation 3.30

$$\begin{aligned} \bar{e}_i^2 = {} & \dot{\mathbf{d}}_{Re}^{\mathrm{T}}(\mathbf{N_R}^{\mathrm{T}}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{N_R})_i\dot{\mathbf{d}}_{Re} + \dot{\mathbf{d}}_{Im}^{\mathrm{T}}(\mathbf{N_R}^{\mathrm{T}}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{N_R})_i\dot{\mathbf{d}}_{Im} \\ & - 2(\mathrm{v}_{L_{Re}}\mathbf{y}^{\mathrm{T}}\mathbf{N_R})_i\dot{\mathbf{d}}_{Re} - 2(\mathrm{v}_{L_{Im}}\mathbf{y}^{\mathrm{T}}\mathbf{N_R})_i\dot{\mathbf{d}}_{Im} + \mathrm{v}_{L_{Re_i}}^2 + \mathrm{v}_{L_{Im_i}}^2 \end{aligned} \tag{3.31}$$

Recognizing that removing the constants $\mathrm{v}_{L_{Re_i}}^2$ and $\mathrm{v}_{L_{Im_i}}^2$ affects the minimum value of $\bar{e}_i^2$, but not where the minimum occurs, the two terms can be dropped. Introducing the terms

$$\begin{aligned} \mathbf{L_i} &= (\mathbf{N_R}^{\mathrm{T}}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{N_R})_i \\ \mathbf{l}_{Re_i} &= (\mathrm{v}_{L_{Re}}\mathbf{N_R}^{\mathrm{T}}\mathbf{y})_i \\ \mathbf{l}_{Im_i} &= (\mathrm{v}_{L_{Im}}\mathbf{N_R}^{\mathrm{T}}\mathbf{y})_i \end{aligned} \tag{3.32}$$

the squared error amplitude for sample point i becomes

$$\bar{e}_i^2 = \dot{\mathbf{d}}_{Re}^{\mathrm{T}}\mathbf{L_i}\dot{\mathbf{d}}_{Re}^{\mathrm{T}} + \dot{\mathbf{d}}_{Im}^{\mathrm{T}}\mathbf{L_i}\dot{\mathbf{d}}_{Im}^{\mathrm{T}} - 2\mathbf{l}_{Re_i}^{\mathrm{T}}\dot{\mathbf{d}}_{Re} - 2\mathbf{l}_{Im_i}^{\mathrm{T}}\dot{\mathbf{d}}_{Im} \tag{3.33}$$

The vectors $\mathbf{l}_{Re_i}$ and $\mathbf{l}_{Im_i}$ have the *Re* and *Im* subscripts since they contain the components of the experimental velocity data, $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$, respectively. Equation 3.33 is the squared error amplitude for one laser sample location. In order to make the best fit between the modeled velocity field and all of the laser sample locations, the squared errors should be summed for all the sample points. This sum, for n sample locations is

$$\begin{aligned}\sum_{i=1}^{n} \bar{e}_i^2 = \dot{\mathbf{d}}_{Re}^{\mathrm{T}}\mathbf{L_1}\dot{\mathbf{d}}_{Re} + \dot{\mathbf{d}}_{Re}^{\mathrm{T}}\mathbf{L_2}\dot{\mathbf{d}}_{Re} + \cdots + \dot{\mathbf{d}}_{Re}^{\mathrm{T}}\mathbf{L_n}\dot{\mathbf{d}}_{Re} \\ + \dot{\mathbf{d}}_{Im}^{\mathrm{T}}\mathbf{L_1}\dot{\mathbf{d}}_{Im} + \dot{\mathbf{d}}_{Im}^{\mathrm{T}}\mathbf{L_2}\dot{\mathbf{d}}_{Im} + \cdots + \dot{\mathbf{d}}_{Im}^{\mathrm{T}}\mathbf{L_n}\dot{\mathbf{d}}_{Im} \\ - 2\mathbf{l}_{Re_1}^{\mathrm{T}}\dot{\mathbf{d}}_{Re} - 2\mathbf{l}_{Re_2}^{\mathrm{T}}\dot{\mathbf{d}}_{Re} - \cdots - 2\mathbf{l}_{Re_n}^{\mathrm{T}}\dot{\mathbf{d}}_{Re} \\ - 2\mathbf{l}_{Im_1}^{\mathrm{T}}\dot{\mathbf{d}}_{Im} - 2\mathbf{l}_{Im_2}^{\mathrm{T}}\dot{\mathbf{d}}_{Im} - \cdots - 2\mathbf{l}_{Im_n}^{\mathrm{T}}\dot{\mathbf{d}}_{Im} \quad (3.34)\end{aligned}$$

The node velocities, $\dot{\mathbf{d}}_{Re}$ and $\dot{\mathbf{d}}_{Im}$, do not change for each sample location, and the information for the velocity, location and the direction cosines of each point is carried in $\mathbf{L_i}$, $\mathbf{l}_{Re_i}$, and $\mathbf{l}_{Im_i}$. Consequently, the matrices and vectors for all the sample locations can be grouped and added together so that the sum of squared error can be written as

$$\sum_{i=1}^{n} \bar{e}_i^2 = \dot{\mathbf{d}}_{Re}^{\mathrm{T}}\mathbf{L}_\Sigma\dot{\mathbf{d}}_{Re} + \dot{\mathbf{d}}_{Im}^{\mathrm{T}}\mathbf{L}_\Sigma\dot{\mathbf{d}}_{Im} - 2\mathbf{l}_{Re_\Sigma}^{\mathrm{T}}\dot{\mathbf{d}}_{Re} - 2\mathbf{l}_{Im_\Sigma}^{\mathrm{T}}\dot{\mathbf{d}}_{Im}$$

where

$$\begin{aligned}\mathbf{L}_\Sigma &= \mathbf{L_1} + \mathbf{L_2} + \cdots + \mathbf{L_n} \\ \mathbf{l}_{Re_\Sigma} &= \mathbf{l}_{Re_1} + \mathbf{l}_{Re_2} + \cdots + \mathbf{l}_{Re_n} \\ \mathbf{l}_{Im_\Sigma} &= \mathbf{l}_{Im_1} + \mathbf{l}_{Im_2} + \cdots + \mathbf{l}_{Im_n} \quad (3.35)\end{aligned}$$

The equation can be further simplified if the the following vectors and matrix are substituted

$$\mathbf{Q} = \begin{bmatrix} \mathbf{L}_\Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_\Sigma \end{bmatrix} \qquad \mathbf{c} = \begin{Bmatrix} \mathbf{l}_{Re_\Sigma} \\ \mathbf{l}_{Im_\Sigma} \end{Bmatrix} \qquad \dot{\mathbf{d}} = \begin{Bmatrix} \dot{\mathbf{d}}_{Re} \\ \dot{\mathbf{d}}_{Im} \end{Bmatrix}$$

Now the function whose minimization produces the velocity field that most closely matches the velocities measured by the laser is

$$\chi^2 = \dot{\mathbf{d}}^{\mathrm{T}}\mathbf{Q}\dot{\mathbf{d}} - 2\mathbf{c}^{\mathrm{T}}\dot{\mathbf{d}} \quad (3.36)$$

where the sum of the squared error amplitude, $\sum_{i=1}^{n} \bar{e}_i^2$, has been replaced by the symbol χ^2. The matrix $\mathbf{Q}$ and the vector $\mathbf{c}$ carry all of the relevant experimental data: the measured velocity magnitudes and the direction cosines.

3.4 Assembling the Global Matrices

Up to this point, the algorithm has been formulated showing how the matrices and vectors of experimental data($\mathbf{Q}$ and $\mathbf{c}$) for individual elements are constructed. The matrices and vectors for each element must be assembled into a global matrix and global vector before a solution for the whole structure can be found.

For each measurement location on the structure, a decision must be made as to which element the measurement is located in. If the measurement is located on a border between elements(an unlikely event), the measurement should only contribute to one of the elements, otherwise the effect is as if more than one measurement at that location on the structure was made. Once the appropriate element for a measurement is selected, the contribution of the test data for sample i to the element matrix($\mathbf{L^e_i}$) or vectors($\mathbf{l}^e_{Re_i}$ and $\mathbf{l}^e_{Im_i}$) is made, here for element e.

For each element e, $\mathbf{L^e_\Sigma}$, $\mathbf{l}^e_{Re_\Sigma}$, and $\mathbf{l}^e_{Im_\Sigma}$ are constructed by summing the data for all the samples that occurred in the domain of that particular element. Then, the element matrices/vectors are combined, using a process called the direct stiffness method by Bathe (1982), into a global matrix/vector. In this process, each element matrix/vector adds its contribution to a matrix vector representing the degrees of freedom for the entire structure. These global matrices/vectors are represented as $\mathbf{L}_{\Sigma_{global}}$, $\mathbf{l}_{Re_{\Sigma_{global}}}$ and $\mathbf{l}_{Im_{\Sigma_{global}}}$ where

$$\begin{aligned}\mathbf{L}_{\Sigma_{global}} &= \mathbf{L^1_\Sigma} + \mathbf{L^2_\Sigma} + \cdots + \mathbf{L^m_\Sigma}\\ \mathbf{l}_{Re_{\Sigma_{global}}} &= \mathbf{l}^1_{Re_\Sigma} + \mathbf{l}^2_{Re_\Sigma} + \cdots + \mathbf{l}^m_{Re_\Sigma}\\ \mathbf{l}_{Im_{\Sigma_{global}}} &= \mathbf{l}^1_{Im_\Sigma} + \mathbf{l}^2_{Im_\Sigma} + \cdots + \mathbf{l}^m_{Im_\Sigma}\end{aligned} \tag{3.37}$$

and m is the number of elements used to represent the structure. Once these global matrices/vectors are created, then the matrix $\mathbf{Q}$ and the vector $\mathbf{c}$ are created, where

$$\mathbf{Q} = \begin{bmatrix} \mathbf{L}_{\Sigma_{global}} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{\Sigma_{global}} \end{bmatrix} \qquad \mathbf{c} = \begin{Bmatrix} \mathbf{l}_{Re_{\Sigma_{global}}} \\ \mathbf{l}_{Im_{\Sigma_{global}}} \end{Bmatrix}$$

The resulting equation which represents the squared error over the whole structure is the same as that introduced in Equation 3.36

$$\chi^2 = \dot{\mathbf{d}}^{\mathrm{T}}\mathbf{Q}\dot{\mathbf{d}} - 2\mathbf{c}^{\mathrm{T}}\dot{\mathbf{d}} \tag{3.36}$$

where $\dot{\mathbf{d}}$ is the vector of node responses for the entire structure, not just one element.

Equation 3.36 is identical to the least squares variational statement presented by Montgomery (1994), with two exceptions. The first is that the equation presented here does not include the test data variance term, but it could be added easily. The second difference is that in Montgomery's work, the finite element nodes were at the surface of the structure. For the algorithm presented in this research, the nodes do not have to be located at the structure's surface. For nodes located on a surface other than the one struck by the laser, a transformation has been introduced so the node velocities can be related to the velocities measured at the surface by the laser(Equation 3.14). The minimum of the error function can be found by setting the derivative with respect to the node velocities equal to zero,

$$\frac{\partial \chi^2}{\partial \dot{\mathbf{d}}} = 2\mathbf{Q}\dot{\mathbf{d}} - 2\mathbf{c} = 0$$

The resulting solution, $\dot{\mathbf{d}} = \mathbf{Q}^{-1}\mathbf{c}$, used with the shape functions, provides a model of the velocities across the whole structure. Since Equation 3.36 can be solved for node velocities with no input from a traditional FE model based on an analytical theory, it is called the experimental model.

3.5 Finite Element Mechanics Model for Harmonic Response

An experimental model based on the finite element framework was developed in Section 3.3. The FE model based on an analytical theory is introduced in this section. This analytical model contains the parameters that will be updated.

As stated earlier, the finite element method is a technique to produce approximate solutions to boundary value problems. The finite element method is applicable to any problem for which a differential equation governs. In all cases, a set of simultaneous equations results from applying the finite element method. When applied to structural dynamics problems, the equations of motion produced by the finite element method are identical to those for a multiple degree of freedom discrete system. The equation of motion for a linear system with multiple degrees of freedom is

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{q} \tag{3.38}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, $\mathbf{K}$ is the stiffness matrix, and $\mathbf{q}$ is the force vector. The vector $\mathbf{x}$ is the response, and the dots represent derivatives with respect to time.

The equations for a harmonic response will have a special form. If the force vector has the form $\mathbf{q} = (\mathbf{f}_{Re} + i\mathbf{f}_{Im})e^{i\omega t}$, then the reponse will have the form

$$\begin{aligned}
\mathbf{x} &= -\frac{i}{\omega}\dot{\mathbf{d}}e^{i\omega t} \\
\dot{\mathbf{x}} &= \dot{\mathbf{d}}e^{i\omega t} \\
\ddot{\mathbf{x}} &= i\omega\dot{\mathbf{d}}e^{i\omega t}
\end{aligned}$$

where $\dot{\mathbf{d}} = \dot{\mathbf{d}}_{Re} + i\dot{\mathbf{d}}_{Im}$. The response is written in terms of the velocities of the degrees of freedom since that is the form of the experimental model. Inserting this solution into Equation 3.38, canceling out the $e^{i\omega t}$, and inserting $\dot{\mathbf{d}} = \dot{\mathbf{d}}_{Re} + i\dot{\mathbf{d}}_{Im}$ results in

$$\left(i\omega\mathbf{M} + \mathbf{C} - \frac{i}{\omega}\mathbf{K}\right)(\dot{\mathbf{d}}_{Re} + i\dot{\mathbf{d}}_{Im}) = \mathbf{f}_{Re} + i\mathbf{f}_{Im}$$

Even though the $e^{i\omega t}$ has been cancelled out, it is understood that the solution of this equation is a harmonic response. This is the frequency domain form of the equation.

As shown by Zienkiewicz (1977), this equation may be manipulated so that it can be solved just like a linear static problem. Expanding the equation of motion results in

$$\mathbf{C}\dot{\mathbf{d}}_{Re} + i\left(\omega\mathbf{M} - \frac{1}{\omega}\mathbf{K}\right)\dot{\mathbf{d}}_{Re} + \left(-\omega\mathbf{M} + \frac{1}{\omega}\mathbf{K}\right)\dot{\mathbf{d}}_{Im} + i\mathbf{C}\dot{\mathbf{d}}_{Im} = \mathbf{f}_{Re} + i\mathbf{f}_{Im}$$

Recognizing that the components on the left side of the equation that include i must equal the components on the right side that include i, the equation can be separated into two simultaneous equations, presented here in matrix form

$$\begin{bmatrix} \omega\mathbf{M} - \frac{1}{\omega}\mathbf{K} & \mathbf{C} \\ \mathbf{C} & -(\omega\mathbf{M} - \frac{1}{\omega}\mathbf{K}) \end{bmatrix} \left\{ \begin{array}{c} \dot{\mathbf{d}}_{Re} \\ \dot{\mathbf{d}}_{Im} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{f}_{Im} \\ \mathbf{f}_{Re} \end{array} \right\} \tag{3.39}$$

Letting

$$\mathbf{Z} = \begin{bmatrix} \omega\mathbf{M} - \frac{1}{\omega}\mathbf{K} & \mathbf{C} \\ \mathbf{C} & -(\omega\mathbf{M} - \frac{1}{\omega}\mathbf{K}) \end{bmatrix}$$

and

$$\dot{\mathbf{d}} = \left\{ \begin{array}{c} \dot{\mathbf{d}}_{Re} \\ \dot{\mathbf{d}}_{Im} \end{array} \right\} \qquad \mathbf{f} = \left\{ \begin{array}{c} \mathbf{f}_{Im} \\ \mathbf{f}_{Re} \end{array} \right\}$$

the equation can be written compactly as $\mathbf{Z}\dot{\mathbf{d}} = \mathbf{f}$, where $\mathbf{Z}$ is the impedance matrix, $\dot{\mathbf{d}}$ is the vector of velocities and $\mathbf{f}$ is the force vector. The impedance matrix is not positive definite, but it will be symmetric provided the damping matrix is symmetric. Even though $\mathbf{Z}$ is not the impedance matrix traditionally found in vibrations testing, the terminology and symbol are used here since a relationship between force and velocity is being expressed.

The *Re* and *Im* subscripts for the forces and velocities have the same meaning as those introduced in Chapter 2. The two components are there to carry the magnitude and the phase. The force vector $\mathbf{f}$ can be thought of as being made up of two vectors, $\mathbf{f}_{Im}$and $\mathbf{f}_{Re}$. The first components of these vectors carry the magnitude and phase of the force applied to the first degree of freedom, the second components correspond to the second degree of freedom, and so on. The same relationship holds true for $\dot{\mathbf{d}}$ and the components of $\dot{\mathbf{d}}_{Re}$ and $\dot{\mathbf{d}}_{Im}$. As in Chapter 2, the reference force will only have components in $\mathbf{f}_{Re}$. The experimentally measured force inputs will be placed in the force vector for the update algorithm, so the FE model must have nodes at positions corresponding to the input locations. The response, $\dot{\mathbf{d}}_{Re}$ and $\dot{\mathbf{d}}_{Im}$, will automatically have its phase relative to the reference force for that frequency. The finite element method will produce equations in the form of Equation 3.39 regardless of the geometry and loading of the structure.

The form of the vector $\dot{\mathbf{d}}$ is the same as that used in the experimental model. This formulation has been developed specifically so that there is a direct correspondence between the experimental and analytical models, making their combination convenient.

3.6 Optimization Formulations Considered

Thus far, the experimental model has been developed having the form

$$\chi^2 = \dot{\mathbf{d}}^{\mathrm{T}}\mathbf{Q}\dot{\mathbf{d}} - 2\mathbf{c}^{\mathrm{T}}\dot{\mathbf{d}}$$

This is the sum of the squared error between the measured velocity values and the surface defined by the shape functions and node velocities. The matrix $\mathbf{Q}$ and the vector $\mathbf{c}$ contain the data gathered during the test: the sample locations(in the form of shape function values), the direction cosines for each sample location, and the velocity components for each sample location, $\mathrm{v}_{L_{Re}}$and $\mathrm{v}_{L_{Im}}$. The finite element model for the structure results in equations having the form $\mathbf{Z}\dot{\mathbf{d}} = \mathbf{f}$. The geometry, stiffness, mass, and damping characteristics are held in the impedance matrix, $\mathbf{Z}$. With the exception of the characteristics controlled by the parameters that will be updated, the impedance matrix is a known quantity. This matrix, along with the deformation mechanics that underlie the theory of the model, control the response the model predicts for a given load. The analytical and experimental models share the vector $\dot{\mathbf{d}}$. This consistency between the two models is used to update parameters in the FE model. Since both the analytical and experimental models can be solved separately for $\dot{\mathbf{d}}$, the formulation to update the parameters in the impedance matrix will largely be controlled by how $\dot{\mathbf{d}}$ is found. Three formulations were considered to bring the experimental and analytical models together through the common response vector $\dot{\mathbf{d}}$. These three model formulations are described in the following subsections.

3.6.1 Response Constrained By the FE Model

The node velocities can be solved for in the FE model, $\dot{\mathbf{d}} = \mathbf{Z}^{-1}\mathbf{f}$, and then substituted into the experimental model, the sum of squared error equation. Minimizing the resulting equation gives the lowest error consistent with behavior the analytical model allows. By solving the analytical model for $\dot{\mathbf{d}}$, the node velocities must take values that are allowed by the mechanics of the structural theory that underlies the FE model. Letting $\mathbf{Y} = \mathbf{Z}^{-1}$ and substituting into the squared error of Equation 3.36 results in

$$\Pi_{exp} = \mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Q}\mathbf{Y}\mathbf{f} - 2\mathbf{c}^{\mathrm{T}}\mathbf{Y}\mathbf{f} \tag{3.40}$$

The symbol $\mathbf{Y}$ is used for the mobility matrix in experimental vibrations references. Even though the matrix $\mathbf{Y}$ in this research is not exactly the same as that used in experimental work, the analogy still holds— it is the inverse of the impedance matrix that was defined earlier. Using a term from the field of optimization, Π_{exp} is the objective function. By minimizing this objective function, the parameters in the FE model can be updated. As the optimization algorithm attempts to find the minimum value for Π_{exp}, the parameters of the FE model, located in $\mathbf{Y}$, are changed. As the parameters change, the response, $\dot{\mathbf{d}}$, also changes. The constants that are not free to change are the measured data: the input forces located in $\mathbf{f}$, the velocity components($\mathrm{v}_{L_{Re}}$and $\mathrm{v}_{L_{Im}}$) at each SLDV sample location, the direction cosines, and the sample location information.

In this formulation, there must be trust in the deformation mechanics of the analytical model. Since equilibrium is enforced, the response must make physical sense in terms of this analytical model. Most of the errors in this model must be represented in the parameters chosen for update in order to achieve a good result. Otherwise, the model will be updated, but the changes will be made to rectify inadequacies that aren't included in the update parameters.

3.6.2 Response Constrained By Experimental Result

The second formulation considered solves for the minimum of the experimental squared error and then attempts to minimize the error in equilibrium of the FE model. The minimum in the experimental squared error occurs when the derivative of χ^2 with respect to $\dot{\mathbf{d}}$ is zero. This gives the solution $\dot{\mathbf{d}} = \mathbf{Q}^{-1}\mathbf{c}$. This is the experimental model solution. In contrast to the first formulation, the vector $\dot{\mathbf{d}}$ takes the value that corresponds to the best fit of the response surface to the experimental data. This value is set regardless of the mechanics that are assumed in the analytical model. The error in equilibrium in the FE model can be expressed as

$$\Pi_{equil} = \left(\mathbf{Z}\dot{\mathbf{d}} - \mathbf{f}\right)^{\mathrm{T}} \left(\mathbf{Z}\dot{\mathbf{d}} - \mathbf{f}\right)$$

Inserting the node velocities from the experimental model solution,

$$\Pi_{equil} = \left(\mathbf{Z}\mathbf{Q}^{-1}\mathbf{c} - \mathbf{f}\right)^{\mathrm{T}} \left(\mathbf{Z}\mathbf{Q}^{-1}\mathbf{c} - \mathbf{f}\right)$$

Here, the response, $\dot{\mathbf{d}}$, has been set by experimentally measured values. Only those parameters available to vary in the FE model can be changed to minimize the error in equilibrium.

This formulation, Π_{equil}, is much like the first formulation except on one important point— equilibrium in the FE model is not strictly enforced. Like the first formulation, any errors in the FE model must be accounted for in the update parameters. In contrast to the first formulation though, inadequacies in the analytical model can result in equilibrium not being satisfied. This unbalanced equilibrium can occur rather than the parameters being forced to values strictly due to unaccounted for errors in the analytical models.

3.6.3 Balance Experimental Error and FE Equilibrium Error

The final formulation attempts to balance the error in the experimental model and the analytical model. The response is set by neither the experimental model or the analytical model. This formulation is a composite function of the squared error in the experimental data and the FE equilibrium condition squared error:

$$\Pi_{composite} = w_1 \left(\dot{\mathbf{d}}^{\mathrm{T}}\mathbf{Q}\dot{\mathbf{d}} - 2\mathbf{c}^{\mathrm{T}}\dot{\mathbf{d}}\right) + w_2 \left(\mathbf{Z}\dot{\mathbf{d}} - \mathbf{f}\right)^{\mathrm{T}} \left(\mathbf{Z}\dot{\mathbf{d}} - \mathbf{f}\right) \tag{3.41}$$

The weights, w_1 and w_2, control what error will dominate. If, due to its size relative to the FE error, the experimental error dominates, the composite objective function becomes(approximately)

$$\Pi_{composite} \approx \left(\dot{\mathbf{d}}^{\mathrm{T}}\mathbf{Q}\dot{\mathbf{d}} - 2\mathbf{c}^{\mathrm{T}}\dot{\mathbf{d}}\right) \tag{3.42}$$

Minimizing this function results in the same solution as the second formulation. If the error in the equilibrium of the FE model dominates, the composite objective function becomes

$$\Pi_{composite} \approx \dot{\mathbf{d}}^{\mathrm{T}}\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\dot{\mathbf{d}} - 2\mathbf{f}^{\mathrm{T}}\mathbf{Z}\dot{\mathbf{d}} + \mathbf{f}^{\mathrm{T}}\mathbf{f} \tag{3.43}$$

The minimum for this equation occurs anywhere $\mathbf{Z}\dot{\mathbf{d}} = \mathbf{f}$. This is an underdetermined set of equations since the update parameters allow $\mathbf{Z}$ to vary. There is no unique solution. Clearly, the weights must be chosen so that one error does not dominate, but a balance occurs.

In this formulation, $\Pi_{composite}$, the weights are very important. If the FE error dominates, there is no unique solution since $\mathbf{Z}\dot{\mathbf{d}} = \mathbf{f}$ is underdetermined. If the experimental error dominates, the formulation degenerates into the second formulation. It is very difficult to chose the weights correctly since the relative error values will change as the optimization progresses.

3.6.4 Conclusions on the Formulations

In choosing the formulation to be used for the update algorithm in this thesis, the relative merits of each of the three formulations presented in this section were considered. The first formulation was chosen for the update algorithm in this research since it places the burden for errors on the analytical model and the parameters chosen for updating. If the updated model produces unsatisfactory results, there are two possible culprits(assuming the experimental data is sound and complete enough): the theory that underlies the analytical model, or the parameters chosen for update. For the remainder of this document, instead of writing Π_{exp}, the formulation will be represented simply as Π.

3.7 Objective Function Formulation for Multiple Tests

The experimental error function given in Equation 3.36 is for a single test. That is, the function represents data from a test with one forcing location and at one frequency. The model update formulation can be generalized to include multiple tests on the same structure. Indeed, when data from more than one test is used, the resulting model from the update should be superior. Altering the equation to include multiple tests

$$\chi^2 = W_1(\dot{\mathbf{d}}_\mathbf{1}{}^\mathrm{T}\mathbf{Q_1}\dot{\mathbf{d}}_\mathbf{1} - 2\mathbf{c_1}{}^\mathrm{T}\dot{\mathbf{d}}_\mathbf{1}) + W_2(\dot{\mathbf{d}}_\mathbf{2}{}^\mathrm{T}\mathbf{Q_2}\dot{\mathbf{d}}_\mathbf{2} - 2\mathbf{c_2}{}^\mathrm{T}\dot{\mathbf{d}}_\mathbf{2}) \\ + \cdots + W_n(\dot{\mathbf{d}}_\mathbf{n}{}^\mathrm{T}\mathbf{Q_n}\dot{\mathbf{d}}_\mathbf{n} - 2\mathbf{c_n}{}^\mathrm{T}\dot{\mathbf{d}}_\mathbf{n}) \quad (3.44)$$

where the subscripts for the matrices and vectors denote data for each different test. Each test could be a different excitation frequency or have a different excitation location. The data for different frequencies could be gathered during one experiment since the linear regressions described in Chapter 2 are capable of processing input and response signals that contain multiple frequencies. The W_i's are weights added to ensure that each test has equal importance in the minimization. This prevents data from one test dominating if its squared error is much larger than that of the other tests.

For tests where the structure's operating shape is assumed to be the same, such as performing the same test with the laser viewing position changed, the test data can be combined. For example, test 2 in Equation 3.44 could actually be made up of 2a and 2b, so that $\mathbf{Q_2} = \mathbf{Q_{2a}} + \mathbf{Q_{2b}}$ and $\mathbf{c_2} = \mathbf{c_{2a}} + \mathbf{c_{2b}}$. This is possible since the node velocities are assumed to be the same, so that

$$\dot{\mathbf{d}}_\mathbf{2}{}^\mathrm{T}\mathbf{Q_{2a}}\dot{\mathbf{d}}_\mathbf{2} + \dot{\mathbf{d}}_\mathbf{2}{}^\mathrm{T}\mathbf{Q_{2b}}\dot{\mathbf{d}}_\mathbf{2} = \dot{\mathbf{d}}_\mathbf{2}{}^\mathrm{T}(\mathbf{Q_{2a}} + \mathbf{Q_{2b}})\dot{\mathbf{d}}_\mathbf{2} = \dot{\mathbf{d}}_\mathbf{2}{}^\mathrm{T}\mathbf{Q_2}\dot{\mathbf{d}}_\mathbf{2}$$

The same model parameters are being updated for all the tests, but the FE model inserted into the objective function will be different for each test condition. This is due to the fact that the impedance matrix, **Z**, is a function of the frequency.

Also, the force may be different for each test and the correct one should be used. This makes the form of the objective function for multiple tests

$$\Pi = W_1(\mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Q}\mathbf{Y}\mathbf{f} - 2\mathbf{c}^{\mathrm{T}}\mathbf{Y}\mathbf{f})_1 + W_2(\mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Q}\mathbf{Y}\mathbf{f} - 2\mathbf{c}^{\mathrm{T}}\mathbf{Y}\mathbf{f})_2 + \cdots + W_n(\mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Q}\mathbf{Y}\mathbf{f} - 2\mathbf{c}^{\mathrm{T}}\mathbf{Y}\mathbf{f})_n \quad (3.45)$$

3.8 Choosing Parameters to Update

Choosing parameters to update in FE models requires physical insight and the use of engineering judgment. There is currently no systematic technique to choose parameters that require update in a model. Generally, the choice should involve deciding where the greatest uncertainties lie in the model. This will change from situation to situation. The number of parameters to update in a model should be kept small to ensure the problem is tractable. Once the parameters are chosen, they should be included in the update formulation in a form to which the experimental data is sensitive so that the updating is accurate. If the data is insensitive to a parameter, the value of the parameter may be changed haphazardly, with little regard to the information in the data. If the wrong parameters are chosen for update, the optimization will still attempt to minimize Π. New values for the parameters will be found, but these new values will not reflect an improvement in the model. At best, choosing parameters is difficult and should be done utilizing as much theoretical and experimental experience(prior knowledge) as possible.

3.9 Chapter Summary

This chapter centered on the development of an objective function whose minimization will result in an updated model

$$\Pi = \mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Q}\mathbf{Y}\mathbf{f} - 2\mathbf{c}^{\mathrm{T}}\mathbf{Y}\mathbf{f}$$

The components of $\mathbf{f}$, $\mathbf{Q}$, and $\mathbf{c}$ contained measurements recorded during the vibration test of the structure being modeled. The matrix $\mathbf{Y}$ is the inverse of the impedance matrix from the FE formulation for harmonic response modeling. The objective function Π results from combining the experimental model and the FE model, which can occur because of the common representation for $\dot{\mathbf{d}}$ in both models.

The next chapter covers some computational issues involved in minimizing Π which results in an updated FE model. Finding the minimum of Π is essentially finding the solution to a nonlinear least squares problem. While the experimental model, χ^2, represented a linear least squares problem, Π is nonlinear due to the update parameters in the mobility matrix, $\mathbf{Y}$, which generally appear as nonlinear terms. The update problem could have been posed as a linear least squares problem with the constraint that $\mathbf{Z}\dot{\mathbf{d}} = \mathbf{f}$. Instead the vector of node velocities were solved for, $\dot{\mathbf{d}} = \mathbf{Z}^{-1}\mathbf{f} = \mathbf{YF}$. This solution was placed in the experimental model, resulting in Π. This was beneficial since many of the variables have been eliminated, leaving just the update parameters in the FE model as variables. Also, unconstrained problems are generally easier to solve. The next chapter includes a discussion about performing tests to validate that the solution of the nonlinear least squares problem does result in an improved FE model.

Chapter 4

Calculating and Validating the Model

In the previous chapter, a function was formulated to facilitate updating an analytical model based on experimental data. This function is

$$\Pi = \mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Q}\mathbf{Y}\mathbf{f} - 2\mathbf{c}^{\mathrm{T}}\mathbf{Y}\mathbf{f}$$

It was developed by inserting the analytical model, a finite element model, into an expression of the experimental squared error, the experimental model χ^2. The result is an expression of the error between the predictions of the analytical model and the measured velocity values. In fact, the experimental model can be thought of as simply a method to condense the experimental data into the node degrees of freedom that are present in the finite element model. The experimental model was originally introduced as something more than a data condensing method since it can be solved for a velocity field on its own. In most of the literature available on updating models, the experimental data must be expanded since there are usually far more FE degrees of freedom than measurements. The scanning capability of the SLDV allows many more measurement locations on a structure than a test with accelerometers, so here the opposite situation occurs.

The parameters in the model will be updated by finding the minimum of the function Π. If the parameters selected for the model update are truly where the error in the model was and the mechanics of the model can represent the motions that actually occurred during the test, the updated model will be an improvement. This chapter covers some of the issues in the process used in finding the minimum value for Π. Choosing the routine used for the optimization involves considering the objective function and the relative merits of applying different routines to the specific problem in this research. Analytically calculated derivatives were used in the optimization to make the numerical routine more efficient. Update parameters must be scaled for the optimization so that the data is equally sensitive to each parameter; otherwise, the routine has little chance of finding the optimum solution. Following the optimization, some methods of validation of the model must be applied to ensure it is *better*. Some techniques to check the validity of the updated mode are presented at the end of this chapter.

4.1 Optimization Routine

The field of optimization addresses the problems of finding the minimum of an objective function. Many times an optimization algorithm is used as a design tool to find the best values for a system's parameters where the objective function is a measure of the system's performance. Usually there are constraints on allowable values for the parameters and on the objective function. For example, the depth of a beam may be limited by space considerations or the stress at a joint may be limited to stay below some maximum allowable value. The methods of optimization have been adapted in this research for the updating algorithm.

Arora (1989) defines the standard optimization problem as:

> Find a vector of design variables $\mathbf{x}$ to minimize the cost function $f = f(\mathbf{x})$ subject to the equality constraints $h_j = 0$ for $j = 1$ to p and the inequality constraints $g_i \leq 0$ for $j = 1$ to m.

Problems that don't have constraints are unconstrained optimization problems and those that are constrained are called constrained optimization problems. If the objective function, f, and the constraints, g_i and h_j, are linear in x, then the problem is called a linear programming problem. If they are nonlinear in x, then the optimization problem is a nonlinear problem. These are important distinctions since there are specialized solution methods that work well for different types of problems.

The objective function Π for the updating algorithm is in general a nonlinear function. The optimization variables are the parameters of the finite element model that are located in $\mathbf{Z}$(and its inverse $\mathbf{Y}$). The problem is unconstrained. Strictly speaking, most of the parameters in the finite element formulation should be constrained to stay in a region of physically reasonable values(i.e.- positive). It is expected that the final solution for these variables will be reasonably close to the initial estimates, so these constraints are left out. The optimization problem for the update algorithm could also be defined as:

$$\text{Find the minimum value for } \chi^2 \text{ subject to the constraints } \mathbf{Z}\dot{\mathbf{d}} = \mathbf{f}.$$

There are two reasons that the formulation with Π is better. The first is that unconstrained optimization problems are much easier to solve than constrained problems. Secondly, by solving the finite element equations for $\dot{\mathbf{d}}$ and inserting them into the cost function, a large number of the variables are removed, making the problem easier to solve.

There are two methods to find the solution of an optimization problem: methods that attempt to find solutions that satisfy optimality criteria, and iterative solutions that search the design space. Optimality criteria are conditions a function must satisfy at its minimum point. They are equivalent to finding the solution of $\frac{dy}{dx} = 0$ to find the minimum of y, a function of x. The optimality criteria for functions of more than one variable and constrained functions, called Kuhn-Tucker conditions, are more complex but the ideas are the same. In the sections on the linear regressions and on the experimental model solution, the solutions were the result of satisfying optimality criteria. The methods that find the optimum by searching the design space do so iteratively, starting at some initial estimate of the optimum point and then steadily moving closer to the true optimum. They terminate when some measure of improvement indicates that the optimum has been found.

For several reasons, in practice numerical search methods are used more often than methods to satisfy optimality criteria. If the objective function or the constraints are very nonlinear, finding solutions to the Kuhn-Tucker conditions is very difficult. Solutions to optimality criteria are also difficult to find when there are many design variables. Lastly, it can be difficult or even impossible to get an equation for the solution of the criteria explicitly in terms of the design variables. Since the objective function is nonlinear for the update algorithm and extracting the variables into an explicit form would be very difficult, a numerical search method will be used to find the minimum of Π.

There are many numerical search methods to find the solution for nonlinear unconstrained optimization problems. They all begin with some initial design and change it iteratively to find the optimum. They can be loosely classified by the level of information required from the cost function(Π) here at each iteration to find the solution: those that use only function evaluations, those that use first derivative information, and those that use second derivative information.

The search methods for unconstrained nonlinear optimization that use only function evaluations are only appropriate when the function is not differentiable or if it is so nonlinear that there is no trust in numerical derivatives. The sequential simplex method and the conjugate search direction method, discussed by Arora, are examples of algorithms using only function evaluations. Other methods that use only function evaluations as they iterate are simulated annealing and genetic algorithms. These two algorithms, which Haftka and Gurdal (1991) provide an overview of, iterate based on a random selection process guided by probabilistic decisions. They are better suited to situations where a global minimum is sought in the face of many local minimums, a situation that arises when variables are constrained to certain discrete values. Since the derivatives of the cost function Π will be available, search methods that use this gradient information are preferred.

Search methods that use derivative information to find new values for the design variables $\mathbf{x}$ at each iteration generally do so using the following equations:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$$

$$\Delta\mathbf{x}_k = \alpha_k d_k$$

where $\mathbf{x}_k$ is the current value for the variables, $\mathbf{x}_{k+1}$ is the value at the next iteration, and $\Delta\mathbf{x}_k$ is the change in the variables. The change is found by first calculating d_k, a desirable direction to move in the design space. Finding this direction requires the cost function value and its gradient. Generally, d_k is chosen so that a move in that direction will result in a lower cost function value. The factor α_k determines the size of the move in the search direction. Finding $\Delta\mathbf{x}_k$ is complicated by the presence of constraints, which must be included in the decisions. This is one of the reasons solutions to constrained problems are harder to find than solutions to unconstrained problems. Finding α_k involves finding the minimum of the cost function along a curve. Finding this minimum is called one dimensional minimization since the objective function f has been reduced to a function of one variable, $f(\alpha)$. Usually α_k is calculated numerically and there are many methods to do so. Some of the methods only use the values of $f(\alpha)$ to estimate its minimum, while others fit a polynomial to values of $f(\alpha)$ and calculate the minimum of this polynomial. Using polynomials is more efficient when gradients of the cost function are available. The difference between the search methods using first derivative information and those using second derivatives is the way they use this information to find the search direction, d_k.

The steepest descent method and the conjugate gradient method, discussed by Arora (1989), are algorithms that use the gradient of the cost function to find the search direction. While the conjugate gradient method is preferred over the steepest descent method since it converges to a solution faster, methods that use second derivative information are preferred over both steepest descent and conjugate gradient since they converge at an even greater rate.

Newton's method is a nonlinear unconstrained optimization algorithm that uses second derivative information to calculate the search direction at each iteration.

It has several drawbacks. The search direction that is calculated is not guaranteed to be in a direction of descent of the cost function, so convergence is not guaranteed. Also, it requires calculation of second derivatives, which can be computationally expensive, especially if finite differences are used to calculate them. When Newton's method works, it converges much faster than the methods using first derivative information, so methods known as quasi-Newton methods were developed to overcome its limitations. The quasi-Newton methods use first derivative information to estimate the second derivatives. This is computationally more efficient than actually calculating them. Also, the methods to estimate the second derivatives are tailored so that the search direction is guaranteed to be in a direction of descent for the cost function at every iteration. The BFGS(Broyden-Fletcher-Goldfarb-Shanno) method to estimate the second derivatives is the most popular since it has been proven to be the most effective. See Arora for details of this algorithm.

Since the cost function for the update algorithm, Π, is nonlinear, and the first derivatives are readily available, the quasi-Newton method is preferred to find its minimum. For the work in this research, a quasi-Newton optimization routine called **fminu** that is supplied with Matlab was used. The **fminu** routine uses the BFGS method by default to estimate the second derivatives using first derivatives and a polynomial search method for the one dimensional minimization at each iteration. Further information about **fminu** can be found in Grace (1992). The next section shows how the first derivatives were supplied so that finite differences were not needed to calculate the gradient.

Since the optimization procedure is not guaranteed to find the global optimum, the routine should be run several times starting from different initial points for the parameters. Several local minimum values for Π may be found which must be compared to decide which is best.

4.2 Analytical Derivatives

The convergence rate of the optimization routine used in this algorithm can be greatly increased by supplying the gradients of the cost function so that they don't have to be calculated using a finite difference approximation. This removes the need to compute them, which can be computationally expensive, and ensures that the derivatives are accurate(so long as the supplied derivatives are accurate). Haftka and Gurdal (1991) provide a comparison of calculating derivatives using finite differences and supplied analytical derivatives, supporting the use of the analytical derivatives.

The variables in the cost function are all located in the FE mobility matrix $\mathbf{Y}$, which is the inverse of the impedance matrix from the FE model. The FE equation can be used to derive the the derivatives of the cost function with respect to any variables that appear in the impedance matrix. Starting with the experimental model before the FE equations are inserted and taking its derivative with respect to a variable x shows that $\frac{\partial \dot{\mathbf{d}}}{\partial x}$ is needed

$$\begin{aligned} \chi^2 &= \dot{\mathbf{d}}^{\mathrm{T}}\mathbf{Q}\dot{\mathbf{d}} - 2\mathbf{c}^{\mathrm{T}}\dot{\mathbf{d}} \\ \frac{\partial \chi^2}{\partial x} &= 2\dot{\mathbf{d}}^{\mathrm{T}}\mathbf{Q}\frac{\partial \dot{\mathbf{d}}}{\partial x} - 2\mathbf{c}^{\mathrm{T}}\frac{\partial \dot{\mathbf{d}}}{\partial x} + \dot{\mathbf{d}}^{\mathrm{T}}\frac{\partial \mathbf{Q}}{\partial x}\dot{\mathbf{d}} - 2\frac{\partial \mathbf{c}^{\mathrm{T}}}{\partial x}\dot{\mathbf{d}} \end{aligned} \tag{4.1}$$

Derivatives of the matrix $\mathbf{Q}$ and the vector $\mathbf{c}$ are included for completeness. If the variable x happens to be a parameter used in the transformation to the surface where the laser makes its measurements(Section 3.2), these derivatives will be needed. Taking the same derivative of the FE equations gives $\frac{\partial \dot{\mathbf{d}}}{\partial x}$:

$$\begin{aligned} \mathbf{Z}\dot{\mathbf{d}} &= \mathbf{f} \\ \frac{\partial \mathbf{Z}}{\partial x}\dot{\mathbf{d}} + \mathbf{Z}\frac{\partial \dot{\mathbf{d}}}{\partial x} &= 0 \\ \frac{\partial \dot{\mathbf{d}}}{\partial x} &= -\mathbf{Z}^{-1}\frac{\partial \mathbf{Z}}{\partial x}\dot{\mathbf{d}} \end{aligned} \tag{4.2}$$

These derivatives are easy to take because of the form of the finite element equations. Derivatives of mode vectors, by comparison, are very difficult to derive and calculate. Remembering that the substitution $\mathbf{Y} = \mathbf{Z}^{-1}$ was made and that $\mathbf{Yf} = \dot{\mathbf{d}}$, then Equation 4.2 becomes

$$\frac{\partial \dot{\mathbf{d}}}{\partial x} = -\mathbf{Y}\frac{\partial \mathbf{Z}}{\partial x}\mathbf{Yf} \tag{4.3}$$

Inserting this equation into Equation 4.1 gives the derivative of the cost function with respect to x:

$$\frac{\partial \Pi}{\partial x} = -2\mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{QY}\frac{\partial \mathbf{Z}}{\partial x}\mathbf{Yf} + 2\mathbf{c}^{\mathrm{T}}\mathbf{Y}\frac{\partial \mathbf{Z}}{\partial x}\mathbf{Yf} + \mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\frac{\partial \mathbf{Q}}{\partial x}\mathbf{Yf} + 2\frac{\partial \mathbf{c}^{\mathrm{T}}}{\partial x}\mathbf{Yf} \tag{4.4}$$

In order to utilize this equation in a numerical computer routine, the matrices $\mathbf{Y}$ and $\frac{\partial \mathbf{Z}}{\partial x}$ must be available.

Although the second derivatives can be calculated in a similar manner to what is shown here, the optimization routine used by this algorithm, the quasi-Newton method, requires only first derivative information. The second derivative information is estimated using the first derivatives. These first derivatives are exact within the precision of the calculation of Equation 4.4. This greatly improves the convergence rate of the optimization algorithm to the best solution. In fact, in the process of performing the simulations presented in the next chapter, it was learned that the optimization routine would not converge to a solution consistently if the exact analytical derivatives were not supplied.

4.3 Scaling the Variables for the Optimization Routine

Due to the disparity in the magnitude of some of the variables in the optimization, rescaling the variables is necessary. Otherwise, the derivatives of the objective function, which are used by the optimization routine to choose a search direction(d_k), would have wildly varying values that would cause one of the variables to dominate. Ideally, scaling would result in:

- The variables taking values between 0.1 and 10,

- The objective function gradients taking values between 0.1 and 10, and,

- The objective function taking values between 0.1 and 10.

Keeping all of these values between 0.1 and 10 ensures that they remain within two orders of magnitude of one another. This hopefully eliminates the loss of accuracy that occurs when math operations are performed with terms that are widely different in magnitude.

Unfortunately, all of the goals listed here can not be attained at the same time. For the numerical optimization routine, the most important goal is to have the objective function gradients taking values between 0.1 and 10. This goal is met by replacing each variable with a new variable that is scaled by the old variable's derivative at the starting point for the optimization. If the initial guess for one of the FE parameters is x_0, then the value of the derivative of the objective function at the starting point is $\left.\frac{\partial \Pi}{\partial x}\right|_{x_0}$. The variable x is replaced by a new variable whose relationship to x is $y = x \left.\frac{\partial \Pi}{\partial x}\right|_{x_0}$. This technique of normalization works since the parameters in the FE model are in equations which are essentially polynomials. The derivative of the objective function with respect to the new scaled variable at the initial point is one. Since the optimum value for the variables are not expected to differ from the initial guesses by more than an order of magnitude, the gradient values stay between 0.1 and 10 as the optimization routine progresses. Even if the initial guess is off by more than an order of magnitude, as it might be for damping, the scale factor should work adequately and is definitely an improvement over the situation without one. Scaling in this manner has the effect of making the data sensitive to changes in each parameter, which is important if the parameters are to be updated with any confidence.

As an example, consider the system of equations that results from the model of the rod structure shown in Figure 4.1. This structure is modeled with three equal length rod elements which results in the global stiffness matrix

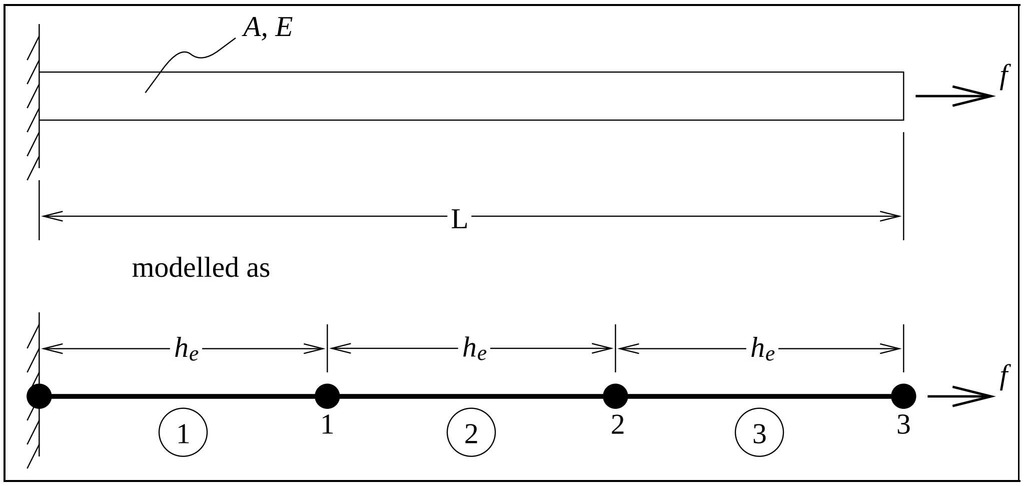

Figure 4.1: Rod Structure Modeled With Three Elements

$$\mathbf{K} = \frac{AE}{h_e}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \tag{4.5}$$

where A is the cross-section area for the rod, E is the elastic modulus, and h_e is the element length. If the length of the whole structure is thirty-six inches, then h_e is 12 inches. The parameters A and E will be the optimization variables. Normally, the impedance matrix, $\mathbf{Z}$, is composed of a mass matrix, a stiffness matrix, and a damping matrix. For simplicity, consider a case where $\mathbf{Z} = \mathbf{K}$, where $\mathbf{K}$ is the matrix given by Equation 4.5. The derivative of the cost function(restating Equation 4.4) is

$$\frac{\partial \Pi}{\partial x} = -2\mathbf{f}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Q}\mathbf{Y}\frac{\partial \mathbf{Z}}{\partial x}\mathbf{Y}\mathbf{f} + 2\mathbf{c}^{\mathrm{T}}\mathbf{Y}\frac{\partial \mathbf{Z}}{\partial x}\mathbf{Y}\mathbf{f} \tag{4.4}$$

Since a transformation is not used in this model, $\frac{\partial \mathbf{Q}}{\partial x} = 0$ and $\frac{\partial \mathbf{c}}{\partial x} = 0$, and those terms are not included in the derivative. For this example, let the other terms in the equation have the following values:

$$\mathbf{Q} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{bmatrix} \qquad \mathbf{c} = \begin{Bmatrix} 4 \\ 5 \\ 6 \end{Bmatrix} \qquad \mathbf{f} = \begin{Bmatrix} 0 \\ 0 \\ 1000 \end{Bmatrix}$$

The derivatives of the cost function with respect to the parameters require the derivatives of the impedance matrix with respect to those parameters. Those derivatives are

$$\frac{\partial \mathbf{Z}}{\partial A} = \frac{E}{h_e}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \qquad \frac{\partial \mathbf{Z}}{\partial E} = \frac{A}{h_e}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

The derivative of the cost function will also require the inverse of the impedance matrix, $\mathbf{Y}$:

$$\mathbf{Y} = \mathbf{Z}^{-1} = \frac{h_e}{AE}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix}$$

Let the initial values for the parameters be $A_0 = 0.125$ in^2 and $E_0 = 30e6\ \frac{\text{lb}}{\text{in}^2}$. Given these initial points for the parameters, the values for the derivatives are

$$\left.\frac{\partial \Pi}{\partial A}\right|_0 = 1.630$$
$$\left.\frac{\partial \Pi}{\partial E}\right|_0 = 6.793e - 9$$

Let the values for the parameters at the next iteration be $A_1 = 0.13$ and $E_1 = 28e6$. At this point, the values for the derivatives are

$$\frac{\partial \Pi}{\partial A} = 1.615$$
$$\frac{\partial \Pi}{\partial E} = 7.497e - 9$$

At both the first and second points, the derivative with respect to the parameter E is much smaller than the derivative with respect to A. A search direction based on these gradient vectors would point along a direction of constant value for E since the derivatives indicate that it has little effect on the value of Π. During the course of the optimization, it is very unlikely that E would change. If it did, the derivative $\frac{\partial \Pi}{\partial E}$ is so small in comparison to the derivative for A that the result would probably be unsatisfactory. This is due to the problem in the relative scales of the parameters. The scaling procedure proposed earlier will fix this problem. The new scaled variables would be

$$y_1 = A \left.\frac{\partial \Pi}{\partial A}\right|_0 \qquad y_2 = E \left.\frac{\partial \Pi}{\partial E}\right|_0$$

For the new optimization problem defined with the variables y_1 and y_2, the derivatives of Π require the values of

$$\frac{\partial \mathbf{Z}}{\partial y_1} = \frac{y_2}{\left(\frac{\partial \Pi}{\partial A}\Big|_0\right)\left(\frac{\partial \Pi}{\partial E}\Big|_0\right) h_e} \begin{bmatrix} -2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\frac{\partial \mathbf{Z}}{\partial y_2} = \frac{y_1}{\left(\frac{\partial \Pi}{\partial A}\Big|_0\right)\left(\frac{\partial \Pi}{\partial E}\Big|_0\right) h_e} \begin{bmatrix} -2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

At the initial point, $y_{1_0} = 0.2038$ and $y_{2_0} = 0.2038$, the derivatives are

$$\frac{\partial \Pi}{\partial y_1}\Big|_0 = 1 \qquad \frac{\partial \Pi}{\partial y_2}\Big|_0 = 1$$

At the next iteration point, $y_{1_1} = 0.2120$ and $y_{2_1} = 0.1902$, the derivatives are

$$\frac{\partial \Pi}{\partial y_1}\Big|_1 = 0.9904 \qquad \frac{\partial \Pi}{\partial y_2}\Big|_1 = 1.104$$

These derivatives do not exhibit the huge disparity that was present when the original variables, A and E, were used. As the optimization algorithm approaches a solution, all of of the derivatives will approach zero, but they will do so because an optimal point is close, not because bad scaling placed some of the derivatives near zero initially. Once the optimization routine has reached a solution, the original parameters are found

$$A^* = \frac{y_1^*}{\frac{\partial \Pi}{\partial A}\Big|_0} \qquad E^* = \frac{y_2^*}{\frac{\partial \Pi}{\partial E}\Big|_0}$$

In the process of creating a computer program to test the algorithm presented in this research, it was learned that the optimization routine would not consistently provide good results without the scaling routine shown here. It is absolutely necessary to scale the parameters for the update algorithm to have a chance to work.

4.4 Validating the Model

There are many measures available to test the validity of a model. In the area of correlating models for structural response, there are several measures available to compare experimental response vectors to vectors from analytical predictions. Since the updating algorithm presented here is based on a least squares formulation, many useful statistical tests are available. These tests for the validity of the model are based on statistical measures for regression. Tests using the response vectors and tests based on regression statistics can both be used in the framework of the update algorithm shown in this research.

4.4.1 Response Vector Tests

Correlation measures that were developed for vibrations use the response vectors to measure the agreement, or lack of it, between analytical predictions and test results. Some of these correlations were developed for modal vector comparisons, but they are equally valid for vectors of the operating shape at a given frequency.

The Frequency Response Assurance Criteria(FRAC), demonstrated by Heylen and Lammens (1996), compares the shape of the response vectors without taking into account the overall scale differences between the test and analytical vectors. The FRAC is defined as

$$FRAC = \frac{\left|\phi_{exp}^{\mathrm{T}}\phi_{FE}\right|^2}{(\phi_{exp}^{\mathrm{T}}\phi_{exp})(\phi_{FE}^{\mathrm{T}}\phi_{FE})} \tag{4.6}$$

where | | signifies the modulus of a complex number, and the transpose is taken to be the conjugate transpose. The FRAC is defined equivalently to the Modal Assurance Criteria(MAC), introduced by Allemang and Ewins (1983). The difference is that the FRAC matches the operating shape response vectors at a certain frequency, while the MAC correlates mode shapes.

Mitchell (1998) proposed the Inverse Modal Assurance Criteria(IMAC) as a modification to the MAC. The inverse of the responses are used so that the correlation is more sensitive to differences between vectors around the nodes(here defined as areas of no vibration, not finite element nodes). This measure came from the observation that subtle changes in the shape are noticeable in the areas of the nodes.

The Coordinate Model Error Function(CoMEF), given by Catbas et al. (1998), gives the weighted error between the analytical and experimental vectors. Relatively low weights are given to areas where the response is low since the data may be near the experimental noise floor. This correlation factor, like the FRAC, gives information on differences in the shape of the response.

The Coordinate Modal Assurance Criteria(CoMAC), presented by Lieven and Ewins (1988), shows which spatial locations contribute to a low value for the FRAC. The low values may be due to magnitude differences or phase differences.

The scaling factor is the constant the experimental data would have to be multiplied by to get amplitudes that are comparable to the analytical model prediction amplitudes(Lindholm 1996). This measure is only useful when there is confidence that the shape is fairly well matched between the two vectors. Otherwise the the value for the scaling factor will have little meaning. If the model is overpredicting in some areas, and underpredicting in others, the scaling factor will not give good information.

All of the measures presented here could be used in several ways in the context of the model update algorithm presented in this research:

1. Solve the experimental model on its own. Let the test vector be defined as

$$\phi_{measured} = \left\{ \begin{array}{c} \mathrm{v}_{L_{Re_1}} + i\mathrm{v}_{L_{Im_1}} \\ \mathrm{v}_{L_{Re_2}} + i\mathrm{v}_{L_{Im_2}} \\ \vdots \\ \mathrm{v}_{L_{Re_n}} + i\mathrm{v}_{L_{Im_n}} \end{array} \right\}$$

   where each component of the vector is the response at a sample location. Let ϕ_{exp} be the response at each measurement location as predicted by the experimental model solution. The correlation measures will show how well the shape functions used in the experimental model can fit the data.

2. Solve the experimental model for $\dot{\mathbf{d}}_{exp}$, and let

$$\phi_{exp} = \dot{\mathbf{d}}_{exp_{Re}} + i\dot{\mathbf{d}}_{exp_{Im}}$$

   Solve the analytical finite element model before it has been updated and let

$$\phi_{FE} = \dot{\mathbf{d}}_{FE_{Re}} + i\dot{\mathbf{d}}_{FE_{Im}}$$

The different correlation tests using these two vectors will show how well the model performs before it is updated. This could also be accomplished using the response at each sample location, which could provide more information about the model's performance before it is updated. The usefulness of the extra information would have to be weighed against the effort required to construct the vectors for so many points.

3. The same tests could be performed after the model has been updated. This would provide information about the quality of the new model. These measures could also be used to rank several candidate models.

4.4.2 Regression Tests

In the same way that the time series regressions shown in Chapter 2 are multiple linear regressions, the formulation for the experimental model is a multiple linear regression. The update formulation is a nonlinear regression. There are a number of tests that arise from the statistics of regressions that can be used to validate the resulting models. The following tests are applicable to linear regression models if the error on the data is distributed normally. They are also applicable to the nonlinear regression if an extra condition is satisfied— the confidence interval for the parameters must be limited to a region where the model can be approximated as linear.

1. Cross-validation can be used to protect against the possibility of overfitting the model to the data from the test. This will ensure that the model will work in general, not just to predict for the conditions of the test. Cross-validation is performed by randomly splitting the data in half following a test, and fitting the model to half the data. Then, the resulting model is used to predict the other half of the data. If the model predicts the other half of the data well, it has passed the test, and a final model should be fit to the entire set of data. If the model doesn't predict well, the model should probably be simplified. For fits with only the experimental model and fits for the nonlinear regression to update the analytical FE model, the data could be split by location on the structure

for one test case, or by test condition(force location or frequency). Following the regression, highly influential observations should be deleted and the model refit.

2. The model can be tested to show whether a significant amount of the variability in the dependent variable is explained for by the model. This test involves looking at the ratio of the error in the dependent variable explained by the model to the error in the dependent variable about the model. This test could be performed on the linear regressions for the frequency domain representation for the velocities at each sample position. It could also be performed on the solution produced by the experimental model and the solution produced by the FE model.

3. Tests similar to the full model test described in item 2 could be used to see if a parameter should be added or eliminated. These tests are called partial F-tests and are used to implement procedures like stepwise regression to find an appropriate model. These tests could be used to add or remove elements in an experimental model. They could also be used to add or remove physical parameters for update in the analytical FE model.

4. There are a series of measures that can be used to quantify the quality of the prediction of the candidate models. One simple method is to examine the residuals at each of the sample locations. If the residuals are small compared to the values for the response, the prediction of the model is good. The coefficient of determination, R^2, determines the percentage of the variation in the velocity that is accounted for by the model. It always increases by adding additional regressor variables, regardless of whether they are of any use or not. The adjusted R^2 corrects for this by falling in value when useless regressor variables are added. Lastly, the C_p statistic is a compromise of not having enough variables and having too many.

The tests in the third and fourth items in the previous list both provide ways to choose from among several candidate models.

If the measurement errors can not be assumed to be distributed normally, Monte Carlo simulations may be used to get confidence intervals on the parameters. This allows inferences to be made to quantify the quality of the fit.

Once the model has been calculated by the update routine, the results should be examined to ensure that the model is *valid.* As was shown, there are several ways to validate the model. A good use of prior knowledge is to examine the values of the updated parameters and compare them to their initial estimates. If the parameters no longer make sense physically, then the whole point of using a physically based model has been lost. It is probably best to use several of the validation techniques listed here. The more tests an updated model passes, the more confidence that can be placed in it. The development of a methodical approach to model validation using some of the techniques presented here would be a significant addition to FE model updating methods. The algorithm presented in this research, since it is based on a least squares method, lends itself to many validation techniques.

Chapter 5

Simulations Demonstrating the Algorithm

Several simulations were run to demonstrate the performance of the update algorithm in the face of different kinds of errors. The first simulation shows how the algorithm reacts to differing levels of error in the laser's velocity measurements. This kind of random error is present in all experimental measurements. The second simulation shows how a systematic error in the laser's position can affect the optimization solution. It is important that the algorithm tolerate some error in the laser's position, since the exact location can never be known. The last simulation shows how the algorithm can be adversely affected when the wrong parameters are selected as the update variables.

Because these simulations were performed using experimental data that came from a structure that exists only in a computer program, there is a limit to what can be learned from them. However, they do demonstrate that simulations run before a test is performed can show how the update model for a specific structure can be expected to perform.

5.1 Generating *Experimental* Data

For all three of the update algorithm tests presented in this chapter, the process of getting experimental data is simulated using a computer program. By simulating the experiments, different cases could be tested quickly and the specific sources of error could be controlled. The response of a cantilever beam was simulated using a finite element model with fourteen beam elements.

The structure simulated is shown in Figure 5.1. The beam had a rectangular cross-section with width b and height h. The length of the beam was L. The values of these dimensions are shown in the figure. The beam was excited at its free end by a harmonically varying force at 200 Hertz. The force acted in the z-direction and had a 22 N amplitude so that the beam vibrated in the x-z plane. Only the transverse motions associated with beam theory were examined. The beam was made of the aluminum alloy 2014-T6. The properties relevant to the dynamics of the beam that were examined are:

density, $\rho = 2800 \ \frac{\mathrm{kg}}{\mathrm{m}^3}$
elastic modulus, $E = 75$ GPa
length of the beam, $L = 1.381$ m
beam cross-section width, $b = 1.27$ cm
beam height, $h = 8.255$ cm
mass proportional damping constant, $\alpha = 0$
stiffness proportional damping constant, $\beta = 1.4e - 5$

The material property values were taken from Beer and Russell (1992). The values for the dimensions and material properties listed here are the known exact values for this structure and were used in the FE model to produce the *experimental* data. Knowing the exact parameter values is a luxury that will not exist for an actual structure, but a necessary luxury in order to test the algorithm. By knowing the exact values, it was possible to gauge the performance of the update algorithm and the quality of the solution.

A FE model of the structure based on Bernoulli-Euler beam theory reproduced

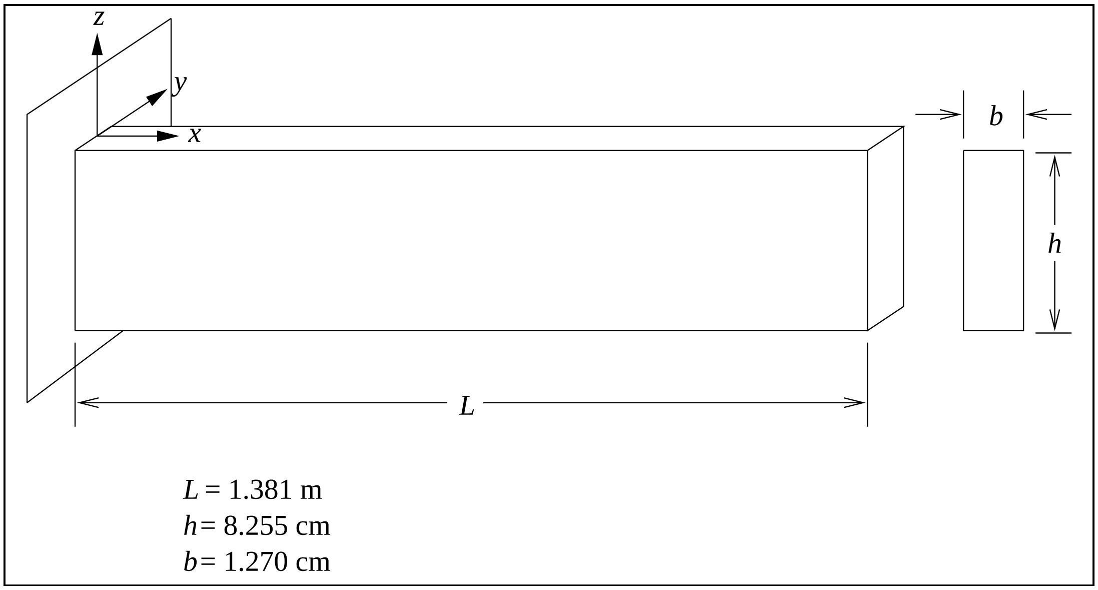

Figure 5.1: Cantilever Beam Structure Used For Simulations

the beam's response to a harmonically varying force, and a computer program simulated the laser sampling points along the beam's length. The *experimental* data that was generated by the program was in the same form as data that would be delivered from an actual test, with the velocity components in frequency domain form for each of the sample locations. For the simulations presented here, the data generating program made four evenly spaced velocity samples in each of the fourteen elements used to simulate the beam. The SLDV was placed 0.35 meters above the beam and 2 meters from the beam's fixed end(referring to the coordinate system shown in Figure 5.1, $x_l = 2.0$ m, $z_l = 0.35$ m).

In order to verify the accuracy of the FE model used to generate the experimental data, the exact analytical solutions for the natural frequencies for the beam were compared to those found by the finite element model. The exact solution was taken from Blevins (1979). Table 5.1 shows that the FE model with fourteen beam elements predicts natural frequencies for the first five modes that match the exact solutions to at least three significant digits. When using finite elements to model dynamic response, the model should have enough elements to represent the response accurately. According to Cook et al. (1989), to accurately model dynamic response, the model should have converged natural frequencies at least three times as high as

Table 5.1: Comparison of Exact Natural Frequencies to FE Model Solution

Mode	Exact Natural Frequency, Hz	FE Natural Frequency, Hz
1	36.19	36.19
2	226.8	226.8
3	635.0	635.1
4	1244	1243
5	2057	2058

the highest excitation frequency. For the simulations the structure was excited at 200 Hertz, so according to this criteria the excitation was well within the range that the model with fourteen elements is accurate.

5.2 Candidate FE Model

The candidate FE model that was updated is based on Bernoulli-Euler beam theory. The model has fourteen elements so that, as shown in the previous section, it can accurately model the response at 200 Hz. Whenever attempting to update a FE model, it is essential that it have enough elements to accurately model the response from which the test data originates. For the simulations presented here, it was known that fourteen elements were required. In practice, the model should be converged before updating is attempted.

The model's impedance matrix, $\mathbf{Z}$, is a function of the excitation frequency, and the mass, stiffness, and damping matrices. These matrices are functions of several parameters, so that $\mathbf{Z} = \mathbf{Z}(\omega,\ \rho,\ E,\ L,\ b,\ h,\ \alpha,\ \beta)$. In the simulations presented in Sections 5.3 and 5.4, E and h were the parameters chosen for update, and the rest of the parameters' assumed values used in the candidate model were exactly the same as those used to generate the experimental data. In Section 5.5, E is the update parameter, but the assumed values for the other parameters were incorrect. In other words, they were not the same as those used to generate the experimental data.

For the simulations presented in the following sections, the model used to create the *experimental* data and the model that is updated are both FE models based on beam theory, so the dynamics represented in the data can be reproduced exactly by the candidate FE model. In a test on a real structure, there is no guarantee that the dynamics represented in the test data can be reproduced by the model being updated. It is incumbent upon the user to ensure that the candidate model to be updated can adequately represent the actual response.

5.3 Random Error On the Data

The first simulation's purpose was to demonstrate the performance of the algorithm in face of random error on the experimental data. The random error was introduced by putting some error on the regressed velocity parameters, $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$, for each sample point. The error applied to the parameters has a normal distribution. In an actual test, the error would be present on each sample in the time series data used in the regression for each sample location. Random error with a normal distribution on the time series data would result in random, normally distributed error in the regression coefficients. Provided the distribution of the error on the time series data is truely normal, modeling the error in the regression coefficients in this fashion is consistent. For the simulation in this section, this was the assumption. The error was added using the Matlab command **randn**. This command generates numbers whose distribution in the long run has a mean of zero and variance of one. These random numbers were scaled so that the appropriate level of error was added to the velocity components.

The form of the added error was

$$\mathrm{v}_{L_{Re}} = \mathrm{v}_{L_{Re_o}} + \mathrm{v}_{L_{Re_o}} \times \mathbf{randn}/100 \tag{5.1}$$

where $\mathrm{v}_{L_{Re_o}}$ is the exact regression value and $\mathrm{v}_{L_{Re}}$ is the value with added error that was used in this simulation. In the form of Equation 5.1, error is added that has a variance that is 1 percent of the value of the actual regression component. Since the function **randn** has a variance of one, $\sigma^2_{randn} = 1$, its standard deviation, σ_{randn}, is also one. Ninety-nine percent of the values of a normally distributed variable will fall roughly within three standard deviations of the mean value for the variable.

Consequently, ninety-nine percent of the numbers produced by **randn** should be in the range -3 to 3. Using Equation 5.1 to add error to the velocity values should result in ninety-nine percent of the values for $\mathrm{v}_{L_{Re}}$ being within plus or minus three percent of the values for $\mathrm{v}_{L_{Re_0}}$. Defining the signal-to-noise ratio(S/N) as

$$S/N = \frac{100}{3\sigma_{ae}} \tag{5.2}$$

the signal-to-noise ratio for Equation 5.1 is 33.3, where σ_{ae} is the standard deviation of the percent error added to the regression components. If the equation to add error is

$$\mathrm{v}_{L_{Re}} = \mathrm{v}_{L_{Re_o}} + \mathrm{v}_{L_{Re_o}} \times \left(\frac{25}{3}\right)\mathbf{randn}/100 \tag{5.3}$$

then S/N is four. Appropriate factors were placed in the equation to add error to the measured velocity values for each of the simulation runs so that S/N was infinite, 100, 20, and 5, respectively.

For the simulations, the height of the beam cross-section, h, and the elastic modulus of the aluminum, E, were the parameters in the FE model that were updated. The optimization routine was run at different starting points for h and E and the error in the final solution was recorded. The solution error was defined as

$$error = \frac{x^* - x_{solution}}{x^*} \times 100 \tag{5.4}$$

where x^* is the known exact value for a variable, and $x_{solution}$ is the value found in the optimization. The solution error defined in this fashion is the percent difference from the known exact solution. For the tests in this section, the error was calculated for both the parameters, h and E, and the node velocity values. The maximum of these values was recorded as the solution error for each of the starting points. At first, the model was run with no error on the velocity components. Then, progressively higher amounts of error were added to the measurements, and the error in the solution was recorded. Since the error added to the velocity measurements was random, the error levels in the solutions were averaged over thirty tests for each starting point.

The results of the four simulations are documented in Tables 5.2, 5.3, 5.4, and 5.5. The values in the top row are the starting points for the parameter E expressed

Table 5.2: Percentage Error in Solution for No Error on the Data, $\sigma_{ae} = 0$

$\frac{h}{h^*} \backslash \frac{E}{E^*}$	0.85	0.95	1.00	1.05	1.15
1.15	0.007	0.04	1.4e-4	5.4e-4	9.2e-5
1.05	1.2e-8	5.7e-5	3.0e-4	4.2e-5	2.7e-4
1.00	2.3e-8	1.5e-8	*dnc*	0.0031	1.7e-4
0.95	*dnc*	3.0e-9	1.5e-5	0.042	3.1e-6
0.85	*dnc*	*dnc*	*dnc*	*dnc*	2.9e-9

Table 5.3: Percentage Error in Solution for $S/N = 100$, $\sigma_{ae} = 0.333$

$\frac{h}{h^*} \backslash \frac{E}{E^*}$	0.85	0.95	1.00	1.05	1.15
1.15	0.062	0.48	0.61	0.53	0.67
1.05	0.86	0.51	0.43	0.67	0.69
1.00	0.56	0.54	0.54	0.53	0.59
0.95	*dnc*	0.60	0.41	0.56	0.69
0.85	*dnc*	*dnc*	*dnc*	*dnc*	0.70

Table 5.4: Percentage Error in Solution for $S/N = 20$, $\sigma_{ae} = 1.67$

$\frac{h}{h^*} \backslash \frac{E}{E^*}$	0.85	0.95	1.00	1.05	1.15
1.15	3.1	2.6	2.8	3.0	3.0
1.05	4.3	2.7	2.1	3.4	3.0
1.00	3.0	2.9	3.1	2.6	2.9
0.95	*dnc*	3.2	2.2	2.6	3.2
0.85	*dnc*	*dnc*	*dnc*	*dnc*	3.8

Table 5.5: Percentage Error in Solution for $S/N = 5$, $\sigma_{ae} = 6.67$

$\frac{h}{h^*} \backslash \frac{E}{E^*}$	0.85	0.95	1.00	1.05	1.15
1.15	14	12	12	12	12
1.05	13	13	12	14	11
1.00	15	12	19	13	8.3
0.95	*dnc*	11	11	11	12
0.85	*dnc*	*dnc*	*dnc*	*dnc*	10

as the ratio to the known exact value, E^*. The values in the first column are the starting points for h expressed as a ratio to the known exact value, h^*. The values in the matrix are the maximum percentage error of the solution compared to the exact value. The symbol *dnc*(did not converge) was placed for starting points where the optimization did not converge to a solution. The optimization algorithm terminates when, after an iteration, the variables, scaled from h and E, have not changed by more than 1e-6 and the objective function, Π, has not changed by more than 1e-4. If these criteria had not been met after one thousand iterations, it was assumed that the routine was not going to converge and the optimization was terminated with *dnc* as the result.

For zero error on the data(infinite S/N), shown in Table 5.2 the optimization finds the solution almost exactly. The optimization did not converge starting from the point ($\frac{h}{h^*} = 1$, $\frac{E}{E^*} = 1$) since the gradients of the objective function at this point(the optimum value) are zero which causes the scale factors to be infinite. This is unlikely to occur in practice, but the possibility of the start point being near enough to the optimum to cause numerical problems should be kept in mind. When the signal-to-noise ratio was set to 100, shown in Table 5.3, the average solution error for the starting points that converged was 0.6 percent. For the tests where $S/N = 20$, shown in Table 5.4, the solution was on average about 3 percent off. With $S/N = 5$, shown in Table 5.5, the solution was on average 12 percent off.

The average solution error for each of these tables is within $2\sigma_{ae}$ of zero, where σ_{ae} is the standard deviation of the percentage error added to the data. These tables show that the optimization solution is only as good as the data that is used. The level of error in the solution is consistently within the expected bounds of the random error applied to the data.

It is interesting to note that the starting points where the optimization would not converge when there was error on the data are the same points where the optimization would not converge with perfect data. This is probably due to the fact that error in the data does not change the shape of the objective function by a great deal. A contour plot of the error function is shown in Figure 5.2. The pattern of light areas

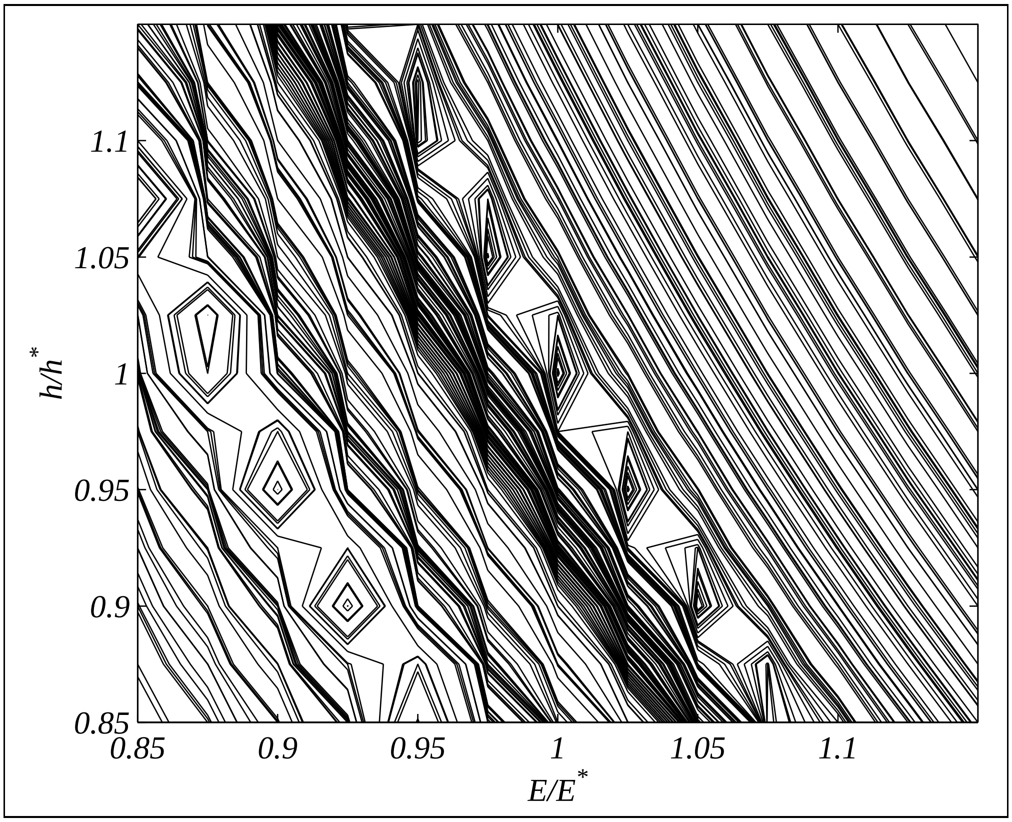

Figure 5.2: Objective Function Contour Plot

with regularly occurring circular bands that travels diagonally approximately from $(\frac{E}{E^*} = 0.85, \frac{h}{h^*} = 0.95)$ to $(\frac{E}{E^*} = 1.05, \frac{h}{h^*} = 0.85)$ are values of the cost function that are very large in comparison to the rest of the areas on the plot. The circular areas of contours in this range of high points are like mountains or a ridge in the parameter space. The optimum solution is located in the center of the plot $(\frac{E}{E^*} = 1, \frac{h}{h^*} = 1)$. It is unlikely that the optimization routine will be able to climb over those high points to get to the optimum if the starting point is on the wrong side of the ridge. All of the table entries where the algorithm did not converge(*dnc*) correspond to start positions in that area. This shows that a simulation performed before a test can show start points that will possibly never yield a good result.

Further examination shows that these high points in the objective function correspond to resonance in the model's predicted response. Figure 5.3 shows a plot of the driving point mobility magnitude for the actual system(known exact solution) and the modeled response for the initial parameter values $(\frac{E}{E^*} = 0.85, \frac{h}{h^*} = 0.85)$. The

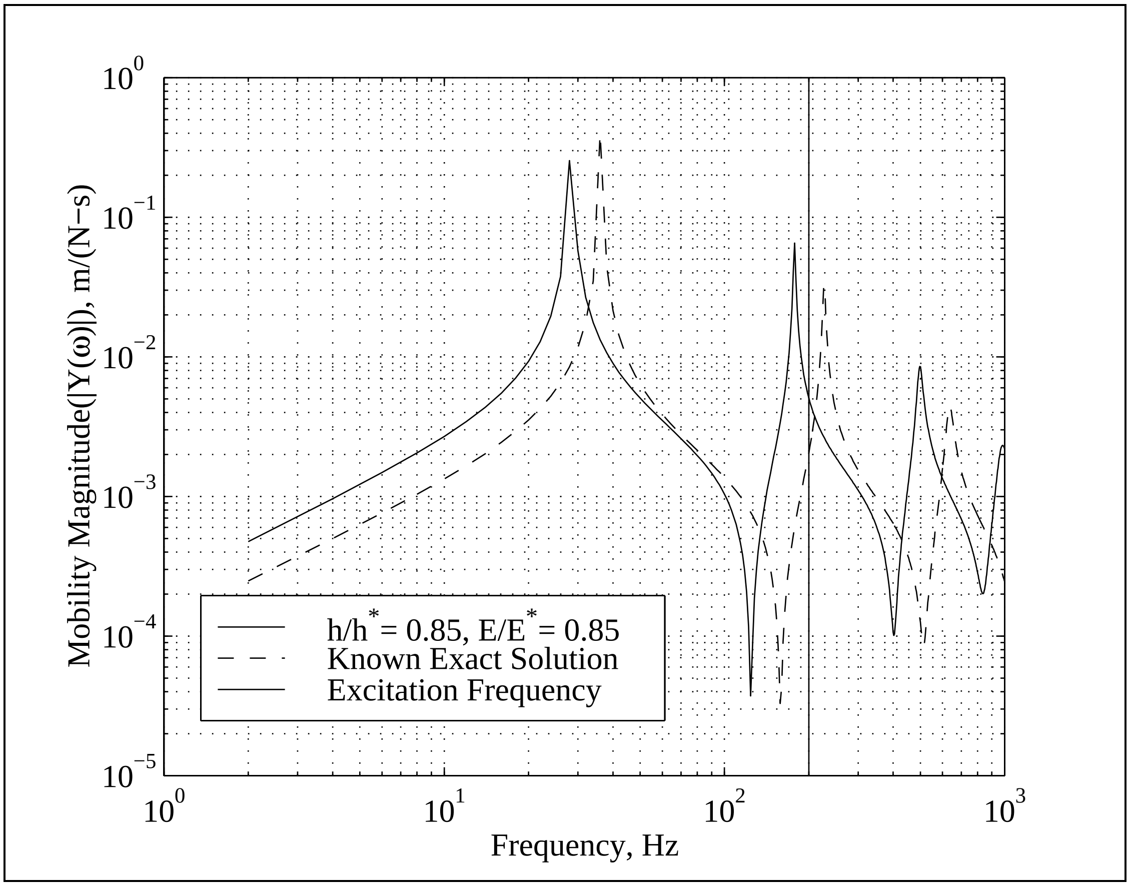

Figure 5.3: Predicted and Actual Driving Point Mobility

vertical line signifies the driving frequency at 200 Hz. As the plot shows, this driving frequency is just below the second natural frequency for the actual system, but just above the second natural freqency for the model. A plot of the signed magnitude operating shape(Figure 5.4) shows that the predicted response is about the same but 180 degrees out of phase to the actual response. As the optimization moves the predicted response curve to match the actual FRF, the model's second natural frequency approaches the driving frequency. The combination of very large response predictions and being 180 degrees out of phase from the actual system response causes the objective function(error between prediction and data) to take on very large vales. The ridge of high points in Figure 5.2 correpond to the model predicting resonance at 200 Hz. Consequently, the optimization for the update algorithm will not converge if the initial model predicts the excitation frequency is on the opposite side of resonance than it actually is for the structure.

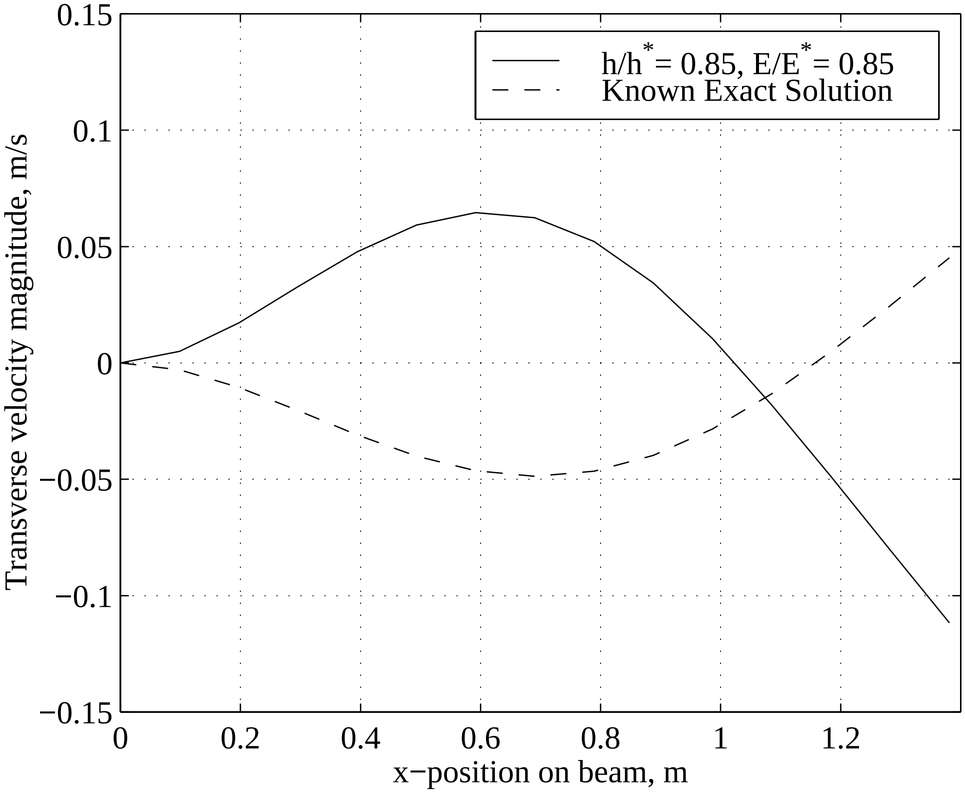

Figure 5.4: Predicted and Actual Operating Shape of the Beam

5.4 Error in the Position of the Laser

The next simulation shows how the algorithm performs when the laser beam origin is actually different from the estimated position. This error manifests itself in sampling the structure at a position different than intended. There are other types of error that can affect where the laser samples: error in the angle the laser points, movement of the sample point due to the motion of the structure, and uncertainty in the exact position of the sample point due to the fact that the laser beam is 1 mm in diameter. This simulation only addresses the error in the laser origin's position. As with the other simulations presented in this chapter, the direction the laser points is exactly known, so the direction cosines are exactly known. Also, the laser only samples from one exact point in the simulations, not from a circle of 1 mm diameter. The motion of the beam in these simulations is assumed to be small enough that the sample point does not change appreciably as the surface moves. There was no error placed on the regression coefficients so that errors in the time signal data did not cloud the effects of error in the laser position.

Figure 5.5 shows how error in the laser position results in an error in the sample position. The x and z coordinates of the assumed laser position are given by x_l and z_l, respectively. The actual laser position is displaced from this position by the distances δ_x and δ_z. For the simulation presented here, $x_l = 2.0$ m and $z_l = 0.35$ m. As stated earlier, the direction of the laser beam, indicated by the angle from vertical, θ, is assumed to be the same for both the assumed and actual origin position. The distance of the error in the sample position is given by $\delta_{\text{sample}} = \delta_x - \delta_z \tan\theta$. Thus, the error changes for each sample, as θ will be different for each sample point. This equation shows that for the same values of x_l, the higher in the z-direction the laser is placed, the less effect that error in the location in the z-direction will have. This is due to the fact that the values of θ, and hence $\tan\theta$, would be smaller. Unfortunately, larger z_l means that the laser is less capable of seeing in-plane(x-direction here) velocities. This trade-off between error in sample position and capability to measure in-plane velocities should be examined before a test is performed.

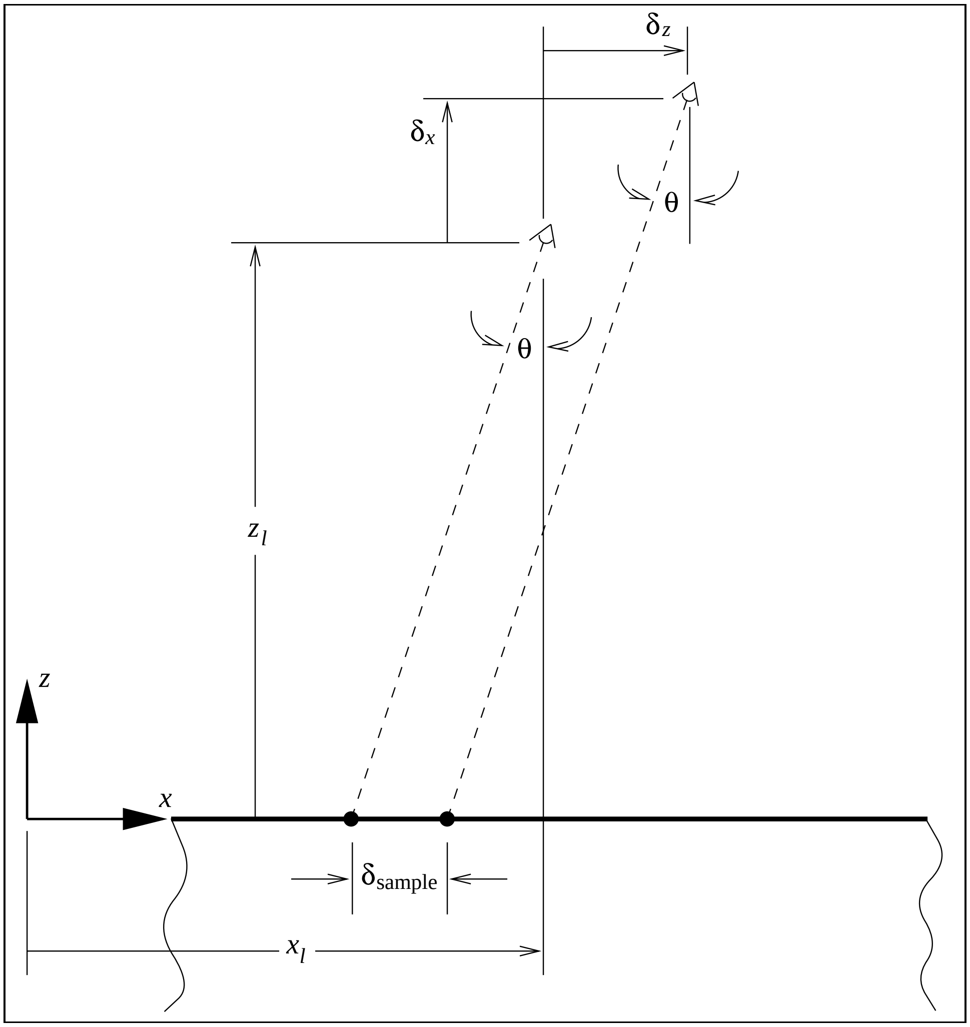

Figure 5.5: Sample Point Position Error

Table 5.6: Error in Parameter Solution for Given Laser Position Error

δ, mm	Percentage Error in h	Percentage Error in E
0	3.21e-5	-8.01e-5
0.1	-0.381	0.838
0.5	-1.26	2.79
1.0	-3.56	8.26
2.0	-6.55	16.0
3.0	-8.99	22.8

The worst case for positive θ angles is when δ_x is positive and δ_z is negative. This was the case modeled, and for the sake of simplicity, $\delta_x = -\delta_z = \delta$. This type of error in the laser position was examined for models with just E as the parameter, just h as the parameter, and both E and h as the parameters. For all the cases, the starting point was ($\frac{E}{E^*} = 0.95$, $\frac{h}{h^*} = 0.95$).

For the cases with E and h as parameters, the error is shown in Table 5.6. As shown, the level of error in the solution became larger than the initial error when $\delta = 1.0$ mm. The error in the sample position on the beam(δ_{sample}) was 6.6 mm at the fixed end when $x_l = 2.0$ m, $z_l = 0.35$ m, and $\delta = 1.0$ mm. Moving the laser to a higher position above the beam reduces the size of θ, thus reducing the size of δ_{sample}. Unfortunately, increasing z_l to 4 m only increased the tolerance for error in the laser position to 2 mm, so that the problem was still present.

The errors reported in the parameters here are relative to the exact known values. Although parameter solution error when h is a parameter alone is lower than the error with E and h as parameters, Figures 5.6 and 5.7 indicate that the solution with E and h as parameters provides a better fit to the data. These figures show the residuals(data – model) of the velocity values for each sample point along the beam's length. The mean squared error and mean error of the residuals are smaller for the solution with E and h as compared to the prediction provided by the model with just h as an update parameter. Given the residuals in the figures, the model with updated E and h would be chosen over the model with just h. The fact that error from the exact values for the parameters is larger would never be known. The plots of the residuals do indicate a problem in the updated model since there is a

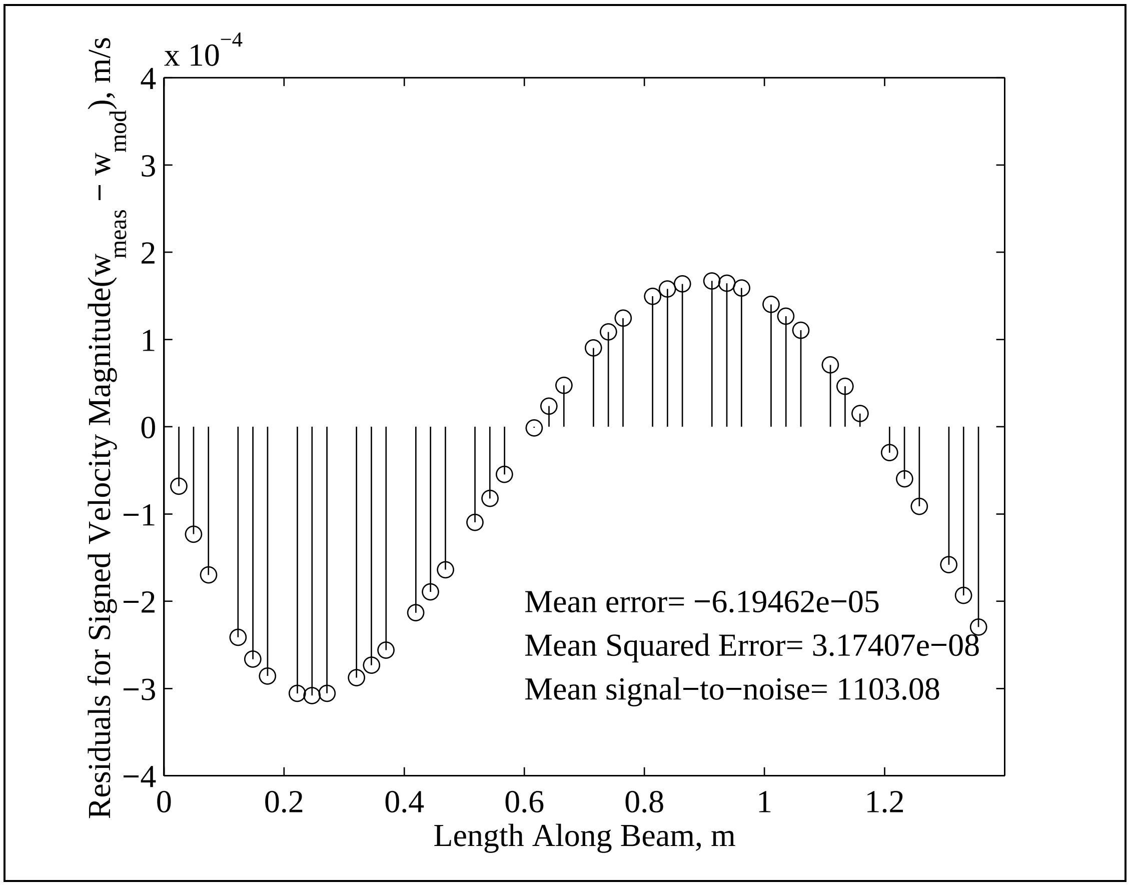

Figure 5.6: Residuals for Model with E and h Updated

very deterministic pattern. A sinusoidal pattern like that shown in the figures for residuals from an actual test might indicate that there is systematic error in the laser position. In both cases, the residuals are small compared to the modeled velocity values. This indicates that the velocity predictions are not bad; however, comparison of the updated parameters to their known values shows that there is error.

When E and h were parameters alone, the error when δ was 2.0 mm was negligible, so that only the case with both parameters updated had large errors. Although the simulation indicates that the position of the laser needs to be pinpointed to an accuracy of 2.0 mm, it is unlikely that this requirement can be met. The results of these simulations show some allowance for error in the laser position must be made if there is any hope of updating some combinations of parameters in the model.

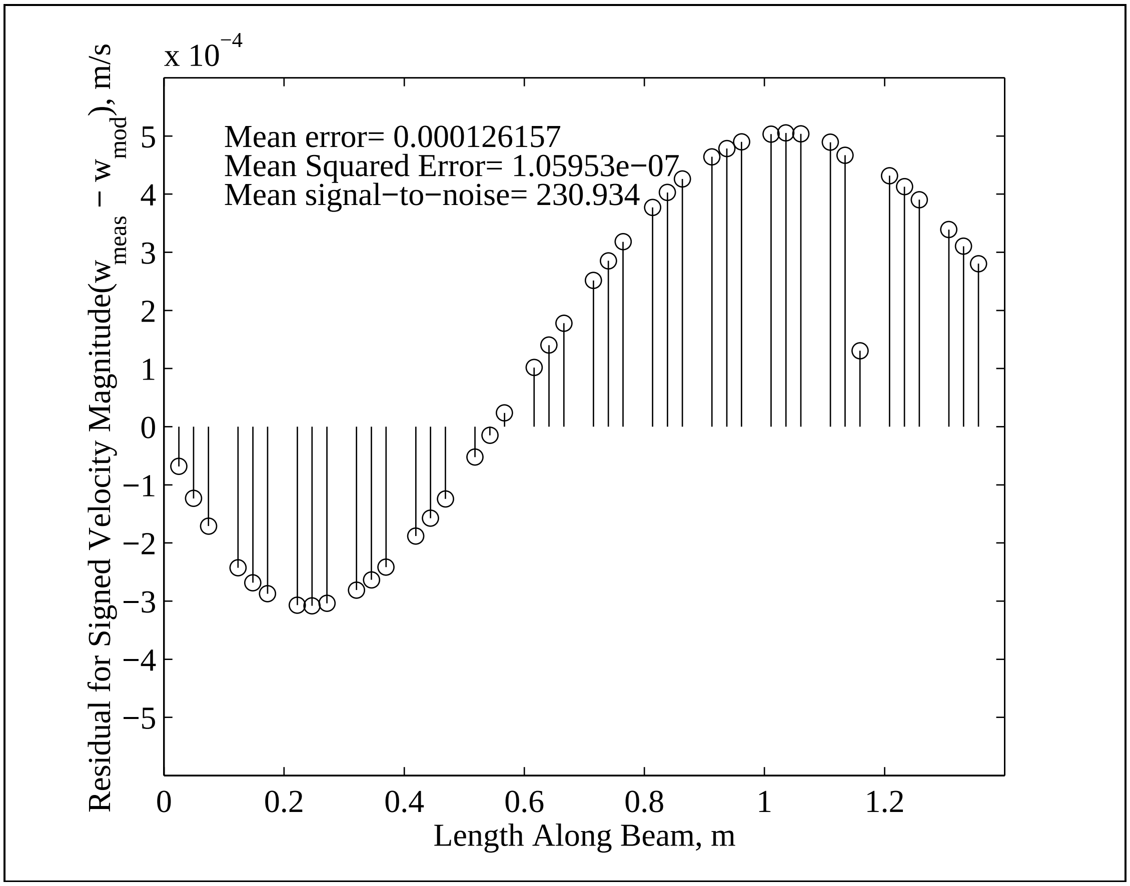

Figure 5.7: Residuals for Model with h Updated

5.5 Choosing the Wrong Parameter for Update

The last set of simulations were performed to examine the quality of the solution when there is unaccounted error in one of the parameters in the FE model. The simulations were run with no error present on the regression components and no inaccuracy in the position of the laser so that only the effects of the wrong parameter would be present. All the simulations were run with E as the optimization variable and from the initial point $\frac{E}{E^*} = 0.99$. The error in the solution for the updated value of E when all the other parameters in the analytical model were set to their correct values was 1.59e-4 percent, where the error is defined as

$$error = \frac{E^* - E_{solution}}{E^*} \times 100$$

Tables 5.7, 5.8, and 5.9 show the percentage difference between the updated value for E and the known exact value of E for different levels of parameter error. Table 5.7 shows that the solution's level of error in E is about the same as the level of error in the density, which wasn't included in the update. The updated solution for E appears to be attempting to make up for the unaccounted for error in ρ.

Table 5.8 shows that even small errors in the height of the beam cross-section can lead to large errors(greater than 5%) in the solution for E. This is probably due to the fact that h is present in the transformation from the neutral axis to the surface, so that changes in h affect not only the FE model, but also the test data.

For Table 5.9, β is the proportional damping constant for the stiffness matrix. The damping matrix is proportional to the values in the stiffness matrix when this constant is used. Table 5.9 shows that errors in the damping, even by an order of magnitude, do not affect the level of error in the parameter E at the solution too adversely for this structure excited at 200 Hz. Errors in the damping are shown to such large values since it is possible, given the lack of knowledge about damping, that the estimate may be off by such a large amount. Table 5.9 also shows that even though there might be small error in the updated value for E, the velocities found by this model are grossly incorrect. The standard deviation of some of the models' prediction errors were on the same order of magnitude as the beam's velocity values

Table 5.7: **Percentage Error in Solution Value for E When ρ is Wrong**

$\frac{\rho_{\text{modelled}}}{\rho_{\text{actual}}}$	Percentage Error for E
0.85	-11.7
0.95	-3.90
1.05	3.90
1.15	11.7

Table 5.8: **Percentage Error in Solution for E When h is Wrong**

$\frac{h_{\text{modelled}}}{h_{\text{actual}}}$	Percentage Error for E
0.85	*dnc*
0.95	12.1
1.05	-10.2
1.15	-26.5

which are around 0.05 m/s, as shown in Figure 5.4. This is due to the fact that near resonance, where this test is being conducted, the damping plays a crucial role in determining the magnitudes of the velocities. This shows that it would pay to experiment with different damping values, or possibly even include damping in the parameters used in the update. This assumes that the proportional damping model is applicable to the problem. If not, some other damping model that would integrate into this algorithm should be used. Table 5.9 also indicates that underestimating the damping provides better velocity magnitudes than overestimating the damping, as the standard deviations for models with underestimated damping are smaller than models with corresponding overestimates.

Table 5.9: **Percentage Error in Solution for E and Velocity Magnitudes When β is Wrong**

$\frac{\beta_{\text{modelled}}}{\beta_{\text{actual}}}$	Percentage Error for E	Standard Deviation of Prediction Error for Velocities, m/s
0.01	0.128	1.28e-4
0.10	0.107	1.09e-4
0.85	4.29e-4	3.18e-5
1.15	-8.70e-3	2.89e-5
10	5.83	0.01037
100	44.1	0.029862

5.6 Summary of Simulation Results

The simulations demonstrated several characteristics of the update algorithm:

1. The updated parameter values resulting from the optimization solution are only going to be as good as the data. The parameters' percentage errors will be within two standard deviations of the percentage error on the data.

2. There is a possibility that the initial gradients will be very small. This will seriously affect the optimization scheme since the scale factors for the variables will be very large, causing numerical problems.

3. If the initial point in the optimization is in a region of the parameter space that is separated from the optimum by a "mountain range," it is unlikely the optimization routine will converge to the correct solution. These high points in the objective function correspond to the model predicting resonance at the excitation frequency. Initializing the update algorithm from several positions in the parameter space should provide the optimization routine the opportunity to find a solution.

4. The algorithm is intolerant to all but the smallest errors in the laser position. Testing parameter combinations before attempting to update them with test data may reveal that they will not work together. Moving the laser so that the beam's angle of incidence is smaller alleviates, but does not eliminate the problem. Some method to allow for error in the laser position must be included for the update algorithm to succeed.

5. If the right parameters are not included in the update, there will be errors in the solution. These errors could prove to be very large. This emphasizes the fact that the choice of parameters is very important.

6. Underestimating damping provided better model predictions than overestimating damping for the structure studied.

The simulations in this chapter have demonstrated the concept of updating using the algorithm presented in this research. Based on the findings in this chapter, it is likely that the update algorithm will work, provided:

- The error on the data is not unreasonably large,
- The initial estimates for the update parameters and the estimates used for parameters not included in the update are not too far off,
- The model is capable of representing the deformations that actually occurred during the test,
- The correct parameters are chosen for the update, and,
- Some method to account for error in the laser position is included.

The first point is an experimental issue that must be dealt with in a rigorous fashion. A quantified estimate of the error in the data, due to noise on the measured signals and due to errors in the laser position, must be available to estimate the quality of the solution. The final three points put the responsibility for the update squarely on the user. These issues involve decisions based on prior knowledge about the structure being modeled. The final point is a requirement that must be satisfied if the algorithm is to succeed.

The next step is to update parameters in a model of an actual structure with known characteristics. Broader conclusions on the performance of the update algorithm could then be drawn.

Chapter 6

Conclusion

6.1 Summary of Thesis

The impetus for the research presented in this thesis is the need for a framework to update finite element(FE) models using vibration test data. This thesis suggests a formulation for updating where experimental data from a scanning laser Doppler vibrometer(SLDV) is posed in a manner consistent with a FE model's representation for harmonic response. The work presented here draws upon the research of Montgomery (1994) and Lindholm (1996), and brings ideas from both together in the development of an update procedure.

This thesis began by showing the relationship between the structural velocity vector, $\dot{\mathbf{u}}_s$, and the quantity that the laser measures, the magnitude of the projection of $\dot{\mathbf{u}}_s$ onto the laser beam direction. Next, it was shown how a harmonic response is measured by the SLDV and processed into the velocity components, $\mathrm{v}_{L_{Re}}$ and $\mathrm{v}_{L_{Im}}$, for every sample location. These components, constituting a frequency domain representation, carry the phase and magnitude of the measured velocity signal at that sample location. The input force is also measured and processed in a similar manner. The phase angle for every sample location is measured relative to a reference force input signal so that the relative phases between sample locations have meaning. It was shown how the regression to process the velocity signal could be extended to process signals that have multiple frequency components. As a result, the data available for updating a model from a single vibration test includes: frequency domain representations for velocity at many locations on the structure and the measured force input(both possibly at multiple frequencies).

An experimental model was developed using finite element shape functions as a basis for a least squares fit to the experimental data. The experimental model efficiently stores the SLDV data from a vibration test in a manner that fully exploits the instrument's capability to automatically make measurements at many locations. This experimental model was developed expressly to pose the data in a form that easily allows an analytical finite element model to be inserted. An analytical finite element model was developed to model harmonic response so that physically meaningful parameters in the model could be updated. Since structural theories usually model the response at the midsurface, a transformation was developed that relates the velocities at the surface where the laser measures to the midsurface velocities. The analytical FE model was inserted into the experimental model resulting in a function that is generally nonlinear in the update parameters. Minimizing the function reduces the prediction error of the finite element model as compared to the test data, and results in an updated model.

Based on the form of the objective function, the quasi-Newton method was chosen to find its minimum. Analytically supplied gradients were first sought to improve the performance of the optimization to minimize the objective function. Preliminary tests of the algorithm indicate that these analytically supplied derivatives are required for a solution to be found. It was also learned that the variables in structural problems must be scaled or the optimization routine has no chance to find the solution. In order to complete the framework of the updating algorithm, validation techniques were discussed. Application of the algorithm presented here will require a method to validate the updated model and quantify the uncertainty associated with the updated parameters.

Simulations were performed to demonstrate the feasibility of the algorithm to update parameters. The simulations showed three key points:

1. The algorithm is tolerant of some error in the data but the resulting solutions

for the parameters cannot be expected to be any better than the quality of the data.

2. The algorithm will tolerate very little error in the position of the laser.

3. The choice of the parameters to update in the model is extremely important.

Overall, the goals of the research were met. A method to update finite element models that is consistent with the experimental model was developed. The algorithm was developed in a manner that is conducive to extending its use with many different types of elements and update parameters. A framework has been constructed in which models can effectively be updated. The feasibility of using this formulation to update FE models has been demonstrated in a limited number of computer simulations.

6.2 Recommendations

Several extensions to the algorithm could make it a much more powerful technique, but the next step is to apply this algorithm to a real structure with relatively well known characteristics. The known characteristics should be updated using the update algorithm presented here, and then checked using independent methods. This would allow an independent check on the solution provided by the update algorithm. In order to make this test, the algorithm would probably need to be extended to use plate or shell elements. This is due to the fact that the these type elements are much more applicable to general structures than the beam elements presented in this thesis.

Allowing the laser location to be a variable may help to alleviate any difficulties caused by not knowing its position exactly. The simulations in Section 5.4 show that the performance of the algorithm can be greatly affected by error in the laser position, so any modification that could reduce these errors would be beneficial.

Making the structure's geometry a variable would allow for imperfections, like out-of-roundness in a cylindrical structure. Variations in the structure's shape would affect its dynamic characteristics and the position the laser strikes the surface. Both of these effects will modify the data and the analytical model's prediction. One way to allow geometry variations is through element-by-element updating. This would also allow the update algorithm to be used in damage detection.

Currently, many vibration tests are performed using free-free boundary conditions(or something close). The experimental model, χ^2, can be used for free-free structures, but the analytical model cannot. While finite element models can model free-free response using modal summation, solutions using direct frequency response cannot deal with these boundary conditions at the moment since they result in a singular matrix. A technique to allow FE models to handle this situation would be very useful for vibration modeling in general, and for this algorithm specifically.

Since harmonic response is being modeled for this update algorithm, damping must be included. Research into the mechanism of damping and appropriate methods to model it are needed. Currently, the most accepted method of modeling damping is through the use of matrices proportional to the mass and stiffness matrices. This is probably due more to the ease of this technique than the fact that the physical mechanism of damping is explained this way.

The parameters used in the update could benefit from having confidence intervals to check the updated values for validity. This is a way of using prior knowledge about the estimated value of a parameter to judge the value the update procedure finds. This concept has been explored by Lindholm (1996) using a Bayesian statistics formulation for updating. These concepts could be incorporated into the algorithm presented here.

The validation techniques could provide much useful information in an experimental test. Their usefulness could be shown in an update using experimental data from a relatively simple structure. One interesting possibility that one of the regression tests suggests is performing a convergence of the experimental model. Regression tests could be used to decide how many elements that the experimental model needs to effectively represent the response. This mesh could be compared to the mesh produced in converging an analytical finite element model to produce the same response. If there are differences, an attempt to explain may show inadequacies in the underlying theory of the analytical FE model.

Finally, it is very important that the normality of the error on the laser data be tested. If the error is not distributed normally, and the degree of nonnormality is severe, then the inferences of the regression tests presented in this research are not appropriate. For mild cases of nonnormal error, the least squares techniques will still work. In the severe cases, more robust estimation techniques must be used in the time signal regressions, the experimental model, and thus in the update formulation.

6.3 Conclusions

Several major conclusions can be drawn from the research to develop this update algorithm. The first is that no matter how good the model updating algorithm is, it is limited by the data that is supplied to it, the candidate model, and the parameters that are given for it to update. The simulations indicate that the solution error of the parameters, in terms of percentage, will fall within two standard deviations of the data's error. If the variance of the error on the data is large, the confidence interval on the updated parameters will also be large. Based on this, establishing the nature of the data is important right from the start. Error in the position of the laser will systematically affect the data. One simulation showed that the laser must be pinpointed to 2 mm accuracy for the update algorithm to find the correct solution. In light of this finding, some method to allow for error in the laser position must be incorporated for the update algorithm to be feasible.

The formulation presented in this thesis places the responsibility on the user to provide the correct model and parameters to update. Starting with the wrong candidate FE model, there is no hope of realistic updating because the model cannot produce the deformations seen in the actual structure, and thus the data. If the wrong parameters are chosen for update, or not enough of the right ones are chosen, there is no guarantee on what the update algorithm will provide as a solution. A methodology to choose parameters for update would be very useful. Finally, placing the validation procedures on firm statistical ground is the only measure that can be hoped to show the degree of the quality of the updated model. The optimization routine used in the update algorithm will almost always provide a solution, the usefulness of this solution must be tested.

Bibliography

Allemang, R. and Ewins, D., 1983, "A Correlation Coefficient for Modal Vector Analysis," in: *Proceedings of the First International Modal Analysis Conference*, Orlando, FL, pp. 110–116.

Arora, J. S., 1989, *Introduction to Optimum Design*, McGraw-Hill Series in Mechanical Engineering, McGraw-Hill, Inc., New York.

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

Beer, F. P. and Russell, Jr., J., 1992, *Mechanics of Materials*, 2nd edn., McGraw-Hill, Inc., New York.

Blevins, R. D., 1979, *Formulas for Natural Frequency and Mode Shape*, Chap. 8, Van Nostrand Reinhold Company Inc., New York, p. 108.

Catbas, F., Aktan, A., Allemang, R., and Brown, D., 1998, "A Correlation Function for Spatial Locations of Scaled Mode Shapes- COMEF," in: *Proceedings of the 16th International Modal Analysis Conference*, Santa Barbara, CA, pp. 1550–1555.

Cook, R. D., 1995, *Finite Element Modeling for Stress Analysis*, John Wiley Sons, Inc., New York.

Cook, R. D., Malkus, D. S., and Plesha, M. E., 1989, *Concepts and Apllications of Finite Element Analysis*, Chap. 19, 3rd edn., John Wiley Sons, New York, p. 582.

Donaldson, B. K., 1993, *Analysis of Aircraft Structures: An Introduction*, McGraw-Hill Series in Aeronautical and Aerospace Engineering, McGraw-Hill, Inc., New York.

Drain, L., 1980, *The Laser Doppler Technique*, John Wiley Sons, Ltd., Chichester, United Kingdom.

Friswell, M. and Mottershead, J., 1995, *Finite Element Model Updating in Structural Dynamics*, Vol. 38 of *Solid Mechanics and Its Applications*, Kluwer Academic Publishers, Norwell, Massachusettes.

Grace, A., 1992, *Optimization Toolbox User's Guide*, The MathWorks, Inc., Natick, Massachussettes.

Haftka, R. T. and Gurdal, Z., 1991, *Elements of Structural Optimization*, Vol. 11 of *Solid Mechanics and Its Applications*, third revised and expanded edn., Kluwer Acedemic Publishers, Norwell, Massachusettes.

Heylen, W. and Lammens, S., 1996, "FRAC: A Consistent Way of Comparing Frequency Response Functions," in: *Proceedings of the International Conference on Identification in Engineering*, pp. 48–57.

Lieven, N. and Ewins, D., 1988, "Spatial Correlation of Mode Shapes, The Coordinate Modal Error Function(COMAC)," in: *Proceedings of the Sixth International Modal Analysis Conference*, Kissimmee, FL, pp. 690–695.

Lindholm, B. E., 1996, "A Bayesian Statistics Approach to Updating Finite Element Models with Frequency Response Data," Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Ljung, L., 1987, *System Identification: Theory for the User*, Chap. 1, Prentice Hall Information and System Sciences Series, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, p. 9, Figure 1.11.

McConnell, K. G., 1995, *Vibration Testing: Theory and Practice*, John Wiley Sons, Inc., New York.

Meirovitch, L., 1997, *Principles and Techniques of Vibrations*, Chap. 1, Prentice-Hall, Inc., Upper Saddle River, New Jersey, p. 14.

Mitchell, L. D., 1998, "Increasing the Sensitivity of the Modal Assurance Criteria(MAC) to Small Mode Shape Changes: The IMAC," in: *Proceedings of the 16th International Modal Analysis Conference*, Santa Barbara, CA, pp. 64–69.

Montgomery, D. E., 1994, "Modeling and Visualization of Laser-Based Three-Dimensional Experimental Spatial Dynamic Response," Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Venter, G., 1995, "Sensitivity Analysis With Respect to Elastic Boundary Conditions and Laser Spatial Variables Within Experimental Spatial Dynamic Modeling," M.S. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Zienkiewicz, O. C., 1977, *The Finite Element Method*, Chap. 20, 3rd edn., McGraw-Hill Book Company Limited, London, pp. 558–559.

Appendix A

Matlab M-files

```
function [max_error, indice]= sim(h_hstar, E_Estar)
% Function to find a solution that minimizes the squared error between FE node
% velocities and velocities measured using a laser Doppeler vibrometer.  The
% optimization problem is solved using the Matlab command fminu.  The structure
% is a fixed-free beam forced at the free end.  The beam can deflect
% transversely and along its axis.
t0 = clock;
m= 14;          % number of elements
g= 3*m+3;       % number of dof before b.c.'s are enforced
p= g-3;         % number of dof after b.c.'s are enforced
mgp= [m g p];
he= 1.381/m;
spe= 4;         % number of samples taken per element
ic= make_ic(m); % Interconnectivity matrix from FE formulation
% Known FE parameters:
known(1)= 200*2*pi;       % driving frequency, w
known(2)= 2800;           % density of the material, rho
%known(3)= 75e9;          % elastic modulus
known(4)= 1.381;          % length of the beam
known(5)= 0.01270;        % beam width
%known(6)= 0.08255;       % beam height
known(7)= 0;              % proportional(to mass) damping constant, alpha
known(8)= 1.4e-5;         % proportional(to stiffness) damping constant, beta
% Starting points for unknown FE parameters for optimization:
E0= E_Estar*75e9;         % elastic modulus
h0= h_hstar*0.08255;      % beam height
x0= [h0; E0];             % Starting point vector for optimization
num_par= 2;               % number of variable parameters in the FE formulation
scales= ones(num_par, 1);  % initialize the scale factors as ones
% Get experimental velocity data:
figure(1)
[vlRe, vlIm, dl_dh, d_actual, laz_er]= gen_exp_data(m, g, p, ic, spe, scales);
%save exp_data.mat vlRe vlIm dl_dh d_actual;
%load exp_data.mat;
% Get the scales for the variables:
scales= find_scale(x0, known, num_par, m, g, p, ic, dl_dh, vlRe, ...
 vlIm, spe, scales);
x0= x0./scales(:);
options(1)= 0;                  % controls output display
options(2)= 1e-6;               % precision required of the variables for term.
options(3)= 1e-4;               % precision required of objective function for
                                % term.
options(7)= 0;                  % line search method- cubic if 1
options(9)= 0;                  % controls gradient check
options(14)= 500*num_par;       % maximum number of iterations
x= fminu('obj_fun', x0, options, 'grad', known, mgp, ic, scales, dl_dh, ...
 vlRe, vlIm, spe);
 % Reporting on the solution:
x= x.*scales(:);
force= make_force(m, g);
Z= make_Imp(known(1), known(2), x(2), known(4), known(5), x(1), ...
 known(7), known(8), m, g, p, ic, ones(num_par, 1));
d_sol= Z\force;
format short g, x= [d_sol; x];
format
run_time= etime(clock, t0)
for i= 1:2*p
   if abs(d_actual(i))>eps
      error(i)= 100*(x(i) - d_actual(i))/d_actual(i);
   else
      if abs(x(i))>eps
         error(i)= 100;
      else
         error(i)= 0;
      end
   end
end
figure(2)
plt= plot_res(d_sol, m, g, p, ic, spe, num_par);  % makes residual plot
error(2*p+1)= 100*(x(2*p+1) - 0.08255)/(0.08255);
error(2*p+2)= 100*(x(2*p+2) - 75e9)/(75e9);
error= error'
h_error= error(2*p+1)
E_error= error(2*p+2)
max_laz_error= max(abs(laz_er))
[max_error, indice]= max(abs(error));
```

```
function ic= make_ic(m)
% Constructs the interconnectivity matrix for beam elements that have 3 dof
% at each node(transverse translation, rotation, and axial translation) given
% the number of elements, m.
ic= zeros(m, 6);
for ele= 1:m
   ic(ele, 1)= 3*ele - 2;
   ic(ele, 2)= 3*ele - 1;
   ic(ele, 3)= 3*ele;
   ic(ele, 4)= 3*ele + 1;
   ic(ele, 5)= 3*ele + 2;
   ic(ele, 6)= 3*ele + 3;
end
```

```
function [vlRe, vlIm, dl_dh, d_actual, laz_er]= gen_exp_data(m, g, p, ic, ...
 spe, scales)
% gen_exp_data.m
% Function that creates "experimental" velocity data measured by the laser using
% a finite element solution for the vibration of a beam with 3 dof at each node.
% A number of samples per element is specified in the variable spe and the
% samples are spread evenly inside each element.  Samples are only taken on the
% interiors of elements so that there is no need for a decision pertaining to
% which element the sample should contribute.  This version has been modified
% to allow for error in the location of the laser.  This error only produces an
% error in the sampled velocity values.  The shape function values placed in the
% experimental matrix and vector(Q and c) are not affected.
% Correct parameter values:
w= 200*2*pi;    % driving frequency
rho= 2800;      % density of the material
E= 75e9;        % elastic modulus
Len= 1.381;     % length of the beam
b= 0.01270;     % beam width
h= 0.08255;     % beam height
alpha= 0;       % proportional(to mass) damping constant
beta= 1.4e-5;   % proportional(to stiffness) damping constant
Z= make_Imp(w, rho, E, Len, b, h, alpha, beta, m, g, p, ic, scales);
force= make_force(m, g);
% Correct FE solution:
d= Z\force;
d_actual= d;
d= [zeros(3, 1); d(1:p); zeros(3, 1); d(p+1:2*p)];
% Stuff to plot actual displaced shape:
he= Len/m; % element length
xd= 0:he:Len;
j= 1;
wmag_star(j)= 0;
for j= 1:m
   wmag_star(j+1)= sign(atan2(d(3*j+2+g), d(3*j+2)))*abs(d(3*j+2) +...
    i*d(3*j+2+g));
end
div= (spe+1)/2; % reciprocal of equal length space between each sample within
                % the elements in terms of the dimensionless element coordinate
                % system
% Assumed position of the laser source w.r.t. the origin of the FE model global
% coodinate system:
laser_x= 2.0;
laser_y= 0;
laser_z= 2.0;
% Error between the assumed laser position and the actual position:
delta= 2e-3;
delta_x= delta;
delta_z= -delta;
% Stuff to plot sampled shape:
j= 1;
xs(j)= 0;
wmag(j)= 0;
laz_er= [];
% Generate "experimental" data:
```

```
dlRe_dh= zeros(g, 1);
dlIm_dh= zeros(g, 1);
vlRe= zeros(m, spe);
vlIm= zeros(m, spe);
for ele= 1:m
   dlRe_dh_ele= zeros(6, 1);
   dlIm_dh_ele= zeros(6, 1);
   for sample= 1:spe
      s= -1 + sample/div;  % Element coordinate system position
      % Shape function values for the current sample point:
      N1r= -1/2*(s - 1);
      N2r= 1/2*(s + 1);
      nr= [N1r; 0; 0; N2r; 0; 0];
      % Derivatives of shape functions for the current sample point:
      dN1b_dx= (-3 + 3*s^2)/(2*he);
      dN2b_dx= 1/4*(-1 - 2*s + 3*s^2);
      dN3b_dx= (3 - 3*s^2)/(2*he);
      dN4b_dx= 1/4*(-1 + 2*s + 3*s^2);
      dnbz_dx= [0; dN1b_dx; dN2b_dx; 0; dN3b_dx; dN4b_dx];
      sample_x= ((ele - 1)*N1r + ele*N2r)*he;  % Assumed sample point position
      sample_y= 0;                             % in FE global coordinate system
      sample_z= 0;                             % (actual sample point location
                                               % is different)
      % Sample point direction cosines
      psi_x= (laser_x - sample_x)/sqrt((laser_x - sample_x)^2 + ...
       ((laser_y - sample_y)^2 + (laser_z - sample_z)^2));
      psi_z= (laser_z - sample_z)/sqrt((laser_z - sample_z)^2 + ...
       ((laser_x - sample_x)^2 + (laser_y - sample_y)^2));
      psi= [psi_x; psi_z];
      % Values to find the actual sample point velocity:
      theta= acos(psi_z);
      b= delta_x - delta_z*tan(theta);
      laz_er= [laz_er; b];
      b_s= 2*b/he;
      if (s+b_s)>1
         error('s is greater than 1')
      end
      N1rd= -1/2*((s+b_s) - 1);
      N2rd= 1/2*((s+b_s) + 1);
      nrd= [N1rd; 0; 0; N2rd; 0; 0];
      dN1b_dxd= (-3 + 3*(s+b_s)^2)/(2*he);
      dN2b_dxd= 1/4*(-1 - 2*(s+b_s) + 3*(s+b_s)^2);
      dN3b_dxd= (3 - 3*(s+b_s)^2)/(2*he);
      dN4b_dxd= 1/4*(-1 + 2*(s+b_s) + 3*(s+b_s)^2);
      dnbz_dxd= [0; dN1b_dxd; dN2b_dxd; 0; dN3b_dxd; dN4b_dxd];
      N1bd= 1/4*(2 - 3*(s+b_s) + (s+b_s)^3);
      N2bd= he/8*(1 - (s+b_s) - (s+b_s)^2 + (s+b_s)^3);
      N3bd= 1/4*(2 + 3*(s+b_s) - (s+b_s)^3);
      N4bd= he/8*(-1 - (s+b_s) + (s+b_s)^2 + (s+b_s)^3);
      nbzd= [0; N1bd; N2bd; 0; N3bd; N4bd];
      % Transformed interpolation matrix for the current sample point:
      N_Trd= [nrd'- h/2*dnbz_dxd'
                nbzd'         ];
      % Surface velocities(different from the velocities at the assumed sample
      % point):
      uRe= N_Trd*d(3*ele-2:3*ele+3);
      uIm= N_Trd*d(3*ele-2+g:3*ele+3+g);
      j= j+1;
      xs(j)= sample_x;
      wmag(j)= sign(atan2(uIm(2), uRe(2)))*abs(uRe(2) + i*uIm(2));
      %  Velocity magnitudes reported by the laser, vlRe and vlIm:
      rdm= 1*randn/3;  % random error on the data
      vlRe(ele, sample)= psi'*uRe; % + psi'*uRe*rdm/100;
      vlIm(ele, sample)= psi'*uIm; % + psi'*uIm*rdm/100;
      % Derivatives:
      dN_Tr_dh= [-1/2*dnbz_dx'
                 zeros(1, 6) ];
      dlRe_dh_ele= dlRe_dh_ele + vlRe(ele, sample)*dN_Tr_dh'*psi;
      dlIm_dh_ele= dlIm_dh_ele + vlIm(ele, sample)*dN_Tr_dh'*psi;
   end
   % Placing the element derivative data into the global vectors:
   for a= 1:6
      dlRe_dh(ic(ele, a))= dlRe_dh(ic(ele, a)) + dlRe_dh_ele(a);
      dlIm_dh(ic(ele, a))= dlIm_dh(ic(ele, a)) + dlIm_dh_ele(a);
   end
end
% Enforce the boundary condition:
dlRe_dh= enf_bc_vec(dlRe_dh, g);
dlIm_dh= enf_bc_vec(dlIm_dh, g);
% Put the derivatives into one matrix.  The first column is dlRe_dh, the second
% is dlIm_dh.
dl_dh= [dlRe_dh dlIm_dh];
% Plot actual and sampled shapes:
plot(xs, wmag, 'o', xd, wmag_star, '--')
set(gca, 'FontName', 'Times')
set(gca, 'FontSize', 12)
title(['delta= ', num2str(delta)])
xlabel('Length Along Beam, m')
ylabel('Velocity Magnitude, m/s')
legendtext= legend('Sampled', 'Actual');
legendhandle = get(legendtext, 'Children');
legendhandle = legendhandle(2:2:length(legendhandle));
set(legendhandle, 'FontName', 'Times')
set(legendhandle, 'FontUnits', 'points')
set(legendhandle, 'FontSize', 12)
%plot(xs, 1000*[laz_er(1); laz_er])
%xlabel('Length Along Beam, m')
%ylabel('Laser Error, mm')
```

```
function Z= make_Imp(w, rho, E, Len, b, h, alpha, beta, m, g, p, ic, scales)
% Function to create the FE impedance matrix.
K= make_Stiff(rho, E, Len, b, h, m, p, ic, scales);
M= make_Mass(rho, Len, b, h, m, p, ic, scales);
C= alpha*M + beta*K;
% Enforce the boundary condition:
K= enf_bc_matrix(K, g);
M= enf_bc_matrix(M, g);
C= enf_bc_matrix(C, g);
Z= [w*M - 1/w*K      C
        C       -w*M + 1/w*K];
```

```
function K= make_Stiff(rho, E, Len, b, h, m, p, ic, scales)
% Assembles the global stiffness matrix from the element stiffness
% matrix for a beam element with 6 dof.
he= Len/m;
% Axial stiffness matrix:
ka= (E*scales(2)*b*h*scales(1)/he)*[1 -1; -1 1];
% Bending stiffness matrix:
kbz= (E*scales(2)*b*(h*scales(1))^3/(6*he^2))*[ 6/he   3     -6/he   3
                                                3      2*he  -3      he
                                               -6/he  -3      6/he  -3
                                                3      he    -3      2*he];
% Element stiffness matrix:
kele= [ka(1, 1)  0          0          ka(1, 2)  0          0
       0         kbz(1, 1)  kbz(1, 2)  0         kbz(1, 3)  kbz(1, 4)
       0         kbz(2, 1)  kbz(2, 2)  0         kbz(2, 3)  kbz(2, 4)
       ka(2, 1)  0          0          ka(2, 2)  0          0
       0         kbz(3, 1)  kbz(3, 2)  0         kbz(3, 3)  kbz(3, 4)
       0         kbz(4, 1)  kbz(4, 2)  0         kbz(4, 3)  kbz(4, 4)];
% Assemble the global stiffness matrix:
K= zeros(3*m+3, 3*m+3);
for ele= 1:m
   for a= 1:6
      for b= 1:6
         K(ic(ele, a), ic(ele, b))= K(ic(ele, a), ic(ele, b)) + kele(a, b);
      end
   end
end
```

```
function M= make_Mass(rho, Len, b, h, m, p, ic, scales)
% Assembles the global mass matrix from the element mass matrix for a beam
% element with 6 dof.
he= Len/m;
ma= (rho*b*h*scales(1)*he/6)*[2 1; 1 2];  % Axial mass matrix
% Bending mass matrix:
mb= (rho*b*h*scales(1)*he^2/420)*[ 156/he   22     54/he   -13
                                   22        4*he  13      -3*he
                                   54/he     13    156/he  -22
                                  -13      -3*he  -22       4*he ];
% Element mass matrix:
mele= [ma(1, 1)  0         0         ma(1, 2)  0         0
       0         mb(1, 1)  mb(1, 2)  0         mb(1, 3)  mb(1, 4)
       0         mb(2, 1)  mb(2, 2)  0         mb(2, 3)  mb(2, 4)
       ma(2, 1)  0         0         ma(2, 2)  0         0
       0         mb(3, 1)  mb(3, 2)  0         mb(3, 3)  mb(3, 4)
       0         mb(4, 1)  mb(4, 2)  0         mb(4, 3)  mb(4, 4)];
% Assemble global mass matrix:
M= zeros(3*m+3, 3*m+3);
for ele= 1:m
   for a= 1:6
      for b= 1:6
         M(ic(ele, a), ic(ele, b))= M(ic(ele, a), ic(ele, b)) + mele(a, b);
      end
   end
end
```

```
function matrix= enf_bc_matrix(matrix, g)
% This function returns the matrix with the boundary condition enforced.
matrix= matrix(4:g, 4:g);
```

```
function force= make_force(m, g)
% This function returns the force vector for the FE formulation.
% Force vectors:
force_Re= zeros(g, 1);
force_Im= zeros(g, 1);
force_Re(3*m+2)= 22;     % fz at the end of the beam
% Enforce the boundary condition:
force_Re= enf_bc_vec(force_Re, g);
force_Im= enf_bc_vec(force_Im, g);
force= [force_Im; force_Re];
```

```
function vec= enf_bc_vec(vec, g)
% This function returns the vector with the boudary condition enforced.
vec= vec(4:g);
```

```
function scales= find_scale(x, known, num_par, m, g, p, ic, dl_dh, vlRe, ...
 vlIm, spe, scales)
% This function finds the scale factors for the variables.
L= make_L(known(4), x(1), m, g, p, ic, spe, scales);
Q= [L          zeros(p)
    zeros(p)   L       ];
[lRe, lIm]= make_l(known(4), x(1), m, g, p, ic, vlRe, vlIm, spe, scales);
c= -2*[lRe; lIm];
dL_dh= make_dL_dh(known(4), x(1), m, g, p, ic, spe, scales);
dQ_dh= [ dL_dh   zeros(p)
        zeros(p)  dL_dh  ];
dc_dh= -2*scales(1)*[dl_dh(:, 1); dl_dh(:, 2)];
Z= make_Imp(known(1), known(2), x(2), known(4), known(5), x(1), ...
  known(7), known(8), m, g, p, ic, scales);
Y= inv(Z);
dZ_dh= make_dZ_dh(known(1), known(2), x(2), known(4), known(5), ...
  x(1), known(7), known(8), m, g, p, ic, scales);
dZ_dE= make_dZ_dE(known(1), known(2), known(4), known(5), x(1), ...
 known(8), m, g, p, ic, scales);
force= make_force(m, g);
df_dh= -2*force'*Y'*Q*Y*dZ_dh*Y*force - c'*Y*dZ_dh*Y*force ...
 + force'*Y'*dQ_dh*Y*force + dc_dh'*Y*force;
df_dE= -2*force'*Y'*Q*Y*dZ_dE*Y*force - c'*Y*dZ_dE*Y*force;
df= [df_dh; df_dE];
for j= 1:length(df)
   if df(j)~= 0
      scales(j)= 1/abs(df(j));
   end
end
```

```
function L= make_L(Len, h, m, g, p, ic, spe, scales)
% Function that creates "experimental" data matrix L.
he= Len/m;  % element length
div= (spe+1)/2;  % reciprocal of equal length space between each sample within
                 % the elements in terms of the dimensionless element coordinate
 % system
% Position of the laser source w.r.t. the origin of the FE model global
% coodinate system:
laser_x= 2.0;
laser_y= 0;
laser_z= 2.0;
% Generate L:
L= zeros(g);
for ele= 1:m
   L_ele= zeros(6);
   for sample= 1:spe
      s= -1 + sample/div;  % Element coordinate system position
      % Shape function values for the current sample point:
      N1r= -1/2*(s - 1);
      N2r= 1/2*(s + 1);
      nr= [N1r; 0; 0; N2r; 0; 0];
      N1b= 1/4*(2 - 3*s + s^3);
      N2b= he/8*(1 - s - s^2 + s^3);
      N3b= 1/4*(2 + 3*s - s^3);
      N4b= he/8*(-1 - s + s^2 + s^3);
      nbz= [0; N1b; N2b; 0; N3b; N4b];
      % Derivatives of shape function values for the current sample point:
      dN1b_dx= (-3 + 3*s^2)/(2*he);
      dN2b_dx= 1/4*(-1 - 2*s + 3*s^2);
      dN3b_dx= (3 - 3*s^2)/(2*he);
      dN4b_dx= 1/4*(-1 + 2*s + 3*s^2);
      dnbz_dx= [0; dN1b_dx; dN2b_dx; 0; dN3b_dx; dN4b_dx];
      % Transformed interpolation matrix for the current sample point:
      N_Tr= [nr'-(h*scales(1))/2*dnbz_dx'
                nbz'      ];
      sample_x= ((ele - 1)*N1r + ele*N2r)*he;  % Sample point position in FE
      sample_y= 0;                             % global coordinate system
      sample_z= 0;
      % Sample point direction cosines
      psi_x= (laser_x - sample_x)/sqrt((laser_x - sample_x)^2 + ...
       ((laser_y - sample_y)^2 + (laser_z - sample_z)^2));
      psi_z= (laser_z - sample_z)/sqrt((laser_z - sample_z)^2 + ...
       ((laser_x - sample_x)^2 + (laser_y - sample_y)^2));
      psi= [psi_x; psi_z];
      Psi= psi*psi';
      L_ele= L_ele + N_Tr'*Psi*N_Tr;
   end
   % Placing the element data into the global matrix:
   for a= 1:6
      for b= 1:6
         L(ic(ele, a), ic(ele, b))= L(ic(ele, a), ic(ele, b)) + L_ele(a, b);
      end
   end
end
```

```
% Enforce the boundary condition:
L= enf_bc_matrix(L, g);
```

```
function [lRe, lIm]= make_l(Len, h, m, g, p, ic, vlRe, vlIm, spe, scales)
% Function that creates "experimental" data vectors lRe and lIm.
he= Len/m;  % element length
div= (spe+1)/2;  % reciprocal of equal length space between each sample within
                 % the elements in terms of the dimensionless element coordinate
                 % system
% Position of the laser source w.r.t. the origin of the FE model global
% coodinate system:
laser_x= 2.0;
laser_y= 0;
laser_z= 2.0;
% Generate "experimental" data:
lRe= zeros(g, 1);
lIm= zeros(g, 1);
for ele= 1:m
   lRe_ele= zeros(6, 1);
   lIm_ele= zeros(6, 1);
   for sample= 1:spe
      s= -1 + sample/div;  % Element coordinate system position
      % Shape function values for the current sample point:
      N1r= -1/2*(s - 1);
      N2r= 1/2*(s + 1);
      nr= [N1r; 0; 0; N2r; 0; 0];
      N1b= 1/4*(2 - 3*s + s^3);
      N2b= he/8*(1 - s - s^2 + s^3);
      N3b= 1/4*(2 + 3*s - s^3);
      N4b= he/8*(-1 - s + s^2 + s^3);
      nbz= [0; N1b; N2b; 0; N3b; N4b];
      % Derivatives of shape function values for the current sample point:
      dN1b_dx= (-3 + 3*s^2)/(2*he);
      dN2b_dx= 1/4*(-1 - 2*s + 3*s^2);
      dN3b_dx= (3 - 3*s^2)/(2*he);
      dN4b_dx= 1/4*(-1 + 2*s + 3*s^2);
      dnbz_dx= [0; dN1b_dx; dN2b_dx; 0; dN3b_dx; dN4b_dx];
      % Transformed interpolation matrix for the current sample point:
      N_Tr= [nr'-(h*scales(1))/2*dnbz_dx'
                nbz'      ];
      sample_x= ((ele - 1)*N1r + ele*N2r)*he;  % Sample point position in FE
      sample_y= 0;                             % global coordinate system
      sample_z= 0;
      % Sample point direction cosines
      psi_x= (laser_x - sample_x)/sqrt((laser_x - sample_x)^2 + ...
       ((laser_y - sample_y)^2 + (laser_z - sample_z)^2));
      psi_z= (laser_z - sample_z)/sqrt((laser_z - sample_z)^2 + ...
       ((laser_x - sample_x)^2 + (laser_y - sample_y)^2));
      psi= [psi_x; psi_z];
      lRe_ele= lRe_ele + vlRe(ele, sample)*N_Tr'*psi;
      lIm_ele= lIm_ele + vlIm(ele, sample)*N_Tr'*psi;
   end
   % Placing the element data into the global vectors:
   for a= 1:6
      lRe(ic(ele, a))= lRe(ic(ele, a)) + lRe_ele(a);
      lIm(ic(ele, a))= lIm(ic(ele, a)) + lIm_ele(a);
   end
```

```
end
% Enforce the boundary condition:
lRe= enf_bc_vec(lRe, g);
lIm= enf_bc_vec(lIm, g);
```

```
function dL_dh= make_dL_dh(Len, h, m, g, p, ic, spe, scales)
% Function to return the derivative of the Q matrix w.r.t. the height of the
% beam.
he= Len/m;      % element length
div= (spe+1)/2; % reciprocal of equal length space between each sample within
                % the elements in terms of the dimensionless element coordinate
                % system
% Position of the laser source w.r.t. the origin of the FE model global
% coodinate system:
laser_x= 2.0;
laser_y= 0;
laser_z= 2.0;
dL_dh= zeros(g);
for ele= 1:m
   dL_dh_ele= zeros(6);
   for sample= 1:spe
      s= -1 + sample/div;  % Element coordinate system position
      % Shape function values for the current sample point:
      N1r= -1/2*(s - 1);
      N2r= 1/2*(s + 1);
      nr= [N1r; 0; 0; N2r; 0; 0];
      N1b= 1/4*(2 - 3*s + s^3);
      N2b= he/8*(1 - s - s^2 + s^3);
      N3b= 1/4*(2 + 3*s - s^3);
      N4b= he/8*(-1 - s + s^2 + s^3);
      nbz= [0; N1b; N2b; 0; N3b; N4b];
      % Derivatives of shape function values for the current sample point:
      dN1b_dx= (-3 + 3*s^2)/(2*he);
      dN2b_dx= 1/4*(-1 - 2*s + 3*s^2);
      dN3b_dx= (3 - 3*s^2)/(2*he);
      dN4b_dx= 1/4*(-1 + 2*s + 3*s^2);
      dnbz_dx= [0; dN1b_dx; dN2b_dx; 0; dN3b_dx; dN4b_dx];
      % Transformed interpolation matrix for the current sample point:
      N_Tr= [nr'-h*scales(1)/2*dnbz_dx'
                nbz'                  ];
      sample_x= ((ele - 1)*N1r + ele*N2r)*he;  % Sample point position in FE
      sample_y= 0;                             % global coordinate system
      sample_z= 0;
      % Sample point direction cosines
      psi_x= (laser_x - sample_x)/sqrt((laser_x - sample_x)^2 + ...
       ((laser_y - sample_y)^2 + (laser_z - sample_z)^2));
      psi_z= (laser_z - sample_z)/sqrt((laser_z - sample_z)^2 + ...
       ((laser_x - sample_x)^2 + (laser_y - sample_y)^2));
      psi= [psi_x; psi_z];
      Psi= psi*psi';
      dN_Tr_dh= [-scales(1)/2*dnbz_dx'
                 zeros(1, 6) ];
      dL_dh_ele= dL_dh_ele + 2*dN_Tr_dh'*Psi*N_Tr;
   end
   % Placing the element derivative into the global matrix:
   for a= 1:6
      for b= 1:6
         dL_dh(ic(ele, a), ic(ele, b))= dL_dh(ic(ele, a), ic(ele, b)) + ...
          dL_dh_ele(a, b);
      end
   end
end
dL_dh= enf_bc_matrix(dL_dh, g);
```

```
function dZ_dh= make_dZ_dh(w, rho, E, Len, b, h, alpha, beta, m, g, p, ...
 ic, scales)
% Function to return the value of the derivative of the impedance matrix w.r.t.
% the beam height, h.
dM_dh= make_Mass(rho, Len, b, 1, m, p, ic, scales);
dK_dh= make_dK_dh(rho, E, Len, b, h, m, p, ic, scales);
dC_dh= alpha*dM_dh + beta*dK_dh;
% Enforce the boundary condition:
dK_dh= enf_bc_matrix(dK_dh, g);
dM_dh= enf_bc_matrix(dM_dh, g);
dC_dh= enf_bc_matrix(dC_dh, g);
dZ_dh= [w*dM_dh - 1/w*dK_dh          dC_dh
               dC_dh         -w*dM_dh + 1/w*dK_dh];
```

```
function dK_dh= make_dK_dh(rho, E, Len, b, h, m, p, ic, scales)
% Returns the derivative of the stiffness matrix w.r.t. the beam height, h.
he= Len/m;
% Derivative of the axial stiffness matrix:
ka= (E*scales(2)*b*scales(1)/he)*[1 -1; -1 1];
% Derivative of the z-dir bending stiffness matrix:
kbz= (3*E*scales(2)*b*h^2*scales(1)^3/(6*he^2))*[ 6/he   3     -6/he   3
                                                  3      2*he  -3      he
                                                 -6/he  -3      6/he  -3
                                                  3      he    -3      2*he];
% Derivative of the element stiffness matrix:
kele= [ka(1, 1)  0          0           ka(1, 2)  0          0
       0         kbz(1, 1)  kbz(1, 2)   0         kbz(1, 3)  kbz(1, 4)
       0         kbz(2, 1)  kbz(2, 2)   0         kbz(2, 3)  kbz(2, 4)
       ka(2, 1)  0          0           ka(2, 2)  0          0
       0         kbz(3, 1)  kbz(3, 2)   0         kbz(3, 3)  kbz(3, 4)
       0         kbz(4, 1)  kbz(4, 2)   0         kbz(4, 3)  kbz(4, 4)];
% Assemble the derivative of the global stiffness matrix:
dK_dh= zeros(3*m+3, 3*m+3);
for ele= 1:m
   for a= 1:6
      for b= 1:6
         dK_dh(ic(ele, a), ic(ele, b))= dK_dh(ic(ele, a), ic(ele, b)) + ...
          kele(a, b);
      end
   end
end
```

```
function dZ_dE= make_dZ_dE(w, rho, Len, b, h, beta, m, g, p, ic, scales)
% Function to return the value of the derivative of the impedance matrix w.r.t.
% the elastic modulus, E.
dK_dE= make_Stiff(rho, 1, Len, b, h, m, p, ic, scales);
dC_dE= beta*dK_dE;
% Enforce the boundary condition:
dK_dE= enf_bc_matrix(dK_dE, g);
dC_dE= enf_bc_matrix(dC_dE, g);
dZ_dE= [-1/w*dK_dE   dC_dE
  dC_dE    1/w*dK_dE];
```

```
function f= obj_fun(x, known, mgp, ic, scales, dl_dh, vlRe, vlIm, spe)
% This function returns the value of the objective function in f.
m= mgp(1);
g= mgp(2);
p= mgp(3);
num_par= length(x);
L= make_L(known(4), x(1), m, g, p, ic, spe, scales);
Q= [   L     zeros(p)
    zeros(p)    L    ];
[lRe, lIm]= make_l(known(4), x(1), m, g, p, ic, vlRe, vlIm, spe, scales);
c= -2*[lRe; lIm];
force= make_force(m, g);
Z= make_Imp(known(1), known(2), x(2), known(4), known(5), x(1), ...
 known(7), known(8), m, g, p, ic, scales);
Y= inv(Z);
f= force'*Y'*Q*Y*force + c'*Y*force;
```

```
function df= grad(x, known, mgp, ic, scales, dl_dh, vlRe, vlIm, spe)
% This function returns the gradients of the objective function.
m= mgp(1);
g= mgp(2);
p= mgp(3);
num_par= length(x);
L= make_L(known(4), x(1), m, g, p, ic, spe, scales);
Q= [L          zeros(p)
    zeros(p)   L       ];
[lRe, lIm]= make_l(known(4), x(1), m, g, p, ic, vlRe, vlIm, spe, scales);
c= -2*[lRe; lIm];
dL_dh= make_dL_dh(known(4), x(1), m, g, p, ic, spe, scales);
dQ_dh= [ dL_dh   zeros(p)
        zeros(p) dL_dh  ];
dc_dh= -2*scales(1)*[dl_dh(:, 1); dl_dh(:, 2)];
Z= make_Imp(known(1), known(2), x(2), known(4), known(5), x(1), ...
  known(7), known(8), m, g, p, ic, scales);
Y= inv(Z);
dZ_dh= make_dZ_dh(known(1), known(2), x(2), known(4), known(5), ...
  x(1), known(7), known(8), m, g, p, ic, scales);
dZ_dE= make_dZ_dE(known(1), known(2), known(4), known(5), x(1), ...
 known(8), m, g, p, ic, scales);
force= make_force(m, g);
df_dh= -2*force'*Y'*Q*Y*dZ_dh*Y*force - c'*Y*dZ_dh*Y*force ...
 + force'*Y'*dQ_dh*Y*force + dc_dh'*Y*force;
df_dE= -2*force'*Y'*Q*Y*dZ_dE*Y*force - c'*Y*dZ_dE*Y*force;
df= [df_dh; df_dE];
```

```
function plt= plot_res(d_sol, m, g, p, ic, spe, num_par)
% Function that creates a plot of the residuals between the measured velocities
% solution velocities.
% Correct parameter values:
w= 200*2*pi;    % driving frequency
rho= 2800;      % density of the material
E= 75e9;        % elastic modulus
Len= 1.381;     % length of the beam
b= 0.01270;     % beam width
h= 0.08255;     % beam height
alpha= 0;       % proportional(to mass) damping constant
beta= 1.4e-5;   % proportional(to stiffness) damping constant
he= Len/m;
scales= ones(num_par, 1);
Z= make_Imp(w, rho, E, Len, b, h, alpha, beta, m, g, p, ic, scales);
force= make_force(m, g);
% Correct FE solution(used to generate the data):
d_actual= Z\force;
d_actual= [zeros(3, 1); d_actual(1:p); zeros(3, 1); d_actual(p+1:2*p)];
d_sol= [zeros(3, 1); d_sol(1:p); zeros(3, 1); d_sol(p+1:2*p)];
div= (spe+1)/2;  % reciprocal of equal length space between each sample within
                 % the elements in terms of the dimensionless element coordinate
                 % system
% Assumed position of the laser source w.r.t. the origin of the FE model global
% coodinate system:
laser_x= 2.0;
laser_y= 0;
laser_z= 2.0;
% Error between the assumed laser position and the actual position:
delta= 2e-3;
delta_x= delta;
delta_z= -delta;
% Initialize variables:
j= 1;
sse= 0;
% Generate "experimental" data:
for ele= 1:m
   for sample= 1:spe
      s= -1 + sample/div;  % Element coordinate system position
      % Shape function values for the current sample point:
      N1r= -1/2*(s - 1);
      N2r= 1/2*(s + 1);
      nr= [N1r; 0; 0; N2r; 0; 0];
      N1b= 1/4*(2 - 3*s + s^3);
      N2b= he/8*(1 - s - s^2 + s^3);
      N3b= 1/4*(2 + 3*s - s^3);
      N4b= he/8*(-1 - s + s^2 + s^3);
      nbz= [0; N1b; N2b; 0; N3b; N4b];
      % Derivatives of shape functions for the current sample point:
      dN1b_dx= (-3 + 3*s^2)/(2*he);
      dN2b_dx= 1/4*(-1 - 2*s + 3*s^2);
      dN3b_dx= (3 - 3*s^2)/(2*he);
      dN4b_dx= 1/4*(-1 + 2*s + 3*s^2);
      dnbz_dx= [0; dN1b_dx; dN2b_dx; 0; dN3b_dx; dN4b_dx];
      N_Tr= [nr' - h/2*dnbz_dx'
                nbz'         ];
      % Surface velocities predicted by the model:
      uRe= N_Tr*d_sol(3*ele-2:3*ele+3);
      uIm= N_Tr*d_sol(3*ele-2+g:3*ele+3+g);
      wmag_sol= sign(atan2(uIm(2), uRe(2)))*abs(uRe(2) + i*uIm(2));
      sample_x= ((ele - 1)*N1r + ele*N2r)*he;  % Assumed sample point position
      sample_y= 0;                             % in FE global coordinate system
      sample_z= 0;                             % (actual sample point location
                                               % is different)
      % Sample point direction cosines
      psi_x= (laser_x - sample_x)/sqrt((laser_x - sample_x)^2 + ...
       ((laser_y - sample_y)^2 + (laser_z - sample_z)^2));
      psi_z= (laser_z - sample_z)/sqrt((laser_z - sample_z)^2 + ...
       ((laser_x - sample_x)^2 + (laser_y - sample_y)^2));
      psi= [psi_x; psi_z];
      % Values to find the actual sample point velocity:
      theta= acos(psi_z);
      b= delta_x - delta_z*tan(theta);
      b_s= 2*b/he;
      if (s+b_s)>1
         error('s is greater than 1')
      end
      N1rd= -1/2*((s+b_s) - 1);
      N2rd= 1/2*((s+b_s) + 1);
      nrd= [N1rd; 0; 0; N2rd; 0; 0];
      dN1b_dxd= (-3 + 3*(s+b_s)^2)/(2*he);
      dN2b_dxd= 1/4*(-1 - 2*(s+b_s) + 3*(s+b_s)^2);
      dN3b_dxd= (3 - 3*(s+b_s)^2)/(2*he);
      dN4b_dxd= 1/4*(-1 + 2*(s+b_s) + 3*(s+b_s)^2);
      dnbz_dxd= [0; dN1b_dxd; dN2b_dxd; 0; dN3b_dxd; dN4b_dxd];
      N1bd= 1/4*(2 - 3*(s+b_s) + (s+b_s)^3);
      N2bd= he/8*(1 - (s+b_s) - (s+b_s)^2 + (s+b_s)^3);
      N3bd= 1/4*(2 + 3*(s+b_s) - (s+b_s)^3);
      N4bd= he/8*(-1 - (s+b_s) + (s+b_s)^2 + (s+b_s)^3);
      nbzd= [0; N1bd; N2bd; 0; N3bd; N4bd];
      % Transformed interpolation matrix for the current sample point:
      N_Trd= [nrd'- h/2*dnbz_dxd'
                nbzd'         ];
      % Surface velocities(different from the velocities at the assumed sample
      % point):
      uRe= N_Trd*d_actual(3*ele-2:3*ele+3);
      uIm= N_Trd*d_actual(3*ele-2+g:3*ele+3+g);
      xs(j)= sample_x;
      wmag_meas= sign(atan2(uIm(2), uRe(2)))*abs(uRe(2) + i*uIm(2));
      error(j)= wmag_meas - wmag_sol;
      sse= sse + (wmag_meas - wmag_sol)^2;
      sn(j)= abs(wmag_sol/error(j));
      j= j+1;
   end
end
error_bar= mean(error);  % Mean error(Mean of the residuals)
mse= sse/(spe*m - 2);     % Mean squared error(Variance of residuals)
snr= mean(sn);
```

```
% Make the residual plot:
plt= stem(xs, error);
axis([0 1.4 -4e-4 4e-4])
set(gca, 'FontName', 'Times')
set(gca, 'FontSize', 12)
xlabel('Length Along Beam, m')
ylabel('Residuals for Signed Velocity Magnitude(w_{meas} - w_{mod}), m/s')
t= text(0.6, -2e-4, ['Mean error= ', num2str(error_bar, '%5g')]);
h= get(t);
set(t, 'FontName', 'Times')
set(t, 'FontUnits', 'points')
set(t, 'FontSize', 12)
t= text(0.6, -2.5e-4, ['Mean Squared Error= ', num2str(mse, '%5g')])
h= get(t);
set(t, 'FontName', 'Times')
set(t, 'FontUnits', 'points')
set(t, 'FontSize', 12)
t= text(0.6, -3.0e-4, ['Mean signal-to-noise= ', num2str(snr, '%5g')])
h= get(t);
set(t, 'FontName', 'Times')
set(t, 'FontUnits', 'points')
set(t, 'FontSize', 12)
fs= 12;
x= 5.65;
y= 4.46;
set(gcf, 'Units', 'inches');
pos= get(gcf, 'Position');
ppos=get(gcf, 'PaperSize');
set(gcf, 'Position', [pos(1) pos(2) x y], ...
 'DefaultTextFontSize', fs, 'DefaultAxesFontSize', fs);
set(gcf, 'PaperPosition', [ppos(1) ppos(2) x y]);
print -deps -loose resEh.eps
```

Vita

Eric ten Siethoff

Eric graduated with a Bachelor of Science degree from the Department of Aerospace Engineering at Virginia Tech in May 1996. During his undergraduate studies, Eric spent several semesters away from school working in internship positions at three different companies: Grumman Space Station Integration Division, ARINC Research Corporation, and General Dynamics Electric Boat Division. Immediately following completion of his undergraduate degree, Eric enrolled in graduate school in the Department of Mechanical Engineering at Virginia Tech. After recieving a Master of Science degree, Eric will be employed as an engineer at MPR Associates Incorporated, a consulting firm located in Alexandria, Virginia.