

MATLAB Implementation of an Operational Modal Analysis Technique for Vibration-Based Structural Health Monitoring

by

Alejandro P. Ojeda

Bachelor of Science in Electrical Science and Engineering, MIT, 2010

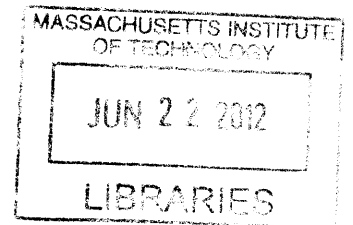
Master of Engineering in Electrical Engineering and Computer Science, MIT, 2011

Submitted to the Department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Civil and Environmental Engineering
at the Massachusetts Institute of Technology


June 2012

© 2012 Massachusetts Institute of Technology. All rights reserved.

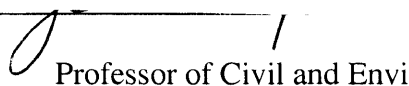
ARCHIVES



Author _____


Department of Civil and Environmental Engineering
May 09, 2012

Certified by _____


Jerome J. Connor
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by _____


Heidi M. Nepf
Chair, Departmental Committee for Graduate Students

MATLAB Implementation of an Operational Modal Analysis Technique for Vibration-Based Structural Health Monitoring

by
Alejandro P. Ojeda

Submitted to the
Department of Civil and Environmental Engineering on May 09, 2012

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Civil and Environmental Engineering

ABSTRACT

Vibration-based structural health monitoring (SHM) has become an attractive solution for the global monitoring and evaluation of damage in structures. Numerous damage detection schemes used in vibration-based SHM require knowledge of the modal properties of the structure under evaluation in its current state. The technique of operational modal analysis allows for these modal properties to be obtained by using the structure's dynamic response to ambient excitation. Using *MATLAB*, a type of operational modal analysis technique called time domain decomposition (TDD) based on [15] was implemented. The *MATLAB* TDD implementation was applied to the dynamic responses from two finite element models of simply-supported beams and their modal frequencies and shapes were extracted. The first three modal frequencies were obtained with less than 6 percent error from the actual values and the fundamental mode shape values obtained contained negligible deviations from the actual mode shape values. However, the higher order mode shapes obtained were more inaccurate, suggesting limitations to the current *MATLAB* TDD implementation. Lastly, changes to the moment of inertia of the simply-supported beam models were used to simulate damage in the finite element models and cause their fundamental mode frequency to change. The *MATLAB* TDD implementation was able to distinguish changes in the fundamental frequency of both finite element models with a resolution of approximately 1.7 radians per second (7.2 percent).

Thesis Supervisor: Jerome J. Connor
Title: Professor of Civil and Environmental Engineering

Acknowledgments

I would like to thank Professor Jerome Connor for his advice and supervision over the course of the project. I would also like to thank Mr. Pierre Ghisbain for his technical assistance and help at different stages of the project. Lastly, I would also like to thank my family and Nan McGarry for their constant help, support, and encouragement throughout the research project and the production of this document.

Table of Contents

1 Introduction	11
1.1 Overview	12
2 Structural Health Monitoring	13
2.1 Structural Health Monitoring Overview.....	13
2.1.1 Operational Evaluation.....	15
2.1.2 Data Acquisition and Normalization.....	15
2.1.3 Feature Extraction	16
2.1.4 Damage Detection and Model Development.....	16
2.2 Vibration-Based Structural Health Monitoring	17
2.2.1 Sources of Excitation.....	18
2.2.2 Data Acquisition Instruments.....	19
2.2.3 Damage Detection Based on Modal Parameters.....	20
2.2.4 Damage Detection Based on Matrix Methods.....	21
2.2.5 Damage Detection Based on Machine Learning.....	22
3 Vibration-Based Modal Analysis	23
3.1 Experimental Modal Analysis.....	23
3.2 Operational Modal Analysis	25
3.2.1 Frequency Domain OMA.....	25
3.2.2 Time Domain OMA.....	26
3.3 Time Domain Decomposition Theoretical Implementation.....	27
3.4 Time Domain Decomposition Software Implementation.....	31

4 TDD Implementation Results	35
41 Finite Element Model Description.....	35
42 11-Node Simply-Supported Beam.....	39
4.2.1 Symmetric Excitation (11-Node Model).....	40
4.2.2 Asymmetric Excitation (11-Node Model).....	45
4.2.3 Actual & Extracted Modal Properties Comparison (11-Node Model)....	47
43 21-Node Simply-Supported Beam.....	49
4.3.1 Symmetric Excitation (21-Node Model).....	50
4.3.2 Asymmetric Excitation (21-Node Model).....	52
4.3.3 Actual & Extracted Modal Properties Comparison (21-Node Model)...	53
44 Damage Detection.....	56
5 Conclusion	59
5.1 Summary.....	59
Appendix A	61
A-1 TDD <i>MATLAB</i> Code Implementation.....	61
A-2 Modal Frequency Finder Function.....	63
A-3 Band-Pass Filter Function	63
Appendix B	65
B-1 OpenSees Simply Supported Beam Model (11-Nodes).....	65
B-2 OpenSees Simply Supported Beam Model (21-Nodes).....	67
B-3 OpenSees Code for Modal Properties Extraction (11-Nodes).....	69
Bibliography	72

List of Figures

2-1 Classification Levels for SHM Structures.....	14
2-2 Advantages and Disadvantages of Excitation Methods for Dynamic Analysis.....	17
2-3 Overview of a Vibration-Based SHM Scheme.....	19
3-1 Overview of the <i>MATLAB</i> TDD Implementation	32
3-2 Band-Pass Filter Magnitude Response.....	34
4-1 Simply-Supported Beam Properties.....	36
4-2 Gaussian White Noise Excitation	37
4-3 Gaussian White Noise Ground Acceleration to Excite Symmetric Modes.....	38
4-4 Gaussian White Noise Ground Acceleration to Excite Asymmetric Modes.....	38
4-5 Actual Modal Frequencies and Shapes Obtained from OpenSees.....	39
4-6 11-Node, 10-Element Model Description.....	40
4-7 Time History Acceleration Response Plot of Node Number 11.....	41
4-8 Extracted Values for the First and Third Modal Frequencies (11-Node Model).....	42
4-9 Power Spectrum Plot for Node Number 3 (11-Node Model).....	43
4-10 Power Spectrum Plot for Node Number 6 (11-Node Model).....	43
4-11 Fundamental Mode Shape Obtained (11-Node Model).....	44
4-12 Third Mode Shape Obtained (11-Node Model).....	44
4-13 Extracted Values for the Second Mode Frequency (11-Node Model).....	45
4-14 Power Spectrum Plot for Node Number 11 (11-Node Model).....	46
4-15 Second Mode Shape Obtained (11-Node Model).....	46

4-16 Mode Frequency Comparison (11-Node Model).....	47
4-17 Mode Shapes Extracted (11-Node Model).....	47
4-18 Fundamental Mode Comparison (11-Node Model).....	48
4-19 Second Mode Comparison (11-Node Model).....	48
4-20 Third Mode Comparison (11-Node Model).....	49
4-21 Extracted Values for the First and Third Modal Frequencies (21-Node Model).....	50
4-22 Power Spectrum Plot for Node Number 3 (21-Node Model).....	51
4-23 Fundamental Mode Shape Obtained (21-Node Model).....	51
4-24 Third Mode Shape Obtained (21-Node Model).....	52
4-25 Extracted Values for the Second Mode Frequency (21-Node Model).....	52
4-26 Second Mode Shape Obtained (21-Node Model).....	53
4-27 Mode Frequency Comparison (21-Node Model).....	54
4-28 Mode Shapes Extracted (21-Node Model).....	54
4-29 Fundamental Mode Comparison (21-Node Model).....	55
4-30 Second Mode Comparison (21-Node Model).....	55
4-31 Third Mode Comparison (21-Node Model).....	56
4-32 Actual Modal Frequencies of <i>Damaged</i> Models.....	57
4-33 Actual and Extracted Fundamental Mode Frequencies of <i>Damaged</i> Models.....	58

Chapter 1

Introduction

Any structure, whether it is a building, home, bridge, road, etc., undergoes physical deterioration over time. Physical deterioration is a result of accumulated damage that can be due to structural aging, exposure to environmental elements, increased operational loads, extreme weather effects, etc. Damage can be defined as changes in the material and/or geometric properties of a structure that affect its current or future performance. These may include changes to any structural connections or boundary conditions as well. As damage accumulates over time in a structure, functional deficiencies or even failure may result.

Virtually all public and private entities that are owners or operators of critical civil infrastructure are interested in detecting and monitoring damage on demand in their assets using a combination of hardware and software tools. Among other things, such monitoring capabilities allow for the maintenance and repair of assets to be managed more effectively. This is because a condition-based, as opposed to a time-based, maintenance philosophy can be implemented. A condition-based maintenance philosophy may allow for capital and human resources to be allocated more efficiently in part because maintenance is only performed if needed instead of following a periodic schedule.

The United State's bridge infrastructure is one example where the monitoring of structural damage would be especially beneficial. The US highways contain more than 590,000 bridges with over 3.2 billion square feet of bridge deck [1]. Many of these highway bridges serve as critical links to airports, international borders, and military installations and must be tested and evaluated regularly. The large and costly undertaking to inspect, maintain and repair these bridges requires effective management efforts and efficient allocation of resources. Having the capability of monitoring the structural condition or *health* of these bridges on demand would be an invaluable tool for such an endeavor.

1.1 Overview

The thesis is structured as follows: Chapter 2 presents an overview of structural health monitoring and vibration-based structural health monitoring. The first half of Chapter 3 discusses vibration-based modal analysis including various experimental and operational modal analysis (OMA) techniques. The second half details the theoretical and *MATLAB* implementation of a type of OMA technique called time domain decomposition (TDD). The results obtained from applying the TDD technique to the dynamic responses of two finite element models are presented and discussed in Chapter 4. Lastly, Chapter 5 includes a summary of the thesis as well as the conclusions reached.

Chapter 2

Structural Health Monitoring

2.1 Structural Health Monitoring Overview

A condition or damage identification strategy for new and rehabilitated aerospace, mechanical, and civil infrastructure is referred to as structural health monitoring (SHM). A properly designed SHM scheme would determine the serviceability, reliability and durability of a structure in order to allow owners or decision makers to effectively allocate resources [2]. The use of an SHM scheme would allow for the periodic investigation of structures during operation and give insight into the determination of which structures would need occasional maintenance, repair, retrofit or replacement.

SHM schemes are commonly classified based on the robustness or capability of the scheme. The larger the classification level, the more sophisticated the scheme. Hence, a level IV scheme would be much more robust than a level I scheme. Typically, a level I SHM scheme has the capability to determine if damage is present in a structure. A level II scheme can identify damage and determine its location. A level III scheme is capable of identifying damage,

determining its location, and estimating the severity of the damage. Lastly, a level IV scheme has the capability of identifying damage, determining its location, estimating the severity of the damage and predicting the durability of the structure. Figure 2-1 summarizes the four classification levels for SHM schemes.

Classification Level	Scheme Robustness / Capability
Level I	Identify damage
Level II	Identify damage; determine damage location
Level III	Identify damage; determine damage location; estimate severity
Level IV	Identify damage; determine damage location; estimate severity; estimate the durability of the structure

Figure 2-1: Classification Levels for SHM Schemes

In general, SHM makes use of *in-situ* sensing that allows for an analysis of the response characteristics of a particular structure. SHM can be used for near real-time or short-term evaluation, such as for structural condition screening after an extreme event, as well as long-term evaluation during which the performance of a structure is periodically updated to reflect any damage accumulation.

A common way to organize a discussion on SHM is to describe it as a four-step process, which includes an operational evaluation step, a data acquisition and normalization step, a feature extraction step and a damage detection step.

2.1.1 Operational Evaluation

The first step in SHM is called the operational evaluation step, during which a feasibility analysis is performed. This feasibility analysis will impose limitations on the SHM scheme with regards to the parameters that will be selected for monitoring, the types of instruments that will be used for monitoring, and the manner in which monitoring will be performed.

Operational evaluation encompasses four major tasks: the first is an assessment of life-safety or economic circumstances that justify performing SHM. Once this has been performed, a definition of damage must be developed for the structure under study. The definition of damage tends to be structure specific and takes advantage of any unique features the structure may have. Following this, the condition under which monitoring will take place will be determined. Lastly, any limitations with regards to data acquisition in the operational environment of the structure will be assessed [3].

2.1.2 Data Acquisition and Normalization

The data acquisition and normalization portion of the SHM scheme entails the selection of input excitation methods as well as output data acquisition and processing. The locations, quantities and types of data acquisition sensors will be determined. The data storage media along with the sampling rate will also be selected at this stage. The selections will be driven mainly by the operational environment of the structure and cost considerations.

Data normalization is an important part of SHM due to the fact that data can be obtained under varying conditions. Most data normalization schemes will be specific to each structure, but

a common technique is to normalize the measured outputs by the inputs. Normalization can be performed with relation to a spatial as well as a temporal reference.

2.1.3 Feature Extraction

The identification and extraction of structural features that provide insight into the condition of a structure is performed in the feature extraction step of the SHM scheme. The best types of features to extract are specific to each structure and its operating conditions. The more common feature extraction algorithms are based on relating the structural output response to structural damage observations made. Other methods apply engineering flaws to a structure in order to determine the parameters that are sensitive to damage. Finite element models are sometimes used to perform numerical simulations and gain insight into the condition of the structure. Lastly, due to the large amount of data that will be collected from a given structure, robust data reduction techniques should be implemented.

2.1.4 Damage Detection and Model Development

The last step in an SHM scheme is that of damage detection. The type of damage detection algorithms that are used depend on the type of SHM scheme and the data that is available for use. Commonly extracted features used for damage detection include the modal properties (modal shapes, frequencies and damping) and structural properties (stiffness and flexibility). Machine learning algorithms can be implemented for damage detection if there is incomplete or insufficient knowledge of structural features. Finally, a damage model of the structure will be developed.

2.2 Vibration-Based Structural Health Monitoring

For many structures, and in particular civil structures, SHM schemes that implement global vibration-based methods for assessing damage have become most desirable. These SHM schemes utilize the dynamic characteristics of the structure in order to assess the overall condition. A typical vibration-based SHM scheme uses an excitation source that serves as an input to the structure in order to observe its output vibration response. The scheme then proceeds to extract various features from the vibration response, which will be used for damage detection. Vibration-based SHM schemes may use features such as the modal parameters of the structure, or implement matrix methods or machine learning algorithms to detect damage. Figure 2-2 is a flow chart depicting the various steps involved in a vibration-based SHM scheme.

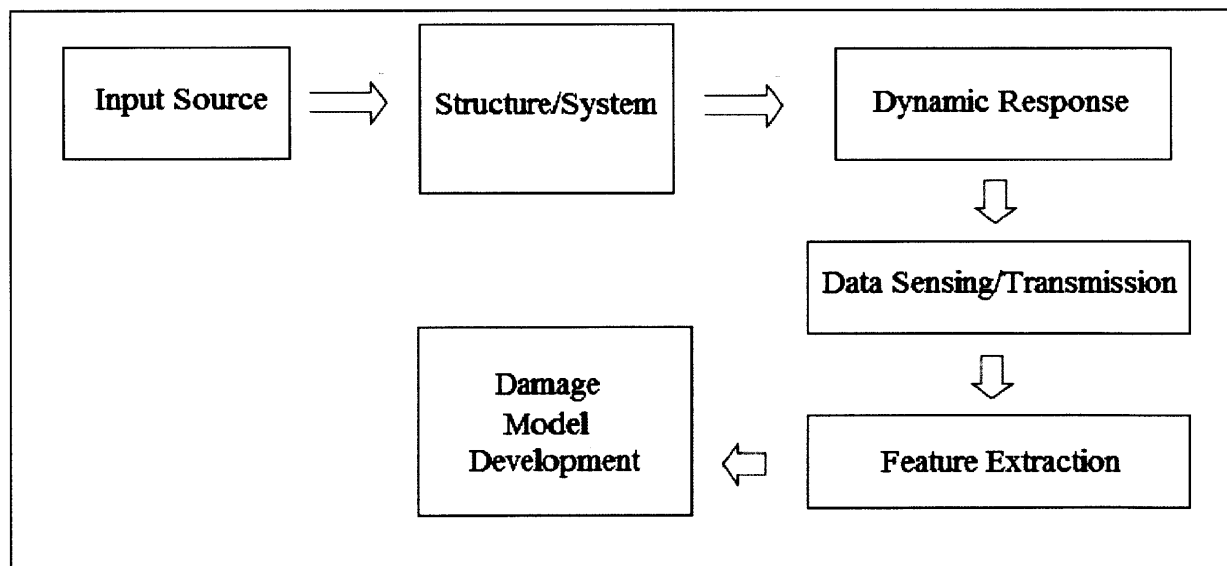


Figure 2-2: Overview of a Vibration-Based SHM Scheme

Vibration-based SHM techniques are different from localized damage detection schemes such as acoustic emission and ultrasound. Most localized damage detection schemes are not very practical for monitoring large civil structures. Ideally, vibration-based SHM schemes would work in tandem with localized damage detection techniques. Vibration-based SHM would first perform a global damage analysis of the structure and, once damage has been identified and located, localized techniques can be used to further investigate the extent of the damage.

2.2.1 Sources of Excitation

Techniques that enable the dynamic testing of structures can be classified as either input-output methods or output only methods.

Input-output methods involve applying a known input, such as a Dirac impulse load, to the structure and measuring its output vibration response. To create a desired forcing function, devices such as impact hammers, shakers or drop weights are used. Output only methods involve measuring the output vibration of the structure in response to ambient excitation. Ambient excitations include loadings that may occur from wind, micro-earthquakes, vehicular and pedestrian traffic, and ocean waves [4]. The ambient excitation is assumed to be a stationary random process containing a flat frequency spectrum [5].

Input-output methods have the advantage of suppressing noise effects on the structural response, which output-only methods do not have. However, input-output solutions tend to be more expensive than output-only solutions and usually create a disruption in the regular use of the structure. Output-only solutions allow for continuous uninterrupted SHM because the

structural response can be measured under operational conditions. Nonetheless, output only methods have the disadvantage that excitation forces are unknown and are not always able to excite the higher frequency modes of the structure. Figure 2-3 summarizes the advantages and disadvantages of input-output and output-only methods.

Method Type	Advantages	Disadvantages
Input-Output	Control over input	Higher Cost No continuous monitoring Disruption in structure use
Output Only	Lower Cost Continuous monitoring No disruption in structure use	No control over input Limited high frequency mode resolution Difficult to extract damping values

Figure 2-3: Advantages and Disadvantages of Excitation Methods for Dynamic Analysis

2.2.2 Data Acquisition Instruments

The most commonly used data acquisition instruments for vibration-based damage detection are acceleration and displacement transducers. These instruments measure the acceleration or displacement time history signals of a structure and convert them into voltage signals. The mass and size of the data acquisition system must be small compared with the mass of the structure in order to not interfere with the output response.

For civil structures, accelerometers tend to be the most widely used data acquisition instruments. This is due to their ability to operate over a large range of frequency responses, their ease of installment

and high measurement accuracy [6]. There are two types of accelerometers commercially available, one which measures acceleration based on capacitance measurements and another which uses piezoelectric deformations.

2.2.3 Damage Detection Based on Modal Parameters

Modal parameters may be extracted from a dynamic analysis of a structure in order to serve as inputs to a vibration-based damage detection technique. The types of modal parameters used can be modal frequencies and/or modal shapes.

Methods that are based on modal frequencies have the inherent assumption that any shifts in the natural frequencies of a given structure indicate that a change in its structural properties has taken place. The advantage of using modal frequencies is that it allows for damage identification to be performed with relatively easy implementations. However, most algorithms that make use of the modal frequencies for damage detection can only provide a qualitative estimation of damage severity and no spatial information on damage. Therefore, only level I SHM schemes can be implemented using modal frequencies.

The use of modal shapes for damage detection allows for spatial information on damage to be obtained, but this requires a minimum number of measurement points in order to properly characterize the mode shapes. The number of measurement points increase with increasing mode frequency. With the use of modal shapes, various damage detection algorithms have the capability of achieving level III robustness.

The use of both modal frequencies and shapes is commonly implemented in SHM schemes as it allows for the detection, localization, and estimation of damage in a structure (level III). The literature on damage detection algorithms is vast and it is not the intent of this thesis to survey the literature. However, some damage detection algorithms that use modal frequencies are [7-8] and some that are based on modal shapes are [9-10].

2.2.4 Damage Detection Based on Matrix Methods

Damage detection algorithms can make use of certain structural properties in order to determine the condition of a structure. Matrices of these structural properties are built from structural parameters and can be used to identify, locate and quantitatively estimate damage. These so called matrix methods compare the undamaged to the damaged system matrices in order to detect damage. Typical matrices that are built and used in damage detection are the stiffness, flexibility, and/or damping matrices.

However, like all damage detection algorithms, there are advantages and disadvantages to matrix methods. In general, the use of system matrices allows a robustness level of up to III to be implemented. The disadvantages are that the structural properties needed to build the matrices are not always precisely known and the solutions are approximations. Moreover, some matrix methods require a minimum level of change in stiffness or flexibility in order to detect a change. An example of a damage detection algorithm that makes use of a structure's global stiffness matrix is [11].

2.2.5 Damage Detection Based on Machine Learning

Machine learning algorithms are used for damage detection when there are measurement or parameter uncertainties and/or insufficient information. Machine learning algorithms can be divided into two categories, either supervised learning or unsupervised learning. If data from both the undamaged and damaged state is available, then supervised statistical learning algorithms are implemented. Supervised learning algorithms infer a classifier or regression function from the given training data, which can be the undamaged data features extracted from the structure of interest. Learning algorithms such as artificial neural networks or genetic algorithms are common in the literature. However, if only data from the undamaged state of a structure is available, unsupervised learning algorithms are implemented. Unsupervised learning algorithms must use data mining methods such as clustering or blind source separation as no training data exists to evaluate a potential solution [12].

In general, learning algorithms can solve complex problems in which there is a lack of data fairly effectively. Most of the algorithms in the literature can achieve up to level III robustness [13-14]. However, limitations to these techniques include convergence issues when handling large volumes of data as well as the fact that the use of training data requires prior knowledge of the damage state of the structure.

Chapter 3

Vibration-Based Modal Analysis

As discussed in the previous chapters, the extraction of structural features is an important part of vibration-based SHM. This chapter and the remainder of the thesis will focus on obtaining the modal properties (both shapes and frequencies) from the structural dynamic output response. In particular, the focus will be on a technique called time domain decomposition, which uses the structure's dynamic response to ambient excitations in order to extract its modal properties.

3.1 Experimental Modal Analysis

Experimental modal analysis (EMA) is a process whereby the modal properties of a structure are extracted. Since the modal properties can be used to model or characterize a

structure, damage detection algorithms can use these features to determine and monitor the condition of said structure.

EMA makes use of a forced input excitation as well as the measured dynamic response of the structure. The input forcing function is typically created using impact hammers, shakers or drop weights. The output dynamic response is usually measured using transducers that measure position, velocity or acceleration.

Compared to other technologies used in generating forced structural input excitation, the impact hammer tends to be more cost effective, more portable and easier to operate. However, the impact hammer also tends to have a poor signal to noise ratio. Another commonly used technology for generating forced input excitation is a drop weight. The drop weight allows for control of the amplitude of the input forcing function and for low frequencies to be excited but, like the impact hammer, also has poor signal to noise ratio and cannot be used for continuous monitoring. Lastly, the shaker can excite higher modal frequencies (up to 100 Hz) than both the impact hammer or drop weight technologies can, but is more expensive and difficult to install [15].

A large number of EMA algorithms have been developed in both the time domain and frequency domain. Most of these algorithms implement single-input-single-output (SISO), single-input-multiple-output (SIMO), or multiple-input-multiple-output (MIMO) techniques. The technique of EMA has been widely used for mechanical and aerospace applications, as well as civil applications.

3.2 Operational Modal Analysis

Similar to EMA, operational modal analysis (OMA) is a technique whereby the modal properties of a structure are extracted. The OMA technique extracts the modal properties of a structure by analyzing its dynamic output response to its operational input excitation. In other words, OMA analyzes a structure's response based solely on ambient loads that it is exposed to [15]. The lack of a need to generate forced input excitation significantly reduces the cost of SHM and allows for SHM to be performed while the structure remains in operation. For civil applications, OMA has become a very appealing technique for the extraction of modal parameters to be used in long term vibration-based SHM studies.

OMA techniques are usually classified as frequency domain OMA or time domain OMA.

A discussion on each of these two areas of OMA follows below:

3.2.1 Frequency Domain OMA

There exist a large number of frequency domain OMA techniques in the technical literature. The simplest technique, called peak-picking [16], extracts the natural frequencies of a structure by finding the peaks of the power spectrum response plot. In theory, each peak corresponds to a natural mode of the structure. The major limitation to this technique is its inability to differentiate between modal frequencies that are very close in value. Furthermore, it also requires the structure to have low damping values.

A second group of frequency domain OMA techniques utilizes the singular value decomposition of the matrix containing the cross spectrum of the structure's output response. Typically, these methods are called frequency domain decomposition (FDD) [17]. After singular value decomposition is performed, the modal shapes and scaled frequencies are obtained. FDD type methods also allow for modal damping to be extracted. These methods have an advantage over peak-picking methods in that neither fairly spaced natural frequency values nor low damping values are required. However, the major drawback of FDD-type techniques is that the values of damping obtained may be biased.

A third group of frequency based OMA methods that allow for the extraction of modal shapes and frequencies use an approximation called the least squares complex frequency (LSCF) domain approximation [15].

3.2.2 Time Domain OMA

Similar to frequency domain OMA, numerous time domain OMA techniques are present in the technical literature. One of the most popular techniques is the natural excitation technique (NExT). The NExT approach assumes that the operational excitation that the structure is exposed to can be modeled as a white noise process. This method establishes the cross correlation between two random dynamic response measurements in the structure and expresses these signals as summations of decaying sinusoidal functions [18]. Once the decaying sinusoidal functions are created, time domain modal identification algorithms such as the eigensystem realization algorithm (ERA) or the polyreference complex exponential technique (PRCE) can be applied in order to obtain the modal parameters.

Along with NExT, other commonly used time domain OMA methods are the autoregressive moving average (ARMA) and stochastic space identification (SSI) methods. The major drawback to time domain OMA methods is that under noisy conditions, the number of vibration modes cannot be estimated accurately. This is due to the fact that since the order of natural modes is unknown, it is difficult to determine the real structural modes from those created by noise effects. Furthermore, due to this mode estimation inaccuracy, many time domain methods are not appropriate for long term monitoring as the algorithms must rely on a user to determine the number of modes manually [15].

3.3 Time Domain Decomposition Theoretical Implementation

A type of OMA technique called time domain decomposition (TDD) is a sort of hybrid between the time domain and frequency domain OMA methods. TDD employs the idea of peak-picking in order to find the modal frequencies and the idea of singular value decomposition in the time domain in order to find the modal shapes. TDD is an attractive technique to implement in this thesis due to the fact that it is computationally efficient, suitable for automation and relatively simple to develop in software. The theoretical framework for the TDD technique is developed in [15] and described below.

The continuous time dynamic acceleration response of a causal, linear time invariant (LTI) system is shown in equation 3-1 as the sum of the modal shapes scaled by their corresponding generalized coordinates.

$$\ddot{\mathbf{u}}(x, t) = \sum_{r=1}^{\infty} \boldsymbol{\varphi}_r(x) \ddot{q}_r(t) \quad (3-1)$$

where $\boldsymbol{\varphi}_r(x)$ is the r^{th} mode shape and $\ddot{q}_r(t)$ is the r^{th} generalized coordinate given as a function of time (t). Using a band-pass filter it is possible to isolate the modal components of the acceleration response. The band-pass filter allows frequencies within a specified range to pass through while attenuating frequencies outside of that range. Applying a band-pass filter with specified cutoff and pass through frequencies to the acceleration response allows for equation 3-2 to be obtained.

$$\ddot{\mathbf{u}}_n(x, t) = \boldsymbol{\varphi}_n(x) \ddot{q}_n(t) \quad (3-2)$$

where $\ddot{\mathbf{u}}_n(x, t)$ is the n^{th} modal contribution to the dynamic acceleration response. The modal frequencies used to create the band-pass filter are obtained using a peak-picking approach as developed by [16].

However, since the acceleration response of real structures will be obtained in a discrete manner, a discrete time representation must be used. If we assume that the system in question contains N_d degrees of freedom and its acceleration time history response contains N_s samples, equation 3-3 can be used as a discrete time representation.

$$[U] = \sum_{r=1}^{N_d} \boldsymbol{\varphi}_r \ddot{q}_r^T \quad (3-3)$$

where U is the response matrix containing N_d rows and N_s columns, $\boldsymbol{\varphi}_r$ the r^{th} mode shape vector containing N_d rows and 1 column, and $\ddot{\mathbf{q}}_r^T$ is the r^{th} generalized coordinate vector containing 1 row and N_s columns. Similar to the continuous time representation, the discrete time acceleration time history response can be filtered at each natural frequency using a band-pass filter. The result is described by equations 3-4 to 3-5, where U_n represents the response matrix for the n^{th} mode.

$$[U_n] = \boldsymbol{\varphi}_n \ddot{\mathbf{q}}_n^T \quad (3-4)$$

$$\begin{bmatrix} \ddot{u}_{1n}(1) & \dots & \ddot{u}_{1n}(N_s) \\ \vdots & \ddots & \vdots \\ \ddot{u}_{N_d n}(1) & \dots & \ddot{u}_{N_d n}(N_s) \end{bmatrix} = \begin{bmatrix} \varphi_{1n} \\ \vdots \\ \varphi_{N_d n} \end{bmatrix} [\ddot{q}_n(1) \quad \dots \quad \ddot{q}_n(N_s)] \quad (3-5)$$

The autocorrelation of the n^{th} mode must then be determined. The n^{th} mode autocorrelation is denoted by E_n and shown in equation 3-6 in terms of the n^{th} mode response matrix.

$$[E_n] = [U_n][U_n]^T \quad (3-6)$$

This operation involves obtaining the square of the n^{th} mode acceleration response matrix, which gives a measurement proportional to the *power* of the n^{th} mode time history signal. This is accomplished by substituting equation 3-4 into 3-6 and rearranging terms. The algebra is shown below from equations 3-7 to 3-11.

$$[E_n] = \boldsymbol{\varphi}_n \ddot{\mathbf{q}}_n^T \ddot{\mathbf{q}}_n \boldsymbol{\varphi}_n^T \quad (3-7)$$

$$[E_n] = \boldsymbol{\varphi}_n Q_n \boldsymbol{\varphi}_n^T \quad (3-8)$$

$$[E_n] = Q_n \boldsymbol{\varphi}_n \boldsymbol{\varphi}_n^T \quad (3-9)$$

$$[E_n] = Q_n \begin{bmatrix} \varphi_{1n} \\ \vdots \\ \varphi_{N_d n} \end{bmatrix} [\varphi_{1n} \quad \dots \quad \varphi_{N_d n}] \quad (3-10)$$

$$[E_n] = Q_n \begin{bmatrix} \varphi_{1n} \varphi_{1n} & \varphi_{1n} \varphi_{2n} & \dots & \varphi_{1n} \varphi_{N_d n} \\ \varphi_{2n} \varphi_{1n} & \ddots & & \vdots \\ \vdots & & & \vdots \\ \varphi_{N_d n} \varphi_{1n} & \dots & \dots & \varphi_{N_d n} \varphi_{N_d n} \end{bmatrix} \quad (3-11)$$

Once the n^{th} mode autocorrelation matrix is obtained, its singular value decomposition (equation 3-12) can be obtained because it is a matrix of rank 1.

$$[E_n] = USV^* \quad (3-12)$$

$$[E_n] = [\mathbf{u}_1 \quad \dots \quad \mathbf{u}_{N_d}] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_{N_d} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_{N_d}^T \end{bmatrix} \quad (3-13)$$

Equation 3-12 depicts the singular value decomposition matrices of E_n . The U matrix contains the eigenvectors, the V^* matrix contains the transposed eigenvectors, and the S matrix contains the eigenvalues of the n^{th} mode autocorrelation matrix. If the measured response contained no noise, then all the eigenvectors and eigenvalues would be zero with the exception of \mathbf{u}_1 and λ_1 . These are the scaled n^{th} mode shape and frequency of the system, respectively.

If noise is present in the response measurement, then other eigenvectors and eigenvalues will be nonzero. Nevertheless, due to the fact that the signal was filtered, the noise contribution to the modal values will be much smaller than the contribution from the natural n^{th} mode.

Lastly, in order to obtain a single degree of freedom representation for the n^{th} mode, equation 3-4 is pre-multiplied by φ_n^T as shown in equation 3-14. The generalized coordinate of the n^{th} mode response is shown in equation 3-15.

$$\varphi_n^T [U_n] = \varphi_n^T \varphi_n \ddot{q}_n^T \quad (3-14)$$

$$\ddot{q}_n^T = \frac{\varphi_n^T [U_n]}{\varphi_n^T \varphi_n} \quad (3-15)$$

This single degree of freedom representation of the n^{th} mode is useful because the mode shape, frequency and damping values can be obtained using time domain modal identification techniques.

3.4 Time Domain Decomposition Software Implementation

The software implementation of the time domain decomposition technique was done in *MATLAB*. The code contains three files all of which are in Appendix-A. The first file in Appendix A-1 is a script file that implements the TDD technique. The second file in Appendix A-2 is a function that the script file calls in order to obtain the modal frequencies of the system. Lastly, the third file in Appendix A-3 is a function that is called by the script file which creates the band-pass filter object in order to filter the acceleration time history data around a given mode frequency. An overview of the TDD implementation in *MATLAB* is shown in figure 3-1 below.

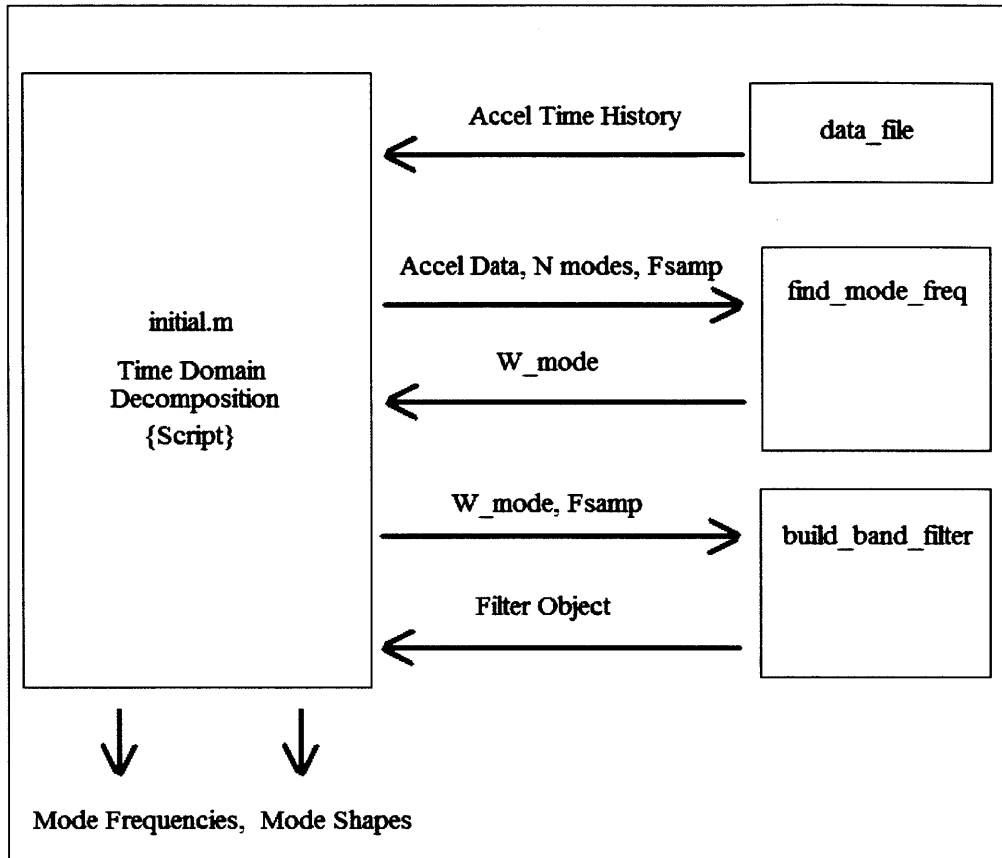


Figure 3-1: Overview of the *MATLAB* TDD Implementation

The *MATLAB* script first proceeds to open, read and arrange the node acceleration time history data contained in a file. In the code shown in Appendix-A, the type of file to be read is a text file, which contains the output acceleration time history data for each node in the structure in a specified column. The script can be amended to open and read almost any file format and furthermore, optional code has been added in a commented section of the script which shows how to read data from an excel file. The script then obtains the specified number of modal frequencies from the `find_mode_freq` function. The modal frequencies are used to create a band-pass filter object using the function `build_band_filter`. Each acceleration time history response is

filtered with the band-pass object and arranged to create the response matrix for each respective mode. From the n^{th} mode response matrix, the n^{th} mode autocorrelation matrix can be found. As mentioned previously, since the each autocorrelation matrix is of rank 1, singular value decomposition can be performed. Lastly, each scaled mode shape is arranged into a matrix denoted by U in the code. The mode shapes can then be scaled as desired. Note that in the code, matrix V is the transpose of U and matrix S contains the eigenvalues of the n^{th} autocorrelation matrix.

As discussed above, the function `find_mode_freq` obtains a specific number of modal frequencies from the structural response. The function takes in as inputs an acceleration time history vector, the data sampling frequency and the number of modes desired. It obtains the power spectrum curve for each node by first computing the Fast Fourier Transform of the acceleration time history signal. Then it multiplies the transformed signal by its complex conjugate and scales the value by the number of samples points contained in the acceleration time history signal. Once the power spectrum curve is obtained, the algorithm finds its peaks. In theory, the peaks correspond to the natural frequencies of the system. Lastly, `find_mode_freq` outputs the frequency values (in radians per second) that correspond to the power spectrum curve peaks.

The function `build_band_filter` takes in as inputs a frequency value and signal sampling frequency value. It proceeds to build a band-pass filter around the given input frequency. The magnitude response function of the filter is shown in figure 3-2. Furthermore, figure 3-2 shows the amount of attenuation that frequencies other than the function input frequency will undergo. Frequencies that are within two percent above or below the function input frequency will not be attenuated. Frequencies that are more than three percent above or below the function input

frequency will be attenuated by more than 120 decibels. Those frequencies in between two and three percent, above and below the function input frequency, will be attenuated somewhere between 0 and 120 decibels. The function implements a Butterworth type of band-pass filter in order to have a flat frequency response in the filter pass-band.

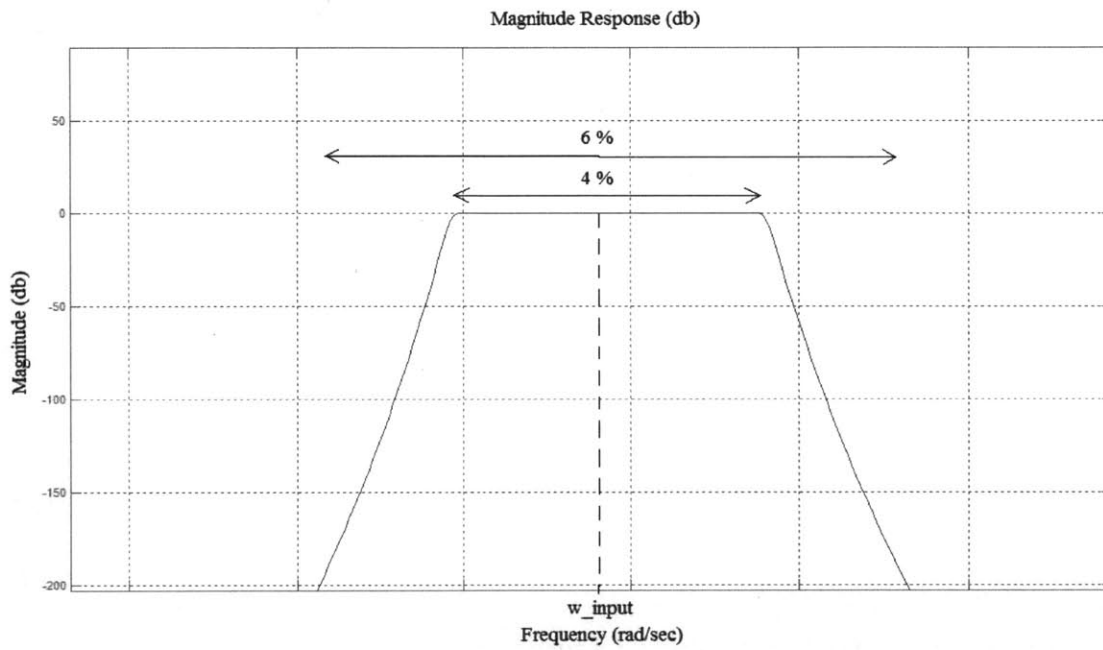


Figure 3-2: Band-Pass Filter Magnitude Response

Chapter 4

TDD Implementation Results

This chapter will provide an overview of the results obtained from the *MATLAB* implementation of TDD technique. The first part of the chapter will describe in detail the finite element models used to generate output dynamic responses. The remainder of the chapter will show the results obtained after applying the TDD implementation to these responses.

4.1 Finite Element Model Description

As previously discussed, the output acceleration response data of a structure under ambient loading is used by the *MATLAB* TDD technique to extract its modal properties. Typically, the acceleration response data is obtained using a series of accelerometers from a particular structure that is built in the field or in a laboratory setting. However, for the purposes of testing and assessing the accuracy of this *MATLAB* TDD implementation, various acceleration

response data sets from two finite element models of a simply-supported beam were used. A finite element model has various advantages over using a real structure to obtain the dynamic responses. One advantage is that the actual modal properties of the structure being model are known exactly and can be used to assess the accuracy of the modal properties extracted using the *MATLAB* TDD implementation. A second advantage is that a finite element model allows great flexibility over the types of loadings that can be applied to the structural model. Lastly, since the finite element model is implemented in software it is an economical solution for testing the TDD implementation.

The type of structure to be modeled in finite element code is a two-dimensional simply-supported beam shown in figure 4-1. The beam is made of steel with a density of 7849.05 [kg/m³] and a modulus of elasticity of 1.999x10¹¹ [N/m²]. The beam has a length of 20 meters, a cross-sectional area of 0.0227 [m²] and a moment of inertia of 8.056x10⁻⁴ [m⁴] about an axis perpendicular to the plane of the page. Furthermore, Rayleigh damping proportional to both stiffness and mass was also modeled.

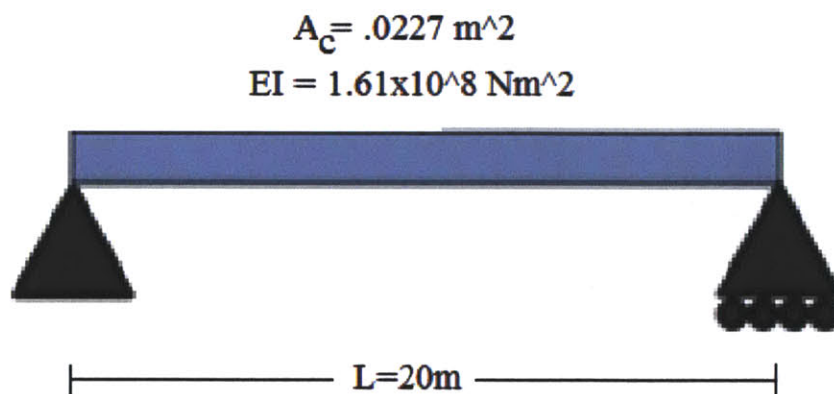


Figure 4-1: Simply-Supported Beam Properties

The open source software framework OpenSees version 2.3.2 was used to construct the finite element models. Two different simply-supported beam models were constructed using OpenSees. The first model contained 11 nodes and the second model contained 21 nodes. For the sake of simplicity, node motion was restricted to the vertical axis only. The OpenSees code for each model can be found in Appendix B.

The input excitation to the models was ground acceleration in the vertical axis modeled as a Gaussian white noise input excitation. A Gaussian white noise excitation is normally distributed with zero mean and finite variance. It is a stationary process with a flat-frequency spectrum which excites all the natural modes of the structure. According to [5], Gaussian white noise excitation is a good approximation of ambient excitation. A typical Gaussian white noise signal is depicted in figure 4-2. The Gaussian white noise signals vary in time and were scaled by the gravitational constant divided by 20 or $.0495 \text{ m/s}^2$.

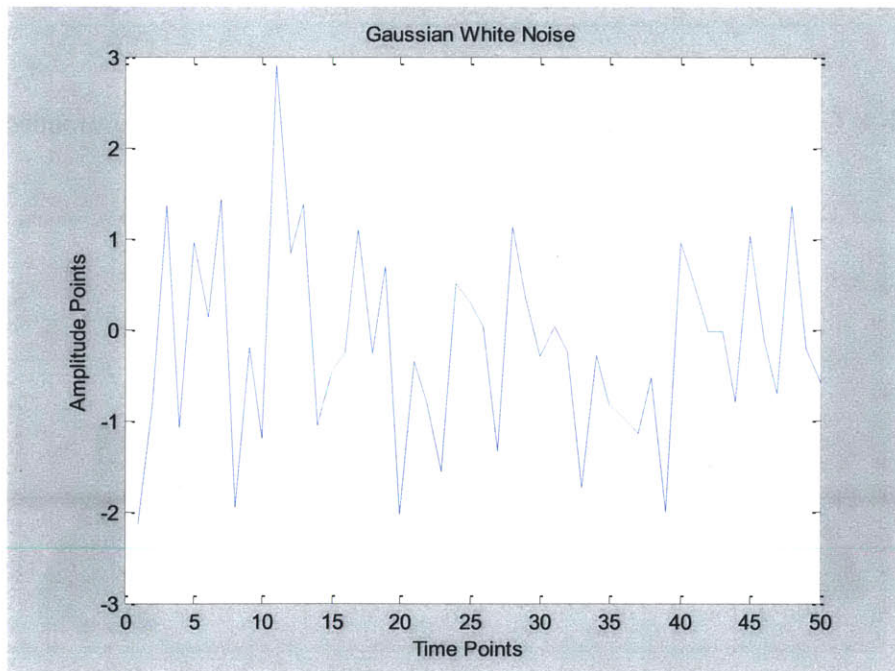


Figure 4-2: Gaussian White Noise Excitation

Due to the fact that in a simply-supported beam the odd numbered modes are symmetric and the even numbered modes are asymmetric, both symmetric and asymmetric ground acceleration time history excitations were used to excite all the modes. The Gaussian white noise excitation was applied as a function of time along the entire length of the beam to excite the symmetric modes in one case and along the left half of the length of the beam to excite the asymmetric modes in another case. Figures 4-3 and 4-4 depict the ground acceleration excitation schemes.

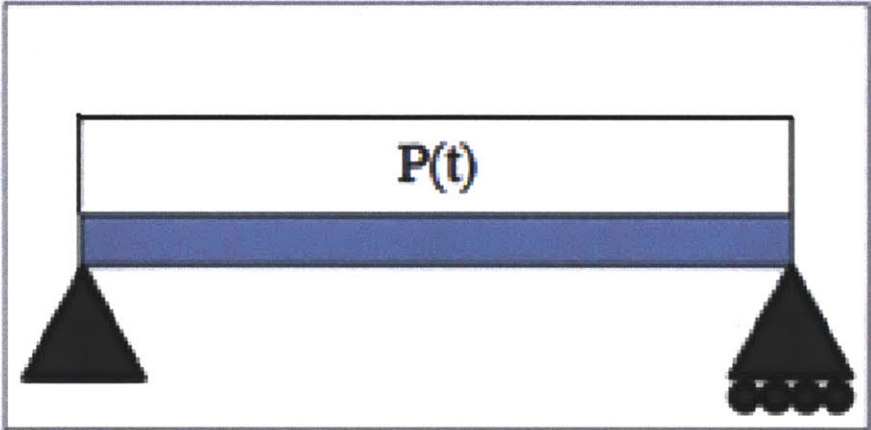


Figure 4-3: Gaussian White Noise Ground Acceleration to Excite Symmetric Modes

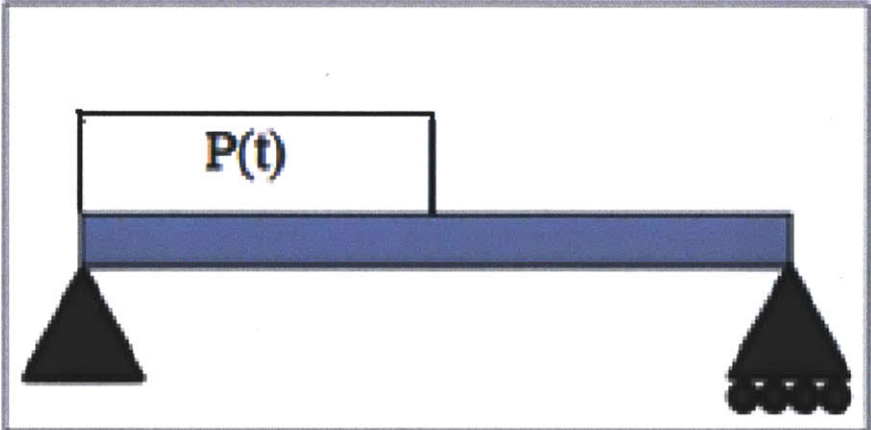


Figure 4-4: Gaussian White Noise Ground Acceleration to Excite Asymmetric Modes

The first three modal shapes and frequencies for this particular simply-supported beam model were obtained from OpenSees using the code in Appendix B-3 and are summarized in figure 4-5. Both the frequencies and the shapes will be used later on in this chapter to assess the accuracy of the values obtained with the *MATLAB* TDD implementation.

Modal Properties	Modal Frequency (rad/sec)	Modal Shape ($0m \leq X \leq 20m$)
Mode 1	23.4575	$\sin\left(\frac{\pi x}{20}\right)$
Mode 2	93.8194	$\sin\left(\frac{2\pi x}{20}\right)$
Mode 3	210.977	$\sin\left(\frac{3\pi x}{20}\right)$

Figure 4-5: Actual Modal Frequencies and Shapes Obtained from OpenSees

4.2 11-Node Simply-Supported Beam Model

The first finite element model created with OpenSees divided the simply-support beam model into 11 nodes and 10 structural elements that link each node. The nodes were spaced evenly along the length of the beam every two meters. Appendix B-1 contains the OpenSees code for this model and figure 4-6 shows how the nodes and structural elements were labeled (nodes are represented by circles and structural elements by straight lines).

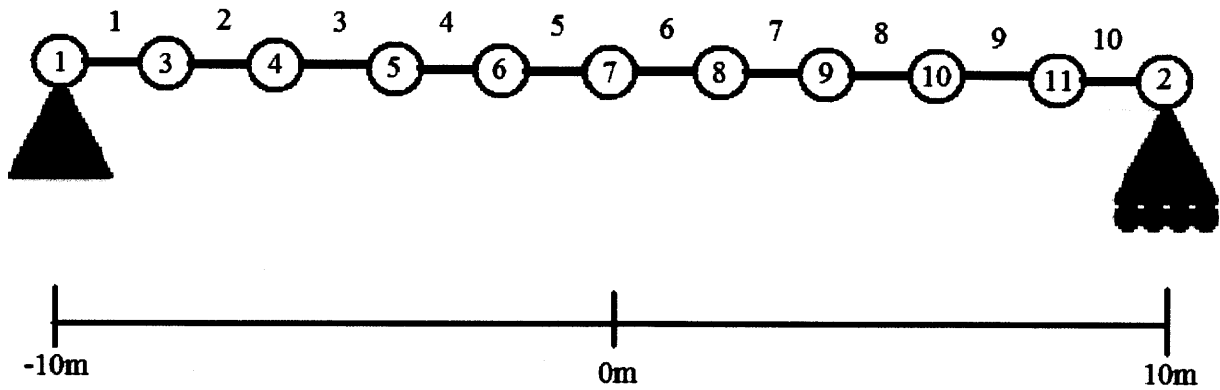


Figure 4-6: 11-Node, 10-Element Model Description

4.2.1 Symmetric Excitation (11-Node Model)

A transient response of the OpenSees finite element simply-supported beam model to symmetrical Gaussian white noise ground acceleration was first simulated. The simulation time step was 1 millisecond and 10,000 time steps were simulated. A typical node time history acceleration response obtained from the OpenSees beam simulation is plotted in figure 4-7.

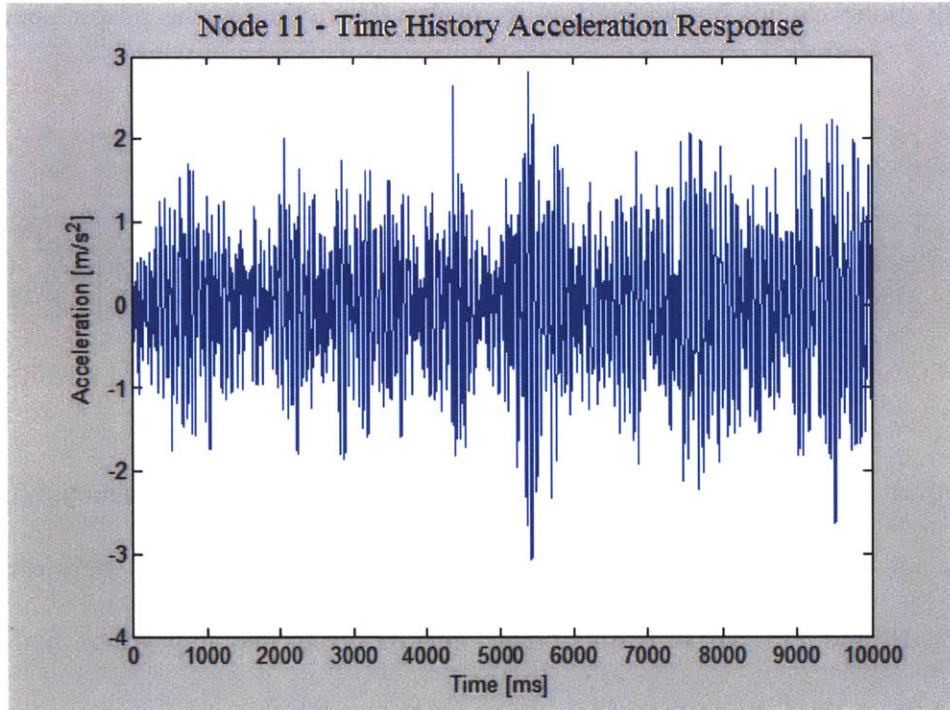


Figure 4-7: Time History Acceleration Response Plot for Node Number 11

Since the input ground acceleration excitation was symmetric, only symmetric modes could be excited. Therefore, the TDD technique implementation was used to obtain the modal properties for the first two odd modes (modes 1 and 3). From each acceleration time history response at a given node, the natural frequencies were extracted. Two different frequencies corresponding to the first and third natural modes of the beam were obtained from each node acceleration time history response. Figure 4-8 summarizes the values that were obtained for the first and third modal frequencies. As mention in chapter 3, the TDD implementation uses a peak-picking approach to extract the frequencies. In theory, the peaks in the power spectrum plot correspond to the natural frequencies of the structure, with the largest peak corresponding to the fundamental frequency. In general, as the mode frequency increases, its corresponding peak in the power spectrum curve will decrease in response to Gaussian white noise excitation. In this

case, the fifth mode cannot be detected as its peak value is below the power spectrum noise level.

Node	3	4	5	6	7	8	9	10	11
$W_1(\text{rad/s})$	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478
$W_3(\text{rad/s})$	214.2566	214.2566	52.1504	52.1504	52.1504	52.1504	214.2566	214.2566	214.2566

Figure 4-8: Extracted Values for the First and Third Modal Frequencies (11-Node Model)

Moreover, figure 4-8 shows that the value for the third mode frequency extracted from each node is not the same. The values extracted from nodes 5 through 8 were different from nodes 3, 4, 9, 10 and 11. These results create uncertainty in deciding which node frequency value is the correct third mode frequency. In order to resolve this issue numerous simulations were run with different Gaussian white noise excitation signals. The extracted third mode frequency value that showed up most frequently was taken to be the correct value. Therefore, the extracted frequency values were 23.2478 radians per second for the first mode and 214.2566 radians per second for the third mode.

Another approach that can be taken to ensure that the correct third mode frequency is extracted is to visually examine the power spectrum plots. In this case, the power spectral plots for node 3 (shown in figure 4-9) and node 6 (shown in figure 4-10) were examined. The power spectrum plot for node 3 contains two distinct and prominent peaks while the power spectrum plot for node 6 contains only one distinct peak. It is likely that the acceleration response of node 6 was driven largely by the fundamental mode and its third mode power spectrum peak was under the power spectrum noise level. By visually examining the power spectrum plots, the modal frequencies can be obtained with more certainty, however, this approach is not suitable for automation.

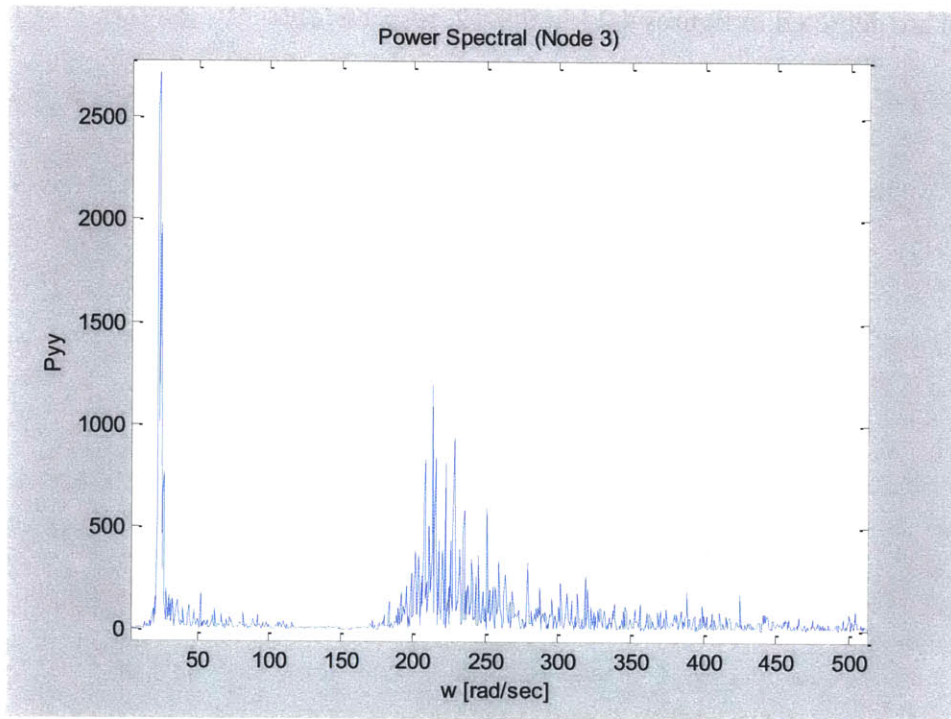


Figure 4-9: Power Spectrum Plot for Node Number 3(11-Node Model)

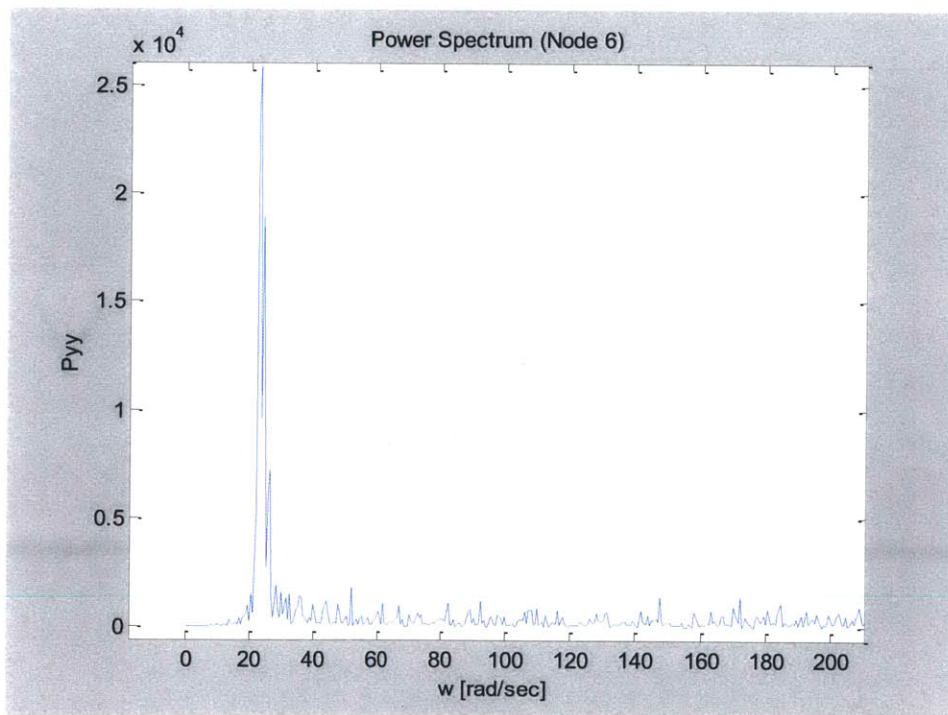


Figure 4-10: Power Spectrum Plot for Node Number 6 (11-Node Model)

After the modal frequencies were extracted, the first and third mode shapes were extracted and are depicted in figures 4-11 and 4-12, respectively.

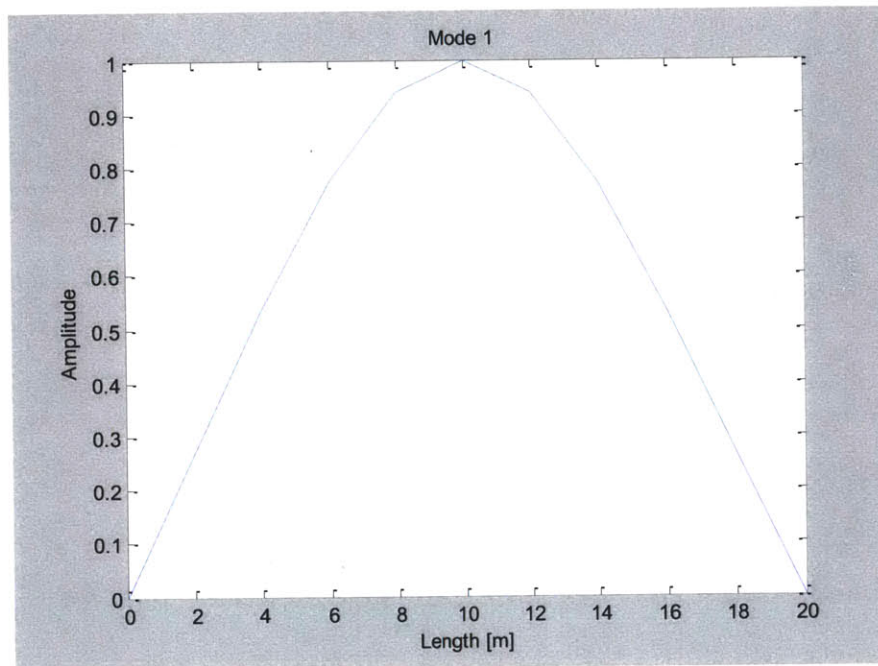


Figure 4-11: Fundamental Mode Shape Obtained (11-Node Model)

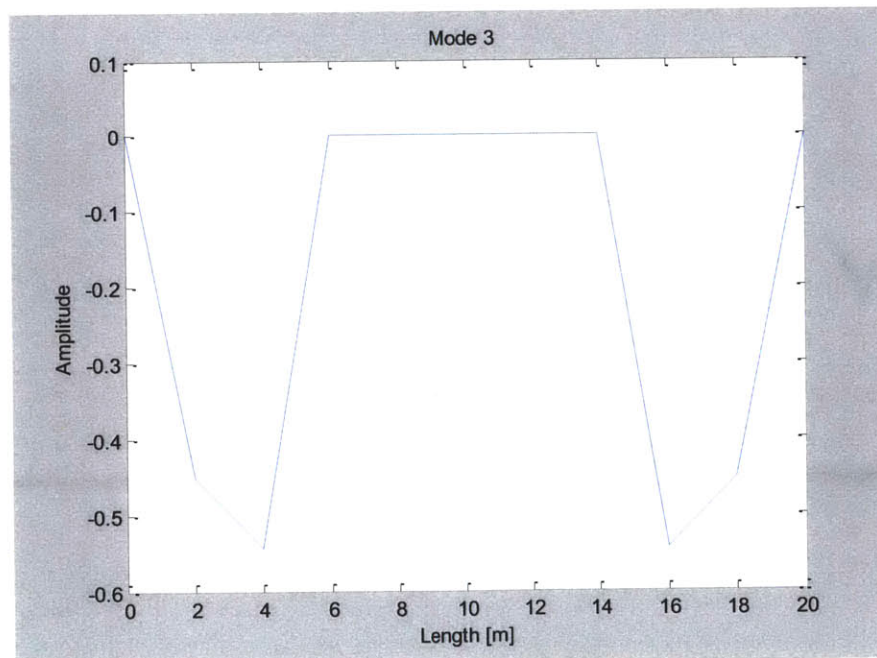


Figure 4-12: Third Mode Shape Obtained (11-Node Model)

4.2.2 Asymmetric Excitation (11-Node Model)

The OpenSees finite element beam model was then simulated subject to asymmetrical Gaussian white noise ground acceleration. Similar to the symmetrical case, the simulation time step was 1 millisecond and 10,000 time steps were simulated. Since the input excitation was asymmetric, only asymmetric modes (even modes) could be excited. Again the TDD technique used a peak-picking approach to extract the second mode frequency. Figure 4-13 summarizes the values that were obtained for the second mode frequency.

Node	3	4	5	6	7	8	9	10	11
$W_2(\text{rad/s})$	52.1504	52.1504	52.1504	52.1504	92.3628	92.3628	92.3628	92.3628	92.3628

Figure 4-13: Extracted Values for the Second Mode Frequency (11-Node Model)

In extracting the second mode frequency values, figure 4-13 shows that different values were obtained. This issue is again resolved by performing several simulations and selecting the value that shows up most frequently to be the correct second mode frequency. In this case that value is 92.3628 radians per second. Figure 4-14 shows the power spectral plot for node number 11 from which the second mode frequency was derived. Lastly, figure 4-15 depicts the shape that was obtained for the second mode.

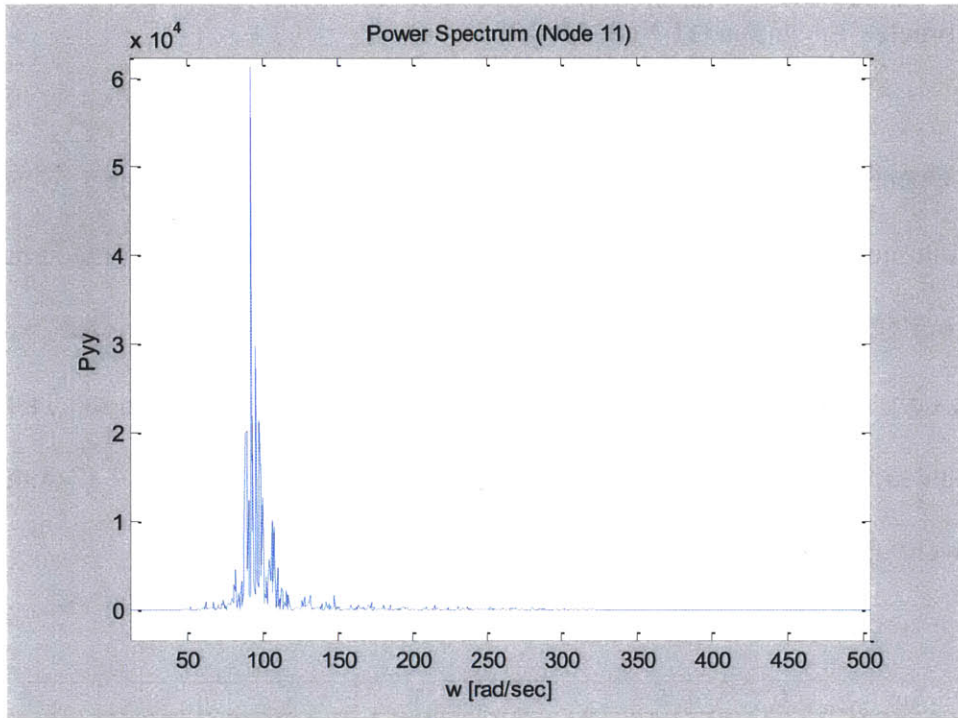


Figure 4-14: Power Spectrum Plot for Node Number 11 (11-Node Model)

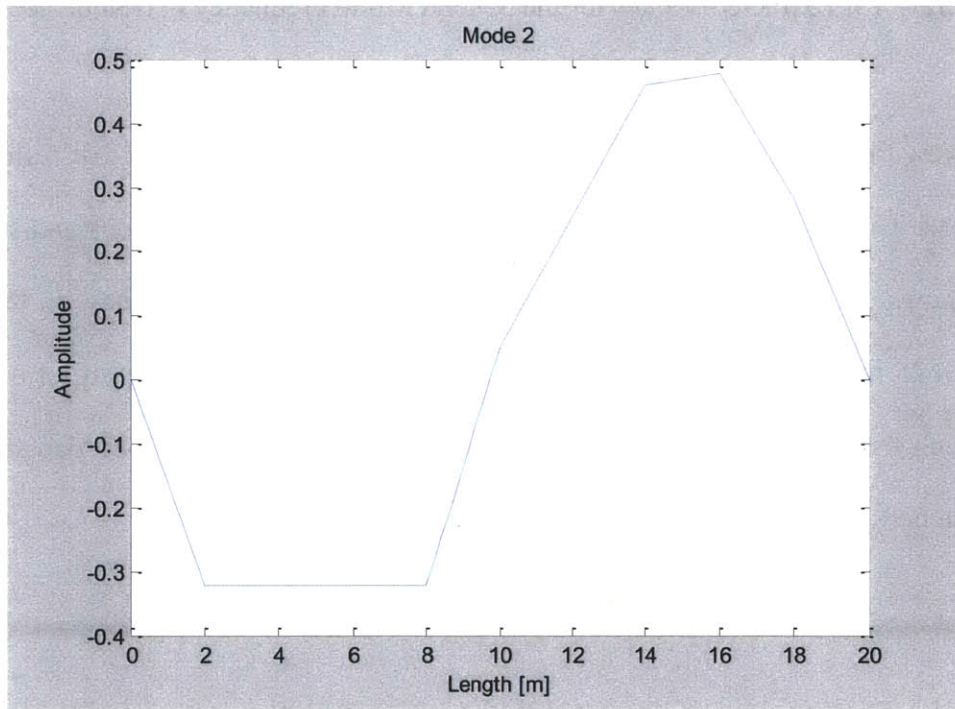


Figure 4-15: Second Mode Shape Obtained (11-Node Model)

4.2.3 Actual and Extracted Modal Properties Comparison (11-Node Model)

Figure 4-16 through 4-20 compare the extracted mode shape and frequency values using the *MATLAB* TDD implementation to the actual mode shape and frequency values obtained from OpenSees. Figure 4-16 depicts that the first three modal frequencies were obtained with less than two percent error. Figures 4-18 through 4-20 offer a visual comparison between the actual and extracted mode shapes values listed in figure 4-17. The extracted fundamental mode shape is almost the same as the actual fundamental mode shape, whereas the extracted second and third mode shapes are close to the actual mode shapes throughout some portions of the beam, but are not as accurate in other portions. The trend to notice is that as the mode number increases, the accuracy decreases. This is expected because as the mode number increases, more nodes are needed in the model to correctly extract the mode shapes.

Mode Frequency	Actual	Extracted	%Error
W ₁ (rad/s)	23.4575	23.2478	0.89
W ₂ (rad/s)	93.8194	92.3628	1.55
W ₃ (rad/s)	210.977	214.2566	1.55

Figure 4-16: Mode Frequency Comparison (11-Node Model)

Mode Shapes			
Length (m)	Fundamental	Second	Third
0	0	0.0000	0.0000
2	0.2672	0.5878	0.8090
4	0.5342	0.5878	0.9713
6	0.7726	0.5878	0.0000
8	0.9397	0.5878	-0.0001
10	1	-0.0090	-0.0001
12	0.9397	-0.4623	-0.0001
14	0.7726	-0.8386	0.0000
16	0.5342	-0.8722	0.9713
18	0.2672	-0.5189	0.8090
20	0	0.0000	0.0000

Figure 4-17: Mode Shapes Extracted (11-Node Model)

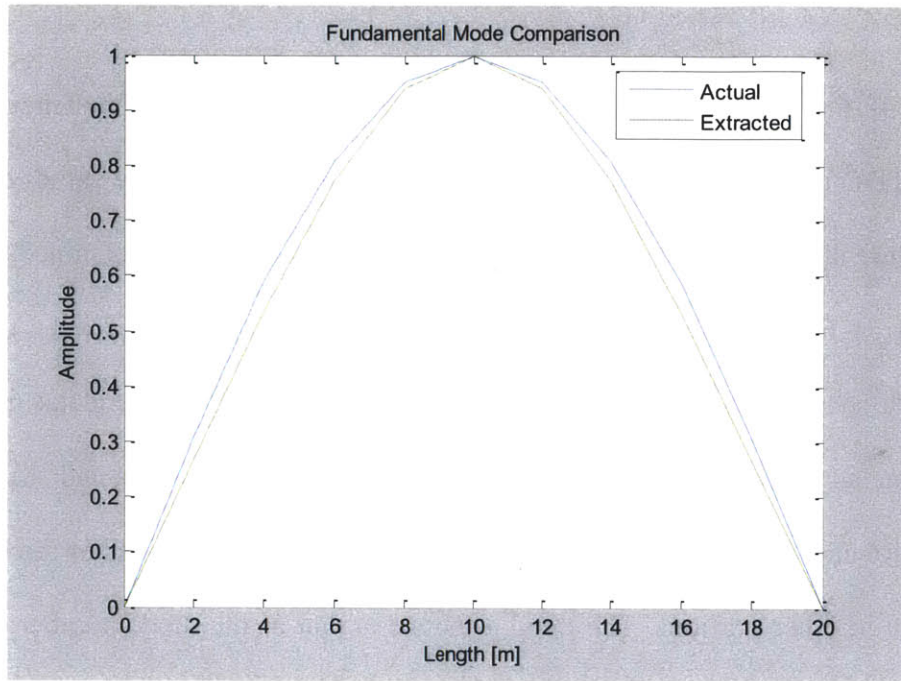


Figure 4-18: Fundamental Mode Comparison (11-Node Model)

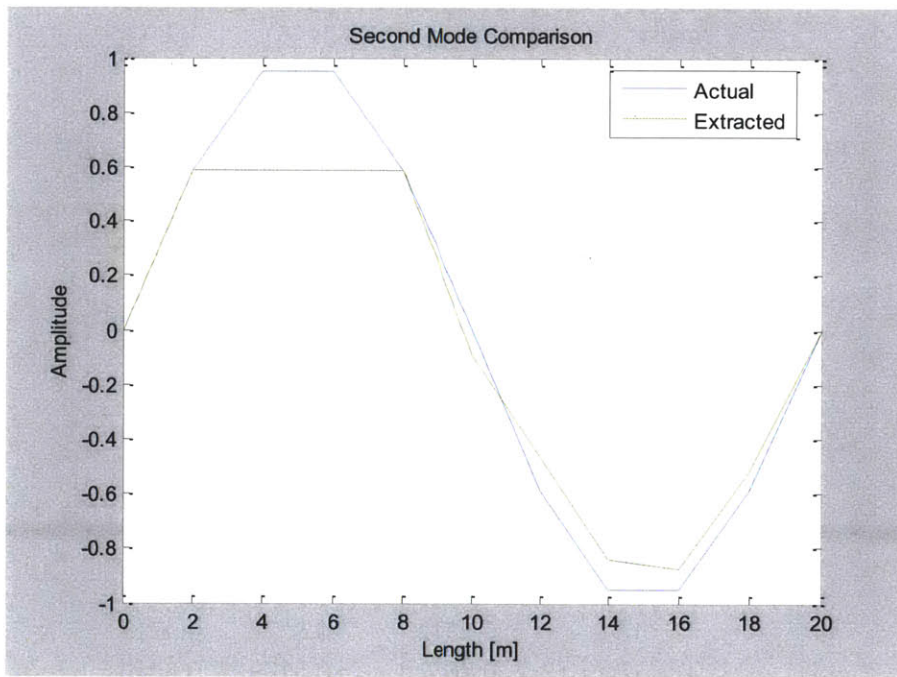


Figure 4-19: Second Mode Comparison (11-Node Model)

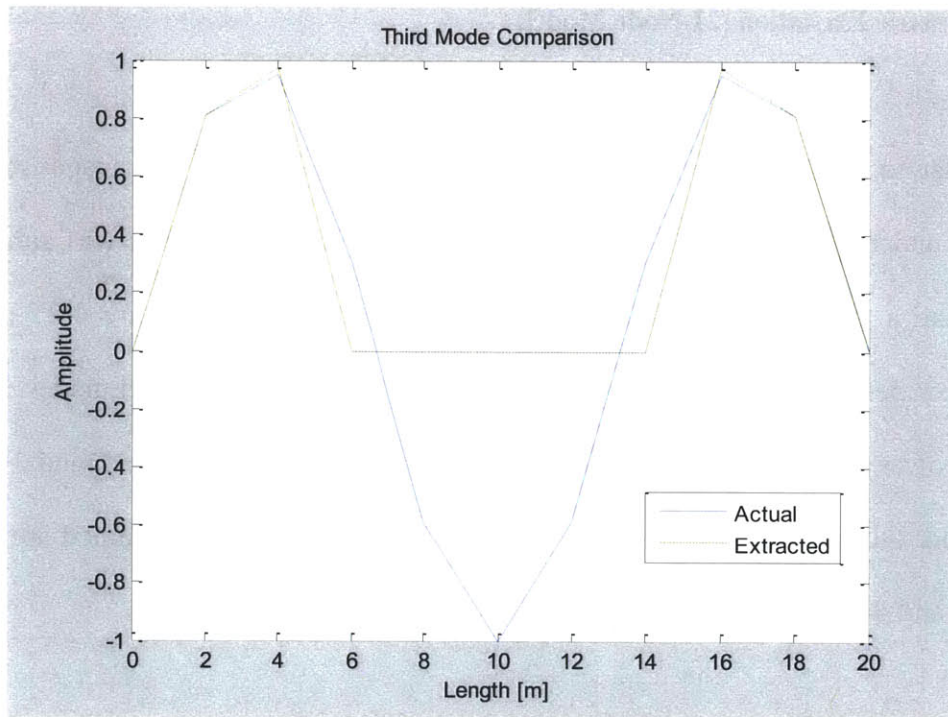


Figure 4-20: Third Mode Comparison (11-Node Model)

4.3 21-Node Simply-Supported Beam

The second finite element model created with OpenSees divided the simply-supported beam model into 21 nodes and 20 structural elements that link each node. The nodes were spaced evenly along the length of the beam every one meter. Appendix B-2 contains the OpenSees code for this model. Similar to figure 4-6, the two support nodes were labeled 1 and 2, while the remainder of the nodes were labeled 3 through 21 from left to right. The structural element links were labeled 1 to 20 from left to right as well.

4.3.1 Symmetric Excitation (21-Node Model)

Similar to the 11-node simply-supported beam model, a transient response of the 21-node model under symmetrical Gaussian white noise ground acceleration was simulated. The simulation had a time step of 1 millisecond and lasted 10 seconds. Figure 4-21 contains the natural frequencies extracted from each node response for the first and third modes. Figure 4-22 depicts the power spectrum for node number 3 from which the first and third modal frequencies were obtained. Furthermore, the first and third mode shapes that were obtained are depicted in figures 4-23 and 4-24, respectively.

Node	3	4	5	6	7	8	9	10	11
W₁(rad/s)	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478
W₃(rad/s)	214.2566	214.2566	214.2566	214.2566	214.2566	229.3363	52.1504	52.1504	52.1504
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
Node	12	13	14	15	16	17	18	19	20 & 21
W₁(rad/s)	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478	23.2478
W₃(rad/s)	52.1504	52.1504	52.1504	229.3363	214.2566	214.2566	214.2566	214.2566	214.2566

Figure 4-21: Extracted Values for the First and Third Modal Frequencies (21-Node Model)

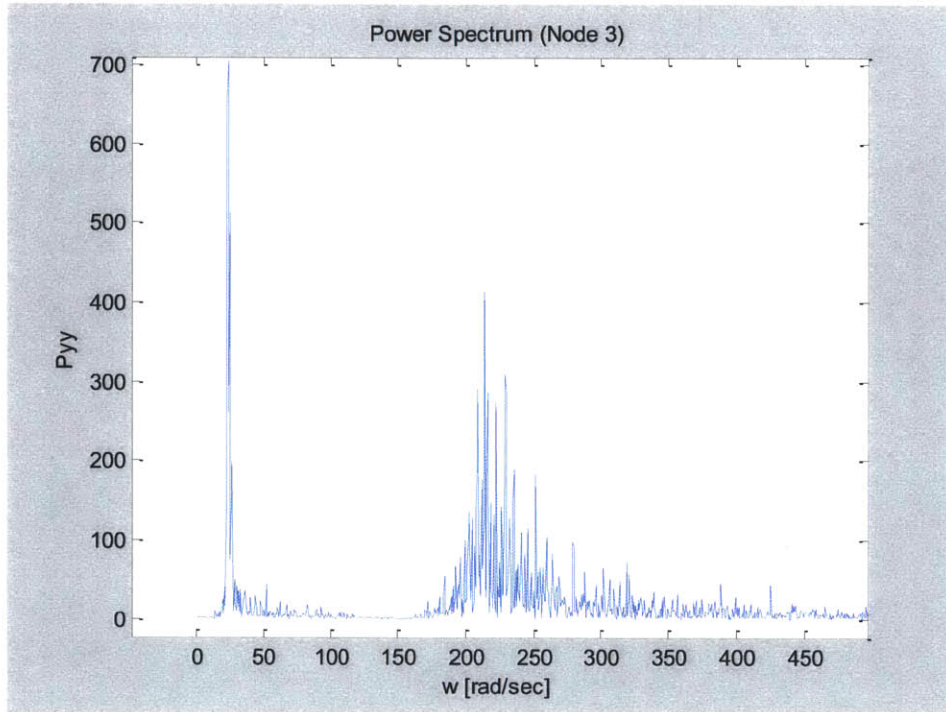


Figure 4-22: Power Spectrum Plot for Node Number 3 (21-Node Model)

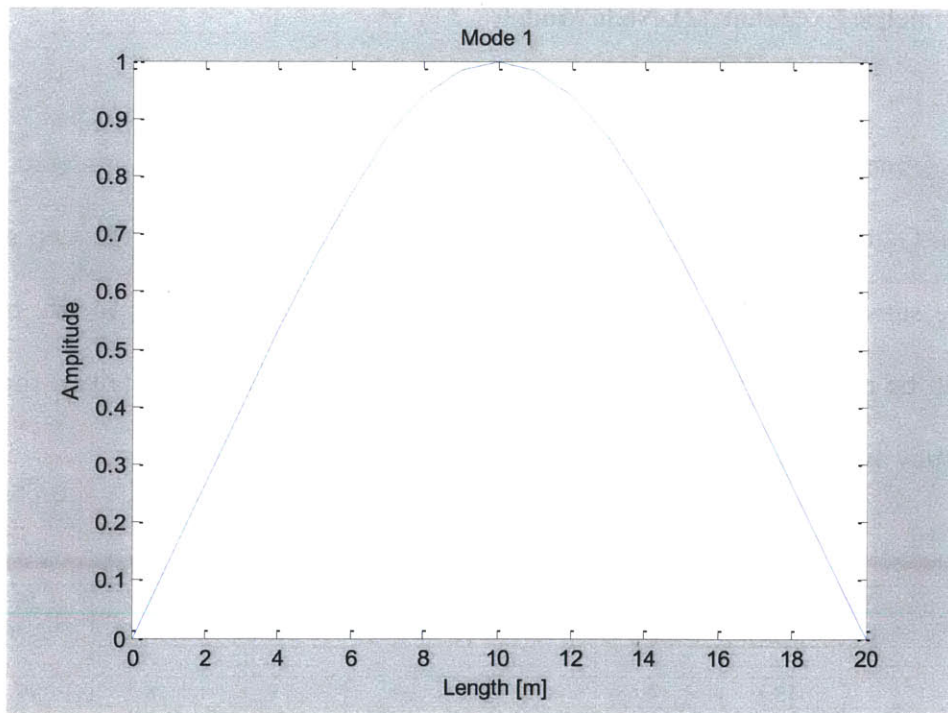


Figure 4-23: Fundamental Mode Shape Obtained (21-Node Model)

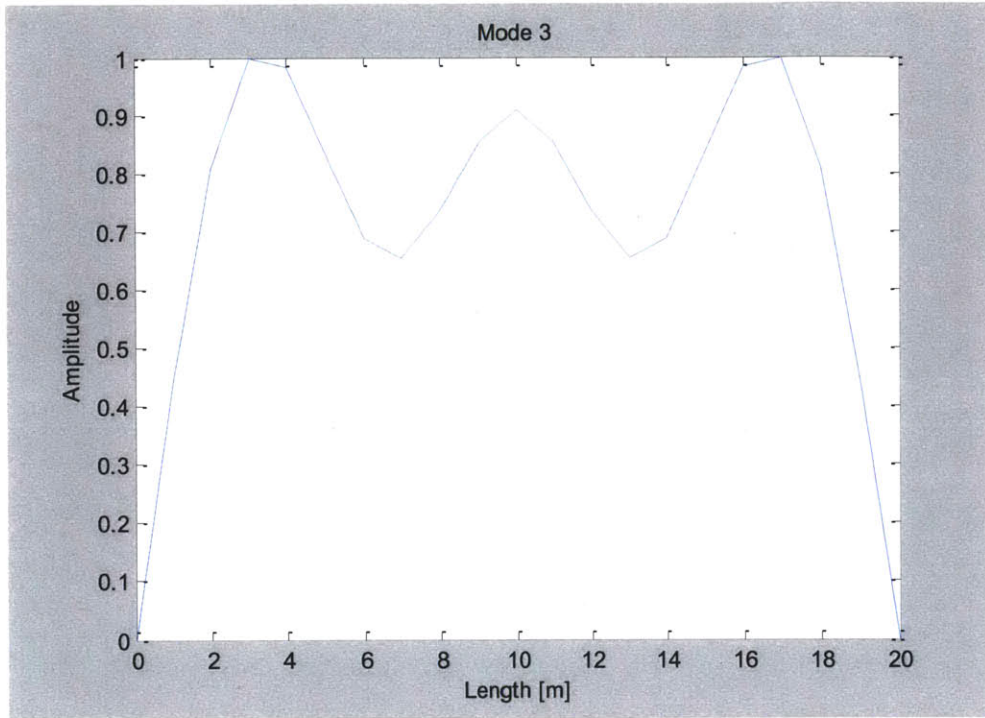


Figure 4-24: Third Mode Shape Obtained (21-Node Model)

4.3.2 Asymmetric Excitation (21-Node Model)

The OpenSees finite element beam model was also then simulated subject to asymmetrical Gaussian white noise ground acceleration. The simulation time step and duration remained the same. Since the input excitation was asymmetric, only asymmetric modes (even modes) could be excited. Figure 4-25 summarizes the values that were obtained for the second mode frequency and figure 4-26 depicts the second mode shape obtained.

Node	3	4	5	6	7	8	9	10	11
$W_2(\text{rad/s})$	92.3628	92.3628	92.3628	92.3628	92.3628	92.3628	92.3628	92.3628	92.3628
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
Node	12	13	14	15	16	17	18	19	20 & 21
$W_2(\text{rad/s})$	3079.389	107.4425	116.2389	116.2389	116.2389	116.2389	116.2389	116.2389	116.2389

Figure 4-25: Extracted Values for the Second Mode Frequency (21-Node Model)

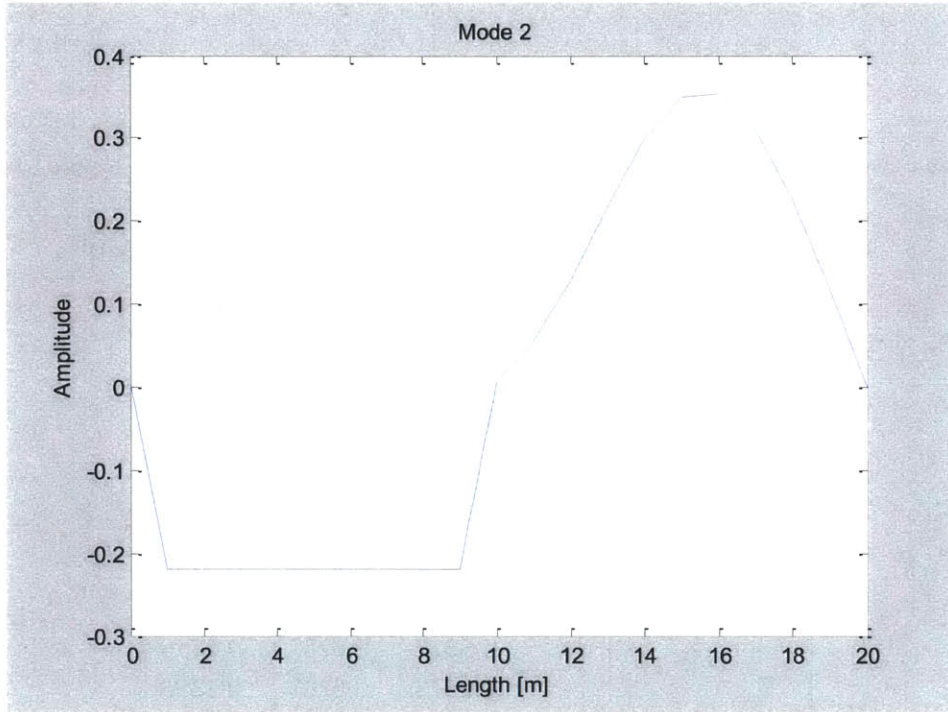


Figure 4-26: Second Mode Shape Obtained (21-Node Model)

4.3.3 Actual and Extracted Modal Properties Comparison (21-Node Model)

Figure 4-27 through 4-31 compare the extracted mode shape and frequency values from the 21-node model using the *MATLAB* TDD implementation to the actual mode shape and frequency values. Figure 4-27 and 4-28 show the modal frequencies and shapes obtained, respectively. The first three modal frequencies were obtained with less than two percent error. Figures 4-29 through 4-31 offer a visual comparison between the actual and extracted mode shapes values. Similar to the 11-node model, the extracted fundamental mode shape is almost the same as the actual fundamental mode shape, whereas the extracted second and third mode shapes are not as accurate. This suggests a poor accuracy in the implementation when obtaining high order mode shapes.

Mode Frequency	Actual	Extracted	%Error
W1 (rad/s)	23.4575	23.2478	0.89
W2 (rad/s)	93.8194	92.3628	1.55
W3 (rad/s)	210.977	214.2566	1.55

Figure 4-27: Mode Frequency Comparison (21-Node Model)

Mode Shapes			
Length (m)	Fundamental	Second	Third
0	0.0000	0.0000	0.0000
1	0.1315	0.6287	0.1261
2	0.2647	0.6287	0.2316
3	0.3992	0.6287	0.2867
4	0.5317	0.6287	0.2816
5	0.6575	0.6287	0.2386
6	0.7711	0.6287	0.1972
7	0.8668	0.6287	0.1872
8	0.9393	0.6287	0.2107
9	0.9846	0.6287	0.2447
10	1.0000	-0.0225	0.2603
11	0.9846	-0.1558	0.2447
12	0.9393	-0.3715	0.2107
13	0.8668	-0.6263	0.1872
14	0.7711	-0.8565	0.1972
15	0.6575	-1.0000	0.2386
16	0.5317	-1.0151	0.2816
17	0.3992	-0.8916	0.2867
18	0.2647	-0.6520	0.2316
19	0.1315	-0.3399	0.1261
20	0.0000	0.0000	0.0000

Figure 4-28: Mode Shapes Extracted (21-Node Model)

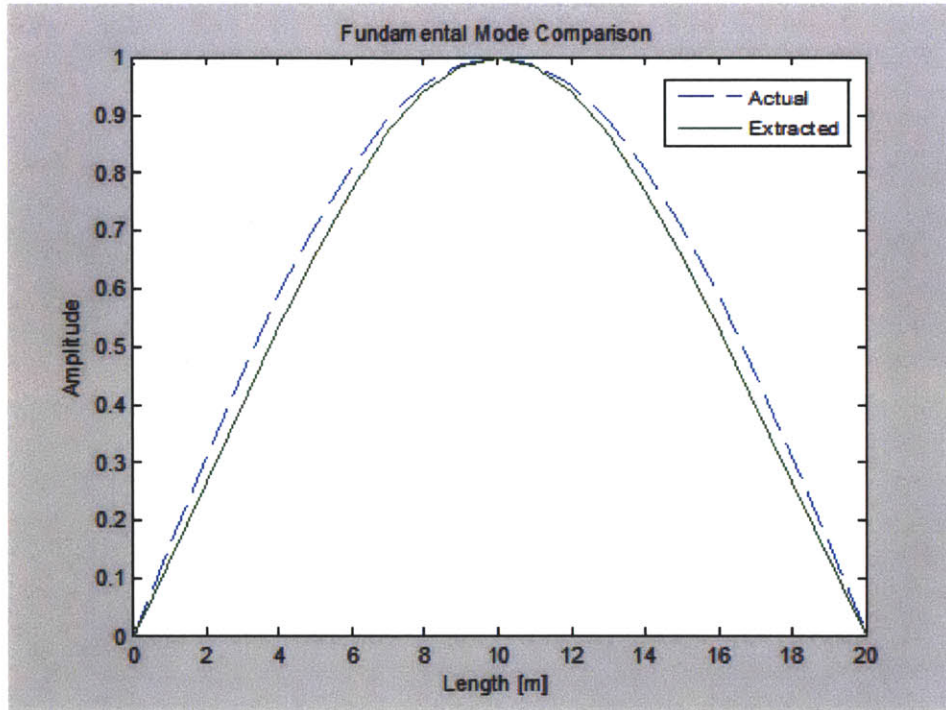


Figure 4-29: Fundamental Mode Comparison (21-Node Model)

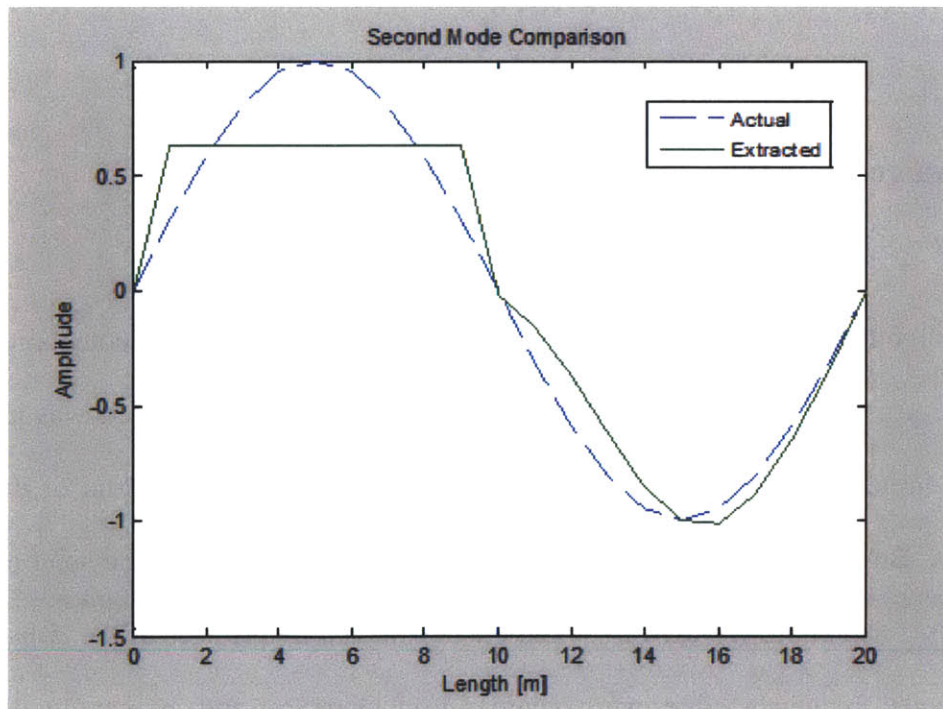


Figure 4-30: Second Mode Comparison (21-Node Model)

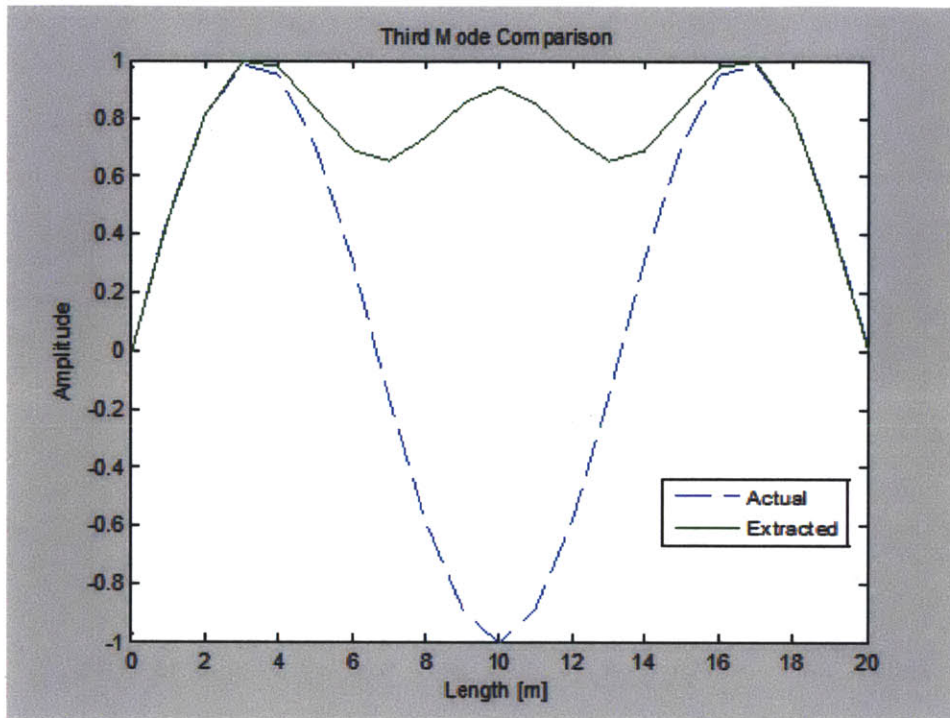


Figure 4-31: Third Mode Comparison (21-Node Model)

4.4 Damage Detection

Although the main objective of this thesis was to use the TTD implementation to extract modal properties, the area of damage detection was briefly explored. Damage was introduced in the 11-node finite element model by way to reducing the moment of inertia of the structural element links. Both the location and severity of the damage varied over the span of the beam. The TDD implementation was used to obtain the new modal frequencies of the *damaged* model. The actual modal frequency values were obtained from OpenSees and compared with the values obtained from the TDD implementation.

Figure 4-32 shows several cases where damaged was introduced into the 11-node model. In each case, the moment of inertia in the specified structural link was reduced by 50 percent. The modal frequency values for the first, second and third modes were obtained from OpenSees using the script in Appendix B-3.

Structural Link	No Damage	Damage Introduced									
		1	2	3	4	5	6	7	8	9	10
W1 (rad/s)	23.4575	23.3817	22.9725	22.3527	21.7886	21.4666	21.4685	21.7907	22.3549	22.9749	23.3842
W2 (rad/s)	93.8194	92.6691	88.2897	86.4383	88.9744	92.8039	92.8054	88.9763	86.4403	88.2913	92.6705
W3 (rad/s)	210.9767	205.7205	196.0547	203.2256	208.9345	198.7208	198.7312	208.9504	203.2424	196.0684	205.7363

Figure 4-32: Actual Modal Frequencies of *Damaged* Models

The *MATLAB* TDD implementation was then applied to each of the *damaged* models and the fundamental mode frequency was extracted. Figure 4-33 shows a comparison of the actual to the extracted fundamental mode frequency for both the *undamaged* and *damaged* 11-nodes models. The TDD implementation obtained the fundamental mode frequency with less than a 6 percent error in every case tried. Furthermore, the TDD implementation was able to distinguish a change in the fundamental frequency when that change was at least 7 percent of the fundamental frequency of the *undamaged* model. In other words, for *damaged* models whose actual fundamental frequency deviated at least 7 percent from the actual *undamaged* fundamental frequency, the TDD implementation was able to detect a change. Otherwise, if the percent change in frequency was less than 7 percent, the extracted fundamental frequency would be the same as that of the *undamaged* model. Lastly, as shown in figure 4-33, the resolution of the TDD implementation when obtaining the fundamental frequency is approximately 1.7 radians per second.

First Mode Frequency [rad/sec]			
Damage Induced	Actual	Extracted	% Error
No	23.4575	23.2478	0.89
Yes	23.3817	23.2478	0.57
Yes	22.9725	22.6195	1.54
Yes	22.3527	22.6195	1.19
Yes	21.7886	22.6195	3.81
Yes	21.4666	22.6195	5.37
Yes	21.1904	20.73450	2.15
Yes	20.9871	20.73450	1.20
Yes	20.3958	20.73450	1.66
Yes	18.6037	18.8496	1.32

Figure 4-33: Actual and Extracted Fundamental Modal Frequencies of *Damaged Models*

Chapter 5

Conclusion

5.1 Summary

The first chapter of this thesis provided a motivation for structural health monitoring. Chapter 2 gives an overview and discusses the various steps involved in creating an SHM scheme. Operational evaluation, data acquisition and normalization, feature extraction and damage model development were some of the facets of SHM that were addressed. The second chapter then focuses on vibration-based SHM and discusses the sources of excitation used to create a dynamic response in a structure, the data acquisition instruments used to record the dynamic response and the various damage detection algorithms used to analyze the dynamic response. The most common damage detection algorithms discussed were based on modal properties, matrix methods and machine learning. Chapter 3 gave an overview of a technique for obtaining the modal properties from a structure's dynamic response called operational modal analysis. Both frequency and time domain operational modal analysis techniques were discussed.

The chapter focused on a particular technique called time domain decomposition (TDD), which uses both frequency and time domain techniques to obtain a structure's modal properties. Using the theoretical implementation of the TDD technique from [15] a *MATLAB* TDD implementation was developed. Chapter 4 presents an overview of the results obtained from the *MATLAB* TDD implementation applied to the dynamic response of two finite element models. The models were created in OpenSees with the first model consisting of a simply-supported beam structure discretized into 11 nodes and 10 elements and the second model consisting of the same simply-supported beam discretized into 21 nodes and 20 elements. The *MATLAB* TDD implementation was able to obtain the first three modal frequencies of both models with less than 6 percent error. The fundamental mode shapes obtained for each model were very close to the actual mode shapes obtained from OpenSees. However, the second and third mode shapes obtained were more inaccurate, which suggests that the current *MATLAB* TDD implementation has limitations in obtaining higher order mode shapes. In the context of damage detection, the *MATLAB* TDD implementation was able to distinguish changes in the fundamental frequency of both finite element models with a resolution of approximately 1.7 radians per second (7.2 percent).

Appendix A

A-1: TDD *MATLAB* Code Implementation

```
% This MATLAB script implements the time domain decomposition technique.  
% The script first reads a given file containing acceleration time history  
% data. It then implements a peak-picking algorithm in order to obtain the  
% modal frequencies from the power spectral plots. Once the modal frequencies  
% are obtained, each acceleration time history signal is filtered using a  
% band-pass filter and the response matrix is built. Lastly, The  
% autocorrelation matrix is found and SVD is performed in order to obtain  
% the scaled modal shapes and frequencies. The SDOF representation is  
% optional and commented out.
```

```
% clears all MATLAB workspace  
clc;  
clear all;  
close all;
```

```
% Opens, reads and arranges time history acceleration data from a text file  
% containing 11 nodes (Output from Opensees)  
S=load('opensees_output.txt');  
a11_rel=S(:,12)';  
a10_rel=S(:,11)';  
a9_rel=S(:,10)';  
a8_rel=S(:,9)';  
a7_rel=S(:,8)';  
a6_rel=S(:,7)';  
a5_rel=S(:,6)';  
a4_rel=S(:,5)';  
a3_rel=S(:,4)';
```

```
% Opens, reads and arranges time history acceleration data form an excel file  
% containing 11 nodes (optional code)  
% [num,txt,row] = xlsread('ss_beam_absolute_n9_standard','H4:H500000');  
% a11_rel=num(100011:110011)';  
% a10_rel=num(90010:100010)';  
% a9_rel=num(80009:90009)';  
% a8_rel=num(70008:80008)';  
% a7_rel=num(60007:70007)';
```

```

% a6_rel=num(50006:60006)';
% a5_rel=num(40005:50005)';
% a4_rel=num(30004:40004)';
% a3_rel=num(20003:30003)';
% data=[a3_rel;a4_rel;a5_rel;a6_rel;a7_rel;a8_rel;a9_rel;a10_rel;a11_rel];

% Accelerometer Time History Data Contained in variable (data)
data=[a3_rel;a4_rel;a5_rel;a6_rel;a7_rel;a8_rel;a9_rel;a10_rel;a11_rel];

fs=1000; %Sampling frequency [Hertz]
Nd = length(data(:,1)); %number of accelerometers
Ns = length(data(1,:)); %number of data points sampled

% Finds the mode frequencies
n=3; % number of modes desired
w =zeros(Nd*n,1);
for i=1:1:Nd
    %outputs the vector of mode frequencies
    w((n*(i-1)+1):(n*i),1) = find_mode_freq(data(i,:), fs, n);
end

fsamp = 2*fs; %sampling frequency
filter_data = zeros(Nd*n,Ns); %Initializes matrices for speed
U = zeros(Nd*n,Nd);
S = zeros(Nd*n,Nd);
V = zeros(Nd*n,Nd);
qdot_dot =zeros(Nd,Ns);
count=0;
% Creates the response matrix Un for each mode
for j=1:1:Nd
    for k=1:1:n
        bandPass = build_band_filter(w(k+count), fsamp);
        filter_data(Nd*(k-1)+j,:) = filter(bandPass, data(j,:));
    end
    count = count+n;
end

count2=1;
for l=1:Nd:Nd*n
    En = filter_data(1:l+Nd-1,:)*filter_data(1:l+Nd-1,:)' ; %Autoregressive
matrix
    [U(1:l+Nd-1,:),S(1:l+Nd-1,:),V(1:l+Nd-1,:)] = svd(En); %Singular Value
Decomposition
    %qdot_dot(count2,:)= (U(1:l+Nd-1,1) '*filter_data(1:l+Nd-1,:))/(U(1:l+Nd-
1,1) '*U(1:l+Nd-1,1));
    count2= count2+1;
end

```

A-2: Modal Frequency Finder Function

```
function [w] = find_mode_freq(data, fs, n)

%
% Function takes in as input a time history vector, the sampling frequency
% and the number of modes of the system. It obtains the power spectrum
% curve from the time history data and finds its peaks. The function then
% outputs the frequency values in rad/sec that correspond to the peaks.
%

N = length(data(1,:));           %Number of sample points in data
DATA = fft(data,N);             %Computes the Fast Fourier Transform
Pyy = DATA.*conj(DATA)/N;      %Computes the power spectral density
Pyy_half = Pyy(1:(floor(N/2)+4)); %Pyy is symmetric, thus take just over
                                %half the signal

f = (fs/N)*(0:(floor(N/2)+3));

%Finds and sorts peaks
[~,locs]=findpeaks(Pyy_half, 'sortstr', 'descend', 'minpeakdistance',10);

%Finds frequencies in radians/second for corresponding peaks
w = zeros(n,1);
for i=1:1:n
    w(i) = f(locs(i))*2*pi;
end
end
```

A-3: Band-Pass Filter Build Function

```
function [ bandPass ] = build_band_filter(w, fs)

%
% Function takes in as input a frequency and signal sampling frequency.
% It proceeds to build a band-pass filter around the given input frequency.
% Frequencies that are more than 3% above or below the band-pass frequency
% will be attenuated by 120 decibels.
%

A_stop1 = 120;                 % Attenuation in the first stopband
F_stop1 = w-.03*w;             % Edge of the stopband
F_pass1 = w-.02*w;             % Edge of the passband
F_pass2 = w+.02*w;             % Closing edge of the passband
F_stop2 = w+.03*w;             % Edge of the second stopband
```

```
A_stop2 = 120;           % Attenuation in the second stopband
A_pass = 1;             % Amount of ripple allowed in the passband

bandPassObj = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2', F_stop1,
                              F_pass1, F_pass2, F_stop2, A_stop1,
                              A_pass, A_stop2, fs);

bandPass = design(bandPassObj, 'butter');

% fvtool(bandPass); % optional for visualization
end
```


Appendix B

B-1: Opensees Simply Supported Beam Model (11 nodes)

```
#####  
## 11 node finite element model of a simply supported beam model with  
## symmetric and asymmetric loading conditions. Software: Opensees v2.3.2  
#####  
  
# Basic dimension definitions  
model basic -ndm 2 -ndf 3  
  
# Node definitions  
node 1 -10 0  
node 2 10 0  
node 3 -8 0  
node 4 -6 0  
node 5 -4 0  
node 6 -2 0  
node 7 0 0  
node 8 2 0  
node 9 4 0  
node 10 6 0  
node 11 8 0  
  
fix 1 1 1 0  
fix 2 0 1 0  
  
# Node mass definitions  
mass 1 0. 178.173 0.  
mass 3 0. 356.347 0.  
mass 4 0. 356.347 0.  
mass 5 0. 356.347 0.  
mass 6 0. 356.347 0.  
mass 7 0. 356.347 0.  
mass 8 0. 356.347 0.  
mass 9 0. 356.347 0.  
mass 10 0. 356.347 0.  
mass 11 0. 356.347 0.  
mass 2 0. 178.173 0.  
  
geomTransf Linear 1  
  
# Element definitions
```

```

element elasticBeamColumn 1 1 3 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 2 3 4 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 3 4 5 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 4 5 6 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 5 6 7 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 6 7 8 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 7 8 9 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 8 9 10 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 9 10 11 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 10 11 2 0.0227 1.999e11 8.056e-4 1

# Rayleigh Damping
rayleigh 1.8644 8.677e-4 0.0 0.0

# Data recording and output
recorder Node -file opensees_output.txt -time -nodeRange 1 11 -dof 2 accel

# Loading excitation and scaling
timeSeries Path 1 -dt 0.005 -filePath opensees_005.txt -factor .04905
pattern UniformExcitation 1 2 -accel 1

# Solver
integrator Newmark 0.5 0.25

# Optional for asymmetric loading
#pattern MultipleSupport 2 {
#groundMotion 1 Plain -accel 1
#imposedMotion 1 2 1
#imposedMotion 3 2 1
#imposedMotion 4 2 1
#imposedMotion 5 2 1
#imposedMotion 6 2 1
#}
#

# General configuration

# Handles boundary conditions
constraints Transformation;
numberer RCM;

# How store and solve the equations in analysis
system BandGeneral;

# Convergence criteria with tolerance, max iterations
test NormUnbalance 1.0e-6 400;

# Solution algorithm
algorithm Newton;

# Define type of analysis
analysis Transient;

# Analysis time duration and step size
set ok [analyze 10000 0.001];

remove recorders

```

B-2: Opensees Simply Supported Beam Model (21 nodes)

```
#####  
## 21 node finite element model of a simply supported beam model with  
## symmetric and asymmetric loading conditions. Software: Opensees v2.3.2  
#####  
  
# Basic dimension definitions  
model basic -ndm 2 -ndf 3  
  
# Node definitions  
node 1 -10 0  
node 2 10 0  
node 3 -9 0  
node 4 -8 0  
node 5 -7 0  
node 6 -6 0  
node 7 -5 0  
node 8 -4 0  
node 9 -3 0  
node 10 -2 0  
node 11 -1 0  
node 12 0 0  
node 13 1 0  
node 14 2 0  
node 15 3 0  
node 16 4 0  
node 17 5 0  
node 18 6 0  
node 19 7 0  
node 20 8 0  
node 21 9 0  
  
fix 1 1 1 0  
fix 2 0 1 0  
  
# Node mass definitions  
mass 1 0. 89.0865 0.  
mass 2 0. 89.0865 0.  
mass 3 0. 178.173 0.  
mass 4 0. 178.173 0.  
mass 5 0. 178.173 0.  
mass 6 0. 178.173 0.  
mass 7 0. 178.173 0.  
mass 8 0. 178.173 0.  
mass 9 0. 178.173 0.  
mass 10 0. 178.173 0.  
mass 11 0. 178.173 0.  
mass 12 0. 178.173 0.
```

```
mass 13 0. 178.173 0.
mass 14 0. 178.173 0.
mass 15 0. 178.173 0.
mass 16 0. 178.173 0.
mass 17 0. 178.173 0.
mass 18 0. 178.173 0.
mass 19 0. 178.173 0.
mass 20 0. 178.173 0.
mass 21 0. 178.173 0.
```

```
geomTransf Linear 1
```

```
# Element definitions
```

```
element elasticBeamColumn 1 1 3 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 2 3 4 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 3 4 5 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 4 5 6 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 5 6 7 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 6 7 8 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 7 8 9 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 8 9 10 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 9 10 11 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 10 11 12 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 11 12 13 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 12 13 14 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 13 14 15 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 14 15 16 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 15 16 17 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 16 17 18 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 17 18 19 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 18 19 20 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 19 20 21 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 20 21 2 0.0227 1.999e11 8.056e-4 1
```

```
# Rayleigh Damping
```

```
rayleigh 1.8644 8.677e-4 0.0 0.0
```

```
# Data recording and output
```

```
recorder Node -file opensees_output.txt -time -nodeRange 1 21 -dof 2 accel
```

```
# Loading excitation and scaling
```

```
timeSeries Path 1 -dt 0.005 -filePath opensees_005.txt -factor .04905
```

```
pattern UniformExcitation 1 2 -accel 1
```

```
# Solver
```

```
integrator Newmark 0.5 0.25
```

```
# Optional for asymmetric loading
```

```
#pattern MultipleSupport 2 {
```

```
#groundMotion 1 Plain -accel 1
```

```
#imposedMotion 1 2 1
```

```
#imposedMotion 3 2 1
```

```
#imposedMotion 4 2 1
```

```
#imposedMotion 5 2 1
```

```
#imposedMotion 6 2 1
```

```
#imposedMotion 7 2 1
```

```
#imposedMotion 8 2 1
```

```

#imposedMotion 9 2 1
#imposedMotion 10 2 1
#imposedMotion 11 2 1
#}
#

#
# General configuration
#

# Handles boundary conditions
constraints Transformation;
numberer RCM;

# How store and solve the equations in analysis
system BandGeneral;

# Convergence criteria with tolerance, max iterations
test NormUnbalance 1.0e-6 400;

# Solution algorithm
algorithm Newton;

# Define type of analysis
analysis Transient;

# Analysis time duration and step size
set ok [analyze 10000 0.001];

remove recorders

```

B-3: Opensees Code for Modal Properties Extraction (11-Nodes)

```

#####
## Script obtains the modal periods of an 11 node finite element model of
## a simply supported beam model. Software: Opensees v2.3.2
#####

# Basic dimension definitions
model basic -ndm 2 -ndf 3

# Node definitions
node 1 -10 0
node 2 10 0
node 3 -8 0
node 4 -6 0
node 5 -4 0
node 6 -2 0
node 7 0 0

```

```

node 8 2 0
node 9 4 0
node 10 6 0
node 11 8 0

fix 1 1 1 0
fix 2 0 1 0

# Node mass definitions
mass 1 0. 178.173 0.
mass 3 0. 356.347 0.
mass 4 0. 356.347 0.
mass 5 0. 356.347 0.
mass 6 0. 356.347 0.
mass 7 0. 356.347 0.
mass 8 0. 356.347 0.
mass 9 0. 356.347 0.
mass 10 0. 356.347 0.
mass 11 0. 356.347 0.
mass 2 0. 178.173 0.

geomTransf Linear 1

# Element definitions
element elasticBeamColumn 1 1 3 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 2 3 4 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 3 4 5 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 4 5 6 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 5 6 7 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 6 7 8 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 7 8 9 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 8 9 10 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 9 10 11 0.0227 1.999e11 8.056e-4 1
element elasticBeamColumn 10 11 2 0.0227 1.999e11 8.056e-4 1

# Rayleigh Damping
rayleigh 1.8644 8.677e-4 0.0 0.0

# Setup number of eigenvalues to obtain
for { set k 1 } { $k <= 4 } { incr k } {
    recorder Node -file [format "modes/mode%i.out" $k] -nodeRange 1 11 -dof 2
    "eigen $k"
}

# General definitions
set lambda [eigen 4];
set omega {}
set f {}
set T {}
set pi 3.141593

foreach lam $lambda {
    lappend omega [expr sqrt($lam)]
    lappend f [expr sqrt($lam)/(2*$pi)]
    lappend T [expr (2*$pi)/sqrt($lam)]
}

```

```
# Write modal periods to a text file
set period "modesperiods.txt"
set Periods [open $period "w"]
foreach t $T {
    puts $Periods " $t"
}
close $Periods
```

```
record
```

Bibliography

- [1] Chase, S. B., and Laman, J. A. (2000). 'Dynamics and Field Testing of Bridges.' *Transportation in The New Millennium: State of the Art and Future Directions, Perspectives from the Transportation Research Board Standing Committees.*
- [2] Sikorsky, C. (1999). 'Development of a Health Monitoring System for Civil Structures Using a Level IV Non-Destructive Damage Evaluation Method.' *Proc., the 2nd International Workshop on Structural Health Monitoring*, 68–81.
- [3] Farrar C.R, Worden K (2007) 'An introduction to structural health monitoring.' *Philosophical Transactions of the Royal Society A*. 365, 303–315.
- [4] Farrar, C.R. and Sohn, H. (2001). 'Condition/Damage Monitoring Methodologies.' Invited Talk, *The Consortium of Organizations for Strong Motion Observation Systems (COSMOS) Workshop, Emeryville, CA, November 14–15, 2001.*
- [5] Salawu, O.S. and Williams, C. (1995). 'Review of Full-Scale Dynamic Testing of Bridge Structures.' *Engineering Structures*, 17(2), 113–121.
- [6] Green, M.F. (1995). 'Modal Test Methods For Bridges: A Review.' *Proceedings of the 13th International Modal Analysis Conference, Nashville, Tennessee*, 1, 552–558.
- [7] Bicanic, N and Chen, H. (1997). 'Damage Identification in Framed Structures Using Natural Frequencies.' *International Journal for Numerical Methods in Engineering*, 40, 4451–4468.
- [8] Ray, L.R. and Tian, L. (1999). 'Damage Detection in Smart Structures Through Sensitivity Enhancing Feedback Control.' *J. Sound and Vibration*, 227(5), 987–1002.
- [9] Law, S.S., Shi, Z.Y. and Zhang, L.M. (1998). 'Structural Damage Detection From Incomplete and Noisy Modal Test Data.' *J. Engineering Mechanics*, 124(11), 1280–1288.
- [10] Stubbs, N., Park, S., Sikorsky, C. and Choi, S. (2000). 'A Global Non-destructive Damage Assessment Methodology for Civil Engineering Structures.' *Intl. J. Systems Science*, 31(11), 1361–1373.
- [11] Escobar, J.A., Sosa, J.J. and Gomez, R. (2001). 'Damage Detection in Framed Buildings.' *Canadian Journal of Civil Engineering*, 28(1), 35–47.

- [12] Adeli, H. and Hung, S. (1995). *Machine Learning – Neural Networks, Genetic Algorithms, and Fuzzy Systems*. John Wiley & Sons.
- [13] Hung, S-L. and Kao, C.Y. (2002). ‘Structural Damage Detection Using the Optimal Weights of the Approximating Artificial Neural Networks.’ *Earthquake Engineering and Structural Dynamics*, 31, 217–234.
- [14] Hao, H. and Xia, Y. (2002). ‘Vibration-based Damage Detection of Structures by Genetic Algorithm.’ *J. Computing in Civil Engineering*, 16(3), 222–229.
- [15] Karbhari, Vistasp M.; Ansari, Farhad (2009). *Structural Health Monitoring of Civil Infrastructure Systems*. Woodhead Publishing.
- [16] Maia, N. M. M., and Montalvão e Silva, J. M. (1997). *Theoretical and experimental modal analysis*, Research Studies Press; Wiley, Hertfordshire, England / New York.
- [17] Brincker, R., Zhang, L., and Anderson, P. (2000). ‘Output-Only Modal Analysis by Frequency Domain Decomposition.’ *International Seminar on Modal Analysis (ISMA25)*, Katholieke Universiteit Leuven, Belgium.
- [18] James III, G. H., Carne, T. G., and Lauffer, J. P. (1993). ‘*The Natural Excitation Technique (NExT) for Modal Parameter Extraction From Operating Wind Turbines.*’ Sandia National Laboratories, Albuquerque, New Mexico, USA.