Investigating Various Modal Analysis Extraction Techniques to Estimate Damping Ratio

Angel Moises Iglesias

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

Master Degree
in
Mechanical Engineering

Alfred L. Wicks
Ricardo A. Burdisso
Larry D. Mitchell

June 29, 2000
Blacksburg, Virginia

Keywords: Vibrations, Damping Ratio, Dynamic Properties

Investigating Various Modal Analysis Extraction Techniques to Estimate Damping Ratio

Angel Moises Iglesias

Abstract

Many researchers have devoted their work to the development of modal analysis extraction techniques in order to obtain more reliable identification of the modal parameters. Also, as a consequence of all this work, there are some other works devoted to the evaluation and comparison of these methods in order to find which one is the most reliable method with respect to certain characteristics.

In this thesis the Rational Fraction Polynomial (RFP) Method, the Prony or Complex Exponential Method (CEM), the Ibrahim Time Domain (ITD) Method, and Hilbert Envelope Method are used to evaluate how the accuracy of the damping ratio is affected with respect to various parameters and conditions. The investigation focuses in the estimation of damping ratio because among the modal parameters, it is the most difficult to model. Each method is evaluated individually in order to understand how the damping ratio estimation is affected with respect to each method when the characteristics of the FRF are changed. Also, they are compared to show that, in general, the Rational Fraction Polynomial Method is a more reliable method than the other methods. To investigate this, a simulated analytical data and an experimental data are processed to estimate the modal parameters, but focusing in the damping ratio. For the simulated analytical data the damping ratio's percent of error were calculated. The highest damping ratio's percent of error of the RFP was 0.0073501%. In the other hand, for the CEM, ITD, and Hilbert Envelope Method their highest damping ratio's percent of error were 83.02%, 99.82%, and 4.077%, respectively.

Acknowledgment

I want to express my gratitude to a lot of people that during the past two years have supported me in some way or another. To my major professor, Dr. Alfred L. Wicks, a special thanks for giving me the opportunity to work with him and for his patience, tolerance, understanding, and encouragement. He was more than a professor to me; he was also a friend. It has been a great pleasure working with him. I am very thankful to my committee, Dr. Larry D. Mitchell, Dr. Mehdi Ahmadian, and Dr. Ricardo Burdisso for their patience, and advices. I really appreciate their time and effort in been in part of my committee. I want to express my gratitude to Sean Fahey for helping me in my research as well for his advices. Another thanks are extended to Cathy Hill for her support and help in many ways. I also want to thank to all those that helped me through the research.

A special thanks goes to my friends Dwight Smith, Bernardo Carnicero, Edwin Ayala, Edgardo Reyes, and Cristina Sanchez. They were encouraging me during my whole research. I want to thank the Thermal Radiation Group of Dr. Mahan for their support and friendship. Also, I am grateful to the Hispanic Community for making me feel like home.

My parents, Jose Manuel Iglesias and Magda Iglesias, receive a very special thanks for been my support and guide in the most difficult times of my life. They were encouraging and supporting me all the time. Without them I would not be able to be what I am.

Also, I want to thank Victor A. Ramos, my godfather, for been such a good godfather by giving me his example of a Christian Catholic friend. He encouraged and supported me in the good and bad days.

I am very grateful with my beloved future wife, Frances Bernier, for encouraging and giving me strength to do what I had to do. She brought me more love and strength to my life.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction and Literature Review

The dynamic response of structures and the transmission of vibrations to the surroundings are critically determined by the damping mechanisms, and its value is very important for the design and analysis of vibrating structures. When the structure is modeled, the stiffness and mass distributions are quite well determined, but there is great uncertainty regarding the energy dissipating mechanism provided by the damping of the structure because it is the least well understood. But, in order to validate these models, the damping must be estimated by experimental modal analysis.

The experimental modal analysis is the analysis of the structural dynamic properties in terms of its modal parameters. These modal parameters can be identified from the measured frequency response function (FRF). This process of identifying parameters is often referred to as curve fitting or parameters estimation. The modal parameters are the measured natural frequencies, damping ratios and mode shapes. But, many parameters and conditions, depending on which modal parameter extraction technique is used, affect the accuracy of estimating damping ratio. Part of this thesis' work is to investigate the capabilities of the modal parameter extraction techniques to do curve fitting.

The goal of this thesis is to investigate how the accuracy of estimating damping ratio changes with respect to: the modal parameter extraction technique, different damping ratios, truncation in the FRF, close modes, separated modes, and two approaches: multiple degree of freedom and single mode approach. Four modal parameter extraction techniques were selected to perform this investigation; these are the Prony or Complex Exponential Method (CEM), the Ibrahim Time Domain (ITD) Method, the Rational Fraction Polynomial Method, and the Hilbert Envelope Method. Regarding the other parameters and conditions, these are selected because these are the characteristics that the real data presents, also the accuracy of estimating damping ratio can be affected depending of the method used and the characteristics of the FRF.

In order to accomplish these goals, a simulated analytical and experimental data were analyzed by estimating the modal parameters. The simulated analytical data is very useful because the exact values are known and the characteristics of the FRF can be varied in order to observe how the damping ratio accuracy is affected. The experimental data was measured at various locations of a structure and then analyzed in the same manner as the simulated analytical data with the difference that for the experimental data the exact values are not know and the error cannot be calculated. For all the methods and cases, the data is regenerated using the estimated modal parameters in order to calculate the residual, or difference, and the standard deviation between the original and regenerated FRF. In order to test if the curve fit behaves in the same manner in experimental data as for the simulate data, the residual and standard deviations are compared between the type of data with respect to the characteristics of FRF and the modal analysis parameter extraction technique.

Literature Review

A lot of work has being devoted to the development and improvement of the modal analysis extraction techniques. As a consequence of all this work, some other work has being performed in regard to compare these techniques and investigate their reliability. Svend Gade and Henrik Herlufsen [1] compared the Digital Filter (DF) techniques vs. the Discrete/Fast Fourier Transform (DFT/FFT) techniques for damping measurements by vibration decay measurements or bandwidth determination of measured modal resonances. Also, these methods are summarized with respect to their advantages and disadvantages. For the DF analysis the damping is estimated from the decay of the free vibration response due to an impact excitation and they address that the advantage of this method is that it is very fast and doesn't have limitations in dealing with very light damped systems. But, due to the poor resolution of DF analyzer it is not well suited for bandwidth determination of measured modal resonances. For the FFT techniques the damping was measured using free vibration decay, curve fit of frequency response function measured using impact excitation and random excitation with shaker, and decay of impulse response function using pseudo random excitation with a shaker. The vibration decay method doesn't have any limitation in regard to low damping, but the limit for the estimation of high damping values comes from the limited transfer rate of spectra, which depends of the measuring system. Also, it requires that the resonances be well separated in the analysis. The curve fit of the frequency response function measured using impact excitation is done in a single-degree-of-freedom approach and uses an exponential window in order to decrease the leakage error. This method has the advantage over the decay methods in regard that high damped systems with high coupling between the modes can be analyzed. The same method is used for random excitation, but due to the low resolution it uses zoom measurements with sufficient resolution to eliminate leakage. The disadvantage of the zoom technique is that it takes to much time for the analysis. For the last FFT techniques the frequency response function is estimated using shaker with a pseudo random force signal, then the decay envelope of the impulse response function for each mode is calculated by isolating the different resonances in the frequency response function. This method is faster than the previous one. They say that in general, FFT analysis is better for heavily damped structures, but it is advantageous to use DF analysis when dealing with lightly damped structures.

M. Imregun compared two different single-degree-of-freedom (SDOF) modal analysis techniques and global multi-degree-of freedom (MDOF) method applied to frequency response function measurements taken on a lightly damped linear structure. For the SDOF, the circle-fit and the line-fit were used to identify the modal properties, which gave very similar results for most of the cases. But, sometimes there was not possible to fit a reliable circle fit of the FRF data. Also, the weak and coupled modes were among the most difficult to analyze. He addresses that the circle-fit method gives reliable results when there are enough data points around resonance and that damping is not too low. In other hand, this method should not be used when the data contains noise around resonance. He found that the global identification method produces a consistent set of modal properties and is much faster than the SDOF approach. He also found that a reliable batch processing of measured FRF data is possible as long as simple, but rigorous, checks are made to ensure the quality of global parameters identification.

In another paper [3], S O'F Fahey and J. Pratt explain how to fit experimental data using single-degree-of-freedom and multiple-degree-of-freedom techniques. The SDOF techniques are the Half-Power and Finite Difference method. They say that these two techniques are attractive when performing quick field analysis or to provide initial estimates for more complex MDOF techniques. For the MDOF techniques they explained the simultaneous frequency-domain method and the rational polynomial method. Finally, they addressed the topic about refitting data to obtain global modal parameters. But in another article [4], they compared different time-domain modal-estimation techniques. These techniques are the Complex Exponential Algorithm, Pisarenko's Harmonic Decomposition, Ibrahim's Time Domain method, and the Eigen-system Realization Algorithm. They developed a numerical example in order to compare and contrast them. They say that there is a difference between the frequency-domain techniques and the time-domain techniques with respect to the system-damping ratio. This difference is that the time-domain techniques generally work better than the frequency-domain techniques when the system damping is less than 0.5 percent. But, the frequency-domain techniques, generally gives more reasonable results when the damping is greater than 4.0 percent.

Nuno Manuel Mendes Maia [5] described and compared some SDOF modal analysis methods. He gave a brief review of the Peak Amplitude, the Quadrature Response, the Maximum Quadrature Component, the Kennedy-Pancu, the Circle-Fittin, the Inverse and Dobson methods. He shows that the Dobson's method is a more refined and more powerful version than the Inverse method. Also, he demonstrates that for practical use, Dobson's method gives better results than the Inverse method and it works better than the Circle Fitting method.

Gilles Collot [6] compares the Half Power Frequency Domain Method, the Hilbert Transform Method and the Half Power Frequency Domain Method based on zoom measurements. There was concluded that third method is time consuming and that should be used when it is absolutely necessary to improve the frequency resolution. For light damping the Hilbert Transform gives better results than the Half Power Frequency Domain Method. Also, there was shown that the first method gives good and fast loss factor calculation when there are good coherence and small frequency resolution.

S. Gade, K. Zaveri, H. Konstantin-Hansen and H. Herlufsen [7] compared and demonstrated a resonant and a non-resonant method to measure damping of visco-elastic materials. There was found that the advantage of the non-resonant method over the resonant method is that it calculates the damping as a continuous function of frequency.

CHAPTER 2

Theory

This chapter develops the theory used to develop the data processing MatLab codes (see Appendix A) described in section 2.4. First, the necessary assumptions are established in order to develop of the next sections, which use this information to establish the analytical equations. Then, the frequency response function (FRF) estimation is explained. After this section, the modal parameters extraction techniques are developed having in mind the assumptions and that these techniques are applied to the measured FRF.

2.1 Assumption

I order to develop the theory applied to the various modal parameters extraction techniques and the frequency response function (FRF) estimation, four basic assumptions have to established about structure:

1. It is a linear system where its dynamic behavior can be described by a second-order differential equation.
2. It is time invariant.
3. Obeys Maxwell's reciprocity theorem.
4. It is underdamped.

2.2 Frequency Response Function Estimation

Assume the following form of the general input/output model.

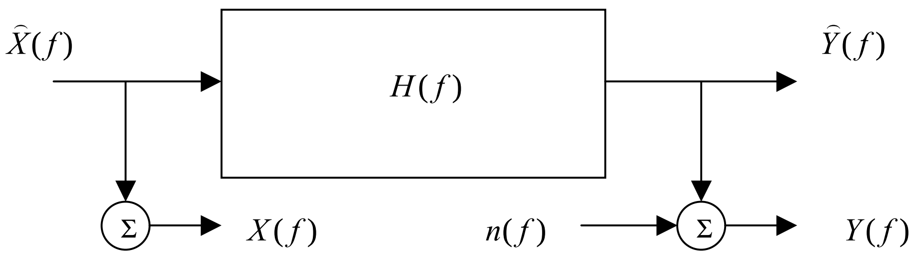


Figure 2.2.1: General input/output model

Then,

$$X(f) = \widehat{X}(f)$$
$$Y(f) = \widehat{Y}(f) + n(f) \tag{2.2.1}$$

where *X(f)* is the measured input, which is equal to the correlated or ideal input, and *Y(f)* is the measured output, which is the sum of the correlated or ideal output plus the uncorrelated output.

The FRF is:

$$H = \frac{\sum_{k=1}^{K} X_k^* Y_k}{\sum_{k=1}^{K} X_k^* X_k} = \frac{\sum_{k=1}^{K} G_{xy}}{\sum_{k=1}^{K} G_{xx}} = \frac{cross - spectrum}{input - autospectrum} = H_1 \tag{2.2.2}$$

where the symbol * means complex conjugate, and H_1 is a complex valued function. The magnitude of the FRF is commonly referred to as gain, and the phase angle is the angle between the output relative to the input, which is obtained from the cross-spectrum in the numerator of the estimator.

The coherence function is:

$$\gamma_{xy}^2 = \frac{\sum_{k=1}^{K} G_{xy} \sum_{k=1}^{K} G_{yx}}{\sum_{k=1}^{K} G_{xx} \sum_{k=1}^{K} G_{yy}} \tag{2.2.3}$$

The coherence function has a value between 0 and 1. When the coherence γ^2 = 1, implies that G_{nn} = 0, which states that there is no uncorrelated content on the output measurement. In the other hand when coherence γ^2 = 0, implies that G_{nn} = G_{yy} and this suggests that the output measurement is composed entirely of uncorrelated content.

A broader explanation regarding the FRF estimation can be found in section 2.3 of reference [9].

2.3 Modal Parameters Extraction Techniques

2.3.1 The Complex-Exponential Method

A complete development of the this method is found in pages 189 to 192 of reference [9].

In the frequency domain, the frequency response function (FRF) in terms of receptance H_{jk} (displacement at point j due to a force at point k) for a linear, viscously damped system with N degree of freedom (DOF) can be given by:

$$H_{jk}(\omega) = \sum_{r=1}^{N} \left(\frac{{}_rA_{jk}}{\omega_r\xi_r + i\left(\omega - \omega_r\sqrt{\left(1-\xi_r^2\right)}\right)} + \frac{{}_rA_{rj}^*}{\omega_r\xi_r + i\left(\omega + \omega_r\sqrt{\left(1-\xi_r^2\right)}\right)} \right) \qquad (2.3.1.1)$$

where ω_r is the natural frequency, ξ_r is the damping ratio and ${}_rA_{jk}$ is the residue corresponding to each mode r; * denotes complex conjugate. Another way of writing equation (2.3.1.1) is

$$H_{jk}(\omega) = \sum_{r=1}^{2N} \frac{{}_rA_{jk}}{\omega_r\xi_r + i(\omega - \omega_r')} \qquad (2.3.1.2)$$

where

$$\begin{aligned} \omega_r' &= \omega_r\sqrt{\left(1-\xi_r^2\right)} \\ \omega_{r+N}' &= -\omega_r' \\ {}_{r+N}A_{jk} &= {}_rA_{jk}^* \end{aligned} \qquad (2.3.1.3)$$

and ω_r is the natural frequency, ξ_r is the damping ratio, and A is the residue.

The CEM works with the corresponding impulse response function (IRF), obtained by an inverse Fourier transform:

$$h_{jk}(t) = \sum_{r=1}^{2N} {}_r A_{jk} e^{s_r t} \tag{2.3.1.4}$$

or,

$$h(t) = \sum_{r=1}^{2N} A'_r e^{s_r t} \tag{2.3.1.5}$$

where $s_r = -\omega_r \xi_r + i\omega'_r$ and the properties in equation (2.3.1.3) hold. The time response $h(t)$ (real-valued) at series of L equally spaced time intervals Δt, is

$$\begin{aligned}
h_0 &= h(0) = \sum_{r=1}^{2N} A'_r \\
h_1 &= h(\Delta t) = \sum_{r=1}^{2N} A'_r e^{s_r(\Delta t)} \\
h_2 &= h(2 \cdot \Delta t) = \sum_{r=1}^{2N} A'_r e^{s_r(2 \cdot \Delta t)} \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
h_L &= h(L \cdot \Delta t) = \sum_{r=1}^{2N} A'_r e^{s_r(L \cdot \Delta t)}
\end{aligned} \tag{2.3.1.6}$$

or,

$$
\begin{aligned}
h_0 &= \sum_{r=1}^{2N} A'_r \\
h_1 &= \sum_{r=1}^{2N} A'_r V_r \\
h_2 &= \sum_{r=1}^{2N} A'_r V_r^2 \\
\vdots \quad & \quad \vdots \qquad \vdots \\
h_L &= \sum_{r=1}^{2N} A'_r V_r^L
\end{aligned}
\tag{2.3.1.7}
$$

with

$$V_r = e^{s_r \Delta t} \tag{2.3.1.8}$$

We know the values of h_i, but we do not know A'_r, V_r. The roots s_r for a underdamped system always occur in complex conjugate pairs, so do the modified variable V_r. There always exists a polynomial in V_r of order L with real coefficients β (called the autoregressive coefficients) such that the following relation is verified:

$$\beta_0 + \beta_1 V_r + \beta_2 V_r^2 + \cdots + \beta_L V_r^L = 0 \tag{2.3.1.9}$$

In order to calculate the coefficients β_j to evaluate V_r, multiply both sides of equation (2.3.1.7) by β_0 to β_L and sum the result. This procedure gives:

$$\sum_{j=0}^{L} \beta_j h_j = \sum_{j=0}^{2N} \left(\beta_j \sum_{r=1}^{2N} A'_r V_r^j \right) = \sum_{r=1}^{2N} \left(A'_r \sum_{j=0}^{L} \beta_j V_r^j \right) \tag{2.3.1.10}$$

The inner summation in the right side of equation (2.3.1.10) is exactly the polynomial in equation (2.3.1.9). Therefore, that summation is going to be equal to zero for each V_r, it follows that

$$\sum_{j=0}^{L} \beta_j h_j = 0 \tag{2.3.1.11}$$

From equation (2.3.1.11) we can calculate the coefficients β_j (h_j is measured). These coefficients are used to calculate the roots of equation (2.3.1.9), V_r, and are calculated as follow: we make $M = L/2$, and n = DOF + 1. There will be n sets of data points h_j, each set shifted one time interval, and β_L is assumed equal to 1. This gives:

$$\begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_{n-1} \\ h_1 & h_2 & h_3 & \cdots & h_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{M-1} & h_M & h_{M+1} & \cdots & h_{n+M-2} \end{bmatrix} \begin{Bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-1} \end{Bmatrix} = - \begin{Bmatrix} h_n \\ h_{n+1} \\ \vdots \\ h_{n+M-1} \end{Bmatrix} \tag{2.3.1.12}$$

or,

$$\underset{M\times n}{[h]}\,\underset{n\times 1}{\{\beta\}} = \underset{M\times 1}{\{h'\}} \tag{2.3.1.13}$$

From this equation it is possible to calculate $\{\beta\}$, as $[h]$ and $\{h'\}$ are known matrices. This can done using pseudo-inverse technique, multiply by $[h]^{\mathrm{T}}$ (transpose), and then solve for $\{\beta\}$. The result is

$$\{\beta\} = \left([h]^T [h]\right)^{-1} \left([h]^T \{h'\}\right) \tag{2.3.1.14}$$

After calculating $\{\beta\}$, it is used to calculate the roots V_r. In order to calculate the natural frequencies, and damping ratios, equation (2.3.1.8) is used, as follows:

$$R_r = \ln(V_r) = s_r \cdot \Delta t$$
$$f_r = \frac{|R_r|}{2\pi\Delta t} \tag{2.3.1.15}$$
$$\xi_r = \sqrt{\frac{1}{1+\left(\frac{\mathrm{Im}\,ag(R_r)}{\mathrm{Re}\,al(R_r)}\right)^2}}$$

With the values of V_r, we can calculate the residues A'_r if equation (2.3.1.7) is written as:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ V_1 & V_2 & \cdots & V_{2N} \\ V_1^2 & V_2^2 & \cdots & V_{2N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ V_1^{2N-1} & V_2^{2N-1} & \cdots & V_{2N}^{2N-1} \end{bmatrix} \begin{Bmatrix} A'_1 \\ A'_2 \\ A'_3 \\ \vdots \\ A'_{2N} \end{Bmatrix} = -\begin{Bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_{2N} \end{Bmatrix} \tag{2.3.1.16}$$

2.3.2 The Hilbert-Envelope Method

The shape of a signal that contains a rapidly oscillating component that varies slowly with time is called “envelope”. With the use of the Hilbert transform, the rapid oscillations can be removed from the signal to produce the representation to the envelope.

The Hilbert transform to *x(t)* is:

$$X_{Hi}(t) = \frac{-1}{\pi t} * x(t) = f(t) * x(t) = F^{-1}\{F(j\omega)X(j\omega)\} \tag{2.3.2.1}$$

The Fourier transform of $(-\pi t)^{-1}$ is i sgn ω_r, which is $+i$ for positive ω and $-i$ for negative ω. The Hilbert transformation is equivalent to a filtering, in which the amplitudes of the spectral components are left unchanged, but their phases are altered by $\pi/2$, positively or negative according to the sign of ω.

Also, the Hilbert transforms of even functions are odd and those of odd functions are even. The cosine component transforms into negative sine components and sine components transform into cosine components.

In order to obtain a more deep explanation regarding the Hilbert transform, it can be found in pages 266 to 272 of reference [11].

The impulse response function of a single-degree-of-freedom system is an exponential damped sinusoid. The Hilbert transform is used to calculate a new time signal from the original signal. Both signals are combined to form the analytical signal as follows,

$$\overline{x(t)} = x(t) - iX_{Hi}(t) \tag{2.3.2.2}$$

The magnitude of the analytic signal is the envelope of the original time signal. When the envelope is plotted in a dB scale, the graph is a line. Then, the slope of the line is related to the damping ratio as will be shown next.

The impulse response function of a single-degree-of-freedom system can be described with the following equation:

$$x(t) = Ae^{-\xi\omega_n t} \sin\left(\omega_n \left(\sqrt{1\text{-}\xi^2}\right)t\right) \tag{2.3.2.3}$$

where ω_n is the natural frequency, ξ is the damping ratio, and A is the residue. The Hilbert transform (from equation 2.3.2.1) of this signal is,

$$X_{Hi}(t) = Ae^{-\xi\omega_n t} \cos\left(\omega_n \left(\sqrt{1\text{-}\xi^2}\right)t\right) \tag{2.3.2.4}$$

The analytic signal is,

$$\overline{x(t)} = Ae^{-\xi\omega_n t}\left(\sin\left(\omega_n\left(\sqrt{1\text{-}\xi^2}\right)t\right) + i\ \cos\left(\omega_n\left(\sqrt{1\text{-}\xi^2}\right)t\right)\right) \tag{2.3.2.5}$$

The magnitude of the analytic signal eliminates the oscillatory component, and gives the envelope as follow,

$$\left|\overline{x(t)}\right| = \sqrt{\left(Ae^{-\xi\omega_n t}\right)^2\left(\sin^2\left(\omega_n\left(\sqrt{1\text{-}\xi^2}\right)t\right) + \cos^2\left(\omega_n\left(\sqrt{1\text{-}\xi^2}\right)t\right)\right)} = Ae^{-\xi\omega_n t} \tag{2.3.2.6}$$

Taking the natural logarithm of each side yields,

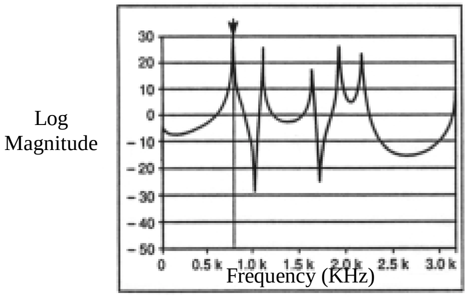
$$\ln\left|\overline{x(t)}\right| = \ln\left(Ae^{-\xi\omega_n t}\right) = \ln(A) - \left(\xi\omega_n\right)t \tag{2.3.2.7}$$

This is the equation of a straight line. If the slope of the line is calculated, we can estimate the damping ratio as follow,

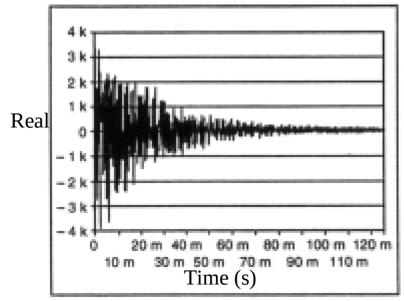
$$\xi = \frac{-\,slope}{\omega_n} \tag{2.3.2.8}$$

Reference [10] gives a brief explanation of how to apply this method to a MDOF FRF, which is explained next.

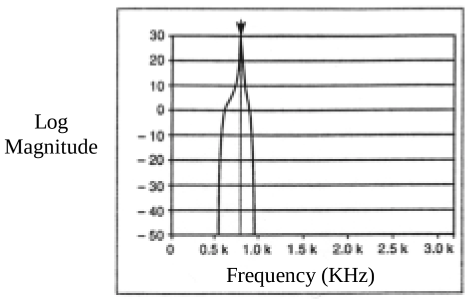
This is a single-degree-of-freedom method. In order to apply it to a multiple-degree-of-freedom (MDOF) FRF the procedure showed in Figure (2.3.2.1) can be used.



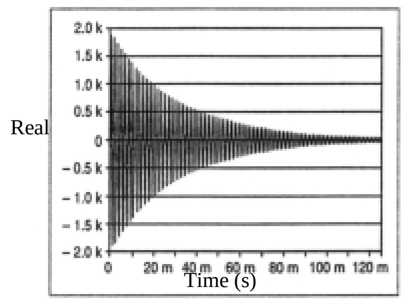
(a) MDOF FRF



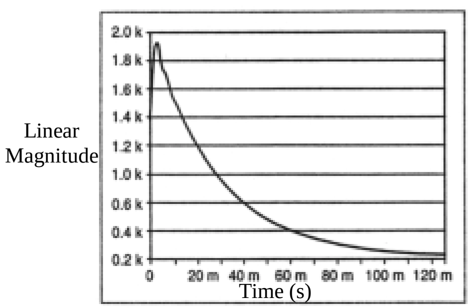
(b) MDOF IRF



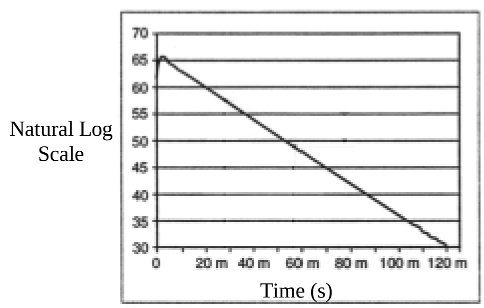
(c) Isolated single-mode



(d) IRF of the isolated single-mode



(e) Envelope of the single-mode's IRF



(f) Envelope in natural log scale

Figure 2.3.2.1: Procedure to apply the Hilbert envelope method to a multiple-degree-of-freedom FRF.

Figure 2.3.2.1(a) shows a frequency response function, and Figure 2.3.2.1(b) shows the corresponding impulse response function. But, this cannot be used to calculate the damping because it contains five exponential damped sinusoids (one for each resonance) superimposed. The damping at each resonance frequency can be determined if we isolate each natural frequency and calculate the impulse response function that corresponds to each one.

Figure 2.3.2.1(c) shows a single resonance that has been isolated from Figure 2.3.2.1(a). Its corresponding IRF is shown in Figure 2.3.2.1(d), which is the Inverse Fourier Transform of the FRF. Figure 2.3.2.1(e) shows the magnitude of the analytic signal of the impulse response function on a linear amplitude scale. Then, using a dB scale, the envelope is a straight line (Figure 2.3.2.1(f)). The Linear Least Square Method can be used to calculate the slope of the line in order to estimate the damping ratio (ξ).

After estimating the damping ratios of each mode, the eigenvalues can be calculated. Then, equation (2.3.1.17), from previous section, can used to estimate the residues.

2.3.3 The Ibrahim Time Domain method

Reference [9] at pages 200 to 202 and reference [12] gives a complete derivation of this method. This method uses the free response of the structure under test. During its free response, the system is assumed to be described by the following equation:

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\}=0 \tag{2.3.3.1}$$

As in the Complex Exponential Method, it is assumed that the solution of this equation is:

$$\{x\}=\{pe^{st}\} \tag{2.3.3.2}$$

whence

$$\left[s^2[M]+s[C]+[K]\right]p=0 \tag{2.3.3.3}$$

For an underdamped structure, the roots s_r of equation (2.3.3.3) are complex and occur in conjugate pairs as,

$$s_r = a_r \pm ib_r = -\xi_r\omega_r \pm i\omega_r\sqrt{1-\xi_{r2}^2} \tag{2.3.3.4}$$

where ξ_r is the damping ratio, and ω_r is the natural frequency in radians per second that correspond to mode r.

The response of a N degree of freedom (DOF) at point i and time t_j is expressed as a summation of the individual responses of each mode:

$$x_i(t_j) = \sum_{r=1}^{2N} p_{ir} e^{s_r t_j} \tag{2.3.3.5}$$

where p_{ir} is the i^{th} component of the eigenvector $\{p_r\}$. The response measured at L instances of time can be expressed in matrix form as

$$\begin{bmatrix} x(t_1) & x(t_2) & \cdots & x(t_L) \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & \cdots & p_{2N} \end{bmatrix} \begin{bmatrix} e^{s_1 t_1} & e^{s_1 t_2} & \cdots & e^{s_1 t_L} \\ e^{s_2 t_1} & e^{s_2 t_2} & \cdots & e^{s_2 t_L} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_{2N} t_1} & e^{s_{2N} t_2} & \cdots & e^{s_{2N} t_L} \end{bmatrix} \tag{2.3.3.6}$$

or,

$$\underset{1\times L}{[X_0]} = \underset{1\times 2N}{[P]}\ \underset{2N\times L}{[\Lambda]} \tag{2.3.3.7}$$

If we consider a response that is shifted one interval Δt with respect to the first, it follows that

$$
\begin{aligned}
&\left[x(t_1+\varDelta t) \quad \cdots \quad x(t_L+\varDelta t)\right] \\
&=\left[p_1 \quad p_2 \quad \cdots \quad p_{2N}\right]\begin{bmatrix} e^{s_1(t_1+\varDelta t)} & e^{s_1(t_2+\varDelta t)} & \cdots & e^{s_1(t_L+\varDelta t)} \\ e^{s_2(t_1+\varDelta t)} & e^{s_2(t_2+\varDelta t)} & \cdots & e^{s_2(t_L+\varDelta t)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_{2N}(t_1+\varDelta t)} & e^{s_{2N}(t_2+\varDelta t)} & \cdots & e^{s_{2N}(t_L+\varDelta t)} \end{bmatrix} \\
&=\left[p_1 \quad p_2 \quad \cdots \quad p_{2N}\right]\begin{bmatrix} e^{s_1\varDelta t} & & & 0 \\ & e^{s_2\varDelta t} & & \\ & & \ddots & \\ 0 & & & e^{s_{2N}\varDelta t} \end{bmatrix}\begin{bmatrix} e^{s_1t_1} & e^{s_1t_2} & \cdots & e^{s_1t_L} \\ e^{s_2t_1} & e^{s_2t_2} & \cdots & e^{s_2t_L} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_{2N}t_1} & e^{s_{2N}t_2} & \cdots & e^{s_{2N}t_L} \end{bmatrix}
\end{aligned}
\tag{2.3.3.8}
$$

or,

$$
\underset{1\times L}{[X_1]} = \underset{1\times 2N}{[P]}\ \underset{2N\times 2N}{[\hat{\varLambda}]}\ \underset{2N\times L}{[\varLambda]} \tag{2.3.3.9}
$$

Consider another response that is shifted two intervals with respect to (2.3.3.5), it is

$$
\begin{aligned}
&\left[x(t_1+2\varDelta t) \quad \cdots \quad x(t_L+2\varDelta t)\right] \\
&=\left[p_1 \quad p_2 \quad \cdots \quad p_{2N}\right]\begin{bmatrix} e^{s_1(t_1+2\varDelta t)} & e^{s_1(t_2+2\varDelta t)} & \cdots & e^{s_1(t_L+2\varDelta t)} \\ e^{s_2(t_1+2\varDelta t)} & e^{s_2(t_2+2\varDelta t)} & \cdots & e^{s_2(t_L+2\varDelta t)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_{2N}(t_1+2\varDelta t)} & e^{s_{2N}(t_2+2\varDelta t)} & \cdots & e^{s_{2N}(t_L+2\varDelta t)} \end{bmatrix} \\
&=\left[p_1 \quad p_2 \quad \cdots \quad p_{2N}\right]\begin{bmatrix} e^{s_1 2\varDelta t} & & & 0 \\ & e^{s_2 2\varDelta t} & & \\ & & \ddots & \\ 0 & & & e^{s_{2N} 2\varDelta t} \end{bmatrix}\begin{bmatrix} e^{s_1t_1} & e^{s_1t_2} & \cdots & e^{s_1t_L} \\ e^{s_2t_1} & e^{s_2t_2} & \cdots & e^{s_2t_L} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_{2N}t_1} & e^{s_{2N}t_2} & \cdots & e^{s_{2N}t_L} \end{bmatrix}
\end{aligned}
\tag{2.3.3.10}
$$

or,

$$
\underset{1\times L}{[X_2]} = \underset{1\times 2N}{[P]}\ \underset{2N\times 2N}{[\hat{\varLambda}]}^{2}\ \underset{2N\times L}{[\varLambda]} \tag{2.3.3.11}
$$

If $m+1$ data sets of responses are measured and if the last set is shifted m intervals with respect to equation (2.3.3.5), the last response is as follow

$$\begin{aligned}
&\left[x(t_1+m\cdot\Delta t) \quad \cdots \quad x(t_L+m\cdot\Delta t)\right] \\
&=\left[p_1 \quad p_2 \quad \cdots \quad p_{2N}\right]\begin{bmatrix} e^{s_1(t_1+m\cdot\Delta t)} & e^{s_1(t_2+m\cdot\Delta t)} & \cdots & e^{s_1(t_L+m\cdot\Delta t)} \\ e^{s_2(t_1+m\cdot\Delta t)} & e^{s_2(t_2+m\cdot\Delta t)} & \cdots & e^{s_2(t_L+m\cdot\Delta t)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_{2N}(t_1+m\Delta t)} & e^{s_{2N}(t_2+m\Delta t)} & \cdots & e^{s_{2N}(t_L+m\Delta t)} \end{bmatrix} \\
&=\left[p_1 \quad p_2 \quad \cdots \quad p_{2N}\right]\begin{bmatrix} e^{s_1 m\cdot\Delta t} & & & 0 \\ & e^{s_2 m\cdot\Delta t} & & \\ & & \ddots & \\ 0 & & & e^{s_{2N} m\cdot\Delta t} \end{bmatrix}\begin{bmatrix} e^{s_1 t_1} & e^{s_1 t_2} & \cdots & e^{s_1 t_L} \\ e^{s_2 t_1} & e^{s_2 t_2} & \cdots & e^{s_2 t_L} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_{2N} t_1} & e^{s_{2N} t_2} & \cdots & e^{s_{2N} t_L} \end{bmatrix}
\end{aligned} \tag{2.3.3.12}$$

or,

$$\underset{1\times L}{[X_m]} = \underset{1\times 2N}{[P]}\,\underset{2N\times 2N}{\left[\hat{\Lambda}\right]}^{m}\,\underset{2N\times L}{[\Lambda]} \tag{2.3.3.13}$$

The responses can be manipulated to solve for the eigenvalues and eigenvectors. First, the measured m responses are grouped in two matrices as follow,

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{m-1} \end{bmatrix} = \begin{bmatrix} P \\ P\hat{\Lambda} \\ P\hat{\Lambda}^2 \\ \vdots \\ P\hat{\Lambda}^{m-1} \end{bmatrix}\Lambda \tag{2.3.3.14}$$

or,

$$\underset{m\times L}{[\Phi]} = \underset{m\times L}{[\Psi]}\,\underset{2N\times L}{[\Lambda]} \tag{2.3.3.15}$$

and

$$\underset{m\times L}{\left[\hat{\Phi}\right]} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_m \end{bmatrix} = \begin{bmatrix} P\hat{\Lambda} \\ P\hat{\Lambda}^2 \\ P\hat{\Lambda}^3 \\ \vdots \\ P\hat{\Lambda}^m \end{bmatrix} \Lambda = \underset{m\times L}{\left[\hat{\Psi}\right]}\underset{2N\times L}{\left[\Lambda\right]} = \underset{m\times L}{\left[\Psi\right]}\underset{2N\times 2N}{\left[\hat{\Lambda}\right]}\underset{2N\times L}{\left[\Lambda\right]} \tag{2.3.3.16}$$

or

$$\underset{m\times L}{\left[\hat{\Phi}\right]} = \underset{m\times L}{\left[\hat{\Psi}\right]}\underset{2N\times L}{\left[\Lambda\right]} = \underset{m\times L}{\left[\Psi\right]}\underset{2N\times 2N}{\left[\hat{\Lambda}\right]}\underset{2N\times L}{\left[\Lambda\right]} \tag{2.3.3.17}$$

Now equations (2.3.3.12) and (2.3.3.14) are manipulated in order to eliminate [Λ], obtaining

$$\left[\hat{\Phi}\right] = \left[\hat{\Psi}\right]\left[\Psi\right]^{-1}\left[\Phi\right] \tag{2.3.3.18}$$

A square matrix [A_s] of order m is defined as follow,

$$\left[A_s\right] = \left[\hat{\Psi}\right]\left[\Psi\right]^{-1} \tag{2.3.3.19}$$

Substituting equation (2.3.3.16) in equation (2.3.3.15),

$$\left[A_s\right]\left[\Phi\right] = \left[\hat{\Phi}\right] \tag{2.3.3.20}$$

This equation can be solved for [A_s] via the pseudo-inverse technique. This technique leads to two expressions, which are,

$$\begin{aligned} \left[A_s\right] &= \left(\left[\hat{\Phi}\right]\left[\hat{\Phi}\right]^T\right)\left(\left[\Phi\right]\left[\hat{\Phi}\right]^T\right)^{-1} \\ \left[A_s\right] &= \left(\left[\hat{\Phi}\right]\left[\Phi\right]^T\right)\left(\left[\Phi\right]\left[\Phi\right]^T\right)^{-1} \end{aligned} \tag{2.3.3.21}$$

A combination of both equations, known as Double Least-Squares (DLS), is used because it leads to a better estimate of the damping factors. This gives,

$$[A_s] = \frac{1}{2}\left[\left([\hat{\Phi}][\hat{\Phi}]^T\right)\left([\Phi][\hat{\Phi}]^T\right)^{-1} + \left([\hat{\Phi}][\Phi]^T\right)\left([\Phi][\Phi]^T\right)^{-1}\right] \qquad (2.3.3.22)$$

From equations (2.3.3.16) and (2.3.3.14),

$$[[A_s][\Psi] - [\Psi][\hat{\Lambda}]] = 0 \qquad (2.3.3.23)$$

This equation can be written for each column vector, [Ψ_r], of [Ψ] as follow,

$$[[A_s][\Psi_r] - e^{s_r \Delta t}[\Psi_r]] = 0 \qquad (2.3.3.24)$$

or

$$[[A_s] - e^{s_r \Delta t}[I]][\Psi_r] = 0 \qquad (2.3.3.25)$$

This equation is an eigenvalue problem. After calculating [A_s] in equation (2.3.3.19), it can be solved as a standard eigenvalue problem, which will give m eigenvalues and eigenvectors.

The relationship between the eigenvalues, $\beta_r + i\gamma_r$, and $s_r = a_r + ib_r$, the roots of equation (2.3.3.3), is used to calculate the damping ratios, and natural frequencies, as follow,

$$\beta + i\gamma = e^{(a+ib)\Delta t} = V_r \qquad (2.3.3.26)$$

This gives,

$$
\begin{aligned}
R_r &= \ln(V_r) \\
f_r &= \frac{|R_r|}{2\pi \cdot \Delta t} \\
\xi_r &= \sqrt{\frac{1}{1+\left(\frac{\mathrm{Im}\,ag(R_r)}{\mathrm{Re}\,al(R_r)}\right)^2}}
\end{aligned}
\tag{2.3.3.27}
$$

With the values of V_r, we can calculate the residues if equation (2.3.1.17) is used, and substituting *h(t)* by *x(t)*.

2.3.4 Rational Fraction Polynomial Method

This method works in the frequency domain. The formulation of the FRF is expressed in the rational fraction form instead of the partial fraction form where the error function is established in a way that the resulting system of equations is linear. Because the resulting linear system of equations involves matrices that are ill conditioned, the Gradient Method is used to minimize this error function and the initial estimate is calculated by using the Least Square Method. The derivation of this method up to equation 2.3.4.12 is found at pages 237 to 239 of reference [9] and in references [14] and [15]. The theory of the Gradient Method was obtained from reference [16].

The FRF, in terms of receptance, for a linear system, with N DOF, and viscous damping can be modeled with the following partial fraction equation:

$$
H(\omega) = \sum_{r=1}^{N} \frac{A_r + i\omega \cdot B_r}{\omega_r^2 - \omega^2 + i2\xi_r\omega_r\omega} \tag{2.3.4.1}
$$

where A_r and B_r are constants.

It can also be expressed in the rational fraction form as follow,

$$H(\omega) = \frac{\sum_{k=0}^{2N-1} a_k (i\omega)^k}{\sum_{k=0}^{2N} b_k (i\omega)^k} \qquad (2.3.4.2)$$

The difference between the analytical FRF $H(\omega)$ and the experimental FRF $H_e(\omega)$ is the error function given by:

$$e_j = \frac{\sum_{k=0}^{2N-1} a_k (i\omega_j)^k}{\sum_{k=0}^{2N} b_k (i\omega_j)^k} - H_e(\omega_j) \qquad (2.3.4.3)$$

Now, the error function is linearized by working with the following modified error function:

$$e'_j = e_j \sum_{k=0}^{2N} b_k (i\omega_j)^k \qquad (2.3.4.4)$$

and making $b_{2N} = 1$. This leads to,

$$e'_j = \sum_{k=0}^{2N-1} a_k (i\omega_j)^k - H_e(\omega_j)\left[\sum_{k=0}^{2N-1} b_k (i\omega_j)^k + (i\omega_j)^{2N}\right] \qquad (2.3.4.5)$$

An error vector is defined for all the L measured frequencies:

$$\{E\} = \begin{Bmatrix} e'_1 \\ e'_2 \\ \vdots \\ e'_L \end{Bmatrix} \qquad (2.3.4.6)$$

Equation (2.3.4.5) expressed in the matrix form becomes,

$$
\{E\} = \begin{bmatrix} 1 & (i\omega_1) & (i\omega_1)^2 & \cdots & (i\omega_1)^{2N-1} \\ 1 & (i\omega_2) & (i\omega_2)^2 & \cdots & (i\omega_2)^{2N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (i\omega_L) & (i\omega_L)^2 & \cdots & (i\omega_L)^{2N-1} \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2N-1} \end{Bmatrix} -
$$

$$
\begin{bmatrix} H_e(\omega_1) & H_e(\omega_1)(i\omega_1) & \cdots & H_e(\omega_1)(i\omega_1)^{2N-1} \\ H_e(\omega_2) & H_e(\omega_2)(i\omega_2) & \cdots & H_e(\omega_2)(i\omega_2)^{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ H_e(\omega_L) & H_e(\omega_L)(i\omega_L) & \cdots & H_e(\omega_L)(i\omega_L)^{2N-1} \end{bmatrix} \begin{Bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{2N-1} \end{Bmatrix} - \begin{Bmatrix} H_e(\omega_1)(i\omega_1)^{2N} \\ H_e(\omega_2)(i\omega_2)^{2N} \\ \vdots \\ H_e(\omega_L)(i\omega_L)^{2N} \end{Bmatrix} \tag{2.3.4.7}
$$

or,

$$
\underset{(Lx1)}{\{E\}} = \underset{(Lx2N)}{[P]} \underset{(2Nx1)}{\{a\}} - \underset{(Lx2N)}{[T]} \underset{(2Nx1)}{\{b\}} - \underset{(Lx1)}{\{W\}} \tag{2.3.4.8}
$$

The equation that will be minimized with the Gradient Method is the squared error function J,

$$
J = \{E^*\}^T \{E\} \tag{2.3.4.9}
$$

where * indicates the complex conjugate. Then, substituting equation (2.3.4.8) in equation (2.3.4.9), and after performing some manipulations, we obtain

$$
\begin{aligned} J = {} & \{a\}^T \operatorname{Re}\left([P^*]^T [P]\right)\{a\} + \{b\}^T \operatorname{Re}\left([T^*]^T [T]\right)\{b\} + \{W^*\}^T \{W\} - \\ & 2\{a\}^T \operatorname{Re}\left([P^*]^T [T]\right)\{b\} - 2\{a\}^T \operatorname{Re}\left([P^*]^T [W]\right) + 2\{b\}^T \operatorname{Re}\left([T^*]^T [W]\right) \end{aligned} \tag{2.3.4.10}
$$

This is an equation that has ill-conditioned matrices, that is why it has to be solved by using the Gradient Method in order to minimize the error function of equation (2.3.4.10). But, the Least Square Method can be used to obtain the initial estimate needed for the Gradient Method.

The Least Square Method is done by taking the derivatives of equation (2.3.4.10) with respect to $\{a\}$ and $\{b\}$ and equaling them to zero gives the following system of equations:

$$\begin{aligned}
&\mathrm{Re}\left(\left[P^*\right]^T[P]\right)\{a\} - \mathrm{Re}\left(\left[P^*\right]^T[T]\right)\{b\} - \mathrm{Re}\left(\left[P^*\right]^T\{W\}\right) = \{0\} \\
&\mathrm{Re}\left(\left[T^*\right]^T[T]\right)\{b\} - \mathrm{Re}\left(\left[T^*\right]^T[P]\right)\{a\} + \mathrm{Re}\left(\left[T^*\right]^T\{W\}\right) = \{0\}
\end{aligned} \tag{2.3.4.11}$$

or,

$$\begin{bmatrix} [Y] & [X] \\ [X]^T & [Z] \end{bmatrix} \begin{Bmatrix} \{a\} \\ \{b\} \end{Bmatrix} = \begin{Bmatrix} \{G\} \\ \{F\} \end{Bmatrix} \tag{2.3.4.12}$$

where

$$\begin{aligned}
&[Y] = \mathrm{Re}\left(\left[P^*\right]^T[P]\right) \\
&[X] = -\mathrm{Re}\left(\left[P^*\right]^T[T]\right) \\
&[Z] = \mathrm{Re}\left(\left[T^*\right]^T[T]\right) \\
&\{G\} = \mathrm{Re}\left(\left[P^*\right]^T\{W\}\right) \\
&\{F\} = -\mathrm{Re}\left(\left[T^*\right]^T\{W\}\right)
\end{aligned} \tag{2.3.4.13}$$

Then solve for the initial values of $\{a\}$ and $\{b\}$. Then use these values to evaluate the gradient. The gradient with respect to $\{a\}$ is the partial derivative of equation (2.3.4.10) with respect to $\{a\}$:

$$\mathrm{Re}\left(\left[P^*\right]^T[P]\right)\{a\} - \mathrm{Re}\left(\left[P^*\right]^T[T]\right)\{b\} - \mathrm{Re}\left(\left[P^*\right]^T\{W\}\right) = \{M\} \tag{2.3.4.14}$$

where [M] is the gradient vector with respect to $\{a\}$.

The gradient with respect to $\{b\}$ is the partial derivative of equation (2.3.4.10) with respect to $\{b\}$:

$$\mathrm{Re}\left([T^*]^T[T]\right)\{b\} - \mathrm{Re}\left([T^*]^T[P]\right)\{a\} + \mathrm{Re}\left([T^*]^T\{W\}\right) = \{N\} \tag{2.3.4.15}$$

where [N] is the gradient vector with respect to $\{b\}$.

Then with equations (2.3.4.14 and (2.3.4.15) the gradient vector is:

$$\{V\} = \begin{Bmatrix} M \\ N \end{Bmatrix} \tag{2.3.4.16}$$

The gradient vector direction is calculated to subtract it to the coefficients in order to move in the direction where the function is minimized. The gradient vector direction is:

$$\{S\} = \frac{-\{V\}}{\|\{V\}\|} \tag{2.3.4.17}$$

where the $\|\{V\}\|$ is the norm of the vector.

Then, the new coefficients are:

$$\begin{Bmatrix} a \\ b \end{Bmatrix} = \{V\} + \{S\} \tag{2.3.4.18}$$

Then the gradient vector and its respective norm is calculated and compared with the desired tolerance. If the value is greater than the tolerance, the gradient vector direction is subtracted to the coefficients in order to compute new coefficients. The process is repeated until the norm of the gradient vector is smaller than the tolerance.

After obtaining the coefficients of rational fraction equation (2.3.4.2), we can calculate the modal parameters. The roots or poles of the denominator polynomial contain the values of the natural frequency and damping ratio. These can be obtained as follow,

$$f_r = \frac{|P_r|}{2\pi}$$
$$\xi_r = -\frac{\mathrm{Re}\,al(P_r)}{|P_r|} \tag{2.3.4.19}$$

where P_r is the pole of mode r.

In order to calculate the residues, the rational fraction is expanded in a partial fraction equation and the numerator becomes a pair of complex conjugate constants, called residues.

Chapter 3

Analysis and Procedure

Two types of analysis were preformed in this research in order to investigate the accuracy of estimating damping ratio by four different modal parameters extraction techniques. The first analysis was done to a simulated analytical data set with known properties and the second was an experimental data. Both data were processed using a MatLab code (see appendix), which was written specifically for this investigation. After this code loads the frequency response function (FRF) data, it asks if it was truncated and how many lines were truncated. If the data is truncated in the FRF, the program adds spectral zero components to replace the same amount of truncated spectral lines in order to obtain the correct sampling frequency when the impulse response function (IRF) is calculated from the FRF. When some frequency range is isolated with a rectangular window, the truncated components are replaced with spectral zero components. This is done because the CEM, ITD, and Hilbert Envelope method estimate the modal parameters in the time domain. In the Hilbert Envelope method, the time range to estimate the damping ratio is specified on the dB-scale plot of the IRF envelope.

3.1 Simulated Analytical Data

The simulated analytical data was generated in two blocks that were divided by two different set of damping ratios. Each block has five degrees of freedom, the same natural frequencies, and the same residues. This data is divided in different cases to estimate the damping ratio. For the CEM, ITD, and RFP methods it was divided in three cases, where two are in a multiple degree of freedom (MDOF) approach and one in a single-mode approach. The MDOF approach was separated in truncation and without truncation in the FRF. The single-mode approach was performed using an isolating rectangular window of 50 spectral lines. As the Hilbert Envelope Method is a single-mode approach, it was divided in two cases using two different isolating rectangular windows of 50 and 20 spectral lines. For each case the natural frequency, damping ratio, FRF curve fit residual, standard deviation, and damping ratio's percent of error with respect to the analytical input damping ratio were calculated. The FRF curve fit residual is the difference between the original FRF and the regenerated signal (curve fit) with respect to each method. The standard deviation is calculated from the difference between the original FRF and the regenerated dignal.

3.2 Experiment

3.2.1 Equipment

The following equipment was used to perform the modal analysis experiment

- Hewlett Packard (HP) Dynamic Signal Analyzer 35665A (Figure 3.2.1.1)
- Aluminum T-plate structure (Figure 3.2.1.2)
- Foam support for the structure (Figure 3.2.1.2)
- Four pounds electromagnetic shaker (Figure 3.2.2.1)
- Accelerometer (Figure 3.2.2.1)
- Amplifier (Figure 3.2.2.1)

Figure 3.2.1.1: Hewlett Packard (HP) Dynamic Signal Analyzer 35665A

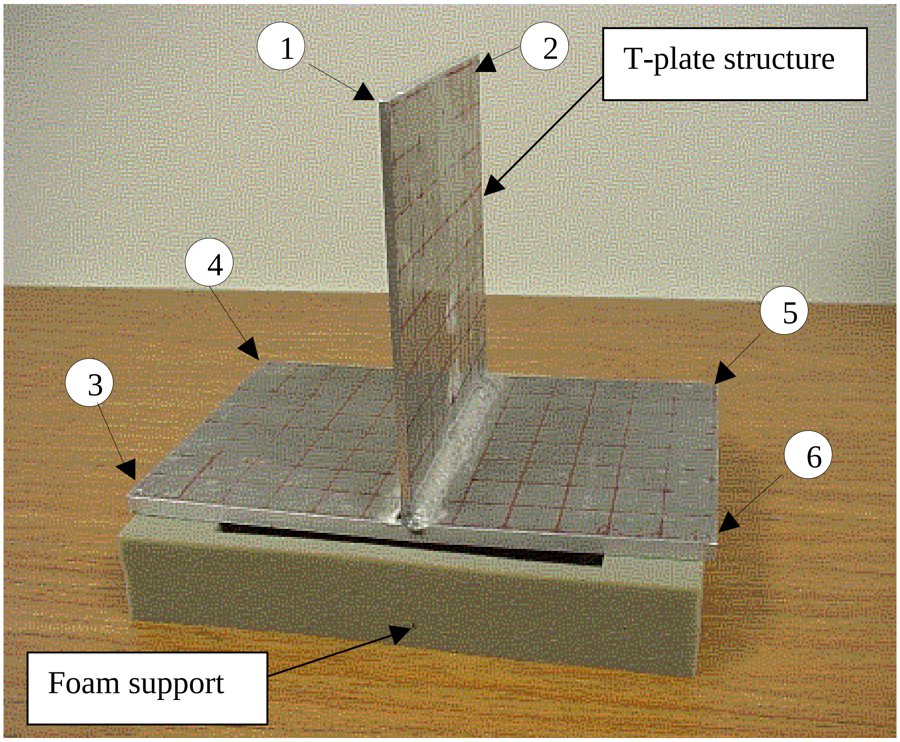


Figure 3.2.1.2: Specified output measuring location points of the T-plate structure

3.2.2 Setup

The damping ratio measurements were performed on a T-plate structure. The T-plate is place on the foam to approximate a freely suspended structure. The electromagnetic shaker was placed at the driving point at location point six (Figure 3.2.1.2). This shaker is used to excite the structure via a swept sine or chirp excitation in a frequency range from 0 to 800 Hz. The response is measured with an accelerometer that is placed normal to surface at each specified point on figure 3.2.1.2. The measurement configuration setup is shown in figure 3.2.2.1.

The HP Dynamic Signal Analyzer was setup to measure 25 averages, to send swept sine from 0 to 800 Hz, to measure the input (channel 1) and output (channel 2), and to calculate the FRF and coherence. The FRF and coherence is displayed at the same time in an upper and lower window, respectively. No window was applied to the signal. After everything is connected and the HP dynamic signal analyzer is setup, the start button, on the analyzer, is pressed in order to start the 25 averaging measurements. When the 25 averages are completed, this data is saved in a floppy disk, and the accelerometer is changed to the next location. The process is the same for all the location points.

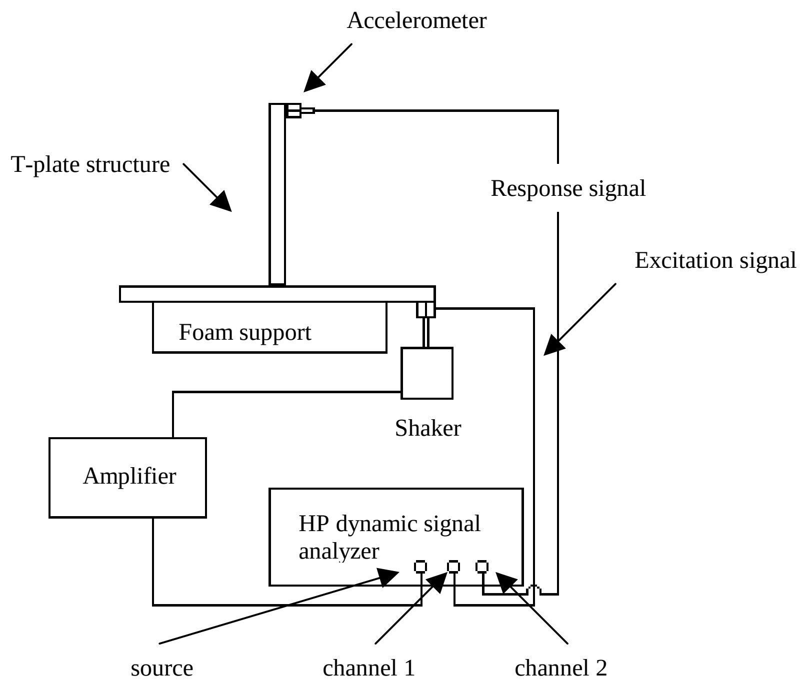


Figure 3.2.2.1: Experiment configuration setup

3.2.3 Analysis

The experimental modal analysis data was measured (output) on six different locations (each corner) of the T-plate structure (Figure 3.2.1.2.), where location six is the driving point (input). The HP dynamic signal analyzer calculated the FRF and coherence in a frequency range from 0 to 800 Hz and 25 averages. To obtain coherence equal to one, a chirp excitation was used because it is a controlled input that doesn't have uncorrelated content. The analyzer calculates the FRF as in chapter 2. After obtaining the FRF data, it is saved in a floppy disk and the data format is changed to ASCII format that is divided in frequency, real, and imaginary columns, 1, 2, and 3, respectively. This data is loaded into the MatLab program and processed using the four modal parameters extraction techniques. Due to the anti-aliasing filter, the FRF is truncated, and the sampling frequency is equal to 2.56 multiplied by the highest spectral frequency line. To obtain the number of truncated spectral lines multiply the sampling frequency by the inverse of the frequency resolution, and then subtract the amount of the FRF spectral lines to this result. For this analysis the CEM, ITD and RFP were used in an MDOF approach, and the Hilbert Envelope method in a single-mode approach. The same values were calculated as in the previous section, except for the damping ratio's percent of error because there is no way to know which is the exact damping ratio. For this reason the coherence has to be equal to one, and the residual between the original and the regenerated (curve fit) signal is calculated to check the curve fit validity, therefore the estimated damping ratio.

CHAPTER 4

Results and Discussion

This chapter has the numerical estimated values of the processed data. The necessary plots are shown in order to demonstrate and discuss how the damping ratio values are affected with respect to some parameters. This chapter is separated in two major sections: the simulated analytical data, and the experimental data

The results are discussed with respect to how the accuracy of the estimated damping ratios is affected and why. The discussion of the simulated analytical data results is more precise because the error was calculated. The experimental data results are the results processed from the experimental FRF measured on the T-plate structure where the error could not be calculated because there is no way to now the exact damping ratio value. In order to evaluate the damping ratio accuracy, the residual between the original data curve and the curve fit were calculated for the simulated analytical and experimental data. Then the local and global standard deviations were calculated from the residual. Having all this in mind is how these results were analyzed.

4.1 Simulated Analytical Data

In this section the parameters of the simulated analytical data sets are presented. Table 4.1.1 gives the data collection parameters, which are the frequency response function (FRF) parameters. The properties of the first and second data sets are presented in tables 4.1.2 and 4.1.3, respectively. The first data set has lower damping ratio than the second data set. Both have the same degree of freedoms equal to five and the same natural frequencies. The impulse response function, FRF, and phase angle plots for each simulated data set is represented on figure 4.1.1 and 4.1.2. The first three modes for each data set are well separated and the last two are close to each other (see figure 4.1.1 and 4.1.2).

Figures 4.1.3 and 4.1.4 show the FRF, phase angle and IRF plots of the truncated first and second data sets. The time leakage is not visible for the first data, but the second data shows it at the end of the IRF (figure 4.1.4(c)). Where the time leakage is due to the truncation in the FRF.

The tables and figures of the estimated parameters are divided by method. The Complex Exponential Method (CEM), the Ibrahim Time Domain (ITD) Method, and the Rational Fraction Polynomial (RFP) Method are organized in two subsections, MDOF and single-mode approach. The Hilbert Envelope Method is a single-mode method. The tables have the values of the estimated damping ratio, the damping ratio's percent of error, and local and global standard deviation with respect to each mode. The damping ratio's percent of error is relative to the damping ratio's value on table 4.1.2 and 4.1.3 with respect to the mode and data set. The standard deviations are calculated from the residual, or difference, between the original and regenerate FRF data. The local standard deviation is calculated around the natural frequencies with a range of 10 spectral lines. The regenerated signal was estimated using the estimated modal parameters. The figures are shown on each section after the tables. The residual plots are the values of the difference between the original and regenerated FRF data plotted with respect to each frequency.

Table 4.1.1: Data collection parameters

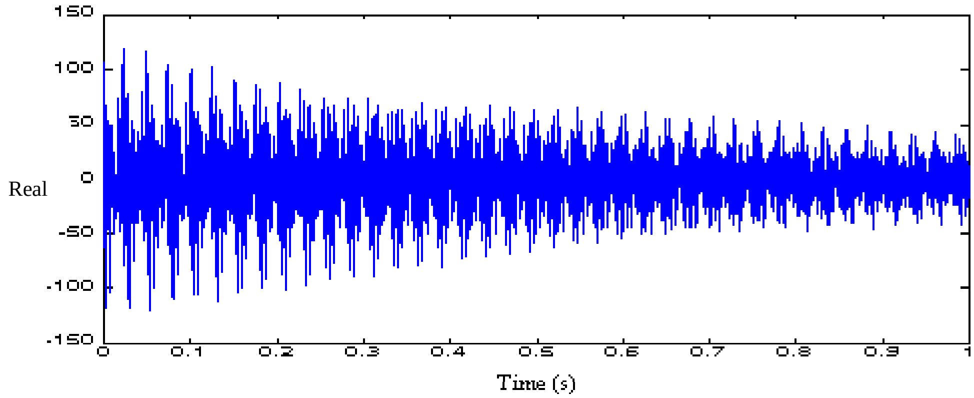
Sample rate	2048 Hz
Number of samples	2048
Frequency resolution	1 Hz
Nyquist frequency	1024 Hz

Table 4.1.2: Properties of the first simulated data set (lower damping ratio)

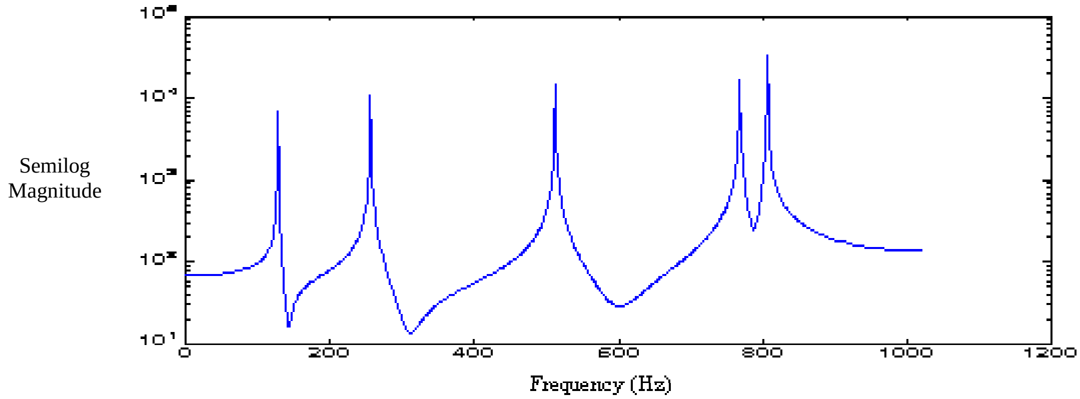
Mode	Residue	Natural Frequency (Hz)	Damping Ratio
1	10	128	0.00100000000
2	20	256	0.00083333333
3	30	512	0.00046296296
4	39	768	0.00030864198
5	50.7	806.4	0.00015432099

Table 4.1.3: Properties of the second simulated data set (higher damping ratio)

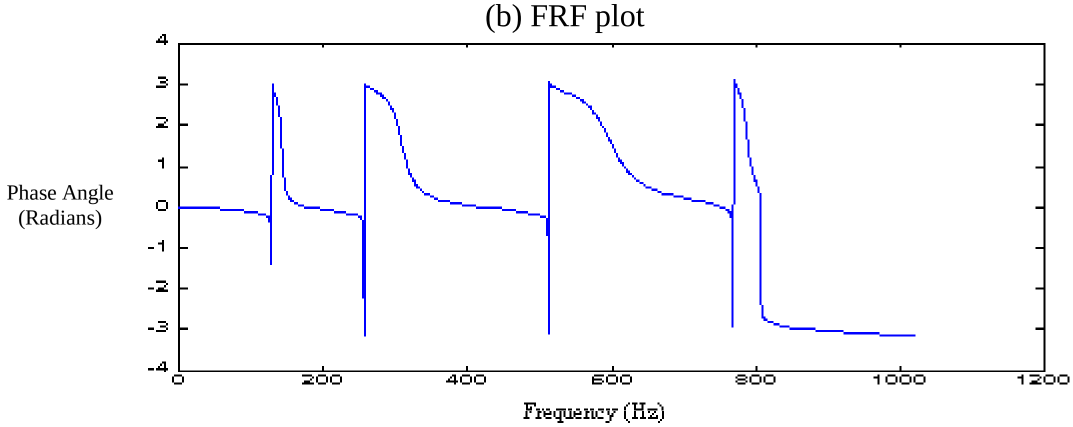
Mode	Residue	Natural Frequency (Hz)	Damping Ratio
1	10	128	0.00400000000
2	20	256	0.00333333333
3	30	512	0.00185185185
4	39	768	0.00123456790
5	50.7	806.4	0.00061728395



(a) IRF plot

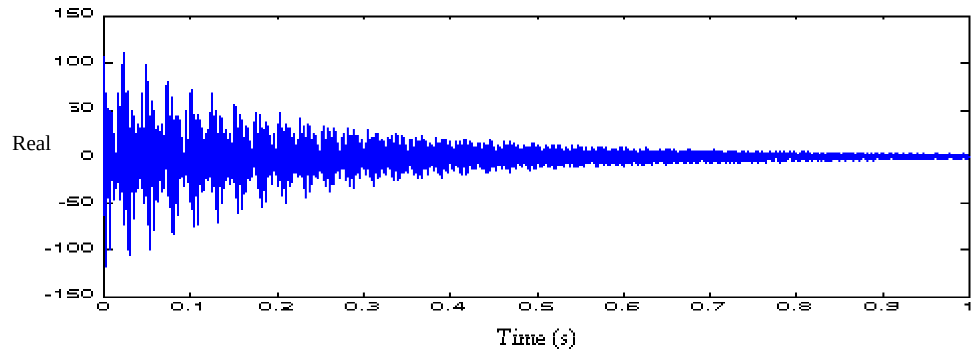


(b) FRF plot

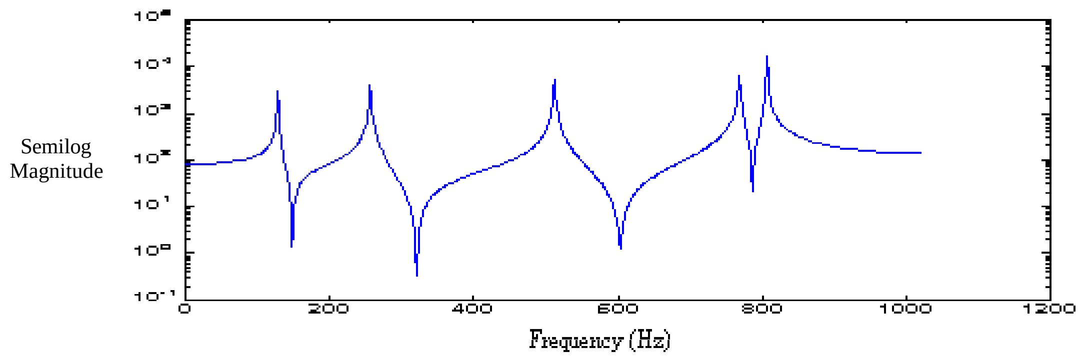


(c) Phase angle plot

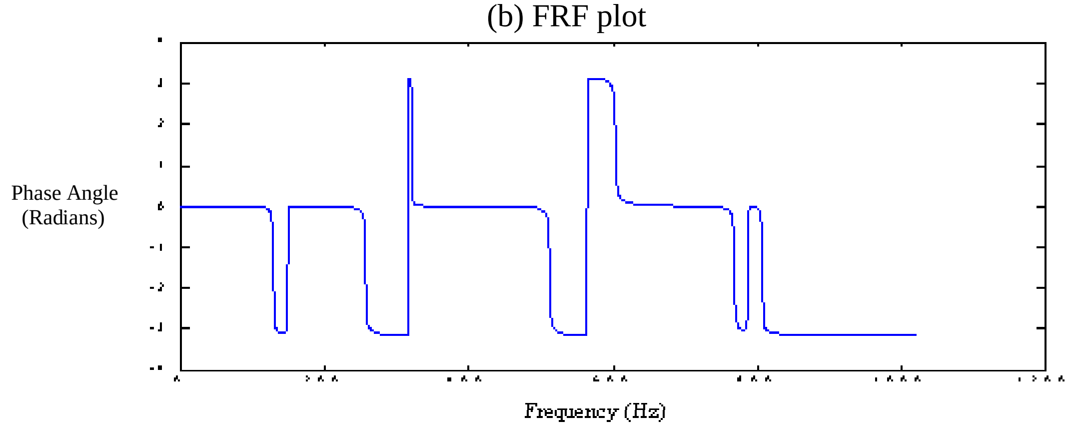
Figure 4.1.1: IRF (a), FRF (b) and Phase Angle (c) of the first simulated data set without truncation in the FRF



(a) IRF plot

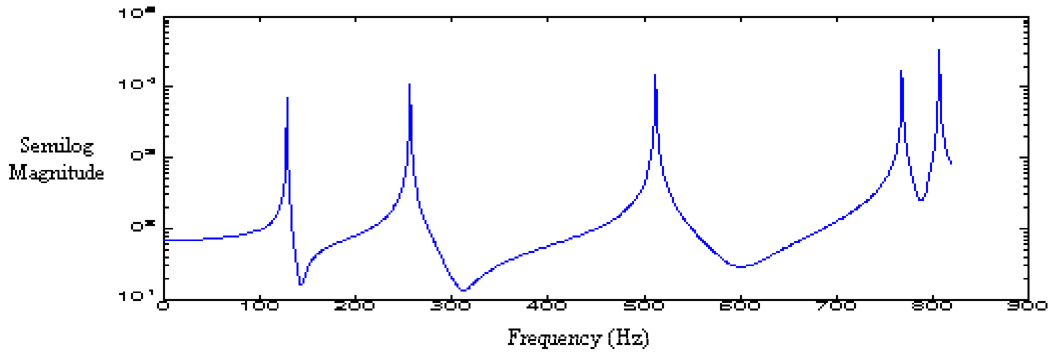


(b) FRF plot

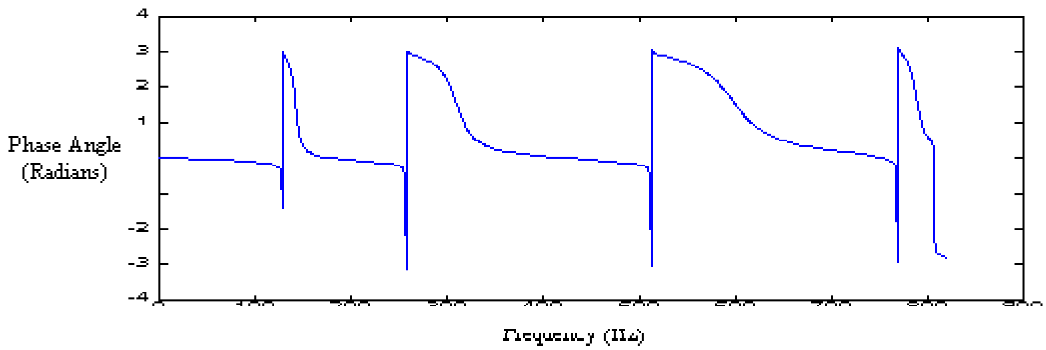


(c) Phase angle plot

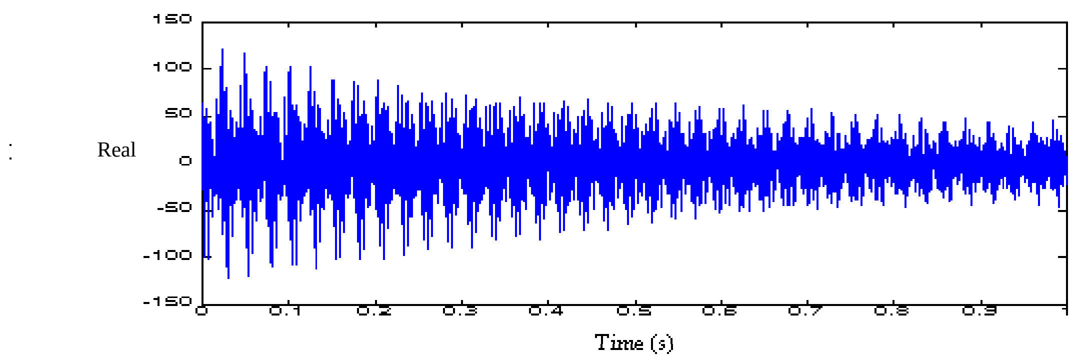
Figure 4.1.2: IRF (a), FRF (b) and Phase Angle (c) of the second simulated data set without truncation in the FRF



(a) FRF plot truncated at 820 Hz

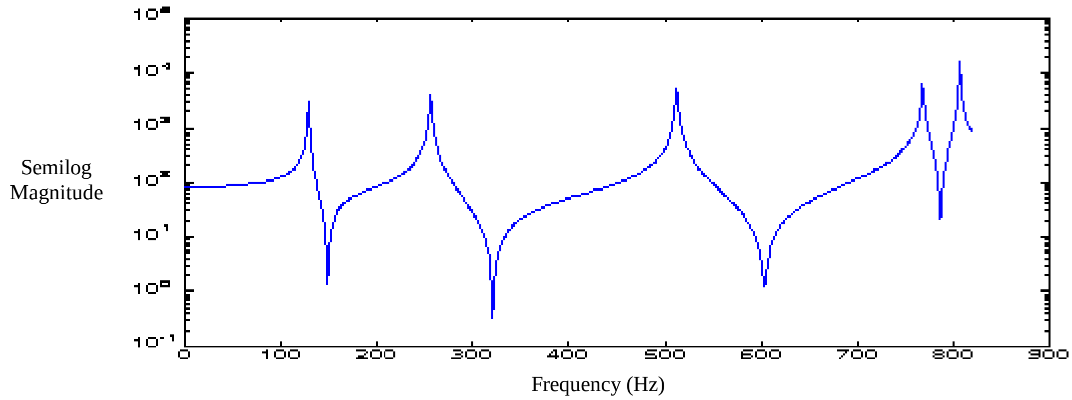


(b) Phase angle plot truncated at 820 Hz

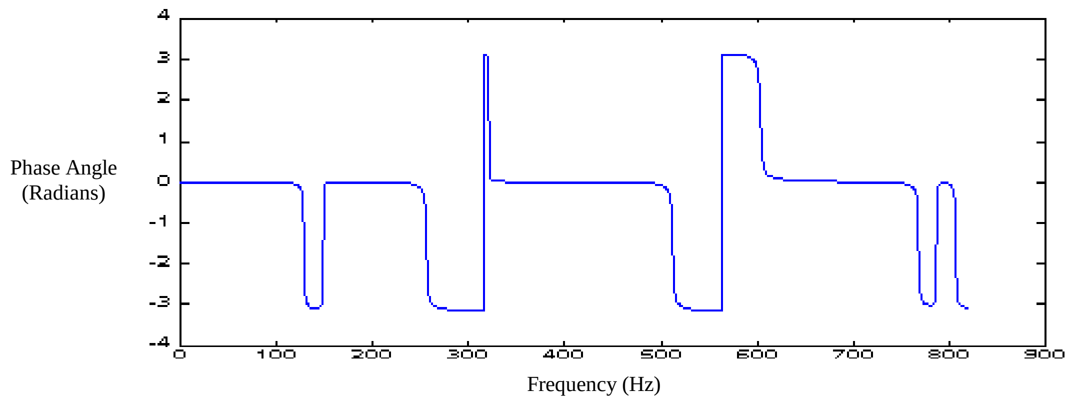


(c) IRF plot calculated from the truncated FRF

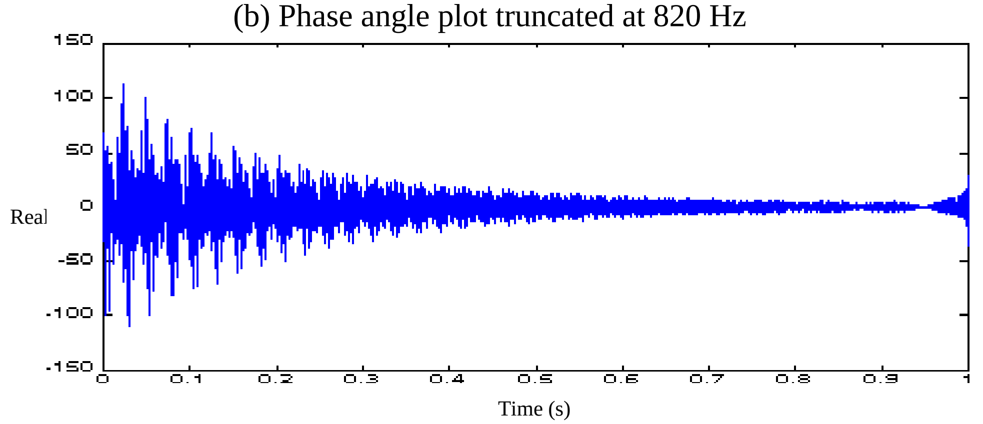
Figure 4.1.3: FRF (a), Phase angle (b) and IRF (c) of the first simulated data set truncated at 820 Hz



(a) FRF plot truncated at 820 Hz



(b) Phase angle plot truncated at 820 Hz



(c) IRF plot calculated from the truncated FRF

Figure 4.1.4: FRF (a), Phase angle (b) and IRF (c) of the second simulated data set truncated at 820 Hz

Complex Exponential Method (CEM) Estimated Parameters

In this method we can observe from the MDOF approach that it gives better damping ratio estimation for higher damping ratios. This is shown in the two cases, with and without truncation, when the damping ratio's percent of error, and the local and global standard deviation of the second data set is smaller than the first data set in each case (see tables 4.1.4 to 4.1.7). Another general behavior, for all the cases in CEM, is that the residual plot (figures 4.1.5(c), 4.1.6(c), 4.1.8(b), and 4.1.9(c)) shows that the residual increases when the natural frequency increases. This is due because of the way the auto-regression works by dividing the IRF in *n* blocks (section 2.3.1). Another fact that affects the estimation of damping ratios is when the mode is close to another mode. This can be noticed because the difference in the percent of error, and local standard deviation between the separated (first three) and close modes (last two) is higher than the difference among the values of the separated modes; the residual plots has a sudden increase at the close modes.

The estimated damping ratios, for the case without truncation, are very accurate for both data sets because the damping ratio's percent of error, and the local and global standard deviation are very small. Also, if we observe figure 4.1.5(a)(b) and 4.1.6(a)(b), they show that the curve fit is the same as the analytical simulated curve.

The numerical values (table 4.1.6 and 4.1.7) and figures (figures 4.1.7 to 4.1.9) of the case with truncation at 820 Hz demonstrate that the accuracy of estimating the damping ratio is affected when the FRF is truncated. The damping ratio's percent of error, and the local and global standard deviation increased for all the modes relative to the case without truncation. But, the effect on the first three modes is not too high; the estimated damping ratios are almost the same as the analytical damping ratios. Figure 4.1.7 shows the FRF analytical curve and curve fit in the same graph for the first data set with truncation, it has a zoom-in at the peak of mode 4 and 5 to show that the curves don't match. Figure 4.1.9, second data set with truncation, doesn't have a zoom-in because it has the same behavior as the first data set. We can say that the truncation affected mode 5 more than mode 4, and this effect is less while the mode is farther from the truncation because of the differences in the statistical values, relative to the case without truncation. The problem with the truncation is that when the IRF is calculated it has time leakage and some information is lost, therefore the estimated damping ratio will deviate from the exact solution.

The numerical values (table 4.1.8 and 4.1.10) and figures (figure 4.1.10) of the single-mode approach demonstrate that while the FRF looses more information the estimated damping ratio deviates from its exact value. It has the same problem as in the case of and MDOF approach with truncated FRF, except that the FRF was truncated at both side of the mode in order to isolate it.

Multiple Degree of Freedom (MDOF) Approach

Table 4.1.4: CEM estimated parameters in a MDOF approach of the first simulated data set without truncation

Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128.00000002	0.00100000002	0.00000238082	0.000051445	
2	256.00000000	0.00083333340	0.00000859204	0.000072014	
3	512.00000000	0.00046296299	0.00000564190	0.000075790	0.00023630737
4	768.00000006	0.00030864207	0.00003213200	0.000709280	
5	806.39999989	0.00015432097	0.00001101768	0.001458200	

Table 4.1.5: CEM estimated parameters in a MDOF approach of the second simulated data set without truncation

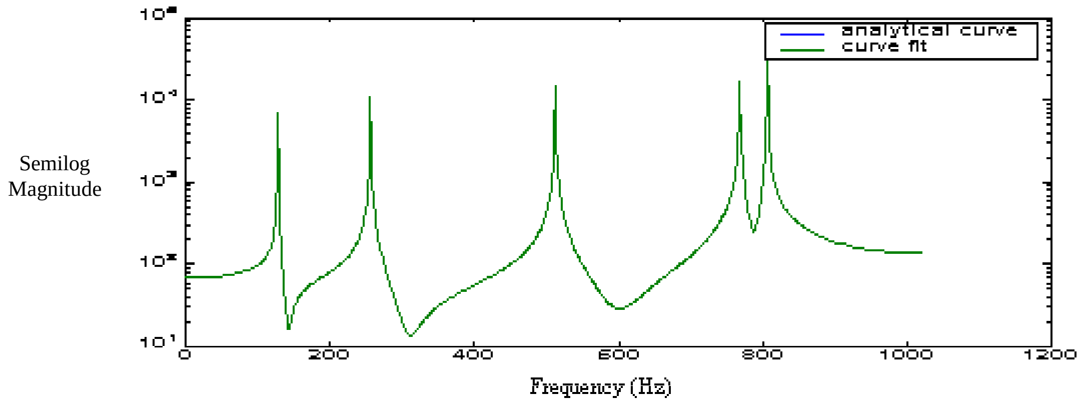
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128.00000010	0.00399999987	0.00000328901	0.000080776	
2	256.00000003	0.00333333367	0.00001024504	0.000071750	
3	512.00000004	0.00185185194	0.00000452935	0.000057242	0.00013462825
4	768.00000031	0.00123456808	0.00001487944	0.000439940	
5	806.39999981	0.00061728395	0.00000082170	0.000803670	

Table 4.1.6: CEM estimated parameters in a MDOF approach of the first simulated data set truncated at 820 Hz

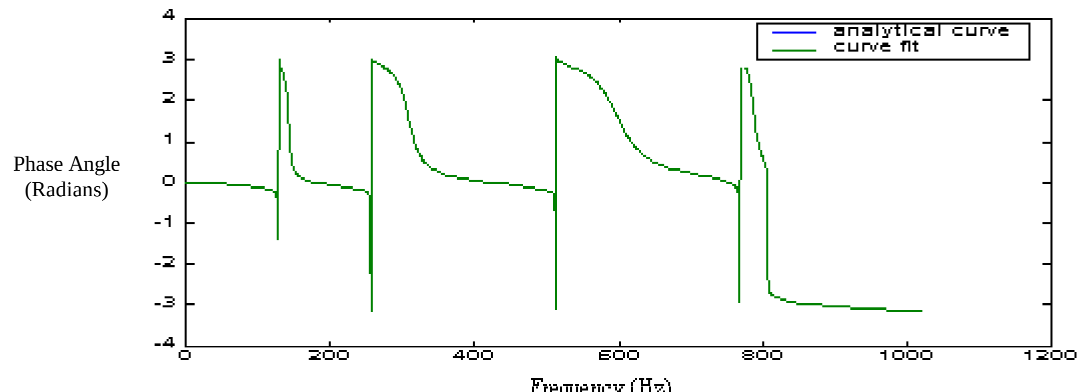
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	127.99998088	0.00100051012	0.05101232596	18.0164	
2	255.99997161	0.00083351397	0.02167665827	19.4848	
3	512.00007479	0.00046282326	0.03017541793	27.5647	117.9329441
4	768.01489261	0.00032293615	4.63131310315	147.5440	
5	806.43911356	0.00019440415	25.97389219865	671.0049	

Table 4.1.7: CEM estimated parameters in a MDOF approach of the second simulated data set truncated at 820 Hz

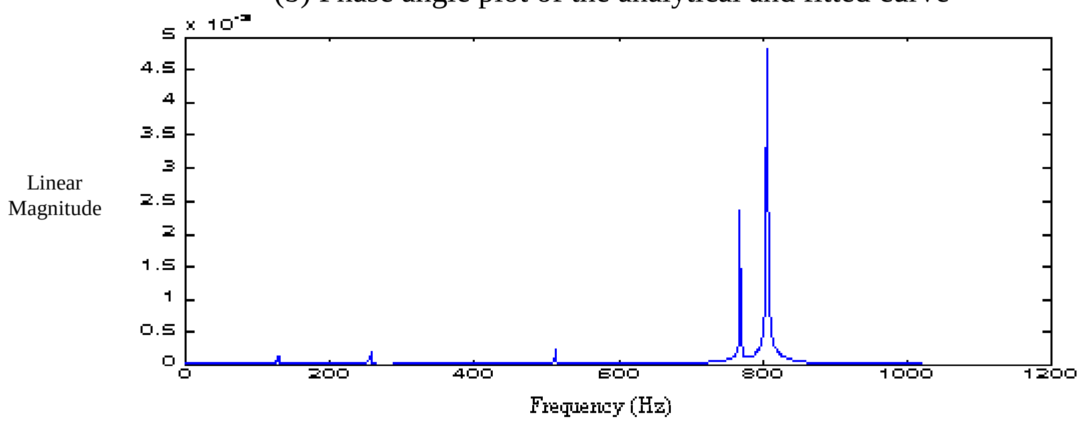
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	127.99996014	0.00400114359	0.02858964501	21.2748	
2	255.99991989	0.00333386134	0.01584006567	22.9280	
3	512.00021992	0.00185141162	0.02377229833	31.6119	91.7418743
4	768.05015678	0.00127455279	3.23877615600	107.9156	
5	806.49818506	0.00069758168	13.0082324809	485.0540	



(a) FRF plot of the analytical and fitted curve

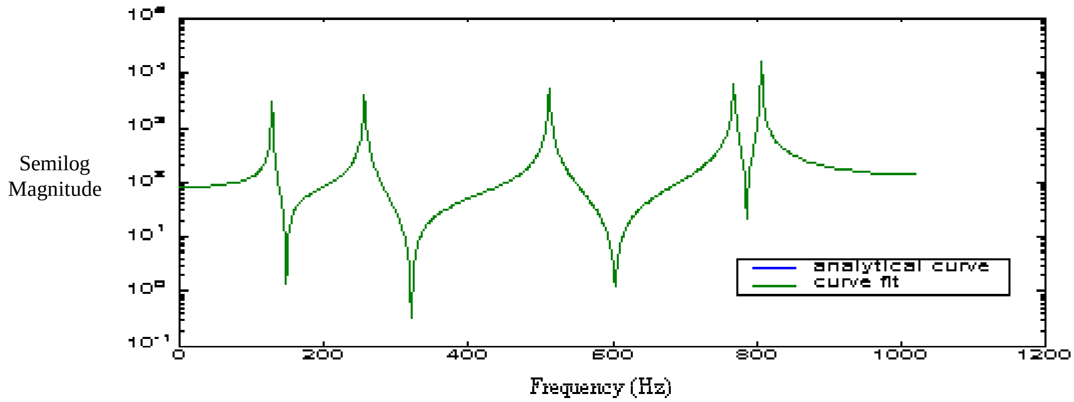


(b) Phase angle plot of the analytical and fitted curve

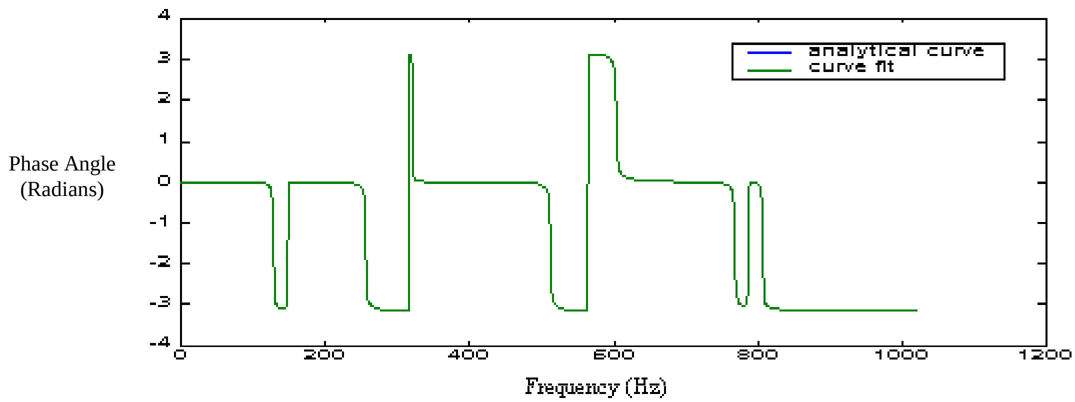


(c) Residual plot

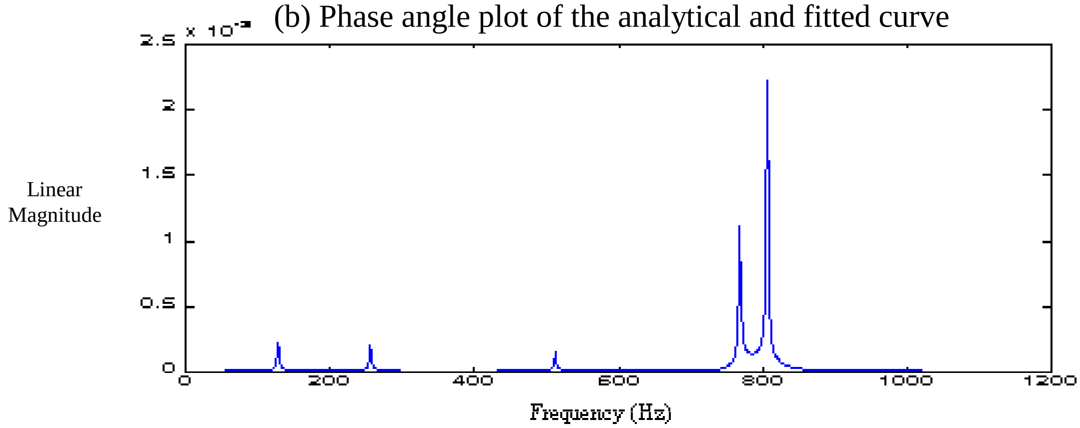
Figure 4.1.5: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and CEM curve fit performed to the first simulated data without truncation



(a) FRF plot of the analytical and fitted curve



(b) Phase angle plot of the analytical and fitted curve



(c) Residual plot

Figure 4.1.6: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and CEM curve fit performed to the second simulated data without truncation

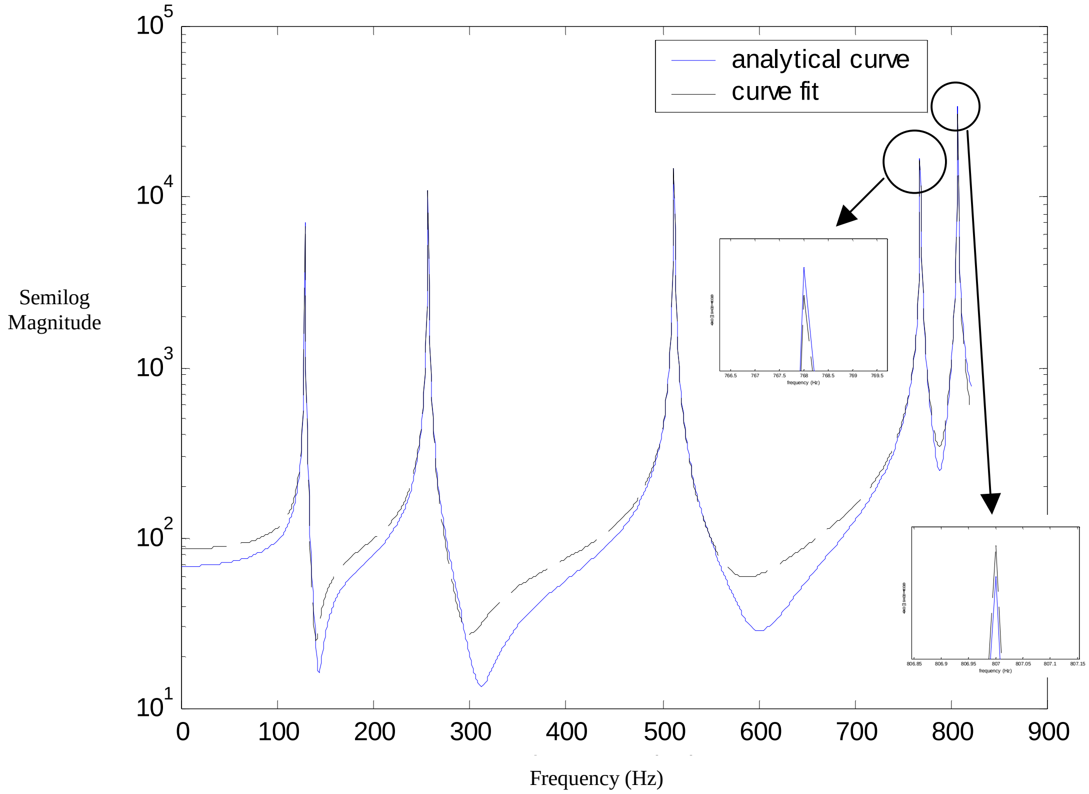
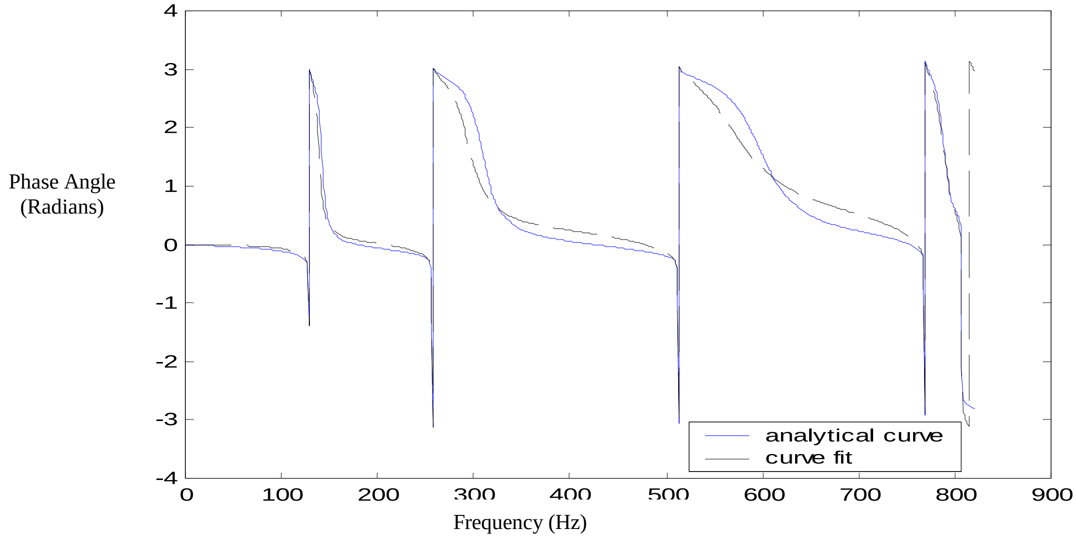
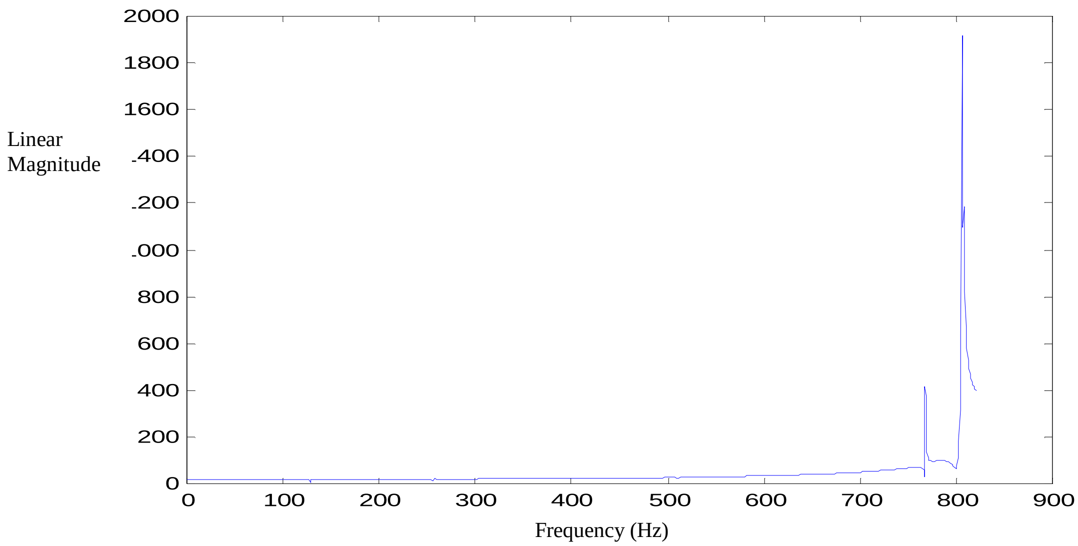


Figure 4.1.7: FRF plot of the analytical curve and CEM curve fit performed to the first simulated data truncated at 820 Hz

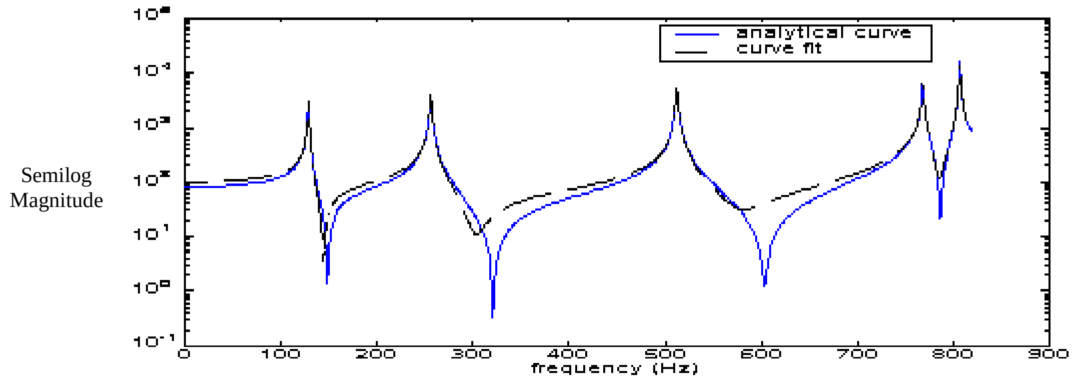


(a) Phase angle plot of the analytical and fitted curve

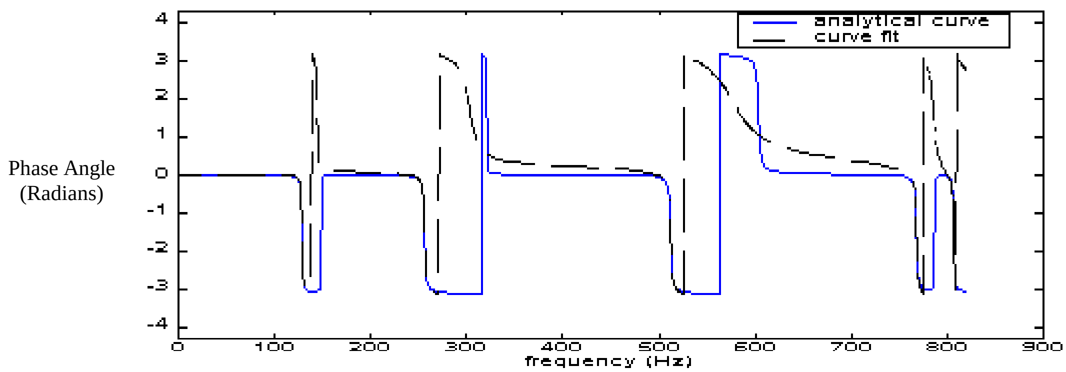


(b) Residual plot

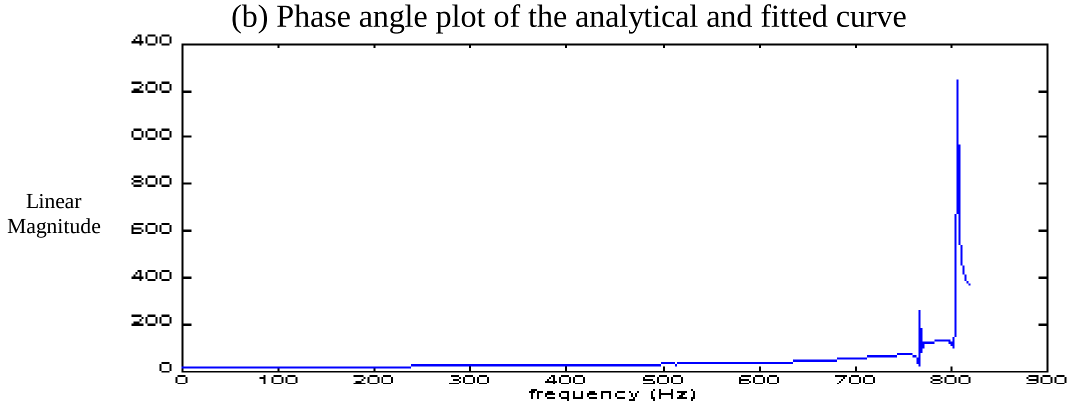
Figure 4.1.8: Phase angle (a) and Residual (b) plots of the analytical curve and CEM curve fit performed to the first simulated data truncated at 820 Hz



(a) FRF plot of the analytical and fitted curve



(b) Phase angle plot of the analytical and fitted curve



(c) Residual plot

Figure 4.1.9: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and CEM curve fit performed to the second simulated data truncated at 820 Hz

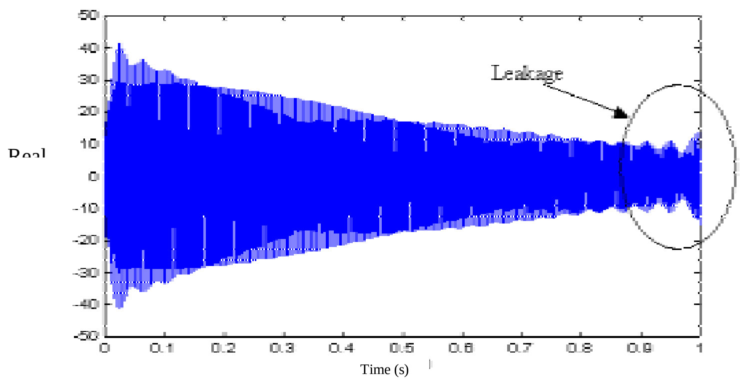
Single Mode Approach

Table 4.1.8: CEM estimated parameters of the first simulated data set in a single-mode approach with an isolating rectangular window of 51 spectral lines

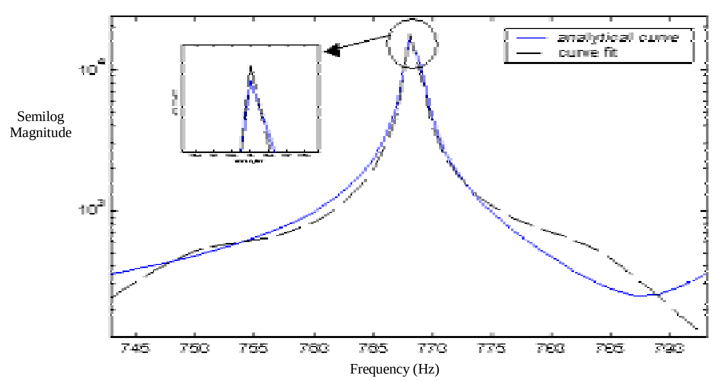
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Standard Deviation (+ - 25 Spectral Lines)
1	128.10984721	0.00035871133	64.12886674597	229.9660
2	255.92366412	0.00050175807	39.78903196674	307.9028
3	511.99243992	0.00084732274	83.02171254053	570.5418
4	767.95115265	0.00020224175	34.47367252959	459.0667
5	806.46152785	0.00005710131	62.99835266054	865.4528

Table 4.1.9: CEM estimated parameters of the second simulated data set in a single-mode approach with an isolating rectangular window of 51 spectral lines

Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Standard Deviation (+ - 25 Spectral Lines)
1	128.20835219	0.00273545441	31.61363968872	153.1857
2	255.87600839	0.00224271375	32.71858735378	187.8851
3	511.73576583	0.00310072664	67.43923843020	293.3045
4	767.86879634	0.00093558755	24.21740881082	292.1937
5	806.48688785	0.00043516735	29.50288884182	531.2362



(a) Single mode IRF plot



(b) Single mode isolated FRF plot with the analytical and fitted curve

Figure 4.1.10: The 768 Hz natural frequency IRF (a) and FRF plot (b) with the analytical and fitted curve of the CEM single-mode approach of the first simulated data set

Ibrahim Time Domain (ITD) Method Estimated Parameters

This method shows the same behavior as the CEM with respect that it works better for higher damping ratios and that the accuracy of the estimated damping ratio is affected in the same way when the FRF is truncated for the MDOF and single-mode approach. To corroborate this, observe from table 4.1.10 to 4.1.13 how the damping ratio's percent of error, and local and global standard deviation change when the FRF is truncated. Also, when the FRF is truncated in the MDOF approach, it affects more the modes that are near the truncation. Because this method works in the time domain, it has the same problem as the CEM with respect to truncation due to the lost of information when the IRF is calculated.

The residual plots (figure 4.1.11 and 4.1.12), the damping ratio's percent of error, and the local standard deviation (table 4.1.9 and 4.1.10) of the MDOF approach without truncation show that the accuracy of the estimated damping ratio doesn't depend if the natural frequency increases, it is affected if there are close modes (mode 4 and 5).

Multiple Degree of Freedom (MDOF) Approach

Table 4.1.10: ITD estimated parameters in a MDOF approach of the first simulated data set without truncation

Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128.00000005	0.00099999997	0.00000316577	0.000134570	
2	256.00000001	0.00083333340	0.00000818810	0.000084776	
3	512.00000001	0.00046296298	0.00000354197	0.000087589	0.00013296604
4	768.00000010	0.00030864201	0.00001087546	0.000749560	
5	806.39999997	0.00015432101	0.00001666532	0.000500050	

Table 4.1.11: ITD estimated parameters in a MDOF approach of the second simulated data set without truncation

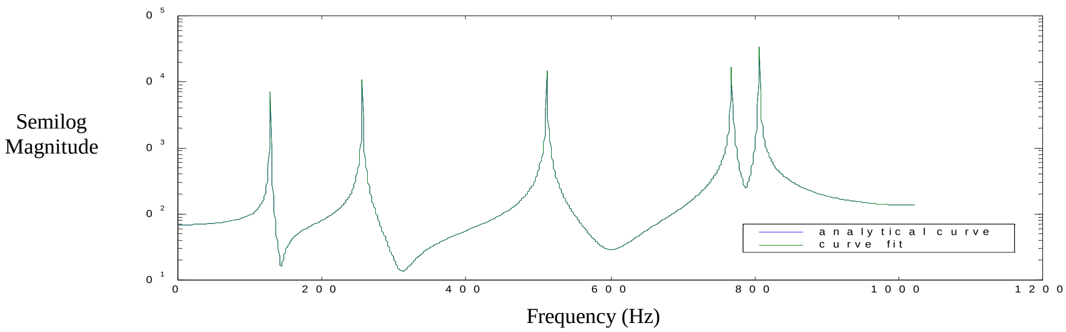
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128.00000012	0.00399999981	0.00000477448	0.000100270	
2	256.00000004	0.00333333369	0.00001068740	0.000075970	
3	512.00000004	0.00185185194	0.00000450143	0.000059715	0.00011486165
4	768.00000032	0.00123456808	0.00001487944	0.000446540	
5	806.39999985	0.00061728397	0.00000331379	0.000633970	

Table 4.1.12: ITD estimated parameters in a MDOF approach of the first simulated data set truncated at 820 Hz

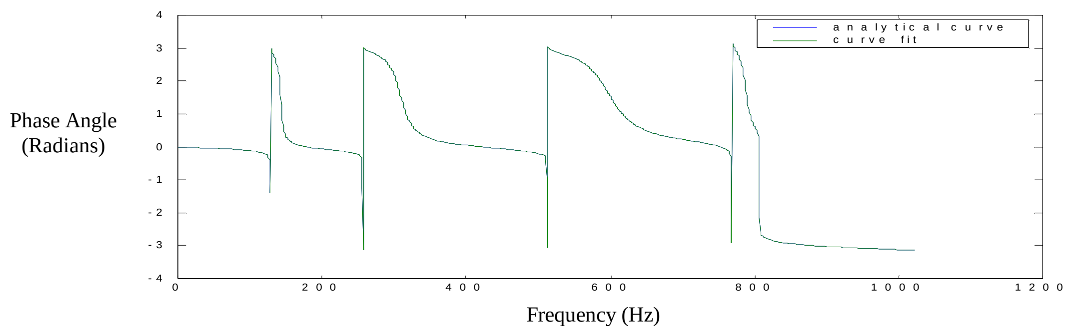
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	127.99999976	0.00100005500	0.00549958966	16.8171	
2	255.99999760	0.00083333474	0.00016877183	18.2278	
3	511.99998566	0.00046296427	0.00028145092	24.9213	39.3162230
4	768.00332066	0.00031407865	1.76148115113	62.2212	
5	806.40514861	0.00014188749	8.05690921548	172.5245	

Table 4.1.13: ITD estimated parameters in a MDOF approach of the second simulated data set truncated at 820 Hz

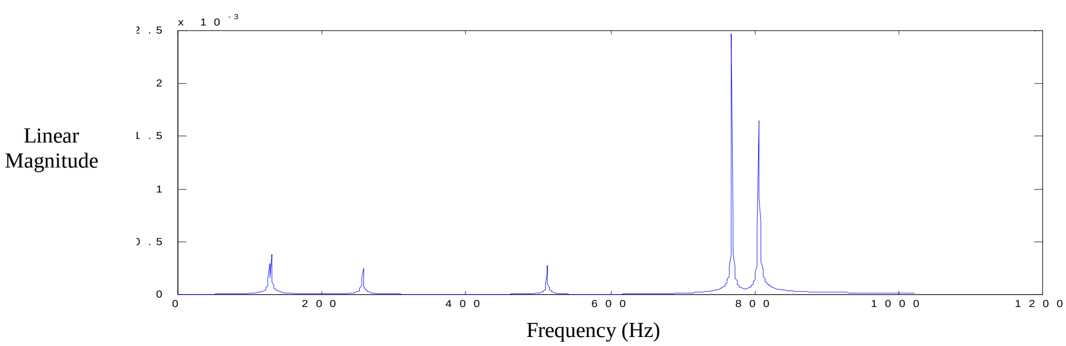
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	127.99999767	0.00399999699	0.00007530750	13.7279	
2	256.00000096	0.00333334121	0.00023636548	16.6116	
3	511.99999976	0.00185183864	0.00071328902	23.1991	27.0435102
4	768.01351384	0.00124852907	1.13085506865	62.7624	
5	806.41112259	0.00059942870	2.8925499435	117.5541	



(a) FRF plot of the analytical and fitted curve

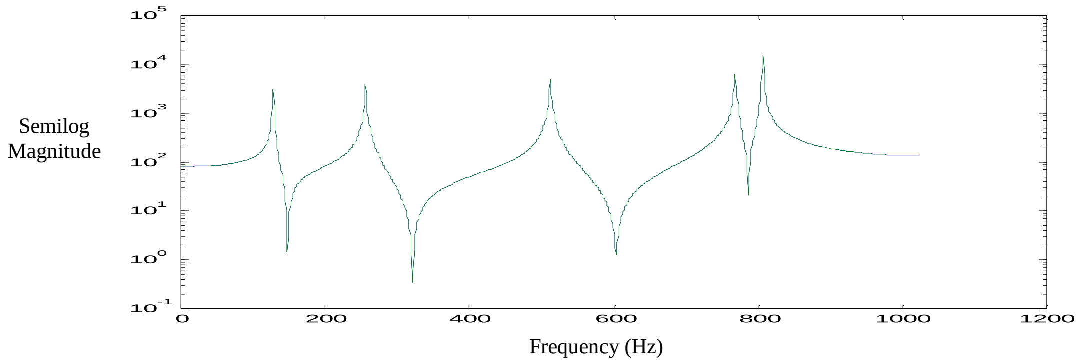


(b) Phase angle plot of the analytical and fitted curve

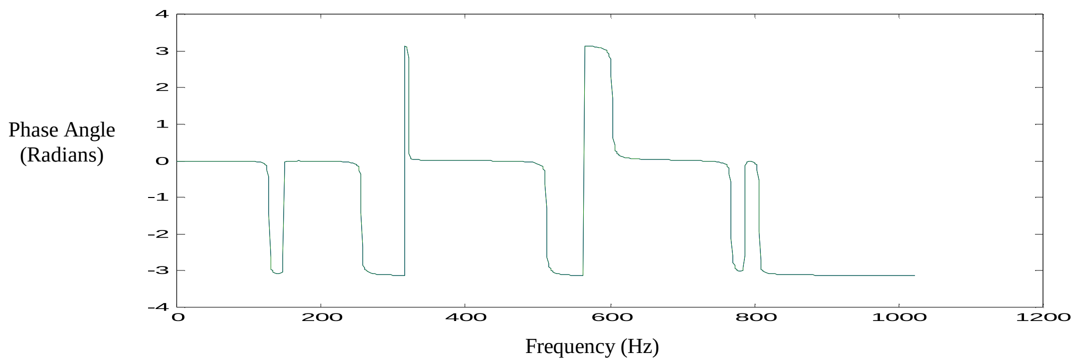


(c) Residual plot

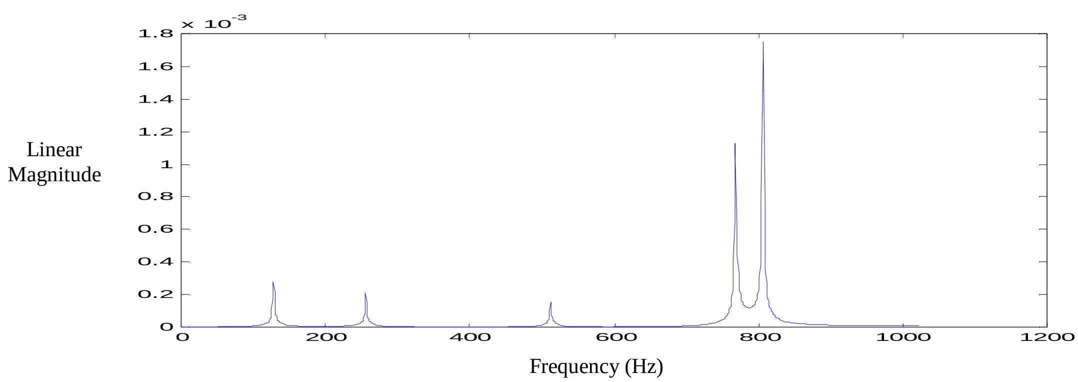
Figure 4.1.11: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and ITD curve fit performed to the first simulated data without truncation



(a) FRF plot of the analytical and fitted curve



(b) Phase angle plot of the analytical and fitted curve



(c) Residual plot

Figure 4.1.12: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and ITD curve fit performed to the second simulated data without truncation

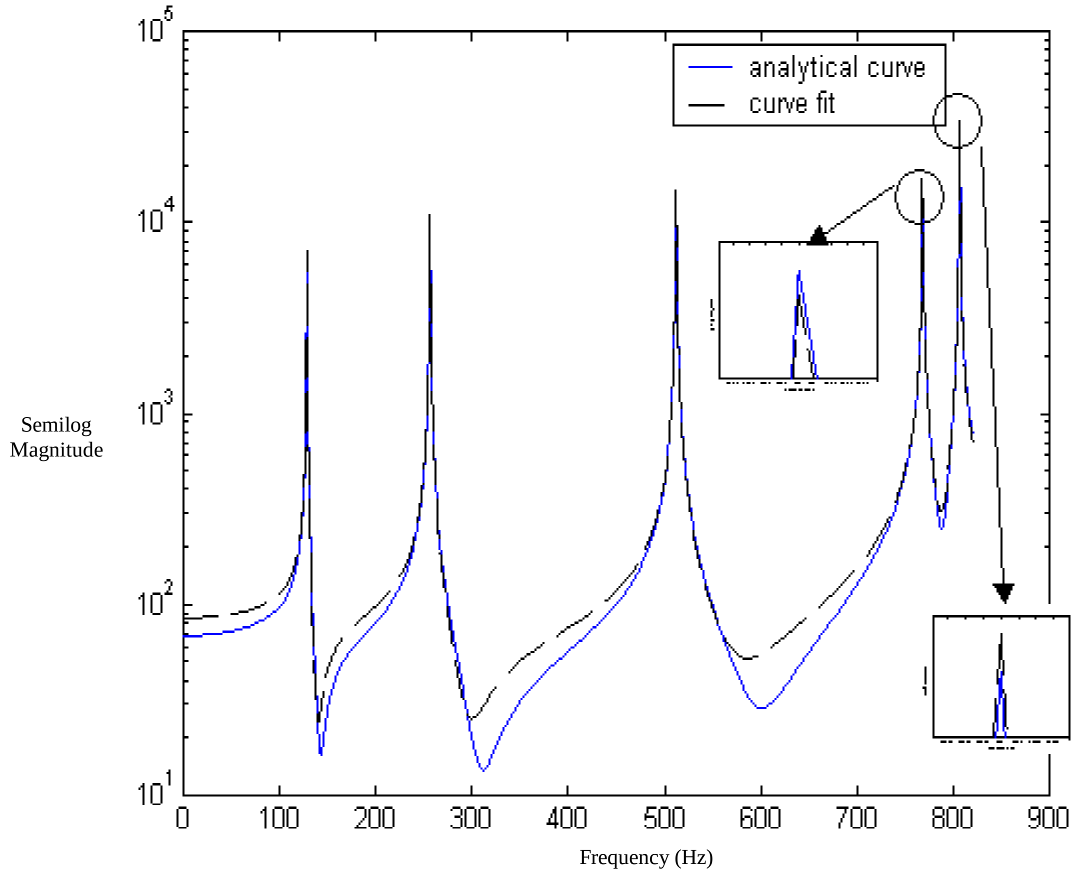
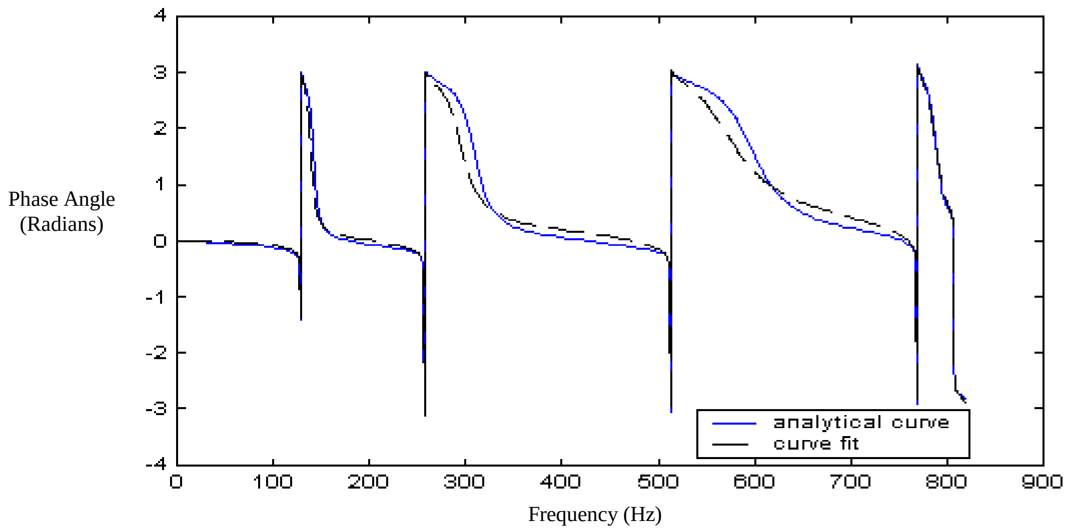
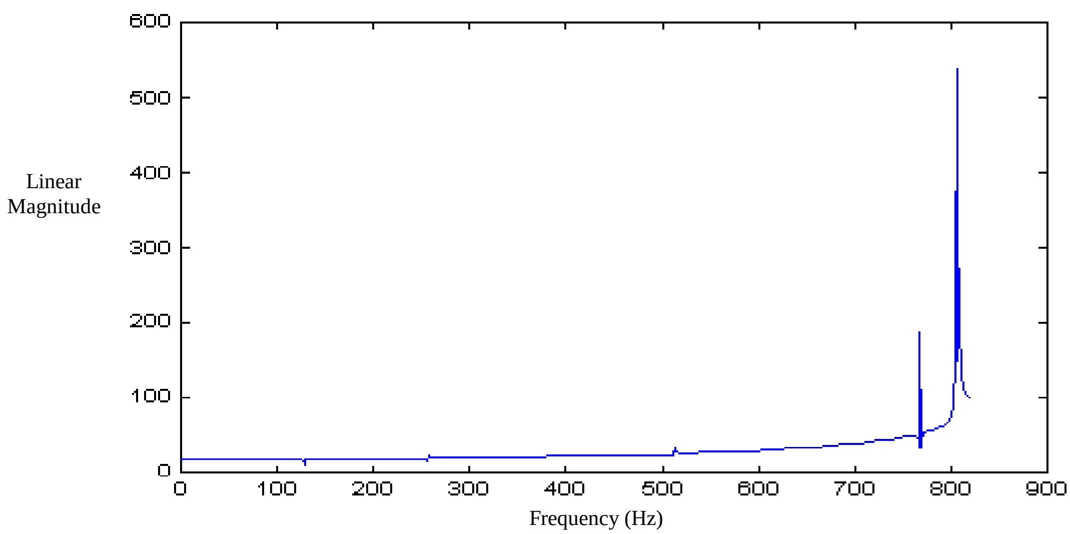


Figure 4.1.13: FRF plot of the analytical curve and ITD curve fit performed to the first simulated data truncated at 820 Hz

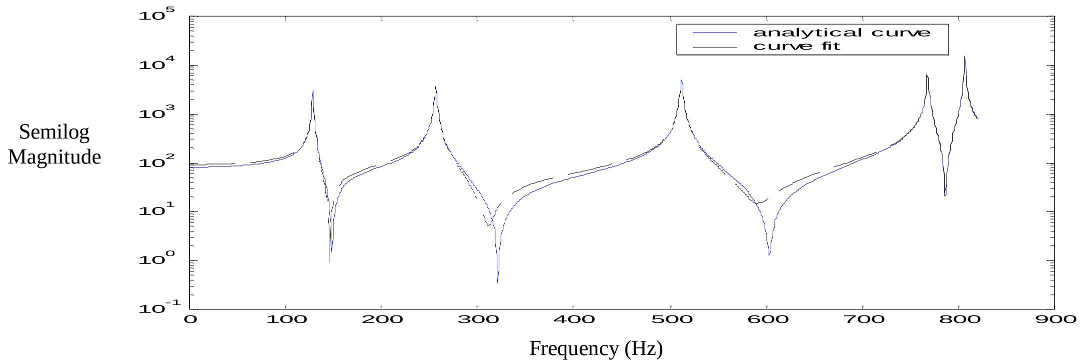


(a) Phase angle plot of the analytical and fitted curve

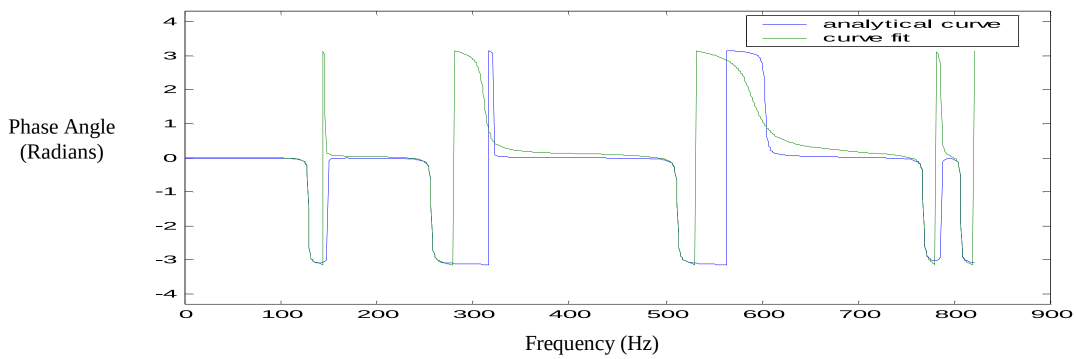


(b) Residual plot

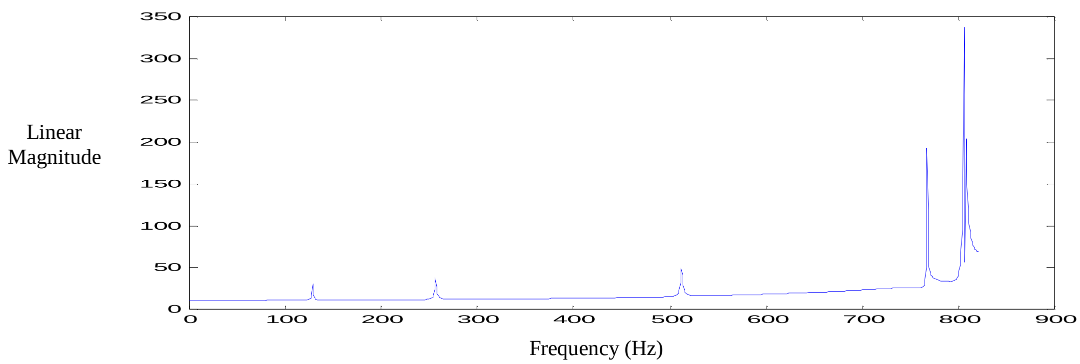
Figure 4.1.14: Phase angle (a) and Residual (b) plots of the analytical curve and CEM curve fit performed to the first simulated data truncated at 820 Hz



(a) FRF plot of the analytical and fitted curve



(b) Phase angle plot of the analytical and fitted curve



(c) Residual plot

Figure 4.1.15: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and ITD curve fit performed to the second simulated data truncated at 820 Hz

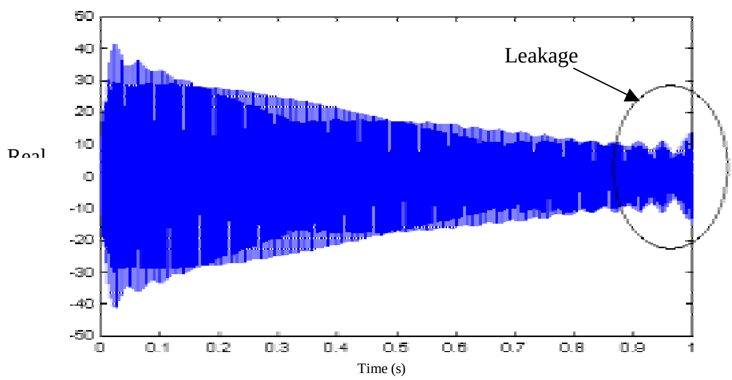
Single Mode Approach

Table 4.1.14: ITD estimated parameters of the first simulated data set in a single-mode approach with an isolating rectangular window of 51 spectral lines

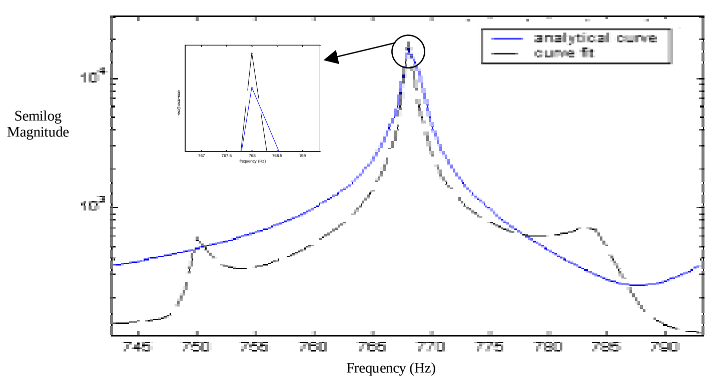
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Standard Deviation (+ - 25 Spectral Lines)
1	127.88059141	0.00100527540	0.52753989352	207.8755
2	255.93965232	0.00074460234	10.64771896814	173.5117
3	512.02215946	0.00000080397	99.82634246698	935.8428
4	767.84983064	0.00009289731	69.90127097531	1093.8693
5	806.57463015	0.00004526414	70.66883801213	1703.2066

Table 4.1.15: ITD estimated parameters of the second simulated data set in a single-mode approach with an isolating rectangular window of 51 spectral lines

Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Standard Deviation (+ - 25 Spectral Lines)
1	128.07348866	0.00340447071	14.88823214028	81.6296
2	255.95591449	0.00391270322	17.38109646774	85.4354
3	516.17320789	0.00022887929	87.64051823927	1135.9111
4	767.69280405	0.00045311581	63.29761966550	782.6443
5	806.67158535	0.00015660812	74.62948502851	1589.6020



(a) Single mode IRF plot



(b) Single mode isolated FRF plot with the analytical and fitted curve

Figure 4.1.16: The 768 Hz natural frequency IRF (a) and FRF plot (b) with the analytical and fitted curve of the CEM single-mode approach of the first simulated data set.

Rational Fraction Polynomial (RFP) Method Estimated Parameters

With this method, the estimated damping ratios are not affected by the truncation in the MDOF and single-mode approach. It gives an accurate and consistent damping ratio estimation; all the damping ratio's percent of error, and the local standard deviation (tables 4.1.16 to 4.1.21) are small. The highest residuals and local standard deviations, for all the cases, occur at the close modes (mode 4 and 5), but these values are still small. Also, if we observe the figures (figures 4.1.17 to 4.1.21), they show that the analytical curve and curve fit match exactly even when the FRF is truncated. The truncation in the FRF doesn't affect this method because it works in the frequency domain and doesn't have to calculate the IRF to estimate the damping ratio.

Multiple Degree of Freedom (MDOF) Approach

Table 4.1.16: RFP estimated parameters in a MDOF approach of the first simulated data set without truncation

Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128.06253066	0.00100000312	0.00031197330	0.0080943	
2	256.12506119	0.00083333536	0.00024299413	0.0056654	
3	512.25012257	0.00046296410	0.00024604482	0.0082612	0.01226093549
4	768.37518338	0.00030864187	0.00003523283	0.0088717	
5	806.79394265	0.00015432060	0.00024884641	0.0063880	

Table 4.1.17: RFP estimated parameters in a MDOF approach of the second simulated data set without truncation

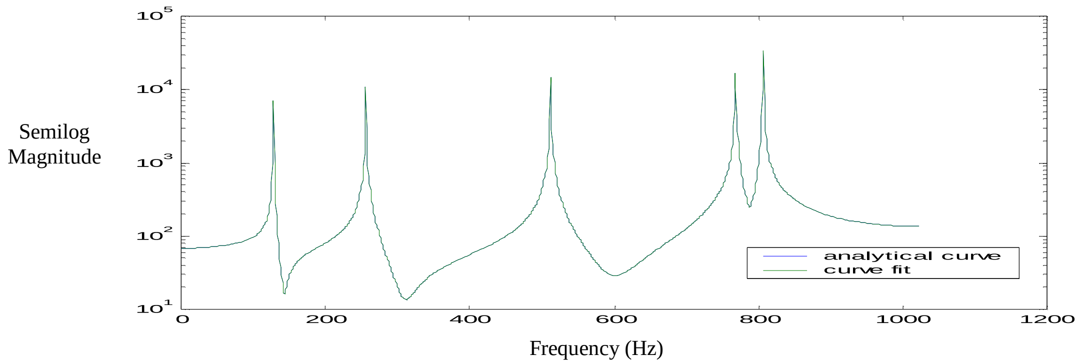
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128.06253298	0.00399997976	0.00050604430	0.0057564	
2	256.12506001	0.00333331683	0.00049524654	0.0074841	
3	512.25011905	0.00185185707	0.00028204966	0.0048564	0.00968554114
4	768.37518720	0.00123456769	0.00001714405	0.0103330	
5	806.79394344	0.00061728340	0.00008883918	0.0090598	

Table 4.1.18: RFP estimated parameters in a MDOF approach of the first simulated data set truncated at 820 Hz

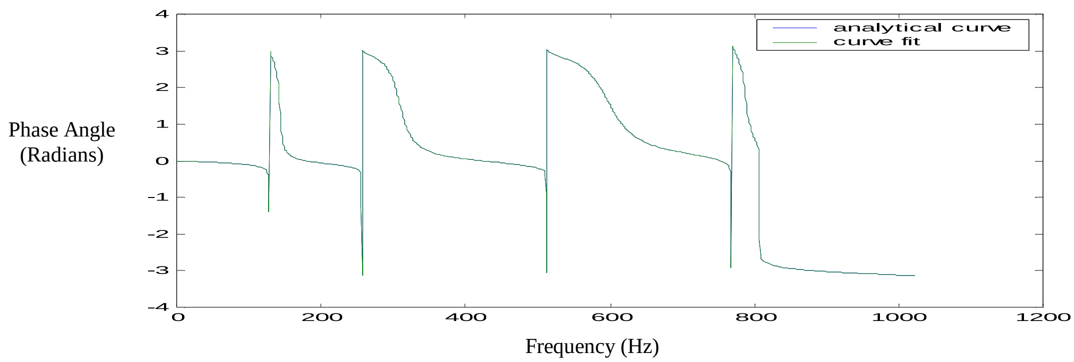
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128.06253113	0.00100000456	0.00045595495	0.0050468	
2	256.12506167	0.00083333440	0.00012810812	0.0056179	
3	512.25012188	0.00046296335	0.00008348859	0.0045485	0.0082805
4	768.37518334	0.00030864149	0.00015770589	0.0051799	
5	806.79394215	0.00015432090	0.00005645518	0.0219100	

Table 4.1.19: RFP estimated parameters in a MDOF approach of the second simulated data set truncated at 820 Hz

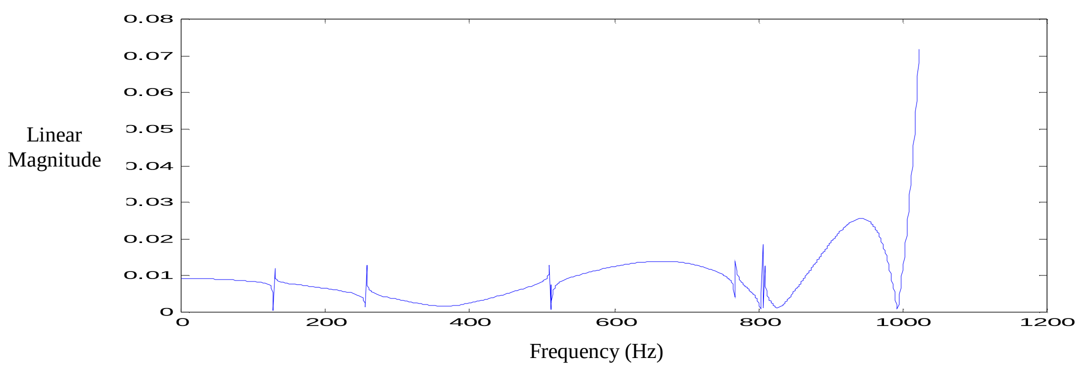
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128.06252989	0.00399997060	0.00073501491	0.0057526	
2	256.12505611	0.00333332977	0.00010702787	0.0069943	
3	512.25012656	0.00185185507	0.00017397622	0.0064013	0.0105746
4	768.37517738	0.00123457120	0.00026713136	0.0099079	
5	806.79394252	0.00061728605	0.0003394010	0.0284890	



(a) FRF plot of the analytical and fitted curve

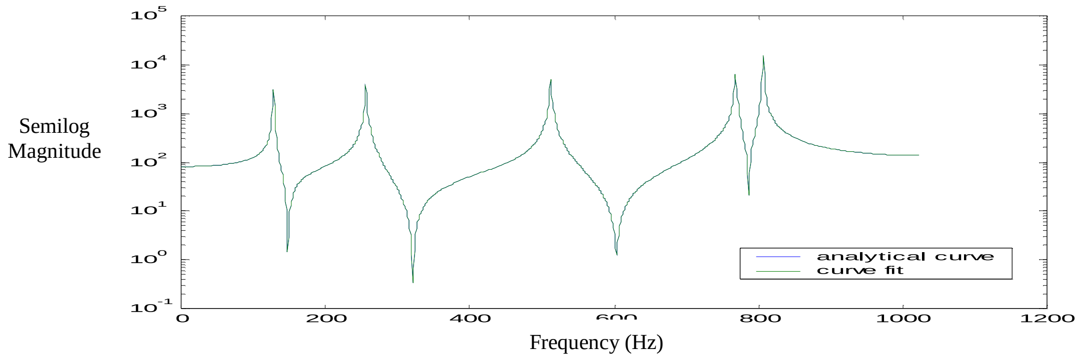


(b) Phase angle plot of the analytical and fitted curve

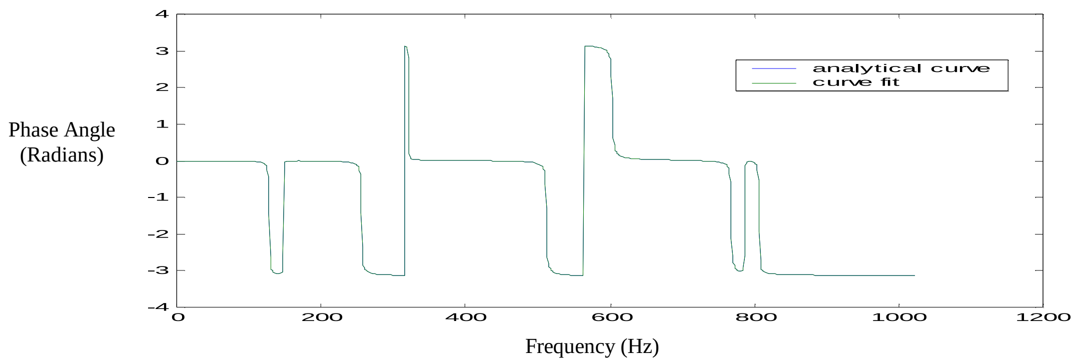


(c) Residual plot

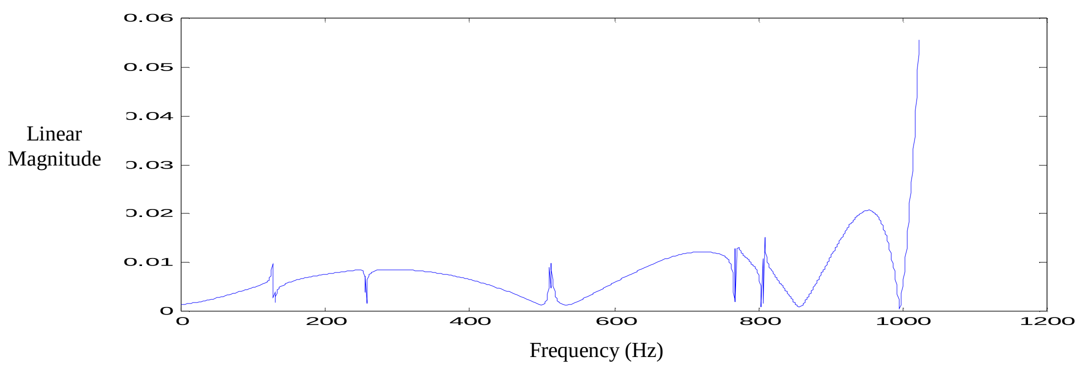
Figure 4.1.17: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and RFP curve fit performed to the first simulated data without truncation



(a) FRF plot of the analytical and fitted curve

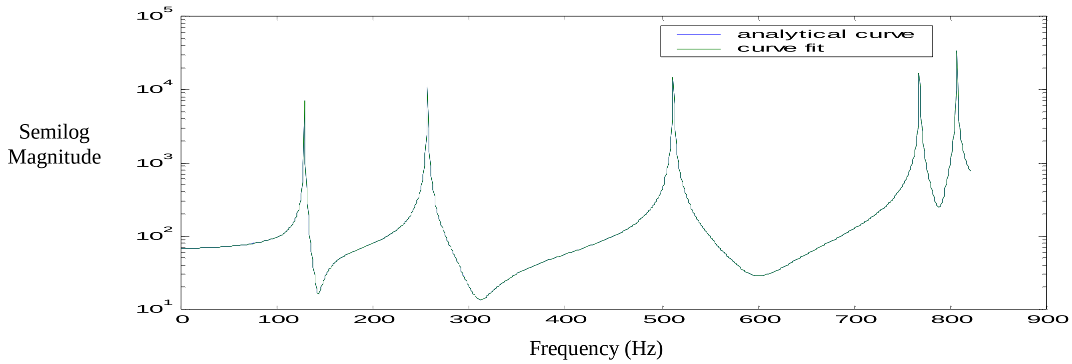


(b) Phase angle plot of the analytical and fitted curve

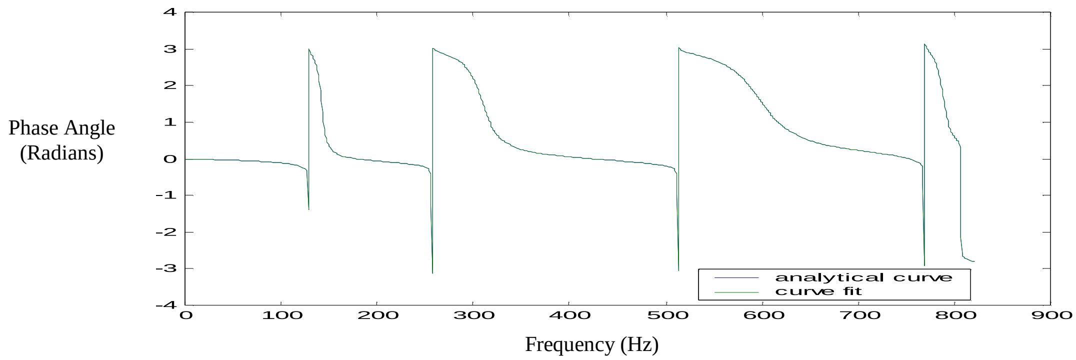


(c) Residual plot

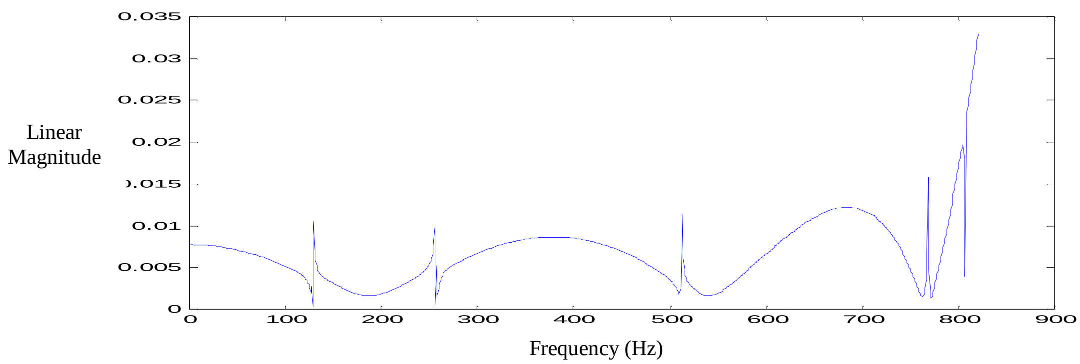
Figure 4.1.18: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and RFP curve fit performed to the second simulated data without truncation



(a) FRF plot of the analytical and fitted curve

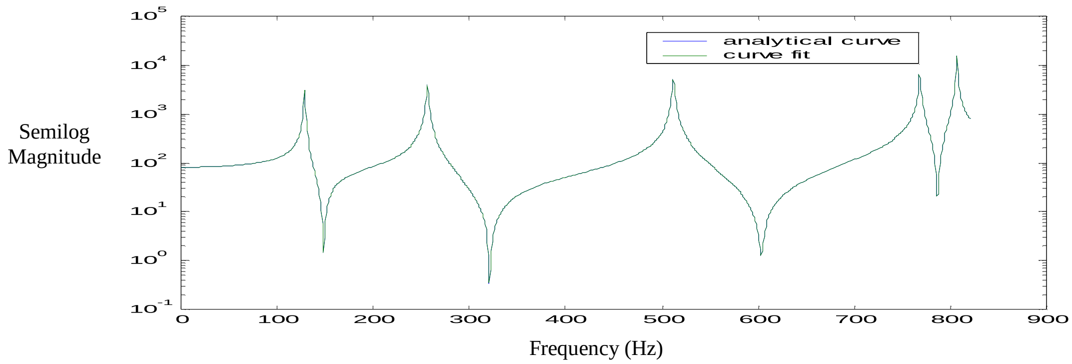


(b) Phase angle plot of the analytical and fitted curve

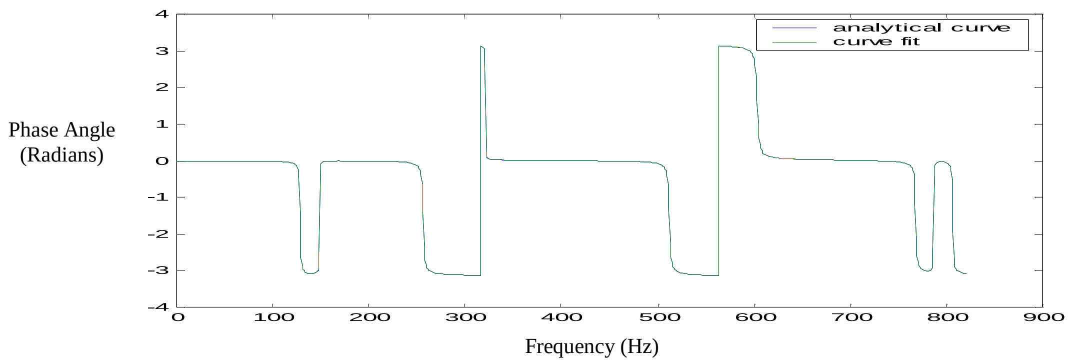


(c) Residual plot

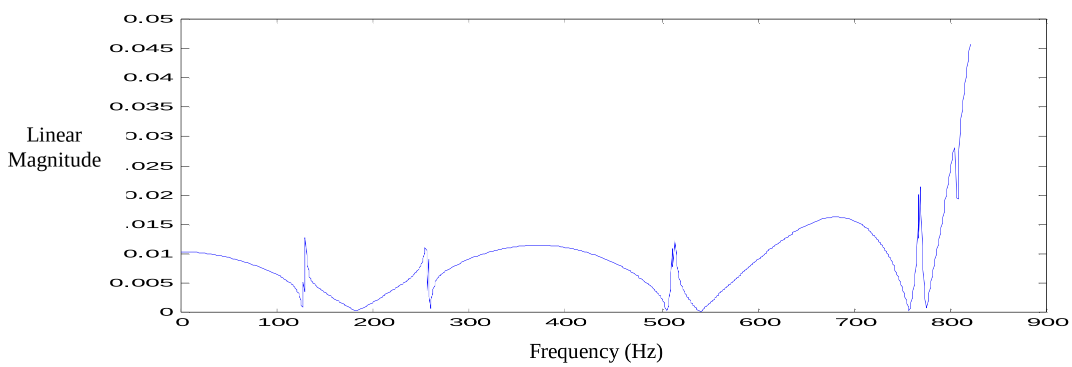
Figure 4.1.19: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and RFP curve fit performed to the first simulated data truncated at 820 Hz



(a) FRF plot of the analytical and fitted curve



(b) Phase angle plot of the analytical and fitted curve



(c) Residual plot

Figure 4.1.20: FRF (a), Phase angle (b) and Residual (c) plots of the analytical curve and RFP curve fit performed to the first simulated data truncated at 820 Hz

Single Mode Approach

Table 4.1.20: RFP estimated parameters of the first simulated data set in a single-mode approach with an isolating rectangular window of 51 spectral lines

Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Standard Deviation (+ - 25 Spectral Lines)
1	128.06253052	0.00099999997	0.00000282990	0.000767
2	256.12506106	0.00083333334	0.00000036829	0.018810
3	512.25012227	0.00046296346	0.00010838284	0.005170
4	768.37518395	0.00030864290	0.00029847349	0.017995
5	806.79394293	0.00015432280	0.00117312011	0.084062

Table 4.1.21: RFP estimated parameters of the second simulated data set in a single-mode approach with an isolating rectangular window of 51 spectral lines

Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Standard Deviation (+ - 25 Spectral Lines)
1	128.06253044	0.00399999968	0.00000791775	0.000322
2	256.12506107	0.00333333332	0.00000041574	0.000004
3	512.25012238	0.00185185233	0.00002602007	0.001303
4	768.37518447	0.00123457176	0.00031221145	0.009317
5	806.79393672	0.00061728402	0.00001116715	0.080318

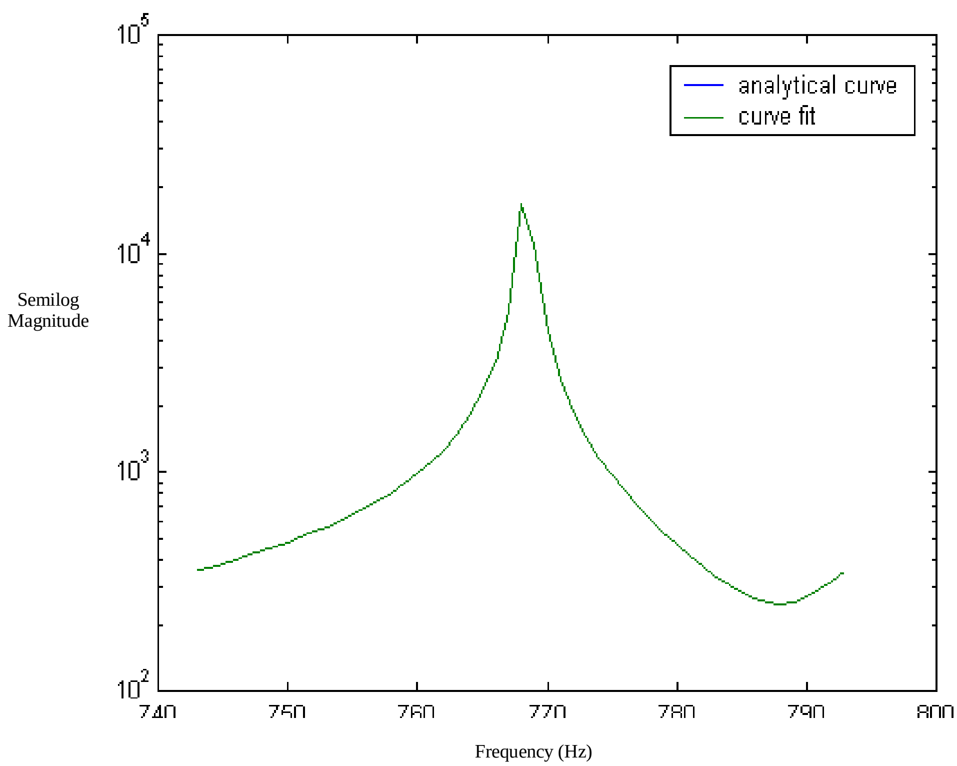


Figure 4.1.21: Single mode isolated FRF plot with the analytical and fitted curve performed with the RFP single-mode approach of the first simulated data set.

Hilbert Envelope Method Estimated Parameters

This method estimates the damping ratio by isolating each mode in the FRF, then plotting the envelope of the single-mode IRF in natural logarithmic scale (see figures 4.1.22 to 4.1.25) in order to obtain a line from where the slope contains the damping ratio value. The damping ratio's percent of error (tables 4.1.22 to 4.1.25) decreases and the IRF envelope plots are better when the spectral frequency range of the isolating rectangular window increases. The first data set has better plots than the second data set because the anti-resonance frequencies of the second data set influence more on each mode because its magnitude is lower. Also, the plots are worst when there are close modes (mode 4 and 5). This means that the estimated damping ratio accuracy is affected when the mode has a close mode or a low magnitude anti-resonance frequency, or both.

Single Mode Approach

Table 4.1.22: Hilbert Envelope Method estimated parameters of the first simulated data set with an isolating rectangular window of 51 spectral lines

Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128	0.00100026103	0.02610326674	6.7698	
2	256	0.00083185355	0.17757421532	8.0564	
3	512	0.00046275518	0.04488005193	12.5110	714.5111004
4	768	0.00030878784	0.04726027649	88.7805	
5	806.4	0.00015307492	0.80745330489	4984.4902	

Table 4.1.23: Hilbert Envelope Method estimated parameters of the second simulated data set with an isolating rectangular window of 51 spectral lines

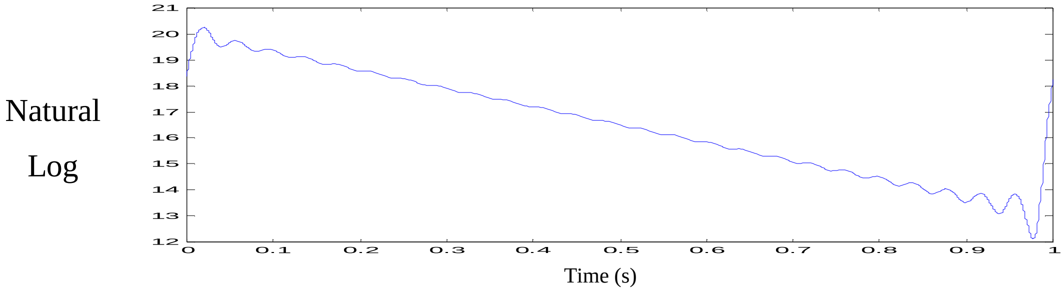
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128	0.00399272221	0.18194470630	7.2857000	
2	256	0.00332687640	0.19370797934	8.2854000	
3	512	0.00184572569	0.33081251600	13.2535000	236.7294818
4	768	0.00122962328	0.40051471297	76.1895000	
5	806.4	0.00061685640	0.06926328159	1635.2180000	

Table 4.1.24: Hilbert Envelope Method estimated parameters of the first simulated data set with an isolating rectangular window of 21 spectral lines

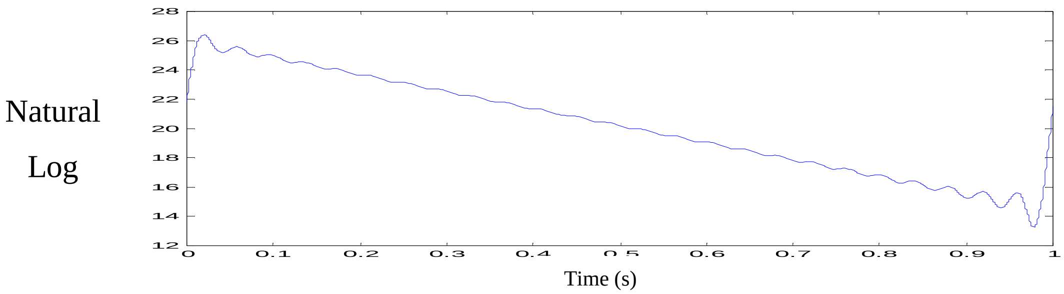
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128	0.00098722759	1.27724085383	7.8668000	
2	256	0.00082295677	1.24518769691	13.7461000	
3	512	0.00045916922	0.81944852009	18.0788000	715.7985333
4	768	0.00030409868	1.47202742503	98.0368000	
5	806.4	0.00014821987	3.95352476975	4993.3044000	

Table 4.1.25: Hilbert Envelope Method estimated parameters of the second simulated data set with an isolating rectangular window of 21 spectral lines

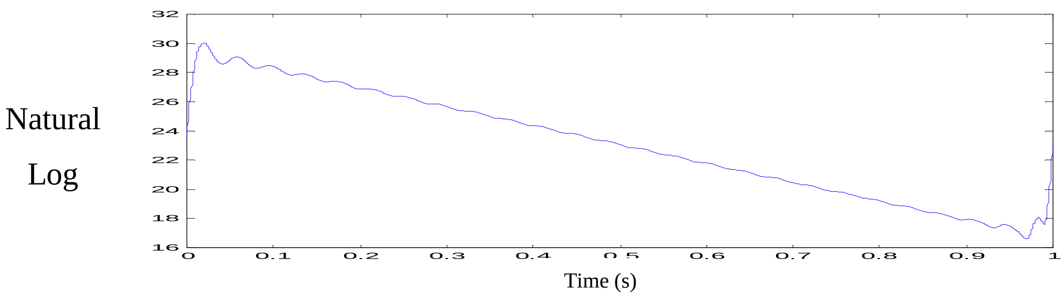
Mode	Natural Frequency (Hz)	Damping Ratio	Damping Ratio's Percent of Error (%)	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
1	128	0.00390492491	2.37687726144	12.0840000	
2	256	0.00319743558	4.07693260897	29.7805000	
3	512	0.00180192173	2.69622674009	31.4477000	238.6429083
4	768	0.00120929493	2.04711079885	86.3498000	
5	806.4	0.00060590208	1.8438629953	1647.4846000	



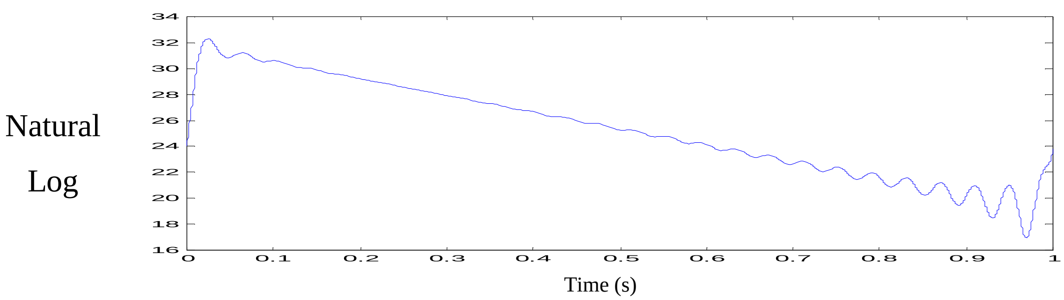
(a) 128 Hz



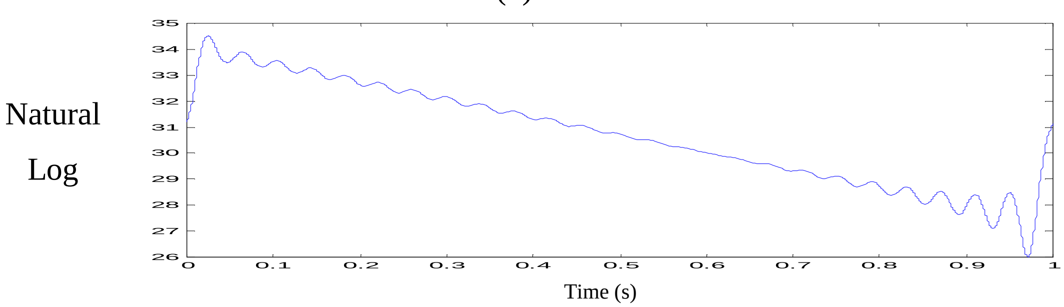
(b) 256 Hz



(c) 512 Hz

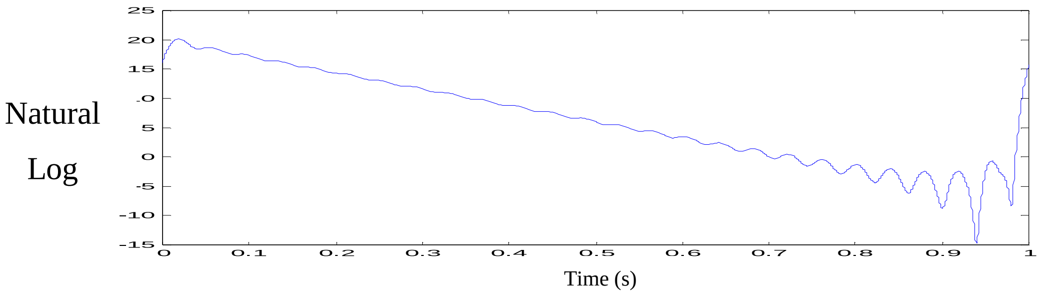


(d) 768 Hz

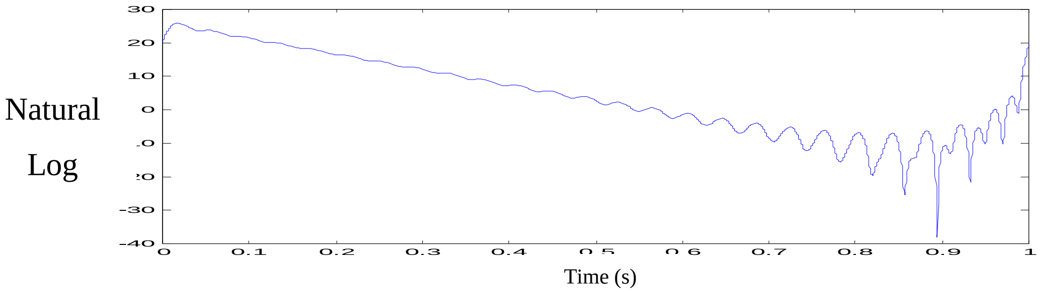


(e) 806 Hz

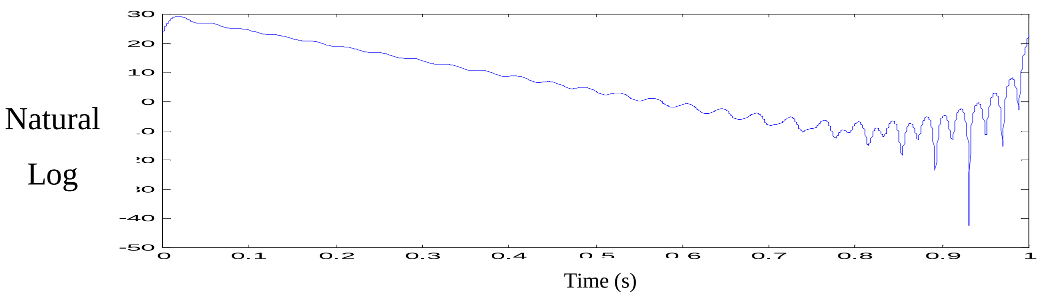
Figure 4.1.22: IRF envelope of the first data with isolating window of 51 spectral lines



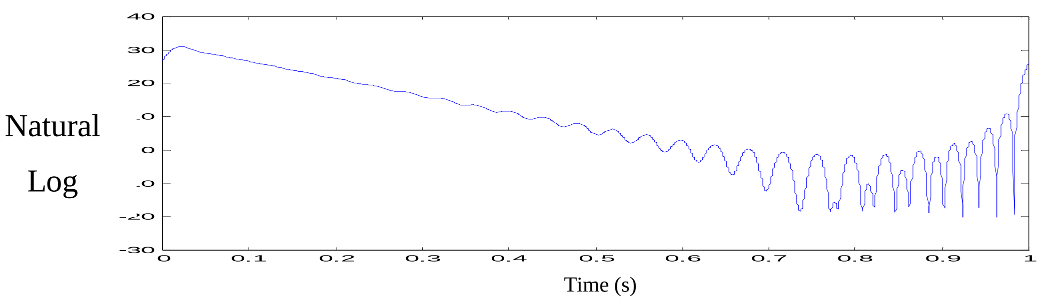
(a) 128 Hz



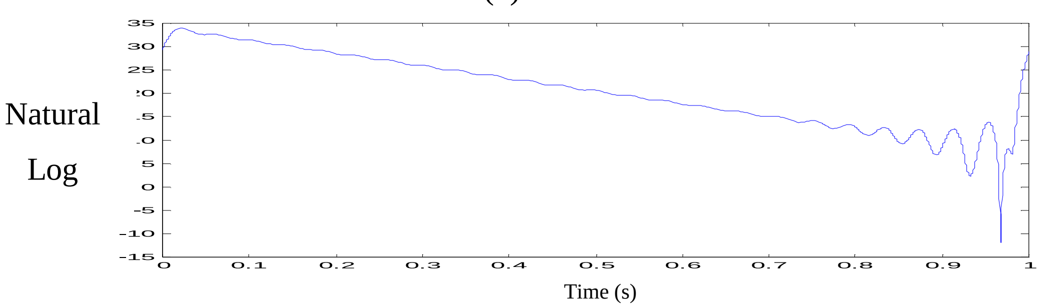
(b) 256



(c) 512

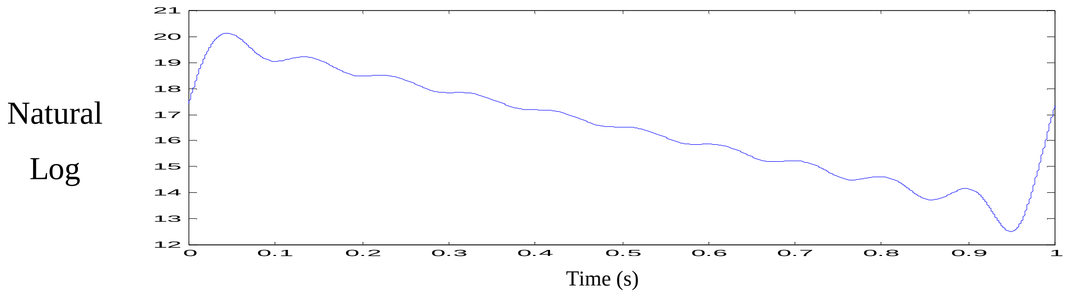


(d) 768 Hz

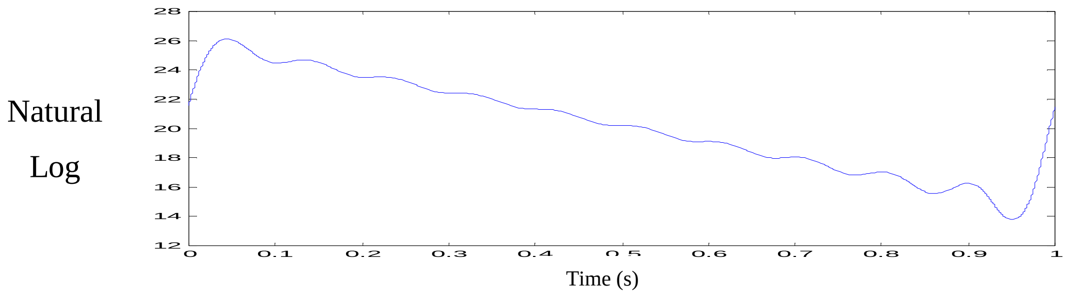


(e) 806 Hz

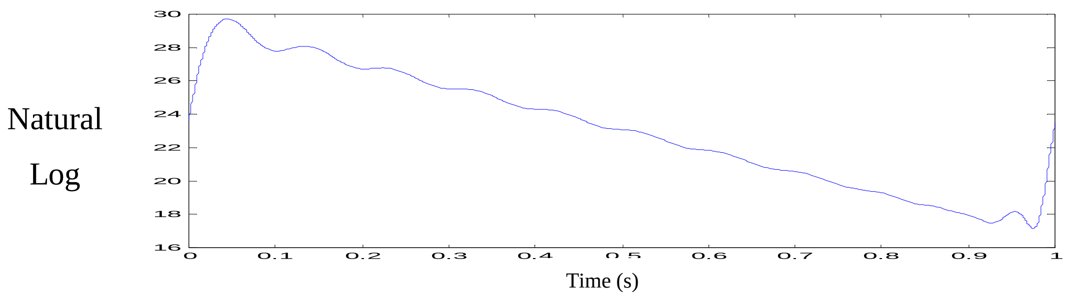
Figure 4.1.23: IRF envelope of the second data with isolating window of 51 spectral lines



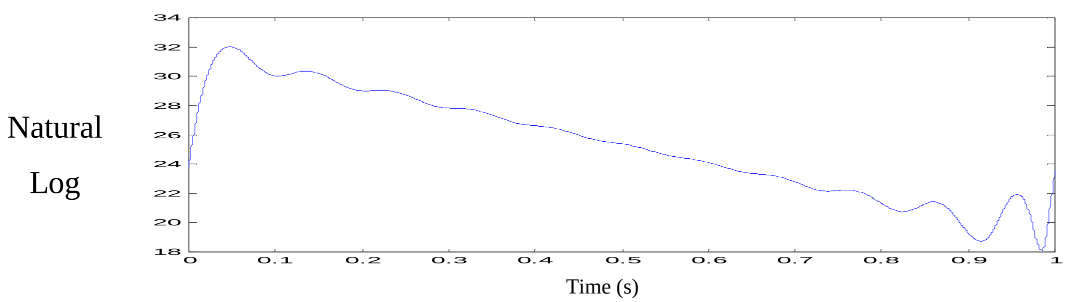
(a) 128 Hz



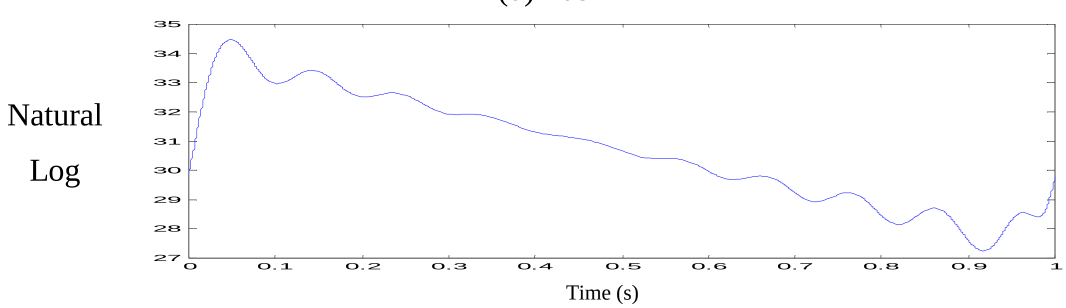
(b) 256 Hz



(c) 512 Hz

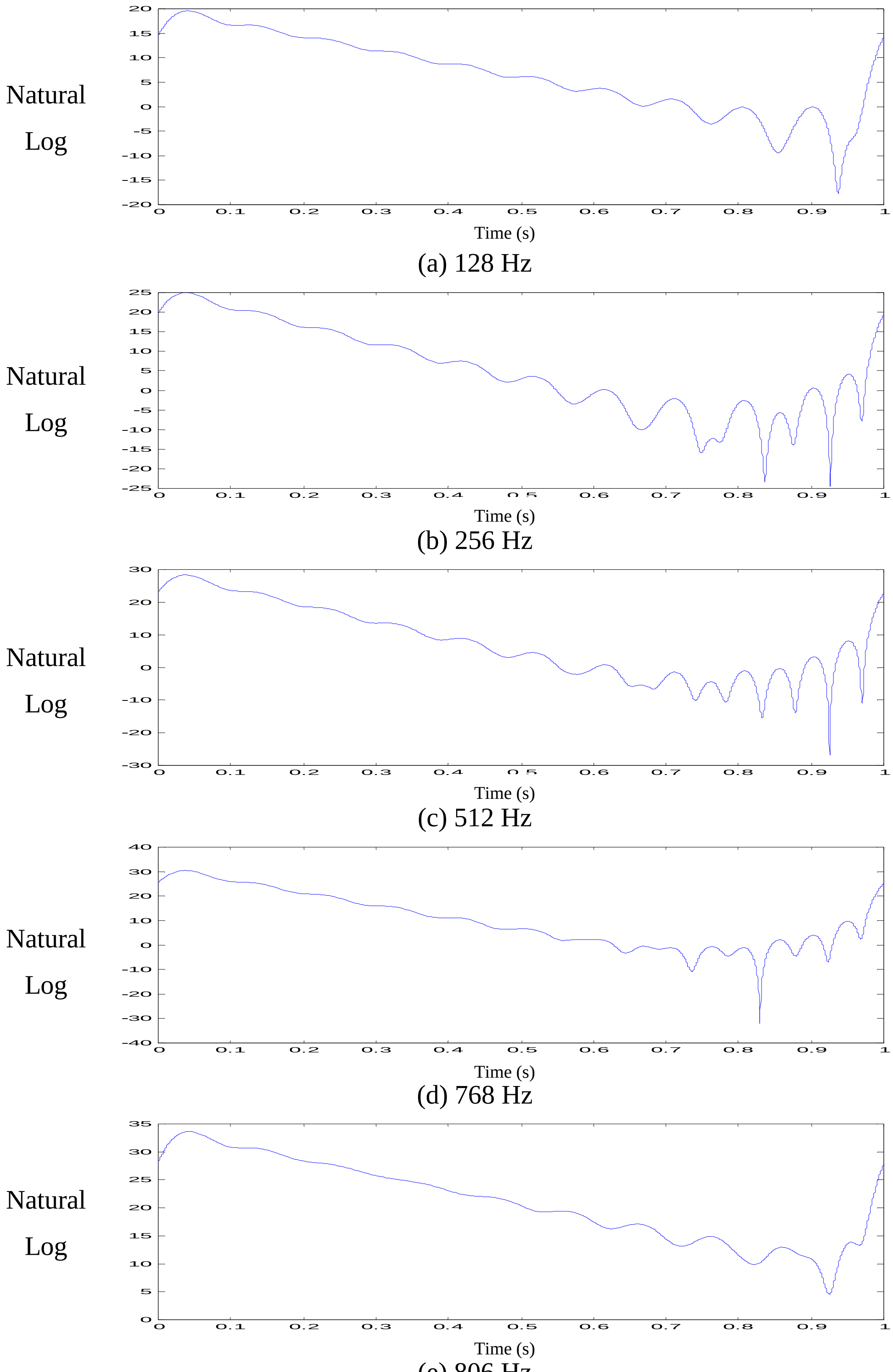


(d) 768 Hz



(e) 806 Hz

Figure 4.1.24: IRF envelope of the first data with isolating window of 21 spectral lines



(a) 128 Hz

(b) 256 Hz

(c) 512 Hz

(d) 768 Hz

(e) 806 Hz

Figure 4.1.25: IRF envelope of the second data with isolating window of 21 spectral lines

4.2 Experimental Data

This section presents the estimated values from the T-plate's experimental data. These values are separated by output measurement location point. Each section has each method's estimated parameters organized by tables and then the FRF, phase angle, and IRF plots are shown in regard to the structure's output measurement point. The tables in this section present the values of the estimated natural frequency, damping ratio, and the local and global standard deviation. In comparison to the previous section's tables, the following tables don't have the damping ratio's percent of error because the exact damping ratio value cannot be calculated for real structures. In the other hand, the residual and standard deviations are calculated in order to validate the estimated modal parameters.

At the end of this section, figures 4.2.7(a) and 4.2.7(b) show the best and worst coherence plots, respectively, are shown in order to demonstrate that this is good data.

When the estimated damping ratios for each location with respect to the natural frequency were inconsistent among the methods was because there were close modes, or close anti-resonance frequency with or without low magnitude, or both. The Hilbert Envelope Method was affected more by the anti-resonance frequency with low magnitude even if it was not close to the mode, but it affected more the closest mode to it. Also, if there was a mode near the maximum frequency of the FRF, the CEM and ITD method's estimated damping ratio were affected because of the truncation (223 truncated spectral lines). But, with all these problems, the RFP method always had the lowest local standard deviation among the methods with respect to each mode, and the lowest global standard deviation for each FRF's output measurement location point processed. There is no way to know which is the most exact estimated damping ratio, but residual and standard deviations are good guides.

First Location

Table 4.2.1: CEM estimated parameters on the structure's first output measurement location point

Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
139.39503567359	0.00072912177	0.052026	
252.74218887631	0.00122082717	0.055784	
500.83039727767	0.00217666487	0.170240	0.111842071
525.32979623439	0.00066968354	0.265570	
661.06111696236	0.00332874117	0.093230	
776.10716234617	0.00262571494	0.369090	

Table 4.2.2: ITD estimated parameters on the structure's first output measurement location point

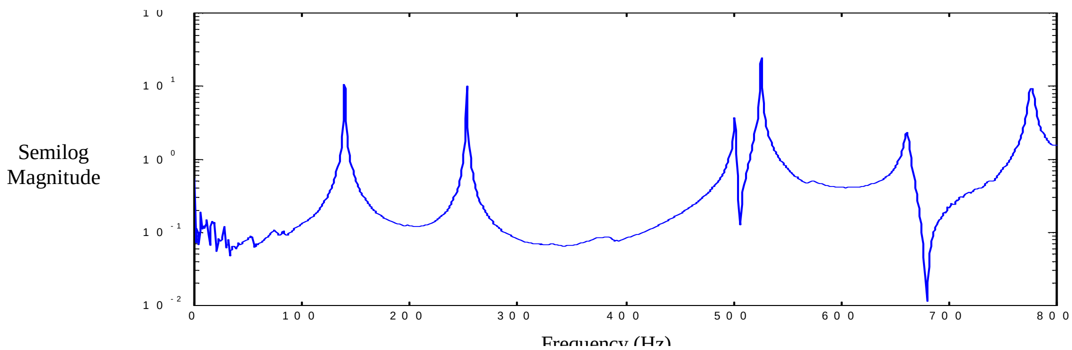
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
139.41319857252	0.00078111379	0.153500	
252.73217116949	0.00122367752	0.086380	
500.69207435945	0.00144415706	0.274510	0.345771862
525.35611343718	0.00067920729	0.451360	
661.08343317263	0.00439011067	0.113180	
775.94567248597	0.00139384631	1.840600	

Table 4.2.3: RFP estimated parameters on the structure's first output measurement location point

Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
139.46013352719	0.00070780397	0.040979	
252.87972317807	0.00121675344	0.026440	
501.23962448564	0.00182051091	0.042104	0.037435634
525.56899714450	0.00070743485	0.057459	
661.18542057729	0.00362327753	0.034222	
776.47223507694	0.00299321968	0.104210	

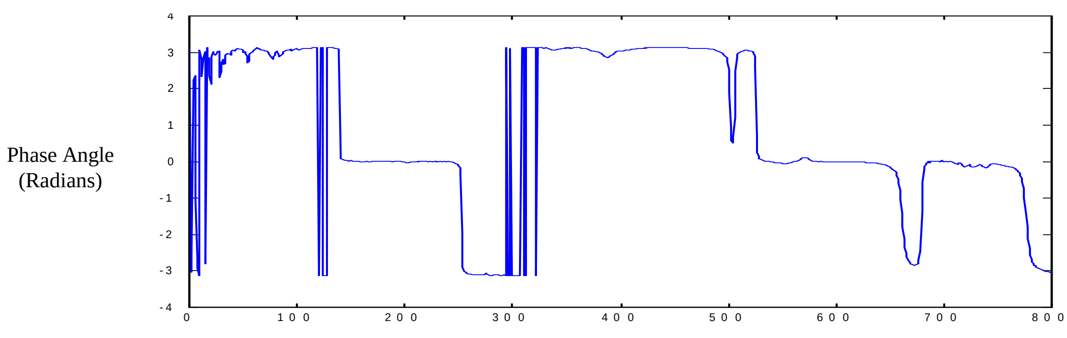
Table 4.2.4: Hilbert Envelope Method estimated parameters on the structure's first output measurement location point

Natural Frequency (Hz)	Damping Ratio	Isolating Window Range	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
139	0.00069532596	25	2.097700	
253	0.00117389618	25	0.786700	
501	0.00177620338	10	0.476120	0.958119229
526	0.00070516143	20	5.342200	
661	0.00378268379	20	0.121200	
776	0.00303534773	20	0.274580	



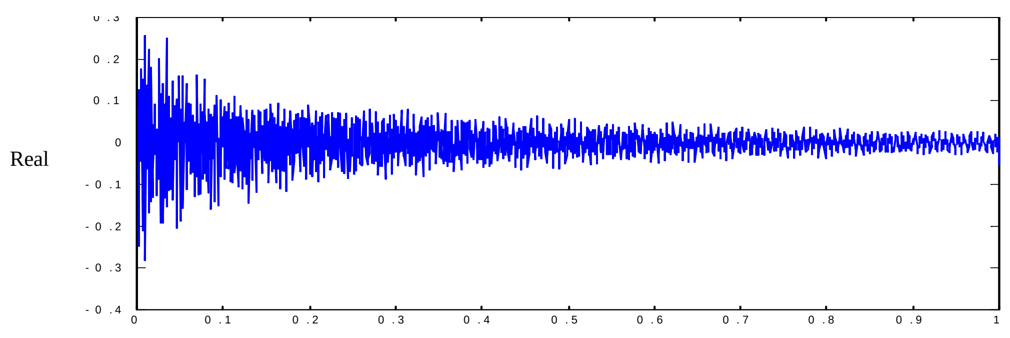
Frequency (Hz)

(a) FRF plot



Frequency (Hz)

(b) Phase angle plot



Time (s)

(c) IRF plot calculated from the FRF

Figure 4.2.1: FRF (a), Phase angle (b) and IRF (c) of the structure's first output measurement location point

Second Location

Table 4.2.5: CEM estimated parameters on the structure's second output measurement location point

		Local Standard Deviation	Global
Natural Frequency (Hz)	Damping Ratio	(+ - 10 Spectral Lines)	Standard Deviation
139.39999469361	0.00097514020	0.081321	
251.87709312619	0.00231908200	0.067059	
500.89289336951	0.00778637708	0.116840	0.088579715
525.04626509145	0.00091282392	0.266490	
661.55290497059	0.00200344168	0.037994	
772.75126405374	0.00555592381	0.076129	

Table 4.2.6: ITD estimated parameters on the structure's second output measurement location point

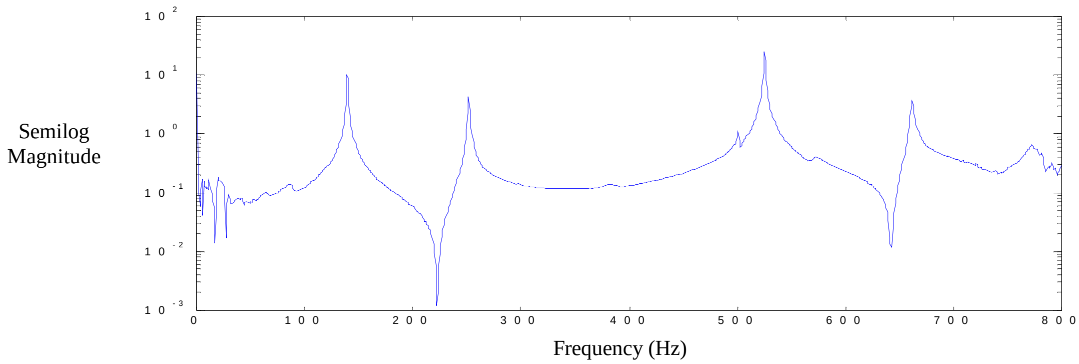
		Local Standard Deviation	Global
Natural Frequency (Hz)	Damping Ratio	(+ - 10 Spectral Lines)	Standard Deviation
139.41831042927	0.00106358899	0.170830	
251.90870813348	0.00241962699	0.047591	
514.90082562075	0.00297033338	0.274750	0.477102965
524.91869285102	0.00071001782	1.629100	
661.24510288169	0.00231360912	0.157110	
772.92162727152	0.00208557423	0.256850	

Table 4.2.7: RFP estimated parameters on the structure's second output measurement location point

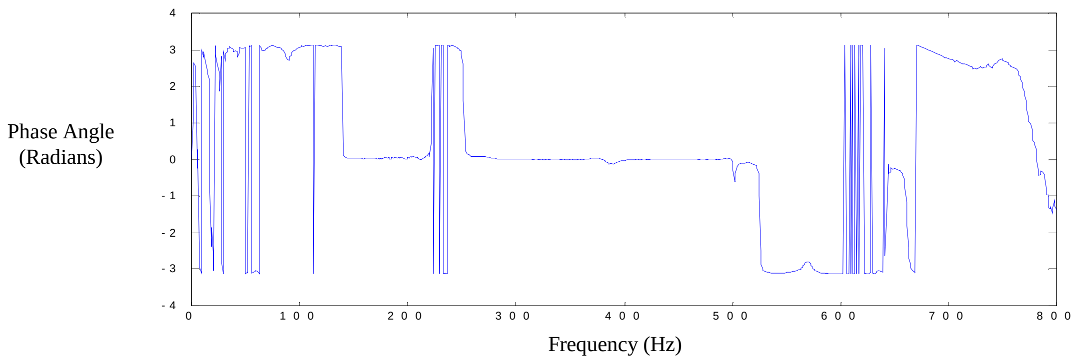
		Local Standard Deviation	Global
Natural Frequency (Hz)	Damping Ratio	(+ - 10 Spectral Lines)	Standard Deviation
139.46908267781	0.00089399332	0.081321	
252.02851621968	0.00243603222	0.067059	
501.18754232972	0.00182476484	0.116840	0.065656224
525.32738302445	0.00095031818	0.266490	
661.46913464390	0.00216705481	0.037994	
774.66259251183	0.00766892357	0.076129	

Table 4.2.8: Hilbert Envelope Method estimated parameters on the structure's second output measurement location point

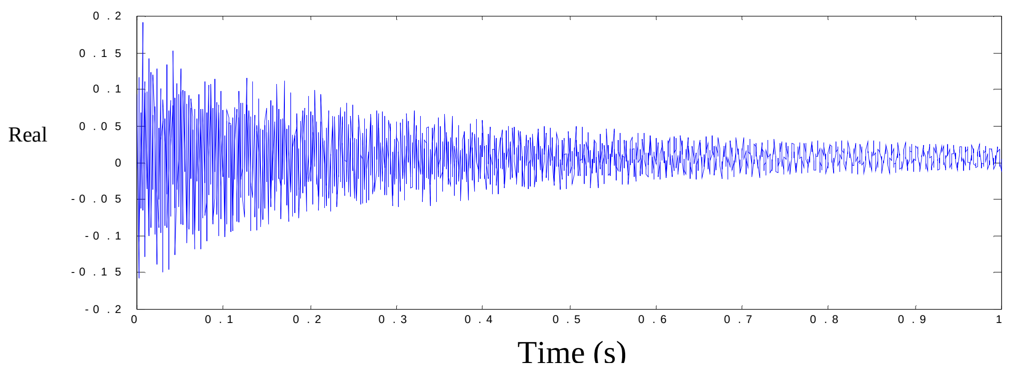
		Isolating Window	Local Standard Deviation	Global
Natural Frequency (Hz)	Damping Ratio	Range	(+ - 10 Spectral Lines)	Standard Deviation
139	0.00069532596	25	2.006100	
252	0.00117389618	25	0.105230	
501	0.00177620338	10	0.054455	0.530361913
525	0.00070516143	15	0.563520	
662	0.00378268379	20	0.249140	
773	0.00303534773	20	0.541450	



(a) FRF plot



(b) Phase angle plot



(c) IRF plot calculated from the FRF

Figure 4.2.2: FRF (a), Phase angle (b) and IRF (c) of the structure's second output measurement location point

Third Location

Table 4.2.9: CEM estimated parameters on the structure's third output measurement location point

Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
140.98082842701	0.00050980488	0.114290	
260.96693663261	0.00034922231	0.116320	
499.33967427463	0.00191484902	0.242750	0.585593808
520.47924775275	0.00101380046	0.563710	
665.80302400246	0.00086911748	0.321210	

Table 4.2.10: ITD estimated parameters on the structure's third output measurement location point

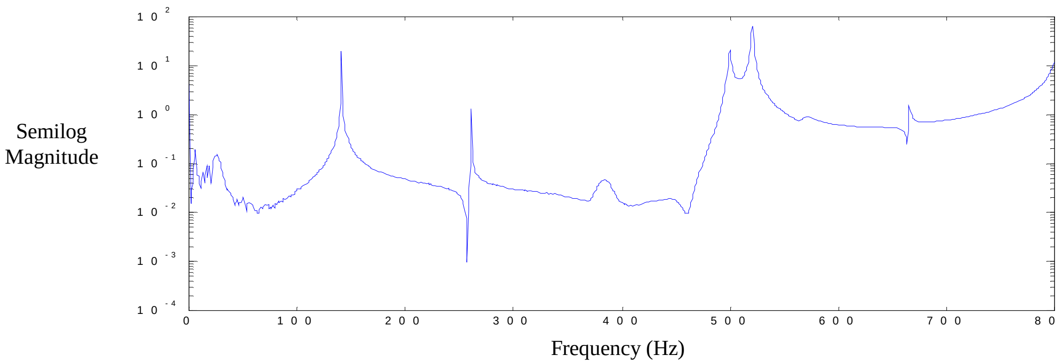
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
140.98816591457	0.00049930618	0.076221	
260.90416590858	0.00028797381	0.113690	
499.72285691875	0.00150856110	2.039300	0.961773176
520.53877601790	0.00079538578	2.959800	
655.48537491167	0.01118129887	0.362160	

Table 4.2.11: RFP estimated parameters on the structure's third output measurement location point

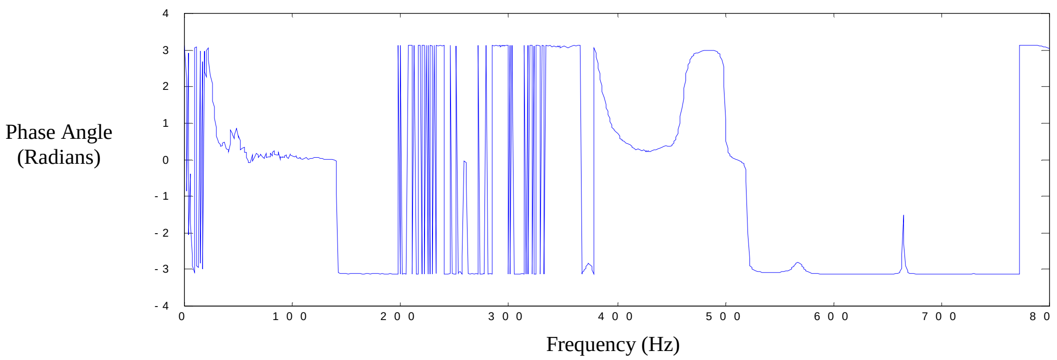
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
141.05364000729	0.00051429893	0.011078	
261.08397682728	0.00021825649	0.012388	
499.55062287839	0.00189527446	0.150660	0.041189462
520.70774991386	0.00102213710	0.160820	
665.76254502648	0.00090237728	0.004652	

Table 4.2.12: Hilbert Envelope Method estimated parameters on the structure's third output measurement location point

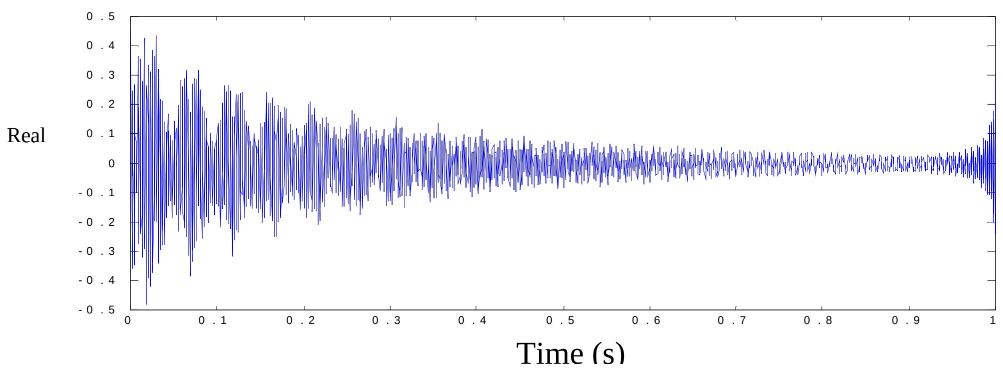
Natural Frequency (Hz)	Damping Ratio	Isolating Window Range	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
141	0.00051272141	25	0.116380	
261	0.00021808858	25	0.048281	
500	0.00198531784	10	2.830200	1.892790416
521	0.00100578354	15	9.188700	
666	0.00100836288	20	0.508210	



(a) FRF plot



(b) Phase angle plot



(c) IRF plot calculated from the FRF

Figure 4.2.3: FRF (a), Phase angle (b) and IRF (c) of the structure’s third output measurement location point

Fourth Location

Table 4.2.13: CEM estimated parameters on the structure's fourth output measurement location point

Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
140.98690429174	0.00067752341	0.100780	
260.95093431720	0.00029059256	0.110580	
497.46419850365	0.00163717371	0.265090	0.634951744
522.55998851097	0.00106541522	0.492490	
665.35896166737	0.00086185004	0.308880	

Table 4.2.14: ITD estimated parameters on the structure's fourth output measurement location point

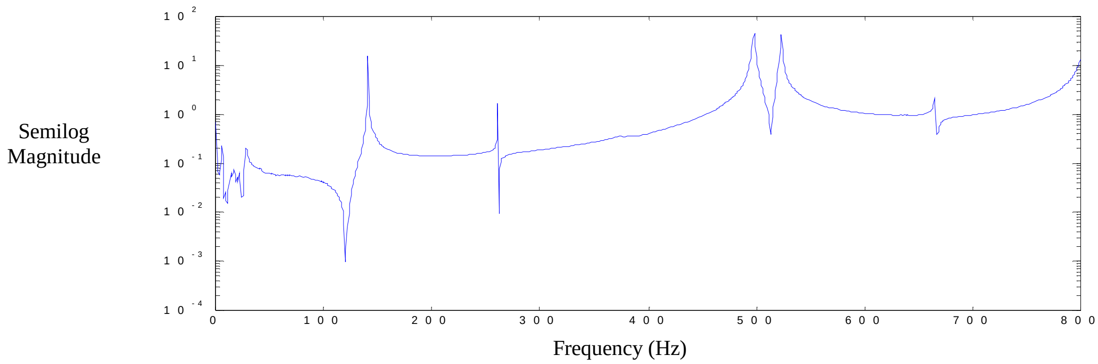
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
140.98288879466	0.00069159171	0.050243	
260.97133553794	0.00001098022	0.043899	
497.44392305449	0.00163506052	0.386700	0.492887238
522.57763864120	0.00103422581	0.539330	
666.00918378415	0.00046299209	0.355390	

Table 4.2.15: RFP estimated parameters on the structure's fourth output measurement location point

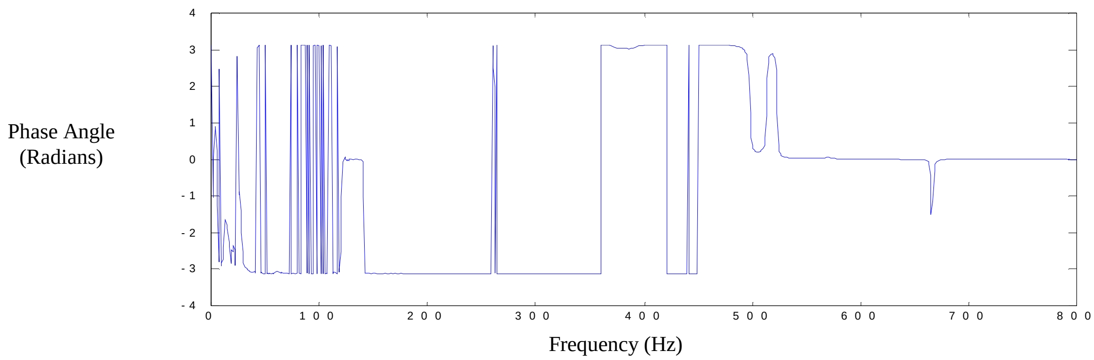
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
141.05946645125	0.00067996340	0.019277	
261.07577466861	0.00023893859	0.003368	
497.72048939382	0.00161745018	0.213130	0.065115797
522.84354899954	0.00110829922	0.275280	
665.67822966886	0.00090249448	0.004390	

Table 4.2.16: Hilbert Envelope Method estimated parameters on the structure's fourth output measurement location point

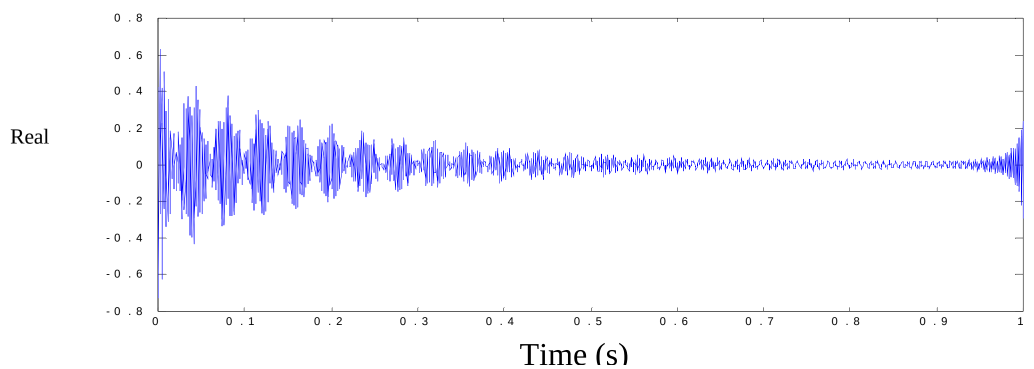
Natural Frequency (Hz)	Damping Ratio	Isolating Window Range	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
141	0.00067304468	25	0.076372	
261	0.00026974090	25	0.086058	
498	0.00164534078	10	4.737800	1.540290298
523	0.00109769388	15	4.191600	
665	0.00094483935	20	0.482340	



(a) FRF plot



(b) Phase angle plot



(c) IRF plot calculated from the FRF

Figure 4.2.4: FRF (a), Phase angle (b) and IRF (c) of the structure's fourth output measurement location point

Fifth Location

Table 4.2.17: CEM estimated parameters on the structure's fifth output measurement location point

Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
140.96309880351	0.00050317835	0.136680	
260.93695017610	0.00016255225	0.132960	
499.45110322944	0.00162551180	0.361230	0.553379565
519.84382254393	0.00087391690	0.934380	
665.61312883428	0.00112139427	0.348870	

Table 4.2.18: ITD estimated parameters on the structure's fifth output measurement location point

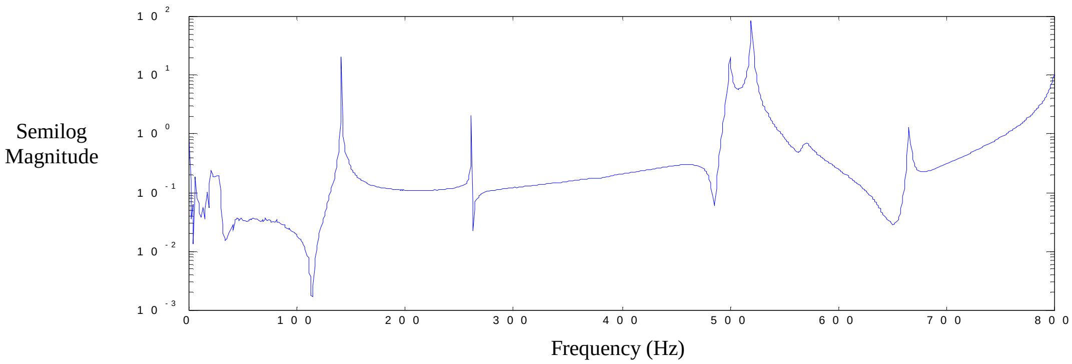
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
140.96343477894	0.00055898578	0.042682	
260.93329565599	0.00016212660	0.020593	
499.17354862137	0.00234380842	1.326500	0.861828916
519.86666802659	0.00133672979	4.159800	
661.80411254112	0.00859471412	0.430570	

Table 4.2.19: RFP estimated parameters on the structure's fifth output measurement location point

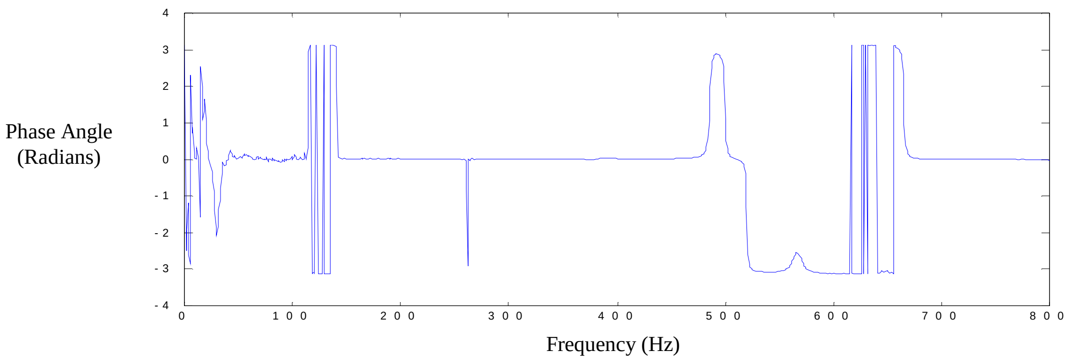
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
141.03620398058	0.00053654705	0.019767	
261.02256817507	0.00017655619	0.332920	
499.65966346407	0.00177351979	0.097540	0.188883354
520.11960985140	0.00092999996	0.251810	
665.60746814208	0.00090428020	0.012650	

Table 4.2.20: Hilbert Envelope Method estimated parameters on the structure's fifth output measurement location point

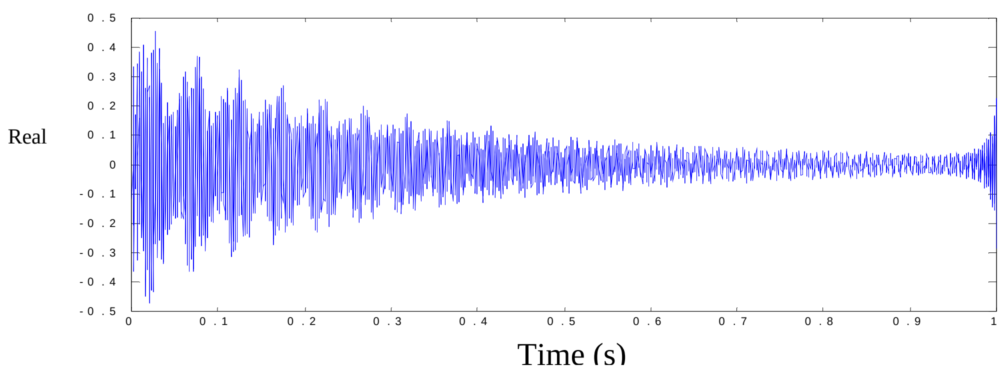
Natural Frequency (Hz)	Damping Ratio	Isolating Window Range	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
141	0.00054126029	25	0.247140	
261	0.00026968167	25	0.062160	
500	0.00168864209	10	2.410100	1.115060674
520	0.00095191949	15	3.243200	
666	0.00090971216	20	0.373090	



(a) FRF plot



(b) Phase angle plot



(c) IRF plot calculated from the FRF

Figure 4.2.5: FRF (a), Phase angle (b) and IRF (c) of the structure's fifth output measurement location point

Sixth Location

Table 4.2.21: CEM estimated parameters on the structure's sixth output measurement location point

Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
140.95281110473	0.00074491825	0.207880	
260.92596422559	0.00016143674	0.228590	
497.12839919202	0.00213572781	0.421090	0.587366539
522.12643740060	0.00118793341	0.502700	
665.21500992041	0.00068536346	0.497790	

Table 4.2.22: ITD estimated parameters on the structure's sixth output measurement location point

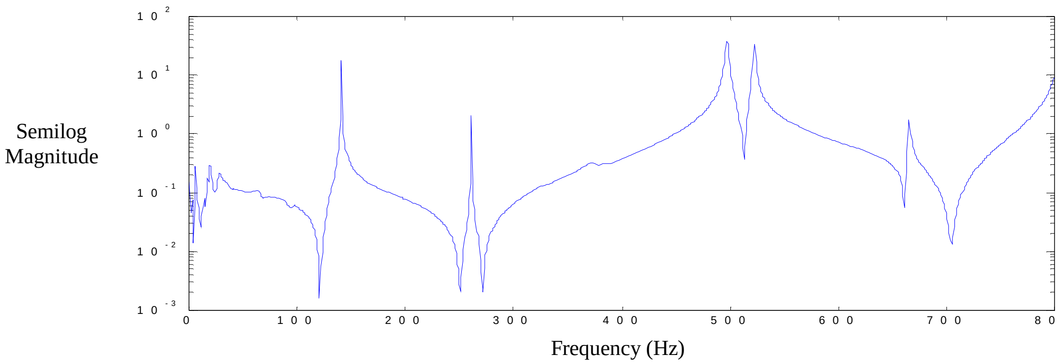
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
140.97791192576	0.00048545079	0.307670	
260.41520643888	0.00015683507	0.366760	
497.85414106582	0.00329219828	5.030200	1.502781877
521.38474284828	0.00225193450	5.995400	
666.03145705058	0.00361548407	0.661970	

Table 4.2.23: RFP estimated parameters on the structure's sixth output measurement location point

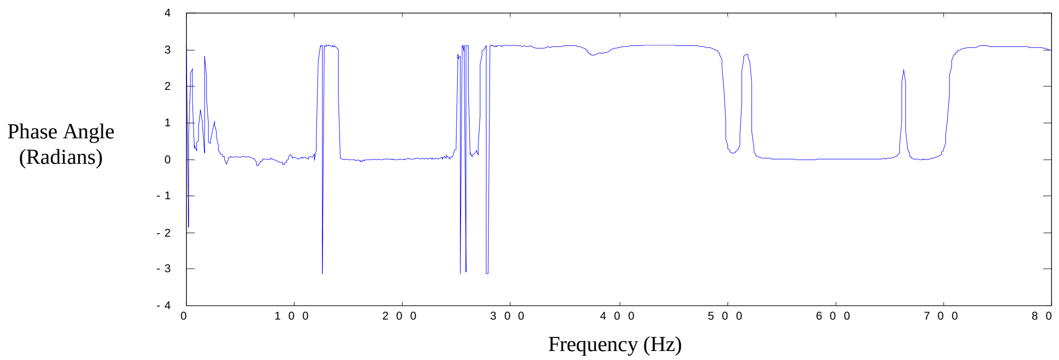
Natural Frequency (Hz)	Damping Ratio	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
141.02459255842	0.00072246544	0.012537	
261.03524431840	0.00024821459	0.005264	
497.37760526718	0.00207223997	0.240980	0.056263602
522.41137483553	0.00119362831	0.184150	
665.53410212401	0.00095570468	0.007291	

Table 4.2.24: Hilbert Envelope Method estimated parameters on the structure's sixth output measurement location point

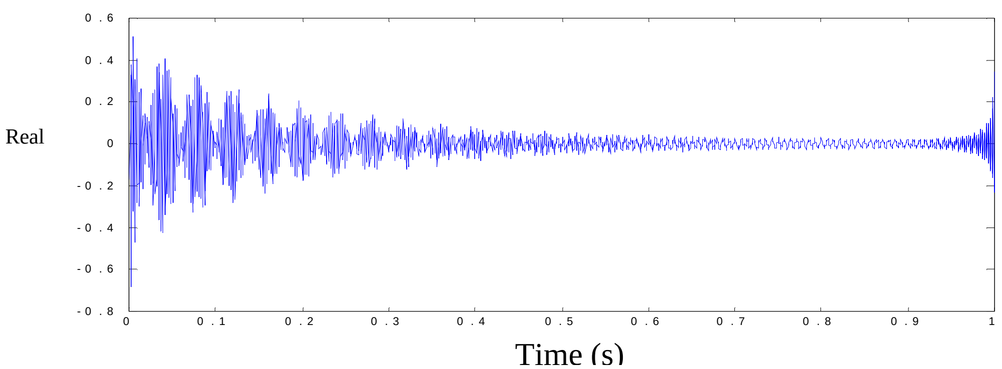
Natural Frequency (Hz)	Damping Ratio	Isolating Window Range	Local Standard Deviation (+ - 10 Spectral Lines)	Global Standard Deviation
141	0.00054126029	25	0.247140	
261	0.00026968167	25	0.062160	
500	0.00168864209	10	2.410100	1.115060674
520	0.00095191949	15	3.243200	
666	0.00090971216	20	0.373090	



(a) FRF plot



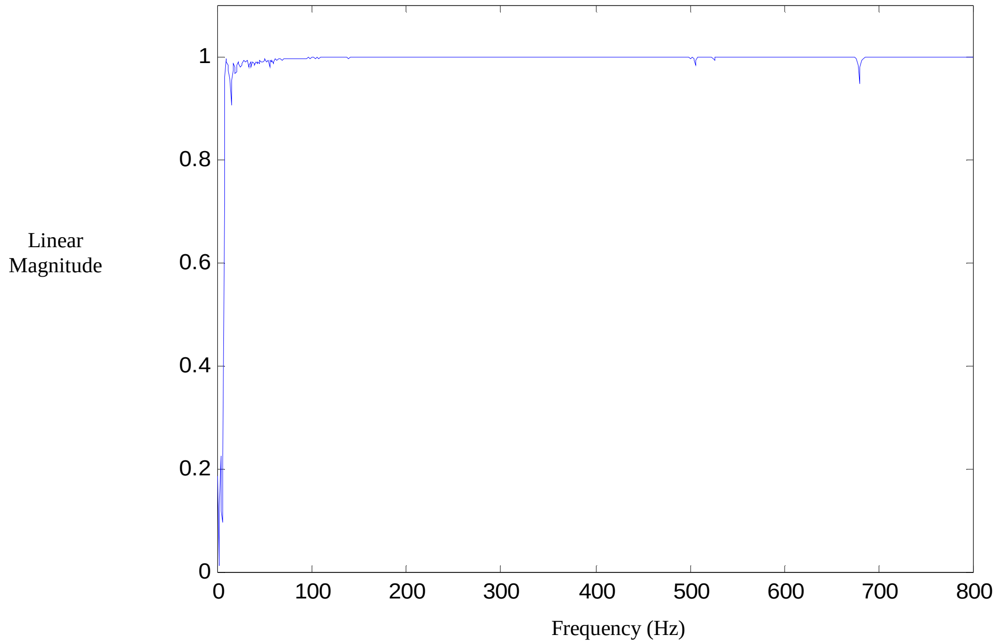
(b) Phase angle plot



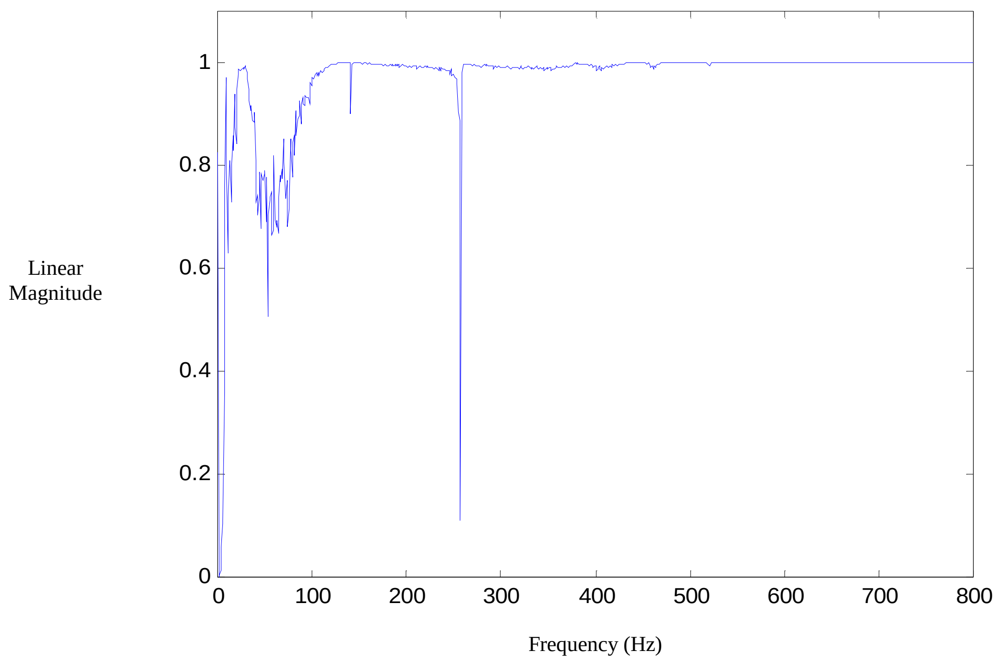
(c) IRF plot calculated from the FRF

Figure 4.2.6: FRF (a), Phase angle (b) and IRF (c) of the structure's sixth output measurement location point

Experimental Coherence



(a) First Location



(b) Third Location

Figure 4.2.7: Coherence of the (a) First and (b) Third Location

CHAPTER 5

Conclusions and Recommendations

The present investigation provides results that can be used to understand how and when the accuracy of the damping ratio is affected with respect to which method is used. Based on the discussion (chapter 5) and the results (chapter 4) the following conclusions can be made:

- In general, the Rational Fraction Polynomial Method gives the most accurate damping ratio estimate. This the most reliable method because it works in the frequency domain and doesn't have to calculate the IRF. Therefore, the truncation in the FRF doesn't affect the damping ratio estimation.
- The Hilbert Envelope Method gives a more reliable damping ratio estimation than the CEM and ITD when the FRF is truncated.
- When using the CEM and ITD method as a single-mode method the operator has to be very careful because when the mode is isolated the FRF looses a lot of information and the calculated IRF will have leakage error.
- If the CEM or ITD method is used to process a truncated FRF, its estimated damping ratio of the highest natural frequencies near to the maximum FRF frequency will not be exact.
- CEM and ITD methods are more suitable for heavily damped systems than for lightly damped systems.

The following recommendations are made for future work in order to try to improve the accuracy of the damping ratio estimation:

- Apply the cosine taper data window from reference [18] (page 146) to the FRF to:
  - ✓ isolate each single-mode in the Hilbert Envelope method
  - ✓ isolate each single-mode when the CEM and ITD method are used in the single-mode approach.

- ✓ compensate the FRF truncation when the CEM and ITD are used in the multiple degree of freedom approach.

- If there is time leakage and the CEM or ITD method is used, specify the IRF time range where parameters are going to be estimated. This can be done to avoid using the time leakage when the parameters are estimated from the IRF.
- Perform experiments with some other excitations in order to observe if the method's performance depends of the excitation, due to the frequency range input.

List of References

1. Svend Gade and Henrik Herlufsen, “Digital Filter Techniques vs. FFT Techniques for Damping Measurements”, Proceedings of the International Modal Analysis Conference, pp. 1056-1064 1990.

2. M. Imregun, “A Comparison of SDOF and Global MDOF Modal Analysis Techniques when applied to a Lightly-Damped Linear Structure”, Proceedings of the International Modal Analysis Conference, pp. 435-441 1991.

3. S. O’F. Fahey and J. Pratt, “Frequency Domain Modal Estimation Techniques”, EXPERIMENTAL TECHNIQUES, pp. 33-37, September/October 1998).

4. S. O’F. Fahey and J. Pratt, “Time Domain Modal Estimation Techniques”, EXPERIMENTAL TECHNIQUES, 45-49 (November/December 1998).

5. Nuno Manuel Mendes Maia, “Reflections on Some SDOF Modal Analysis Methods”, Proceedings of the International Modal Analysis Conference, pp. 658-666, 1990.

6. Gilles Collot, “A New Method for Loss Factor Measurements, Using the Hilbert Transform Implementation in a Software to Automize the ‘Oberst’ Method”, International Modal Analysis Conference Proceedings, pp. 389-393

7. S. Gade, K. Zaveri, H. Konstantin-Hansen and H. Herlufsen, “Complex Modulus and Damping Measurements Using Resonant and Non-Resonant Methods”, Proceedings of the International Modal Analysis Conference, pp.1458-1462, 1995.

8. A.L Wicks’ Handouts of the Signal Processing for Experimental Analysis at Virginia Tech’s Mechanical Engineering Department, Fall Semester of 1998.

9. Nuno Manuel Mendes Maia and JúlioMontalvão e Silva, Eds., Theoretical and Experimental Modal Analysis, John Wiley & Sons Inc., New York,

10. N. Thrane, J. Wismer, H. Konstantin-Hansen, and S. Gade, "Practical use of the Hilbert Transform", Bruel & Kjær.

11. Ronald N. Bracewell, The Fourier Transform and its Applications, second edition, McGraw-Hill, Inc., USA, 1986.

12. S. R. Ibrahim and E. C. Mikulcik, "A Method for the Direct Identification of Vibration Parameters from the Free Response", Sounds & Vibrations Bulletin, pp. 183-198, 1977.

13. Code of the MatLab Software's "infreqs" function, revision 1.2, The MathWorks, Inc., 1998.

14. M. H. Richardson and D. L. Formenti, "Parameter Estimation from Frequency Response Measurements using Rational Fraction Polynomials", Proceedings of the International Modal Analysis Conference, pp.167-181, 1982.

15. S. O'F. Fahey, "Rational Fraction Polynomial Frequency Domain Methods for Extracting Modal Parameters, Notes given to me by the author; Ph.D. candidate at Virginia Tech, April 1997.

16. John H. Mathews and Kurtis D. Fink, Numerical Methods Using MatLab, third edition, Prentice Hall, Inc., NJ, 1999.

17. Duane Hanselman and Bruce Littlefield, The Student Edition of MatLab: Version 5, User's Guide, NJ, 1997.

18. D. E. Newland, An Introduction to Random Vibrations, Spectral & Wavelet Analysis, third edition, Longman Singapore Pte. Ltd., Singapore, 1993.

19. William T. Thomson and Marie Dillon Dahleh, Theory of Vibration with Applications, fifth edition, Prentice Hall, Inc., NJ, 1998.

20. Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab, Signal & Systems, second edition, Prentice Hall, Inc., NJ, 1997.

21. Thomas G. Bechwith, Roy D. Marangoni, and John H. Lienhard V, Mechanical Measurements, fifth edition, Addison-Wesley Publishing Company, Inc., USA, 1995.

Appendix A

Data Processing Codes

The following MatLab codes were used to process the data. MatLab created all the plots and calculated the values for the tables, excluding the damping ratio percent of errors. While these codes were processing the data, they were displaying and creating an output result file.

These codes were created for the simulated analytical analysis, and the experimental analysis. The simulated analytical codes first generated the data with the specific parameters and then calculated the values with the specified method. The experimental data was obtained from the HP dynamic signal analyzer, but it was changed to ASCII format with three columns before processing it in MatLab.

A.1 Code for the Simulated Analytical Data

Generating Data

```
clear
clc
format long g
close all hidden
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%---- Signal Parameters ----%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
df = 1; % Frequency Resolution
tt = 1/df;  % Total time
L = tt*2048; % Sampling Frequency Number
t = linspace(0,tt,L);
dt = t(3) - t(2);
N = L/2;
f = linspace(0,df*(N-1),N);

%##############################################################%
%----------- Simulated Analytical Data Generation  ------------%
%##############################################################%

% First Mode
  A1 = 10;
  E1 = input('First damping ratio: ');
  fn1 = df*N/8;
  wn1 = fn1*2*pi;
  a1 = E1*wn1;
  wd1 = wn1*sqrt(1 - E1^2);
% Second Mode
  A2 = 2*A1;
  E2 = E1/1.2;
  fn2 = 2*fn1;
  wn2 = fn2*2*pi;
  a2 = E2*wn2;
  wd2 = wn2*sqrt(1 - E2^2);
% Third Mode
  A3 = 1.5*A2;
  E3 = E2/1.8;
  fn3 = 2*fn2;
  wn3 = fn3*2*pi;
  a3 = E3*wn3;
  wd3 = wn3*sqrt(1 - E3^2);
% Fourth Mode
  A4 = 1.3*A3;
  E4 = E3/1.5;
  fn4 = 1.5*fn3;
  wn4 = fn4*2*pi;
  a4 = E4*wn4;
  wd4 = wn4*sqrt(1 - E4^2);
% Fifth Mode
  A5 = 1.3*A4;
  E5 = E4/2;
  fn5 = 1.05*fn4;
  wn5 = fn5*2*pi;
  a5 = E5*wn5;
  wd5 = wn5*sqrt(1 - E5^2);

x = (A1*exp(-a1*t).*sin(wd1*t) + A2*exp(-a2*t).*sin(wd2*t) + A3*exp(-a3*t).*sin(wd3*t)...
   + A4*exp(-a4*t).*sin(wd4*t) + A5*exp(-a5*t).*sin(wd5*t));

  % Calculating the Frequency Response Function (FRF)
```

```
    x_dft = fft(x);

  % Calculating the Impulse Response Function (IRF)
    x_ift = ifft(x_dft);


%--- Displaying Freqs. and Damping Ratios ---%

disp('Natural Frequencies and Damping Ratios for the data with two close modes')
Natural_frequency_Damping_ratio = [ fn1 E1 ; fn2 E2 ; fn3 E3 ; fn4 E4 ; fn5 E5]

fig = 1;
p_fig = menu('Plot graphs?','Yes','No');
if p_fig == 1
%#########################################################%
%---------------------------------------------------------%
% Graphing Data                                           %
%---------------------------------------------------------%
%#########################################################%

%### Plotting the FRFs and IRFs in the same plot for each data ###%
%----- Data with two close modes (x2) -----%

figure(fig)
fig = 1 + fig;
subplot(2,1,1)
semilogy(f,abs(x_dft(1:N)))
title(sprintf('FRF with two close modes'))
xlabel('frequency (Hz)')
subplot(2,1,2)
plot(t,x)
title(sprintf('IRF with two close modes'))
xlabel('time (seconds)')
clear w* a* tt A* p_fig
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%-----------------------------%%%%%
%%%%% Estimating the Damping Ratio %%%%%
%%%%%-----------------------------%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


method = MENU('Choose method','ITD','CEM','RFP','Hilbert');
if method == 1
   itd_analysis_cm_tr
elseif method == 2
   cem_analysis_cm_tr
elseif method == 3
   rfp_analysis_cm_tr
else
   hil_analysis_cm_tr
end
```

Complex Exponential Method

```
format long g
%%%%%%%%%%%%%%%%%%%%%%%

method = 'cem';
frf = x_dft;
dofm = menu('Approach','MDOF','SDOF');
if dofm == 1
   diary (sprintf('%s_%1.0d_Mdof_results.m',method,E1))
   disp('MDOF  Simulated Data')
else
   diary (sprintf('%s_%1.0d_Sdof_results.m',method,E1))
   disp('SDOF  Simulated Data')
end
```

```
   disp([sprintf('\n')])
   disp('CEM METHOD')
   disp([sprintf('\n')])
   disp('Two Close Modes')
   disp([sprintf('\n')])
   disp('Natural Frequencies and Damping Ratios for the data with two close modes')
   Natural_frequency_Damping_ratio = [ fn1 E1 ; fn2 E2 ; fn3 E3 ; fn4 E4 ; fn5 E5 ]

W_TR = length(frf)/2;;
N = length(frf)/2;
frf = conj(frf(1:N)');  % The HP analyzer just gives the positive freq components
f = f(1:N);
df = f(3) - f(2);

% Before getting to this point I need to select the data
% and need freq. matrix

%----------- Specifying the freq range of curve fit -----------%
wind = menu('Do you want to specify the freq. range for the curve fit?','Yes','No');
if wind == 1
  disp('Frequency Range Specified')
  specify = menu('How do you want to specify the freq. range?','Point on Graph','Type
it');
   if specify == 1
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (minimum frequency)')
      [x_frm1,y]=ginput(1);
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (maximum frequency)')
      [W_TR,y]=ginput(1);
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq
= %8.4g',x_frm1,W_TR)
   else
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      x_frm1 = input('Minimum Frequency (Hz): ');
      W_TR = input('Maximum Frequency (Hz): ');
     sprintf('The selected frequency range is:\n\tMinimum freq = %d\n\tMaximum freq = %d'
,x_frm1,W_TR)
   end

   % Isolating the Frequency Range
      x_frm1 = round(x_frm1/df + 1);
      W_TR = round(W_TR/df + 1);
      frf_F1 = zeros(x_frm1-1,1);                   % Putting zeros before the isolated
FRF components
      frf_F1(x_frm1:W_TR) = frf(x_frm1:W_TR);             % Isolated FRF components
      frf_F1(W_TR+1:N) = ones(N-(W_TR),1);      % Putting zeros after the isolated FRF
components
      % Adding the conjugate components to the FRF
         frf_F1(N+1) = real(frf_F1(N));
         frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));
         [r,c] = size(frf_F1);
         if r < c
            frf = conj(frf_F1');
         else
            frf = frf_F1;
         end
         clear frf_F1

else
   % Adding the conjugate components to the FRF
     frf(N+1) = real(frf(N));
     frf(N+2:2*N) = conj(frf(N:-1:2));
     x_frm1 = 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(fig + 1)
```

```
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
dof = input('How many DOF?: ');                              %%%  D  O  F

diary off                                        % Turns diary off
%------- Calculating the Impulse Response Function from the FRF Inverse -------%
irf = real(ifft(frf));

%-- Time parameters --%
t = linspace(0,1/df,2*N);
dt = t(2)-t(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------- Processing Data --------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L = length(irf);
M = L/2;
n = dof*2; % this value is equal to - DOF*2

for r = 1:n
   h1(:,r) = real(irf(r:M-1+r));
end
for r = 1:M
   hv1(r,:) = -real(irf(n+r));
end

B1 = inv(h1'*h1)*(h1'*hv1);

B1(n+1,1) = 1;
B1v = B1(n+1:-1:1);
V_cem = roots(B1v);

%--- Calculating the Natural Freq & Damping Ratio ---%
n = length(V_cem);
for r = 1:n
   wn_cem(r) = abs(log(V_cem(r)))/dt;
   Fn_cem(r) = wn_cem(r)/(2*pi);
   Damp_ratio_cem(r) = sqrt(1/(((imag(log(V_cem(r)))/real(log(V_cem(r))))^2)+1));
end

%--------- Calculating eigenvector ---------%
for r = 0:(2*N - 1)
   Va_cem(r+1,:) = [conj(V_cem').^r ];
end

Ar_cem = (inv(conj(Va_cem')*Va_cem)*conj(Va_cem')*(irf));

%---------- Calcualting the IRF Curve Fit ----------%
x_cem = Va_cem*Ar_cem;

%---------- Calcualting the FRF Curve Fit ----------%
frf_cem = fft(x_cem);

%--------- Calculating the Residual ---------%
%-- FRF --%
Residual = frf(x_frm1:W_TR) - frf_cem(x_frm1:W_TR);
%-- IRF --%
ResidualT = real(x_cem) - irf;

%--------- Plotting ---------%
figure(fig + 2)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)),f(x_frm1:W_TR),abs(frf_cem(x_frm1:W_TR)))

figure(fig + 3)
plot(f(x_frm1:W_TR)',angle(frf(x_frm1:W_TR)),f(x_frm1:W_TR)',angle(frf_cem(x_frm1:W_TR)))

figure(fig + 4)
plot(f(x_frm1:W_TR)',imag(frf(x_frm1:W_TR)),f(x_frm1:W_TR)',imag(frf_cem(x_frm1:W_TR)))

figure(fig + 5)
subplot(2,1,1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
```

```
subplot(2,1,2)
plot(f(x_frm1:W_TR),abs(Residual))

figure(fig + 6)
plot(t,irf,t,real(x_cem))


figure(fig + 7)
plot(t,ResidualT)

diary on                  % Turns diary ON

%--------- Displaying Results ---------%
Residues_Eigenvalues_cem = [ Ar_cem V_cem ]
format long g
Natural_freq_Damping_ratio_cem = [ Fn_cem' Damp_ratio_cem' ]

%--------- Calculating & Displaying the Standard Deviation ---------%
% Frequecy domain (Residual)
Curvefit_Standard_deviation_FRF = sqrt((Residual'*Residual)/length(Residual))

% Time domain (ResidualT)
Curvefit_Standard_deviation_IRF = sqrt((ResidualT'*ResidualT)/length(ResidualT))

% Around each natural frequency in the FRF
stdn = menu('Standard Deviation around each nat. freq.','Yes','No');
if stdn == 1
   std_nat_freq
end

diary off                                 % Turns diary off

if dofm == 1
   sprintf('The Output file (results) is %s_%1.0d_Mdof_results.m',method,E1)
else
   sprintf('The Output file (results) is %s_%1.0d_Sdof_results.m',method,E1)
end
```

Ibrahim Time Domain Method

```
format long g
%%%%%%%%%%%%%%%%%%%%%%%

method = 'itd';
frf = x_dft;
dofm = menu('Approach','MDOF','SDOF');
if dofm == 1
   diary (sprintf('%s_%1.0d_Mdof_results.m',method,E1))
   disp('MDOF  Simulated Data')
else
   diary (sprintf('%s_%1.0d_Sdof_results.m',method,E1))
   disp('SDOF  Simulated Data')
end
   disp([sprintf('\n')])
   disp('ITD METHOD')
   disp([sprintf('\n')])
   disp('Two Close Modes')
   disp([sprintf('\n')])
   disp('Natural Frequencies and Damping Ratios for the data with two close modes')
   Natural_frequency_Damping_ratio = [ fn1 E1 ; fn2 E2 ; fn3 E3 ; fn4 E4 ; fn5 E5 ]

W_TR = length(frf)/2;;

N = length(frf)/2;
frf = conj(frf(1:N)');  % The HP analyzer just gives the positive freq components
f = f(1:N);
df = f(3) - f(2);

% Before getting to this point I need to select the data
```

```
% and need freq. matrix

%----------- Specifying the freq range of curve fit -----------%
wind = menu('Do you want to specify the freq. range for the curve fit?','Yes','No');
if wind == 1
   disp('Frequency Range Specified')
   specify = menu('How do you want to specify the freq. range?','Point on Graph','Type
it');

   if specify == 1
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (minimum frequency)')
      [x_frm1,y]=ginput(1);
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (maximum frequency)')
      [W_TR,y]=ginput(1);
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq
= %8.4g',x_frm1,W_TR)
   else
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      x_frm1 = input('Minimum Frequency (Hz): ');
      W_TR = input('Maximum Frequency (Hz): ');
      sprintf('The selected frequency range is:\n\tMinimum freq = %d\n\tMaximum freq =
%d',x_frm1,W_TR)
   end

   % Isolating the Frequency Range
      x_frm1 = round(x_frm1/df + 1);
      W_TR = round(W_TR/df + 1);
      frf_F1 = zeros(x_frm1-1,1);                  % Putting zeros before the isolated
FRF components
      frf_F1(x_frm1:W_TR) = frf(x_frm1:W_TR);            % Isolated FRF components
      frf_F1(W_TR+1:N) = ones(N-(W_TR),1);      % Putting zeros after the isolated FRF
components
      % Adding the conjugate components to the FRF
         frf_F1(N+1) = real(frf_F1(N));
         frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));
         [r,c] = size(frf_F1);
         if r < c
         frf = conj(frf_F1');
         else
            frf = frf_F1;
         end
         clear frf_F1

else
   % Adding the conjugate components to the FRF
   frf(N+1) = real(frf(N));
   frf(N+2:2*N) = conj(frf(N:-1:2));
   x_frm1 = 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(fig + 1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
dof = input('How many DOF?: ');                             %%%  D  O  F

diary off                                        % Turns diary off

%------- Calculating the Impulse Response Function from the FRF Inverse -------%
irf = real(ifft(frf));

%-- Time parameters --%
t = linspace(0,1/df,2*N);
dt = t(2)-t(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%------------- Processing Data -------------%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L = length(irf);
M = L/2;
n = 2*dof +1; % n = DOF*2+1
for r = 1:n-1
   x1_itd(r,:) = [real(irf(r:L-(n-r)))]';
   x2_itd(r,:) = [real(irf(r+1:L-(n-(r+1))))]';
end
A = (1/2)*( (x2_itd*x2_itd')*inv((x1_itd*x2_itd')) +
(x2_itd*x1_itd')*inv(x1_itd*x1_itd'));

[Ar_itd,V_itd] = eig(A);

%--- Calculating the Natural Freq & Damping Ratio ---%
n = length(V_itd);
for r = 1:n
   wn_itd(r) = abs(log(V_itd(r,r)))/dt;
   Fn_itd(r) = wn_itd(r)/(2*pi);
   Damp_ratio_itd(r) = sqrt(1/(((imag(log(V_itd(r,r)))/real(log(V_itd(r,r))))^2)+1));
end

%--------- Calculating eigenvector ---------%
rr =1;
for r = 1:n
   % Filtering the eigenvalues; needs the values to be greater than one
   if abs(real(V_itd(r,r)))<= 1 & abs(imag(V_itd(r,r)))<= 1
      r
      V_itd_2(rr) = V_itd(r,r);
      inda(rr) = r;
      rr = rr + 1;
   end

end

L = length(irf);
for r = 0:L-1
   Va_itd(r+1,:) = [ V_itd_2.^r ];
end

Ar_itd_2 = (inv(conj(Va_itd')*Va_itd)*conj(Va_itd')*(irf));

%---------- Calcualting the IRF Curve Fit ----------%
x_itd= Va_itd*Ar_itd_2;

%---------- Calcualting FRF Curve Fit ----------%
frf_itd = fft(x_itd);

%--------- Calculating the Residual ---------%
%-- FRF --%
Residual = frf(x_frm1:W_TR) - frf_itd(x_frm1:W_TR);

%-- IRF --%
ResidualT = real(x_itd) - irf;

%--------- Plotting ---------%
figure(fig + 2)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)),f(x_frm1:W_TR),abs(frf_itd(x_frm1:W_TR)))

figure(fig + 3)
plot(f(x_frm1:W_TR)',angle(frf(x_frm1:W_TR)),f(x_frm1:W_TR)',angle(frf_itd(x_frm1:W_TR)))

figure(fig + 4)
plot(f(x_frm1:W_TR)',imag(frf(x_frm1:W_TR)),f(x_frm1:W_TR)',imag(frf_itd(x_frm1:W_TR)))

figure(fig + 5)
subplot(2,1,1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
subplot(2,1,2)
plot(f(x_frm1:W_TR),abs(Residual))

figure(fig + 6)
```

```
plot(t,real(x_itd),t,irf)

figure(fig + 7)
plot(t,ResidualT)

diary on               % Turns diary ON
%--------- Displaying Results ---------%
Residues_and_Effective_Eigenvalues_itd = [ Ar_itd_2 conj(V_itd_2') ]
format long g
Natural_freq_Damping_ratio_itd = [ Fn_itd' Damp_ratio_itd' ]

%--------- Calculating & Displaying the Standard Deviation ---------%
% Frequecy domain (Residual)
Curvefit_Standard_deviation_FRF = sqrt((Residual'*Residual)/length(Residual))

% Time domain (ResidualT)
Curvefit_Standard_deviation_IRF = sqrt((ResidualT'*ResidualT)/length(ResidualT))

% Around each natural frequency in the FRF
stdn = menu('Standard Deviation around each nat. freq.','Yes','No');
if stdn == 1
   std_nat_freq
end

diary off                                  % Turns diary off

if dofm == 1
   sprintf('The Output file (results) is %s_%1.0d_Mdof_results.m',method,E1)
else
   sprintf('The Output file (results) is %s_%1.0d_Sdof_results.m',method,E1)
end
```

Rational Fraction Polynomial Method

```
format long g
%%%%%%%%%%%%%%%%%%%%%%%

method = 'rfp';
frf = x_dft;
dofm = menu('Approach','MDOF','SDOF');
if dofm == 1
   diary (sprintf('%s_%1.0d_Mdof_results.m',method,E1))
   disp('MDOF  Simulated Data')
else
   diary (sprintf('%s_%1.0d_Sdof_results.m',method,E1))
   disp('SDOF  Simulated Data')
end
   disp([sprintf('\n')])
   disp('RFP METHOD')
   disp([sprintf('\n')])
   disp('Two Close Modes')
   disp([sprintf('\n')])
   disp('Natural Frequencies and Damping Ratios for the data with two close modes')
   Natural_frequency_Damping_ratio = [ fn1 E1 ; fn2 E2 ; fn3 E3 ; fn4 E4 ; fn5 E5 ]

W_TR = length(frf)/2;;

N = length(frf)/2;
frf = (frf(1:W_TR));  % The HP analyzer just gives the positive freq components
f = f(1:W_TR);
df = f(3) - f(2);

% Before getting to this point I need to select the data
% and need freq. matrix

%----------- Specifying the freq range of curve fit -----------%
wind = menu('Do you want to specify the freq. range for the curve fit?','Yes','No');
if wind == 1
   disp('Frequency Range Specified')
   specify = menu('How do you want to specify the freq. range?','Point on Graph','Type
it');
```

```
   if specify == 1
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (minimum frequency)')
      [x_frm1,y]=ginput(1);
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (maximum frequency)')
      [W_TR,y]=ginput(1);
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq
= %8.4g',x_frm1,W_TR)
   else
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      x_frm1 = input('Minimum Frequency (Hz): ');
      W_TR = input('Maximum Frequency (Hz): ');
      sprintf('The selected frequency range is:\n\tMinimum freq = %d\n\tMaximum freq =
%d',x_frm1,W_TR)
   end

   % Isolating the Frequency Range
      x_frm1 = round(x_frm1/df + 1);
      W_TR = round(W_TR/df + 1);
      frf_F1 = ones(x_frm1-1,1);                  % Putting ones before the isolated FRF
components (mult by mean of irf)
      frf_F1(x_frm1:W_TR) = frf(x_frm1:W_TR);     % Isolated FRF components
      frf_F1(W_TR+1:N) = ones(N-(W_TR),1);        % Putting ones after the isolated FRF
components (mult by mean of irf)
      %--------------------%
      [r,c] = size(frf_F1);
         if r > c
            frf = conj(frf_F1');
         else
            frf = frf_F1;
         end
       clear frf_F1
else
   x_frm1 = 1;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(fig + 1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
dof = input('How many DOF?: ');                              %%%  D  O  F

diary off                   % Turns diary OFF

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%------------- Processing Data -------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

w = 2*pi*f(x_frm1:W_TR);
% Scaling the frequency from 0 to 1
% Dividing by the maximum frequency
wi = w/max(w);
n = dof*2;
wt = ones(1,length(frf(x_frm1:W_TR)));
iter = 100;
tol = 0;
[A,B] = invfreqs(frf(x_frm1:W_TR),wi,n,n,wt,iter,tol);
[R_rfp,P_rfp,K] = residue(A,B); % Residues & Poles, respectively


%--- Calculating the Natural Freq & Damping Ratio ---%
Damp_ratio_rfp = -real(P_rfp)./(abs(P_rfp));
% Here the natural frequency is multiplied by
% the maximum in because the frequencies were
% scaled from 0 to 1 to avoid problems in
% the invfreqs function
```

```
Fn_rfp = abs(P_rfp)*max(w)/(2*pi);

%---------- Calcualting the FRF Curve Fit ----------%
frf_rfp = freqs(A,B,wi);

%--- Adding the conjugate components to the FRF and zeros in the truncated ---%
   % Experimental FRF
   frf(1:x_frm1-1) = 0;
   frf(N+1) = 0;
   frf(N+2:2*N) = conj(frf(N:-1:2));
   % Curve Fit
   frf_rfp(x_frm1:W_TR) = frf_rfp;
   frf_rfp(1:x_frm1 -1) = zeros(1,x_frm1 -1);
   frf_rfp(N+1) = 0;
   frf_rfp(N+2:2*N) = conj(frf_rfp(N:-1:2));

%--- Calculating the Impulse Response Function from the FRF Inverse ---%
irf = real(ifft(frf));
irf_rfp = real(ifft(frf_rfp));

%--------- Calculating the Residual ---------%
%-- FRF --%
Residual = frf(x_frm1:W_TR) - frf_rfp(x_frm1:W_TR);
%-- IRF --%
ResidualT = irf - irf_rfp;

%--------- Plotting ---------%
figure(fig + 2)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)),f(x_frm1:W_TR),abs(frf_rfp(x_frm1:W_TR)))

figure(fig + 3)
plot(f(x_frm1:W_TR),angle(frf(x_frm1:W_TR)),f(x_frm1:W_TR),angle(frf_rfp(x_frm1:W_TR)))

figure(fig + 4)
plot(f(x_frm1:W_TR),imag(frf(x_frm1:W_TR)),f(x_frm1:W_TR),imag(frf_rfp(x_frm1:W_TR)))

figure(fig + 5)
subplot(2,1,1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
subplot(2,1,2)
plot(f(x_frm1:W_TR),abs(Residual))

figure(fig + 6)
t = linspace(0,1/df,2*N);
plot(t,irf,t,irf_rfp)

figure(fig + 7)
plot(t,(ResidualT))

diary on                     % Turns diary ON


%--------- Displaying Results ---------%
Residues_and_Poles_rfp = [ R_rfp P_rfp ]
format long g
Natural_freq_Damping_ratio_rfp = [ Fn_rfp Damp_ratio_rfp ]

%--------- Calculating & Displaying the Standard Deviation ---------%
% Frequecy domain (Residual)
Curvefit_Standard_deviation_FRF = sqrt((Residual*Residual')/length(Residual))

% Time domain (ResidualT)
Curvefit_Standard_deviation_IRF = sqrt((ResidualT*ResidualT')/length(ResidualT))

% Around each natural frequency in the FRF
stdn = menu('Standard Deviation around each nat. freq.','Yes','No');
if stdn == 1
   std_nat_freq_rfp
end

diary off                                  % Turns diary off
```

```
if dofm == 1
   sprintf('The Output file (results) is %s_%1.0d_Mdof_results.m',method,E1)
else
   sprintf('The Output file (results) is %s_%1.0d_Sdof_results.m',method,E1)
end
```

Hilbert Envelope Method

```
format long g
%%%%%%%%%%%%%%%%%%%%%%

method = 'hil';
frf = x_dft;
diary (sprintf('%s_%1.0d_results.m',method,E1))

   disp('Hilbert Envelope METHOD')
   disp([sprintf('\n')])
   disp('Two Close Modes')
   disp([sprintf('\n')])
   disp('Natural Frequencies and Damping Ratios for the data with two close modes')
   Natural_frequency_Damping_ratio = [ fn1 E1 ; fn2 E2 ; fn3 E3 ; fn4 E4 ; fn5 E5 ]

W_TR = length(frf)/2;;

N = length(frf)/2;
frf = conj(frf(1:N)');  % The HP analyzer just gives the positive freq components
f = f(1:N);
df = f(3) - f(2);

% Before getting to this point I need to select the data
% and need freq. matrix

%----------- Specifying the freq range of curve fit -----------%
wind = menu('Do you want to specify the freq. range for the curve fit?','Yes','No');
if wind == 1
   disp('Frequency Range Specified')
   specify = menu('How do you want to specify the freq. range?','Point on Graph','Type
it');
   if specify == 1
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (minimum frequency)')
      [x_frm1,y]=ginput(1);
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (maximum frequency)')
      [W_TR,y]=ginput(1);
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq
= %8.4g',x_frm1,W_TR)
   else
      figure(fig + 1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      x_frm1 = input('Minimum Frequency (Hz): ');
      W_TR = input('Maximum Frequency (Hz): ');
      sprintf('The selected frequency range is:\n\tMinimum freq = %d\n\tMaximum freq =
%d',x_frm1,W_TR)
   end

   % Isolating the Frequency Range
      x_frm1 = round(x_frm1/df + 1);
      W_TR = round(W_TR/df + 1);
      frf_F1 = zeros(x_frm1-1,1);                  % Putting zeros before the isolated
FRF components
      frf_F1(x_frm1:W_TR) = frf(x_frm1:W_TR);        % Isolated FRF components
      frf_F1(W_TR+1:N) = zeros(N-(W_TR),1);          % Putting zeros after the isolated
FRF components
      % Adding the conjugate components to the FRF
         frf_F1(N+1) = real(frf_F1(N));
```

```
         frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));
         frf = frf_F1;
         clear frf_F1

else
   % Adding the conjugate components to the FRF
   frf(N+1) = real(frf(N));
   frf(N+2:2*N) = conj(frf(N:-1:2));
   x_frm1 = 1;
end


figure(fig + 1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
%%%%%%%%%%%%%%%%%%%%%%%%

dof = input('How many DOF?: ');                              %%%  D  O  F

%-- Time parameters --%
    t = linspace(0,1/df,2*N);
    dt = t(2)-t(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------- Processing Data --------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for n = 1:dof
   figure(fig + 1)
   semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
   sprintf('Type natural frequency (Hz) # %d',n)
   fn(n) = input('');
   disp('')
   %---------- Selecting in which way the freq. range will be specified ----------%
   specify = menu('How do you want to specify the freq. range?','Range of window','Point 
on Graph','Type it');
   if specify == 1
      % (+-) Value for the isolating window centered at the natural freq
      R = input('Type the range (spectral lines) of the window (+ -): ');
      % Isolate specified freq range
      Comp_f = abs(round(fn(n)/df + 1));        % Componenet in 'f' where the nat freq is 
located
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq 
= %8.4g',f(Comp_f-R),f(Comp_f+R))

      frf_F1 = zeros(Comp_f-R-1,1);                             % Putting zeros before 
the isolated FRF components
      frf_F1(Comp_f-R:Comp_f+R) = frf(Comp_f-R:Comp_f+R);        % Isolated FRF 
components
      frf_F1(Comp_f+R+1:N) = zeros(N-(Comp_f+R),1);              % Putting zeros after 
the isolated FRF components

   elseif specify == 2 | specify == 3
      if specify == 2
         figure(fig + 1)
         semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
         title('Select the first point (minimum frequency)')
         [x_frmh1,y]=ginput(1);
         figure(fig + 1)
         semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
         title('Select the first point (maximum frequency)')
         [x_frmh2,y]=ginput(1);
         sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum 
freq = %8.4g',x_frmh1,x_frmh2)
      else
         figure(fig + 1)
         semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
         x_frmh1 = input('Minimum Frequency (Hz): ');
         x_frmh2 = input('Maximum Frequency (Hz): ');
         sprintf('The selected frequency range is:\n\tMinimum freq = %8.4g\n\tMaximum 
freq = %8.4g',x_frm1,x_frm2)
      end
```

```
      % Isolating the Frequency Range
      x_frmh1 = round(x_frmh1/df + 1);
      x_frmh2 = round(x_frmh2/df + 1);
      frf_F1 = zeros(x_frmh1-1,1);                         % Putting zeros before the
isolated FRF components (mult by mean of irf)
      frf_F1(x_frmh1:x_frmh2) = frf(x_frmh1:x_frmh2);      % Isolated FRF components
      frf_F1(x_frmh2+1:N) = zeros(N-(x_frmh2),1);          % Putting zeros after the
isolated FRF components (mult by mean of irf)
      % Adding the conjugate components to the FRF
         frf_F1(N+1) = real(frf_F1(N));
         frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));

   end

   % Adding the conjugate components to the FRF
      frf_F1(N+1) = real(frf_F1(N));
      frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));

   % Separating the DFT in order to change the phase
      for nn = 2:N;
         frf_F1_j(nn) = j*frf_F1(nn);          % Mult by the complex number j
      end

      for nn = N+2:2*N;
         frf_F1_j(nn) = -j*frf_F1(nn);      % Mult by the complex number -j the conjugate
part of the DFT
      end

   % Impulse Response Function of the isolated FRF
      irf_F1 = (ifft(frf_F1));
      irf_F1 = conj(irf_F1');

   % Hilbert transform (quadrature function).  The phase was changed previously
      hil_F1 = (ifft(frf_F1_j));

   % Envelope of the decaying signal
      env_F1 = irf_F1 - j*hil_F1;
      env_F1_dB = 20*log10(abs(env_F1));         % dB scale for the envelope's magnitude

   % Least Square Method to calculate slope, and then damping ratio
      specifyT = menu('How do you want to specify the IRF time range?','Point on
Graph','Type it');

      if specifyT == 1
         figure(fig + n + 1)
         plot(t,env_F1_dB)
         title('Select two points in the time domain (x axis)')
         [x,y]=ginput(2);
         sprintf('The selected time range is:\n\tMinimum time: %8.5g \n\tMaximum time:
%8.5g',x(1),x(2))
      else
         figure(fig + n + 1)
         plot(t,env_F1_dB)
         disp('The selected time range is:')
         x(1) = input('   Minimum time: ');
         x(2) = input('   Maximum time: ');
      end


      [np] = round(x/dt + 1);
      t1 = t(np(1):np(2));
      env_F1_dB1 = env_F1_dB(np(1):np(2));
      m = length(t1);               % Calculates the amount of points data
      sum_x = sum(t1);              % Summatory of the t points (x components)
      sum_y = sum(env_F1_dB1);      % Summatory of the env_dB points (y components)
      sum_x_sq = dot(t1,t1);        % Summatory of the square value of each t points (x
components)
      sum_xy = dot(t1,env_F1_dB1);  % Summatory of the multiplication of t and env_dB
points (x and y components)
      LQ1 = [m sum_x; sum_x sum_x_sq];
      LQ2 = [sum_y ; sum_xy];
```

```
        LQ3 = inv(LQ1) * LQ2;
        slope = LQ3(2);
        % Calculating damping ratio
        damping_F1(n) = -slope/(20*log10(exp(1)))         ;
        damping_ratio_F1(n) = damping_F1(n)/(fn(n)*(2*pi))  ;

        % Plotting the Hilbert Process in four plots (for each mode)
        figure(fig + n + 1)
        subplot(2,2,1)
        plot(f,abs(frf_F1(1:W_TR)))
        title(sprintf('Isolated Natural Frequency # %d',n))

        subplot(2,2,2)
        plot(t,real(irf_F1))
        title('Impulse Response Function')

        subplot(2,2,3)
        plot(t,abs(env_F1))
        title('Envelope of the Impulse Response Function')

        subplot(2,2,4)
        plot(t,env_F1_dB)
        %title('dB Scale of the Envelope')
        xlabel('time (s)')
        ylabel('dB scale')
end
%--------- Calculatin the Exp. IRF ---------%
irf = real(ifft(frf));

%--------- Calculating the eigenvalues ---------%
V_hil = exp((-damping_ratio_F1 + i*sqrt(1 - damping_ratio_F1.^2)).*fn*2*pi*dt);
rr = length(V_hil);
V_hil(rr+1:2*rr) = conj(V_hil);

%--------- Calculating eigenvector ---------%
L = length(t);
for r = 0:L-1
   Va_hil(r+1,:) = [conj(V_hil).^r ];
end

Ar_hil = (inv(conj(Va_hil')*Va_hil)*conj(Va_hil')*(irf));

%---------- Calcualting the IRF Curve Fit ----------%
x_hil = Va_hil*Ar_hil;

%---------- Calcualting FRF Curve Fit ----------%
frf_hil = fft(x_hil);

%--------- Calculating the Residual ---------%
%-- FRF --%
Residual = frf(x_frm1:W_TR) - frf_hil(x_frm1:W_TR);
%-- IRF --%
ResidualT = real(x_hil) - irf;

%--------- Plotting ---------%
figure(fig + dof + 2)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)),f(x_frm1:W_TR),abs(frf_hil(x_frm1:W_TR)))

figure(fig + dof + 3)
plot(f(x_frm1:W_TR),angle(frf(x_frm1:W_TR)),f(x_frm1:W_TR),angle(frf_hil(x_frm1:W_TR)))

figure(fig + dof + 4)
plot(f(x_frm1:W_TR),imag(frf(x_frm1:W_TR)),f(x_frm1:W_TR),imag(frf_hil(x_frm1:W_TR)))

figure(fig + dof + 5)
subplot(2,1,1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
subplot(2,1,2)
plot(f(x_frm1:W_TR),abs(Residual))

figure(fig + dof + 6)
```

```
plot(t,real(x_hil),t,irf)

figure(fig + dof + 7)
plot(t,ResidualT)

%--------- Displaying Results ---------%
Residues_Eigenvalues_hil = [ Ar_hil conj(V_hil') ]
Natural_freq_Damping_ratio_hil = [ fn' damping_ratio_F1' ]


%--------- Calculating & Displaying the Standard Deviation ---------%
% Frequecy domain (Residual)
Curvefit_Standard_deviation_FRF = sqrt((Residual'*Residual)/length(Residual))

% Time domain (ResidualT)
Curvefit_Standard_deviation_IRF = sqrt((ResidualT'*ResidualT)/length(ResidualT))

% Around each natural frequency in the FRF
stdn = menu('Standard Deviation around each nat. freq.','Yes','No');
if stdn == 1
   std_nat_freq
end

diary off                                   % Turns diary off

sprintf(' The Output file (results) is %s_%1.0d_results.m',method,E1)
```

Standard Deviation

```
clear x_frm1
disp('Standard Deviation around each nat. freq.')
disp([sprintf('\n')])
np = input('   How many modes were identified? ');
disp([sprintf('\n')])
for r = 1:np
   NT = input(sprintf('\tType natural freq # %d (integer number): ',r));
   x_frm1 = round((NT - min(f))/df + 1);      % Index Matrix of Nat. Freq.
   disp([sprintf('\t\t') 'Standard Deviation = ' ...
         num2str(sqrt((Residual(x_frm1-10:x_frm1+10)'...
         *Residual(x_frm1-10:x_frm1+10))/length(Residual(x_frm1-10:x_frm1+10))))])
   disp([sprintf('\n')])
end
```

Standard Deviation for the RFP

```
clear x_frm1
disp('Standard Deviation around each nat. freq.')
disp([sprintf('\n')])
np = input('   How many modes were identified? ');
disp([sprintf('\n')])
for r = 1:np
   NT = input(sprintf('\tType natural freq # %d (integer number): ',r));
   x_frm1 = round((NT - min(f))/df + 1);      % Index Matrix of Nat. Freq.
   disp([sprintf('\t\t') 'Standard Deviation = ' ...
         num2str(sqrt((Residual(x_frm1-10:x_frm1+10)...
         *Residual(x_frm1-10:x_frm1+10)')/length(Residual(x_frm1-10:x_frm1+10))))])
   disp([sprintf('\n')])
end
```

A.2 Code for Experimental Data

Selecting Method

```
close all hidden
method = MENU('Choose method','ITD','CEM','RFP','Hilbert');
if method == 1
```

```
   itd_analysis2  % file-name if the itd method
elseif method == 2
   cem_analysis2  % file-name if the cem method
elseif method == 3
   rfp_analysis2  % file-name if the rfp method
else
   hil_analysis2  % file-name if the hilbert method
end
```

Complex Exponential Method

```
clear
clc
close all hidden
format long
%%%%%%%%%%%%%%%%%%%%%%%
ls *dat
method = 'cem';

disp('Write between quotes the name of the selected data (do not write extension)')
data = input('');

frf = load (sprintf('%s.dat',data));

[r,c] = size(frf);

if c == 3

diary (sprintf('%s_%s_results.m',method,data))

disp('CEM_METHOD')
sprintf('\nSelected data is: %s\n',data)

f = frf(:,1);
df = f(3) - f(2);
frf = frf(:,2) + i*frf(:,3);
W_TR = length(frf);
N = W_TR;

%----------- Truncation in the FRF -----------%
trf = menu('Is this FRF truncated in the frequency domain?','Yes','No');
if trf == 1
   spl = input('How many spectral lines were truncated? ');
   frf(N+spl) = 0;
   N = length(frf);
   disp(sprintf('FRF with %d truncated spectral lines',spl))
end

% Before getting to this point I need to select the data
% and need freq. matrix

%----------- Specifying the freq range of curve fit -----------%
wind = menu('Do you want to specify the freq. range for the curve fit?','Yes','No');
if wind == 1
   disp('Frequency Range Specified')
   specify = menu('How do you want to specify the freq. range?','Point on Graph','Type
it');
   mean_irf = mean(real(ifft(frf)));
   if specify == 1
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (minimum frequency)')
      [x_frm1,y]=ginput(1);
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (maximum frequency)')
      [W_TR,y]=ginput(1);
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq
= %8.4g',x_frm1,W_TR)
   else
```

```
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      x_frm1 = input('Minimum Frequency (Hz): ');
      W_TR = input('Maximum Frequency (Hz): ');
      sprintf('The selected frequency range is:\n\tMinimum freq = %d\n\tMaximum freq =
%d',x_frm1,W_TR)
   end

   % Isolating the Frequency Range
      x_frm1 = round((x_frm1 - min(f))/df + 1);
      W_TR = round((W_TR - min(f))/df + 1);
      frf_F1 = zeros(x_frm1-1,1);     % Putting zeros before the isolated FRF components
      frf_F1(x_frm1:W_TR) = frf(x_frm1:W_TR);             % Isolated FRF components
      frf_F1(W_TR+1:N) = ones(N-(W_TR),1);    % Putting zeros after the isolated FRF
components
      % Adding the conjugate components to the FRF
         frf_F1(N+1) = real(frf_F1(N));
         frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));
         frf = frf_F1;
         clear frf_F1

else
   % Adding the conjugate components to the FRF
   frf(N+1) = real(frf(N));
   frf(N+2:2*N) = conj(frf(N:-1:2));
   x_frm1 = 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
dof = input('How many DOF?: ');                             %%%  D  O  F

diary off                                        % Turns diary off
%------- Calculating the Impulse Response Function from the FRF Inverse -------%
irf = real(ifft(frf));

%-- Time parameters --%
t = linspace(0,1/df,2*N);
dt = t(2)-t(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------- Processing Data --------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L = length(irf);
M = L/2;
n = dof*2; % this value is equal to - DOF*2

for r = 1:n
   h1(:,r) = real(irf(r:M-1+r));
end
for r = 1:M
   hv1(r,:) = -real(irf(n+r));
end

B1 = inv(h1'*h1)*(h1'*hv1);

B1(n+1,1) = 1;
B1v = B1(n+1:-1:1);
V_cem = roots(B1v);


%--- Calculating the Natural Freq & Damping Ratio ---%
n = length(V_cem);
for r = 1:n
   wn_cem(r) = abs(log(V_cem(r)))/dt;
   Fn_cem(r) = wn_cem(r)/(2*pi);
   Damp_ratio_cem(r) = sqrt(1/(((imag(log(V_cem(r)))/real(log(V_cem(r))))^2)+1));
end

%--------- Calculating eigenvector ---------%
```

```
for r = 0:(2*N - 1)
   Va_cem(r+1,:) = [conj(V_cem').^r ];
end

Ar_cem = (inv(conj(Va_cem')*Va_cem)*conj(Va_cem')*(irf));

%---------- Calcualting the IRF Curve Fit ----------%
x_cem = Va_cem*Ar_cem;

%---------- Calcualting the FRF Curve Fit ----------%
frf_cem = fft(x_cem);

%--------- Calculating the Residual ---------%
%-- FRF --%
Residual = frf(x_frm1:W_TR) - frf_cem(x_frm1:W_TR);
%-- IRF --%
ResidualT = real(x_cem) - irf;

%--------- Plotting ---------%
figure(2)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)),f(x_frm1:W_TR),abs(frf_cem(x_frm1:W_TR)))

figure(3)
plot(f(x_frm1:W_TR)',angle(frf(x_frm1:W_TR)),f(x_frm1:W_TR)',angle(frf_cem(x_frm1:W_TR)))

figure(4)
plot(f(x_frm1:W_TR)',imag(frf(x_frm1:W_TR)),f(x_frm1:W_TR)',imag(frf_cem(x_frm1:W_TR)))

figure(5)
subplot(2,1,1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
subplot(2,1,2)
plot(f(x_frm1:W_TR),abs(Residual))

figure(6)
plot(t,real(x_cem),t,irf)

figure(7)
plot(t,ResidualT)

diary on                     % Turns diary ON

%--------- Displaying Results ---------%
Residues_Eigenvalues_cem = [ Ar_cem V_cem ]
format long g
Natural_freq_Damping_ratio_cem = [ Fn_cem' Damp_ratio_cem' ]

%--------- Calculating & Displaying the Standard Deviation ---------%
% Frequecy domain (Residual)
Curvefit_Standard_deviation_FRF = sqrt((Residual'*Residual)/length(Residual))

% Time domain (ResidualT)
Curvefit_Standard_deviation_IRF = sqrt((ResidualT'*ResidualT)/length(ResidualT))

% Around each natural frequency in the FRF
stdn = menu('Standard Deviation around each nat. freq.','Yes','No');
if stdn == 1
   std_nat_freq
end

diary off                                  % Turns diary off

sprintf(' The Output file (results) is %s_%s_results.m',method,data)

else
   error('Can not process data')
   disp('The FRF data needs to be in three columns')
   disp('   - First column: frequency')
   disp('   - Second column: Real part')
   disp('   - Third column: Imaginary part')
end
```

Ibrahim Time Domain Method

```
clear
clc
close all hidden
format long
%%%%%%%%%%%%%%%%%%%%%%%
ls *dat
method = 'itd';
disp('Write between quotes the name of the selected data (do not write extension)')
data = input('');

frf = load (sprintf('%s.dat',data));

[r,c] = size(frf);

if c == 3

diary (sprintf('%s_%s_results.m',method,data))

disp('ITD_METHOD')
sprintf('\nSelected data is: %s\n',data)

f = frf(:,1);
df = f(3) - f(2);
frf = frf(:,2) + i*frf(:,3);
W_TR = length(frf);
N = W_TR;

%----------- Truncation in the FRF -----------%
trf = menu('Is this FRF truncated in the frequency domain?','Yes','No');
if trf == 1
   spl = input('How many spectral lines were truncated? ');
   frf(N+spl) = 0;
   N = length(frf);
   disp(sprintf('FRF with %d truncated spectral lines',spl))
end
% Before getting to this point I need to select the data
% and need freq. matrix

%----------- Specifying the freq range of curve fit -----------%
wind = menu('Do you want to specify the freq. range for the curve fit?','Yes','No');
if wind == 1
   disp('Frequency Range Specified')
   specify = menu('How do you want to specify the freq. range?','Point on Graph','Type
it');
   mean_irf = mean(real(ifft(frf)));
   if specify == 1
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (minimum frequency)')
      [x_frm1,y]=ginput(1);
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (maximum frequency)')
      [W_TR,y]=ginput(1);
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq
= %8.4g',x_frm1,W_TR)
   else
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      x_frm1 = input('Minimum Frequency (Hz): ');
      W_TR = input('Maximum Frequency (Hz): ');
      sprintf('The selected frequency range is:\n\tMinimum freq = %d\n\tMaximum freq =
%d',x_frm1,W_TR)
   end

   % Isolating the Frequency Range
      x_frm1 = round((x_frm1 - min(f))/df + 1);
      W_TR = round((W_TR - min(f))/df + 1);
```

```
      frf_F1 = zeros(x_frm1-1,1);                  % Putting zeros before the isolated
FRF components
      frf_F1(x_frm1:W_TR) = frf(x_frm1:W_TR);             % Isolated FRF components
      frf_F1(W_TR+1:N) = ones(N-(W_TR),1);      % Putting zeros after the isolated FRF
components
      % Adding the conjugate components to the FRF
         frf_F1(N+1) = real(frf_F1(N));
         frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));
         frf = frf_F1;
         clear frf_F1

else
   % Adding the conjugate components to the FRF
   frf(N+1) = real(frf(N));
   frf(N+2:2*N) = conj(frf(N:-1:2));
   x_frm1 = 1;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
dof = input('How many DOF?: ');                              %%%  D  O  F

diary off                                        % Turns diary off

%------- Calculating the Impulse Response Function from the FRF Inverse -------%
irf = real(ifft(frf));

%-- Time parameters --%
t = linspace(0,1/df,2*N);
dt = t(2)-t(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%------------- Processing Data -------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L = length(irf);
M = L/2;
n = 2*dof +1; % n = DOF*2+1
for r = 1:n-1
   x1_itd(r,:) = [real(irf(r:L-(n-r)))]';
   x2_itd(r,:) = [real(irf(r+1:L-(n-(r+1))))]';
end
A = (1/2)*( (x2_itd*x2_itd')*inv((x1_itd*x2_itd')) +
(x2_itd*x1_itd')*inv(x1_itd*x1_itd'));

[Ar_itd,V_itd] = eig(A);


%--- Calculating the Natural Freq & Damping Ratio ---%
n = length(V_itd);
for r = 1:n
   wn_itd(r) = abs(log(V_itd(r,r)))/dt;
   Fn_itd(r) = wn_itd(r)/(2*pi);
   Damp_ratio_itd(r) = sqrt(1/(((imag(log(V_itd(r,r)))/real(log(V_itd(r,r))))^2)+1));
end

%--------- Calculating eigenvector ---------%
rr =1;
for r = 1:n
   % Filtering the eigenvalues; needs the values to be greater than one
   if abs(real(V_itd(r,r)))<= 1 & abs(imag(V_itd(r,r)))<= 1
      V_itd_2(rr) = V_itd(r,r);
      inda(rr) = r;
      rr = rr + 1;
   end

end

L = length(irf);
for r = 0:L-1
```

```
   Va_itd(r+1,:) = [ V_itd_2.^r ];
end

Ar_itd_2 = (inv(conj(Va_itd')*Va_itd)*conj(Va_itd')*(irf));

%---------- Calcualting the IRF Curve Fit ----------%
x_itd= Va_itd*Ar_itd_2;

%---------- Calcualting FRF Curve Fit ----------%
frf_itd = fft(x_itd);

%--------- Calculating the Residual ---------%
%-- FRF --%
Residual = frf(x_frm1:W_TR) - frf_itd(x_frm1:W_TR);

%-- IRF --%
ResidualT = real(x_itd) - irf;

%--------- Plotting ---------%
figure(2)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)),f(x_frm1:W_TR),abs(frf_itd(x_frm1:W_TR)))

figure(3)
plot(f(x_frm1:W_TR)',angle(frf(x_frm1:W_TR)),f(x_frm1:W_TR)',angle(frf_itd(x_frm1:W_TR)))

figure(4)
plot(f(x_frm1:W_TR)',imag(frf(x_frm1:W_TR)),f(x_frm1:W_TR)',imag(frf_itd(x_frm1:W_TR)))

figure(5)
subplot(2,1,1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
subplot(2,1,2)
plot(f(x_frm1:W_TR),abs(Residual))

figure(6)
plot(t,real(x_itd),t,irf)

figure(7)
plot(t,ResidualT)

diary on                  % Turns diary ON
%--------- Displaying Results ---------%
Residues_and_Effective_Eigenvalues_itd = [ Ar_itd_2 conj(V_itd_2') ]
format long g
Natural_freq_Damping_ratio_itd = [ Fn_itd' Damp_ratio_itd' ]

%--------- Calculating & Displaying the Standard Deviation ---------%
% Frequecy domain (Residual)
Curvefit_Standard_deviation_FRF = sqrt((Residual'*Residual)/length(Residual))

% Time domain (ResidualT)
Curvefit_Standard_deviation_IRF = sqrt((ResidualT'*ResidualT)/length(ResidualT))

% Around each natural frequency in the FRF
stdn = menu('Standard Deviation around each nat. freq.','Yes','No');
if stdn == 1
   std_nat_freq
end

diary off                                  % Turns diary off

sprintf(' The Output file (results) is %s_%s_results.m',method,data)

else
   error('Can not process data')
   disp('The FRF data needs to be in three columns')
   disp('   - First column: frequency')
   disp('   - Second column: Real part')
   disp('   - Third column: Imaginary part')
end
```

Rational Fraction Polynomial Method

```
clear
clc
close all hidden
format long
%%%%%%%%%%%%%%%%%%%%%%%
ls *dat
method = 'rfp';
disp('Write between quotes the name of the selected data (do not write extension)')
data = input('');

frf = load (sprintf('%s.dat',data));

[r,c] = size(frf);

if c == 3

diary (sprintf('%s_%s_results.m',method,data))

disp('RFP_METHOD')
sprintf('\nSelected data is: %s\n',data)

f = frf(:,1);
df = f(3) - f(2);
frf = frf(:,2) + i*frf(:,3);
W_TR = length(frf);
N = W_TR;

%----------- Truncation in the FRF -----------%
trf = menu('Is this FRF truncated in the frequency domain?','Yes','No');
if trf == 1
   spl = input('How many spectral lines were truncated? ');
   %frf(N+spl) = 0;
   %N = length(frf);
   N = W_TR + spl;
   disp(sprintf('FRF with %d truncated spectral lines',spl))
end


% Before getting to this point I need to select the data
% and need freq. matrix

%----------- Specifying the freq range of curve fit -----------%
wind = menu('Do you want to specify the freq. range for the curve fit?','Yes','No');
if wind == 1
   disp('Frequency Range Specified')
   specify = menu('How do you want to specify the freq. range?','Point on Graph','Type 
it');
   mean_irf = mean(real(ifft(frf)));
   if specify == 1
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (minimum frequency)')
      [x_frm1,y]=ginput(1);
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (maximum frequency)')
      [W_TR,y]=ginput(1);
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq 
= %8.4g',x_frm1,W_TR)
   else
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      x_frm1 = input('Minimum Frequency (Hz): ');
      W_TR = input('Maximum Frequency (Hz): ');
      sprintf('The selected frequency range is:\n\tMinimum freq = %d\n\tMaximum freq = 
%d',x_frm1,W_TR)
   end

   % Isolating the Frequency Range
```

```
      x_frm1 = round((x_frm1 - min(f))/df + 1);
      W_TR = round((W_TR - min(f))/df + 1);
      frf_F1 = ones(x_frm1-1,1);                 % Putting ones before the isolated FRF
components (mult by mean of irf)
      frf_F1(x_frm1:W_TR) = frf(x_frm1:W_TR);    % Isolated FRF components
      frf_F1(W_TR+1:N) = ones(N-(W_TR),1);       % Putting ones after the isolated FRF
components (mult by mean of irf)
      %--------------------%
         frf = frf_F1;
         clear frf_F1
else
   x_frm1 = 1;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
dof = input('How many DOF?: ');                            %%%  D  O  F

diary off              % Turns diary OFF

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------- Processing Data --------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

w = 2*pi*f(x_frm1:W_TR);
% Scaling the frequency from 0 to 1
% Dividing by the maximum frequency
wi = w/max(w);
n = dof*2;
wt = ones(1,length(frf(x_frm1:W_TR)));
iter = 50;
tol = 0;
[A,B] = invfreqs(frf(x_frm1:W_TR),wi,n,n,wt,iter,tol);
[R_rfp,P_rfp,K] = residue(A,B); % Residues & Poles, respectively


%--- Calculating the Natural Freq & Damping Ratio ---%
Damp_ratio_rfp = -real(P_rfp)./(abs(P_rfp));
% Here the natural frequency is multiplied by
% the maximum in because the frequencies were
% scaled from 0 to 1 to avoid problems in
% the invfreqs function
Fn_rfp = abs(P_rfp)*max(w)/(2*pi);

%---------- Calcualting the FRF Curve Fit ----------%
frf_rfp = freqs(A,B,wi);

%--- Adding the conjugate components to the FRF and zeros in the truncated ---%
   % Experimental FRF
   frf(1:x_frm1-1) = 0;
   frf(N+1) = 0;
   frf(N+2:2*N) = conj(frf(N:-1:2));
   % Curve Fit
   frf_rfp(x_frm1:W_TR) = frf_rfp;
   frf_rfp(1:x_frm1 -1) = zeros(1,x_frm1 -1);
   frf_rfp(N+1) = 0;
   frf_rfp(N+2:2*N) = conj(frf_rfp(N:-1:2));

%--- Calculating the Impulse Response Function from the FRF Inverse ---%
irf = real(ifft(frf));
irf_rfp = real(ifft(frf_rfp));

%--------- Calculating the Residual ---------%
%-- FRF --%
Residual = frf(x_frm1:W_TR) - frf_rfp(x_frm1:W_TR);
%-- IRF --%
ResidualT = irf_rfp - irf;

%--------- Plotting ---------%
```

```
figure(2)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)),f(x_frm1:W_TR),abs(frf_rfp(x_frm1:W_TR)))

figure(3)
plot(f(x_frm1:W_TR),angle(frf(x_frm1:W_TR)),f(x_frm1:W_TR),angle(frf_rfp(x_frm1:W_TR)))

figure(4)
plot(f(x_frm1:W_TR),imag(frf(x_frm1:W_TR)),f(x_frm1:W_TR),imag(frf_rfp(x_frm1:W_TR)))

figure(5)
subplot(2,1,1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
subplot(2,1,2)
plot(f(x_frm1:W_TR),abs(Residual))

figure(6)
t = linspace(0,1/df,2*N);
plot(t,irf,t,irf_rfp)

figure(7)
plot(t,(ResidualT))

diary on               % Turns diary ON


%--------- Displaying Results ---------%
Residues_and_Poles_rfp = [ R_rfp P_rfp ]
format long g
Natural_freq_Damping_ratio_rfp = [ Fn_rfp Damp_ratio_rfp ]

%--------- Calculating & Displaying the Standard Deviation ---------%
% Frequecy domain (Residual)
Curvefit_Standard_deviation_FRF = sqrt((Residual'*Residual)/length(Residual))

% Time domain (ResidualT)
Curvefit_Standard_deviation_IRF = sqrt((ResidualT'*ResidualT)/length(ResidualT))

% Around each natural frequency in the FRF
stdn = menu('Standard Deviation around each nat. freq.','Yes','No');
if stdn == 1
   std_nat_freq
end

diary off                                  % Turns diary off

sprintf(' The Output file (results) is %s_%s_results.m',method,data)

else
   error('Can not process data')
   disp('The FRF data needs to be in three columns')
   disp('   - First column: frequency')
   disp('   - Second column: Real part')
   disp('   - Third column: Imaginary part')
end
```

Hilbert Envelope Method

```
clear
clc
close all hidden
format long g
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ls *dat
method = 'hil';
disp('Write between quotes the name of the selected data (do not write extension)')
data = input('');

frf = load (sprintf('%s.dat',data));

[r,c] = size(frf);
```

```
if c == 3

diary (sprintf('%s_%s_results.m',method,data))

disp('Hilbert Envelope Method')
sprintf('\nSelected data is: %s\n',data)

f = frf(:,1);
df = f(3) - f(2);
frf = frf(:,2) + i*frf(:,3);
W_TR = length(frf);
N = W_TR;

%----------- Truncation in the FRF -----------%
trf = menu('Is this FRF truncated in the frequency domain?','Yes','No');
if trf == 1
   spl = input('How many spectral lines were truncated? ');
   frf(N+spl) = 0;
   N = length(frf);
   disp(sprintf('FRF with %d truncated spectral lines',spl))
end

% Before getting to this point I need to select the data
% and need freq. matrix

%----------- Specifying the freq range of curve fit -----------%
wind = menu('Do you want to specify the freq. range for the curve fit?','Yes','No');
if wind == 1
   disp('Frequency Range Specified')
   specify = menu('How do you want to specify the freq. range?','Point on Graph','Type
it');
   if specify == 1
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (minimum frequency)')
      [x_frm1,y]=ginput(1);
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      title('Select the first point (maximum frequency)')
      [W_TR,y]=ginput(1);
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq
= %8.4g',x_frm1,W_TR)
   else
      figure(1)
      semilogy(f(1:W_TR),abs(frf(1:W_TR)))
      x_frm1 = input('Minimum Frequency (Hz): ');
      W_TR = input('Maximum Frequency (Hz): ');
      sprintf('The selected frequency range is:\n\tMinimum freq = %d\n\tMaximum freq =
%d',x_frm1,W_TR)
   end

   % Isolating the Frequency Range
      x_frm1 = round(x_frm1/df + 1);
      W_TR = round(W_TR/df + 1);
      frf_F1 = zeros(x_frm1-1,1);                  % Putting zeros before the isolated
FRF components
      frf_F1(x_frm1:W_TR) = frf(x_frm1:W_TR);        % Isolated FRF components
      frf_F1(W_TR+1:N) = zeros(N-(W_TR),1);          % Putting zeros after the isolated
FRF components
      % Adding the conjugate components to the FRF
         frf_F1(N+1) = real(frf_F1(N));
         frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));
         frf = frf_F1;
         clear frf_F1

else
   % Adding the conjugate components to the FRF
   frf(N+1) = real(frf(N));
   frf(N+2:2*N) = conj(frf(N:-1:2));
   x_frm1 = 1;
```

```
end


figure(1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
%%%%%%%%%%%%%%%%%%%%%%%%

dof = input('How many DOF?: ');                              %%%  D  O  F

%-- Time parameters --%
   t = linspace(0,1/df,2*N);
   dt = t(2)-t(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------- Processing Data --------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for n = 1:dof
   figure(1)
   semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
   sprintf('Type natural frequency (Hz) # %d',n)
   fn(n) = input('');
   disp('')
   %---------- Selecting in which way the freq. range will be specified ----------%
   specify = menu('How do you want to specify the freq. range?','Range of window','Point
on Graph','Type it');
   if specify == 1
      % (+-) Value for the isolating window centered at the natural freq
      R = input('Type the range (spectral lines) of the window (+ -): ');
      % Isolate specified freq range
      Comp_f = abs(round(fn(n)/df + 1));        % Componenet in 'f' where the nat freq is
located
      sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum freq
= %8.4g',f(Comp_f-R),f(Comp_f+R))

      frf_F1 = zeros(Comp_f-R-1,1);    % Putting zeros before the isolated FRF components
      frf_F1(Comp_f-R:Comp_f+R) = frf(Comp_f-R:Comp_f+R);     % Isolated FRF components
      frf_F1(Comp_f+R+1:N) = zeros(N-(Comp_f+R),1);     % Putting zeros after the
isolated FRF components

   elseif specify == 2 | specify == 3
      if specify == 2
         figure(1)
         semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
         title('Select the first point (minimum frequency)')
         [x_frmh1,y]=ginput(1);
         figure(1)
         semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
         title('Select the first point (maximum frequency)')
         [x_frmh2,y]=ginput(1);
         sprintf('The selected frequency range is:\n\tMinimum freq  = %8.4g\n\tMaximum
freq = %8.4g',x_frmh1,x_frmh2)
      else
         figure(1)
         semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
         x_frmh1 = input('Minimum Frequency (Hz): ');
         x_frmh2 = input('Maximum Frequency (Hz): ');
         sprintf('The selected frequency range is:\n\tMinimum freq = %8.4g\n\tMaximum
freq = %8.4g',x_frm1,x_frm2)
      end
      % Isolating the Frequency Range
      x_frmh1 = round(x_frmh1/df + 1);
      x_frmh2 = round(x_frmh2/df + 1);
      frf_F1 = zeros(x_frmh1-1,1);                        % Putting zeros before the
isolated FRF components (mult by mean of irf)
      frf_F1(x_frmh1:x_frmh2) = frf(x_frmh1:x_frmh2);    % Isolated FRF components
      frf_F1(x_frmh2+1:N) = zeros(N-(x_frmh2),1);        % Putting zeros after the
isolated FRF components (mult by mean of irf)
      % Adding the conjugate components to the FRF
         frf_F1(N+1) = real(frf_F1(N));
         frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));
```

```
   end

   % Adding the conjugate components to the FRF
      frf_F1(N+1) = real(frf_F1(N));
      frf_F1(N+2:2*N) = conj(frf_F1(N:-1:2));

   % Separating the DFT in order to change the phase
      for nn = 2:N;
         frf_F1_j(nn) = j*frf_F1(nn);          % Mult by the complex number j
      end

         for nn = N+2:2*N;
            frf_F1_j(nn) = -j*frf_F1(nn);         % Mult by the complex number -j the
conjugate part of the DFT
        end

   % Impulse Response Function of the isolated FRF
      irf_F1 = (ifft(frf_F1));
      irf_F1 = conj(irf_F1');

   % Hilbert transform (quadrature function).  The phase was changed previously
      hil_F1 = (ifft(frf_F1_j));

   % Envelope of the decaying signal
      env_F1 = irf_F1 - j*hil_F1;
      env_F1_dB = 20*log10(abs(env_F1));         % dB scale for the envelope's magnitude

   % Least Square Method to calculate slope, and then damping ratio
      figure(n + 1)
      plot(t,env_F1_dB)
      title('Select two points in the time domain (x axis)')
      [x,y]=ginput(2);
      [np] = round(x/dt + 1);
      t1 = t(np(1):np(2));
      env_F1_dB1 = env_F1_dB(np(1):np(2));
      m = length(t1);                      % Calculates the amount of points data
      sum_x = sum(t1);                     % Summatory of the t points (x components)
      sum_y = sum(env_F1_dB1);             % Summatory of the env_dB points (y components)
      sum_x_sq = dot(t1,t1);               % Summatory of the square value of each t
points (x components)
      sum_xy = dot(t1,env_F1_dB1);         % Summatory of the multiplication of t and
env_dB points (x and y components)
      LQ1 = [m sum_x; sum_x sum_x_sq];
      LQ2 = [sum_y ; sum_xy];
      LQ3 = inv(LQ1) * LQ2;
      slope = LQ3(2);
      % Calculating damping ratio
      damping_F1(n) = -slope/(20*log10(exp(1)))         ;
      damping_ratio_F1(n) = damping_F1(n)/(fn(n)*(2*pi))  ;

      % Plotting the Hilbert Process in four plots (for each mode)
      figure(n + 1)
      subplot(2,2,1)
      plot(f,abs(frf_F1(1:N-spl)))
      title(sprintf('Isolated Natural Frequency # %d',n))

      subplot(2,2,2)
      plot(t,real(irf_F1))
      title('Impulse Response Function')

      subplot(2,2,3)
      plot(t,abs(env_F1))
      title('Envelope of the Impulse Response Function')

      subplot(2,2,4)
      plot(t,env_F1_dB)
      title('dB Scale of the Envelope')
end
%--------- Calculatin the Exp. IRF ---------%
irf = real(ifft(frf));
```

```
%--------- Calculating the eigenvalues ---------%
V_hil = exp((-damping_ratio_F1 + i*sqrt(1 - damping_ratio_F1.^2)).*fn*2*pi*dt);
rr = length(V_hil);
V_hil(rr+1:2*rr) = conj(V_hil);

%--------- Calculating eigenvector ---------%
L = length(t);
for r = 0:L-1
   Va_hil(r+1,:) = [conj(V_hil).^r ];
end

Ar_hil = (inv(conj(Va_hil')*Va_hil)*conj(Va_hil')*(irf));

%---------- Calcualting the IRF Curve Fit ----------%
x_hil = Va_hil*Ar_hil;

%---------- Calcualting FRF Curve Fit ----------%
frf_hil = fft(x_hil);

%--------- Calculating the Residual ---------%
%-- FRF --%
Residual = frf(x_frm1:W_TR) - frf_hil(x_frm1:W_TR);
%-- IRF --%
ResidualT = real(x_hil) - irf;

%--------- Plotting ---------%
figure(dof + 2)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)),f(x_frm1:W_TR),abs(frf_hil(x_frm1:W_TR)))

figure(dof + 3)
plot(f(x_frm1:W_TR),angle(frf(x_frm1:W_TR)),f(x_frm1:W_TR),angle(frf_hil(x_frm1:W_TR)))

figure(dof + 4)
plot(f(x_frm1:W_TR),imag(frf(x_frm1:W_TR)),f(x_frm1:W_TR),imag(frf_hil(x_frm1:W_TR)))

figure(dof + 5)
subplot(2,1,1)
semilogy(f(x_frm1:W_TR),abs(frf(x_frm1:W_TR)))
subplot(2,1,2)
plot(f(x_frm1:W_TR),abs(Residual))

figure(dof + 6)
plot(t,real(x_hil),t,irf)

figure(dof + 7)
plot(t,ResidualT)

%--------- Displaying Results ---------%
Residues_Eigenvalues_hil = [ Ar_hil conj(V_hil') ]
Natural_freq_Damping_ratio_hil = [ fn' damping_ratio_F1' ]


%--------- Calculating & Displaying the Standard Deviation ---------%
% Frequecy domain (Residual)
Curvefit_Standard_deviation_FRF = sqrt((Residual'*Residual)/length(Residual))

% Time domain (ResidualT)
Curvefit_Standard_deviation_IRF = sqrt((ResidualT'*ResidualT)/length(ResidualT))

% Around each natural frequency in the FRF
stdn = menu('Standard Deviation around each nat. freq.','Yes','No');
if stdn == 1
   std_nat_freq
end


diary off                                  % Turns diary off

sprintf(' The Output file (results) is %s_%s_results.m',method,data)
```

```
else
   error('Can not process data')
   disp('The FRF data needs to be in three columns')
   disp('   - First column: frequency')
   disp('   - Second column: Real part')
   disp('   - Third column: Imaginary part')
end
```

Standard Deviation

```
clear x_frm1
disp('Standard Deviation around each nat. freq.')
disp([sprintf('\n')])
np = input('   How many modes were identified? ');
disp([sprintf('\n')])
for r = 1:np
   NT = input(sprintf('\tType natural freq # %d (integer number): ',r));
   x_frm1 = round((NT - min(f))/df + 1);      % Index Matrix of Nat. Freq.
   disp([sprintf('\t\t') 'Standard Deviation = ' ...
         num2str(sqrt((Residual(x_frm1-10:x_frm1+10)'*Residual(x_frm1-
10:x_frm1+10))/length(Residual(x_frm1-10:x_frm1+10))))])
   disp([sprintf('\n')])
end
```

Vitta

My name is Angel Moises Iglesias. I was born in Santurce, Puerto Rico and raised in Carolina, Puerto Rico. Actually, my parents, Jose Manuel Iglesias and Magda Iglesias, live in Carolina, Puerto Rico. I am the youngest of three brothers, the eldest, Jose Manuel Iglesias, the father of 2 years old son, Gabriel Iglesias, and the other, Carlos Juan Iglesias, father a 6 year old daughter, Carina Iglesias.

After graduating from high school, Colegio Espiritu Santo, I received a Bachelo of Science in Mechanical Engineering from the Polytechnic University of Puerto Rico at Hato Rey, Puerto Rico in June of 1998. Then, I started my mechanical engineering graduates studies at Virginia Tech in August of 1998. On June of 2000 I completed my requirements for my Master of Science in Mechanical Engineering.

I am going to start working in July of 2000 at General Electric Aircraft Engines in Cincinnati, Ohio. As part of my most important plans, I am going to get married on May 26, 2001 with Frances Bernier.