Damping and resonance analysis of a composite cantilever beam

David Friesorger

Degree Thesis
Plastics Technology
2014

DEGREE THESIS	
Arcada	
Degree Programme:	Plastics Technology
Identification number:	
Author:	David Friesorger
Title:	Damping and resonance analysis of a cantilever beam
Supervisor (Arcada):	Rene Hermann
Commissioned by:	

This thesis was written in order to study the possibility of obtaining certain mechanical properties, more specifically damping coefficient and resonance frequency, of a material through a non destructive testing method. The testing method implemented was that of a cantilever composite beam, made of carbon fiber, vibrating after an initial displacement. A piezo electric transducer is attached to the beam and a computer’s sound card via a 2.5 mm audio jack, and used to collect vibrational data, which could then be analysed using a Fast Fourier Transform (FFT) and time data analysis through the use of MATLAB. In the case of this study it was found that for a light weight and stiff cantilever beam, the FFT method of analysis provides a resonance frequency with 10 times more certainty than the time domain analysis. The FFT method also provides a damping coefficient which has 10% less variance than that of the time domain analysis.

Keywords:	damping, resonance, frequency, Fast Fourier Transform, vibrational analysis, cantilever beam, non-destructive testing
Number of pages:	45
Language:	English
Date of acceptance:	

CONTENTS

1 Introduction 7

1.1 Background 7

1.2 Objectives 7

2 Literature Review 8

2.1 Definition of terms 8

2.1.1 Piezoelectric Transducer 8

2.1.2 Stress (σ) 8

2.1.3 Strain (ε) 9

2.1.4 Young’s Modulus (E) 10

2.1.5 Damping 11

2.1.6 Damping Ratio (ζ) 11

2.1.7 Natural frequency (ω_o) 11

2.1.8 Damping in FFT analysis ($\Delta\omega$) 12

2.1.9 Damping in time data analysis ($\zeta\omega_0$) 12

2.1.10 MATLAB 12

2.1.11 Audacity 13

2.1.12 Fast Fourier Transform (FFT) 13

3 Data collection method 14

3.1 Sample rate 16

3.2 Signal/Noise Flow 17

3.2.1 Setup (cantilever beam + clamping device) 18

3.2.2 Piezo-electric Transducer 18

3.2.3 Sound card 19

3.2.4 Power Source 19

3.2.5 Audacity 20

4 Analysis 23

4.1 MATLAB frequency data analysis 23

4.1.1 MATLAB Fast Fourier Transform 24

4.2 Obtaining resonance frequency with FFT 25

4.2.1 MATLAB Peak Find 28

4.2.2 MATLAB Full Width at Half Maximum 33

4.3 MATLAB Time Data Analysis 37

4.3.1 Obtaining resonance frequency using time data 37

4.3.2 Obtaining damping coefficient from time data 41

5 Results 44

6 Discussion AND Conclusion.. 45

7 Suggestion For Further Work 46

References 47

8 Appendices 48

Figures

Figure 1 Visual representation of mechanical stress 8
Figure 2 Visual representation of strain........ 9
Figure 3 Stress strain graph depicting Young's Modulus 10
Figure 4 Sound data obtained displayed as amplitude vs time [s] (after having applied low and high pass filters)........ 13
Figure 5 Frequency data converted from orignal sound data displayed as intensity vs.frequency........ 14
Figure 6 Drawing of experimental setup 15
Figure 7 Flow of data from source to analysis 17
Figure 8 Starting setup for experimental data collection. With important settings highlighted in yellow........ 21
Figure 9 Low pass filter options in Audacity 21
Figure 10 Exporting of single signal chosen in Audacity, into usable format by MATLAB. 22
Figure 11 Graph displaying degree of error 24
Figure 12 MATLAB script used to implement Fast Fourier Transform, and display graphically 25
Figure 13 Example of MATLAB graphical representation of data after applying FFT.27
Figure 14 MATLAB script used to locate points at peaks 28
Figure 15 Graphical representation of **I=find(spectra>14)** command 29
Figure 16 Window of MATLAB after using the **I=find(spectra>14)** command........ 30
Figure 17 Command window of MATLAB displaying the use of the **spectra (:)** command 31
Figure 18 Graphical representation of the **spectra (:)** command........ 31
Figure 19 Command window of MATLAB portraying the frequency at a specific data point........ 32
Figure 20 MATLAB script used to determine Full Width at Half Maximum (ω_o)........ 33
Figure 21 Side by side graphs showing the result of of selected_d=(spectra(:) command. 34
Figure 22 Graph displaying the function of **Plot(selected_d-max(selected_d)/2);.**... 35
Figure 23 MATLAB script used to obtain resonance frequency from time data......... 37

Figure 24 Graphical display of peak finding in time domain, displayed as amplitude vs. time. 38
Figure 25 MATLAB command window displaying results of [y,i] function. 39
Figure 26 MATLAB script used to determine damping coefficient from time data. 42
Figure 27 Method used of finding damping coefficient from time data in MATLAB. 43

Tables

Table 1 Settings used in Audacity 20
Table 2 Results from data analysis of 50 signals 44

1 INTRODUCTION

1.1 Background

The purpose of this thesis is to use vibrational data to extract a material's mechanical properties. Nondestructive testing methods encompass a large variety of analysis techniques to be used in industry to evaluate material properties without altering the piece being tested. Among these methods is acoustic emission testing. This method of testing relies on an applied force to a system to produce elastic energy which propagates throughout the system as an elastic wave. The wave is then monitored by one or more transducer placed throughout the system, which records the signal in a readable data format. This method of testing can be used to locate problem areas in existing structures, to monitor the condition of structures and the effects of ageing, and to determine material and structural properties. This thesis will use this method of testing to determine wether time data analysis or Fast Fourier analysis will provide results with less variance.

1.2 Objectives

There are 5 objectives which this thesis seeks to fulfill;

- To optimize the data collection method, through the use of filters both hardware and software.
- To obtain accurate frequency data of a composite beam using Fast Fourier transform analysis.
- To obtain accurate damping ratio data of a composite beam using 2nd order time analysis.
- To determine the most precise analysis methods with which to obtain damping and frequency data.
- To extract mechanical properties of a material, specifically natural frequency and damping coefficient, through analysis of vibrational data obtained.

2 LITERATURE REVIEW

2.1 Definition of terms

2.1.1 Piezoelectric Transducer

This testing method relies on a piezoelectric ceramic transducer in order to convert the vibration of the piece in the experiment into an electric signal to be read. The piezoelectric transducer converts mechanical vibrations into electrical energy, and electrical pulses into mechanical vibrations(NDT Resource Center 1996.)

2.1.2 Stress (σ)

Stress is the force per unit area on a body. It is measured in pascals (Pa) or newtons per metre squared $\frac{N}{m^2}$. The stress of the rod in figure 2 is found by dividing the force by the cross sectional area of the cylinder (Ross p.54-57.)

$$\sigma = \frac{F}{A} \qquad \text{[Eq. 1]}$$

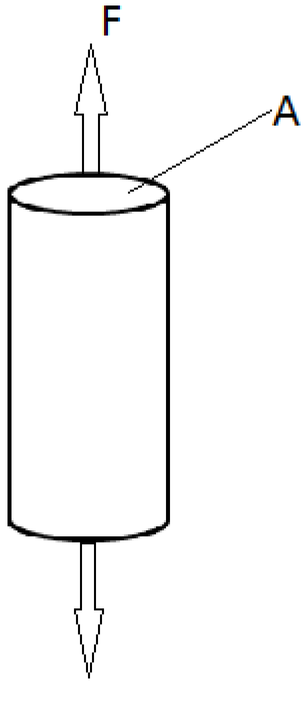


Figure 1 *Visual representation of mechanical stress*

2.1.3 Strain (ε)

Strain is the ratio of total deformation *ΔL* of an object to that of the initial length *L*. It is a unit less term expressed by the formula(Ross p.54-57.):

$$\varepsilon = \frac{\Delta L}{L} = \frac{\ell - L}{L} \qquad \text{[Eq. 2]}$$

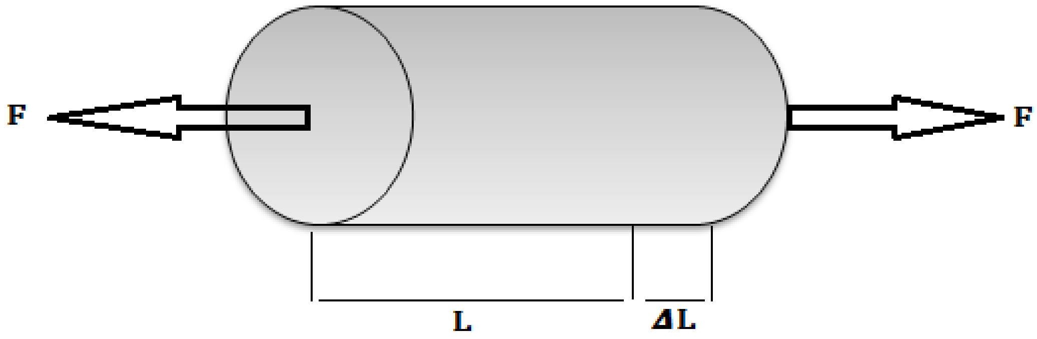


Figure 2 *Visual representation of strain.*

2.1.4 Young’s Modulus (E)

Also known as elastic modulus, Young’s modulus is a measure of stiffness of a material. The Young’s modulus or E of a material is the ratio of stress along an axis over the strain, under the range in which Hooke’s Law holds, and is measured in Pascals. The slope of the stress-strain curve at any point is known as the tangent modulus. The tangent modulus of the linear portion of a stress strain graph of an elastic isotropic material is the Young’s modulus (Ross p.56,59.)

$$E = \frac{tensile\ stress}{strain} = \frac{\sigma}{\varepsilon} \qquad \text{[Eq. 3]}$$

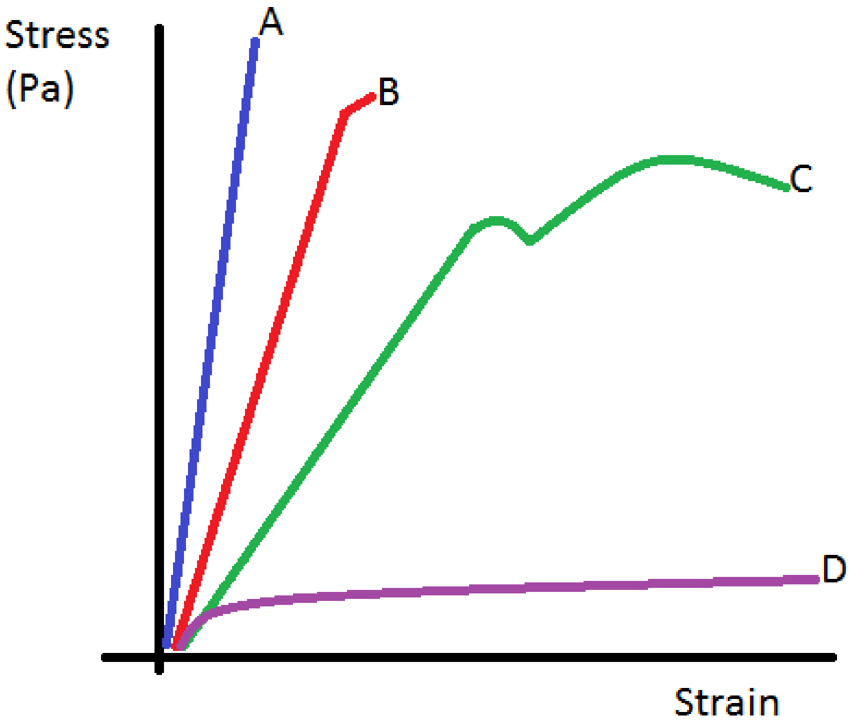


Figure 3 *Stress strain graph depicting Young’s Modulus*

In the above graph, the elastic region in each materials curve is where the Young’s Modulus is determined. As can be seen in the plastic materials curve (D), the elastic region is very small.

2.1.5 Damping

Damping in mechanical engineering is the influence which causes an oscillatory system to either not oscillate or reduce a system's amplitude. In the case of this thesis we will deal with an underdamped system. In an underdamped system oscillations are achieved with the amplitude gradually reducing to zero, as opposed to a critically or over damped system, where the system will return to equilibrium without oscillating (Knight p.395-399.)

In an underdamped situation the system will oscillate at the natural damped frequency $\omega_{d,}$ where the natural damped frequency is a function of the natural frequency and the damping ratio (Knight p.395-399.)

$$x(t) = e^{\zeta \omega o t}(Acos(\omega dt) + Bsin(\omega dt)) \text{ [Eq. 4]}$$

$$\omega_d = \omega_o\sqrt{1 - 2\zeta^2} \qquad \text{[Eq. 5]}$$

2.1.6 Damping Ratio (ζ)

Damping ratio is a dimensionless measure which describes how oscillations in a system decay. It is defined as the ratio of the damping coefficient to the critical damping coefficient. $\zeta = \frac{c}{c_c}$ [Eq. 6] Where $c_c = 2m\sqrt{k/m}$ [Eq. 7] and k is the spring constant and m is the mass (Knight p.395-399.)

2.1.7 Natural frequency (ω_o)

Natural frequency is the frequency at which a system oscillates in the absence of an outside or damping force. The natural frequency ω_o is defined as follows.

$$\omega_o = \sqrt{k/m} = \alpha\sqrt{\frac{EI}{\rho AL^4}} \quad \text{[Eq. 8]}$$

Where E is Young’s modulus, I is the moment of inertia of a beam, A is the cross sectional area, and L is the length of the beam (Knight p.395-399.)

2.1.8 Damping in FFT analysis (*Δ*ω)

In the case of the FFT, finding damping data is a result of finding the bandwidth of a resonance peak. The terms are related as follows. Where *c* is the damping coefficient, *k* is the spring constant, and *m* is the mass.

$$\Delta\omega = \sqrt{\frac{k}{m}} * \frac{c}{\sqrt{mk}} = \frac{c}{m}. \quad \text{[Eq. 9]}$$

2.1.9 Damping in time data analysis (ζω$_0$)

In the case of the time data analysis, the damping data is related to the damping ratio ζ and the natural frequency ω_o as follows.

$$\zeta\omega_0 = \frac{c}{2\sqrt{mk}} * \sqrt{\frac{k}{m}} = \frac{c}{2m} \quad \text{[Eq. 10]}$$

2.1.10 MATLAB

MATLAB is a numerical computing environment and a programming language, which was developed by MathWorks in the 80’s. MATLAB uses matrices to create plot functions and run numerical algorithms. MATLAB allows large data sets to be analysed and can assist researchears by displaying functions applied to large data sets both numerically and graphically (MATLAB 1994.) In this thesis MATLAB is used to apply a Fast Fourier Transform to the experimental data set, in order to find the resonance frequency of the tested composite laminate.

2.1.11 Audacity

Audacity is an open source digital audio editor and recording software application. Audacity allows for recording and a large range of post-processing of audio (Audacity 2000.) In this thesis Audacity is used to record the experimental samples, and also to apply low and high pass filters to the experimental data. The program is then used to convert the audio from a .wav a, to a .mat text file which is readable by MATLAB.

2.1.12 Fast Fourier Transform (FFT)

A Fast Fourier Transform converts time data to frequency data. In the case of this thesis it is used to to transform the time data of the laminates vibrational signal into frequency data viewable through MATLAB. Below is an example of the experimental time data compared to its frequency data obtained through the use of FFT.

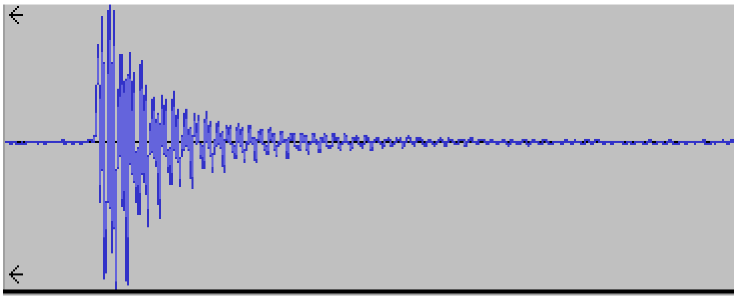


Figure 4 *Sound data obtained displayed as amplitude vs time [s] (after having applied low and high pass filters)*

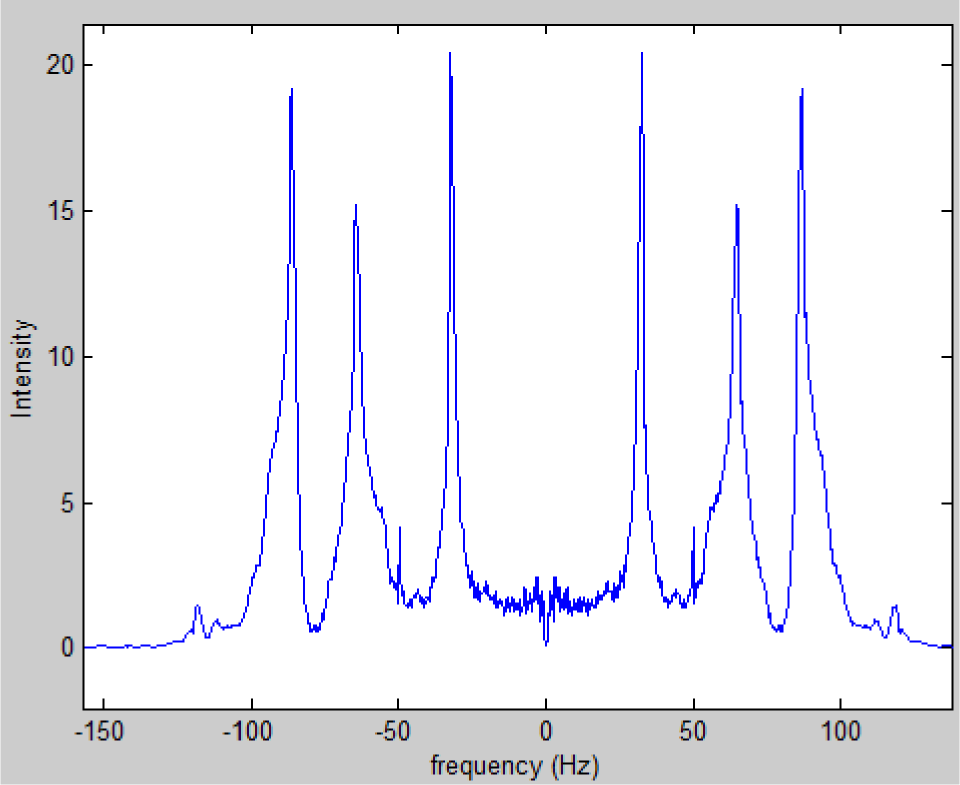


Figure 5 *Frequency data converted from orignal sound data displayed as intensity vs.frequency.*

3 DATA COLLECTION METHOD

In this experiment, data is collected via a piezoelectric transducer which is attached with a 2 bar bracket brace to a composite beam as shown in figure 6. The piezoelectric transducer is then connected with a high pass filter in series, to a computers sound card through standard 2.5 mm audio jack. The composite beam is then stabilized with a large vice in order to reduce background noise from the vibrating composite member. The program *Audacity*, which records the audio data inputted from the transducer is then started. The experiment is then carried out by displacing the beam a certain distance, in order to excite a harmonic oscillation. The piezoelectric transducer then converts the mechanical vibrations incited from the motion into an electrical signal, and is recorded into the Audacity program. Due to the presence of high frequency background noise, it is necessary to use a low pass filter within the Audacity program. A low pass filter is a function used to remove all sound or noise above a certain frequency, in the case of this experiment everything above 100 Hz is removed as it is all background noise and static from the sound card being used for recording. After the recording of each oscillation, the desired data set from each sample is selected and saved in a format appropriate to be used with Microsoft Excel or MATLAB. In the case of MATLAB the file is saved as .mat file.

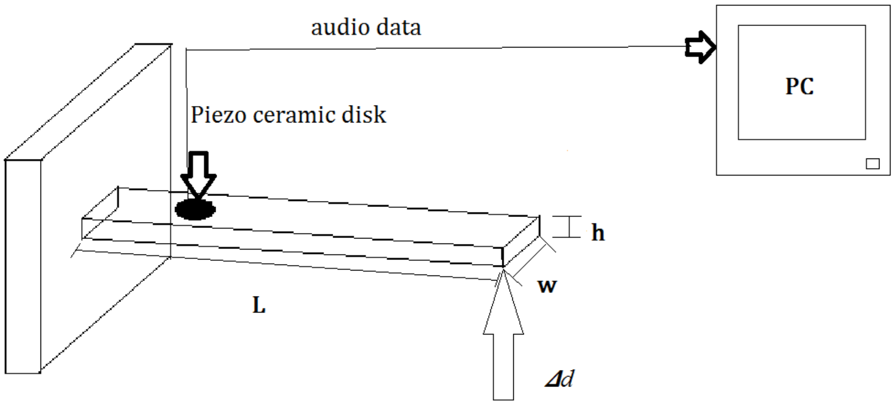


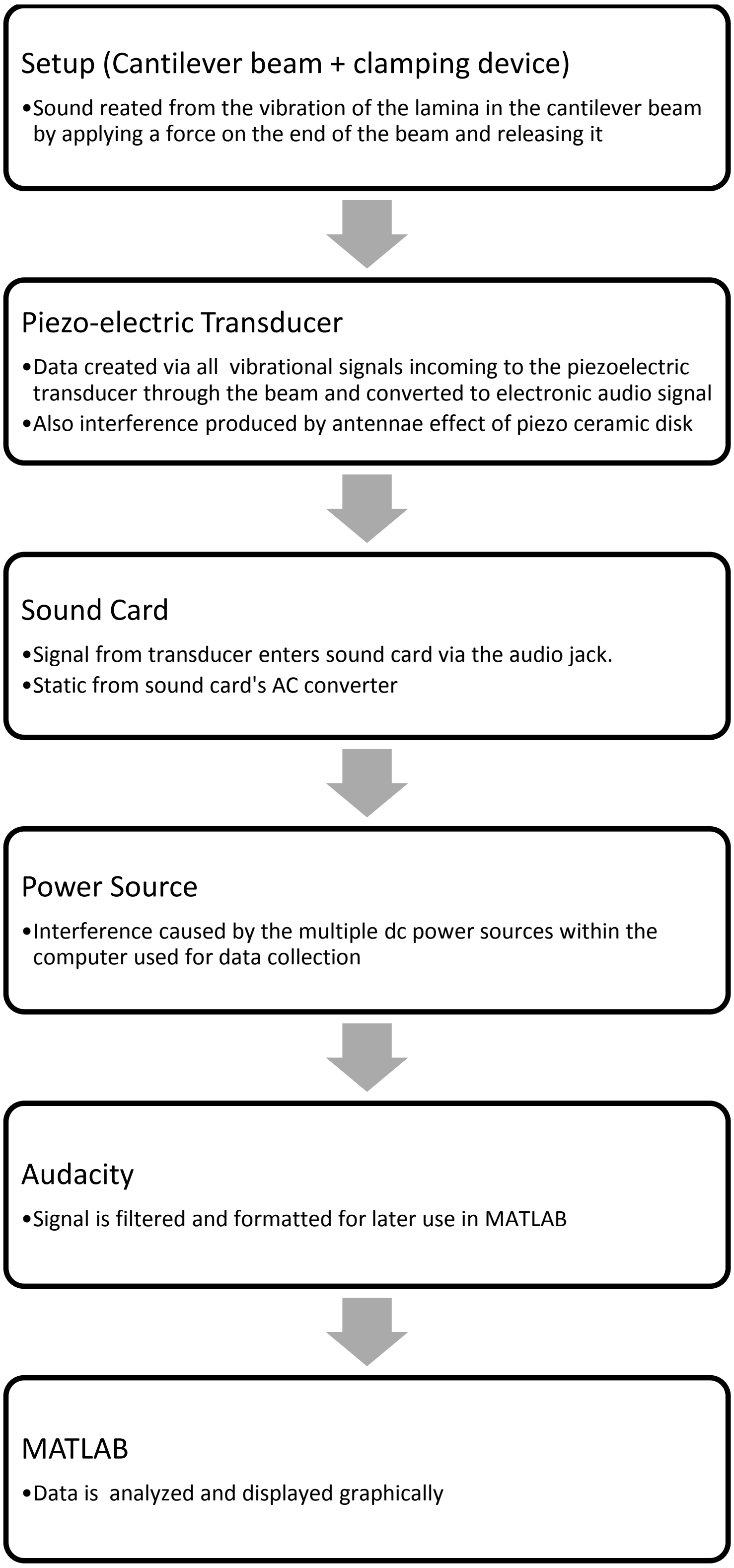
Figure 6 *Drawing of experimental setup*

Above is a diagram showing the experimental setup. The beam with length *L*, width *w*, and height *h,* is secured at one end and displaced an amount *Δd*. There is then a piezo ceramic disk attached to the beam, which is then wired and plugged into the audio jack on the pc’s sound card.

3.1 Sample rate

When determining the sample rate, it is necessary to choose a sampling rate which will prove a sufficient number of points per wave. It is important to note that a minimum of 2 points per wave is required to recreate the data graphically. Thus it was determined that 10 points per wave would be sufficient. As it turns out that the smallest sample rate available in audacity is 8000 points per second, it was chosen, and it happens to provide a greater number of points than 10 per wave, thus it was deemed sufficient.

3.2 Signal/Noise Flow



Setup (Cantilever beam + clamping device)

- Sound reated from the vibration of the lamina in the cantilever beam by applying a force on the end of the beam and releasing it

Piezo-electric Transducer

- Data created via all vibrational signals incoming to the piezoelectric transducer through the beam and converted to electronic audio signal
- Also interference produced by antennae effect of piezo ceramic disk

Sound Card

- Signal from transducer enters sound card via the audio jack.
- Static from sound card's AC converter

Power Source

- Interference caused by the multiple dc power sources within the computer used for data collection

Audacity

- Signal is filtered and formatted for later use in MATLAB

MATLAB

- Data is analyzed and displayed graphically

Figure 7 *Flow of data from source to analysis*

At each step in the total data collection system the original signal is mixed with noise. Physical background noise being sound originating from somewhere other than the vibrating part, which influences the data collected through the parts vibration. Electrical background noise being static or electrical signals from the computers parts themselves, which mixes with or in some cases covers the signal to be collected. Each issue was addressed and overcome in the completed data collection process.

3.2.1 Setup (cantilever beam + clamping device)

As the data to be collected for the purpose of this thesis originates from a vibrating composite laminae, the original signal to be collected must be pure. In order to reduce mechanical background noise in the first step of the data collection process the experiment was performed in a quiet room with no sound influence other than that of the vibrating part. The part was also placed in a vice which is used to test audio parts, in order to reduce any secondary vibration of the system due to the initial excitation. The noise that enters does not effect the outcome of the data as it can be filtered at various stages of the process.

3.2.2 Piezo-electric Transducer

At the sensor stage, the mechanical signal caused by the vibration of the lamina is converted into an electrical signal which is delivered to the computer by being plugged into the audio jack of the computer. As the piezo-electric device acts as an antanna for surrounding electronic signals, there is also a form of electronic background noise produced from the device, as it receives various signals. Including signals from the various powers sources inside the computer.

3.2.3 Sound card

This is an important step as the quality of the sound card being used is not ideal, which means that a great deal of background noise is created and mixed with the final signal which is read via the Audacity program. The sound card and Audacity program together convert the audio signal into a readable data format. At this stage it is important to determine the importance of the data collected, by determining which parts of the signal were actual data, and which were noise to be removed with the use of high and low pass filters. This was done by using an oscilloscope, as an oscilloscope collects a signal with no non experimental noise, to determine which part of the experimental signal was being produced by the actual vibration of the cantilever beam, thereby determining which signals were to be removed with the use of high and low pass filters within the Audacity program.

3.2.4 Power Source

As a European desktop computer uses a power source that has a utility frequency of 50 Hz, this presents a specific interference that shows up on every signal taken from this setup. This signal frequency is emitted to the piezo-electric ceramic which acts as an antenna and portrays this signal in the resultant collected data. As shown in the results, this signal does not interfere with the experiment as the 50 Hz frequency does not collide with the experimentally found natural frequency of the vibrating composite laminae. However it is important to note the appearance of the phenomena in the recorded signals and resultant graphs.

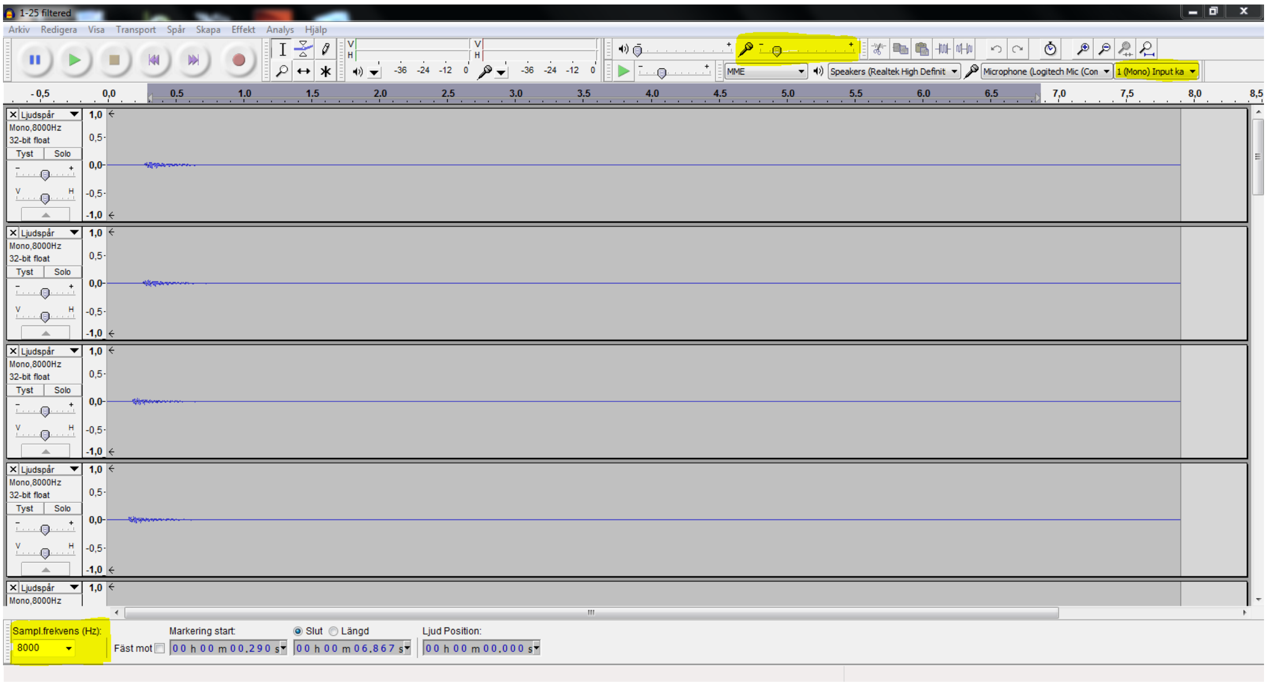
3.2.5 Audacity

As the signal is created it is read into the Audacity program. Audacity displays the signal in a graphical format with the signal displayed as a graph of amplitude and time. In Audacity it is also possible to use low pass filters to reduce the background noise and static from the actual signal. It is then possible to select the portion of the signal desired convert it to a format to be read by MATLab.
The settings used in Audacity were as follows.

Table 1 *Settings used in Audacity*

Mic in	**Mono**
Data collection rate	**8,000 points per second**
Volume	**0.2**
Low pass filter	**Rolloff: 48 dB** **Cutoff frequency: 100 Hz**

Each signal obtained is made usable in MATLab by exporting the signal to a **.MAT4** format with the use of the **32 bit float option**.


1-25 filtered
Arkiv Redigera Visa Transport Spår Skapa Effekt Analys Hjälp
MME
Speakers (Realtek High Definit
Microphone (Logitech Mic (Con
1 (Mono) Input ka
Ljudspår
Mono,8000Hz
32-bit float
Tyst Solo

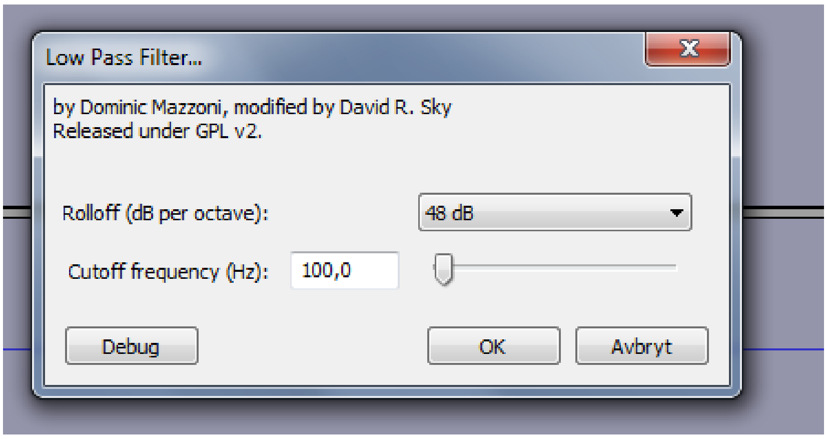
Sampl.frekvens (Hz): 8000

Fäst mot □ Markering start: 00 h 00 m 00,290 s

● Slut ○ Längd 00 h 00 m 06,867 s

Ljud Position: 00 h 00 m 00,000 s

Figure 8 *Starting setup for experimental data collection. With important settings highlighted in yellow.*

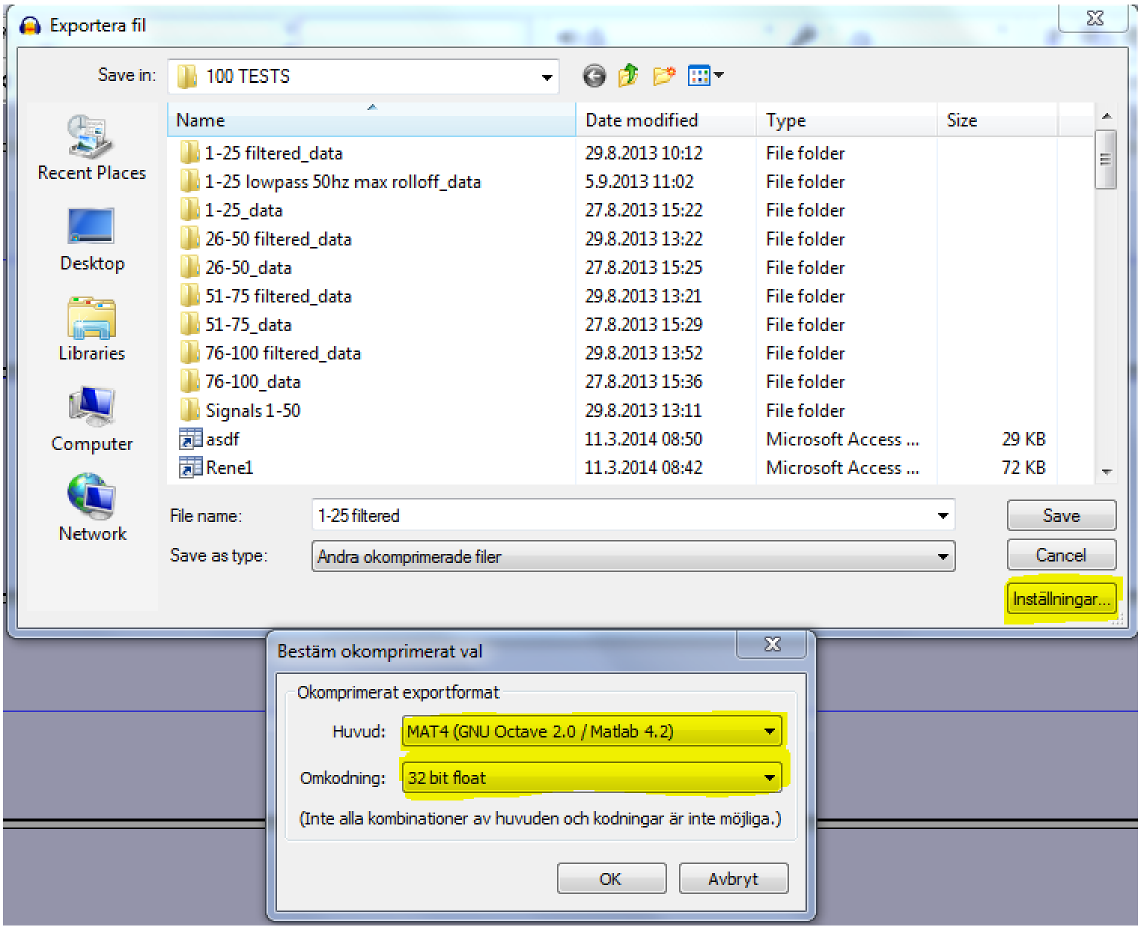

Low Pass Filter...

by Dominic Mazzoni, modified by David R. Sky
Released under GPL v2.

Rolloff (dB per octave): 48 dB

Cutoff frequency (Hz): 100,0

Figure 9 *Low pass filter options in Audacity*



Save in: 100 TESTS

Name	Date modified	Type	Size
1-25 filtered_data	29.8.2013 10:12	File folder	
1-25 lowpass 50hz max rolloff_data	5.9.2013 11:02	File folder	
1-25_data	27.8.2013 15:22	File folder	
26-50 filtered_data	29.8.2013 13:22	File folder	
26-50_data	27.8.2013 15:25	File folder	
51-75 filtered_data	29.8.2013 13:21	File folder	
51-75_data	27.8.2013 15:29	File folder	
76-100 filtered_data	29.8.2013 13:52	File folder	
76-100_data	27.8.2013 15:36	File folder	
Signals 1-50	29.8.2013 13:11	File folder	
asdf	11.3.2014 08:50	Microsoft Access ...	29 KB
Rene1	11.3.2014 08:42	Microsoft Access ...	72 KB

File name: 1-25 filtered

Save as type: Andra okomprimerade filer

Okomprimerat exportformat

Huvud: MAT4 (GNU Octave 2.0 / Matlab 4.2)

Omkodning: 32 bit float

(Inte alla kombinationer av huvuden och kodningar är inte möjliga.)

***Figure 10** Exporting of single signal chosen in Audacity, into usable format by MATLAB.*

4 ANALYSIS

4.1 MATLAB frequency data analysis

As is stated previously, the objective of this thesis is to use both time and frequency data to obtain resonance frequency and damping coefficient, and to compare the accuracy and precision of these two methods. The following is the method used to obtain the desired information while using the frequency data from the implementation of FFT. The method used to extract damping data relies on the following function **df=samplerate/max(size(wavedata));.** As can be seen this determination of df, which is the step of the frequency between each point, is inversely proportional to the size of the data collected, or the number of points collected. As the samplerate (8000) stays the same throughout all the experiments, **“df”** depends solely on the duration of the sample taken. As the material being tested is a stiff, light weight material, the vibrational time of the piece is low, and the only effect we can impose on “**df**” is by changing samplerate.

This is important to note, as it will affect the results when finding the band width at half maximum for the peaks of the FFT data, which is then used to determine damping coefficient. Because the method chosen to find bandwidth at half height relies on *df* and the number of points above the half height line, the accuracy of the result will include a degree of error of +- 1 data point. This error can be seen in the graph below as the points on the right side of the peak are not sufficient to determine the point on the line that intersects 0.

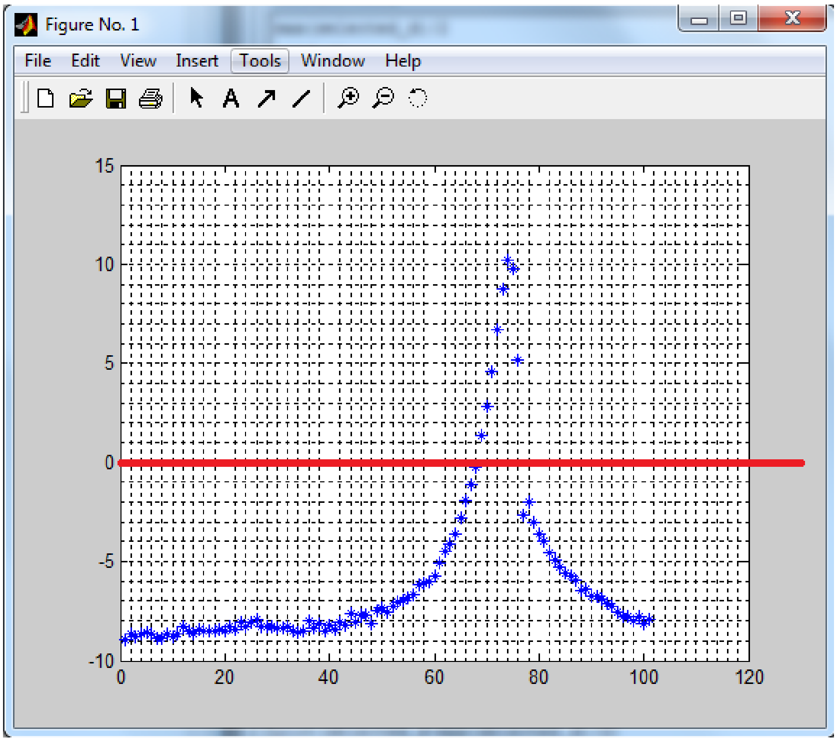
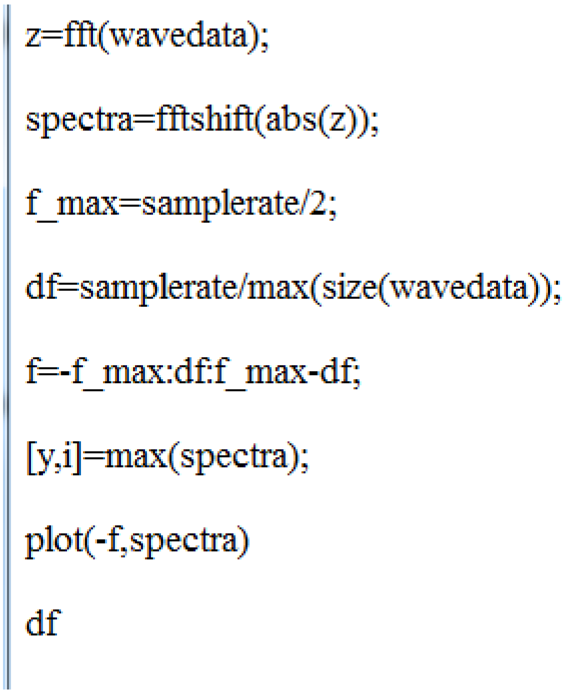


Figure 11 *Graph displaying degree of error*

4.1.1 MATLAB Fast Fourier Transform

MATLAB is the program used in this thesis to analyse the data received from the experiment. Through the use of MATLAB it is possible to apply a Fast Fourier Transform to the sound data, in order to display graphically the resonance frequency of the composite lamina being studied. The script used to implement the FFT and display the data graphically is as follows.

4.2 Obtaining resonance frequency with FFT



```
z=fft(wavedata);

spectra=fftshift(abs(z));

f_max=samplerate/2;

df=samplerate/max(size(wavedata));

f=-f_max:df:f_max-df;

[y,i]=max(spectra);

plot(-f,spectra)

df
```

Figure 12 *MATLAB script used to implement Fast Fourier Transform, and display graphically*

Below, the purpose of each step in the above MATLAB script will be described.

z=fft(wavedata); This operation creates a data set "z", which is the modified data set of x after having applied the FFT.

spectra=fftshift(abs(z)); This operation rearranges the outputs of the fft(x) command by moving the zero-frequency component to the center of the graphical array. This is useful to display the FFT graphically with both sides of 0 shown. This is determined by creating a new data set "spectra."

f_max=samplerate/2; This operation determines the Nyquist frequency of the signal, which in this case is 4000 Hz, it is found by dividing the sample rate by 2.

df=samplerate/max(size(wavedata)); This operation determines the width of a step on the graph. It determines it by dividing the samplerate, which is in this case 8000 points per second, by the size of the “wavedata" set.

f=-f_max:df:f_max-df; This operation determines the range of frequencies displayed and the step size between each plotted point.

[y,i]=max(spectra); This command is used to determine the max point of a data set, in this case the data set “spectra.”

plot(-f/f(i),spectra); This command creates a spectra plot of frequency vs intensity.

df; The final command displays the df or step size of the data set. This is important as it will be used later in order to determine the full width at half maximum of the studied peak, which can then be used to determine damping properties.

After executing the above script in MATLAB on a signal file which has been filtered and exported from the Audacity program, we are presented with a spectra as follows.

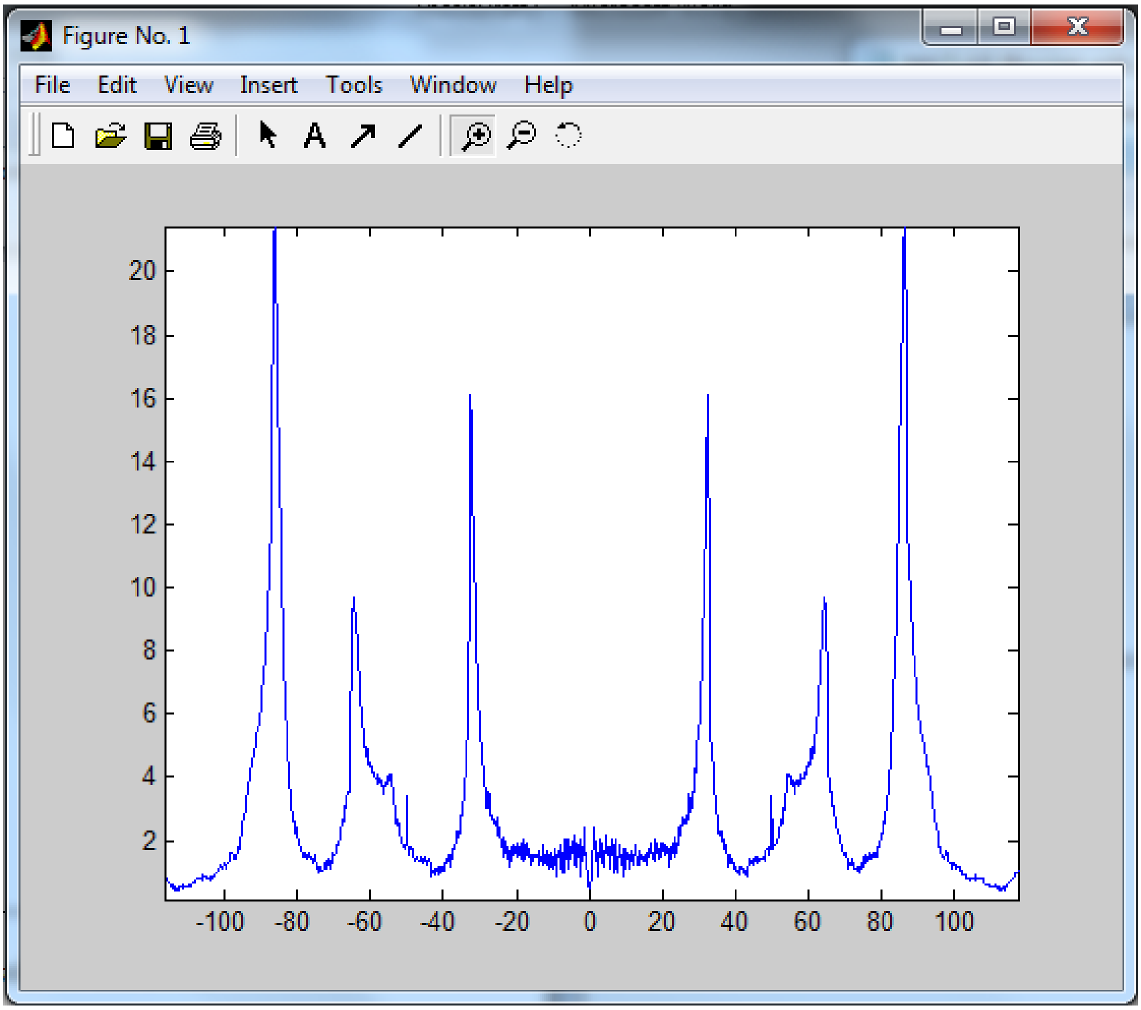


Figure 13 *Example of MATLAB graphical representation of data after applying FFT.*

The Spectra above shows three obvious peaks, one at approximately 32 Hz, 63 Hz, and 85 Hz respectively. Each peak following the initial 32 Hz peak represents a fourier overtone of the 32 Hz peak. There is also minor peak at 50 Hz which represents the aforementioned representation of the European utility power frequency. In the case of this thesis we will only refer to the original peak around the 32 Hz mark, as it is the first mode resonance frequency of the laminate part.

4.2.1 MATLAB Peak Find

After the calculation of the FFT of the experimental data it is necessary to use further MATLAB functions to locate the exact peak frequency from the graph. In order to do this the following script and actions were used.

```
I=find(spectra>14)

spectra(    :     )

spectra (    )

f (    )
```

Figure 14 *MATLAB script used to locate points at peaks*

As the frequency can be found at the peak, the only pertinent information is found above a certain range. The above operators in MATLAB are used to locate the sought after information.

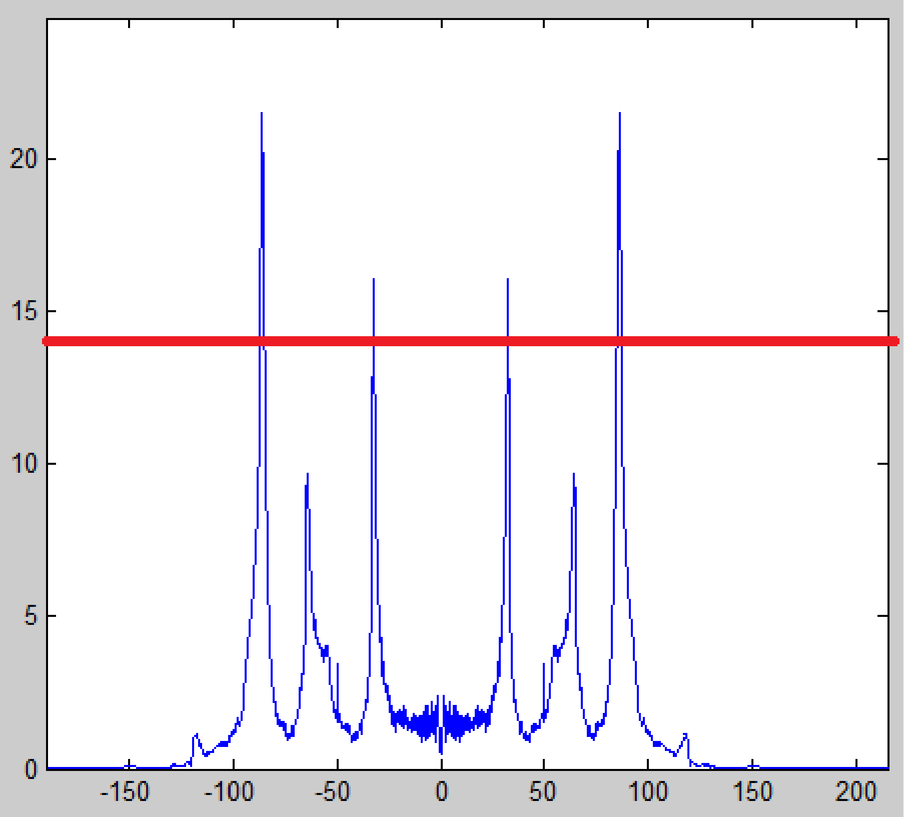
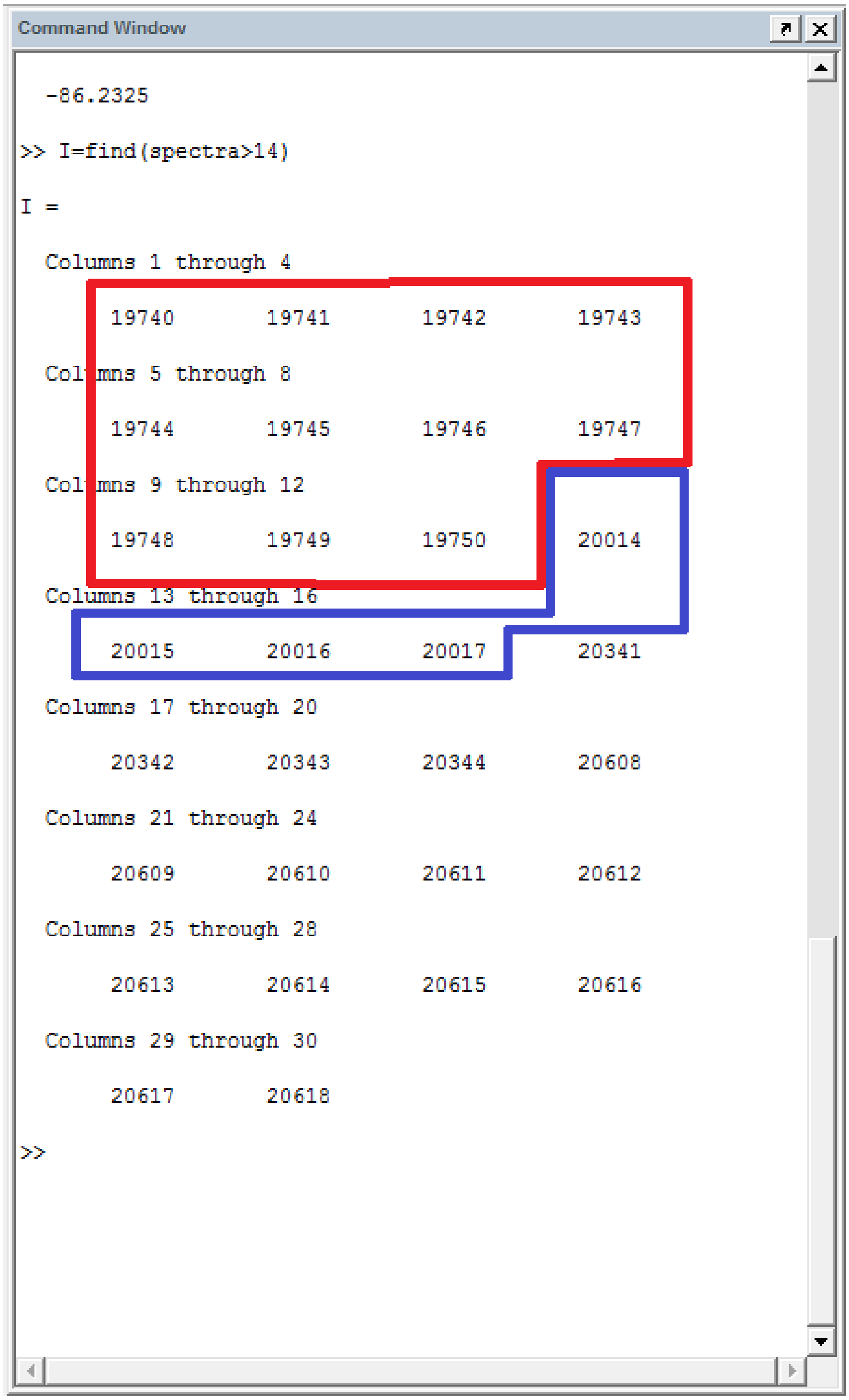


***Figure 15** Graphical representation of **I=find(spectra>14)** command*

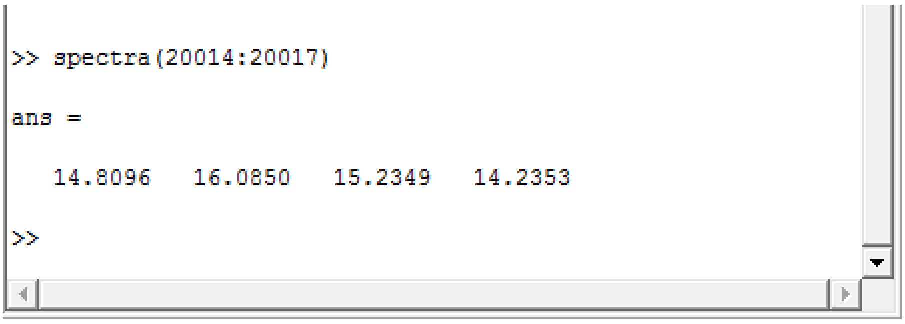
I=find(spectra>14); After applying this operator in MATLAB the following window is displayed. This window displays the plotted points above an intensity of 14. The red boxed series of numbers represents the large 80 Hz peak, and the blue boxed series represents the 32 Hz peak, which holds the desired information. We know this by using the **f(   );** function to determine the range of frequencies inside each series of values obtained through the **I=find** function.



```
  -86.2325

>> I=find(spectra>14)

I =

  Columns 1 through 4

       19740        19741        19742        19743

  Columns 5 through 8

       19744        19745        19746        19747

  Columns 9 through 12

       19748        19749        19750        20014

  Columns 13 through 16

       20015        20016        20017        20341

  Columns 17 through 20

       20342        20343        20344        20608

  Columns 21 through 24

       20609        20610        20611        20612

  Columns 25 through 28

       20613        20614        20615        20616

  Columns 29 through 30

       20617        20618

>>
```

Figure 16 *Window of MATLAB after using the* ***I=find(spectra>14)*** *command*

spectra(:); Using this command it is possible to find the intensity at specific points. This is used to locate the string of points that encompass the desired peak, which is in this case the peak around 32 Hz. Using this operator displays the following information.



```
>> spectra(20014:20017)

ans =

   14.8096   16.0850   15.2349   14.2353

>>
```

*Figure 17 Command window of MATLAB displaying the use of the **spectra (:   )** command*

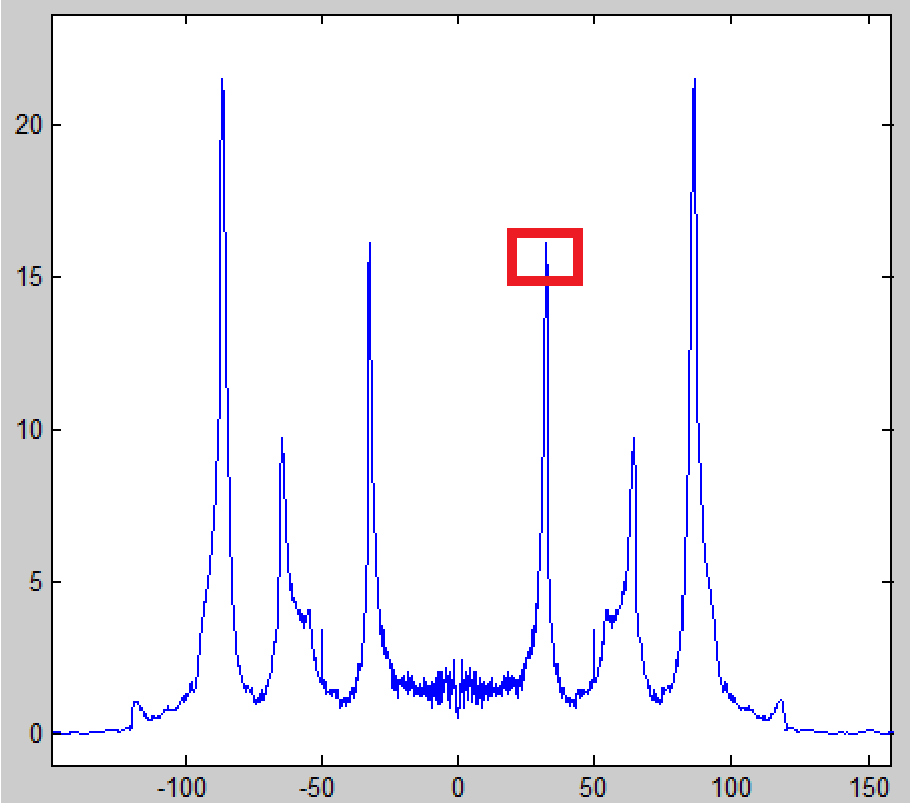
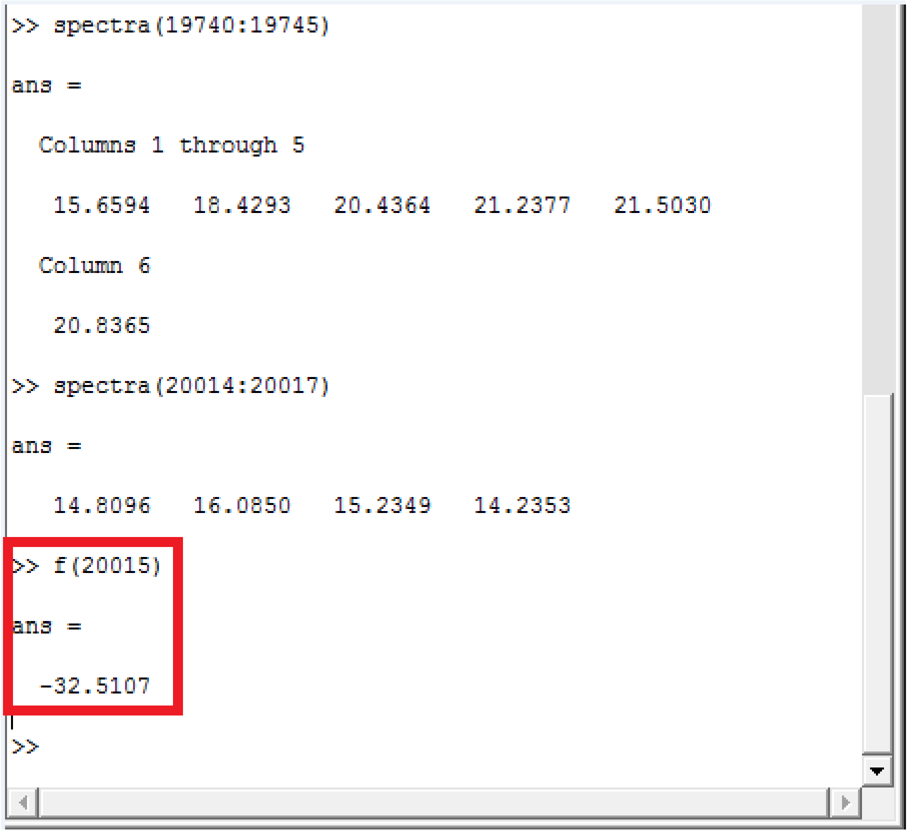


***Figure 18** Graphical representation of the **spectra (:   )** command*

As can be seen abvoe, these values represent the intensity of the signal at the peak around 32 Hz.

spectra (   ); Using this command it is then possible to pinpoint the desired point by determining the point that contains the highest intensity in each curve.

f (   ); After determining the correct point, this command is then used to determine the frequency at the corresponding point. As is shown below.

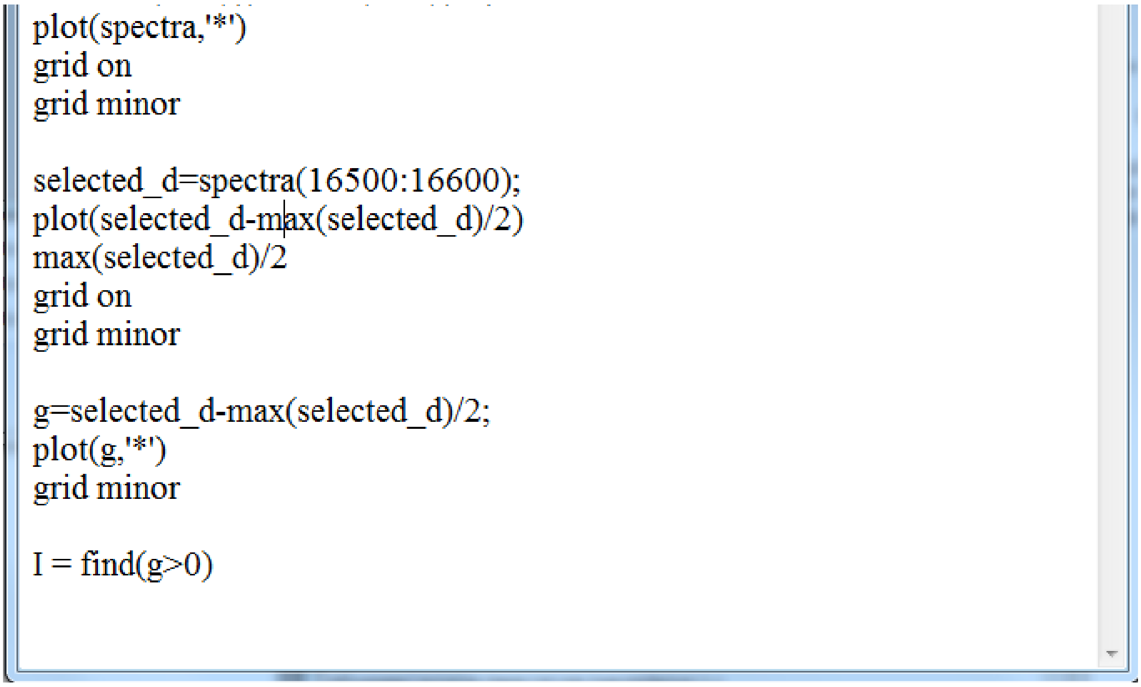


```
>> spectra(19740:19745)

ans =

  Columns 1 through 5

   15.6594   18.4293   20.4364   21.2377   21.5030

  Column 6

   20.8365

>> spectra(20014:20017)

ans =

   14.8096   16.0850   15.2349   14.2353

>> f(20015)

ans =

  -32.5107

>>
```

Figure 19 *Command window of MATLAB portraying the frequency at a specific data point.*

4.2.2 MATLAB Full Width at Half Maximum

As it is desired to find the damping coefficient from the FFT data set, after applying the FFT function to the data set, the following commands were used to obtain the pertinent information.



```
plot(spectra,'*')
grid on
grid minor

selected_d=spectra(16500:16600);
plot(selected_d-max(selected_d)/2)
max(selected_d)/2
grid on
grid minor

g=selected_d-max(selected_d)/2;
plot(g,'*')
grid minor

I = find(g>0)
```

Figure 20 *MATLAB script used to determine Full Width at Half Maximum (ω_o).*

Plot(spectra,’*’)

grid on

grid minor: These commands in conjuction with each other will plot the variable spectra, which was previously created from the data after having applied the FFT. It creates a graph with the x axis as indices rather than frequencies, which allows for the number of points to be determined within a field. The ‘*’ command, creates a graph with star shaped points instead of a continuous line, in order to more easily identify the amount of data points in the graph. The grid on and grid minor commands simply add a grid to the graph which helps with determining the correct search fields.

Selected_d=spectra(:); This command creates a data set determined from the “spectra” data set, in which one can focus on a smaller area of the “spectra” data set by creating a new data set, with the name selected_d which only shows a portion of “spectra” determined by the index points selected. This is done in order to remove the overtone frequency data from consideration when applying the later functions.

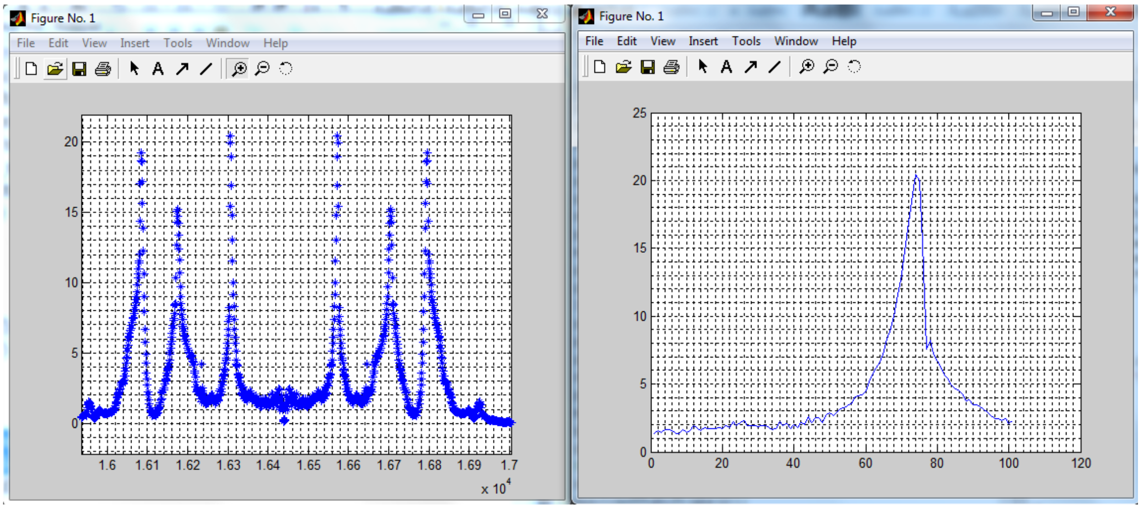


Figure 21 *Side by side graphs showing the result of of selected_d=(spectra(:   ) command.*

As can be seen above, the **Selected_d=spectra(:   );** command limits the data set, to the initial peak and its outlying data points.

Plot(selected_d-max(selected_d)/2); This command plots the previously determined data set, in a way so that the 0 line, occurs at half of the peaks height, this is done so as to make it easier to determine the band width at half height of the peak.

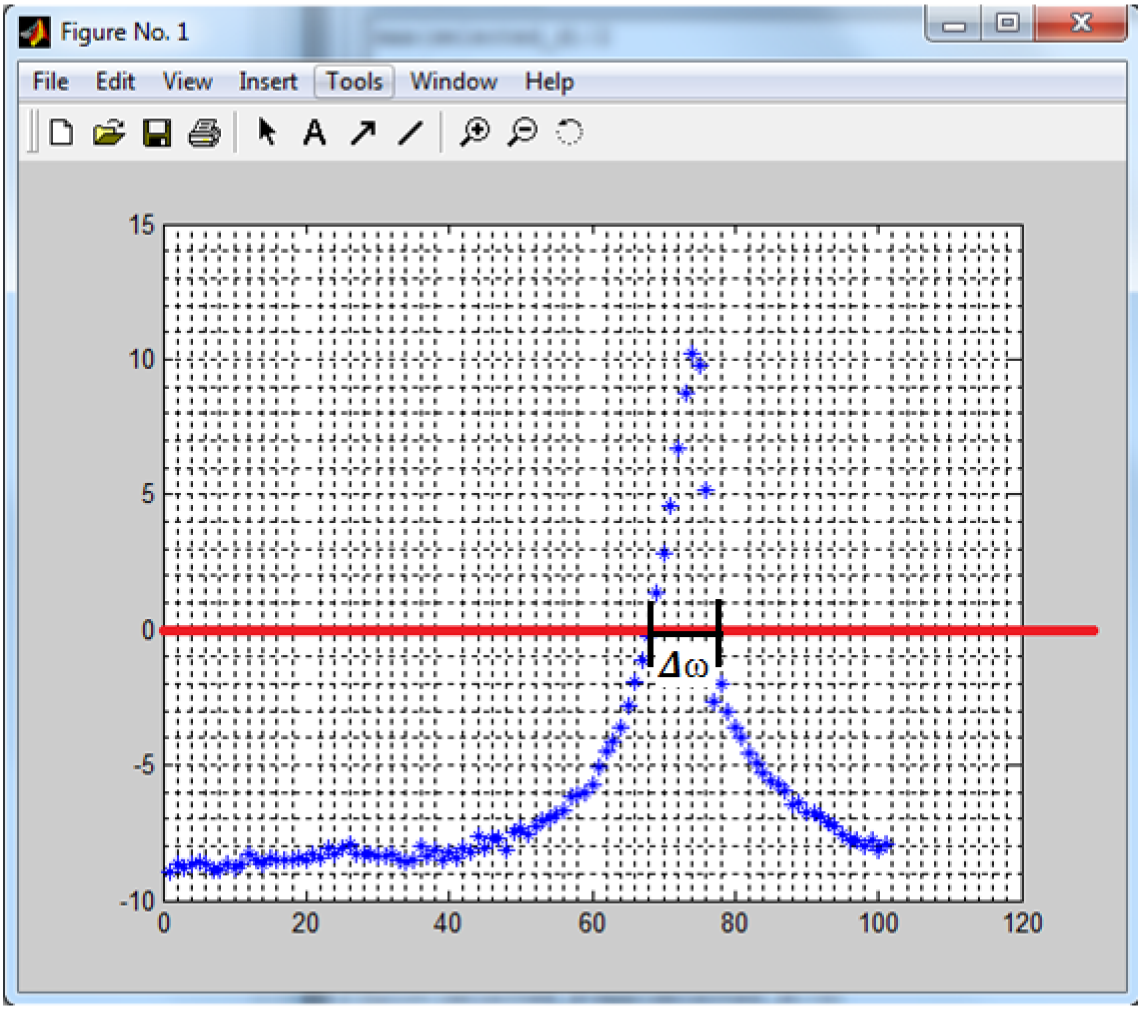


Figure 22 Graph displaying the function of ***Plot(selected_d-max(selected_d)/2);.***

g=selected_d-max(selected_d)/2; This command creates another variable data set to be considered, in this case it takes the previously determined data from "selected_d" and assigns it a name, that being "g".

I = find(g>0); The I=find(g>0) command, operates on the data set "g" to find all points of the "g" data set that are greater than 0, and shows which data points those are.

After implementing the **I = find(g>0);** MATLAB displays the number of points above the zero line, using this number along with the step length (df), which was discovered earlier during the FFT application, we are able to determine the bandwidth at half maximum of the FFT function.

This is done by multiplying **df** by the number of points found above the half height.

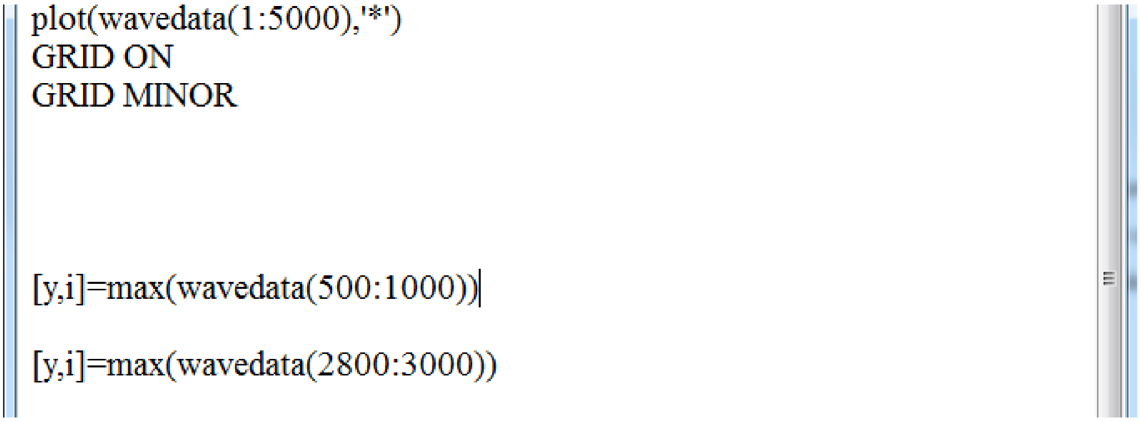
$$Bandwidth = df * (n > (\frac{peakmax}{2})) \qquad \text{[Eq.11]}$$

This value is found with a degree of error of +- 2 points, due to the possibility of a lack of sufficient points.

4.3 MATLAB Time Data Analysis

As is stated previously, the objective of this thesis is to use both time and frequency data to obtain resonance frequency and damping coefficient, and to compare the accuracy and precision of these two methods. The following is the method used to obtain the desired information while using the time data.

4.3.1 Obtaining resonance frequency using time data



```
plot(wavedata(1:5000),'*')
GRID ON
GRID MINOR




[y,i]=max(wavedata(500:1000))

[y,i]=max(wavedata(2800:3000))
```

Figure 23 *MATLAB script used to obtain resonance frequency from time data.*

Plot(wavedata(1:5000),'*'): This command plots the data set in its orginal form, imported from the Audacity program. The number series 1:5000 determines that it will plot the 1st through the 5000th point.

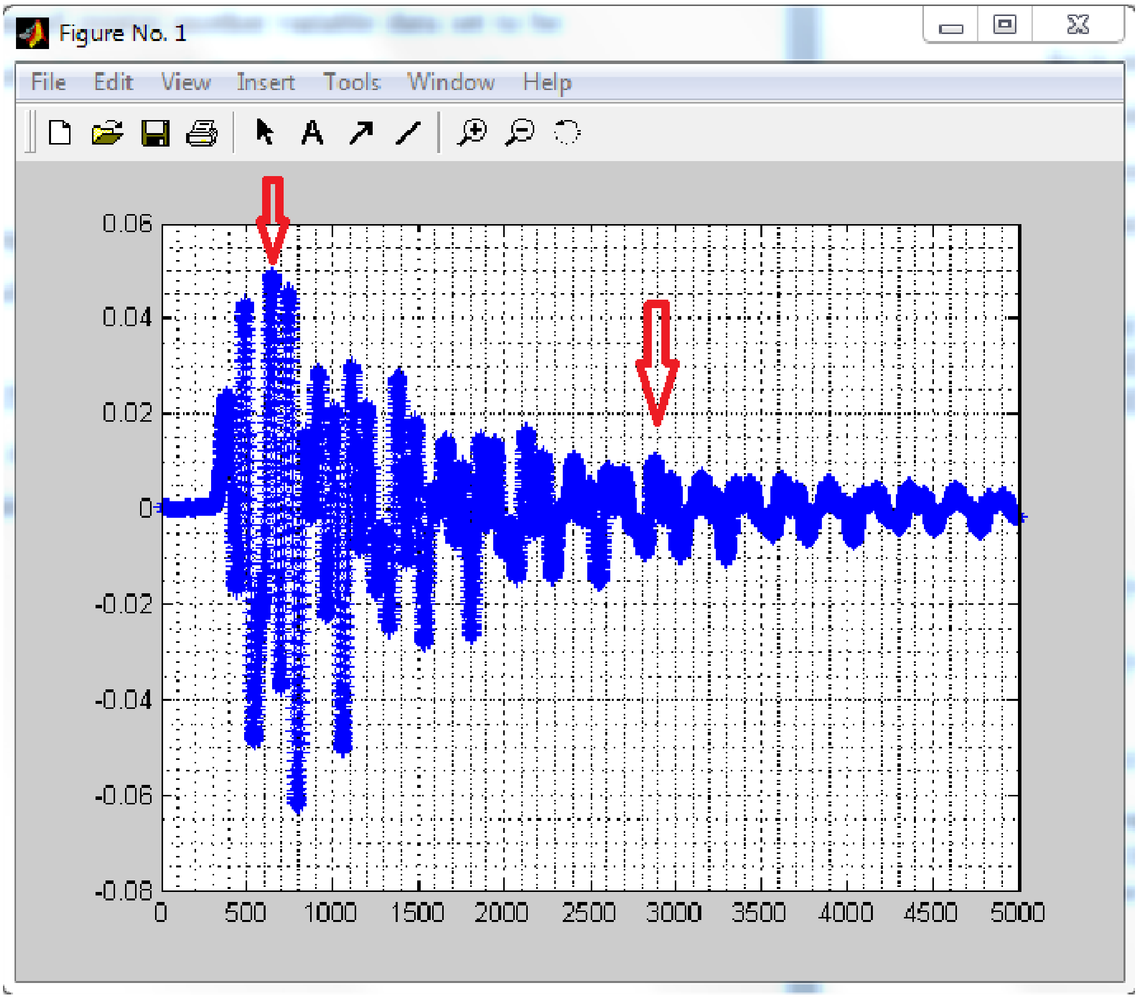


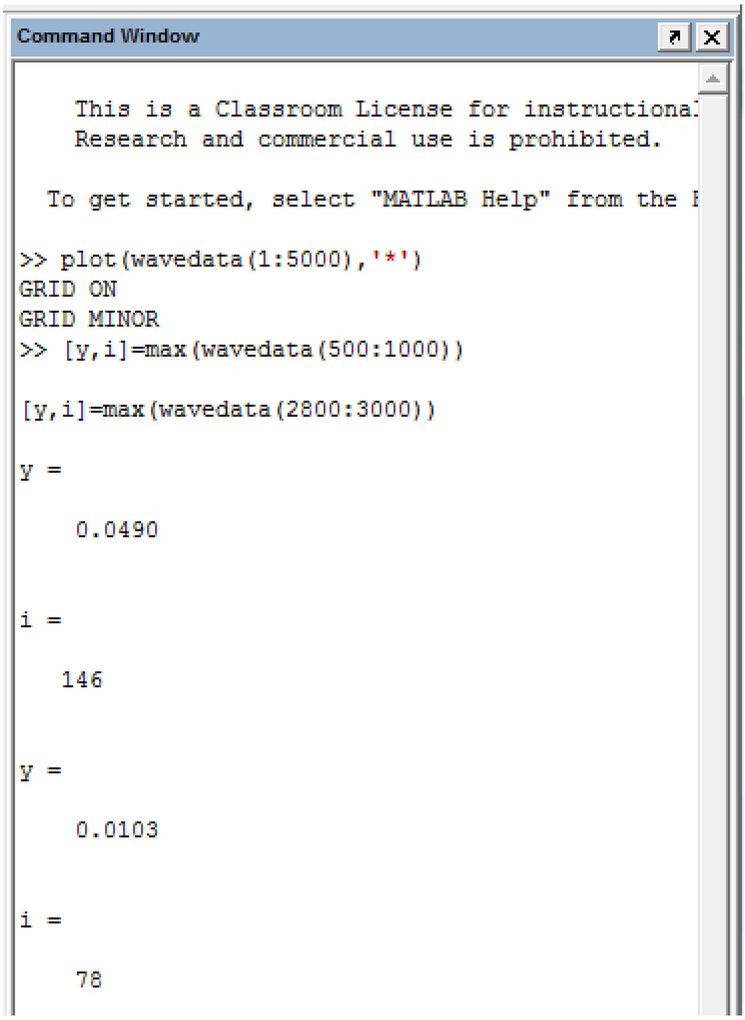
Figure 24 *Graphical display of peak finding in time domain, displayed as amplitude vs. time.*

With the above graph it is possible to determine by sight the 1st and 10th peak from each graph and their index ranges. For example in the above graph, the first peak takes place between the 500th and 1000th indices, while the 10th peak occurs between the 2800th and 3000th point.

By measuring the number of indeces between two peaks we can determine the period *T*. In order to reduce the error when finding *T* we measure the points between 10 peaks, so we can find the average of 10 periods.

[y,i]=max(wavedata(500:1000))

[y,i]=max(wavedata(2800:3000)): Using the **[y,i]=max(:)** function in matlab, it is possible to determine the max point within a range of numbers. We use this to find the peak points for the 1st and 10th peak in the time data.



```
    This is a Classroom License for instructional
    Research and commercial use is prohibited.

  To get started, select "MATLAB Help" from the H

>> plot(wavedata(1:5000),'*')
GRID ON
GRID MINOR
>> [y,i]=max(wavedata(500:1000))

[y,i]=max(wavedata(2800:3000))

y =

    0.0490


i =

   146


y =

    0.0103


i =

    78
```

***Figure 25** MATLAB command window displaying results of [y,i] function.*

Above is an image showing the format which MATLAB presents the max points and their locations. For the first peak, we can see that the **y** value is the height of the peak, where as the **I** value is that point's position in relation to the search fields beginning point. In the above case, the first peak's search field is from 500 to 1000, so the **I** value being 146, represents the first peaks position as 146 points from 500, resulting in **646**. The same search is made for the 10th peak's position, which in the above case shows it being 78 points from the 2800th point, resulting in its position being at point 2878.

With the knowledge of the fact that

$$f = \frac{1}{T} \qquad \text{[Eq. 12],}$$

$$T = \frac{\Delta t}{samplerate\ (8000)} \qquad \text{[Eq. 13] ,}$$

and that $\frac{T}{N(10)} = \frac{1}{f}$ [Eq. 14].

We can determine from the experimental data that the frequency

$$f = \frac{(N)(8000)}{\Delta t} \qquad \text{[Eq. 15]}$$

In the case of the first experimenta data, $f = \frac{(10)(8000)}{2506}$ = 31.9234 Hz

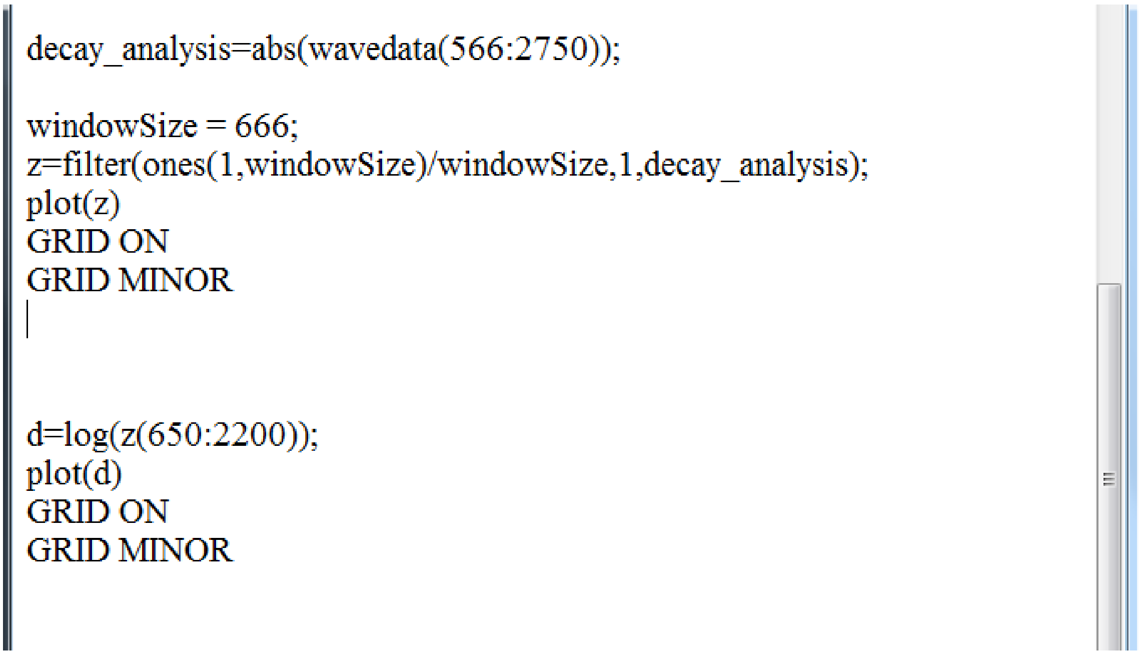
4.3.2 Obtaining damping coefficient from time data

When considering time data analysis it is necessary to use the second order differential equation:

$$y(t) = y_0 e^{\zeta t}(\cos(\omega t) + \sin(\omega t))$$

It is then possible to remove the second term "$(\cos(\omega t) + \sin(\omega t))$" which is high frequency noise by applying absolute value, and a low pass filter. Which will leave the first term, $y(t) = y_0 e^{\zeta t}$ ➔ $\frac{y(t)}{y_0} = e^{\zeta t}$.

We then logarithmize $\frac{y(t)}{y_0} = e^{\zeta t}$, which will then provide us with ln $\left(\frac{y(t)}{y_0}\right) = -\zeta t$. Which provides us with an equation in slope intercept form y= ax + b, where x is the damping coefficient ζ. We then fit a linear regression to find the slope ζ.



```
decay_analysis=abs(wavedata(566:2750));

windowSize = 666;
z=filter(ones(1,windowSize)/windowSize,1,decay_analysis);
plot(z)
GRID ON
GRID MINOR



d=log(z(650:2200));
plot(d)
GRID ON
GRID MINOR
```

Figure 26 *MATLAB script used to determine damping coefficient from time data.*

The above command list is used to obtain the damping coefficient from the time domain data set. After using the former commands it is possible to use the **i** indices obtained from the **[y,i]=max(wavedata(500:1000))** command to determine the relevant field of data points to be studied, those being the points between the initial and 10th peak.

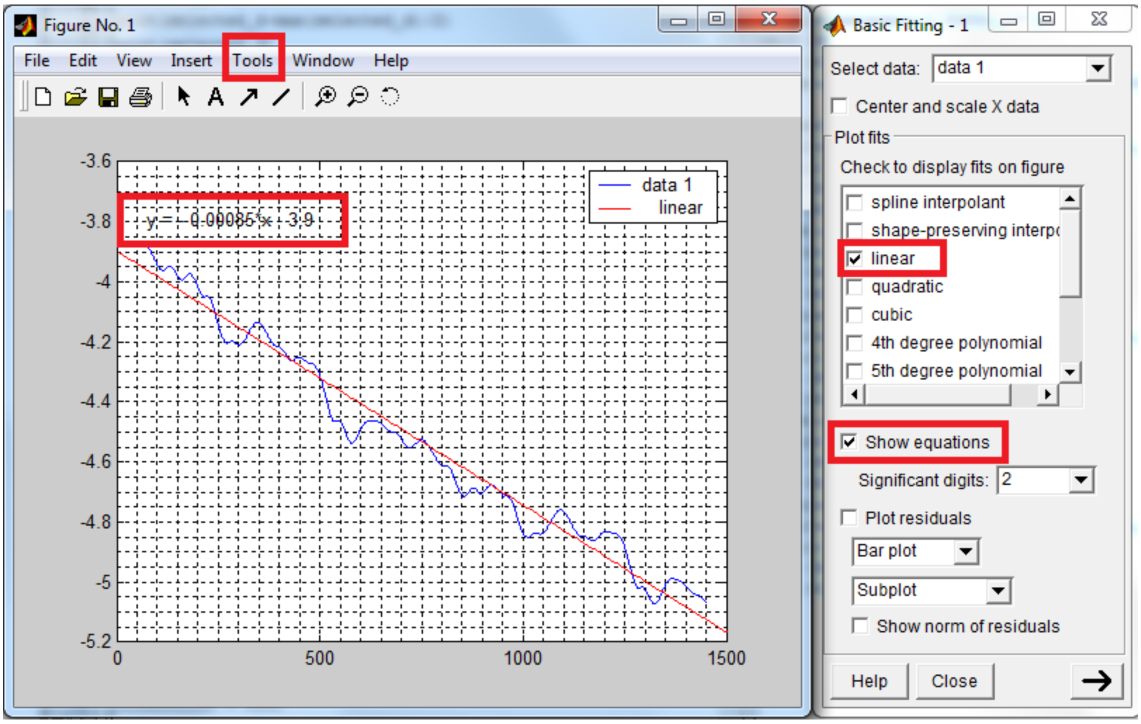
decay_analysis=abs(wavedata(566:2750)); This command is used to create a new data set known as “decay_analysis,” which consists of the absolute value of the data points 566 through 2750 from the “wavedata” data set.

windowSize = 666;
z=filter(ones(1,windowSize)/windowSize,1,decay_analysis);
plot(z): This series of commands acts as a digital filter, to further remove unnecessary data from the time data set. It also creates a new data set “z,” which is a data set consisting of the filtered data from “decay_analysis.”

d=log(z(650:2200));

plot(d): These commands apply the natural logarithm to the ”z” data set in order to create the new data set ”d.” We can then apply a linear regression to the “d” data set in order to determine the damping coefficient of the composite laminate.

The regression is plottable through the matlab toolkit by selecting in the “MATLAB Figure window” **Tools-→Basic Fitting→linear and checking show equations.** The slope is shown in the red box over the graph, in this case it is shown as **y = .00086x * 3.9.** As seen .00086 is the resulting slope, and represents the damping coefficient as determined by the time data analysis.



y = - 0.00085*x - 3.9

Select data: data 1

□ Center and scale X data

☑ linear

☑ Show equations

Significant digits: 2

□ Plot residuals

Bar plot

Subplot

□ Show norm of residuals

Figure 27 Method used of finding damping coefficient from time data in MATLAB.

5 RESULTS

After collecting 50 samples and applying the above MATLAB operations, the average, standard deviation, and relative error was calculated for each of the values obtained. The results are as follows.

Table 2 *Results from data analysis of 50 signals*

	Average	**Standard Deviation**	**Relative Error (StDev/Average)**
Time Data			
Resonance frequency F(0) (Hz)	35,72163061	0,889721929	0,024907092
Damping Coefficient ((N*s)/m)	0,0007354	2,94999E-05	0,040114106
FFT Data			
Resonance frequency F(0) (Hz)	32,53709	0,09423	0,002896072
Band width at half height ($\Delta\omega$)	2,105106	0,143154	0,034001662

As is shown above, the relative error and standard deviation for both the frequency, and the damping data is lower when taken from the FFT data. From referring to the last column in the above table and compare the results it can be seen that f(0) is found with about 9 times less variance from the FFT method. Also the damping related data can be found with 1.3 times less variance.

6 DISCUSSION AND CONCLUSION

After reviewing the results from the tests completed, it is made obvious that the more stable method when trying to obtain a material's resonance frequency and damping coefficient, is that of the FFT analysis. As can be seen in the results, the FFT analysis provides a resonance frequency with a standard deviation and relative error which is 10 times smaller than that of the time domain analysis, with values of 0.00289 and 0.0249 respectively.

In the determination of the damping coefficient and its relative data, the FFT method proved again to be the more stable analysis method, as it provided a value with 13 percent less relative error than that of the time data analysis, with values of 0.034 and 0.04 respectively.

Based on the results provided, it can be determined that in the case of a lightweight and stiff material, the best non destructive analysis method involves the use of Fast Fourier Transform to provide resonance frequency and damping coefficient.

While this conclusion is found, it is important to note that the same results may not be shown for materials with different stiffness to mass ratios.

7 SUGGESTION FOR FURTHER WORK

If it is desired to take this work further, one could pursue the following studies.

- To study FFT data at overtone peaks
- To collect data from a battery driven system, to avoid switched mode power noise affecting the data collected
- To study a double clamped system, rather than a cantilever beam system
- To study poisson ratio and shear properties by torsional vibrating in a double clamped system.
- To study material ageing with the current method

REFERENCES

Knight, Randall. 2013, Physics for Scientists and Engineers, Third Edition, Pearson Education, Inc.: James Smith, 1277 pages.

Ross C.T.F. 1996, Mechanics of Solids, Prentice Hall Europe: Prentice Hall Europe, 456 Pages.

*Audacity*2014, 16.04.2014-last update. Available: http://audacity.sourceforge.net/ [2014, 15.02.2014].

*MathWorks*2014, 16.04.2014-last update. Available: http://www.mathworks.se/products/matlab/ [2014, 15.02.2014].

*Piezoelectric Transducers*2007, 20.12.2007-last update [Homepage of NDT Resource Center], [Online]. Available: http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/EquipmentTrans/piezotransducers.htm [2014, 15.02.2014].

8 APPENDICES

Fast Fourier Transform Analysis Data From MATLAB

Sample #	f(o)	df	VALUE	Bandwidth		
1	32,4827	0,2433	8	1,9464		
2	32,5443	0,2267	10	2,267	**Average f(0)**	**STDEV**
3	32,4287	0,23	9	2,07	32,53709	0,09423
4	32,6105	0,2029	11	2,2319		
5	32,355	0,1549	15	2,3235	**Rel ERR**	
6	32,433	0,1523	15	2,2845	0,002896072	
7	32,3835	0,1459	14	2,0426		
8	32,5501	0,1674	13	2,1762		
9	32,6015	0,161	12	1,932		
10	32,4889	0,1679	12	2,0148	**Average Bandwidth**	**STDEV**
11	32,5137	0,163	12	1,956	2,105106	0,143154
12	32,6298	0,1704	12	2,0448		
13	32,3975	0,1558	13	2,0254		
14	32,4387	0,1287	19	2,4453	**Rel ERR**	
15	32,552	0,1497	14	2,0958	0,034001662	
16	32,4505	0,1506	13	1,9578		
17	32,4358	0,1552	13	2,0176		
18	32,496	0,1494	13	1,9422		
19	32,5519	0,144	13	1,872		
20	32,7079	0,1518	13	1,9734		
21	32,518	0,1455	17	2,4735		
22	32,5947	0,143	15	2,145		
23	32,6144	0,1362	17	2,3154		
24	32,6067	0,1328	15	1,992		
25	32,3856	0,152	15	2,28		
26	32,3741	0,1998	11	2,1978		
27	32,4639	0,1877	12	2,2524		
28	32,5737	0,1825	11	2,0075		
29	32,6407	0,1949	10	1,949		
30	32,5413	0,1838	12	2,2056		
31	32,6838	0,1862	11	2,0482		
32	32,539	0,1914	11	2,1054		
33	32,6407	0,1949	11	2,1439		
34	32,5281	0,1925	12	2,31		
35	32,6996	0,1885	12	2,262		
36	32,3832	0,1916	11	2,1076		
37	32,5551	0,186	10	1,86		
38	32,5617	0,1819	12	2,1828		
39	32,4963	0,1846	11	2,0306		
40	32,6187	0,1891	11	2,0801		
41	32,5549	0,1774	11	1,9514		
42	32,6097	0,1772	12	2,1264		
43	32,6863	0,1716	13	2,2308		
44	32,4989	0,1879	11	2,0669		
45	32,6405	0,176	12	2,112		
46	32,3994	0,177	12	2,124		
47	32,5746	0,1756	12	2,1072		
48	32,605	0,1681	12	2,0172		
49	32,5548	0,1696	12	2,0352		
50	32,6591	0,1742	11	1,9162		

Time domain Analysis Data From MATLAB

Sample #	**wo**	**Damping coefficient**				
1	31,9234	0,00077		**Average w(0)**		**STDEV**
2	35,8262	0,0008		35,72163061		0,889721929
3	37,5059	0,00077				
4	36,3141	0,0008		**REL ERROR**		
5	35,8262	0,00078		0,024907092		
6	35,8102	0,00077				
7	35,8102	0,00077				
8	35,8102	0,00074				
9	35,8102	0,00073		**Average D.C.**		**STDEV**
10	37,4883	0,00072		0,0007354		2,94999E-05
11	35,8102	0,00074				
12	35,8102	0,00074		**REL ERROR**		
13	35,8102	0,00074		0,040114106		
14	35,8102	0,00076				
15	35,8102	0,00075				
16	35,7942	0,00077				
17	35,8102	0,00074				
18	35,8102	0,00075				
19	35,8102	0,00073				
20	35,8102	0,00073				
21	35,7942	0,00074				
22	35,7942	0,00074				
23	35,7942	0,00073				
24	35,9228	0,00077				
25	35,8102	0,00078				
26	35,8102	0,00076				
27	35,8102	0,00074				
28	35,8102	0,00076				
29	35,7942	0,00072				
30	35,8102	0,00072				
31	35,7942	0,00069				
32	35,8102	0,0007				
33	35,8102	0,00071				
34	35,8102	0,00069				
35	35,7942	0,0007				
36	31,5582	0,00071				
37	35,7782	0,0007				
38	35,7942	0,0007				
39	35,7942	0,0007				
40	35,7942	0,00071				
41	35,7782	0,0007				
42	35,7782	0,0007				
43	35,7782	0,00069				
44	35,8102	0,00069				
45	35,7942	0,00074				
46	35,7942	0,00072				
47	35,7942	0,00073				
48	35,8102	0,00073				
49	35,9228	0,00076				
50	35,8102	0,00074				