Title Page

UNIVERSITY OF SALFORD

School of Computing Science and Engineering

Vibration of a Cantilever plate

Lab Report

Name: Joshua Jones

Course Title: MEng Mechanical Engineering with Industrial Experience

Course Code: E/ME/S Mechanical Engineering E3

Module: Control and Dynamics Engineering

Supervisor: Prof D. Howard, Dr James Gardiner

Table of Contents

Introduction

This lab is designed to explore multi degree of freedom systems using an electrodynamic shaker to excite a cantilever plate to vibrate over a range of frequencies. By applying the assumption of lumped masses making up the system the system complexity is reduced to 6 degrees of freedom corresponding to the bolts locate in the plate, as concentrated sources of mass. A matlab script is used to predict the natural frequencies of this reduced 6-degree-of-freedom system and the shaker excites the plate at these frequencies to verify whether the real system behaved in the same fashion at the natural frequencies predicted. This matlab script has been ported to the Python programming language; this can be found in appendix B.

There were some differences noted in the results, which arise due to the assumption of reduced degrees of freedom when in reality there are infinite.

The effects of the vibrations on the plate were illustrated by loading the top face of the plate with sand, which would in turn vibrate and converge on the nodes of the system and come to rest where no vibrations occur.

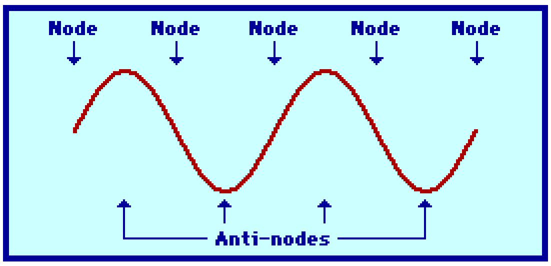


Figure 2 illustration of vibration with nodes and antinodes

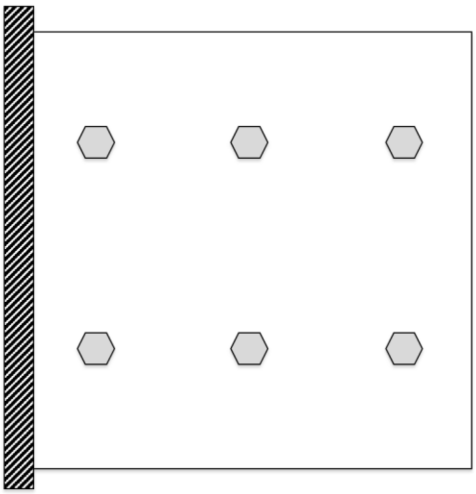


Figure 1 diagram of the experimental system

Aims

The aims of this lab report are to capture and report the findings and results of the experiment conducted, titled "Vibration of a cantilever plate lab". The experiment was designed to investigate the natural frequencies and vibration modes of an m-degree of freedom system. Specifically referring to figure 1 the cantilever plate with six bolts, these bolt positions were used to reduce the problem from an infinite degree of freedom system to an analytically approachable 6 degree of freedom problem.

Objectives

The Objectives of the laboratory are to by way of shaker apparatus to excite the system across a range of frequencies, determine and observe the natural frequencies, corresponding modes and compare the visually determined values with a theoretically predicted set of values.

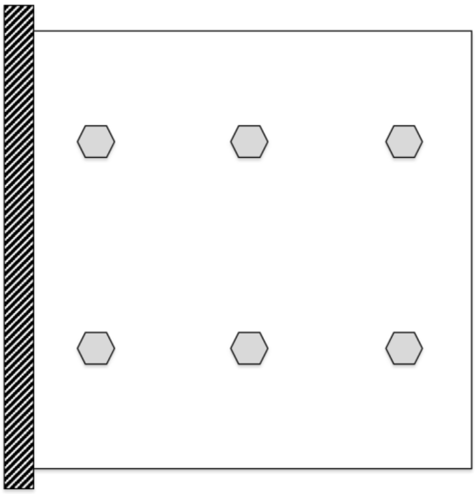
Once these frequencies are determined, the aim is to explore the systems frequency response and the mode shapes produced.

Secondary objectives are to develop a theoretical model and confirm the provided equations, along with developing predicted results with which to validate the experimental results, and to explore possible sources of error introduction within the test apparatus, which may lead to the measured results differing from the predicted results.

The python programming language has been implemented to produce a similar script that predicts the natural frequencies of the same 6-dof system and yields the same natural frequencies, time has not been spend implementing the mode shape plots, these are visualised using Solidworks simulation for the 6 expected modes and 3 additional modes.

Engineering Science of mechanical vibration of a cantilever plate

Problem Description



Replicated from figure 1 diagram of the experimental system

The motion of a system (ignoring damping) can be described by the second order differential equation: $[M]\underline{\ddot{x}} + [K]\underline{x} \ = \underline{0}$

Where [M] is the $n \times n$ mass matrix of the system [K] is the $n \times n$ stiffness matrix and $\underline{x}$ is the $n \times 1$ column vector of the displacements.

While the real system has a practically infinite number of degrees of freedom the problem can be reduced by assuming the mass [M] in the system is acting in a lumped parameter concentrated and acting only at the position of the six bolts shown in figure 1.

This reduces the mass matrix [M] and the stiffness matrix [K] to 6×6 matrices, which can be analysed using simple linear algebra. (For larger matrices, methods that are more exotic may have to be employed to reduce computational run time.)

For a repeated action on the system, the displacement matrix can be described by assuming sinusoidal behaviour and written as: $\underline{x} = \underline{X}Asin(\omega t + \varphi)$

Differentiated twice the equation becomes $\underline{\ddot{x}} = -\omega^2\underline{X}Asin(\omega t + \phi)$ or $\underline{\ddot{x}} = -\omega^2\underline{x}$

Replacing $\underline{\ddot{x}}$ with this result in $[M]\underline{\ddot{x}} + [K]\underline{x} \ = \underline{0}$ to yield $-[M]\omega^2\underline{x} + [K]\underline{x} \ = \underline{0}$

This substitution reduces the differential equation to a homogenous linear equation, which can then be solved using a linear algebra approach rather than a differential equation approach, the choice for doing so is that linear algebra methods scale well and produce computationally implementable functions. Differential equations can be solved at scale but must be approached with numerical methods and approximations in many cases.

Rearranging the result to replace the displacement vector with the relative modal vector yields the following:

$-[M]\omega^2\underline{x} + [K]\underline{x} = \underline{0} \rightarrow \quad ([K] - [M]\omega^2)\underline{x} = \underline{0}$

Given that $\lambda = \omega^2$ *and* $\frac{\underline{x}}{Asin(\omega t+\phi)} = \underline{X}$ the above becomes:

$([K] - [M]\lambda)\underline{X} = \underline{0}$

To produce the characteristic equation in eigenvalue form multiplying the equation by $[M]^{-1}$ produces the identity matrix [I] (because$[M] \times [M]^{-1} = [I]$)

The equation becomes $[A - \lambda I]\underline{X} = \underline{0}$ (where $A = [M]^{-1}[K]$)

Using a mathematical software package the eigenvalue and Eigen vectors have been determined.

Eigenvalues are a special set of scalars associated with a linear system of equations (i.e., a matrix equation) that are sometimes also known as characteristic roots, characteristic values (Hoffman and Kunze 1971), proper values, or latent roots (Marcus and Minc 1988, p. 144). (Weisstein, 2016)

The determination of the eigenvalues and eigenvectors of a system is extremely important in physics and engineering, where it is equivalent to matrix diagonalization and arises in such common applications as stability analysis, the physics of rotating bodies, and small oscillations of vibrating systems, to name only a few. Each eigenvalue is paired with a corresponding so-called eigenvector. Eigenvectors are a special set of vectors associated with a linear system of equations (i.e., a matrix equation) that are sometimes also known as characteristic vectors, proper vectors, or latent vectors (Marcus and Minc 1988, p. 144). (Weisstein, 2016)

Assumptions

Assumption 1: While the real plate exhibits infinite degrees of freedom, the model is a lumped mass model concentrated around six bolts in the plate.

Assumption 2: the addition of the bolt mass should concentrate the mass and allow the real model to relate to the theoretical lumped mass approach better.

Assumption 3: the addition of sand for visualisation purposes will not affect the system and the sand mass is assumed negligible.

Analysis Approach

The theoretical approach to the analysis is to make the above assumptions to develop a closed form reduced problem and with implementation of a mathematical package, solve the system using matrix methods to determine the natural frequencies and corresponding mode shapes.

Matlab and python have been used to develop a 6-dof undamped free vibration representative model of the plate system.

This section describes the working of the code used referring to the file Six_DF_Plate.m

The matlab and python code is split into distinct sections; initially system properties are calculated based on the plate geometry, the mass is divided by 6 then multiplied by the identity matrix to produce a matrix form of the mass distribution across the system (at each bolt).

The code is set up to either calculate the compliance matrix from the load and displacement matrices by $\frac{[displacement]}{[load]} = [h]$ or using some predefined compliance matrix values, the user can calculate the specific compliance values suggested (as seen in figure 4 p13.) taking an average and inputting into the existing compliance matrix [h].

Because the goal is to develop the equation $[A - \lambda I]\underline{X} = \underline{0}$ (where $A = [M]^{-1}[K]$) for an eigenvalue solution, the mass matrix [M] and the compliance matrix [H] are inverted and multiplied to give A, which then feeds into the .eig() function in the standard matlab library or the numpy numerical python library.

.eig() must be assigned to two variables because the function returns two pieces of information, namely the eigenvalues and Eigen vectors, both are produced in matrix form. (the data type for these variables is similar to the array data structure, which is convenient for matrix operations.)

Initially eig(A) returns a diagonal matrix for the eigenvalues λ

```
  1.0e+06 *

    0.0082         0         0         0         0         0
         0    0.0680         0         0         0         0
         0         0    0.1946         0         0         0
         0         0         0    0.6971         0         0
         0         0         0         0    1.6523         0
         0         0         0         0         0    3.2915
```

Figure 3 Raw eigenvalue results

This matrix is flattened using the ones (1, 6) command into a 1x6 vector for ease of reading on output, and for unit conversion. `Hz= sqrt(ones(1,6)*d)/(2*pi)` Following this, the remaining code is for plotting the 3D plot of the eigenvectors.

Apparatus and Test Procedure

The apparatus used in this lab is a cantilever plate with six boltholes, six nut and bolt pairs, a tray of sand, and an electrodynamic shaker with frequency control.

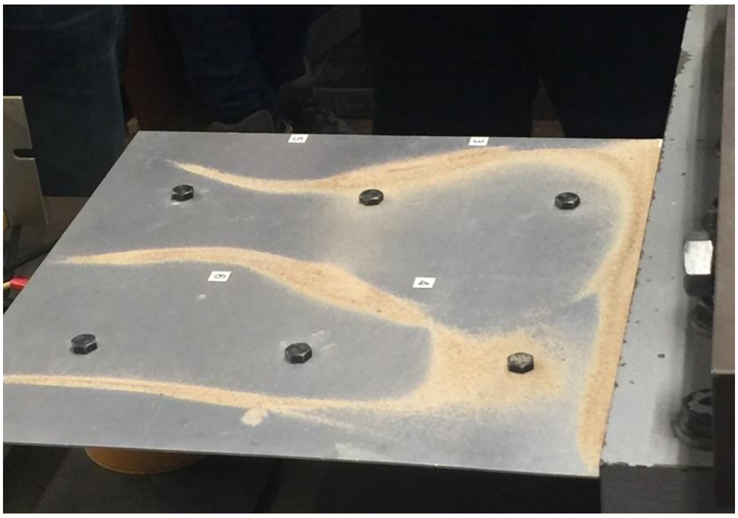


Figure 4 Test set up showing one of the modes illustrated by the application of sand

Using a ruler to measure the plate length and width and a micrometre to measure the plate thickness the system properties were determined.

A simple digital scale was used to determine the bolt mass.

plate parameters	
hole diameter	6mm
bolt mass	93g
plate width	306mm
plate length	306mm
material density	2770kg/m^3

The geometry relations are:

$$boltHoleMass = \pi * (holeRadius^2) * plateThickness * densAL * 1 \times 10^{-9}$$

$$plateMass \ = \ densAL * plateWidth^2 * plateThickness * 1 \times 10^{-9}$$

$$lumpedMass \ = \frac{plateMass}{6} + \ boltMass - boltHoleMass$$

The python (ported from matlab) code provides the result of these calculations as seen in figure 6, replicated below:

Table 1 calculated mass parameters

The mass of the bolt is 0.0093 kg
The density of aluminum is 2770 kg/m^3
The mass of the bolt hole (material removed) is: 0.0001 kg
The mass of plate is: 0.415 kg
The lumped system mass at each of the six points is: 0.0783 kg

The mass concentration leads to the mass matrix, which is a 6x6 matrix with zeroes everywhere except the diagonal, which is populated along the diagonal with the lumped system mass at each point (0.0783kg)

Test Procedure and experimental results

To determine the initial compliance matrix of the system the known standard mass was determined using a digital scale, this mass was then secured by thread to the underside of one of the 6 mass points, the deflection at another mass point of the system was recorded with a depth gauge. The nomenclature in table 2 indicates the measure point and load point.

Table 2 measured deflections

load	0.501	kg

measure/load	displacement mm	displacement m
1,6	0.20	0.00020
2,5	0.19	0.00019
3,3	0.93	0.00093
3,4	0.63	0.00063
4,3	0.63	0.00063
4,4	-0.94	-0.00094
5,2	-0.22	-0.00022
6,1	-0.13	-0.00013

The results in table 1 allow the average values of theoretically identical measurements to be used in the compliance matrix, compliance is calculated as the $\frac{displacement}{load}$

Table 3 Calculated values of compliance

measure/load	absolute compliance values m/N x10^-6
1,6	40.69333301
2,5	38.65866636
3,3	189.2239985
3,4	128.183999
4,3	128.183999
4,4	191.2586651
5,2	44.76266631
6,1	26.45066646
(**a**) avg 16,61,25,52	37.6413
(**b**) avg 34,43	128.184
(**c**) avg 33,44	190.241

$$\begin{bmatrix} & & & & & a \\ & & & & a & \\ & & c & b & & \\ & & b & c & & \\ & a & & & & \\ a & & & & & \end{bmatrix}$$

Figure 5 illustration of the user added values in the compliance matrix

The compliance matrix has been augmented with the user calculated values and placed as illustrated and above in bold, where:

- a is the average of matrix items 16, 61, 52, and 25.
- b is the average of matrix items 43,34
- c is the average of matrix items 44,33

The remaining matrix entries were calculated previously and are not part of the lab set up, the matrix was only augmented as illustrated above, the other matrix values are assumed correct.

The electrodynamic shaker was used to excite the system, and sand was used to illustrate the mode shapes. The frequency was increased across the range to capture the first six modes, and the resonant (natural) frequencies were recorded, including 3 additional mode found (mode modes are expected because the real system has infinite degrees of freedom.

Theoretical Calculations and Results

From the mass matrix and the compliance matrix, the following matrix can be computed by taking the inverse of both matrices and multiplying the results as seen in figure 4.

The matrix A can then be operated on to find the eigenvalue and eigenvectors of the system, the eigenvalues calculated are the square of the natural frequencies.

$$\begin{bmatrix} 2106733.42285937 & -476632.98154716 & -626937.43861343 & 447497.22250821 & 205802.99807407 & -234038.50894849 \\ -476632.98154716 & 2106733.42285937 & 447497.22250821 & -626937.43861343 & -234038.50894849 & 205802.99807407 \\ -626937.43861343 & 447497.22250821 & 644355.93986929 & -460395.58803087 & -307611.61959202 & 239762.66916581 \\ 447497.22250821 & -626937.43861343 & -460395.58803087 & 644355.93986929 & 239762.66916581 & -307611.61959202 \\ 205802.99807407 & -234038.50894849 & -307611.61959202 & 239762.66916581 & 204829.49247582 & -163775.71901257 \\ -234038.50894849 & 205802.99807407 & 239762.66916581 & -307611.61959202 & -163775.71901257 & 204829.49247582 \end{bmatrix}$$

Figure 6 Matrix A = [M]^-1[H]^-1 for predicted 6-dof behavior

Using the matlab standard library or the numpy numerical python library implementations of the .eig() function the Eigen values can be calculated.

Each Eigenvalue is equal to the square of a particular natural frequency (mode), and the natural frequency can be computed by taking the square root, the resulting values are in rad/s, the values have been converted to Hz in the table below:

Table 4 table of natural frequencies calculated using python and matlab slight difference in mode 4 is due to rounding errors by reporting the values to 4sig figs.

mode	matlab	python	type
mode 1	14.451	14.451	bending
mode 2	41.5079	41.5079	torsion
mode 3	70.2077	70.2076	bending
mode 4	132.8862	**132.885**	torsion
mode 5	204.5792	204.5792	bending
mode 6	288.7484	288.7502	torsion

Each mode corresponds to a mode shape described as eigenvectors representing displacements, the plots of the eigenvectors simulating the plate can be found in appendix D.

The modes are described as their shape and occurrence, and for example, mode 3 is the second bending mode. The concept of orders are important to note, where mode 3 order 2 for example is $2 \times 70.2077 = 140.4154Hz$

Solidworks results

A solution using the Solidworks simulation package has been produced using the plate dimensions but neglecting the bolt assembly, for this reason, the natural frequencies are not expected to be accurate but the visualisations have proven more useful than the sand on the plate surface. The displacement field has been plotted on the surface of the part where blue is the node position where the sand would collect in the lab. Red is max displacement.

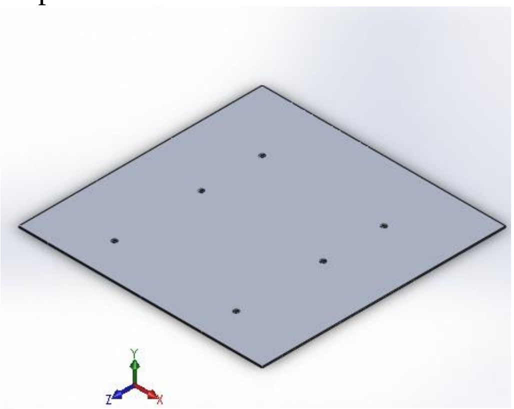


Figure 7 Plate geometry

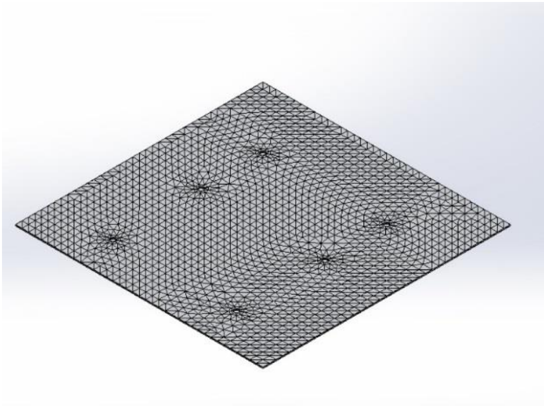


Figure 8 plate mesh

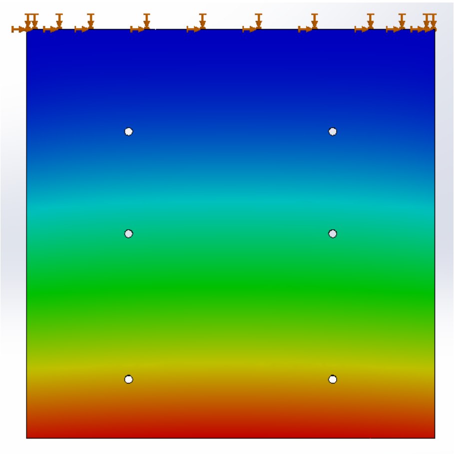


Figure 10 mode 1 displacements

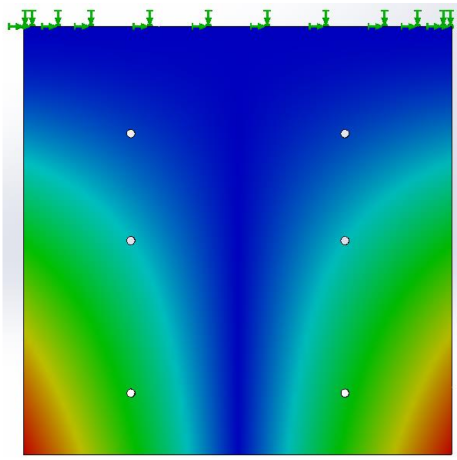


Figure 9 mode 2 displacements

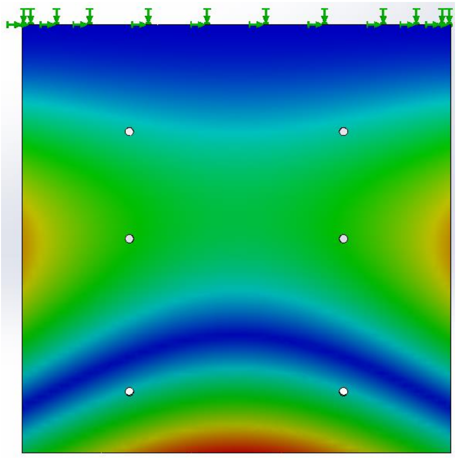


Figure 14 mode 3 displacement

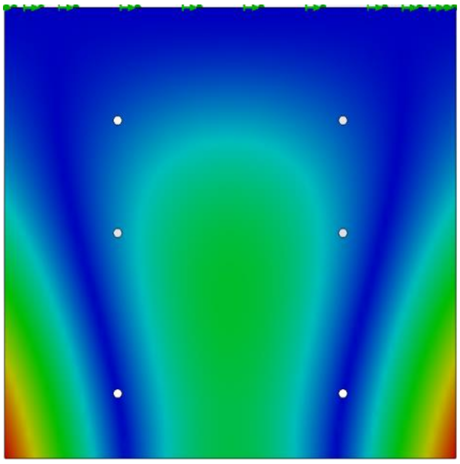


Figure 13 mode 4 displacement

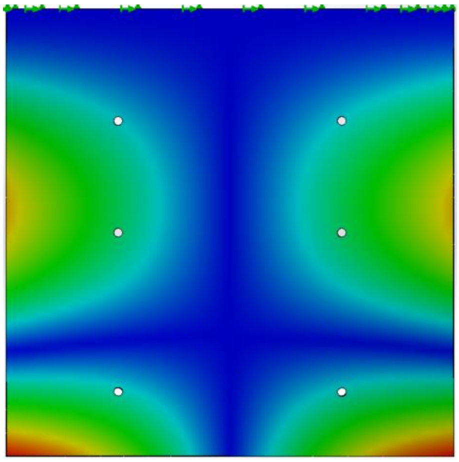


Figure 12 mode 5 displacements

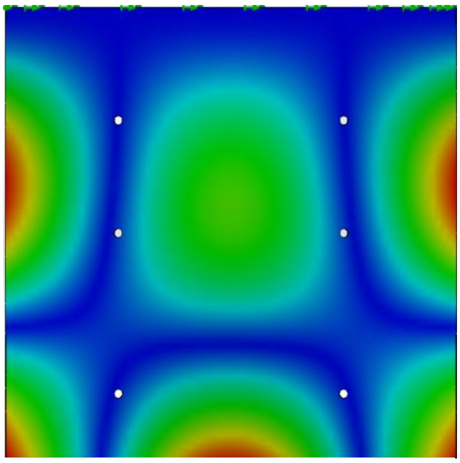


Figure 11 mode 6 displacements

All collected results

Table 5 table of all collected results

mode	matlab	python	Solidworks	experimental in lab	type
mode 1	14.451	14.451	14.748	19.5	bending
mode 2	41.5079	41.5079	35.65	36.3	torsion
mode 3	70.2077	70.2076	89.972	84.76	bending
mode 4	132.8862	132.885	115.09	124.4	torsion
mode 5	204.5792	204.5792	130.68	214	bending
mode 6	288.7484	288.7502	229.1	275	torsion
				109	additional
				244.8	additional
				354.3	additional

For the purposes of the lab the correct real values should be considered as the experimental results, the correct theoretically determined values should be considered the matlab or python values. In both cases, the model considers a 6-dof plate with bolts present; the Solidworks values should be considered as indicative only as the bolts mass is not included in the modal analysis.

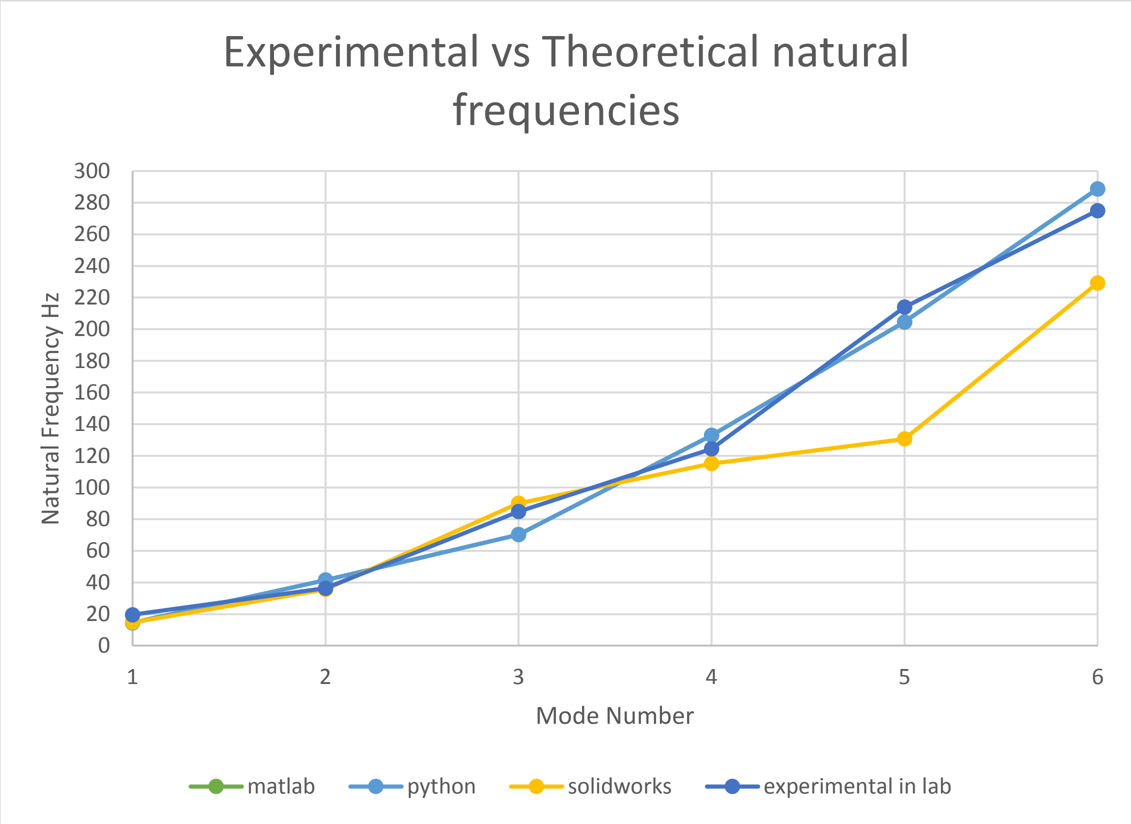


Figure 15 plot of natural frequency vs mode number

Discussion of Results

The results confirm that the matlab numerical implementation of the theoretical system model is a close relation to the experimental data; similarly, the python model reflects the results as expected (although there are some rounding discrepancies in the value of mode 4, highlighted in table 4.

The Solidworks model provides accurate predictions 1,2,3,and 4 however mode 5 and 6 results are too dissimilar to be considered representative, this is mainly due to modelling the whole mass distribution, and not a lumped mass model, a further reason is the lack bolts providing any concentrated mass at the 6 bolt hole points.

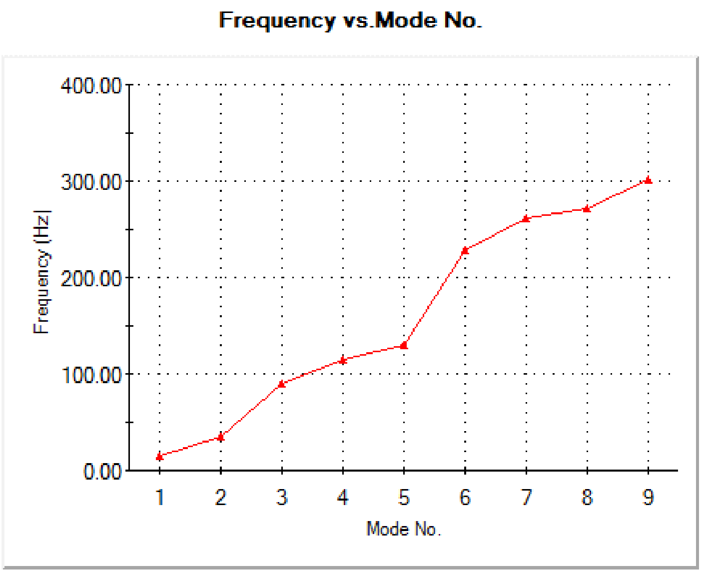


Figure 16 Solidworks natural modes

Solidworks reports more degrees of freedom than the matlab code as seen in figure 15, the additional modes noted in the experiment may correspond to these high modes.

Additional mode 244 Hz theoretically measured occurs between mode 7 and 8 when calculated in Solidworks modal analysis.

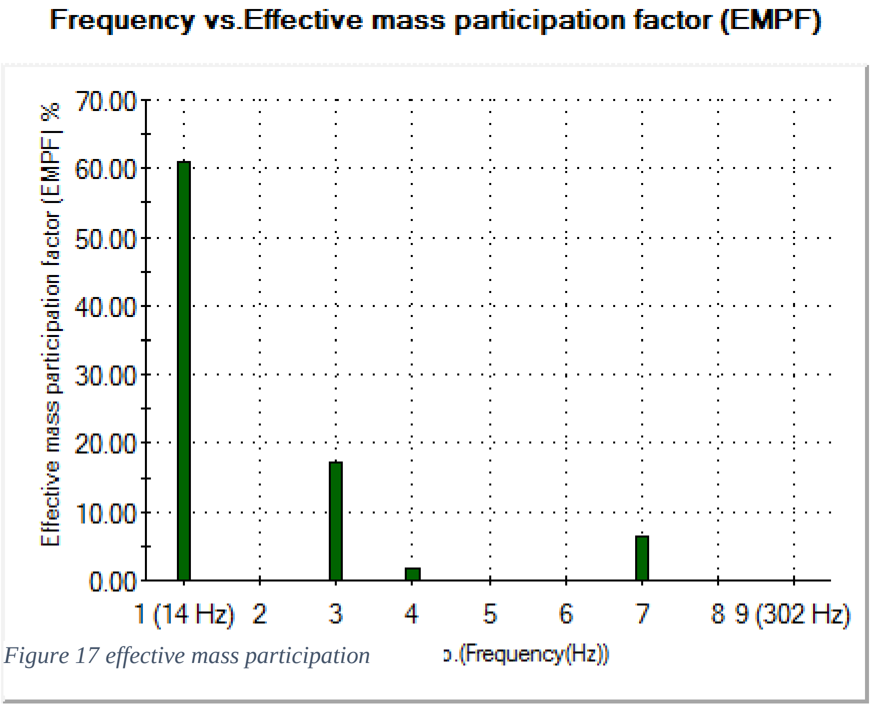


Figure 17 effective mass participation

Figure 16 shows the effective mass participation of the respective modes, this plot indicates the percentage of the mass being displaced each natural frequency, this is a potential indicator of the extent the natural frequency effects the plate and could serve to illustrate how issues such as fatigue and crack propagation might develop in practice.

When conducting a real modal analysis a cumulative graph can show to what extent do the analyzed frequency range capture possible resonance points,

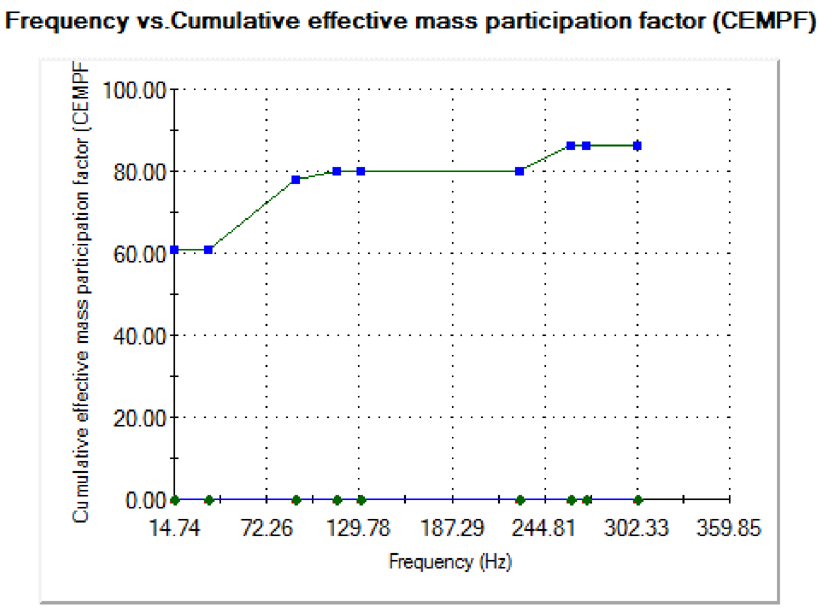


Figure 18 cumulative graph of mass participation

If this cumulative value is low, it is suggested to explore more modes that may cause unforeseen issues. In the analysis of structures one must include enough modes to account for 90 percent of the mass. (Wilson, 2006)

Table 6 Eigenvectors

	mode 1	mode 2	mode 3	mode 4	mode 5	mode 6
x1	-0.0435	0.0544	0.074	0.3644	-1.3054	-0.6035
x2	-0.0435	0.0544	0.074	0.3644	-1.3054	-0.6035
x3	-0.2925	0.3724	0.6381	0.4991	0.0854	0.335
x4	-0.2925	0.3724	0.6381	0.4991	0.0854	0.335
x5	-0.6423	0.5986	-0.2956	-0.3437	0.0087	-0.1535
x6	-0.6423	-0.5986	-0.2956	-0.3437	0.0087	0.1535

Table 6 shows the eigenvectors arranged in columns for each mode in mode 1 the first bending mode the displacements are steadily greater in the –ve y direction showing the tip of the plate bending the most(x5,x6) and the bolt holes near the clamp (x1,x2) bending the least.

The experiment has shown the difference between the theoretical limited degree of freedom model and the experimental testing of a infinite degree of freedom real world plate, the key points of the experiment are that a 6DOF model can capture the real world performance well, figure 15 shows the numerical solutions (python, matlab) correlate very closely with the measured results. Some natural frequencies within the range are not captured in the 6DOF solution as highlighted by the additional values in table 5, one of these values is similar to the Solidworks prediction 7^{th} and 8^{th} modes and may well be another natural frequency that is predictable, however solidworks' accuracy is questionable for the 5^{th} and 6^{th} modes and casts some doubt on this assumption.

Sources of Error

The discrepancy between the number of degrees of freedom of the model (6) and the real world case is an assumption that cannot completely and accurately capture the frequency response of the real system, the lumped mass distribution across the plate.

The measurement of the plate is a source of error, as the plate material is non-homogenous using a $volume \times density$ method to calculate the mass will not be as accurate as removing the plate from the clamp to determine the mass of the full plate with bolts assembled.

Using more bolts and more holes would allow the creation of a higher DOF system and produce a more accurate mass distribution matrix representing the system.

Within the code, ensuring a consistent use of data types to avoid rounding and truncation errors in the calculations should be considered.

(Wilson, 2006) Suggests a more accurate method of calculating the eigenvectors exists, stating that: "*In general, there is no direct relationship between the accuracy of the eigenvalues and eigenvectors and the accuracy of node point displacements and member forces.*"

Conclusions

The experimental and theoretical work has covered a simple situation involving a cantilever plate to demonstrate the differences in experimental and theoretical multi degree of freedom vibration analysis.

The theoretical results are in agreement with the experimental findings; however, future testing should make use of additional error mitigating initiatives such as the use of high degrees of freedom if only to develop a more accurate lumped mass matrix.

As indicated by (Wilson, 2006) the cumulative mass participation factor should be greater than 90% for a satisfactory modal analysis, more degrees of freedom could highlight natural frequencies of interest which may be important in design and reliability factors.

Bibliography

Ansys-help-files. (2016). *Ansys Help Files.*

Bathe, J. K. (1976). *NUMERICAL METHODS IN FINITE ELEMENT ANALYSIS.* Pentice Hall.

Courant, R. (1943). Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*.

Gavin, H. P. (May 13, 2014). *Mathematical Properties of Stiffness Matrices.* Department of Civil and Environmental Engineering Duke University.

Hartog, J. D. (1985). *Mechanical Vibrations.* Dover.

Meriam, K. (1993). *Engineering Mechanics Voume 2: Dynamics 3rd Ed.* Wiley.

Stroud, K. (2011). *Advanced Engineering Mathematics 5th Ed.* Palgrave Macmillan.

University-of-colorado. (2015). *Introduction to Finite Element Methods* . Boulder city, CO: Department of Aerospace engineering sciences.

Appendix A: porting matlab script to python

To aid in understanding the matlab code has been ported to the python programming language, the output of the matlab code has been used as validation.

Hz <1x6 double>

	1	2	3	4	5	6
1	14.4510	41.5079	70.2077	132.8862	204.5792	288.7484

Figure 19 Reported frequency results from MATLAB

Appendix B shows the ported python code, which outputs the correct frequency values for the theoretical values predicted using the experimental measurements to augment the compliance matrix.

```
----Python console output----
The mass of the bolt is 0.0093 kg
The density of aluminium is 2770 kg/m^3
The mass of the bolt hole (material removed) is: 0.0001 kg
The mass of plate is: 0.415 kg
the lumped system mass at each of the six points is: 0.0783 kg
compliance matrix average value 1 is: 37.6413330322 [a]
compliance matrix average value 2 is: 128.183998975 [c]
compliance matrix average value 3 is: 190.241331811 [b]
mode 1 is: 14.451 hz
mode 2 is: 41.5079 hz
mode 3 is: 70.2076 hz
mode 4 is: 132.885 hz
mode 5 is: 204.5792 hz
mode 6 is: 288.7502 hz
```

Figure 20 Python script output results

$$\begin{bmatrix} & & & & & a \\ & & & & a & \\ & & c & b & & \\ & & b & c & & \\ & a & & & & \\ a & & & & & \end{bmatrix}$$

Figure 21 illustration of the user added values

The compliance matrix has been augmented with the user calculated values and placed as illustrated and above in bold, where:

- a is the average of matrix items 16, 61, 52, and 25.
- b is the average of matrix items 43,34
- c is the average of matrix items 44,33

These values are the same as in figure 9 showing the matlab result.

```
[d,v] = numpy.linalg.eig(invLumpedMass*stiffnessMatrix)
```

The function takes matrix objects as the arguments and assigns two return variables with the eigenvalues or natural frequencies (d) and the eigenvector matrix

This operation yields:

d = [3291579.330, 1652276.842, 697126.125, 194593.345, 68017.6882, 8244.3792]

```
v =

   -0.0435    0.0544    0.0740    0.3644   -0.7019   -0.6035
   -0.0435   -0.0544    0.0740   -0.3644   -0.7019    0.6035
   -0.2925    0.3724    0.6381    0.4991    0.0854    0.3350
   -0.2925   -0.3724    0.6381   -0.4991    0.0854   -0.3350
   -0.6423    0.5986   -0.2956   -0.3437    0.0087   -0.1535
   -0.6423   -0.5986   -0.2956    0.3437    0.0087    0.1535
```

Which are the eigenvalues (in units of (rad/s)^2 and the corresponding eigenvectors.

Appendix B: Python code

```
#Vibration of a cantilever plate
#this python script is written as a port of the matlab script used in the
lab
#to determine the natural frequencies of the 6 degree of freedom system
import math #standard mathematical library
import numpy

#this version is the lab session code to determine the theoretical values

#functions
def format(x):
    return ('%.4f' % x).rstrip('0').rstrip('.')#function for formatting
to 4 decimal places and stripping redundant trailing zeros

def h(displacement):#calculate compliance values m/N x10^-6 converts
displacements to compliances for a given load
    load= 0.501 #load in units: kg
    compliance =1e6 *displacement/(1000*load*9.81)#constants here are
unit conversions
    return compliance

#functions
```

Figure 23 Python functions

```
#inputs
densAL=2770
holeRadius = 3 # mm
plateThickness = 1.6#mm
plateWidth= 306#mm
boltMass = 9.3*1e-3 # camelCase variable declarations

#this new section of code simplifies the compliance matrix augmentation
with the experimentally obtained results:
#input the measured compliance values here: be careful to enter absolute
(+ve) values in mm#
#always enter floats rather than integers if the value is 1 enter 1.0 for
example

#displacements in units: mm

d_16=0.2

d_25=0.19

d_33=0.93

d_34=0.63

d_43=0.63

d_44=0.94

d_52=0.22

d_61=0.13

#inputs
```

Figure 22 Python inputs

```
#calculations
boltHoleMass=math.pi*(holeRadius**2)*plateThickness*densAL*1e-
9;#variables should be self descriptive | e-9 is converting the m^3 to
mm^3 unit is kg

plateMass = densAL*plateWidth**2*plateThickness*1e-9#unit is kg

lumpedMass = plateMass/6 + boltMass -boltHoleMass

lumpedMassMatrix= numpy.eye(6)*lumpedMass #build a diagonal matrix with
the lumped mass value along the diagonal line, all over values 0
#calculations

#calculate the average compliance values for the flexibility matrix
avg1=numpy.mean(numpy.array([h(d_16),h(d_61),h(d_52),h(d_25)]))#better
way to take the average
avg2=numpy.mean(numpy.array([h(d_34),h(d_43)]))
avg3=numpy.mean(numpy.array([h(d_44),h(d_33)]))

flexibility=1e-6*numpy.matrix([
[11.5, 2.7, 28.7, 16.0, 46.0, avg1],
[2.7, 11.5,16.0,28.7, avg1,46.0],
[28.7, 16.0,avg3,avg2,317.0,240.0],
[16.0, 28.7,avg2,avg3,240.0,317.0],
[46.0, avg1, 317.0, 240.0,714.0,575.0],
[avg1, 46.0,240.0, 317.0, 575.0, 714.0],
]
)

#inverse the lumped mass matrix
invLumpedMass = numpy.linalg.inv(lumpedMassMatrix)
#inverse the compliance matrix stiffness matrix
stiffnessMatrix=numpy.linalg.inv(flexibility)

#compute the eigenvalue and eigen vector matrix for the two inverses
multiplied

[d,v] = numpy.linalg.eig(invLumpedMass*stiffnessMatrix)

#calculate the natural frequency(takes the rad/s values and reports the
converted hz values
hz = numpy.sqrt( (numpy.ones((1,6))*d))/(2*numpy.pi)#numpy ones just
flattens the diagonal matrix to become a collumn 6x1 matrix
```

Figure 7 python calculations section

```
print '----Python console output----'
print 'The mass of the bolt is '+str(format(boltMass))+' kg' #str is used
because the print keyword only takes arguements of the same type so bolt
mass is conveted from a float to a string
print 'The density of aluminium is '+str(densAL)+' kg/m^3'
print 'The mass of the bolt hole (material removed) is: '+
str(format(boltHoleMass))+' kg' #matches the matlab output
print 'The mass of plate is: '+ str(format(plateMass))+' kg' #matches the
matlab output
print 'the lumped system mass at each of the six points is: '+
str(format(lumpedMass))+' kg'

for i in xrange(0,6):#saves time typing multiple print lines, accesses
the hz(i) values stored in the array hz
    print 'mode '+str((i+1))+' is: '+str(format(hz.item(5-i))) +' hz'
```

Figure 24 python output section

Appendix C: MATLAB Printouts

h =

1.1500e-05	2.7000e-06	2.8700e-05	1.6000e-05	4.6000e-05	$\boxed{3.7641e-05}$
2.7000e-06	1.1500e-05	1.6000e-05	2.8700e-05	$\boxed{3.7641e-05}$	4.6000e-05
2.8700e-05	1.6000e-05	$\boxed{1.9024e-04}$	$\boxed{1.2818e-04}$	3.1700e-04	2.4000e-04
1.6000e-05	2.8700e-05	$\boxed{1.2818e-04}$	$\boxed{1.9024e-04}$	2.4000e-04	3.1700e-04
4.6000e-05	$\boxed{3.7641e-05}$	3.1700e-04	2.4000e-04	7.1400e-04	5.7500e-04
$\boxed{3.7641e-05}$	4.6000e-05	2.4000e-04	3.1700e-04	5.7500e-04	7.1400e-04

Figure 25 Matlab compliance matrix

Appendix D: Matlab Graphics

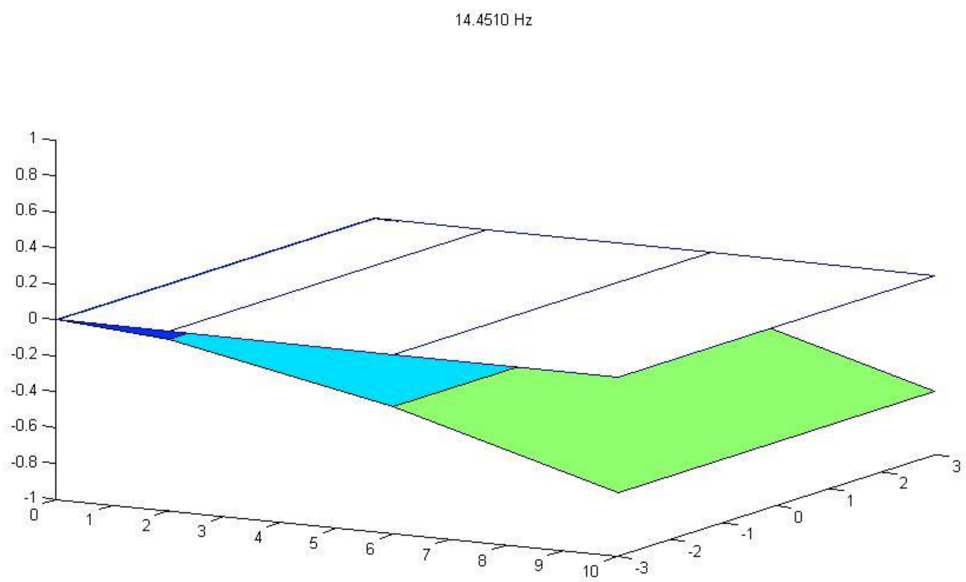


Figure 26 theoretical mode 1 visualisation

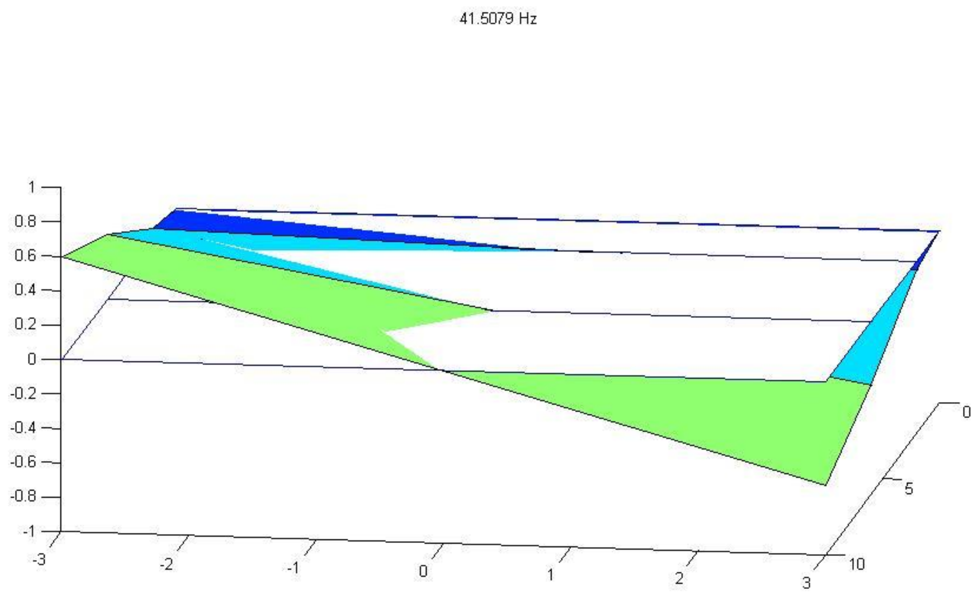


Figure 27 theoretical mode 2 visualisation

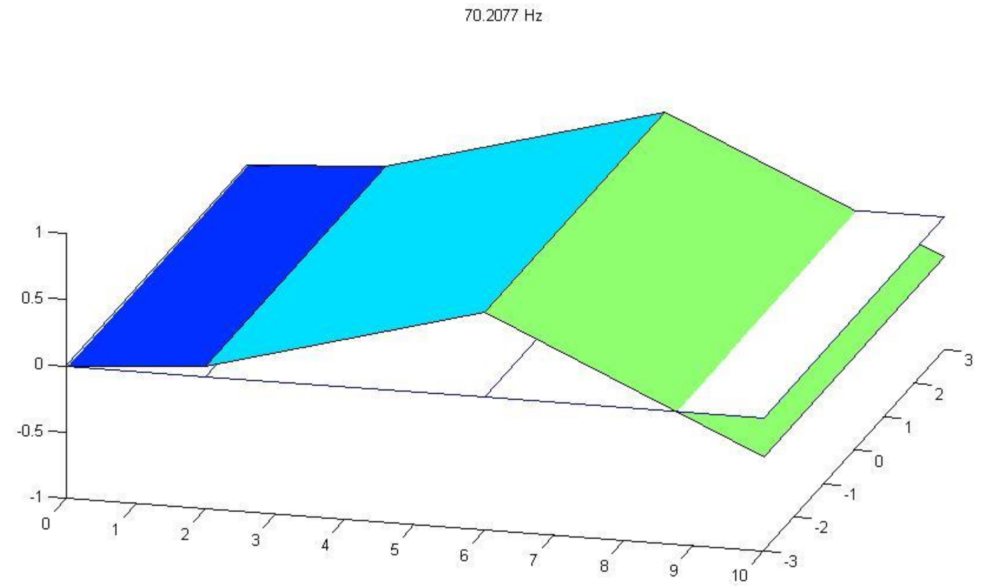


Figure 28 theoretical mode 3 visualisation

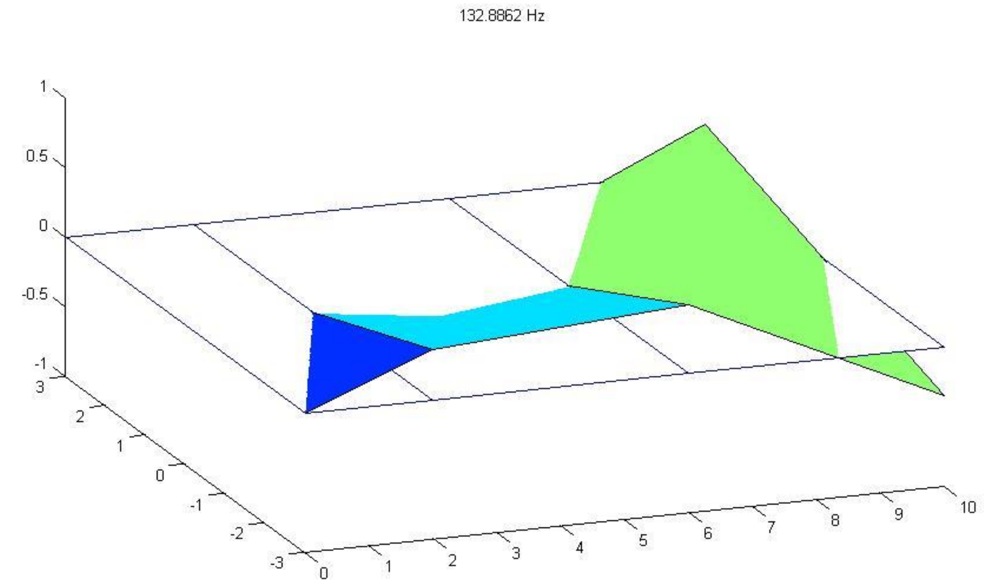


Figure 29 theoretical mode 4 visualization

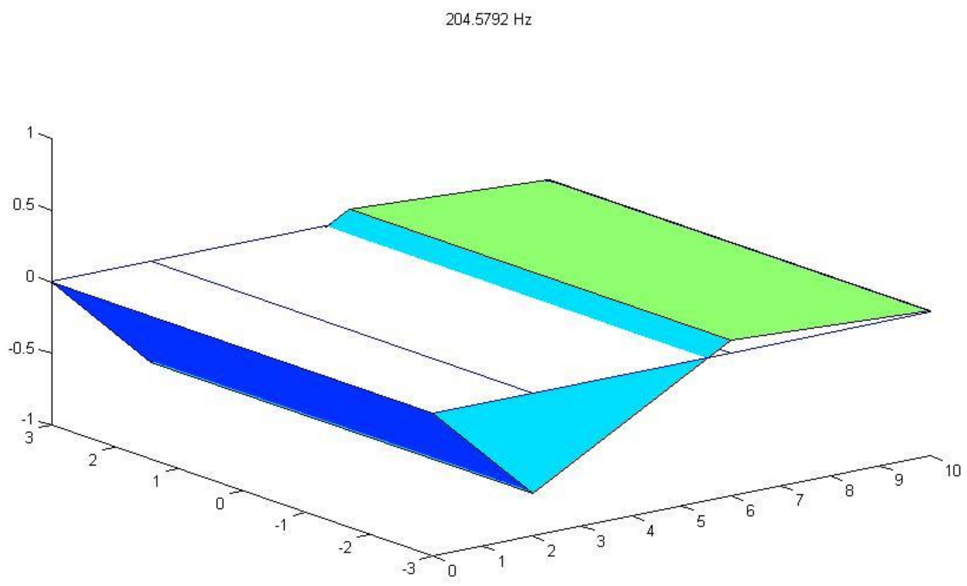


Figure 30 theoretical mode 5 visualisation

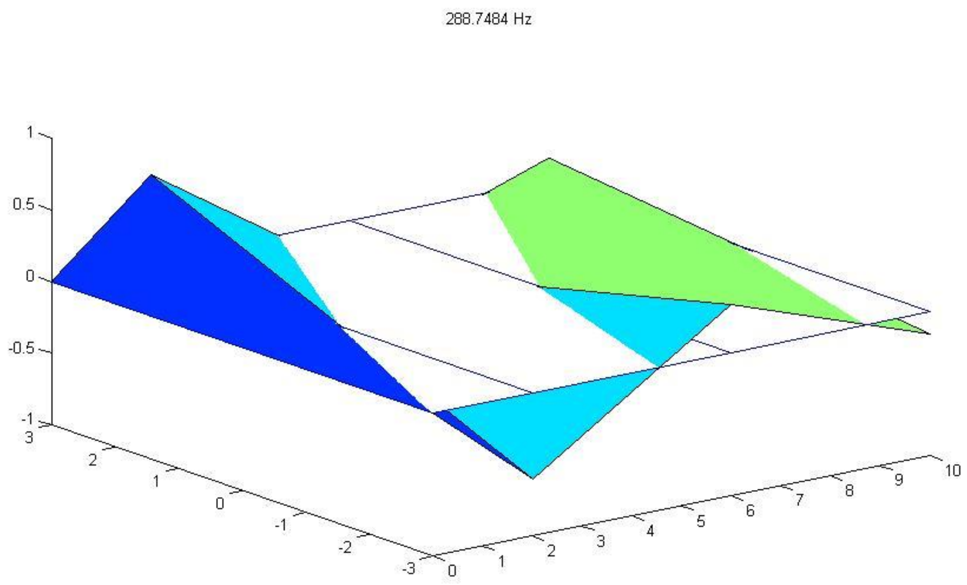


Figure 31 theoretical mode 7 6 visualisation

Appendix D: answers to lab questions

1. What is a normal mode of free vibration?

Normal mode vibrations are free vibrations that depend only on the mass and stiffness of the system and how they are distributed. A normal mode oscillation is defined as one in which each mass of the system undergoes harmonic motion of same frequency and passes the equilibrium position simultaneously.

How many normal modes are in a 2-dof system? (question 2i)

A 2 degree of freedom system will have two normal modes, their mode shapes will show the elements in two possible states, either synchronous or asynchronous.

2ii The equation for the first normal mode excited only will take the form:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} A_i \sin(2t + \theta_i)$$

2iii

The modal vectors are the column vectors $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and they describe the relative vibration amplitudes of the degrees of freedom.

2iv

This system contains 2 natural frequencies, because there a two degrees of freedom.

The first natural frequency is 2 and the second natural frequency is 5 the values are in radians/sec.

2v the values are arbitrary constants based on the initial conditions.

2vi $x_1$will be sinusoidal and x_2 will behave in a cosine manner due to being out of phase with x_1, the values $A_{i\ and\ A_j}$ will determine this.

3 the plate has a lumped mass model assumption using 6 masses to give the system 6 degrees of freedom

4 in reality the plate has practically infinite degrees of freedom and only approximations can be made using a model because it can only be formed with a reduced n-dof model.

5 experimentally more degrees of freedom were found because an infinite number exist and within the test range more than the first 6 existed, Solidworks simulation indicates one of the frequencies is between the 7th and 8th mode and might be one of those discovered additionally.

6 the eigenvectors correspond to the natural frequency squared in units of radians/second.

7 the Eigen vectors correspond to the relative amplitudes of the degrees of freedom and can be used to plot the mode shapes.