

The Computational Investigation of the Wind-Induced Vibration of Overhead Conductors

Avern Malcolm Athol-Webb

Submitted in fulfilment of the academic requirements for the degree of Master of Science in Mechanical
Engineering

November, 2013

Supervisor: Professor Glen Bright

Co-Supervisor: Dr. Richard Loubser

As the candidate's Supervisor I agree/do not agree to the submission of this thesis.

Signed: _____

Date: _____

DECLARATION

Ideclare that

- (i) The research reported in this dissertation/thesis, except where otherwise indicated, is my original work.
- (ii) This dissertation/thesis has not been submitted for any degree or examination at any other university.
- (iii) This dissertation/thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
- (iv) This dissertation/thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
 - a) their words have been re-written but the general information attributed to them has been referenced;
 - b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.
- (v) Where I have reproduced a publication of which I am an author, co-author or editor, I have indicated in detail which part of the publication was actually written by myself alone and have fully referenced such publications.
- (vi) This dissertation/thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation/thesis and in the References sections.

Signed: _____

ACKNOWLEDGEMENTS

I must extend my sincerest thanks to my Supervisor, Professor Glen Bright and my Co-Supervisor Dr. Richard Loubser for all their support and advice. I would also like to thank the Vibration Research Testing Centre for their financial support as well as Pravesh Moodley of the V.R.T.C for his assistance throughout this endeavour.

I would also like to thank the U.K.Z.N. Aerospace Systems Research Group (ASReG) for kindly making available their computer resources.

Dedicated to S and to K

For getting me going and keeping me going

ABSTRACT

The reliable transmission of high-voltage electricity through overhead electrical conductors is a cornerstone of the modern industrialised world. Any situation or phenomenon that impedes the functioning of this network warrants investigation. An example of which is the occurrence of aeolian vibration. This is caused by airflow over the conductor breaking off into alternating turbulent vortices. These vortices can cause an alternating lift force on the conductor, resulting in unwanted vibrations and damaging fatigue loading.

The Vibration Research Testing Centre (V.R.T.C.) of the University of Kwa-Zulu Natal is investigating the effects of this problem experimentally by oscillating an overhead conductor in a test facility. An electrodynamic shaker is used in a frequency and amplitude range equivalent to that produced by measured wind power input. This method is limited because only a single point force input to the conductor is possible as well as the limited span length. The aim of this research was to investigate the effects of aeolian vibrations and to develop a model that can verify the results of the V.R.T.C. This model can also be used to analyse scenarios that cannot be experimentally tested. A mathematical simulation of an overhead conductor subject to various wind power and single point oscillator inputs was developed. The mathematical simulation was performed using the MATLAB computing environment in the form of a finite element model. The model consists of a number of beam elements, arranged linearly to form a cable model, with suitable end conditions and driving inputs. The system was solved using a varying time-step 4th order Runge-Kutta solving method. The results of the model were compared to tests performed at the V.R.T.C. on a sample conductor length.

TABLE OF CONTENTS

LIST OF FIGURES.....	ix
LIST OF TABLES.....	xi
Ch. 1: Introduction.....	1
1.1 Introduction	1
1.2 Vibration Research Testing Centre	2
1.3 Problem Statement	3
1.4 Research Objectives	4
1.5 Research Publications	4
1.6 Dissertation Overview	4
1.7 Chapter Summary	5
Ch. 2: Literature Survey.....	6
2.1 Introduction	6
2.2 Aeolian Vibrations	6
2.2.1 Vortex Shedding	7
2.2.2 Strouhal Number	8
2.2.3 Lock-in	9
2.2.4 Aeolian vibration as a force input	10
2.3 Conductor Modelling	11
2.3.1 Energy Balance Principle and Power Method	11
2.3.2 Shaker Investigations	13
2.3.3 Finite Element Methods	14
2.3.4 Damping	15
2.3.5 Stiffness	16
2.3.6 Tension	17
2.4 Chapter Summary	18
Ch. 3: Numerical Simulation.....	19
3.1 Introduction	19
3.2 Self-Imposed Restrictions	19
3.3 Conductor Finite Element Analysis	20
3.3.1 Applying the Finite Element Method	20
3.3.2 The Planar Frame Element	22
3.3.2.1 Interpolation	23
3.3.3 Stiffness	23

3.3.3.1 Local Stiffness Matrix	23
3.3.3.2 Bending Stiffness	24
3.3.4 Mass Matrix	25
3.3.5 Damping	26
3.3.5.1 Rayleigh Damping	26
3.3.6 External Force Vector	27
3.3.6.1 Tension	27
3.3.6.2 Gravity	28
3.3.6.3 Distributed Wind Load	28
3.3.7 Boundary Conditions	30
3.3.8 Number of elements	31
3.3.9 Transformation Matrix	32
3.3.10 Element Assembly	33
3.4. The External Shaker	36
3.5 Initial Value	38
3.6 MATLAB	39
3.6.1 Structure of the simulation	39
3.6.2 ODE45 and Reduction of Order	39
3.6.3 Applying the Boundary Conditions	40
3.6.4 Shaker driving function	40
3.6.4.1 Constant Frequency	40
3.6.4.2 Time Varying Frequency	41
3.6.4 Convergence and Time to Solve	43
3.7 Chapter Summary	45
Ch. 4: Results and Discussion.....	46
4.1 Introduction	46
4.2 Method of Investigation	46
4.2.1 Conductor Type	47
4.2.2 Varying the Tension	47
4.3 Steady State Displacement	48
4.3.1 Catenary equation	48
4.3.2 Catenary Compared To Model and Measured Results	49
4.3.3 Model vs. Measured Results	51
4.4 Vibration Measurement	53
4.5 Swept Frequency Shaker Excitation	53
4.5.1 Frequency Sweep Data Sampling	53

4.5.2 Rabbit Sweep Results	54
4.5.3 Pelican Sweep Results	57
4.6 Constant Frequency Shaker Excitation	59
4.7 Steady vs. Sweep Excitation	63
4.8 Wind Force Excitation	66
4.9. Factors Influencing the Model Results	68
4.9.1 Effects of Linearisation	68
4.9.2 Effects of damping	68
4.9.3 The effects of bending stiffness	69
4.10 Effectiveness of Model	69
4.10.1 Distributed vs. Single Point Loading	69
4.10.2 Steady State Position	69
4.10.3 Swept Frequency Results	70
4.10.4 Constant Frequency Results	70
4.11 Chapter Summary	71
Ch. 5: Case Study Results Comparison.....	72
5.1 Introduction	72
5.2 Frequency Response Function	72
5.3 Test Setup	72
5.4 Results	73
5.5 Chapter Summary	75
Ch. 6: Conclusions.....	76
6.1 Introduction	76
6.2 Research Outcomes	76
6.3 Limitations	78
6.4 Recommendations	78
6.5 Research Summary	79
REFERENCES.....	80
APPENDICES.....	83
A: Interpolation Polynomials	83
B: Results Graphs	84
C: MATLAB Coding	95
D: Conductor Specifications	145
E: Shaker Specifications	146

LIST OF FIGURES

1.1:	Example of Conductor Fatigue Failure	2
2.1:	Vortex Development in a Trailing Wake	8
2.2:	Variation of Strouhal Number with Reynolds	9
2.3:	Maximum Power Input Coefficients as a Function of Relative Amplitude	12
2.4:	Test Setup for Shaker Investigations	13
2.5:	Side View and Cross Section of Typical Conductor Section	16
3.1:	X vs. Y Displacement at Various Reduced Frequencies	20
3.2:	The Local Coordinate System	21
3.3:	Degrees of Freedom	22
3.4:	Conductor Cross Section	25
3.5:	Values for “a” Constant	26
3.6:	Values for “b” Constant	27
3.7:	Tension as applied to elements	28
3.8:	Damping vs. Mode Numbers for 10700N	29
3.9:	Damping vs. Mode Numbers for 15860N	29
3.10:	Conductor Clamping Mount	31
3.11:	Local vs. Global Coordinates	32
3.12:	Conductor Element Assembly	33
3.13:	Matrix Assembly	34
3.14:	The TIRAvib shaker connected to “Tern”	36
3.15:	Flexible Coupling Connected to “Tern”	37
3.16:	Flexible Coupling Stiffness	38
3.17:	Total Frequency Sweep	42
3.18:	Close View Showing Steps	42
3.19:	Time to Solve vs. Number of Elements	43
3.20:	Number of Elements vs. Relative Accuracy	44

4.1:	Catenary Curve, $a = 20$	48
4.2:	Comparison Between Model and Catenary Curve (Tern)	49
4.3:	Comparison Between Model and Catenary Curve Close View (Tern)	50
4.4:	Comparison Between Measured and Catenary Curve (Tern)	50
4.5:	Rabbit Steady Model Results	51
4.6:	Rabbit Steady Measured Results	52
4.7:	Accelerometer Positions	53
4.8:	Rabbit 2775N Sweep, Model Vs Test	54
4.9:	Rabbit 3700N Sweep, Model Vs Test	55
4.10:	Rabbit 4630N Sweep, Model Vs Test	56
4.11:	Pelican 8120N Sweep, Model vs. Test	57
4.12:	Pelican 11200N Sweep, Model vs. Test	58
4.13:	Pelican 13070N Sweep, Model vs. Test	58
4.14:	Constant Frequency Displacement - Rabbit 2775N	60
4.15:	Constant Frequency Displacement - Rabbit 3700N	60
4.16:	Constant Frequency Displacement - Rabbit 4630N	61
4.17:	Constant Frequency Displacement - Pelican 8120N	61
4.18:	Constant Frequency Displacement Pelican 11200N	62
4.19:	Constant Frequency Displacement – Pelican 13070N	62
4.20:	Pelican 13070N Steady vs. Sweep Test Results	63
4.21:	Pelican 8120N Steady vs. Sweep Test Results	64
4.22:	Pelican 11200N Steady vs. Sweep Test Results	64
4.23:	Rabbit 2775N Steady vs. Sweep Test Results	65
4.24:	Rabbit 3700N Steady vs. Sweep Test Results	65
4.25:	Rabbit 4630N Steady vs. Sweep Test Results	66
4.26:	Pelican Modelled Wind Response	67
4.27:	Rabbit Modelled Wind Response	67
5.1:	Experimental Frequency Response Function of Grosbeak	73
5.2:	Grosbeak 16481N FRF Model Results	74
5.3:	Grosbeak 21778N FRF Model Results	74

LIST OF TABLES

4.1:	Conductor Specifications	47
4.2:	Tension Settings per Conductor	47
4.3:	Measurement Points Along Span	49
4.4:	Rabbit Steady State Results Comparison	52
4.5:	Pelican Steady State Results Comparison	52
4.6:	Tern Steady State Results Comparison	52
5.1:	Conductor Specifications	73

CHAPTER 1

INTRODUCTION

1.1 Introduction

The modern world is dependent on a functioning and effective electrical generation infrastructure that can reliably produce and deliver electricity to the end user without significant outages. A major component of this delivery network is the overhead high voltage conductor cable. Supported by pylons, these conductors transmit high voltage electricity from the generation source to municipal substations where it is then step down transformed to lower voltages before distribution further to local users. South Africa alone has “...over 25000km of overhead power lines with voltages of 132 kV to 765 kV.”¹ It is for this reason that any situation or phenomena that impedes the functioning of this network warranted further investigation. Such a problem is that of aeolian vibrations.

The term “aeolian vibration” refers to the occurrence of wind passing over and around a taut cable, causing the cable to shed multiple alternating vortices in the cables wake. These vortices cause a lift force with alternating sign to act on the cable due to an unsteady pressure distribution on the surface, resulting in a rhythmic oscillation in a direction normal to the wind flow direction. If the frequency of this vortex shedding matches a resonant frequency of the cable, the lift force can cause notable vibrations to develop. The amplitude of the vibrations are relatively small with a maximum amplitude in the order of 1 diameter of the conductor [1]. However, due to the frequency at which it occurs, there can be enough energy to cause noticeable fatigue damage to the cable if left unchecked, usually most obvious at the cable-tower interface. This will reduce transmission effectiveness and can eventually lead to complete cable failure. Figure 1.1 shows an example of near catastrophic fatigue failure of a conductor.

¹Eskom Power Series - Eskom Research Group <http://www.eskom.co.za> (20/09/2011)



Figure 1.1: Example of Conductor Fatigue Failure (Image Courtesy of V.R.T.C.)

This problem is not a new one, with workable solutions having already been found for certain aspects of it. But a number of aspects of the behaviour of conductors under the effect of aeolian vibrations remain unclear and the best way to analyse the behaviour of conductors under sinusoidal energy input from an air flow is to apply a similar force using a shaker. The V.R.T.C. tests conductors by using an electrodynamic shaker to do this, allowing various physical parameters of a conductor to be tested.

This is a time-consuming process that requires a section of conductor of limited length to be erected and tested, which is an expensive prospect and potentially unnecessary for certain investigations. A number of physical limitations are inherent in the test setup, such as set span length (84.6m for V.R.T.C.); an inability to apply a distributed load to the conductor under test; size considerations of the shaker restricting the shakers placement along the length of the conductor and a limitation of the frequency range of the shaker (specifically below 5Hz for the TIRAvib shaker used). It was these considerations that lead to the proposed research objectives.

1.2 Vibration Research Testing Centre

Launched in 2000, the V.R.T.C. is a testing facility situated on the Westville campus of the University of Kwa-Zulu Natal. One of only four such facilities in the world and the only one of its kind in the southern hemisphere, it provides a valuable resource for investigating the

behaviour of vibrating conductors within a controlled environment. It has the capability of testing a conductor with a span of 84.6m in a temperature controlled room, free from external interference effects. The end connection points for the span are mounted onto large concrete blocks set deep into the foundation, preventing movement of the boundary points. The facility was built in line with recommendations from the IEEE Std 563-1978 for the Measurement of Conductor Self-Damping.

The V.R.T.C. provided a valuable resource for this research by granting a method of comparing the results of the computational model to experimental data. Experiments performed at the V.R.T.C. were replicated using the computational model to allow a closeness of fit between model and real to be determined and providing insight into possible deficiencies in the simulations.

1.3 Problem Statement

The investigation and characterisation of the behaviour of conductors when subjected to different force inputs and boundary conditions is a topic worthy of consideration due to the impact that a failure of a conductor can have to various downstream consumers. A common scenario that develops is a cyclic pattern of vibrations due to a fluid-structure interaction known as aeolian vibrations in which a resonance condition results from wind energy input in the form of a fluctuating pressure coefficient. These vibrations can substantially decrease the lifespan of an installed conductor through increasing the fatigue loading. A number of investigations have been performed into various sub-components of this problem, such as the fluid flow interaction with the conductor, analysis of conductor geometry and its effects on conductor dynamics, including self-damping.

These investigations are a basis for the development of a computational model that can provide a method to analyse the motion of a conductor under a range of inputs force inputs. This model should be capable of using the supplied physical parameters of a conductor as given by the manufacturer as the model inputs combined with the chosen span length and position as well as the applied external tension and produce a good approximation of the conductor's response to wind force and single point shaker conditions.

1.4 Research Objectives

The research objectives were:

1. To perform a comprehensive literature survey of relevant research in the field.
2. To research and develop a computational model capable of providing a first order approximation of the behaviour of overhead conductors under a variety of loading conditions and compare this to results from the V.R.T.C.
3. Use the computational model to analyse the effects of using a shaker platform with flexible coupling to apply a force input to the overhead conductor.
4. Analyse and discuss the differences between a single point load vs. a distributed load over a conductor length.
5. Conclude with recommendations for further research.

1.5 Research Publications

One research publication was produced as listed below:

A. Athol-Webb, G.Bright, R. Loubser. “Computational Investigation of the Wind-Induced Vibration of Overhead Conductors”.18th International Congress on Sound and Vibration, Rio De Janerio, Brazil. July 2011.

1.6 Dissertation Overview

Chapter 2

In this chapter the literature relevant to the research project was presented and discussed. The role of aeolian vibrations as a force input to the overhead conductor was detailed. Various equations of motion for conductors were presented and the method chosen for this analysis was introduced.

Chapter 3

The finite element model for the investigation of the conductor motion was analysed in detail. The stiffness, mass and damping matrices were derived for this model including the effects of conductor properties as moment of inertia and externally applied tension. The method of implementing the model in MATLAB was shown.

Chapter 4

This chapter presented the results of tests performed at the VRTC and compared them to results obtained by the computational model. The effects of the shaker as a force input are considered and compared to a distributed load. The use of the model to investigate conductors in scenarios not replicable in the VRTC was demonstrated.

Chapter 5

The model was compared to published results of a conductor under test using a single-point shaker. The frequency response function for the conductor under test at different tensions were found using the model results and compared against the results of the published results of Castello & Matt.

Chapter 6

The outputs of the research project were summarised and compared against the stated aims and objectives. Possible shortcomings were addressed and recommendations for future research were outlined as well as avenues for expansion of this specific research area.

1.7 Chapter Summary

A brief overview of the contents of this research dissertation has been presented and the relevance of the research to industry was considered. The nature and capabilities of the V.R.T.C. was outlined and its importance to this research project. The problem space has been defined and the research objectives were outlined within the scope of the problem space.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

The subject of aeolian vibrations and its effect on conductors is one that has received a large amount of attention over the years. A number of studies have been performed attempting to characterise the nature of aeolian vibrations and to determine methods to replicate and prevent the damaging results that such vibrations can cause if significant aeolian vibrations develop.

In this chapter the nature of aeolian vibrations is discussed with the various factors that affect the formation and behaviour of vortex induced conductor vibration presented. In the second part a review of important concepts and investigations into the modelling of the conductors was presented. An overview of the important physical parameters of the conductor motion was considered.

2.2 Aeolian Vibrations

A strong wind is capable of causing damage to structures through overloading them in direct loading. Wind is also capable of causing damage by inducing a dynamic response, when a wind of insufficient strength to produce direct damage interacts with the structure to produce a resonant response resulting in unwanted and potentially damaging vibrations. It is this dynamic response that will be detailed further.

According to Riera & Davenport (1998) there are three main sources of aerodynamic excitation which result in a dynamic response of the structure:

- a turbulent flow which can cause resonant responses in the direction of the flow and perpendicular to the flow
- a constant or near constant flow which is at a flow rate that will result in vortices being shed in the wake of the structure, causing a perpendicular vibration characteristic
- structural motion interacting with the air, which can result in positive or negative aerodynamic damping [2]

The second condition was the most important to this research, the phenomenon known as aeolian vibration.

2.2.1 Vortex shedding

According to Houghton & Carruthers (1976) the energy input from the wind is provided by the occurrence of vortex shedding as wind flows around the conductor. For most investigations of single conductors the theory is based on vortex shedding from a uniform cylinder. This shedding is dependent on the Reynolds number

$$R_e = U_\infty D / \nu \quad (2.1)$$

Where U_∞ is the free-stream fluid flow, D is the diameter of the cylinder and ν is the kinematic viscosity.

Through a Reynolds number range of 1 to 140, a boundary layer develops around the conductor and a turbulent trailing wake forms as a pair of separation points appear in the flow at the conductor surface on the trailing side of the conductor. As the Reynolds number approaches approximately 140, the turbulent wake broadens as the separation points move further towards the leading side of the conductor, until a symmetrical vortex pair appears in the now much larger turbulent wake. When the Reynolds number grows above 140 the symmetric separation points and vortices destabilise and are no longer in symmetry. This destabilisation causes one of the vortices to break off in an alternating pattern when it reaches sufficient strength whilst the other vortex begins to grow. These vortices shed at equal strength with equal spacing between them as they are carried away from the conductor after having broken away. Importantly these vortices have an alternating sign about the axis parallel to the free stream flow. [3]

According to the American Society of Civil Engineers (1987) this phenomena was documented by Theodore von Karmon and his name is now associated with the vortices being carried away, the von Karmon “street” or “wake”. After a Reynolds number of roughly 5×10^5 , boundary layer begins transitioning to turbulent and the vortex shedding transitions to random shedding until a Reynolds number of roughly 3×10^6 is reached, when the shedding frequency returns to a regular pattern as full turbulence is reached. Figure 2.1 shows the various stages of vortex formation and shedding over a range of Reynolds numbers. [4]

This research focuses on the aeolian vibration of single conductors, as opposed to conductor galloping which is a situation most common in bundled conductors. Conductor galloping is characterised by a lower frequency and higher amplitude of vibration as compared to aeolian vibrations. [5]

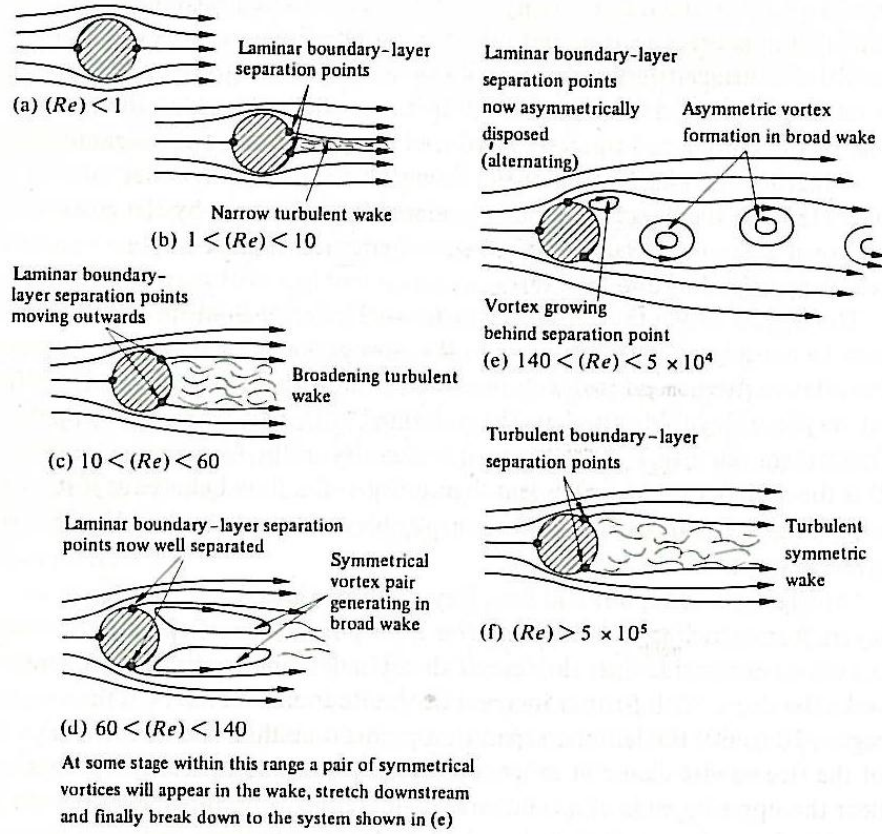


Figure 2.1: Vortex Development in a Trailing Wake (Houghton & Carruthers (1976))

These vortices are regions of lower pressure relative to the free stream flow. This results in a time varying coefficient of pressure that causes an oscillating fluid force to act on the conductor in the across-wind direction. Riera & Davenport (1998) suggested that this fluctuating force oscillates with a frequency equal to the frequency of vortex shedding, f_s . [2]

2.2.2 Strouhal Number

According to the American Society of Civil Engineers (1987) an empirical relation was found by Strouhal to relate the frequency at which the vortices are shed and by implication the frequency at which the fluid force is applied to the conductor. This relation is given as

$$S = \frac{f_s D}{U_\infty} \quad (2.2)$$

where the Strouhal number S is an empirical constant dependent on the cross-section and Reynolds number among others. For the case of a circular cylinder the Strouhal number is given as roughly 0.185 for the lower Reynolds numbers and this is the most common number used for investigations. Figure 2.2 shows the change in Strouhal number as Reynolds number increases and the flow transitions from laminar to turbulent flow. The periodic regime is

associated with the laminar flow and the regular regime is associated with turbulent flow. The random regime of Strouhal numbers occurs during the transition phase between the flow types. For this research the focus was on the laminar flow as the active regime of air flow.

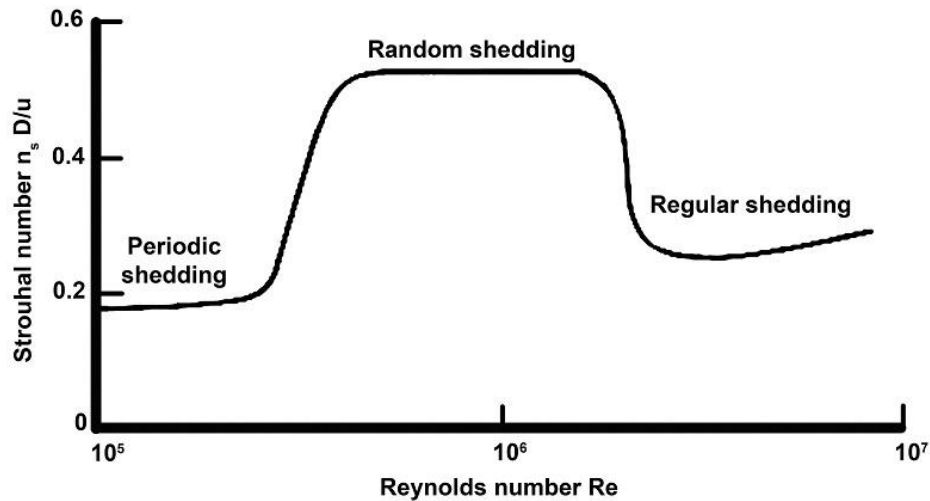


Figure 2.2: Variation of Strouhal Number with Reynolds (American Society of Civil Engineers (1987))

The importance of the Strouhal number is as a base for determining the frequency at which the force from aeolian vibrations is applied to the conductor. As the frequency of vortex shedding approaches the natural frequency of the conductor, the amplitude of vibrations increase to a peak value when the frequency of the exciting force from the wind and the resonance frequency of the physical system align. This leads into a condition known as “Lock-in”. [4]

2.2.3 Lock-in

According to Sarpkaya (2004), as the free stream wind speed U_∞ increases in speed, noticeable vibrations begin to develop as the shed vortices impart energy to the conductor. As the frequency of the exciting wind force nears the resonant frequency of the conductor, the vibration amplitude increases. When the vibration amplitude reaches a certain level, the motion of the conductor begins to have an effect on the vortices being shed from it, resulting in a form of self-excitation. Significantly, a condition called lock-in arises in which even as the wind velocity increases and the vortex shedding frequency should move away from the system resonant frequency, it does not but rather the vortices are still shed at or near the system resonant frequency. This occurs when short occurrences of periodic conductor motion cause the conductor to reach the higher amplitude levels. At these higher levels the steady state vibration occurs and lock-in develops. This condition only ceases when the wind speed changes enough for the lock-in to break down. [6]

The lock-in effect is important when considering the force input to the conductor, as lock-in means that minor variations in wind speed at or near the resonant response frequency do not cause the vortex shedding frequency to change, but instead the response frequency will remain constant. The conductor controls the shedding frequency within this region of meta-stability. This implies that once a lock-in condition is reached the conductor is likely to remain in that state unless a large enough shift in wind speed occurs to disrupt the wake. This lengthens the time at which a conductor is exposed to the peak or near peak vibration amplitudes.

Sarpkaya (2004) suggested that the response at lock-in is self limiting in that the co-efficient of lift due to vortex shedding increases to a maximum for an amplitude to displacement (A/D) ratio of 0.5, after which it decreases rapidly. [6]

2.2.4 Aeolian vibration as a force input

Analyses and treatments of vortex induced vibrations are focused on the fluid effects of the wind as a force on the cylinder in a flow, with the cylinder itself being treated as a single mass object using a single degree of freedom lagrangian equation of motion. The cylinder is sprung and the damping effects are viscous, with the wind as a force input. According to Sarpkaya (2004), this gives the classic equation

$$m\ddot{y} + c\dot{y} + ky = \frac{1}{2}C_y\rho DLU_\infty^2 \quad (2.3)$$

Where y is the displacement transverse to the wind flow, ρ is the density of the fluid, L is the length of the cylinder; m is the mass, c the damping coefficient and k the spring stiffness. This equation is non-linear, due to the amplitude dependence of the lift coefficient C_y . But for small values of the ratio A/D of a sinusoidally excited cylinder at the lock-in frequency, the harmonic oscillator model is commonly used in which the force is ahead of the resultant displacement by a phase angle φ . For a cylinder vibrating in air, the phase angle φ can be obtained from

$$\tan \varphi = 2\zeta \quad (2.4)$$

with ζ being the viscous damping factor.

The harmonic oscillator model is linear when

$$C_y = C_L \sin(2\pi f_s t + \varphi) \quad (2.5)$$

The coefficient C_L is a quantity dependant on a number of factors such as Reynolds number amongst others as noted in the previous sections. [6] For this research the values of C_L were taken from literature values determined using a non-linear finite element flow investigation using direct numerical simulation by Mittal and Kumar (2001) [7].

The linear harmonic oscillator model leads to a response that can be represented by

$$y_r = (A/D) \sin 2\pi f_s t \quad (2.6)$$

with y_r being the reduced transverse displacement. Moe and Wu (1990) suggested that the response is not perfectly sinusoidal. The synchronisation between the force and response is not complete, which can result in variations in peak force up to 10% [8]. How this affected the simulations is detailed in chapter 3.

2.3 Conductor Modelling

The large size and scale of the conductors involved in this problem as well as the their positioning as being strung up between pylons render performing physical experiments a challenging proposition. This has led to a number of theoretical investigations using a range of methods such as finite element analysis and analytical methods. The specific nature of the field also led to the development of measurement techniques and behavioural approximations tailored to investigations of conductor behaviour, such as the Energy Balance Principle (EBP).

2.3.1 Energy Balance Principle and Power Method

The power input from the wind is a measure of the amount of energy injected into the system by the wind due to vortex shedding. This power input has been evaluated in a number of different studies to be a function of the conductor diameter D , the amplitude A and frequency f of the resultant vibration, expressed as

$$P_{input} = f^3 D^4 fnc\left(\frac{A}{D}\right) \quad (2.6)$$

with $fnc\left(\frac{A}{D}\right)$ being one of the functions shown in Figure 2.3, as determined by different researchers [9]. As Figure 2.3 shows, although different researchers have used a variety of different methods to analyse this behaviour, there is a general agreement as to the relationship between relative amplitude and wind power input.

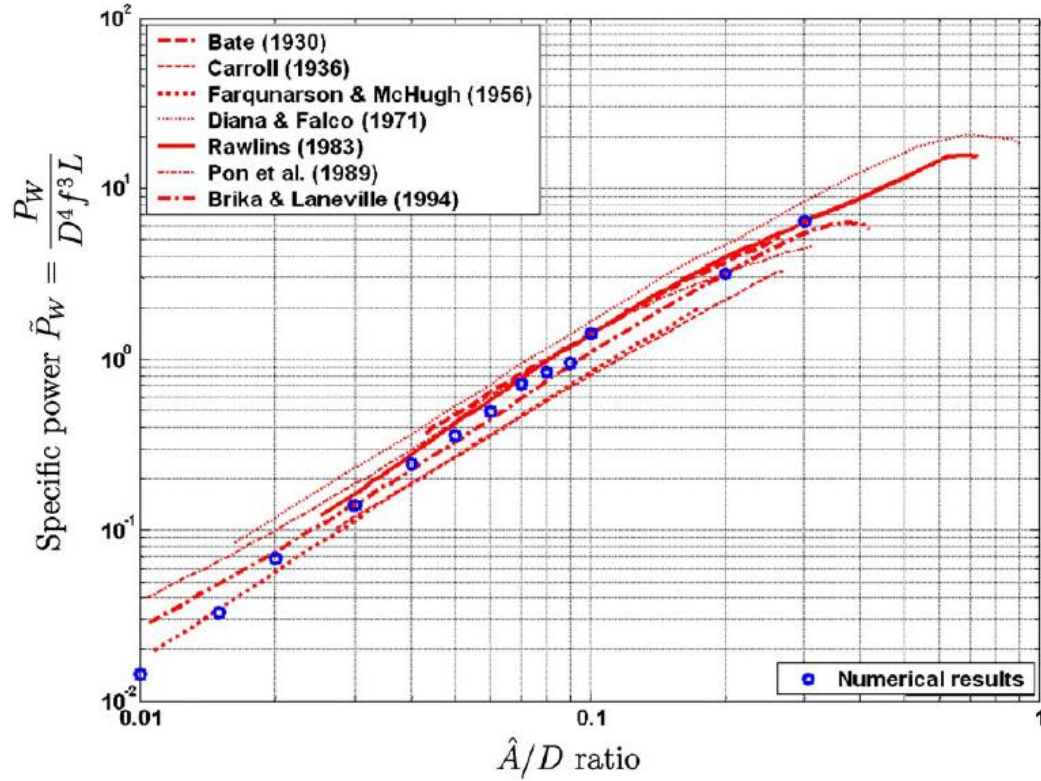


Figure 2.3: Maximum Power Input Coefficients as a Function of Relative Amplitude (Meynen et al. 2005)

The Energy Balance Principle as introduced by CIGRE, is the principle that at steady state vibrations the energy introduced by the wind due to vortex shedding is equivalent to the energy dissipated by the conductor due to damping. As suggested by EPRI (2006) this works best when the wind energy input is a pure sinusoidal input, but due to variations along the span of the cable and in the amplitude this does not occur. Beat patterns in which the amplitude at an antinode rhythmically increase and decrease with a set beat frequency are more likely to occur. The EBP can be used when considering an undamped conductor or damped conductor as well as bundled conductors.

According to Vechiarelli, Currie and Havard (2000), the energy balance principle does not consider the effects of flexural rigidity, which would reduce the resultant bending amplitude. It also implies the energy balance principle does not consider the importance of travelling wave effects. This may not be crucial for a steady state investigation but must be considered when force inputs change their frequency with time, or if a single point force such as a damper or shaker is introduced. Vechiarelli, Currie and Havard (2000) developed an analytic equation solved using a finite difference scheme to analyse the motion of a stockbridge-type damped conductor subject to aeolian vibrations. The model assumed a minimum bending stiffness and a

harmonic distributed force input due to steady state lock-in of the vortex shedding. The conductor was assumed to be a homogenous cylindrical body and that the theory of thin beam is applicable. The results of the simulation were compared against results using the energy balance method. The energy balance method predicted a larger peak to peak displacement but as mentioned does not include some important parameters. [10]

This consideration as well as the potentially wide range of possible applications led to a finite element analysis being chosen as the most suitable method for the investigation of the behaviour of conductors under test at the VRTC as well as being flexible enough to use for other scenarios.

2.3.2 Shaker Investigations

The use of an external shaker applied at a single point to oscillate the conductor is an effective way to excite a conductor when investigating conductor motion subject to a sinusoidal force. Recommendations on the use of a shaker to investigate conductor motion are provided in the IEEE Std 563-1978 “The IEEE Guide on Conductor Self-Damping Measurements”. Figure 2.4 shows the recommended method of testing a conductor using a shaker device.

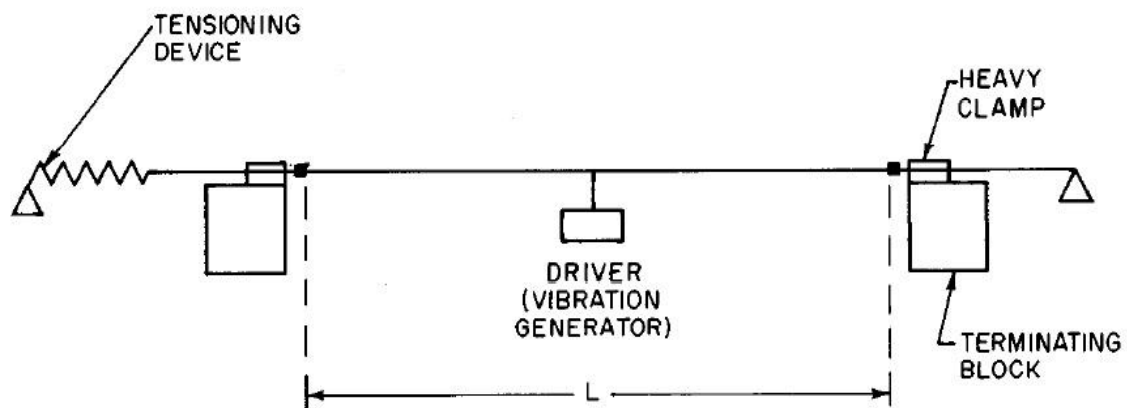


Figure 2.4: Test Setup for Shaker Investigations (IEEE Std 568)

The V.R.T.C. uses this setup for testing conductors, but due to physical limitations cannot place a shaker at the centre of the span. Connecting a shaker to a conductor comes with certain problems, mainly related to the effects of the shaker mass on the moving mass of the conductor itself. For shakers of the electro-mechanical type it is recommended by IEEE 563-1978 that the mass of the moving armature be less than 10% of the mass of the conductor being tested. This is especially important when a stiff connection between armature and conductor is used. As suggested by the Canadian Electrical Association (1988) the effect of armature mass also increases as the stiffness of the conductor increases.

A flexible coupling is a potential solution to some of these problems by isolating the armature from the conductor, so that a conductor-alone resonance can be determined. Care must be taken however to ensure that the flexible coupling is aligned to prevent the conductor oscillating out-of-plane. Also important is ensuring that the conductor is not under a constant load from the shaker coupling by ensuring that the shaker is connected at a neutral position before oscillations begin. During the investigations carried out by Canadian Electrical Association (1988) the shaker was placed at mid-span and at 1.52m away from an end. The mid-span position was chosen in order to achieve a symmetric input which allowed simplified interpretations of the data and faster damping of transients (This is impossible for this current research due to physical restraints of the shaker height). This position can only excite an odd number of nodes, leading to their use of the 1.52m shaker position. [11] For the purposes of this research, the position of the shaker placement at the V.R.T.C. was dictated by the modelling process (see chapter 4).

2.3.3 Finite Element Methods

According to Rao (2011) one of the more popular methods for modelling a conductor is the finite element method. In this technique the system under investigation is discretised into a series of elements of finite length, with each element capable of acting independently. This generates a system of equations with multiple degrees of freedom. This system of equations is in the form of the Lagrangian equations of motion

$$[M]\ddot{Q} + [C]\dot{Q} + [K]Q = F \quad (2.7)$$

With $[M]$ the mass matrix, $[C]$ the damping matrix and $[K]$ the stiffness matrix, F the external load vector representing time varying forces applied to the system. Q is the vector of degrees of freedom, with each element of the vector representing a single degree of freedom. [12]

Tibert (1999) suggested that when modelling a conductor a useful approach is to represent the conductor using a pre-existing finite element type such as bar, parabolic or catenary elements with suitable modifying conditions for bending stiffness, damping and tension. Tibert details the usage of these various basic element types and what approximations are valid for certain loading conditions. These elements were tested under static conditions and not used in time-varying situations such as found in this research. [13] As noted by Wei (1999), a single catenary element can be used to model the static deflection of a conductor cable but is of less use in this research for modelling the dynamic behaviour of a vibrating conductor. [14]

According to Castello & Matt (2011), using an analytic equation based on the homogenous Euler-Bernoulli beam with constant bending stiffness and external applied tension to model a conductor, strong results were achieved using a validation based approach to calibrate the model against a standard shaker-conductor experimental set-up as per the IEEE recommended method for conductor shaker experiments. The equations of motion for the modelled conductor were discretised into a finite element basis. The results were used to calibrate and identify the air and internal damping of the conductor and to identify the bending stiffness of the conductors under investigation at 2 different tensions by using the shaker force input. [15]

According to Gatulli et al. (2004) the non-linear modelling of a cable using analytic and finite element methods was investigated. It was focussed on comparing results between the 2 different methods and the ability of both to capture the complex behaviour of the cables limited to stable branches of oscillation under harmonic loading. It is one of the few papers to consider the out-of-plane response of the cable due to harmonic loading, but this was not vital to this research project. The analytical model was based on a parabola given the assumption of a small sag-to-span ratio with the equations of motion being developed and then discretised for solving. The finite element equations were based on a 3-node isoparametric element and assumed linear elastic behaviour. The agreement between the analytic and FEM eigenvalues was good for the first few modes of vibration, both in and out of plane. However, no comparison to physical testing was made. This investigation showed that FEM can capture the behaviour of conductors in comparison with an analytic approach. [16]

2.3.4 Damping

In any dynamic system a certain amount of energy is lost to the environment through actions such as internal friction, air-damping, heat and noise etc. according to Balachandran and Magrab (2008). These forms of damping can be classified into the four most commonly used damping approximations:

1. Viscous
2. Dry Friction
3. Material/Hysteretic
4. Fluid

These damping methods all share the trait of being defined as a function of the velocity of the system. [17]

For a multiple degree of freedom system the most common choice is the viscous damping model according to Sondipon (2000). [18] This linear proportional technique assumes that the

damping matrix \mathbf{C} is a linear product of the stiffness and mass matrices \mathbf{K} and \mathbf{M} respectively, such that

$$[\mathbf{C}] = a[\mathbf{M}] + b[\mathbf{K}] \quad (2.8)$$

This is known as Rayleigh or Classical damping. [19] Whilst other linear methods of expressing damping coefficient exist, a more detailed study of damping methods is beyond the scope of this current research.

2.3.5 Stiffness

The stiffness of an overhead conductor is a function of a number of parameters. The conductors under investigation were of the Aluminium Conductor Steel Reinforced type, which consist of a bundle of helically wrapped individual strands wrapped around an individual or multiple steel strands as a centre reinforcing element. Figure 2.5 shows a cross section and side view of a typical conductor section. R_1 is the radius of the centre stand, with R_i the radius of a strand in a layer i , r_i the distance from the conductor centre to the centre of a strand in a layer i , F the sum of the forces applied to the conductor in the axial direction and λ the angle of wrap of a strand in the individual layer.

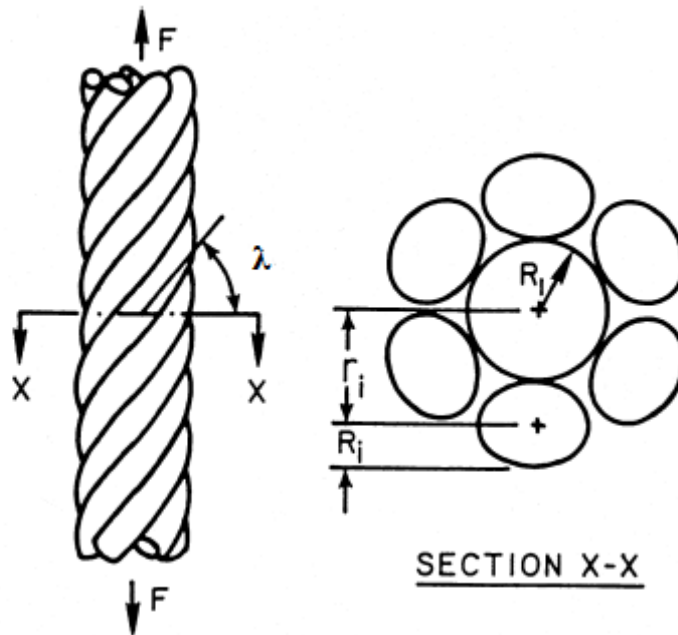


Figure 2.5: Side View and Cross Section of Typical Conductor Section (Tibert 1999)

The stiffness of a conductor affects the motion and steady state shape due to the axial and bending effects. The axial stiffness is a product of the stiffness of the individual strands and the external tension applied as a pre-stress. [13]

The bending effects of a helical cable are complex due to the interaction between the cable strands in a layer as well as between layers as suggested by Papailiou (1995). The bending stiffness EI of a cable in total is not simply the summation of all the individual moments of inertia of each cable strand using the parallel axis theorem, although at maximum the real bending stiffness does approach this as limiting value. According to Papailiou (1995) the bending stiffness is dependent on such factors as the friction between cables, the angle of wrap of the helical stranding and most importantly the curvature of the cable along its length. At high tension and low to minimum relative bending displacement the bending stiffness is at a maximum but as the curvature of the conductor increases the bending stiffness decreases as the individual strands slip relative to each other. [20]

Canadian Electrical Association (1988) suggested that the increase of the conductor bending stiffness due to increasing tension causes the resonant peaks as predicted by vibrating string theory to be shifted to higher frequencies and that as the frequency increases the peaks spread. [11]

2.3.6 Tension

When an overhead conductor is installed, it is installed with a pre-tension applied to it. This tension affects the behaviour of the conductor in a number of ways such as:

- altering the steady state position under gravity
- the speed of wave transmission along the conductor
- the natural frequencies of vibration
- the effects of damping within the conductor

This tension is applied based on the Ultimate Tensile Stress of the conductor in question, with values of tension seldom exceeding 30% of UTS for any conductor. The tension affects the steady state position by increasing the relative stiffness of the conductor which limits the deformation of the conductor under gravity. As noted in Ren (2008), as the tension increases in a tensioned cable, axial stiffness increases and the magnitude of cable sag decreases. [21]

Using vibrating string theory EPRI (2006) suggested that an expression can be derived that gives the natural frequencies of a vibrating string under the effect of a tension T:

$$f_n = \frac{n}{2L} \sqrt{\frac{T}{m_l}} \quad (2.9)$$

where L is the string length, n the mode of vibration and m_l the mass per unit length of the string. The term $\sqrt{\frac{T}{m_l}}$ represents the speed at which a wave travels along the string due to

flexural bending. Increasing the tension will cause this speed to increase. This will also cause the natural frequency of a given mode of vibration to increase as well.

This equation can be modified though to include the effects of bending stiffness, by representing the conductor as a tensioned beam with bending stiffness EI according to Claren and Diana (1969). [22] This leads to the expression:

$$f_n = \frac{1}{2\pi} \sqrt{\left(\frac{n\pi}{L}\right)^2 \frac{T}{m_l} \left[1 + \left(\frac{n\pi}{L}\right)^2 \frac{EI}{T}\right]} \quad (2.10)$$

The bending stiffness term EI has the effect of raising the natural frequency by a small amount, in the range of roughly 1-5% according to EPRI (2006).

Diana et al. (2000) suggested that by increasing the external tension and hence the relative stiffness, the inter-strand slip between cables is reduced which leads to a reduction in the hysteretic losses due to internal friction. This results in a decrease in the internal damping factor for a conductor. [23]

2.4 Chapter Summary

In this chapter an overview of relevant concepts was presented about the behaviour of conductors subject to the effects of aeolian vibrations. The nature of aeolian vibrations including onset and steady-state lock-in was considered as well as its effect on conductor motion. Methods of modelling and investigating conductor behaviour were presented including the Energy Balance Principle and finite element methods in order to analyse and expand on the behaviour of conductors. The principles and limitations that govern the use of shakers to excite conductors during testing was discussed.

CHAPTER 3

NUMERICAL SIMULATION

3.1 Introduction

In this chapter a numerical simulation is presented that was developed in order to investigate aspects of aeolian vibrations and various conductor-shaker interactions. The important components of the finite element method (FEM) are detailed and the methods used to solve the systems of equations are shown.

3.2 Self-Imposed Restrictions

Due to the scope and complexity of the problem, certain simplifying assumptions were necessary. The primary self-imposed restriction was to attempt to linearise the solution space as much as possible. This meant selecting and implementing a linear element model for the finite element method and discounting some of the hysteretic effects that exist within the system where possible. These effects include energy loss at the conductor ends; hysteretic frictional effects of self-damping within the conductor span and air-mass damping. However, some non-linear effects are unavoidable, such as the effect of the external tension, as the tension was applied in the external force vector of the system. The external tension as an element-wise force is dependent on the position of the element which results in a non-linearity. The results of implementing this method of tension are discussed further in Chapter 4.

The model is restricted to the two dimensional case, as the vibrations that are caused by the aeolian vibrations affect the conductor mainly in the vertical plane parallel to its length-wise axis. Cable elements that include the effects of out-of-plane motion in order to represent the motion of the pylons have been developed, but these have limited applications to this research as aeolian vibrations are of insufficient strength to effect the supporting pylons enough to be considered in this research. [24] Mittal and Kumar (2001) showed solutions of the conductor as a two dimensional cross section in a flow field at varying non-dimensionalised vortex frequencies solved using a DNS method. Figure 3.1 shows an X-Y plot of the position of the conductor in a cross-wise flow field as calculated by Mittal and Kumar (2001) at various reduced frequencies of vibration.

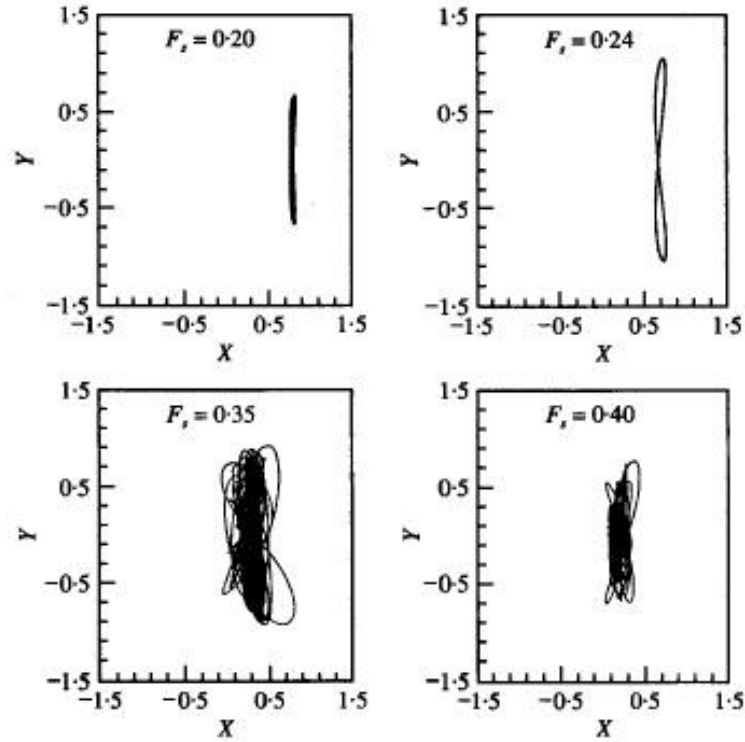


Figure 3.1: X vs Y Displacement at Various Reduced Frequencies (F_s) (Mittal and Kumar (2001))

This shows that the motion of the conductor is mostly restricted to the vertical plane, although in some cases a strong horizontal motion can occur under certain loading conditions. [7]

3.3 Conductor Finite Element Analysis

3.3.1 Applying the Finite Element Method

For this research a finite element method was used to model the conductor. The finite element method is used to discretise a complex problem of infinite degrees of freedom into one of reduced complexity and a finite number of degrees of freedom. It is a method of simplifying a problem that would be too unwieldy to solve outright by approximating it with a known system that can be solved, but in doing so it provides a solution that is an approximation. By increasing the number of degrees of freedom, the model can be made to represent the real system more closely.

Applying the finite element method to the dynamic analysis of structures results in a series of equations that can be represented in the form of the Newtonian equations of motion:

$$[M]\ddot{\underline{Q}} + [C]\dot{\underline{Q}} + [K]\underline{Q} = \underline{F} \quad (3.1)$$

The process of using the finite element method is:

- Discretise the domain into individual local elements
- Derive the mass, stiffness and damping matrix for each element
- Transform the local elements into global coordinates
- Assemble the mass, stiffness and damping matrices
- Derive the external load vector
- Solve for the unknown displacements

A restricted space-frame element based on Euler-Bernoulli beam theory was chosen to model the conductor. The space-frame element was restricted to model the axial and bending stiffness effects of a conductor, but does not consider transverse shear which was assumed to have little effect due to the nature of the conductor. The conductor was discretised into (n) number of elements.

For an element of the bar or beam type lying in the x - y plane, the axial displacement is assigned the variable u , the transverse displacement due to bending v and the slope $= dv/dx$. Figure 3.2 shows the local coordinate system.

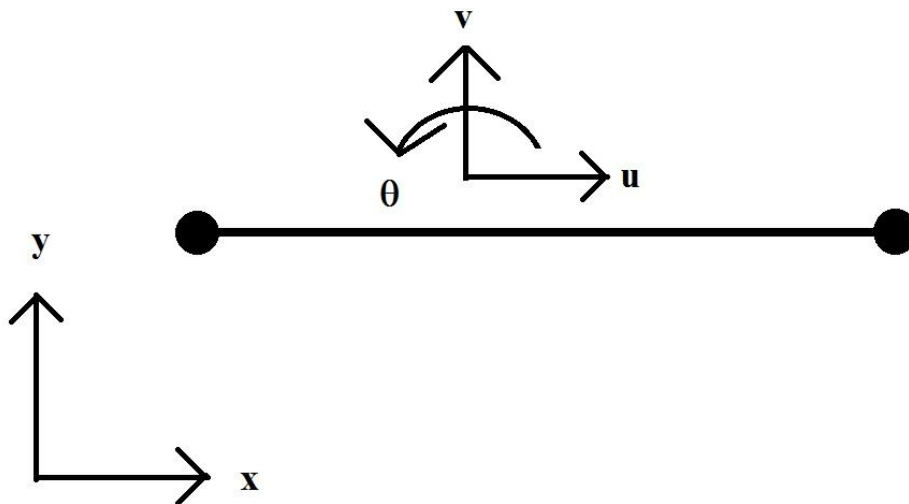


Figure 3.2: The Local Coordinate System

3.3.2 The Planar Frame Element

A space-frame element is an element based on a straight bar. It has deflection in all 3 axes at each of its end nodes and also has rotation about each of its nodes. This results in 12 degrees of freedom. It supports axial and transverse deformation, bending about its ends and torsion and shear. The requirements of the model did not include all of these degrees of freedom and because the investigation is limited to a 2-dimensional plane the element was restricted to 6 degrees of freedom. This leads to what is known as the planar frame element. The result was deflection and bending into the plane perpendicular to the centroidal axis being restricted. Torsion about the centroidal axis was also restricted. Figure 3.3 shows an element of length l lying in the x-y plane, with x parallel to the central axis. The nodal displacements q as the local degrees of freedom are also shown.

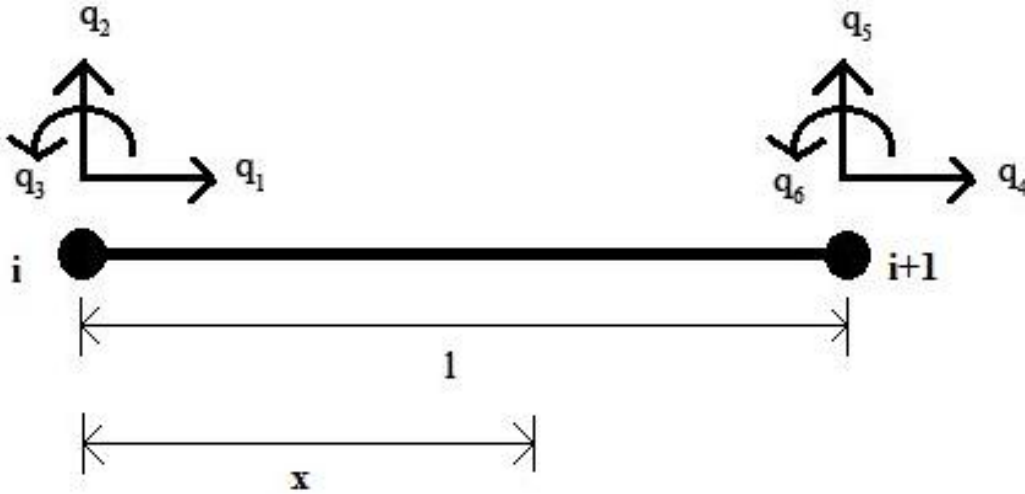


Figure 3.3: Degrees of Freedom

The element has axial displacements along q_1 and q_4 where

$$q_1 = u(x), x = 0 \text{ and } q_4 = u(x), x = l \quad (3.2) \text{ and } (3.3)$$

The transverse displacements at a node are q_2 and q_5 where

$$q_2 = v(x), x = 0 \text{ and } q_5 = v(x), x = l \quad (3.4) \text{ and } (3.5)$$

The slope at each node is represented by q_3 and q_6 where

$$q_3 = \frac{dv}{dx}(x), x = 0 \text{ and } q_6 = \frac{dv}{dx}(x), x = l \quad (3.6) \text{ and } (3.7)$$

The vector $\underline{q}_e = \{q_1 q_2 \dots q_6\}^T$. The characteristic matrices (stiffness, mass and damping) are square matrices of size 6x6.

3.3.2.1 Interpolation

The finite element method is a method of approximating the exact solution to a complex system by subdividing the problem into elements and assuming a function as a solution for each element. According to Rao (2011) these functions are known as interpolation functions. These interpolation functions relate the solution of the nodal degrees of freedom (which are only exact at the nodes themselves) to the rest of the element. As an example, the axial displacement $u(x)$ at a point x along an element of length l , is a product of the nodal variables q_1 and q_4 the interpolation function N such that

$$u(x) = [N] \begin{bmatrix} q_1 \\ q_4 \end{bmatrix} \quad (3.8)$$

$$[N] = \left[\left(1 - \frac{x}{l}\right) \frac{x}{l} \right] \quad (3.9)$$

Interpolation functions used for the planar element were polynomial, with a linear polynomial used for the axial deformation interpolation and a cubic polynomial for the bending and slope deformations. [12] See appendix B for the complete interpolation polynomials.

3.3.3 Stiffness

In the Lagrangian equations of motion, the term $[K]$ denotes the spring stiffness of the structure, based on Hooke's law, such that:

$$[K]\underline{Q} = \underline{F} \quad (3.10)$$

It contains values describing the stiffness between degrees of freedom in the element, with each value in the matrix relating the stiffness between 2 degrees of freedom, for example the value k_{ij} relates the stiffness between degrees of freedom i and j .

3.3.3.1 Local Stiffness Matrix

The planar frame element has a 6x6 stiffness matrix, where

$$[k_e] = \frac{EI_{zz}}{l^3} \begin{bmatrix} \frac{Al^2}{I_{zz}} & & & & & \\ & 12 & & & & \\ 0 & 6l & 4l^2 & & & \\ & & & \text{Symmetric} & & \\ -\frac{Al^2}{I_{zz}} & 0 & 0 & \frac{Al^2}{I_{zz}} & & \\ 0 & -12 & -6l & 0 & 12 & \\ 0 & 6l & 2l^2 & 0 & -6l & 4l^2 \end{bmatrix} \quad (3.11)$$

EI_{zz} is the bending stiffness which will be detailed further in section 3.3.3.2 for a helically wound conductor. A is the cross-sectional area (m^2) of the conductor as given in the manufacturers data sheet for each conductor and l is the element length (m). This stiffness matrix is orientated to the local coordinate system.

3.3.3.2 Bending Stiffness

An important term in the stiffness equations for the beam is the bending stiffness, EI , where E is the Young's Modulus and I is the moment of inertia. For this model, the Young's modulus is taken from the manufacturer's specification sheet, and I must be computed from the conductor geometry, based on the parallel axis theorem. According to Papailiou (1995) the use of the parallel axis theorem is valid as a base assumption. It was shown that a conductor mimics a solid beam of equal cross-sectional geometry at low curvature, due to the effects of inter-strand friction binding the strands together, resulting in the conductor behaving as a single continuous beam. Papailiou (1995) suggests that as the curvature increases the moment of inertia decreases to a minimum, due to a term known as the secondary stiffness. This secondary stiffness reduces the overall bending stiffness as curvature increases due to inter-strand slipping occurring, reducing the cohesiveness of the strands. However, these effects are not considered in this research for two reasons:

1. The secondary stiffness term related to the curvature only shows strong interaction as the curvature exceeds 10^{-2}m^{-1} . The curvature of the catenary of a conductor is not greater than 10^{-4}m^{-1} . It may increase as local waves pass along the conductor, but this leads to point 2
2. The secondary stiffness introduces hysteretic effects, which was not considered in this research.

The equations from Papailiou, 1995 are also dependant on an *a priori* knowledge of the Young's modulus for each individual strand, which is rarely feasible. The bending stiffness was calculated using a global Young's modulus. The $\cos\alpha$ term is introduced, where α is the helix angle of a layer of strand. This reduces the moment of inertia due to the effects of the angle of wrap of the strand in a layer. This angle changes from layer to layer.

Using a modified form of Papailiou's bending stiffness equations:

For a conductor with n_l number of layers and n_s strands in a given layer, the moment of inertia I is

$$I = I_0 + \sum_{n_l} I_l \quad (3.12)$$

Where I_0 is the moment of inertia of the core strand and I_l is the moment of inertia of a layer. The moment of inertia for a layer is given by

$$I_l = n_i I_i \cos \alpha + \frac{n_i}{2} (A_i r_i^2 \cos \alpha^3) \quad (3.13)$$

where I_i is the moment of inertia of a single strand about its own axis; A_i is the area of a strand in the layer; r_i is the distance from the centre of the core to the centre of a strand in the layer. Figure 3.4 shows a cross section of a typical conductor.

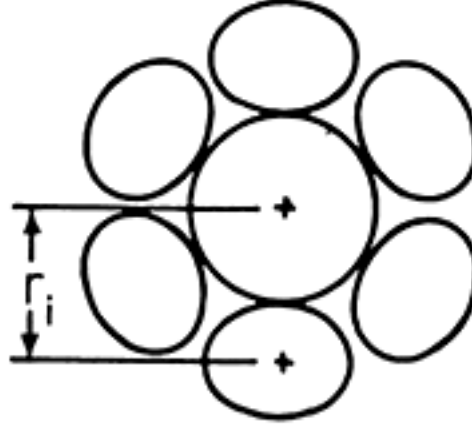


Figure 3.4: Conductor Cross Section (Tibert (1999))

This method resulted in values for the moment of inertia that is the upper end of the potential range.

3.3.4 Mass Matrix

The mass matrix was derived as being a consistent matrix as opposed to a lumped mass matrix. A consistent matrix is one in which the mass is distributed in the element using the same interpolation method as the stiffness matrix. A lumped mass matrix concentrates the mass at the nodes. For a planar frame element the consistent mass matrix is given as:

$$[m_e] = \rho A l \begin{bmatrix} \frac{1}{3} & & & & & \\ 0 & \frac{13}{35} & & & & \\ 0 & \frac{11l}{210} & \frac{l^2}{105} & & & \\ \frac{1}{6} & 0 & 0 & \frac{1}{3} & & \\ 0 & \frac{9}{70} & \frac{13l}{420} & 0 & \frac{13}{35} & \\ 0 & \frac{-13l}{420} & \frac{-l^2}{140} & 0 & \frac{11l}{210} & \frac{l^2}{105} \end{bmatrix} \quad \text{Symmetric} \quad (3.14)$$

Where ρ is the density of the element (kg/m^3), A is the cross sectional area (m^2) and l is the length of the element.

3.3.5 Damping

When simulating a dynamic structure such as this an important consideration is defining the damping characteristics and implementing them in the equations of motion. This presented a challenge as the self-damping behaviour of a conductor is the one of the hardest parameters to define and treat analytically. A number of different methods exist for defining and implementing the damping of a system into the equations of motion, all with limited scope and application as noted in Chapter 2. For this reason the Rayleigh method of damping was chosen as it is the most common and robust method of implementing damping within large systems of equations. Literature data on the Rayleigh coefficients was also available and was used in this research.

3.3.5.1 Rayleigh Damping

Rayleigh damping is a common method of representing damping in continuous systems when modelled by the finite element method. Rayleigh damping has the benefit of decoupling the equations by using modal superposition. [25] Rayleigh Damping is based on viscous damping models and assumes that the damping matrix is a product of the mass and stiffness matrix according to Chowdhury and Dasgupta (2003). It is defined as:

$$[C] = a[M] + b[K] \quad (3.15)$$

Where a and b are pre-determined constants. These constants determine the relative participation of the mass and stiffness matrices in forming the damping matrix. [26] Barbieri et al. (2004) investigated the damping of a conductor and determined the values of a and b for a conductor of type “Ibis” for various lengths and applied tensions. Figures 3.5 and 3.6 show these results.

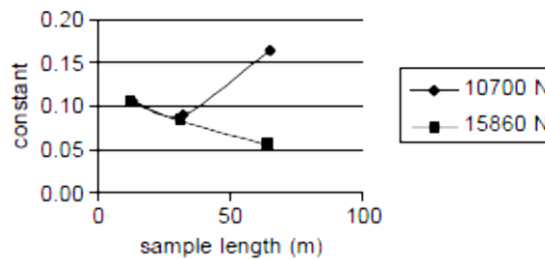


Figure 3.5: Values for “a” Constant (Barbieri et al. (2004))

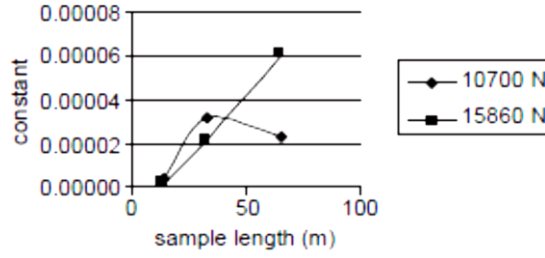


Figure 3.6: Values for "b" Constant (Barbieri et al. (2004))

The damping coefficients of the 65m length were used in this model. Conductors under test with a tension of 12000N and below were modelled using coefficients of $a=0.15$ and $b=0.00002$. For a tension of 12000N and above, values of $a=0.05$ and $b=0.00006$ were used. [27] The effects of varying the damping factors and how important they are in the final simulation are discussed in Chapter 4.

3.3.6 External Force Vector

3.3.6.1 Tension

A conductor strung between two pylons is set to a specific tension T when first installed. This tension was treated as an external force for this model and as such is part of the external force vector \underline{F} .

The tension is a force that works to restore the position of the conductor to an energetic minimum i.e. to a position of least deflection from the line of shortest distance. The tension is applied at the ends of the conductor and acts along the axial component of the conductor. In the absence of gravity and any other forces, the tension would balance between the elements and have zero net result. As the conductor is displaced from the minimum position due to gravity the tension propagates along the axial component with the resultant force developing at the node between two elements. Figure 3.7 shows the interaction between two elements j and $j+1$ and the tension T .

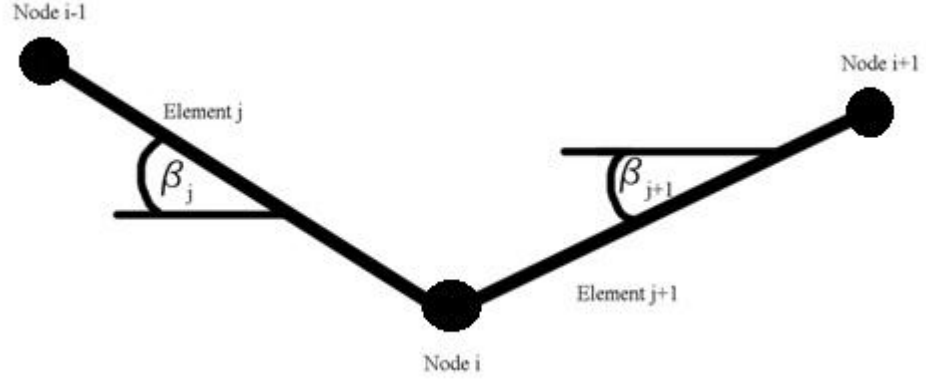


Figure 3.7: Tension as applied to elements

Expressing the resultant forces decomposed into their x and y components at the i^{th} node in the global x and y directions:

$$F_{xi} = T \cos \beta_{j+1} - T \cos \beta_j \quad (3.16)$$

$$F_{yi} = T \sin \beta_{j+1} - T \sin \beta_j \quad (3.17)$$

Where F_{xi} and F_{yi} are the resultant components, β_{j+1} and β_j are the translation angles of the $j+1$ and j elements in the global coordinates respectively. The resultant force components were merged into the global force vector along with the gravity component.

3.3.6.2 Gravity

The effect of gravity on the conductor was incorporated into the external force vector as a distributed load. The force is applied at each node in the $-Y$ global direction. At each node the force in the $-Y$ direction is

$$F_e(Y) = -\frac{\rho L g}{(n-1)} \quad (3.18)$$

Where ρ is the linear density of the conductor in kg/m^3 , L is the total length of the conductor in m, g is the acceleration due to gravity (9.81m/s^2) and n is the number of elements used in the model.

3.3.6.3 Distributed Wind Load

The wind force is a distributed load across the length of the conductor acting at each node in the Y global direction. For a linearised force input the force at a node is

$$F_e(Y) = \frac{1}{2} \frac{C_y \rho D L U_{\infty}^2}{(n-1)} \quad (3.19)$$

with

$$C_y = C_L \sin(2\pi f_s t + \varphi) \quad (2.5)$$

$$\tan \varphi = 2\zeta \quad (2.4)$$

The linearised wind force input (eq 2.5) includes a term for the viscous damping coefficient ζ . This is a single damping coefficient, which did not reconcile with the use of Rayleigh damping which used a damping factor for each mode of vibration. But as can be seen in figure 3.8 and 3.9, the viscous damping factor is associated with the first mode which is the mode most likely to be excited by a distributed force input. This lead to the viscous damping factor of $\zeta = 0.0045$ for the high tension models being used and a $\zeta = 0.0135$ for the low tension models being used.

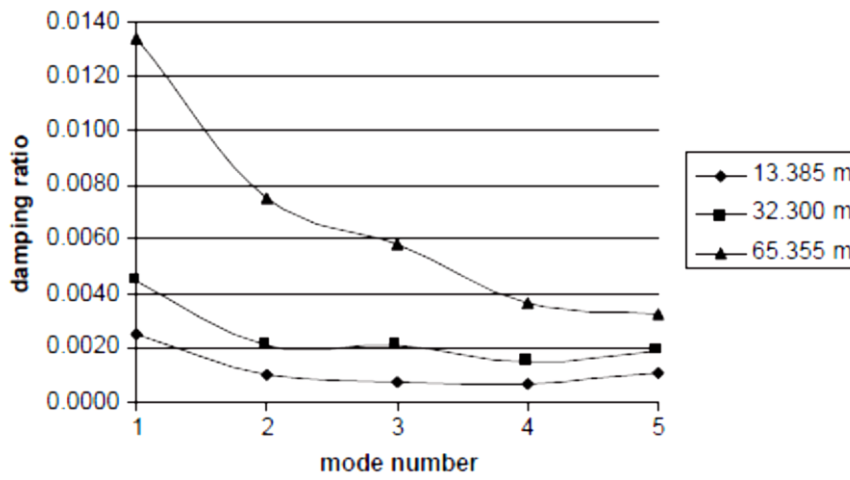


Figure 3.8: Damping vs Mode Numbers for 10700N (Barbieri et al. (2004))

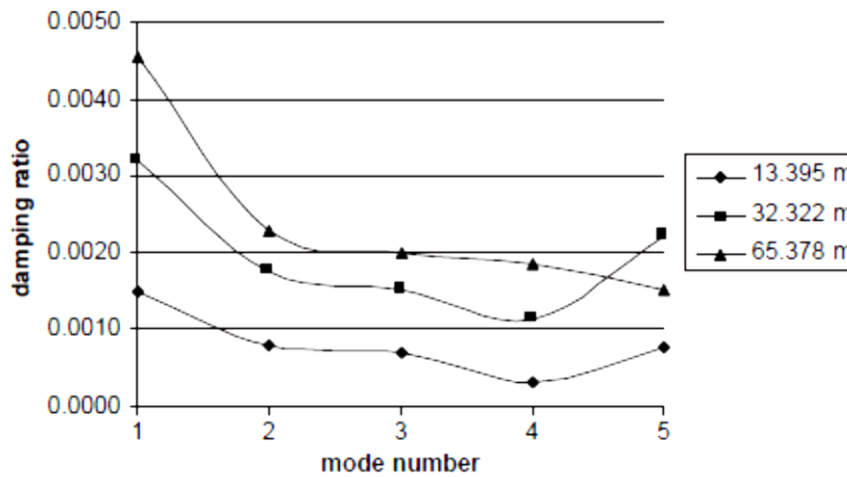


Figure 3.9: Damping vs Mode Numbers for 15860N (Barbieri et al. (2004))

The values for the reduced frequency f_s and C_L where taken from Mittal and Kumar (2001) at a Reynolds number Re of 1000.

3.3.7 Boundary Conditions

Without the proper application of the boundary conditions the model will return results that are rigid body modes, i.e. the structure moves freely in space instead of deforming. For a structural analysis the boundary conditions are nodes where the displacement is specified, either constrained to zero displacement or a displacement of known real value. The boundary conditions for the model were taken from the experimental setup of the V.R.T.C. The end clamps that hold down the ends of the conductor are attached to large concrete blocks sunk into the floor. These prevent translational motion of the conductor ends in any direction. The clamps also constrain the ends of the conductor against rotating at the clamp face due to the non-zero bending stiffness of the conductor. Figure 3.10 shows the clamping terminator mounted onto the concrete base.

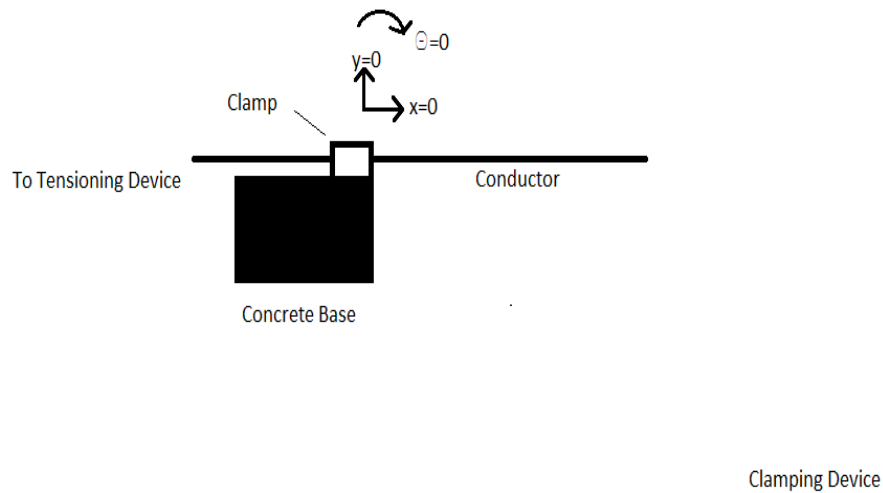


Figure 3.10: Conductor Clamping Mount

Within the global coordinate system the values of the first and last nodes respectively are constrained to:

$$Q_1 = 0 \quad (3.20)$$

$$Q_2 = 0 \quad (3.21)$$

$$Q_3 = 0 \quad (3.22)$$

$$Q_{3n-2} = L \quad (3.23)$$

$$Q_{3n-1} = 0 \quad (3.24)$$

$$Q_{3n} = 0 \quad (3.25)$$

When the shaker was connected in the model the shaker base was treated as a boundary condition of known displacement, with the value of that displacement varying in time as a function of the shaker drive equations.

3.3.8 Number of elements

The finite element method results in a series of coupled equations, with the number of equations dependent on the degrees of freedom per node, the number of nodes per element and number of elements used. A 3-degree- of-freedom, 2 node element would have 6 equations that have to be solved simultaneously, a 10 element simulation has 60 equations to be solved and so forth.

This is not a major difficulty in a steady state scenario, as the system simply has to be iteratively reduced to a solution, using one of the many matrix solving techniques. This allows a large amount of elements to be used. However, in a time-varying solution the time needed to solve the system compounds as the number of elements increases. The problem becomes one of balancing the desired accuracy of the solution with a system that does not require an unfeasibly long time to achieve a solution.

Another constraint in choosing the number of elements was the placement of the shaker when testing at the V.R.T.C. Due to the size of the shaker, there was a limit as to where it could be placed with regards to distance from an end clamp. The height of the end clamps cannot be substantially changed which prevented the shaker being moved too far from the ends as the conductor would hang below the point of any possible shaker attachment. This, in conjunction with the restriction that the shaker connection must coincide with a node of the F.E.M. model in order to properly replicate single point contact limited the test possible test positions. For this research, the shaker was placed 1.41m from a clamped end, which matches to the first free node of a 60 element model, or the 2nd free node of a 120 element model. There was a danger that by

placing the shaker at a node a loss of resolution may occur as the shaker is acting at what will be an anti-node when sufficiently excited to excite the related mode. This was however a restriction of the system that can only be overcome with expanding the number of elements of the model into a realm that would be untenable for all but the most powerful computing resources.

3.3.9 Transformation Matrix

The mass, stiffness and damping matrices and the associated external force vector were derived in a local coordinate system for each element. Before these elements could be assembled into the global matrices they had to be transformed into the global coordinate system using a transformation matrix. Figure 3.11 shows the relation between the local and global coordinate system.

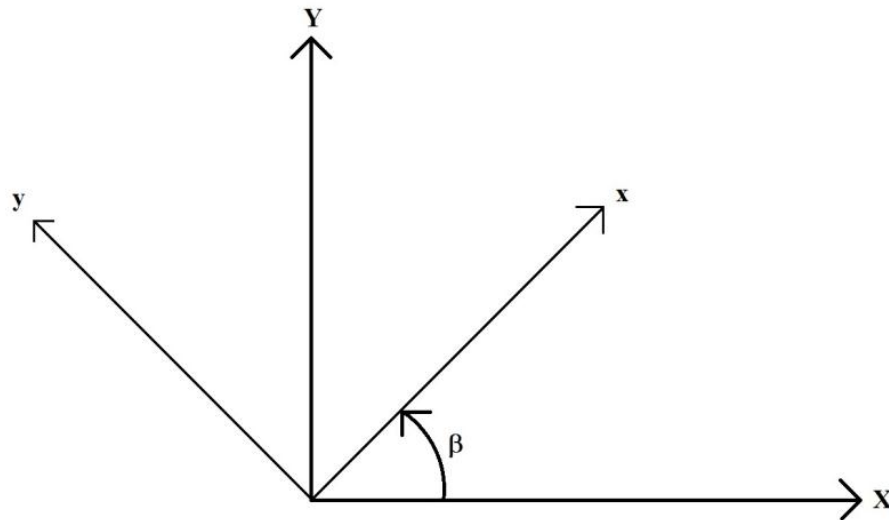


Figure 3.11: Local vs Global Coordinates

Given a local coordinate system $x-y$ orientated at an arbitrary angle β to the global coordinate system $X-Y$, there exists a transformation $[\psi]$ such that

$$\begin{bmatrix} x \\ y \end{bmatrix} = [\psi] \begin{bmatrix} X \\ Y \end{bmatrix} \quad (3.26)$$

With

$$[\psi] = \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \end{bmatrix} \quad (3.27)$$

Applying this to the element matrices, for a local stiffness matrix $[k_e]$ there exists a transformation matrix $[\eta]$ such that

$$[K_e] = [\eta]^T [k_e] [\eta] \quad (3.28)$$

where $[K_e]$ is the stiffness matrix represented in the global co-ordinate system. In a 2-dimensional space, the transformation matrix $[\eta]$ is:

$$[\eta] = \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 & & & \\ -\sin(\beta) & \cos(\beta) & 0 & & \ddots & \\ 0 & 0 & 1 & & & \\ & & & \cos(\beta) & \sin(\beta) & 0 \\ & \ddots & & -\sin(\beta) & \cos(\beta) & 0 \\ & & & 0 & 0 & 1 \end{bmatrix} \quad (3.29)$$

This transformation matrix was also applied to the mass and damping matrices:

$$[M_e] = [\eta]^T [m_e] [\eta] \quad (3.30)$$

$$[C_e] = [\eta]^T [c_e] [\eta] \quad (3.31)$$

The equations of motion for a single element can be represented as:

$$[M_e] \ddot{Q}_e + [C_e] \dot{Q}_e + [K_e] Q_e = F_e \quad (3.32)$$

where the vector Q_e is the degrees of freedom of the element e in the global coordinate system.

3.3.10 Element Assembly

The conductor was modelled as a series of elements connected in sequence. Figure 3.12 shows an example of how a conductor could be modelled using four elements connected together, with five nodes.

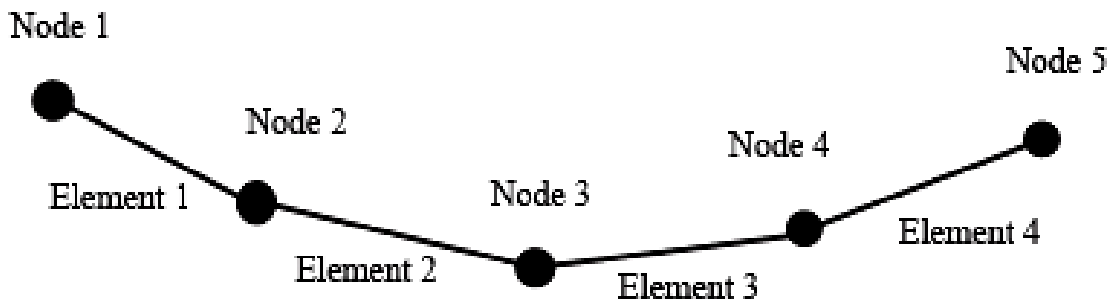


Figure 3.12: Conductor Element Assembly

After the element matrices and vectors were derived in a common global coordinate system, the individual elements had to be assembled into the final system of equations to be solved. Using the stiffness matrix as an example, the global stiffness matrix was formed by the algebraic addition of the individual element matrices after having been transformed into the global coordinates, such that

$$[K] = \sum_{e=1}^n [K_e] \quad (3.33)$$

The model is assembled element by element into the global matrix, with addition taking place at the node shared by two elements. The elements are assembled after they are transformed into the global coordinate system individually. For the assembly process, elements j and $j+1$ share node number i . Figure 3.13 shows how the element matrices are assembled.

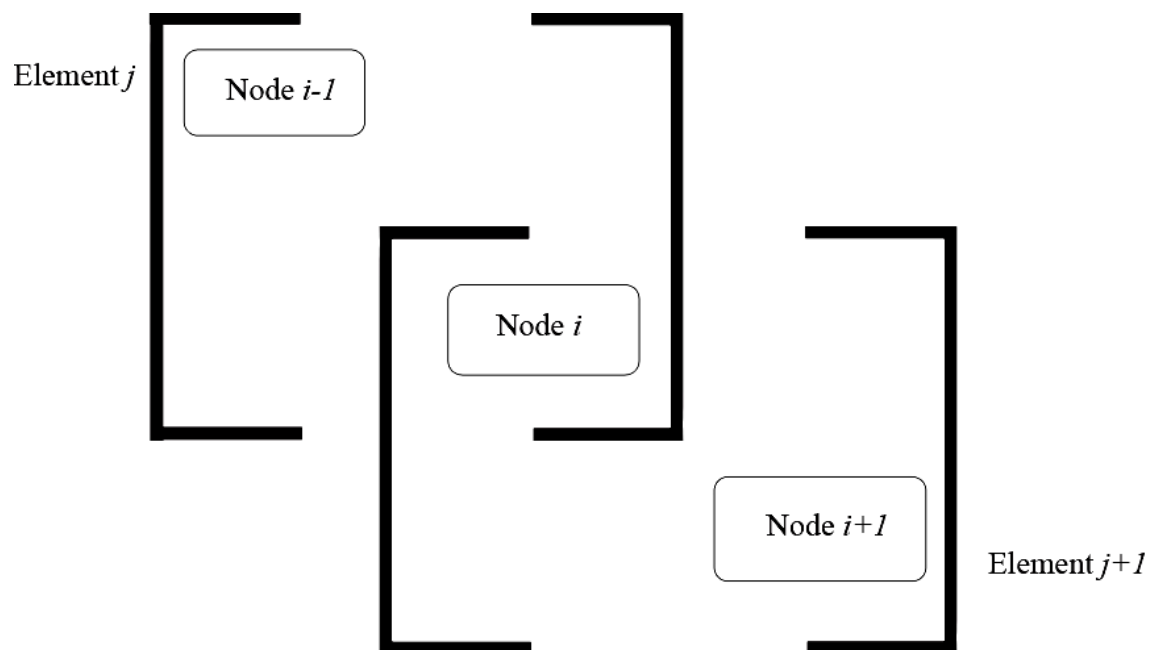


Figure 3.13: Matrix Assembly

Using element numbers 1 and 2 as an example, the partial global stiffness matrix $[K_{\text{partial}}]$ is assembled as

$$[K_{\text{partial}}] = \begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & K_{13}^{(1)} & & K_{14}^{(1)} & K_{15}^{(1)} & K_{16}^{(1)} & & & \\ K_{21}^{(1)} & K_{22}^{(1)} & K_{23}^{(1)} & & K_{24}^{(1)} & K_{25}^{(1)} & K_{26}^{(1)} & & & \\ K_{31}^{(1)} & K_{32}^{(1)} & K_{33}^{(1)} & & K_{34}^{(1)} & K_{35}^{(1)} & K_{36}^{(1)} & & & \\ K_{41}^{(1)} & K_{42}^{(1)} & K_{43}^{(1)} & K_{44}^{(1)} + K_{11}^{(2)} & K_{45}^{(1)} + K_{12}^{(2)} & K_{46}^{(1)} + K_{13}^{(2)} & K_{14}^{(2)} & K_{15}^{(2)} & K_{16}^{(2)} & \\ K_{51}^{(1)} & K_{52}^{(1)} & K_{53}^{(1)} & K_{54}^{(1)} + K_{21}^{(2)} & K_{55}^{(1)} + K_{22}^{(2)} & K_{56}^{(1)} + K_{23}^{(2)} & K_{24}^{(2)} & K_{25}^{(2)} & K_{26}^{(2)} & \\ K_{61}^{(1)} & K_{62}^{(1)} & K_{63}^{(1)} & K_{64}^{(1)} + K_{31}^{(2)} & K_{56}^{(1)} + K_{32}^{(2)} & K_{66}^{(1)} + K_{24}^{(2)} & K_{34}^{(2)} & K_{35}^{(2)} & K_{36}^{(2)} & \\ & & & K_{41}^{(2)} & K_{42}^{(2)} & K_{43}^{(2)} & K_{44}^{(2)} & K_{45}^{(2)} & K_{46}^{(2)} & \\ & & & K_{51}^{(2)} & K_{52}^{(2)} & K_{53}^{(2)} & K_{54}^{(2)} & K_{55}^{(2)} & K_{56}^{(2)} & \\ & & & K_{61}^{(2)} & K_{62}^{(2)} & K_{63}^{(2)} & K_{64}^{(2)} & K_{65}^{(2)} & K_{66}^{(2)} & \end{bmatrix} \quad (3.34)$$

This sequence was repeated for all elements 1 to n until the global matrix was populated. It was this matrix that was used in the final solving process. This algorithm was also applied to the mass matrix:

$$[M] = \sum_{e=1}^n [M_e] \quad (3.35)$$

The external force vector \underline{F} is given by the summation of the individual force components:

$$\underline{F} = \sum_{e=1}^n \underline{F}_e \quad (3.36)$$

Using elements 1 and 2 as an example again, $\underline{F}_{\text{partial}}$ is given as

$$\underline{F}_{\text{partial}} = \begin{bmatrix} F_1^{(1)} \\ F_2^{(1)} \\ F_3^{(1)} \\ F_4^{(1)} + F_1^{(2)} \\ F_5^{(1)} + F_2^{(2)} \\ F_6^{(1)} + F_2^{(2)} \\ F_4^{(2)} \\ F_5^{(2)} \\ F_6^{(2)} \end{bmatrix} \quad (3.37)$$

The external force effects that were derived in the global co-ordinate system, such as gravity and wind force, could be included directly into the global force vector.

3.4. The External Shaker

The primary method of testing overhead conductors at the V.R.T.C. is to apply a sinusoidal force input to the system using an electrodynamic shaker. The shaker that the V.R.T.C. uses is the TIRAvib 56263/LS. It has a frequency range of 5-3000Hz with a maximum displacement of 50.8mm peak to peak. It is used in conjunction with and controlled by the PUMA vibration analysis system which serves as the frequency generator and actively controls the displacement of the shaker base during tests. Figure 3.14 shows the shaker coupled to the Tern conductor.

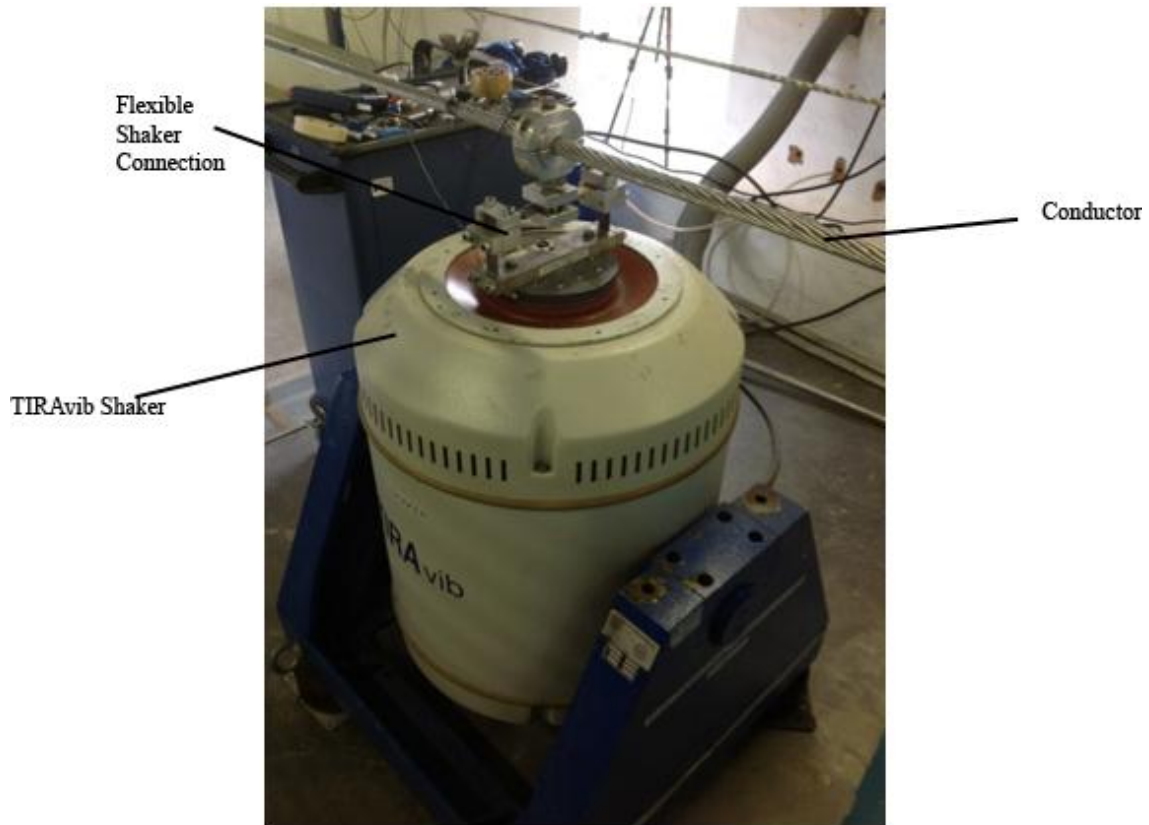


Figure 3.14: The TIRAvib shaker connected to "Tern"

The TIRAvib shaker used for the testing the conductor was used to apply a sinusoidal force to the conductor with a flexible coupling. This coupling is designed to apply a force to the conductor, but theoretically should not noticeably interfere with the behaviour of the conductor as it vibrates i.e. the conductor should dominate the motion of the coupling when sufficiently excited. This however relies on the coupling connection having a mass low enough compared to the conductor and a spring stiffness low enough to yield under noticeable conductor motion. The IEEE Std 563-1978 notes that the mass of the vibration generator should be less than 10% of the conductor mass or otherwise it must be included in the determination of the input power. But the use of the flexible connection effectively isolates the mass of the armature from the conductor. Figure 3.15 shows the flexible coupling.



Figure 3.15: Flexible Coupling Connected to “Tern”

The shaker and coupling was modelled as a single beam element. It connects to a node of user choice on the cable, with the other node being driven by a specified displacement or velocity. The driven node has a single degree of freedom in the vertical direction. The shaker element itself has no length, but it has stiffness and a point mass at the conductor. The driven node replicates the shaker base in that it is not affected by the cable movement but is constrained to follow a constant sinusoidal displacement of varying frequency. The mass of the flexible coupling was treated as a point mass applied at the common node between shaker and conductor. The mass was that of the clamp and its connection to the side clamps. It does not include the connection to the shaker base and the side rising clamps. This was measured to be 0.53kg.

The stiffness of the coupling was measured by applying a series of known weights and measuring the deflection of the clamp. Figure 3.16 shows the graph of applied load and deflection.

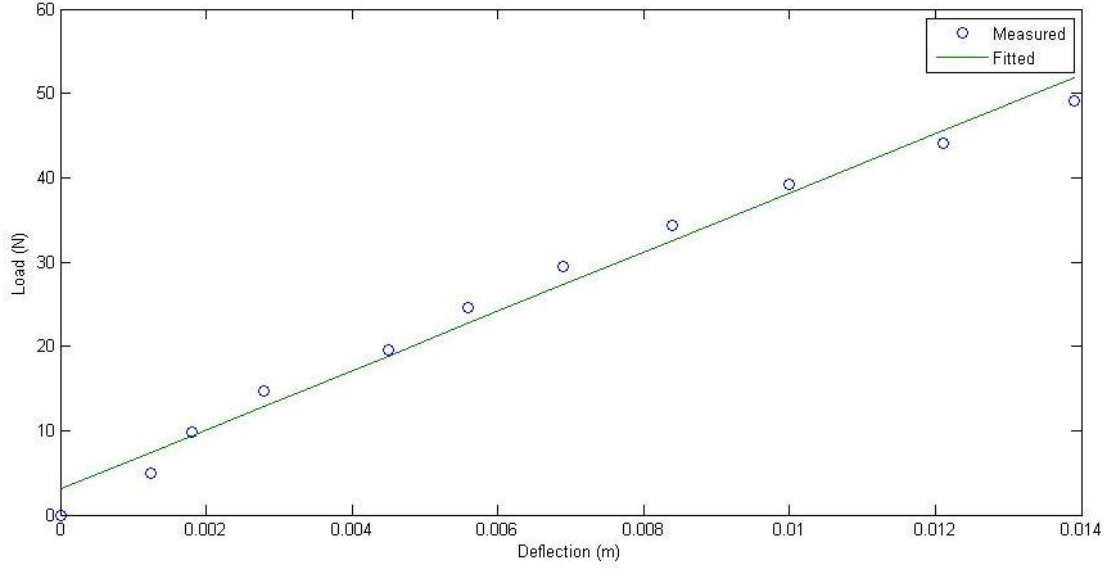


Figure 3.16: Flexible Coupling Stiffness

The data was curve fitted using least squares and a linear polynomial form to determine the best fit slope. The slope reveals the stiffness, as per Hooke's Law, to be 3514.2 N/m. This stiffness was applied between the driven and connecting node in the stiffness matrix. No damping is applied to the shaker as its level of damping is assumed to be far less than that which would have an effect on the conductor behaviour.

3.5 Initial Value

An important factor in modelling the conductor behaviour was the question of sag extensibility. According to Castello & Matt (2011) for conductors with a small sag to span ratio the effects of sag inextensibility are commonly neglected. For this research model to be effective as a simulation under a number of loading types and conductor types and lengths, the effects of sag must be accounted for. [28] It was for this reason that the first step in the simulation process was to determine the steady state position of a specific conductor under a pre-set external tension. This was achieved by setting the damping of the conductor element unfeasibly high within the simulation and releasing it from the initial position of zero sag. The simulation would then model the final position of the conductor under the effects of gravity without any other input, before the shaker was used to excite the conductor.

For the initial value, the shaker base was assigned the same value of displacement as the conductor node it was connected to because the shaker connection was modelled as having zero length. The shaker base node was given indices number "s". For a shaker connected to the 2nd node of the simulation, $Q_s = Q_5$ at time $t = 0$.

3.6 MATLAB

The computational model was created and solved in the MATLAB computing environment. This was chosen due to the flexibility of implementing a solution in MATLAB to solve the various aspects of the problem, such as steady state position, stepped vs. continuous excitations etc. as well as applying the boundary conditions at the edges and the shaker driving functions.

3.6.1 Structure of the simulation

3.6.2 ODE45 and Reduction of Order

The equations were formulated such that they could be solved using the ODE family of time varying solvers included in the MATLAB package. The particular solver chosen for this research was the ODE45 method. This uses the Runge-Kutta 45 method to iteratively solve the system of equations over time, using a variable time step. ODE23tb and other such methods proved to be unstable in terms of the time taken to produce a solution. This led to ODE45 being used as the default method of solving the system of equations.

A limitation of the ODE family of solvers is that they can only solve equations of one degree of order i.e. $\dot{x} = f(x)$. The equations of motion are second order and so needed to be modified before they could be solved by ODE45. This was done by a technique known as reduction of order.

Using the method suggested by Bucknell University, for a system of (m) equations of order (n), the equations can be reduced to a series of (m x n) first order equations. This is done by changing the variables. Beginning with the Lagrangian equation of motion

$$a\ddot{x} + b\dot{x} + cx = 0 \quad (3.38)$$

We can define a vector $\mathbf{P} = [p_1 \ p_2]$ such that:

$$p_1 = x \quad (3.39)$$

$$p_2 = \dot{x} \quad (3.40)$$

Taking the derivative of \mathbf{P} with respect to time gives:

$$\dot{p}_1 = \dot{x} \quad (3.41)$$

$$\dot{p}_2 = \ddot{x} \quad (3.42)$$

We can redefine the equation of motion from eq 3.38:

$$\ddot{x} = -\frac{b}{a}\dot{x} - \frac{c}{a}x \quad (3.43)$$

Into the pair of first order equations:

$$\dot{p}_1 = p_2 \quad (3.44)$$

$$\dot{p}_2 = -\frac{b}{a}p_2 - \frac{c}{a}p_1 \quad (3.45)$$

This pair of equations can be easily solved iteratively by the ODE45 solver. [29]

3.6.3 Applying the Boundary Conditions

The geometric boundary conditions were applied to the model in the MATLAB code by replacing the associated stiffness values in the matrix with unity and incorporating the prescribed displacement into the external force vector.

As suggested by Rao (2011), if a degree of freedom Q_j was constrained to a prescribed value Q_j^* , the external force vector F was modified where:

$$F_i = F_i - K_{ij}Q_j^* \text{ for } i = 1, 2, \dots, M \quad (3.46)$$

$$F_j = Q_j^* \quad (3.47)$$

The rows and columns associated with the constrained degree of freedom were set to zero except the value on the diagonal which was made unity i.e.

$$K_{ji} = K_{ij} = 0 \text{ for } i = 1, 2, \dots, M \quad (3.48)$$

$$K_{jj} = 1 \quad (3.49)$$

The externally driven shaker node was set as a constrained boundary point. The shaker base displacement and velocity was defined explicitly as a function of time before the ODE45 method is called, with the shaker displacement and velocity being referenced from the pre-defined values. Setting the shaker base as a boundary condition prevented the rest of the model affecting its value which was physically comparable to the experimental system in which the PUMA control constantly monitors and controls the amplitude of the shaker base during testing to prevent system feedback from interfering with the shaker motion.

3.6.4 Shaker driving function

3.6.4.1 Constant Frequency

For the investigations at a constant frequency, the shaker base was driven by a simple sine wave driving function of constant amplitude and frequency. The simulation was started at zero shaker displacement relative to the conductor. For a constant frequency f , the shaker base displacement Q_s at a time t_{i+1} is

$$Q_s(t_{i+1}) = Q_s(t_i) + A \sin(2\pi f t_i) \quad (3.50)$$

The velocity of the shaker base \dot{Q}_s at a time t_{i+1} is given as

$$\dot{Q}_s(t_{i+1}) = \dot{Q}_s(t_i) + A2\pi f \cos(2\pi f t_i) \quad (3.51)$$

3.6.4.2 Time Varying Frequency

For the case of a shaker base driven at a frequency that varies with time i.e. $f(t)$, at any time step t_i , the position and velocity of the shaker base at that time was interpolated from the vectors containing the global displacement and velocity over the course of the sweep which had been pre-generated across the time-span before the iterative solving process began. It was necessary to pre-generate the driving variables as the function was too complicated to solve as an explicit function of time from within ODE45 due to the varying time-step. This necessitated the values of the displacement and velocity to be stored as a global variable that ODE45 interpolated from. The driving function is monotonically increasing and well behaved which lead to a quick interpolation function as opposed to a standard interpolation algorithm being used to reduce the computational cost of constantly interpolating the displacement and velocity at each time step.

The sweep tests were performed using a logarithmic function of time, sweeping from the lowest to the highest frequency. A logarithmic sweep is given as

$$f(t) = f_0 B^t \quad (3.52)$$

Where f_0 is the starting frequency and B is the gradient of the sweep. The gradient of the sweep B is given as

$$B = \left(\frac{f_{final}}{f_0} \right)^{\left(\frac{1}{t_{final}} \right)} \quad (3.53)$$

With f_{final} the end frequency of the sweep and t_{final} the time value at which the sweep ends. The sweep displacement and velocity was pre-generated as a series of steps of constant frequency to replicate the control system of the TIRAvib shaker. Figure 3.17 shows an example of a typical frequency sweep. Figure 3.18 shows a close view of the sweep showing the stepped increase.

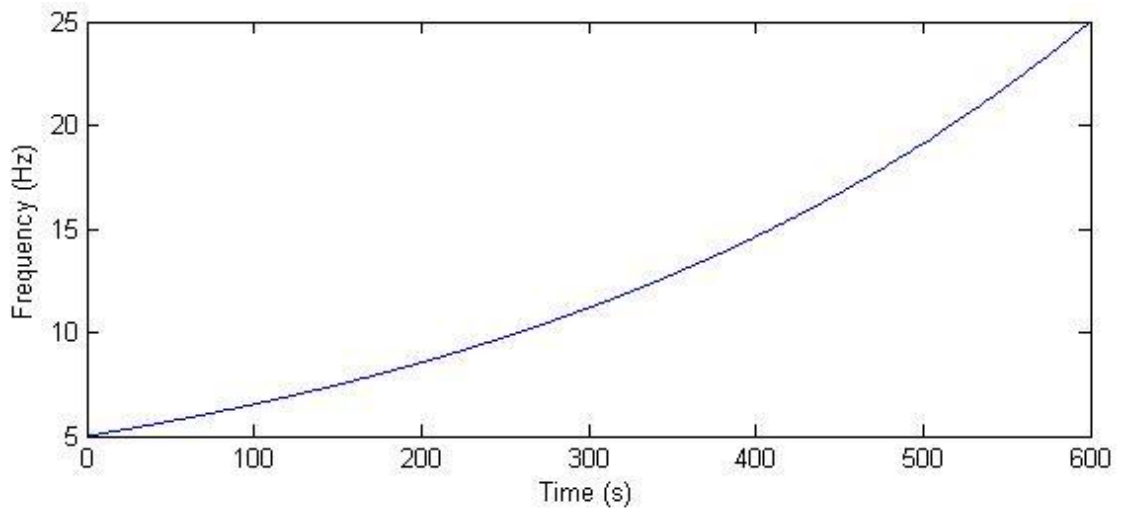


Figure 3.17: Total Frequency Sweep

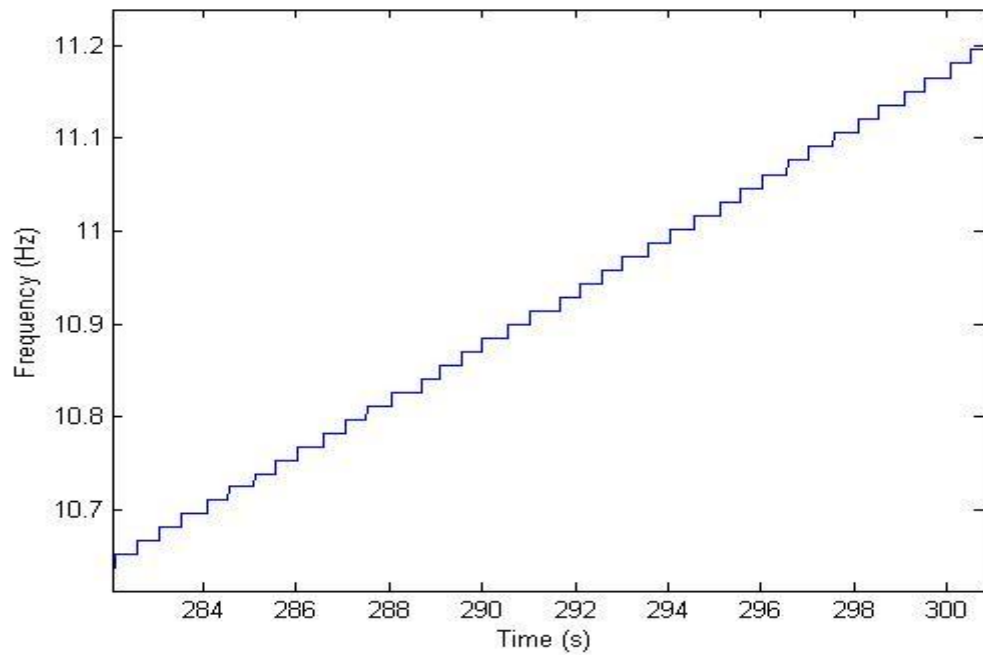


Figure 3.18: Close View Showing Steps

The frequency step increases had to be initiated when the shaker base was at zero velocity i.e. when the shaker base was at maximum displacement at the top or bottom of its travel to prevent discontinuity. The steps occurred after 0.5s intervals, with an average specified increase in frequency of 2 Hz/minute. The experimental test sweeps all began at 5 Hz as that is the lowest frequency the TIRAvib shaker can generate. The model however was not restricted by the same limitation.

3.6.4 Convergence and Time to Solve

As the model detail and number of elements increases in the simulation, the time to solve the model also increases in a non-linear relation. Figure 3.19 shows the relation between the number of elements in a model and the real time taken to solve the same scenario. The scenario in this instance was to release a conductor from a horizontal position at start time 0, under the effects of gravity, to achieve a pre-determined solution end time.

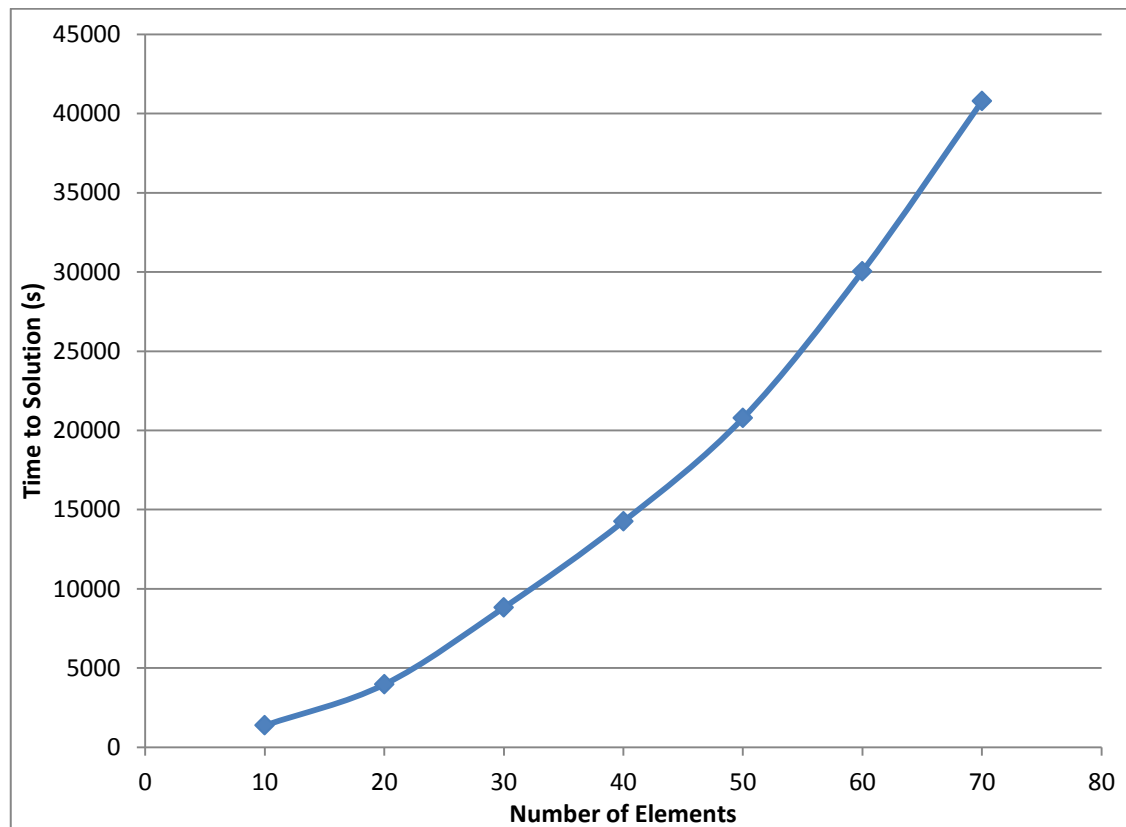


Figure 3.19: Time to Solve vs. Number of Elements

As the number of elements increases in the simulation the accuracy of the results improves. But the accuracy converges to a limiting constant as the number of elements increases. Fig 3.20 shows the accuracy of a conductor under gravity at steady state.

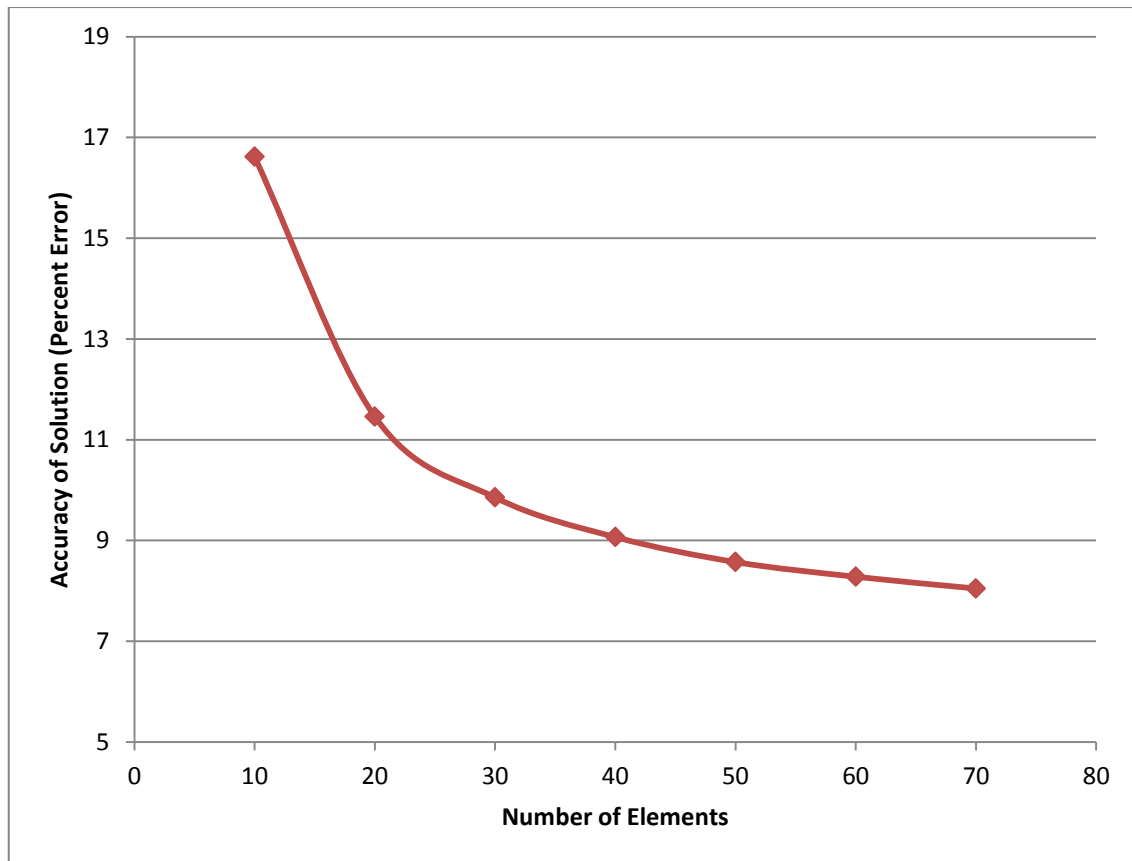


Figure 3.20: Number of Elements vs. Relative Accuracy

This means that increasing the number of elements in the simulation becomes counter-productive after a certain point, because as the time to solve increases there is an associated decrease in the gains in accuracy. The choice is a balance between the desired accuracy against a reasonable time to achieve a solution.

The final number of elements was chosen was 60 elements. This proved to have a sufficient degree of accuracy to capture the main aspects of the conductor behavior whilst still being possible to be solved in a time frame that could be considered reasonable (Approximately two days to solve a sweep lasting 10 minutes using server grade computing resources).

3.7 Chapter Summary

In this chapter the structure of the simulation was introduced and detailed. The method of modelling the conductor was introduced, with the Lagrangian equations of motion being formulated using the finite element method based on a planar frame element and the stiffness, mass and damping matrices being shown. The electrodynamic shaker and flexible coupling was modelled with the various driving functions used to drive the shaker stated. The boundary conditions were derived and the manner of applying the boundary conditions was detailed. The use of MATLAB to solve the system of equations was detailed with the technique of reduction of order shown.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter outlines the testing procedures used in the course of this research and presents the results of the experimental and simulated tests. Steady state displacements are analysed and compared to the ideal catenary for both experimental and model results. The different test methods are detailed and results presented for the experimental and model tests.

The swept and steady excitation results for the model and test conductors are analysed and discussed, with an harmonic oscillating distributed load being applied to the model and the results discussed. A comparison is made between the experimental results of the swept and steady excitations, over the different conductors and applied external tensions.

4.2 Method of Investigation

In order to investigate the effectiveness of the model of conductor behaviour and the effects of the shaker on conductor investigations, a number of separate tests had to be devised and performed on a collection of conductor types. These tests were performed on the conductors at the V.R.T.C. and the same tests were performed on the model:

- Swept Frequency Shaker Excitation: The conductor was excited using the shaker with a swept frequency spectrum
- Steady Frequency Shaker Excitation: The conductor was excited using the shaker at a constant frequency to achieve a steady state vibration
- Wind Force Excitation (Model Only): The conductor model was excited using a distributed wind force input

Before any of these tests could be performed on the model the model had to solve the steady state displacement under gravity for each conductor type and Tension setting. The steady state position was used as the initial value for the model tests. These steady state displacements were also compared to measurements taken of the conductors at VRTC.

4.2.1 Conductor Type

During the course of this research four different conductors were available for testing at the VRTC. All of the conductors were of the Aluminium Conductor Steel Reinforced (ACSR) type, in which the bulk of the conductor is made up of aluminium strands layered helically around a small number of steel strands at the core. A variety of different conductors are available of varying specifications, identified by code names which are standardised between the various conductor manufacturers. Table 4.1 shows the basic details of the conductors tested.

Table 4.1: Conductor Specifications

Conductor Name	Number of Strands (Al/Steel)	Outside Diameter (mm)	Stiffness (N/mm ²)	Density (kg/m)
Tern	45/7	27	66600	1.34
Pelican	18/1	20.7	66200	0.775
Rabbit	6/1	10.05	80400	0.214

Appendix D contains the full technical specifications of each of these conductors. The full set of tests were performed on the Pelican and Rabbit conductors, with only the sweep test having been performed on Tern due to time restrictions imposed by the availability of the VRTC and other research activities being performed.

4.2.2 Varying the Tension

Each conductor was tested at three separate tensions, from low to high tension. As per IEEE standard the tension in the conductor was measured using a load-cell in line with the tensioning load-arm. The chosen tension was set by weighting the load-arm until the desired tension was reached while the clamp that holds the conductor in place to the stationary mounting blocks was loose. Then the clamp was tightened. Table 4.2 shows the tensions for the conductors that were tested.

Table 4.2: Tension Settings per Conductor

Conductor Name	Tension 1 (15% of UTS)	Tension 2 (20% of UTS)	Tension 3 (25% of UTS)
Tern	14800N	19740N	24675N
Pelican	8120N	11200N	13070N
Rabbit	2775N	3700N	4630N

4.3 Steady State Displacement

4.3.1 Catenary equation

For any idealised string or cable strung between two points catenary theory states that under its own weight it will take the shape of a catenary. The catenary shape is described by the equation:

$$y = a \cdot \cosh\left(\frac{x}{a}\right) \quad (4.1)$$

Where a is a dimensionless scaling parameter (Teichman and Mahadevan (2003)). As an example, figure 4.1 is a catenary with a scaling number of $a = 20$. [30]

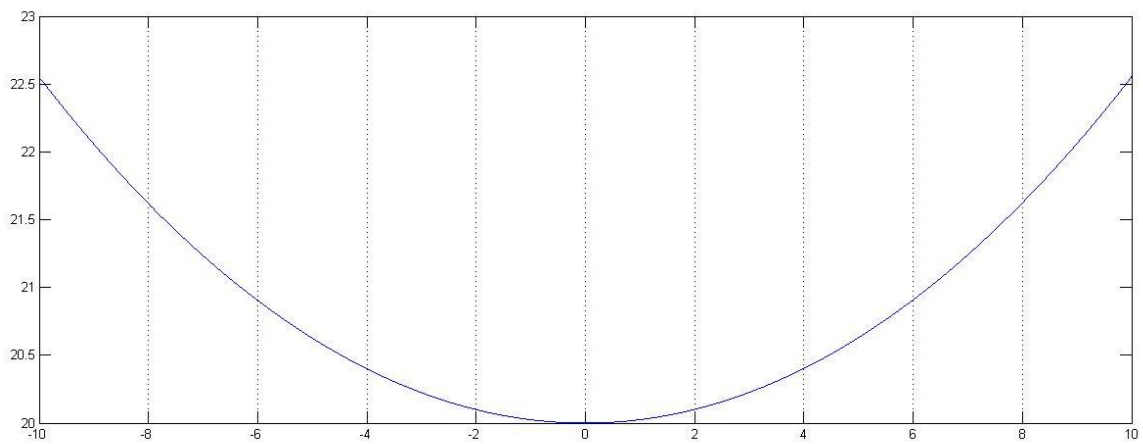


Figure 4.1: Catenary Curve, $a = 20$

Comparing the results of a static simulation under gravity to a catenary of equal end points and centre span sag allows the closeness of fit between the simulated and theoretical results of steady displacement to be determined. The catenary curve however cannot exactly represent a high tension conductor cable as such a cable will have zero gradient at the fixed ends due to non-zero flexural stiffness. This will result in a discrepancy between the simulated and theoretical results near the ends when compared to the catenary curve theory.

Measurements were taken from a clamped end up to the mid-point of the conductor and then duplicated plotting against the model results to the other side to generate a full conductor span. Table 4.3 shows the distances from a clamped end at which displacement measurements were taken.

Table 4.3: Measurement Points Along Span

Fraction of Conductor Length	1/16	1/8	1/4	3/8	1/2
Distance From End	5.2875	10.575	21.15	31.725	42.3

The measurements were biased to more samples being taken closer to the clamped end than the mid section due to the increased relative curvature near the ends.

4.3.2 Catenary Compared To Model and Measured Results

Considering the Tern conductor as a representative example, Figure 4.2 shows a comparison between the conductor model results at a steady state displacement and a pure catenary of equal midpoint displacement.

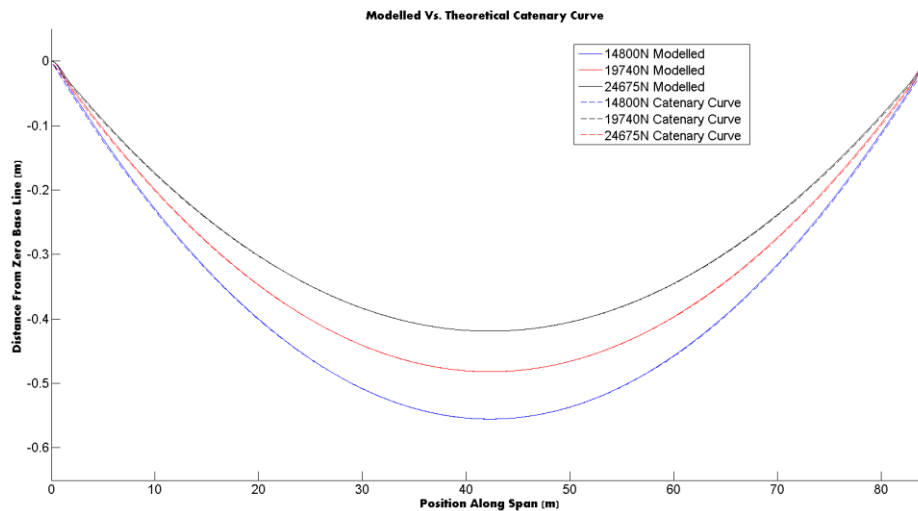


Figure 4.2: Comparison Between Model and Catenary Curve (Tern)

The model shows an agreement with the theory of the idealised hanging cable. Figure 4.3 shows a close view of an end of the comparison. It shows that the main source of discrepancy is the lack of the catenary's support for zero gradient at the ends.

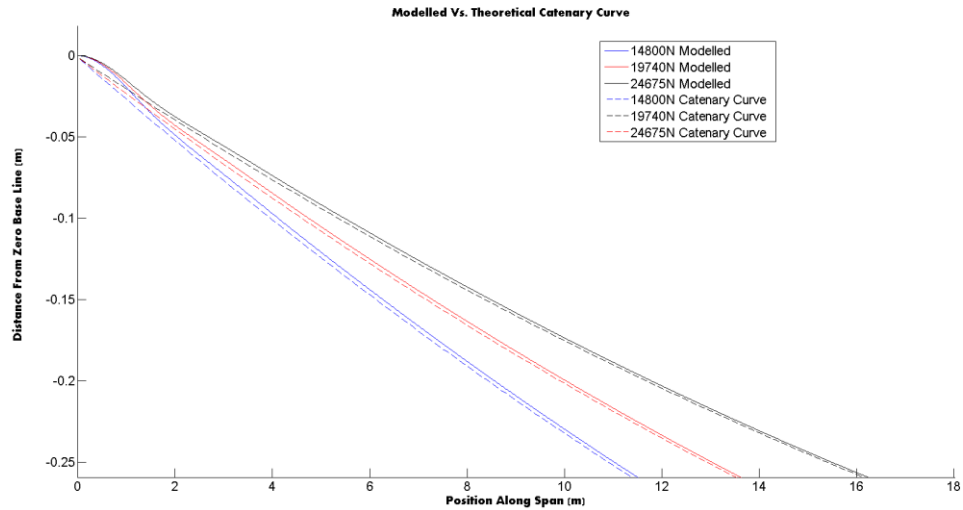


Figure 4.3: Comparison Between Model and Catenary Curve Close View (Tern)

The model results were of a similar nature for the other conductor types.

When the catenary curve was compared in the same manner (matching the midpoint displacement and end values) to the measured displacements a different picture emerged. Figure 4.4 shows a comparison between the measured experimental displacement and the catenary curve.

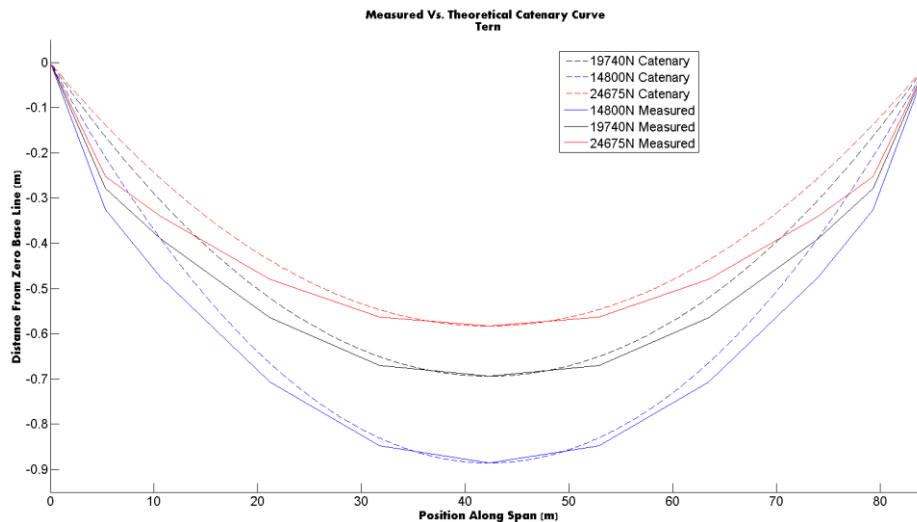


Figure 4.4: Comparison Between Measured and Catenary Curve (Tern)

The measured conductor displacement showed that the real conductor has a shallower gradient in the mid section of the conductor furthest from the ends then the catenary with a sharp increase in the gradient near the ends. These results were mirrored for all conductor types

sampled. This indicates that the behaviour of a high tension overhead conductor does not exactly follow catenary theory.

4.3.3 Model vs. Measured Results

Using the mid-span displacement as a point of comparison, the relative error of the modelled results to the measured displacement can be determined. Using the Rabbit conductor as an example, Figure 4.5 shows the three steady state results as modelled.

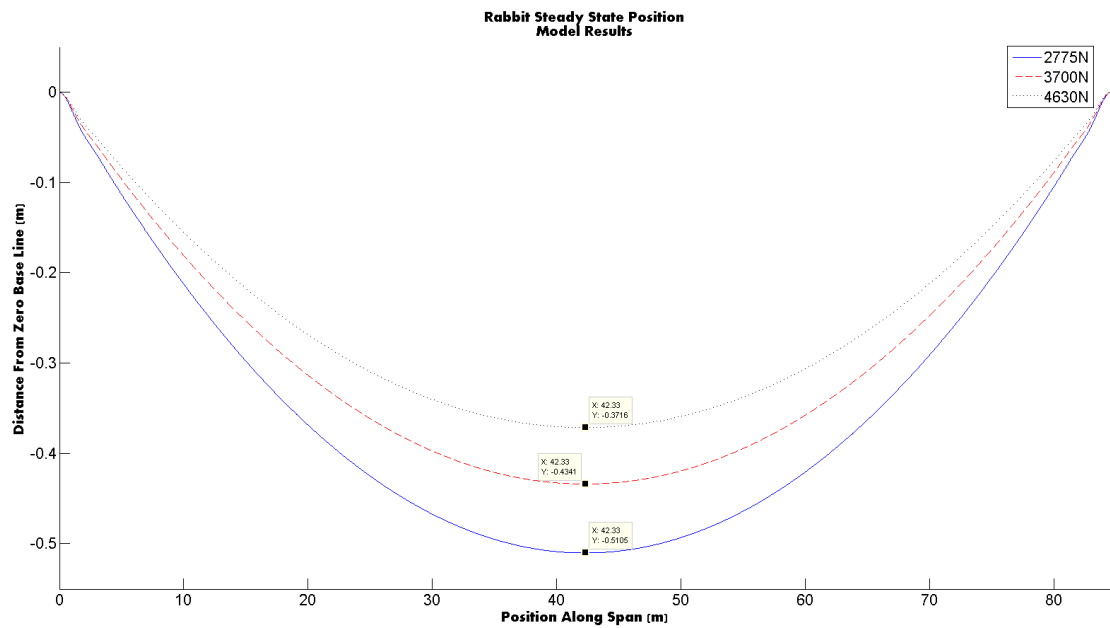


Figure 4.5: Rabbit Steady Model Results

Figure 4.6 shows the three results for the steady state displacement of the Rabbit conductor as measured at the VRTC.

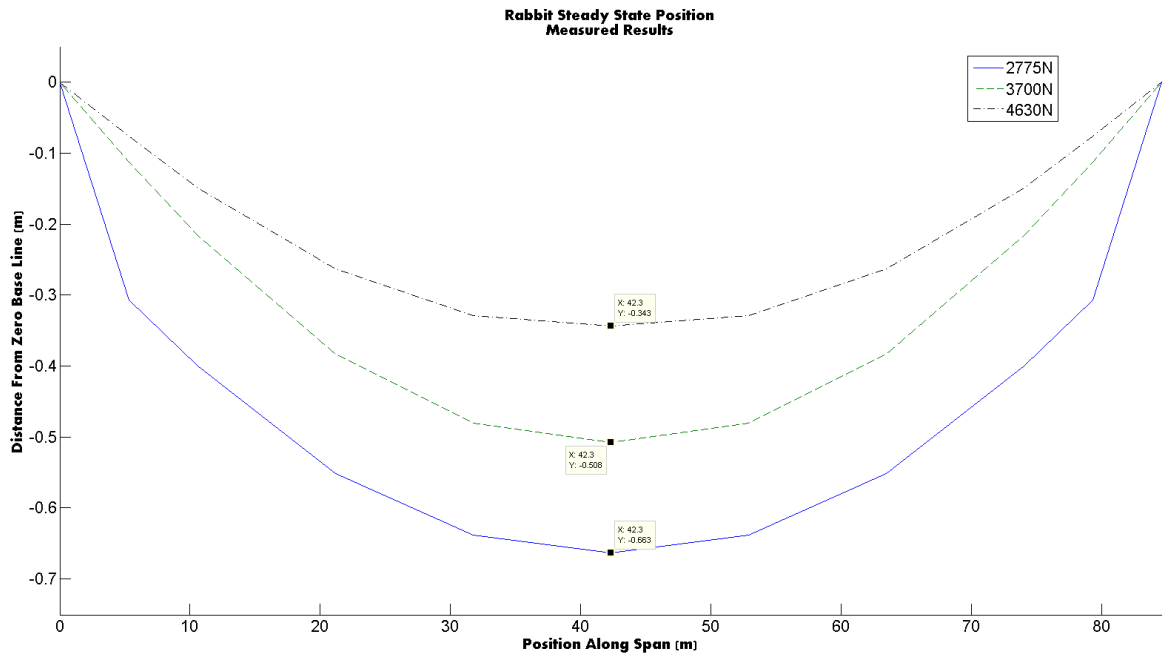


Figure 4.6: Rabbit Steady Measured Results

Tables 4.4, 4.5, 4.6 list the measured and modelled results and the relative error.

Table 4.4: Rabbit Steady State Results Comparison

<u>Tension</u>	<u>Measured</u>	<u>Modelled</u>	<u>Relative Error</u>
2775N	0.663	0.5105	23.00%
3700N	0.506	0.4341	14.21%
4630N	0.343	0.3716	8.34%

Table 4.5: Pelican Steady State Results Comparison

<u>Tension</u>	<u>Measured</u>	<u>Modelled</u>	<u>Relative Error</u>
8120N	0.552	0.5664	2.61%
11200N	0.532	0.486	8.64%
13070N	0.447	0.4433	0.82%

Table 4.6: Tern Steady State Results Comparison

<u>Tension</u>	<u>Measured</u>	<u>Modelled</u>	<u>Relative Error</u>
14800N	0.886	0.5555	37.30%
19740N	0.694	0.482	30.54%
24675N	0.583	0.4191	28.11%

A wide variation in relative accuracy exists between the measured and modelled results for the three conductor types as the tension was varied. For two of the conductor types as the tension was increased the models accuracy improved relative to the measured results. The Pelican results achieved a close agreement, whilst the Tern was not a close fit.

4.4 Vibration Measurement

The vibration behaviour of conductors was measured at the VRTC using accelerometers attached to the conductor at various sites along the conductor's length. An accelerometer was attached at the mid-span of the conductor and also at the site of the shaker connection. Another accelerometer was attached to the shaker base for the PUMA control system feedback. Figure 4.7 shows the position of the two accelerometers attached to the conductor.

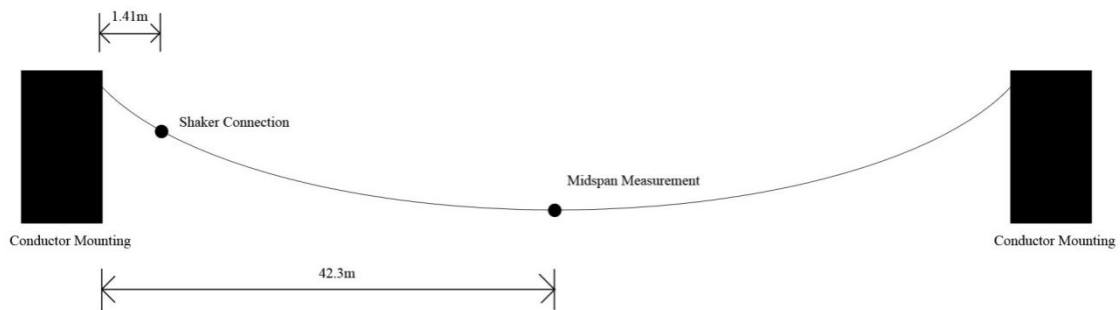


Figure 4.7: Accelerometer Positions

The data from the accelerometers was processed by the PUMA control system and integrated twice to provide the displacement data at the measurement points.

4.5 Swept Frequency Shaker Excitation

One of the primary methods of investigating the dynamic behaviour of conductors at the VRTC is to excite it at a single point connection using an external shaker with a swept frequency range. The vibration results are measured using accelerometers placed at certain points along the span, with the most commonly used being an accelerometer at the midpoint of the span and an accelerometer 89mm from an end to determine the stress at the terminator. For this research the motion at the midpoint was measured and used for comparison between the model and experimental results.

4.5.1 Frequency Sweep Data Sampling

The PUMA control and data analysis system was used to perform the frequency sweep experiments at the VRTC. It controls the frequency and amplitude of the driving shaker as measured by an accelerometer attached to the shaker base. The PUMA system can drive the

shaker in a number of different frequency ramp modes but for this research the shaker was driven using a logarithmic sweep as detailed in Chapter 3.

During a sweep test the PUMA system does not record and display the pure displacement data but instead returns the maximum measured displacement at each individual frequency step throughout the sweep range. The results obtained from the simulation were filtered to match the sampling methods of the PUMA system for the sake of comparison.

4.5.2 Rabbit Sweep Results

Figures 4.8, 4.9 and 4.10 compare the modelled results with that of the measured experimental results for the Rabbit conductor over a frequency range of 5-25Hz using a logarithmic sweep.

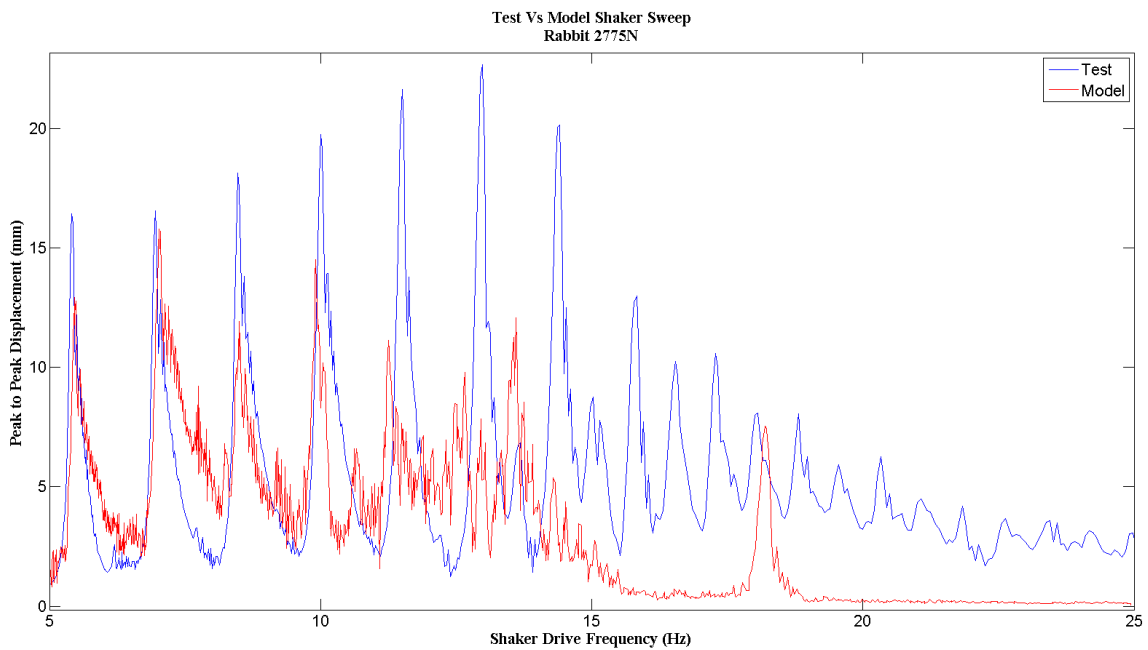


Figure 4.8: Rabbit 2775N Sweep, Model Vs Test

For Rabbit 2775N, the first four peaks of the model results track well with the test results as the peak resonances occur at similar frequencies. The displacement magnitude is of a similar scale to that of the experimental result but not as accurate. The profile of the rise to peak magnitude and roll off for each individual peak of the model results agree reasonably with the experimental results for the lower frequency peaks.

As the frequency increases the accuracy of the model results decreases before the model displacement results drops to a near minimum constant, apart from an out of place harmonic at 18Hz.

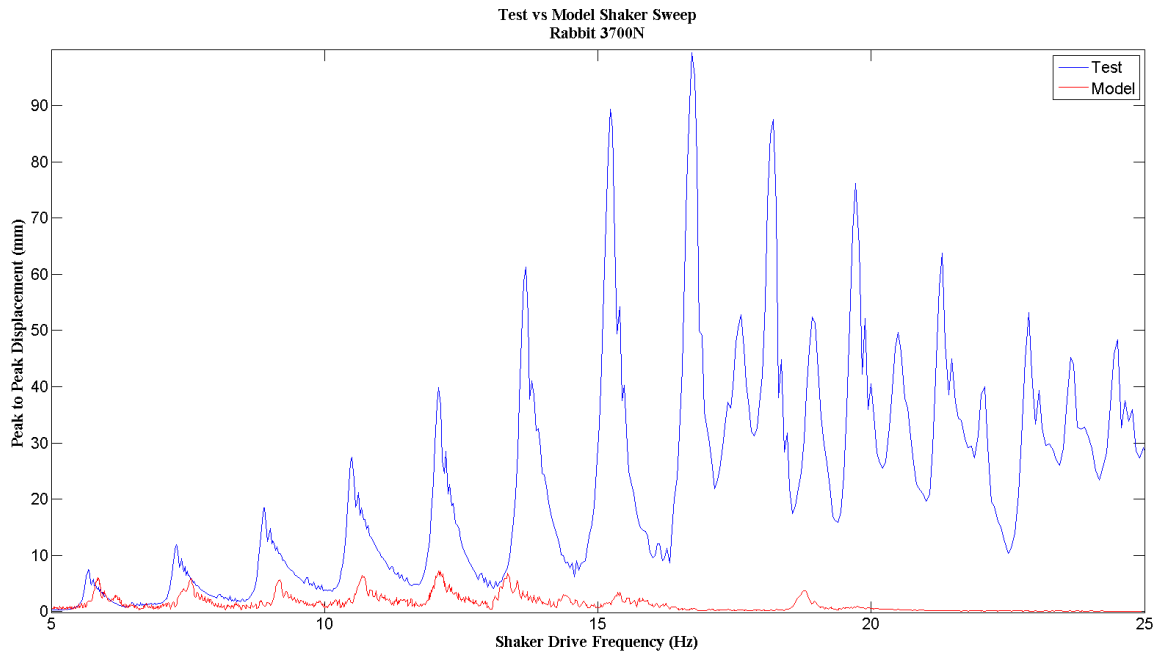


Figure 4.9: Rabbit 3700N Sweep, Model Vs Test

For Rabbit 3700N, the magnitude of the model results are substantially less than the experimental results. The first five resonance peaks of the model match the test result well with regard to the frequency at which they occur, as well as the rise and roll off for each peak.

Like the other Rabbit results, the model results decay to a near constant minimum magnitude after approximately 16 Hz.

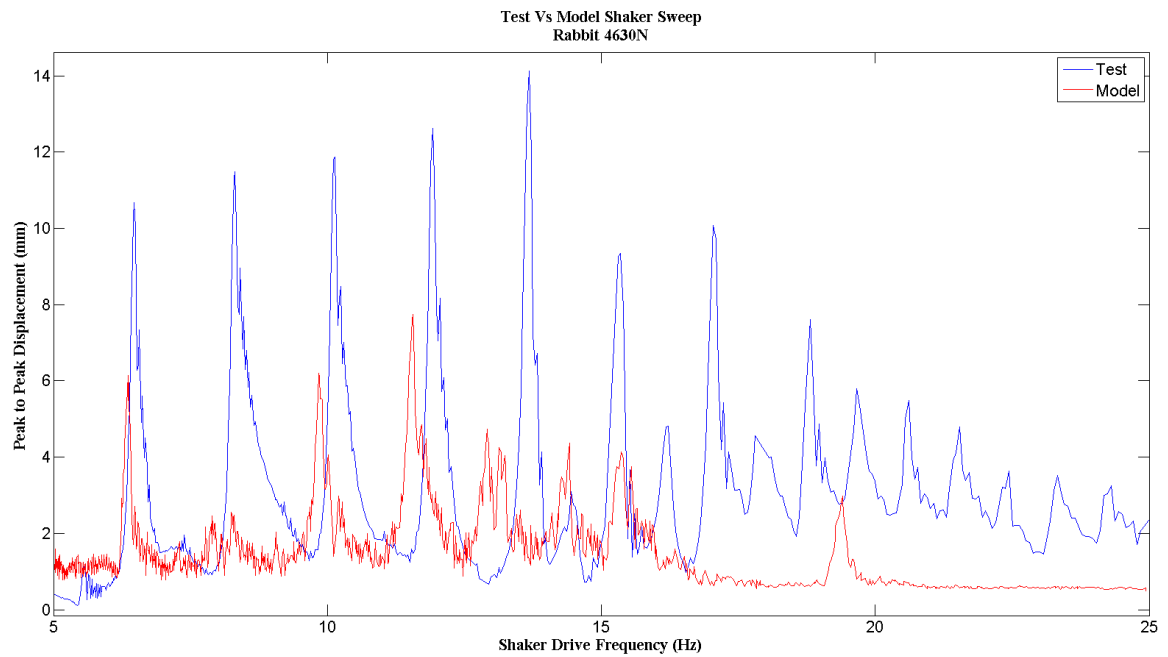


Figure 4.10: Rabbit 4630N Sweep, Model Vs Test

For Rabbit 4630N, the results are the poorest agreement of the three tension settings, with only the first peak being close to the experimental results with respect to the frequency at which occurs. The magnitude of the model results are significantly lower than the experimental results with the model decaying to a near constant minimum magnitude after approximately 16Hz.

4.5.3 Pelican Sweep Results

Figures 4.11, 4.12 and 4.13 compares the modelled results with that of the measured experimental results for the Pelican conductor over a frequency range of 5-25Hz using a logarithmic sweep.

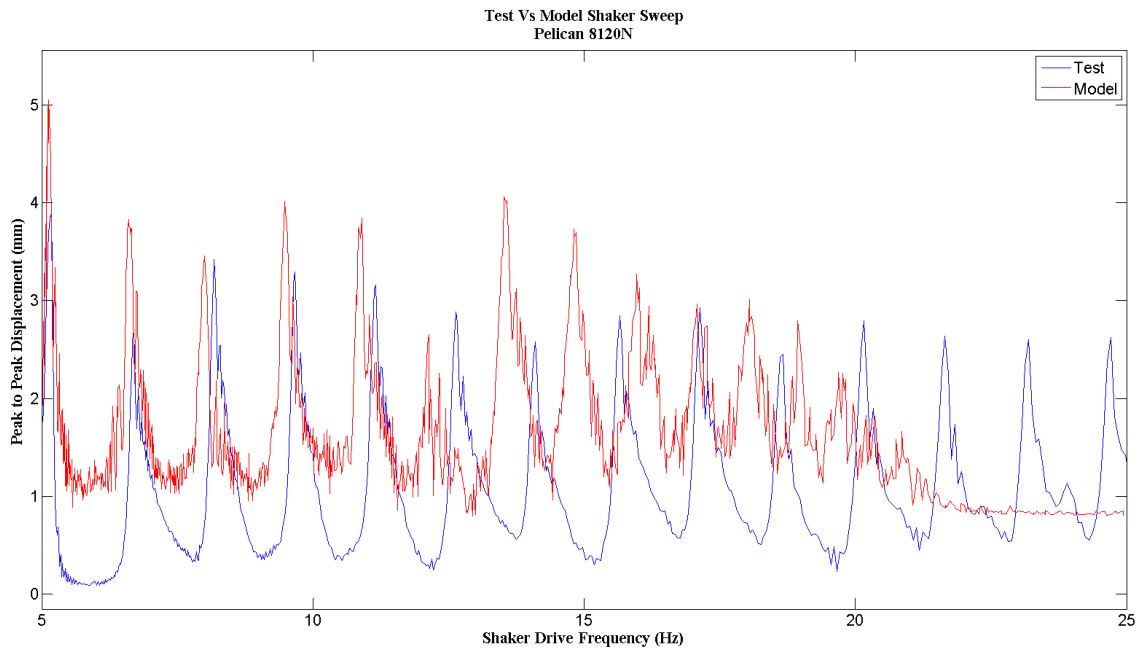


Figure 4.11: Pelican 8120N Sweep, Model vs. Test

For Pelican 8120N, the frequency peaks of the model results match with the experimental results initially but begin to drift quickly, with the gap between the model resonance peaks leading the experimental results increasing as the drive frequency increases. The rise and roll off of each model peak compares favourably to the experimental results.

The mean magnitude of the model results is higher, with the resonance peaks matching the absolute value of the experimental results relatively well. The model results decay to a near constant minimum magnitude after approximately 22Hz.

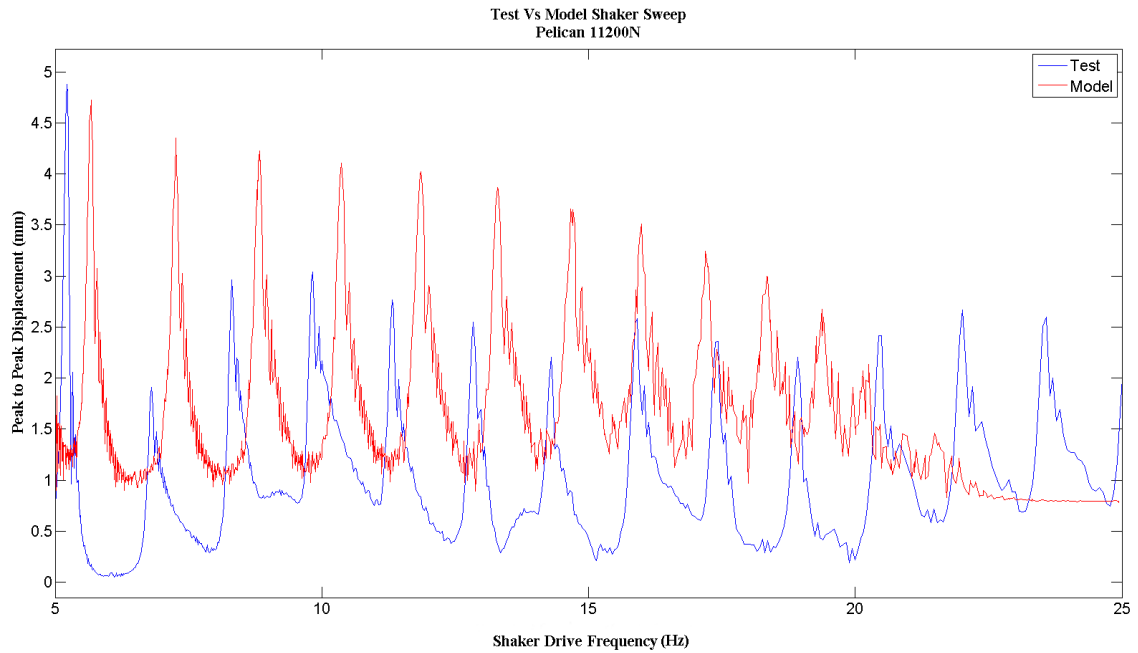


Figure 4.12: Pelican 11200N Sweep, Model vs. Test

For Pelican 11200N, the model results resonance peaks occur at noticeably different frequencies compared to the test results peaks, but the model results maintain a similar spacing between peaks compared to the test results initially before the pitch begins to decrease due to drift.

The mean magnitude of the model results is higher than the experimental results, with the peak model magnitudes being larger than the test results.

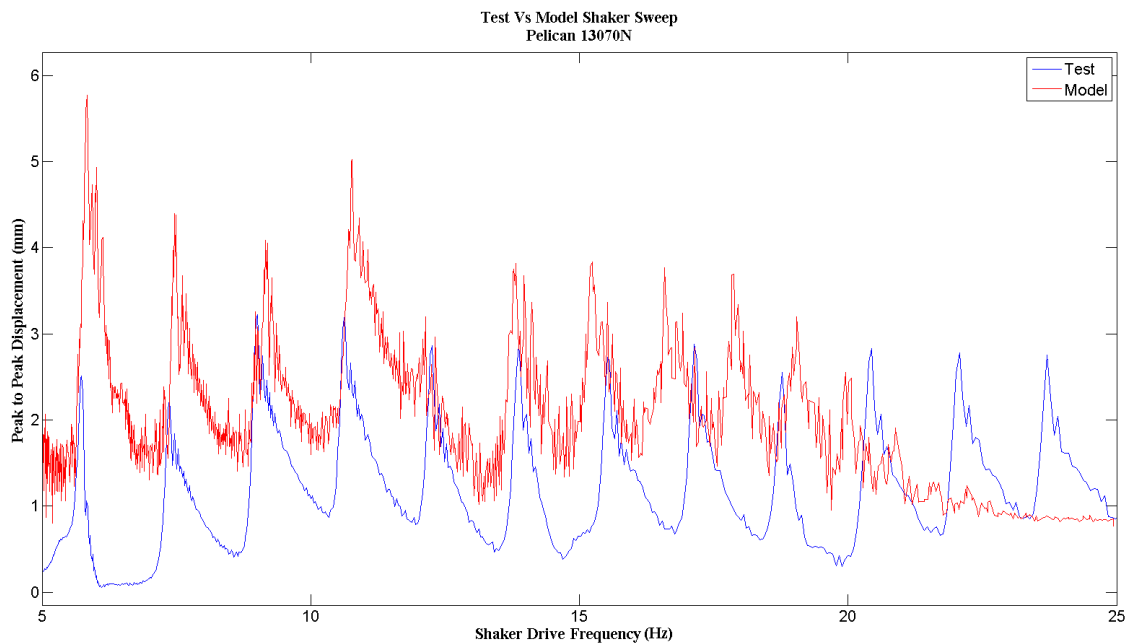


Figure 4.13: Pelican 13070N Sweep, Model vs. Test

The first four peaks of the model results track well with the test results with regard to the frequency value at which the resonance peaks occur. The average magnitude of the model vibration results is higher than that of the test results.

The profile of the rise to peak magnitude and roll off for each individual peak of the model results agree reasonably well. The resonance peaks of the model results continue longer than that of the Rabbit conductor in Fig 4.10 but they begin to drift after the 4th peak, before eventually decaying to a near constant minimum after 22 Hz.

4.6 Constant Frequency Shaker Excitation

The point at which the shaker is connected to the conductor at the VRTC is offset from the centre of the span. This can result in transient effects due to travelling waves being generated. Due to the speed with which the frequency of the exciting source is increased during a sweep test it is possible that the results of the frequency sweep provide displacement results that are different to what would be obtained by exciting the conductor at a constant frequency.

For this reason tests were run with conductors that were set at different tensions and each test was performed with the conductor excited by the shaker at a constant frequency and constant shaker displacement. The data obtained from the accelerometers was in units of gravitational acceleration “g”. This data was converted to m/s^2 and then filtered with a Butterworth filter to remove the DC bias of the accelerometer. The resulting acceleration data was then double integrated to obtain the displacement of the conductor. The following figures show the results of the physical tests and modelled results comparison for the Rabbit and Pelican conductors.

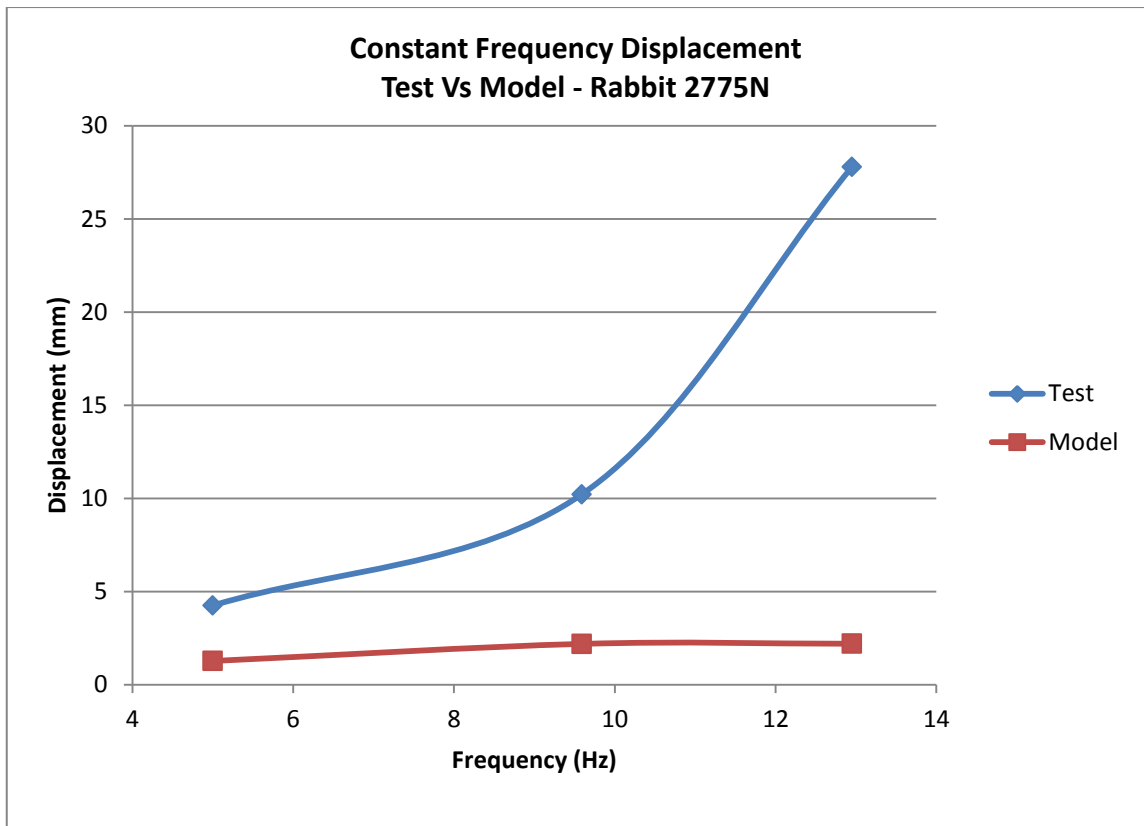


Figure 4.14: Constant Frequency Displacement - Rabbit 2775N

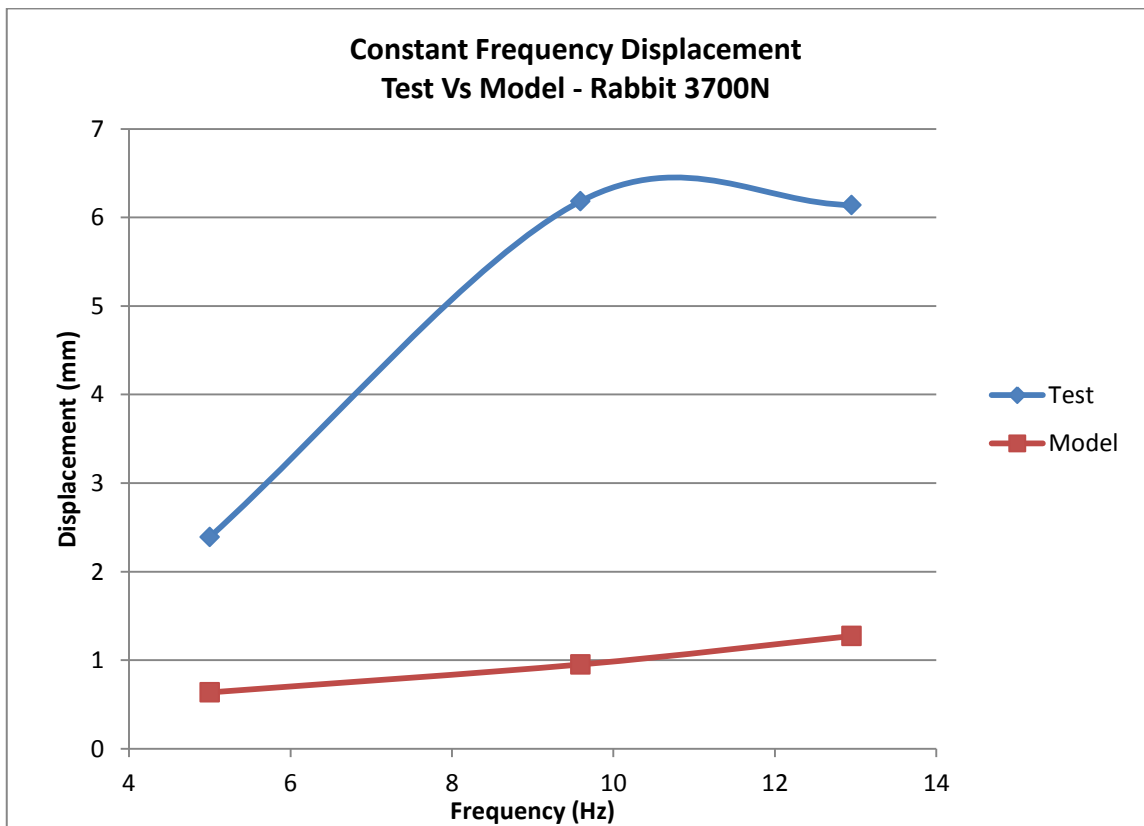


Figure 4.15: Constant Frequency Displacement - Rabbit 3700N

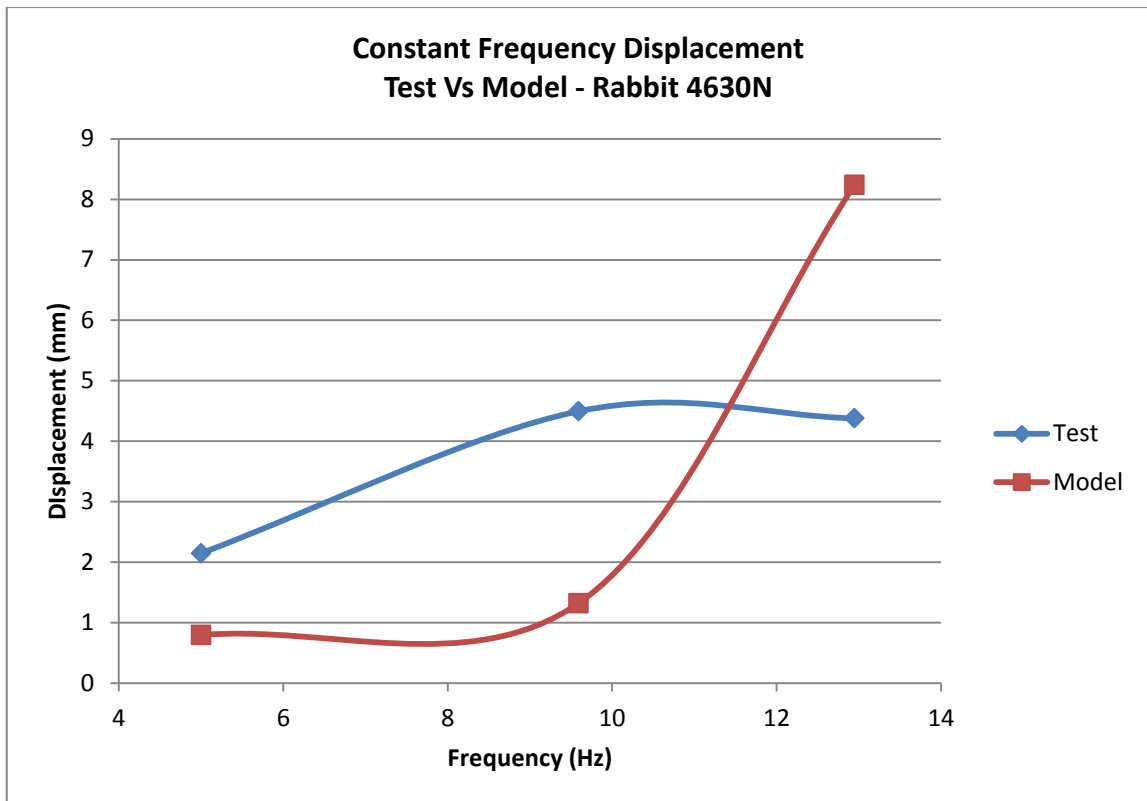


Figure 4.16: Constant Frequency Displacement - Rabbit 4630N

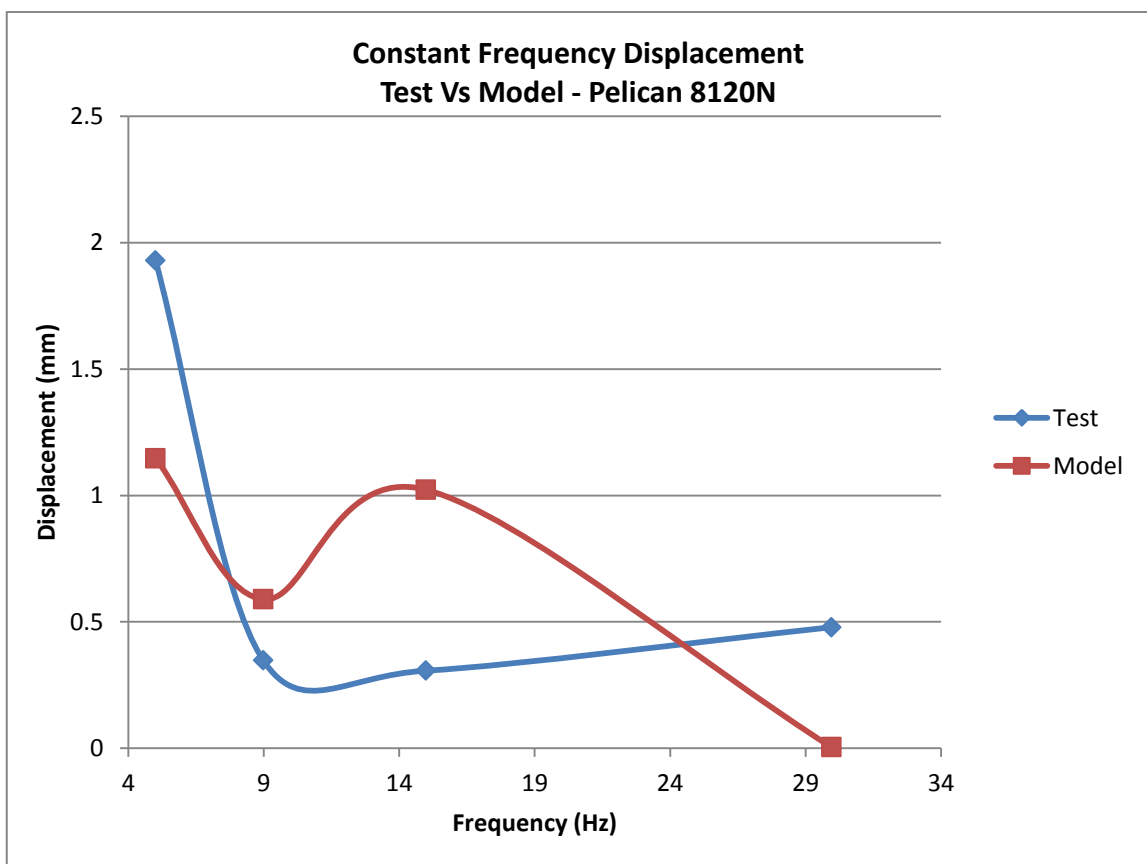


Figure 4.17: Constant Frequency Displacement - Pelican 8120N

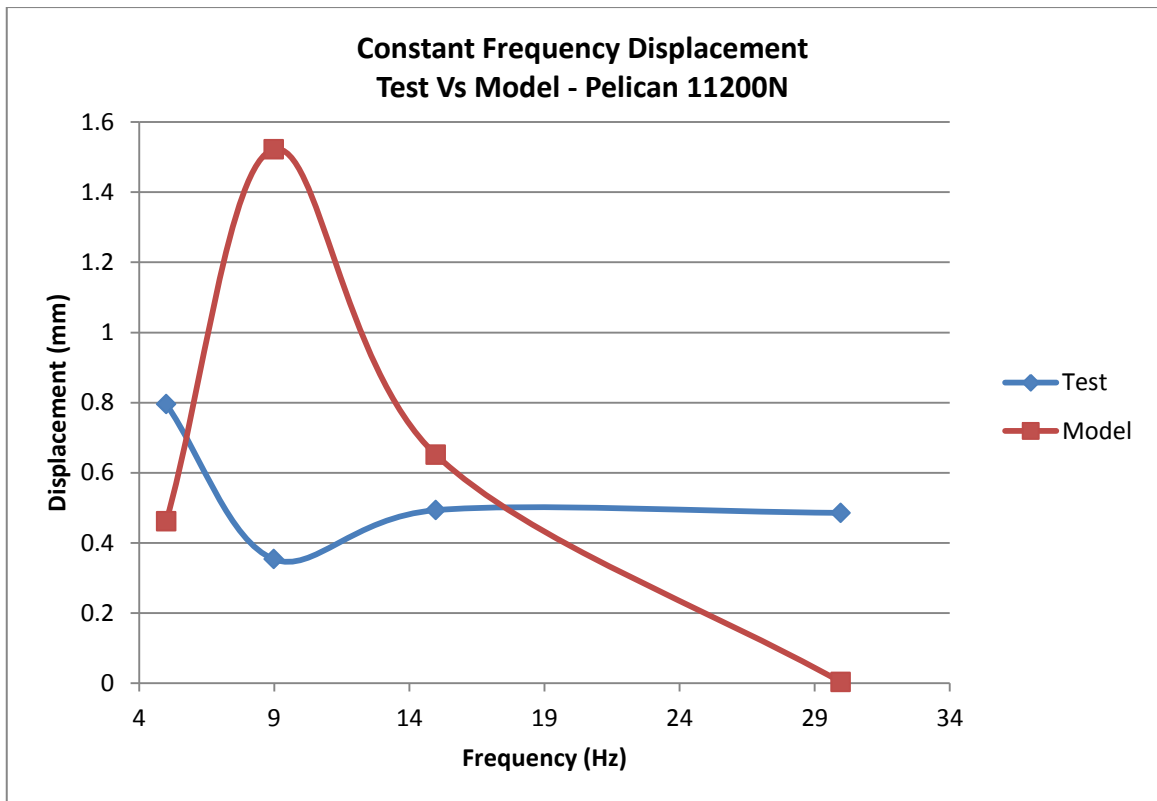


Figure 4.184: Constant Frequency Displacement Pelican 11200N

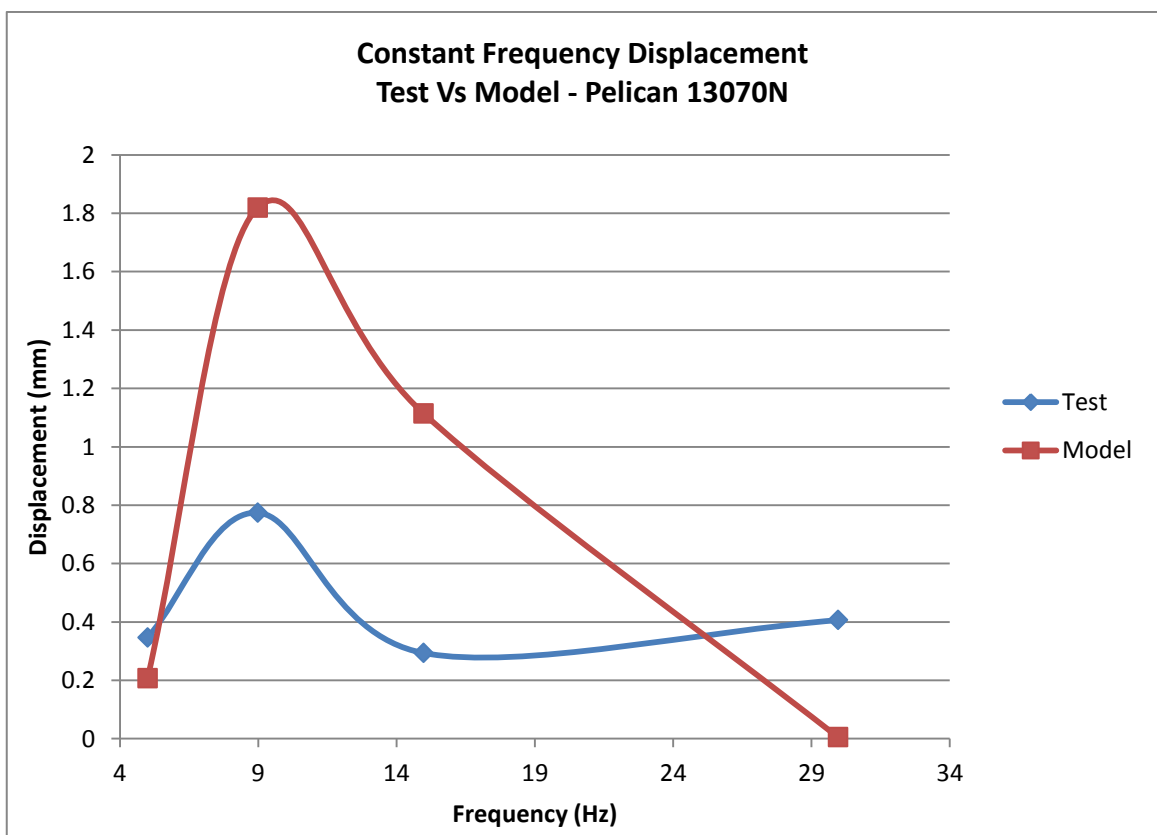


Figure 4.195: Constant Frequency Displacement – Pelican 13070N

A comparison of the results between the model and experimental shows a large range of values for the steady state vibration results. The model results of the steady state vibrations are not satisfactory when compared with the test results, with some of the Pelican model results being near zero. The only discernible trend is the tendency of the Rabbit model results to under predict the response of the experimental constant frequency.

4.7 Steady vs. Sweep Excitation

The method of testing a conductor at a specific tension was to sweep through it at a set tension and analyse the resulting displacement. After the conductor had settled to rest, three to four tests were performed, each at a different constant frequency. These frequencies were the kept the same for all the different tension settings. The following figures show the comparisons between the steady vs. swept drive mode at the same frequencies.

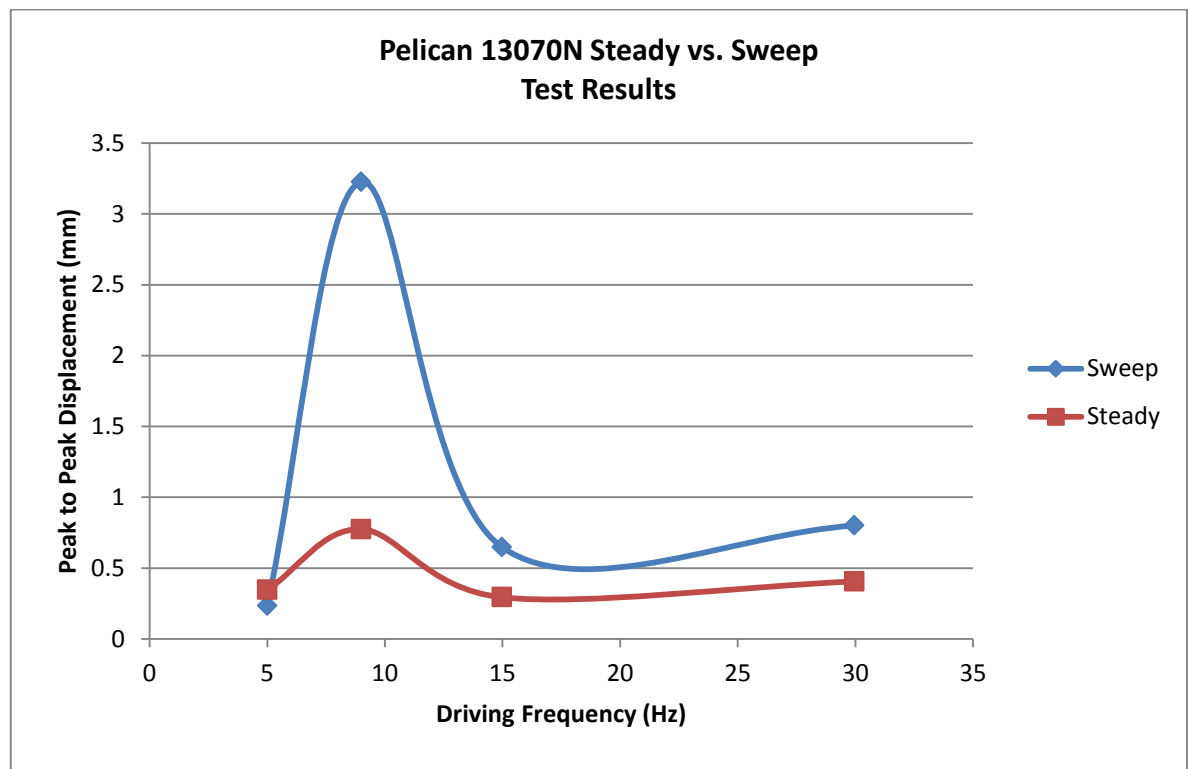


Figure 4.20: Pelican 13070N Steady vs. Sweep Test Results

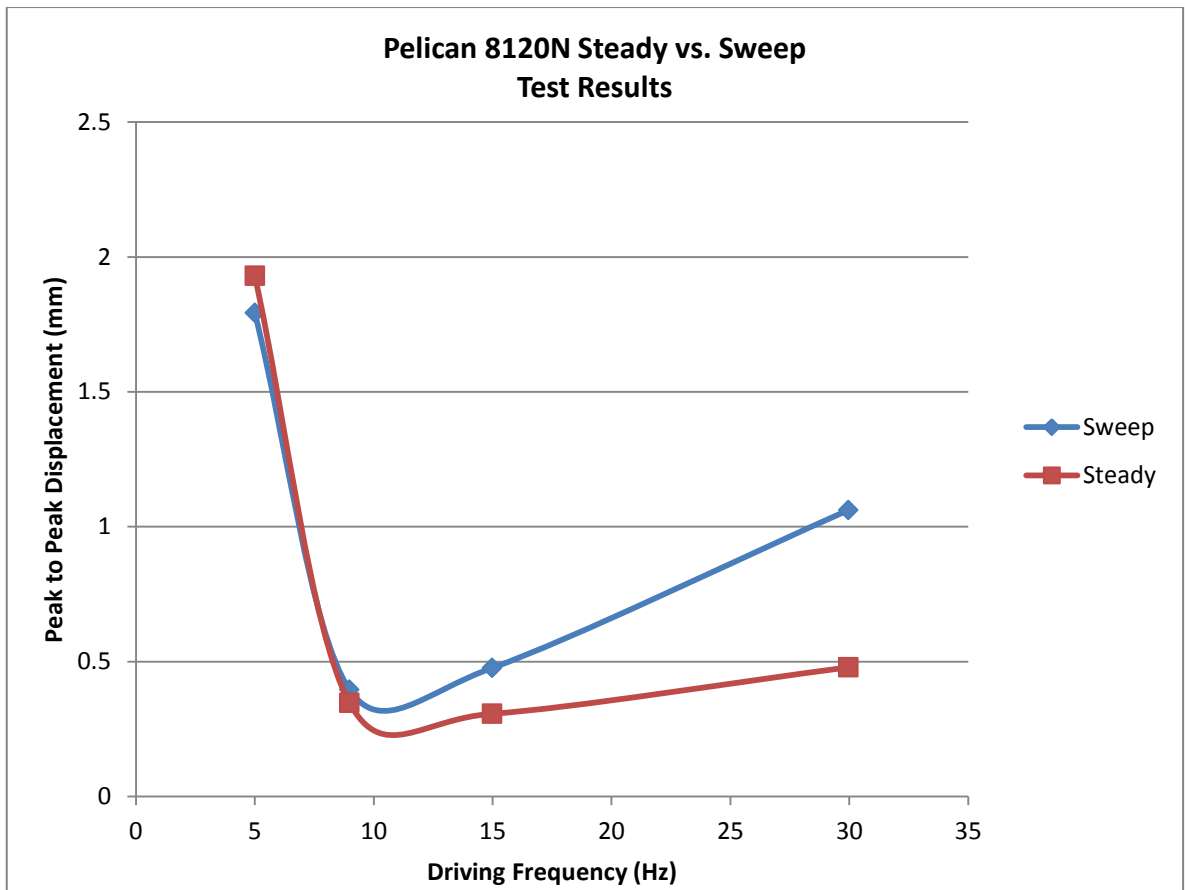


Figure 4.21: Pelican 8120N Steady vs. Sweep Test Results

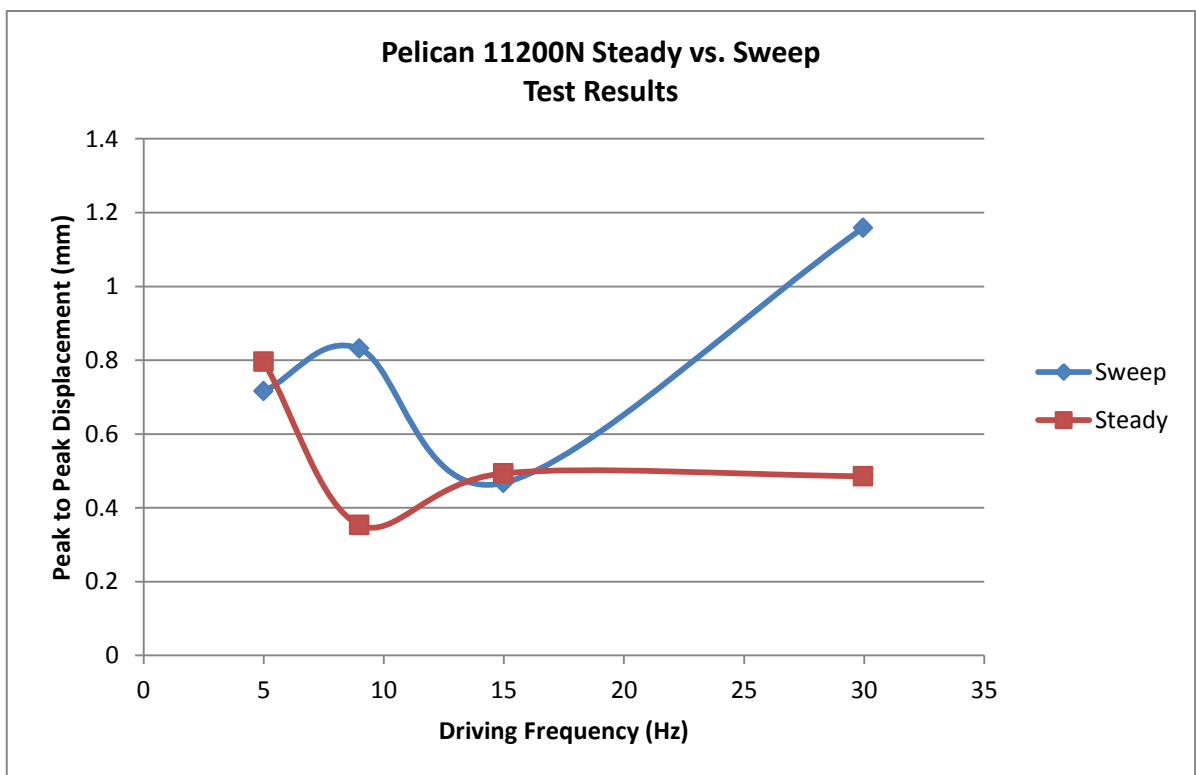


Figure 4.22: Pelican 11200N Steady vs. Sweep Test Results

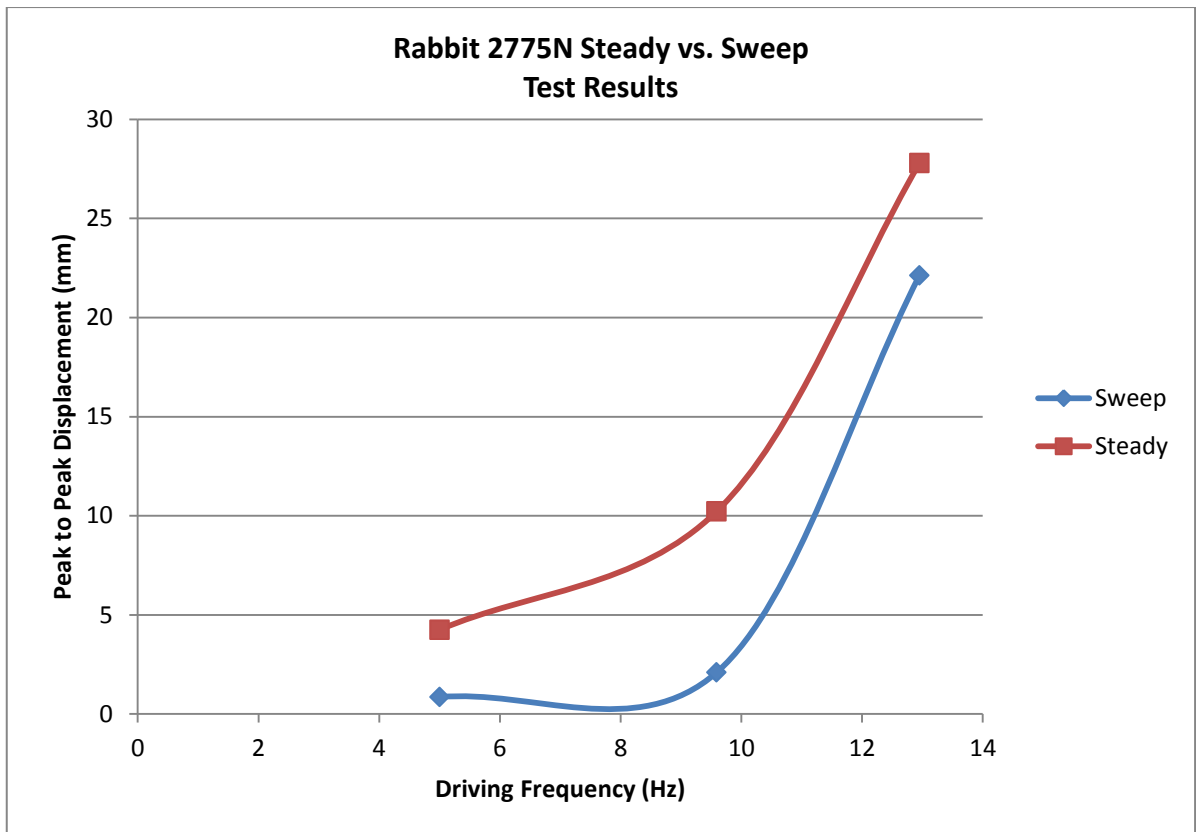


Figure 4.23: Rabbit 2775N Steady vs. Sweep Test Results

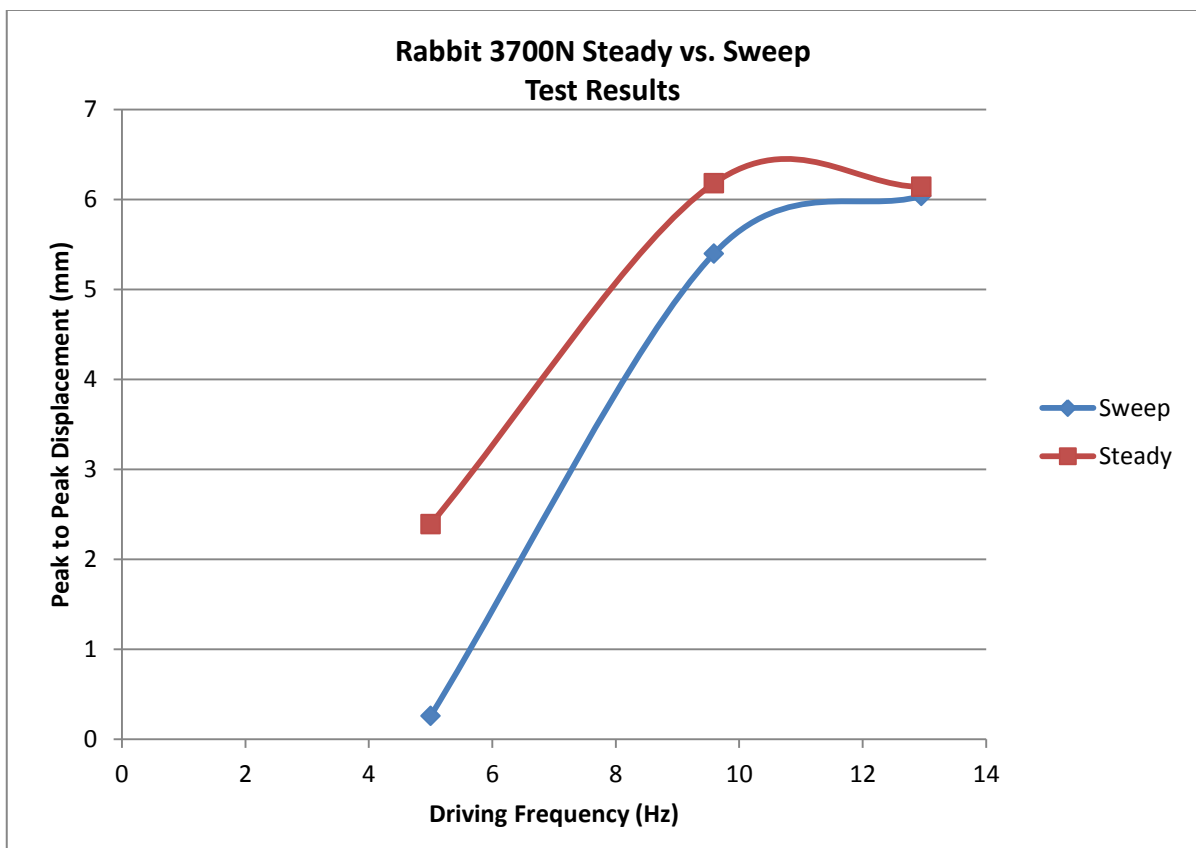


Figure 4.24: Rabbit 3700N Steady vs. Sweep Test Results

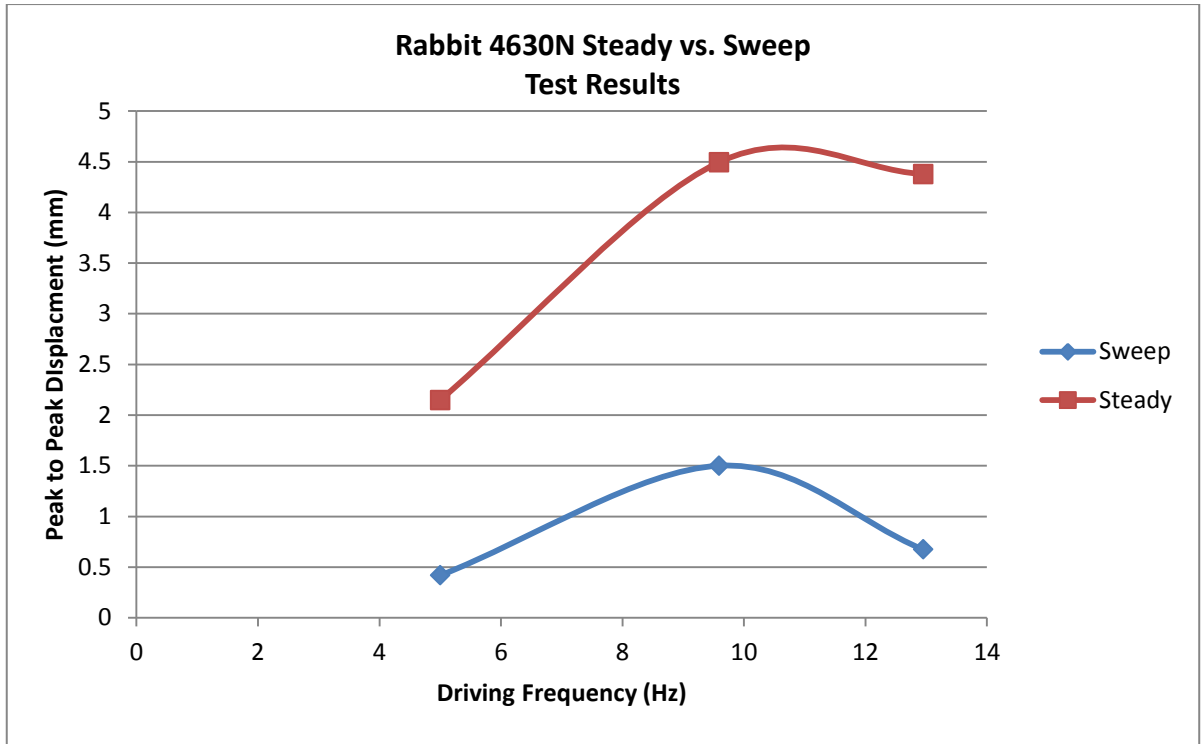


Figure 4.25: Rabbit 4630N Steady vs. Sweep Test Results

A comparison of the steady vs. swept excitations shows that the two different drive methods produce displacement responses at each conductor tension that follow similar trend lines, but of differing magnitudes at the same frequencies. However, the pattern of which drive method cause a larger displacement at a given frequency is not constant between the different conductors. For the Pelican conductor the swept results were higher than that of the steady results, but vice versa for the Rabbit conductor.

4.8 Wind Force Excitation

A sinusoidal force input was applied to the conductor model as a distributed load to replicate the effects of vortex shedding. The Pelican and Rabbit conductors were each modelled under the effects of two different reduced vortex shedding frequencies and corresponding coefficients of lift as taken from Mittal and Kumar (2001) at a Reynolds number of 1000. These two frequencies were:

$$f_s = 0.27 \text{ \& } C_l = 0.4$$

$$f_s = 0.35 \text{ \& } C_l = 2$$

The following figure shows the results of the Pelican conductor as modelled at different tensions and reduced frequencies.

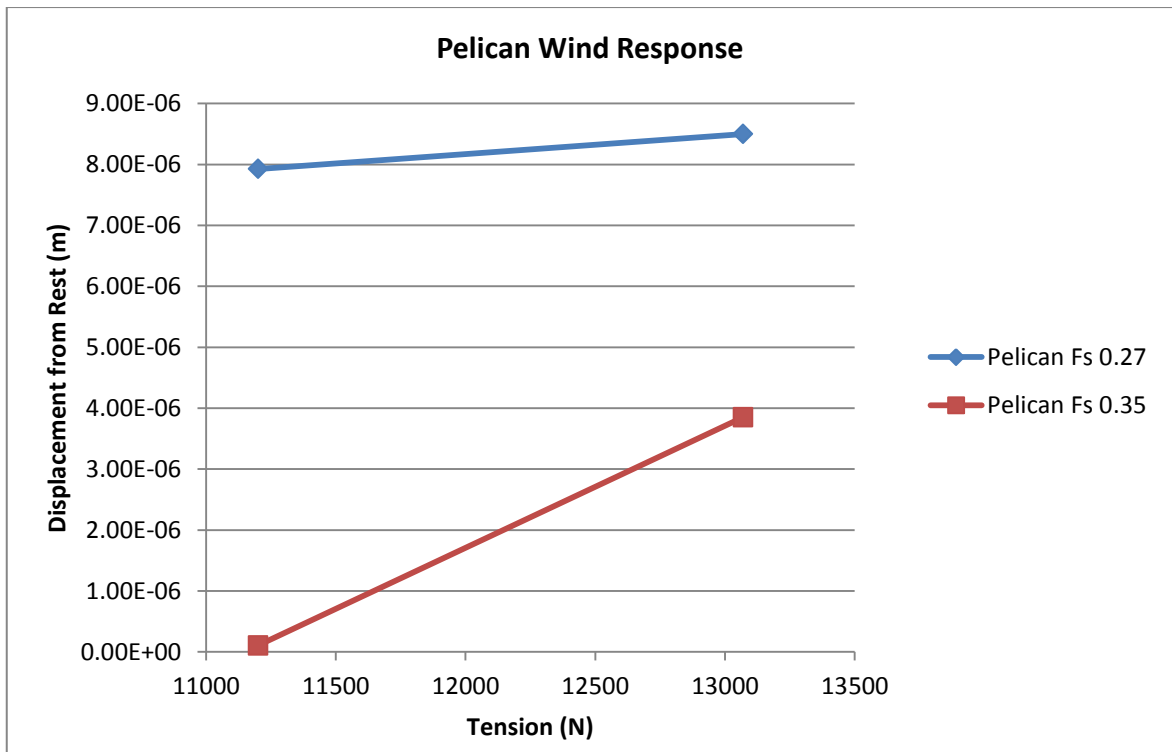


Figure 4.26: Pelican Modelled Wind Response

The following figure shows the results of the Rabbit conductor as modelled at different tensions and reduced frequencies.

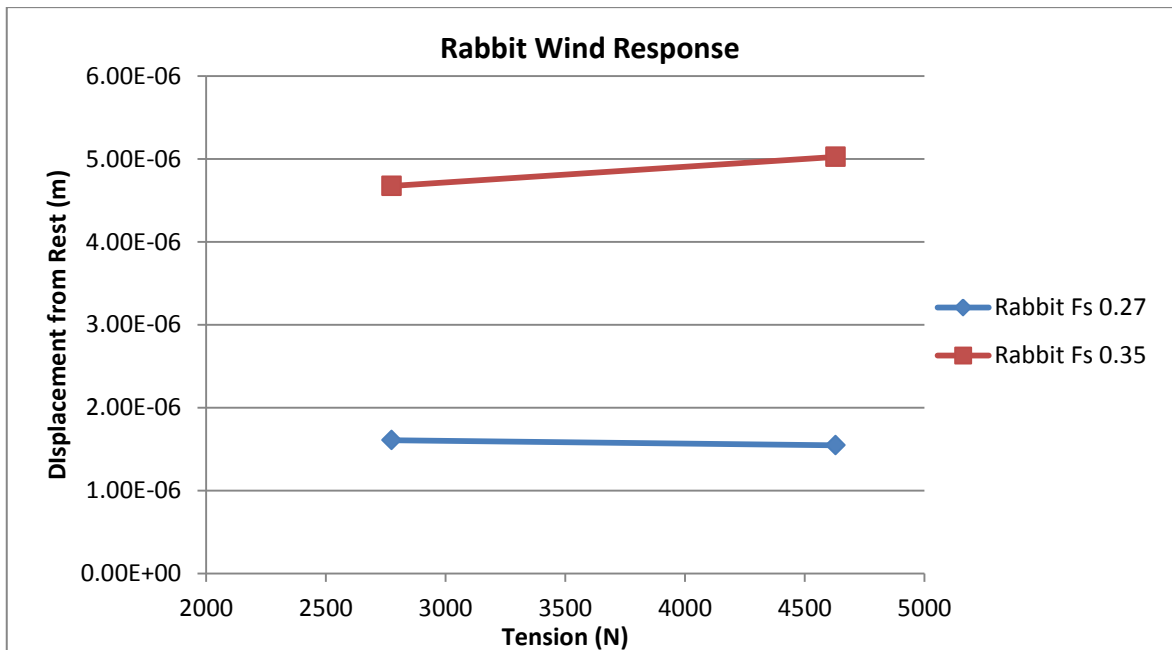


Figure 4.27: Rabbit Modelled Wind Response

Like the steady state displacement, the Pelican and Rabbit conductors show reversed trends to the same loading conditions. For the Pelican Conductor, the lower reduced vortex shedding frequency $f_s = 0.27$ produces a higher response in the model, whilst the Rabbit conductor has a higher amplitude response under the $f_s = 0.35$ loading.

The model results subject to the wind loading are unsatisfactory with the modelled vibration amplitude being far lower than is reasonably expected for a conductor undergoing aeolian vibration, which can have a displacement up to a maximum of one conductor diameter.

4.9. Factors Influencing the Model Results

Certain assumptions were made during the course of this research that had effects on the results obtained from the finite element model. Some of these assumptions were made to reduce the complexity of the model in order to reduce the computational resources needed to obtain a solution in a reasonable time frame. Other assumptions were made to provide reasonable limits on the scope of the research.

4.9.1 Effects of Linearisation

An assumption made at the outset of this research was to linearise the beam model of the conductor and to neglect the non-linear components of the model in order to avoid hysteretic effects. This was done to simplify the derivations of the differential equations as well as to reduce the computational load on the computer resources. By linearising the model, higher order effects that could alter the results were neglected that could have led to a loss of accuracy in certain sections of the research.

4.9.2 Effects of damping

The difficulty in applying a generalised damping model to a finite element model such as this as well as obtaining damping factors to use for the various conductors at their different tensions resulted in the Rayleigh damping method being used.

This method is easy to apply to a generalised dynamic model designed to accommodate a number of different conductor types and loading conditions, especially when the damping factor of the conductor is not known *a priori*. However Rayleigh damping is limited in that it is difficult to control the extent to which different modes are damped, especially at higher modes. It is possible that the damping used in this model was overly aggressive in the higher frequency ranges as evidenced by the sweep results decaying to near constants too early when compared to the experimental results.

Rayleigh damping is a linear combination of the mass and stiffness matrices. For this research two separate damping factors were used, with one set value used for Tensions below 12000N and another set for above 12000N. Rayleigh damping should provide a good method of dealing with variations in the overall damping due to the effects of internal energy losses which are related to the stiffness of the conductor element. However this model would be better suited to using a pair of Rayleigh damping coefficients that is a function of the external tension of the conductor.

4.9.3 The effects of bending stiffness

The bending stiffness EI of the finite element model was found using the parallel axis theorem, with the conductor modelled as a solid beam. The effects of inter-strand slip at increased curvature were neglected because of the decision to neglect hysteretic effects. This resulted in a model that would have been stiffer than reality as the curvature of the conductor increased, as inter-strand slip would have reduced the effective moment of inertia of the experimental conductor. The decision to ignore this inter-strand slip would have contributed to the modelled resonance peaks being shifted away from the peaks determined in the experimental results.

4.10 Effectiveness of Model

It was the intention of this research to develop a model that can obtain data on the behaviour of a conductor under a variety of loading conditions, most of which are impossible to test on a conductor in laboratory conditions. Mixed results were obtained across the different elements of the research that were investigated, with some sections such as the steady state and swept results being more successful for certain conductors than others, such as the applied wind excitation.

4.10.1 Distributed vs. Single Point Loading

The results of the distributed loading wind results were inconclusive, preventing a detailed comparison between the distributed and single point conductor excitation methods. This is most likely due to the generalised use of the wind power input frequency and coefficient of lift, with the applied frequencies of excitation being unable to cause a resonant response.

4.10.2 Steady State Position

The steady state results obtained by the model show a wide variation in the accuracy achieved, for the different conductors tested. The model was shown to be able to replicate a catenary curve as per the catenary theory for hanging conductors, but experimental measurements showed that overhead conductors do not follow exactly catenary theory. The results showed

that as the applied tension increased the model became more accurate. But the model also showed better results for the lighter conductors than the heaviest one, with the neglected non-linear effects of inter-strand slip and such possibly increasing as the conductor weight increased.

4.10.3 Swept Frequency Results

The modelled results of the swept frequency excitation showed were of mixed success. The finite element model was shown to be capable of capturing the basic behaviour of a conductor subject to a sweep test, with the model replicating the profile of the resonance peaks as well as the individual peak shape.

The results were not consistent though, with certain conductors and tension settings for the model achieving results that were of a higher error than others. For the Pelican conductors the modelled results were of a higher displacement as measured at the troughs than the experimental results. The Rabbit modelled well for some tensions but not others. Unfortunately the results obtained by the model for the Tern conductor were unusable, due to the magnitude of the response being too small. This indicates that the model is better suited to smaller weighted conductors of a lower external tension, as it is likely that the larger the conductor becomes, the more the resonant effects become overwhelmed by the overall conductor stiffness.

4.10.4 Constant Frequency Results

The constant frequency results of the model were not in good agreement with the test results. The displacement obtained by the model at a constant frequency was markedly different from the experimental results for certain frequencies. This comparison can be misleading though, as minor variations in the resonance peaks between the model and experimental results would result in large differences when comparing the two at a specific excitation frequency.

This shows that the model is better suited to analysing conductor behaviour over a swept frequency excitation mode as compared to investigations into behaviour at a specific frequency, as small variations in the model can lead to large differences in the resonant response of the conductor.

Chapter Summary

In this chapter the results of the model were presented and compared to the experimental results obtained from the conductors that were tested at the V.R.T.C. The steady state condition of the conductors under gravity was analysed. The procedures that were used to perform swept frequency tests on the conductors were detailed and comparisons were made between the modelled and experimental results.

The effects of sweeping the single shaker excitation frequency as compared to exciting at a constant frequency were discussed, with the behaviour of the different conductors noted. The results of the model under constant frequency excitation were compared against the experimentally obtained results. Also considered were the results of a harmonic distributed wind load applied to the conductor model.

The limitations of the assumptions made during the course of the research were listed and the possible effects they had on the model results were noted. The effectiveness of the derived finite element model in predicting the experimentally obtained results and achieving the stated objectives were considered.

CHAPTER 5

Case Study Results Comparison

5.1 Introduction

The model derived for this dissertation were compared to experimental results obtained in a study by Castello and Matt (2011) of the Universidade Federal do Rio de Janeiro, Brazil. They used a validation based metrics model applied to standard cables. The cables were tested in an IEEE standard facility to test the behaviour of overhead conductors subjected to a shaker excitation. [15]

As part of their research the Frequency Response Functions of the conductor were derived. By comparing their results to those obtained by the model derived during the course of this dissertation, the effectiveness of the model in determining the vibration characteristics of a conductor from limited information can be judged.

5.2 Frequency Response Function

As noted by Storer (1991), the frequency response function is a method of interpreting the resonance characteristics of systems by analysing the measured Fourier transforms of the harmonic frequency response of a system to an applied harmonic input force.

The relation takes the form

$$H_{rs} = \frac{Y_s(\omega)}{X_r(\omega)} \quad (5.1)$$

where $Y_s(\omega)$ is the response at the point of measurement s , $X_r(\omega)$ is the exciting force at a point r and H_{rs} is the resultant frequency response function. [31]

The model FRF results were calculated from the model results by taking the fast Fourier transform of the measured response and forcing input.

5.3 Test Setup

Castello and Matt performed the tests on a sample of Conductor codenamed “Grosbeak”, with the physical properties as detailed in Table 5.1:

Table 5.1: Conductor Specifications

Name	Grosbeak 16481N	Grosbeak 21778N
Cross-Sectional Area	374.34 mm ²	374.34 mm ²
Bending Stiffness EI	514 Nm ²	714.5 Nm ²
Density	1.3027 kg/m	1.3027 kg/m
Percentage UTS	14%	19%

The span used was 51.950m in length, with an electrodynamic shaker placed 1.6m from a terminated end. Measurements were taken from an accelerometer placed 1.4m from the opposite terminated end. For the FRF measurements, the conductor was excited over a frequency range from 5 to 17.5 Hz in a linearly increasing ramp.

5.4 Results

The FRF results by Castello & Matt for the conductors measured at the different tensions are shown in figure 5.1.

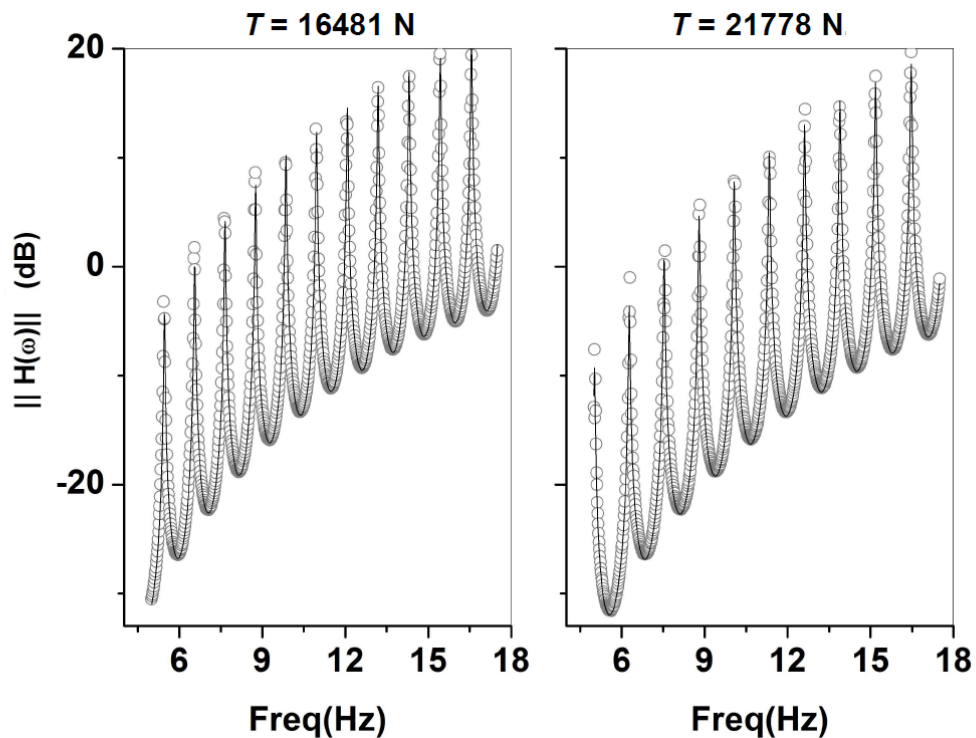


Figure 5.1: Experimental Frequency Response Function of Grosbeak (Castello & Matt)

Using the FEM model derived for this research, the Grosbeak conductor was modelled under the same conditions and excited over the same frequency spectrum within the model space, with

the Frequency Response Function being calculated for the different tensions. Figures 5.2 and 5.3 show the model FRF results.

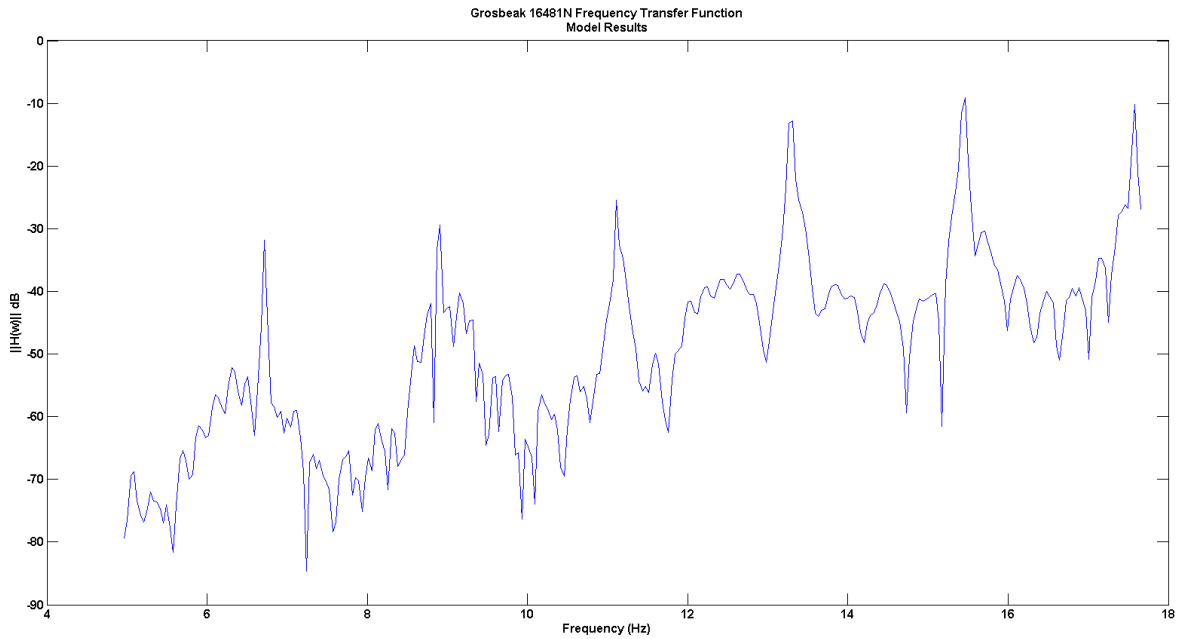


Figure 5.2: Grosbeak 16481N FRF Model Results

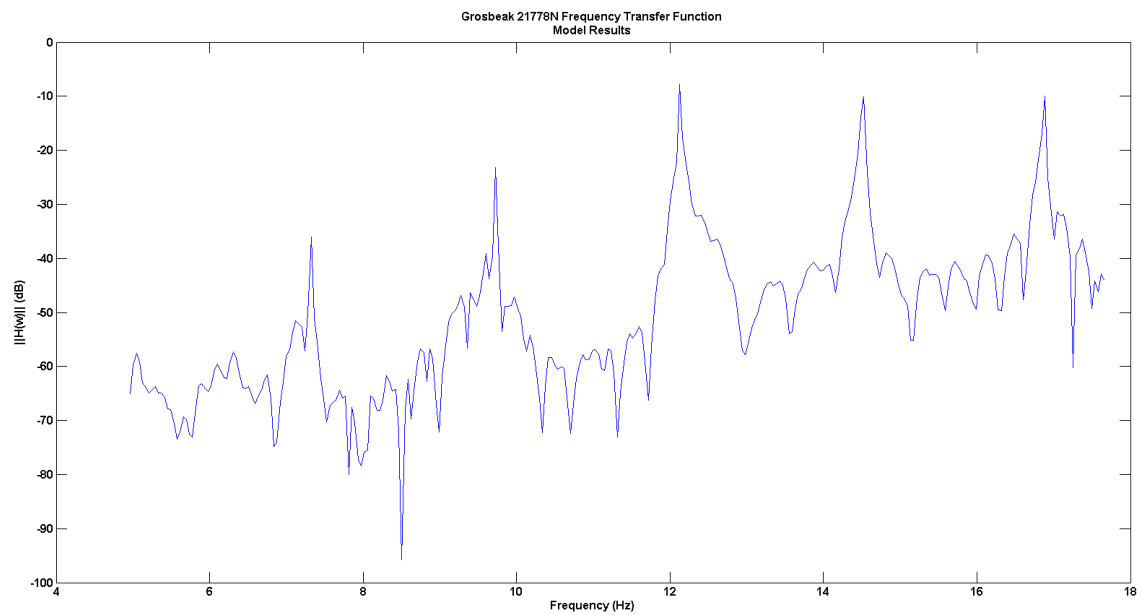


Figure 5.3: Grosbeak 21778N FRF Model Results

A comparison between the model and test result show that the model results cannot capture all the resultant FRF peaks of the conductor as tested by Castello & Matt. The model results follow the correct basic trend for the two tension settings in that the response is characterised with rising peaks and troughs with a global rise in magnitude. The model results are of a lower

overall magnitude in comparison to the experimental results and do not have as strong a global rise.

The model achieved distinct resonant peaks as shown in the FRF results, but did not achieve satisfactory results in the number of resonant peaks achieved throughout the sweep. The frequency at which the resonant peaks occur in the model did not agree well with the experimental results. The model results did achieve a regular spacing in the position of the resonant peaks. The frequency of the peaks in the model shifted higher as the tension in the conductor increased, which matched the trend shown in the experimental results.

Chapter Summary

The results of experimental tests on a conductor using a single point shaker were compared to the model results obtained using the model derived for this research. The frequency response function of the conductor was obtained from the model and compared to the FRF results obtained by Castello & Matt.

The derived finite element model showed that it can obtain an FRF for a conductor that is of the correct general behaviour when compared to the results of Castello & Matt, but the model lacks the accuracy to properly replicate the experimental results.

CHAPTER 6

CONCLUSION

6.1 Introduction

The subject of aeolian vibrations is one characterised by the interactions of different forces and systems. The vibrations occur when wind blowing across the length of the conductor causes vortices to be shed at the structural resonance frequency of the conductor. In this way, the fluid system serves as the force input to the structural system. This research dissertation focused on investigating and modelling the structural system of this fluid-solid interaction, with the effects of the wind serving as a force input to the modelled system.

The aim of this research was to develop a computational model that could be used to investigate the behaviour of overhead conductors subject to various loading conditions. The model was compared to the results obtained from the conductors under test at Vibration Research Testing Centre. It was the intention that the finite element model would provide a method of analysing conductor behaviour in various scenarios that would be difficult to investigate in the real world, as well as shed light on the effects of using a single point shaker when exciting a conductor.

A finite element model based on the linearised equations for beams in bending was derived with the external pre-tensioning load as applied to conductors during installation being included in the model. The shaker and flexible coupling was modelled and included in the model system as a force input. This model was implemented within the MATLAB computing environment.

6.2 Research Outcomes

The model was used to predict conductor behaviour in comparison to an existing experimental conductor installation at the Vibration Research Testing Centre. The model and experimental tests were compared in various conditions:

- Steady State Resting Displacement
- Constant Frequency Single Point Excitation
- Swept Frequency Single Point Excitation
- Distributed Wind Loading
- Frequency Response Function

The model results were of limited success, with varying levels of accuracy achieved. The model was shown to be capable of replicating the basic aspects of the conductor systems but could not adequately match the experimental results under the various conditions.

For the steady state resting displacement, the model adequately replicated the shape of a conductor hanging under the force of gravity, with the external tension causing the proper effects. The accuracy of the results was across the range of conductors was not regular though.

The constant frequency results were of limited accuracy in comparison to the experimental results. They were of a similar scale but showed little pattern in their results. It should be noted that the effectiveness of comparing individual frequency results is limited as a minor shift away from a resonance point will result in a large shift in amplitude, so comparing the model and test results could be misleading.

The swept model results showed that the model is capable of achieving the correct behaviour of a vibrating conductor when compared to the experimental results, in that the model replicated the resonance behaviour. The model lacked accuracy in the value of the frequency at which these resonance points occur, but the spread between peaks and the resonance profile matched the experimental results well.

The distributed wind results were of limited use. The model response using the determined wind force input was of a scale too small to be useful. It is unclear if the results were poor due to the model being overly stiff; the resonance peaks as determined using aeolian theory did not match the resonance patterns of the conductor; or if the wind force as applied was of insufficient strength.

The comparison between swept vs. steady frequency excitation provided an interesting result, in that it showed the difference in magnitude between the two modes of excitation. Also of interest was the uniform trend shown in the responses as they follow the same general curve. Surprisingly the trend of which mode of excitation provided a larger response was not the same for the different conductors.

The frequency response functions were found for a conductor codenamed “Grosbeak” at different tensions and compared to the experimental results obtained by Castello & Matt. The modelled frf’s were not sufficiently accurate to be useful, but the model was shown to be capable of obtaining a frequency response function that follows the correct trend.

6.3 Limitations

The performance of the model in replicating the behaviour of the conductors was shown to be limited in the accuracy of the results. The finite element method applied in this manner is capable of reproducing the basic behavioural elements of different conductors under various loads but was not sufficiently accurate enough to properly replicate the experimental results.

The following are possible reasons for why the model did not achieve satisfactory results.

1. The application of the Rayleigh Damping method to this research was not tuned specifically to the free vibration problem before conducting the forced vibration results, in an attempt to model the vibration characteristics without prior access to a test setup to explicitly determine the damping factors. This resulted in the damping being unsatisfactory and damping the various modes incorrectly.
2. Increasing the element number of the model is limited by the availability of adequate computing power to solve large complex systems over long time frames.
3. The modelled distributed wind force input was of insufficient strength to cause a useful result, possibly due to the damping being too strong for the frequency range of the wind force

By addressing these limitations the accuracy of the model could be improved, particularly in the area of the wind-induced distributed loading.

6.4 Recommendations

Following on from this research certain recommendations can be made for future investigations.

The conductor damping should be improved, with the damping being made a function of a number of model characteristics, such as tension and stiffness of the conductor. The aerodynamic damping effects could also be considered. Model testing could also be limited to smaller sweep frequency ranges which would allow a more focused damping model for a specific frequency range.

The method of applying the external tension to the model should be improved, either by being applied within the model stiffness matrix or as part of the external forcing function. This would improve the final results of the model at rest, which would in turn improve the vibration results by ensuring the pre-loading is correct, which has a strong effect on the stiffness of the system.

The effects of non-linear conductor bending could be investigated, such as conductor slip and inter-strand friction. These result in hysteretic effects within the conductor during bending and

become more prominent as overall conductor bending increases, such as when pre-loaded under gravity.

A more rigorous application of the wind force input to the model system should be developed and implemented. The force input could be modelled such that it includes the effects of wind in the transverse direction of the conductor to more closely model the effects of aeolian vibrations.

Improving the accuracy of the model by increasing the number of elements is not necessarily the best proposition, as the current choice of 60 elements for a conductor 86 metres long resulted in simulations that ran for multiple 24 hour periods. Increasing the number of elements would result in a slight improvement of accuracy at the expense of substantially increased run-time as well as exposing the possibility of the simulation failing due to insufficient computer memory. Increasing the element number by a significant amount would require access to substantially more powerful computer resources.

6.5 Research Summary

This research demonstrated that a finite element model could be derived to investigate the wind-induced vibration of overhead conductors in various loading conditions. The model was designed and used to investigate the behaviour of the overhead conductors under a range of different loading conditions and to analyse the effects of using a single-point shaker when testing an overhead conductor for resonance behaviour. The model was compared to results obtained by other researchers in similar research conditions.

Limitations in scope of the research reduced the models effectiveness in obtaining results that could be used to draw firm conclusions about the behaviour of conductors under the various loading conditions. It was shown however that the model successfully replicated the basic behavioural components of the conductors under test.

The results of the model and test results were discussed and the limitations in the model were considered. Possible avenues to improve and continue the research were presented and detailed.

REFERENCES

1. EPRI (2006) *Transmission Line Reference Book – Wind Induced Conductor Motion*. EPRI: Calif.
2. Reira J.D., Davenport A.G. (ed.) (1998) *Wind Effects on Buildings & Structures*. Taylor & Francis
3. E.L. Houghton, N.B. Carruthers (1976) *Wind Forces On Buildings And Structures: An Introduction*. London: Arnold.
4. American Society of Civil Engineers. Wind Effects Committee (1987) *Wind Loading and Wind-induced Structural Response: A State-of-The-art Report*. New York: American Society of Civil Engineers.
5. Zhang Q. et al. (2000) *Galloping of Bundle Conductor*. Journal of Sound and Vibration. 234(1) p.115-134.
6. Sarpkaya T. (2004) *A Critical Review of the Intrinsic Nature Of Vortex-Induced Vibrations*. Journal of Fluids and Structures. 19 (4) p.389-447
7. Mittal S. and Kumar V. (2001) *Flow-Induced Vibrations of a Light Circular Cylinder at Reynolds Numbers 10^3 to 10^4* . Journal of Sound and Vibration. 245(5) p.923-946.
8. Moe G. and Wu Z.-J (1990) *The Lift Force on a Cylinder Vibrating in a Current*. Journal of Offshore Mechanics and Arctic Engineering. 112(4) p. 297-303.
9. Meynen S. et al. (2005) *On the Numerical Simulation of Vortex-Induced Vibrations of Oscillating Conductors*. Journal of Fluids and Structures 21 p.41-48.
10. Vecciarelli J., Currie I.G. and Havard D.G. (2000) *Computational Analysis of Aeolian Conductor Vibration With A Stockbridge Damper*. Journal of Fluids and Structures. 14(4) p.489-509.
11. Canadian Electrical Association (1988) *Development of Improved Analytical Models for Vibrating Conductor-Damper Systems*. Canadian Electrical Association: Montreal.
12. Rao S. (2011) *The Finite Element Method in Engineering*, 5th Ed. Amsterdam: Elsevier.
13. Tibert G. (1999) 'Numerical Analysis of Cable Roof Structures'. PhD thesis, Royal Institute of Technology, Stockholm.
14. Wei P. et al. (1999) *A Catenary Element for the Analysis of Cable Structures*. Applied Mathematics and Mechanics. 20(5)
15. Castello D. & Matt C. (2011) *A Validation Metrics Based Model Calibration Applied on Stranded Cables*. Journal of the Brazilian Society of Mechanics, Science and Engineering. 33(4) p.417-427

16. Gatulli et al. (2004) *Nonlinear Oscillations of Cables Under Harmonic Loading Using Analytical And Finite Element Models*. Computer Methods in Applied Mechanics and Engineering. 193 p.69-85.
17. Balachandran B. and Magrab E. (2008) *Vibrations 2nd Ed.* London: Cengage Learning.
18. Sondipon A. (2000) 'Damping Models for Structural Vibration'. PhD thesis, Cambridge University, Cambridge.
19. Spears R.E. and Jensen S.R. (2009) *Approach for Selection of Rayleigh Damping Parameters Used for Time History Analysis*. 2009 ASME Pressure Vessels and Piping Division Conference. PVP2009-77257
20. Papailiou K. (1995) 'Bending of Helically Twisted Cables Under Variable Bending Stiffness due to Internal Friction, Tensile Force and Cable Curvature' PhD thesis, Eidgenossische Technische Hochschule Zurich, Zurich.
21. Ren W.X. (2008) *A Parabolic Cable Element for Static Analysis of Cable Structures*. Engineering Computations: International Journal for Computer-Aided Engineering and Software. 25(4) p.366-384.
22. Claren R. and Diana G. (1969) *Dynamic Strain Distribution on Loaded Stranded Cables IEEE Winter Power Meeting*. 69 TP 73.
23. Diana G. et al. (2000) *On the Measurement of Overhead Transmission Lines Conductor Self-Damping*. IEEE Transactions on Power Delivery. 15(1) p.285-292.
24. Xia Y. et al. (2011) *Verification of a Cable Element for Cable Parametric Vibration of One-Cable-Beam System Subject to Harmonic Excitation and Random Excitation*. Advances in Structural Engineering. 14(3) p.589-595.
25. Mohammad D.R.A. et al. (1995) *On the Role of Rayleigh Damping*. Journal of Sound and Vibration. 185(2) p. 207-218.
26. Chowdury I. and Dasgupta S. (2003) *Computation of Rayleigh Damping Coefficients for Large Systems*. [Online] The Electronic Journal of Geotechnical Engineering, 8(C), available from <http://www.ejge.com/2003/JourTOC8C.htm> [Accessed: 22 February 2012]
27. Barbieri et al. (2004) *Dynamical Analysis of Transmission Line Cables. Part 1 – Linear Theory*. Mechanical Systems and Signal Processing. 18(1) p.659-669.
28. Yang J. et al. (2010) *Dynamic Response Analysis and Disaster Prevention of Transmission line under Strong Wind*. 2010 International Conference on Power System Technology.
29. Bucknell University. *Helpful Information For Using MATLAB*. [Online] Available from: <http://www.facstaff.bucknell.edu/maneval/help211/helpmain.html> [Accessed: 18th August 2010].

30. Teichman J. and Mahadevan L. (2003) *The Viscous Catenary*. Journal of Fluid Mechanics. 478 p.71-80.
31. Storer D. (2003) 'Dynamic Analysis of Non-Linear Structures Using Higher Order Frequency Response Functions' PhD thesis, University of Manchester, Manchester.

APPENDIX A

Interpolation Polynomials

$$\underline{q} = [N] \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix}$$

$$N_1 = 1 - \frac{x}{l}$$

$$N_2 = 1 - \frac{3x^2}{l^2} + \frac{2x^3}{l^3}$$

$$N_3 = x - \frac{2x^2}{l} + \frac{x^3}{l^2}$$

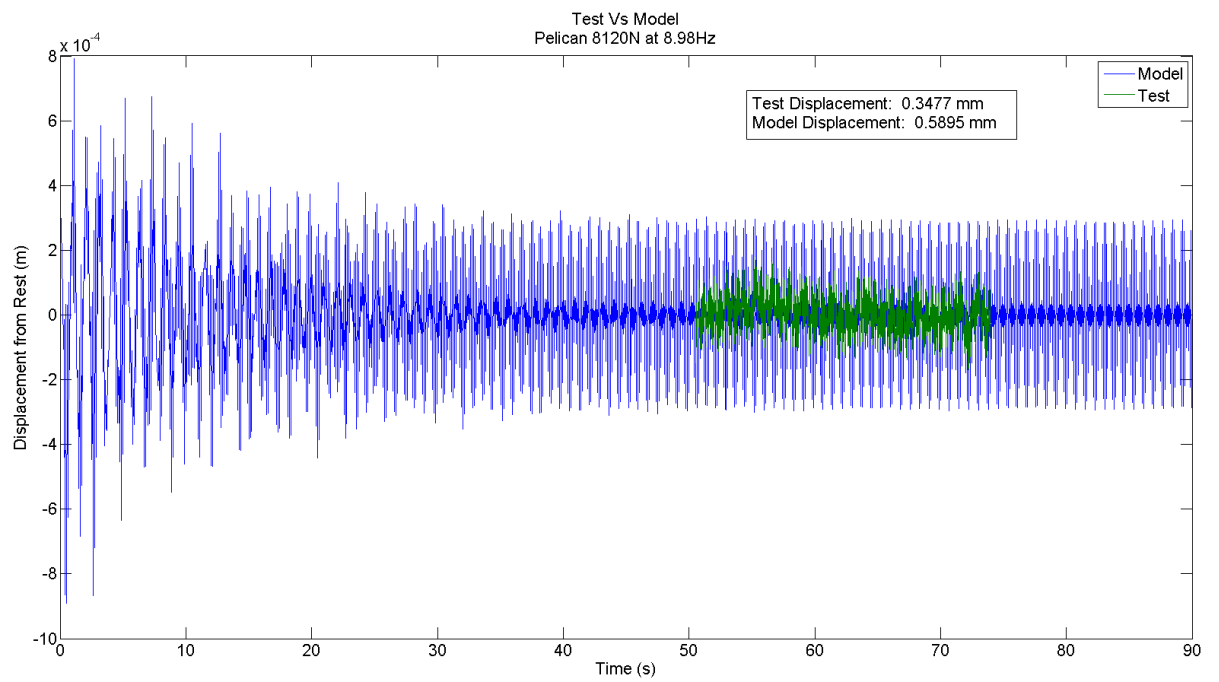
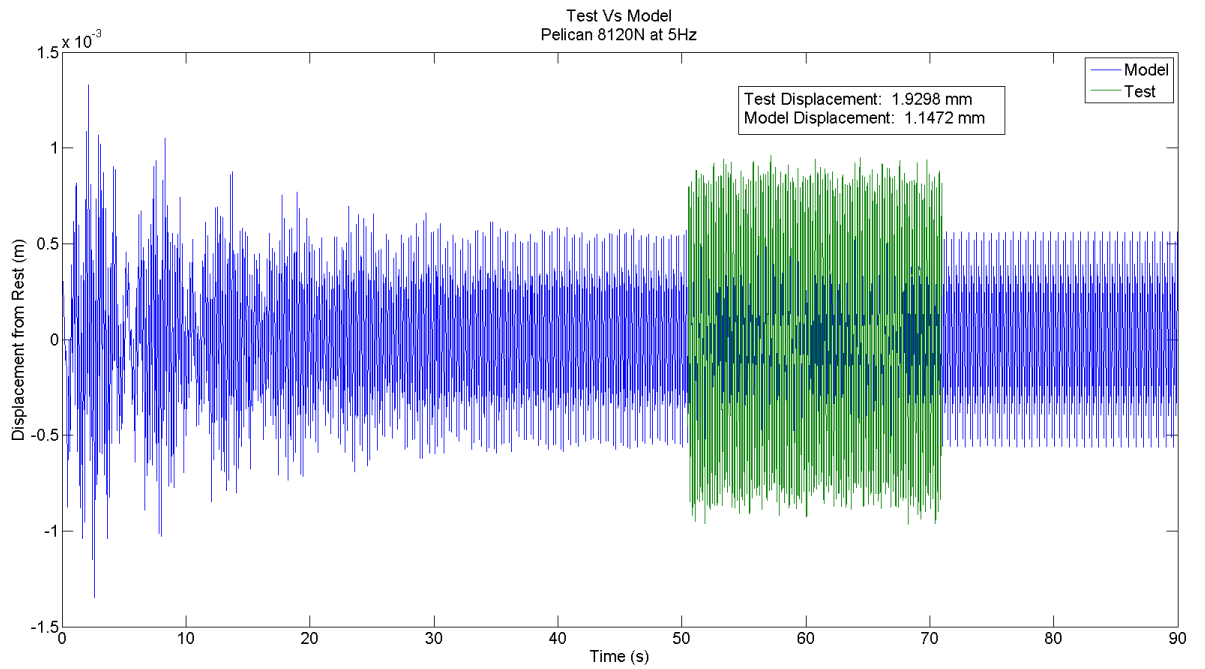
$$N_4 = \frac{x}{l}$$

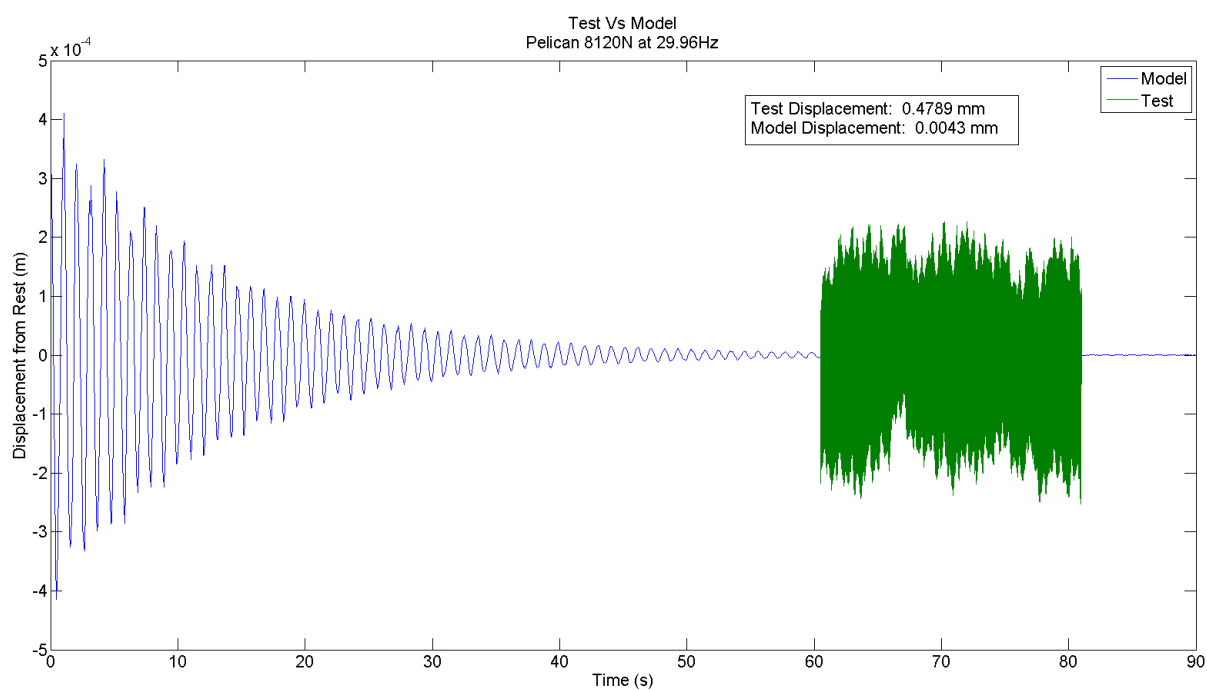
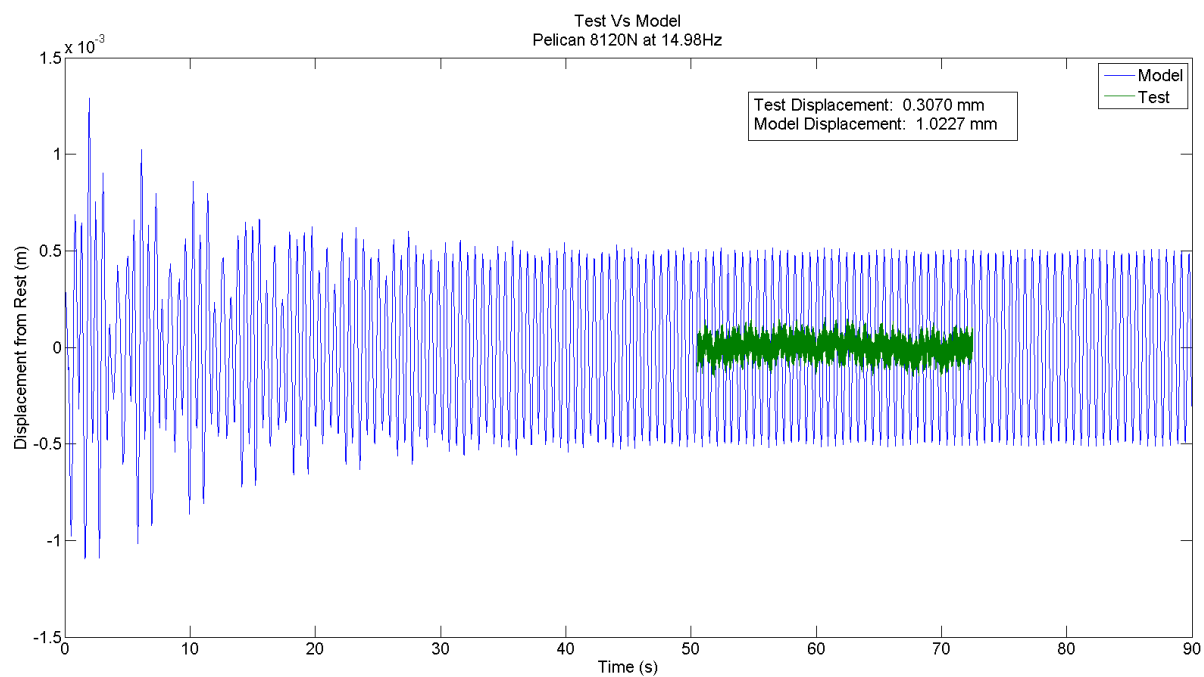
$$N_5 = \frac{3x^2}{l^2} + \frac{2x^3}{l^3}$$

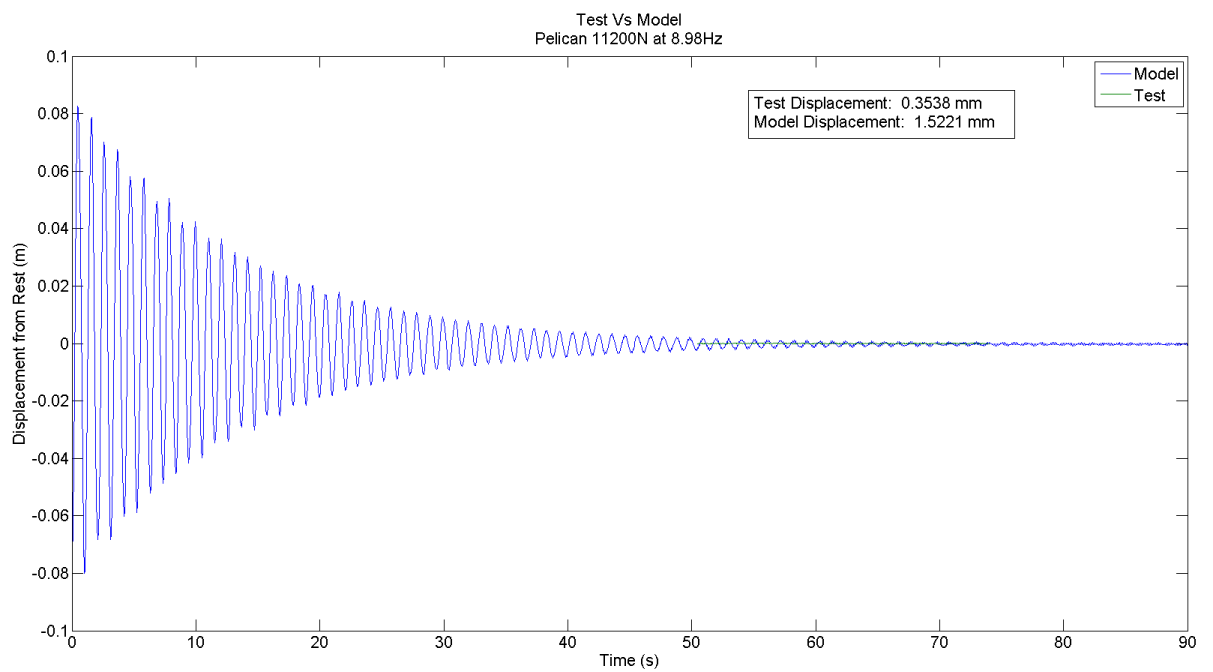
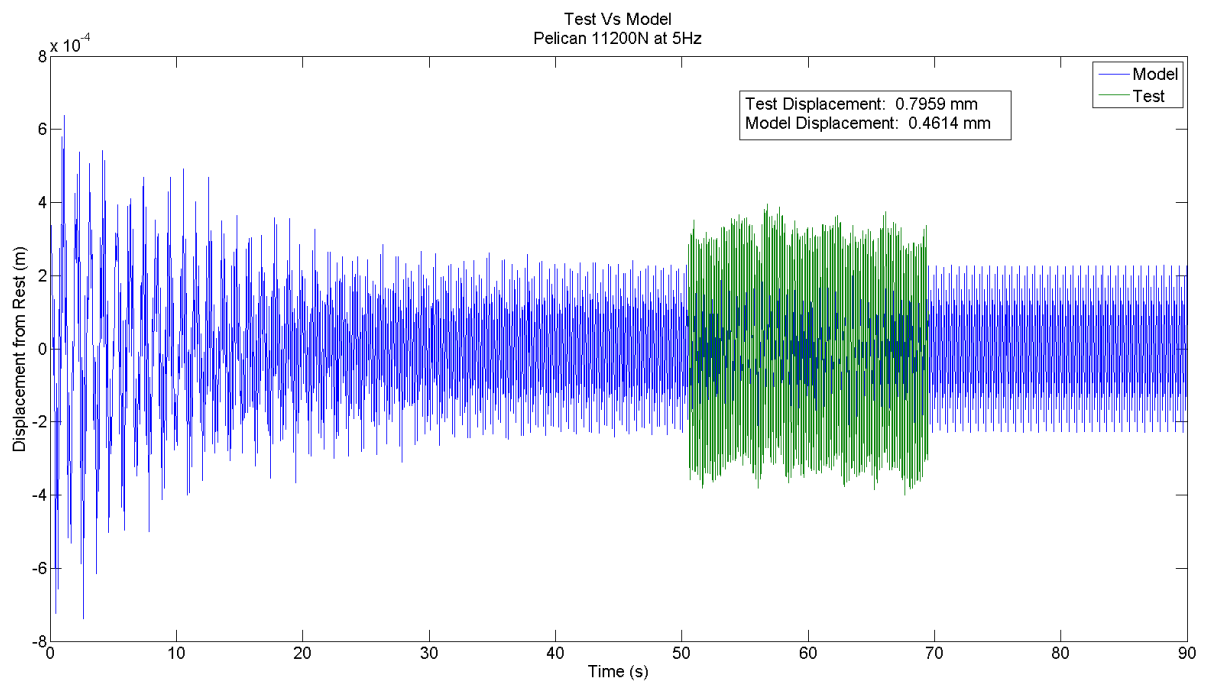
$$N_6 = -\frac{x^2}{l} + \frac{x^3}{l^2}$$

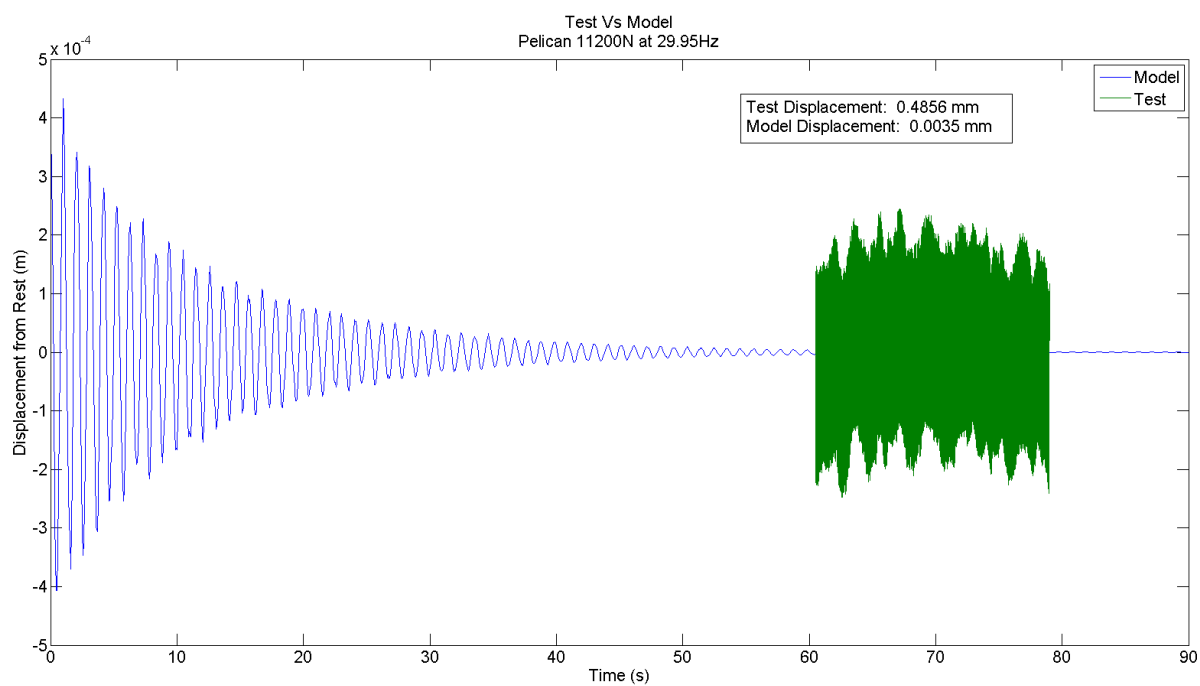
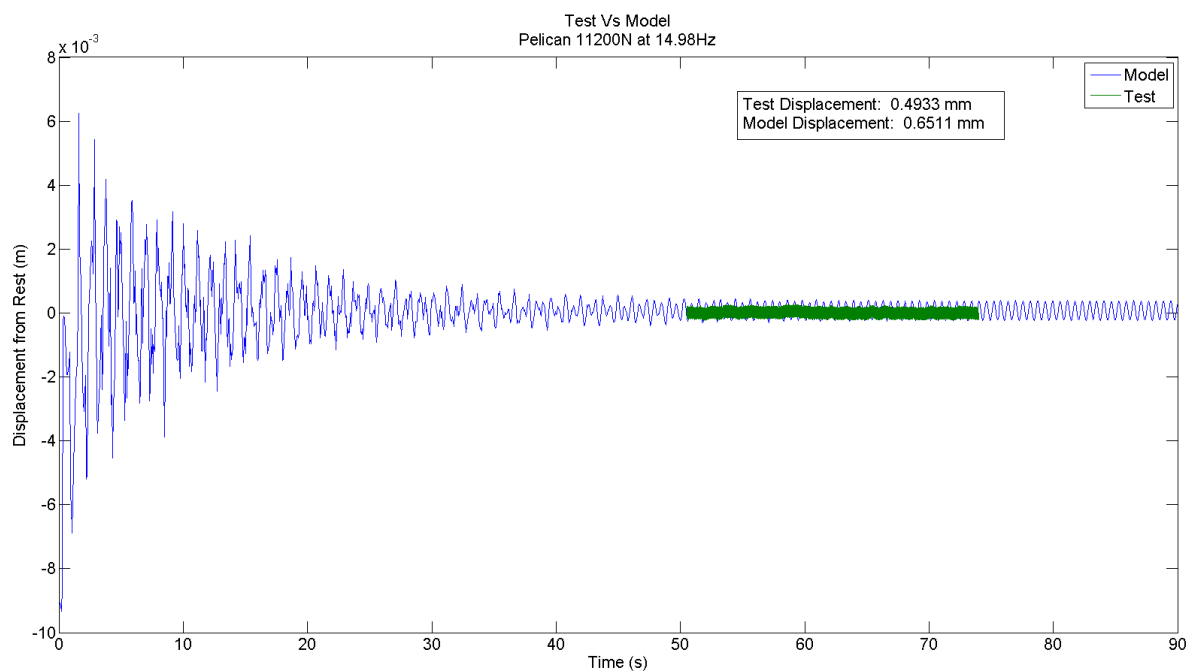
APPENDIX B

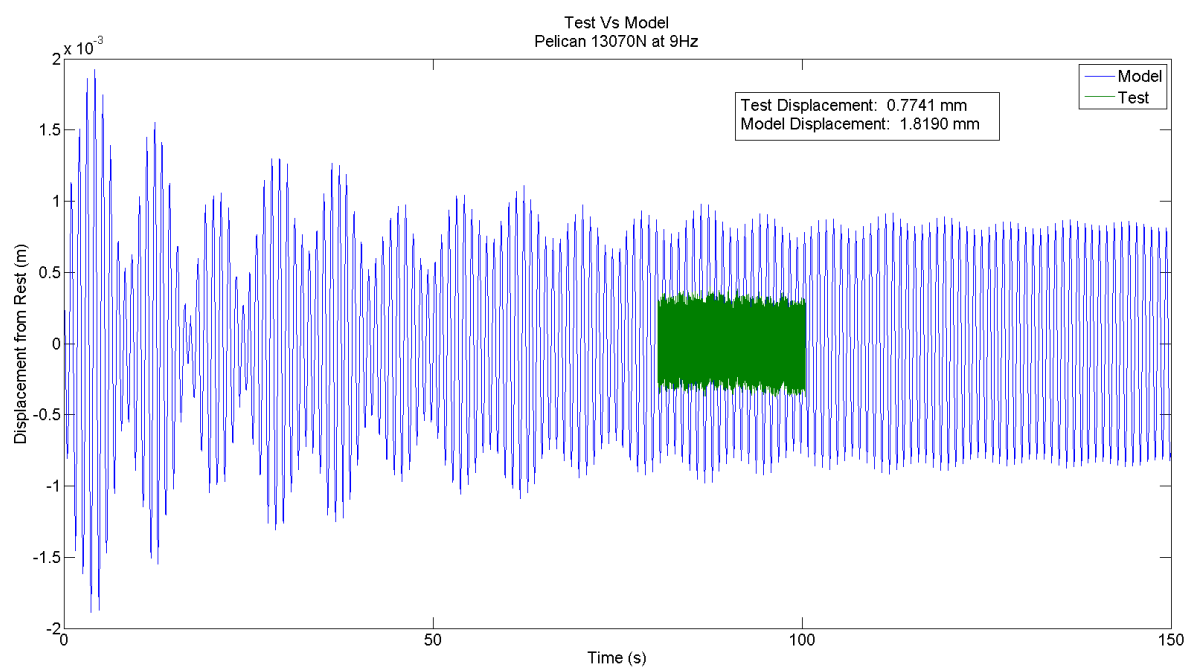
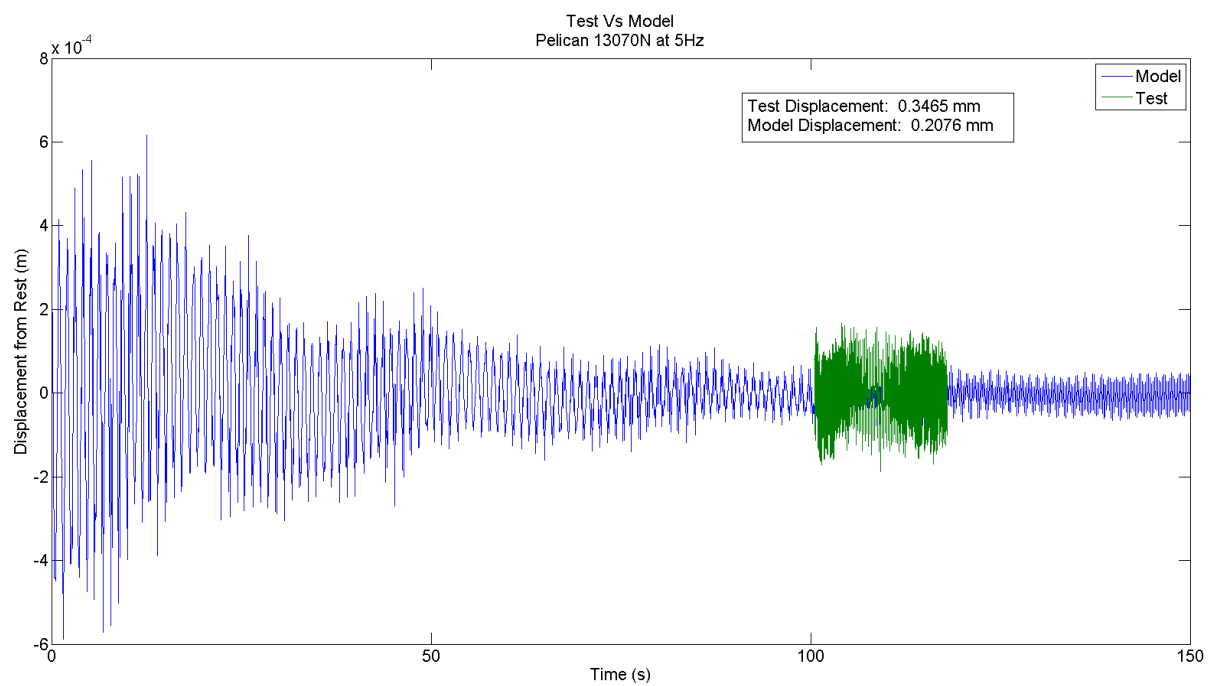
Graphs of Results

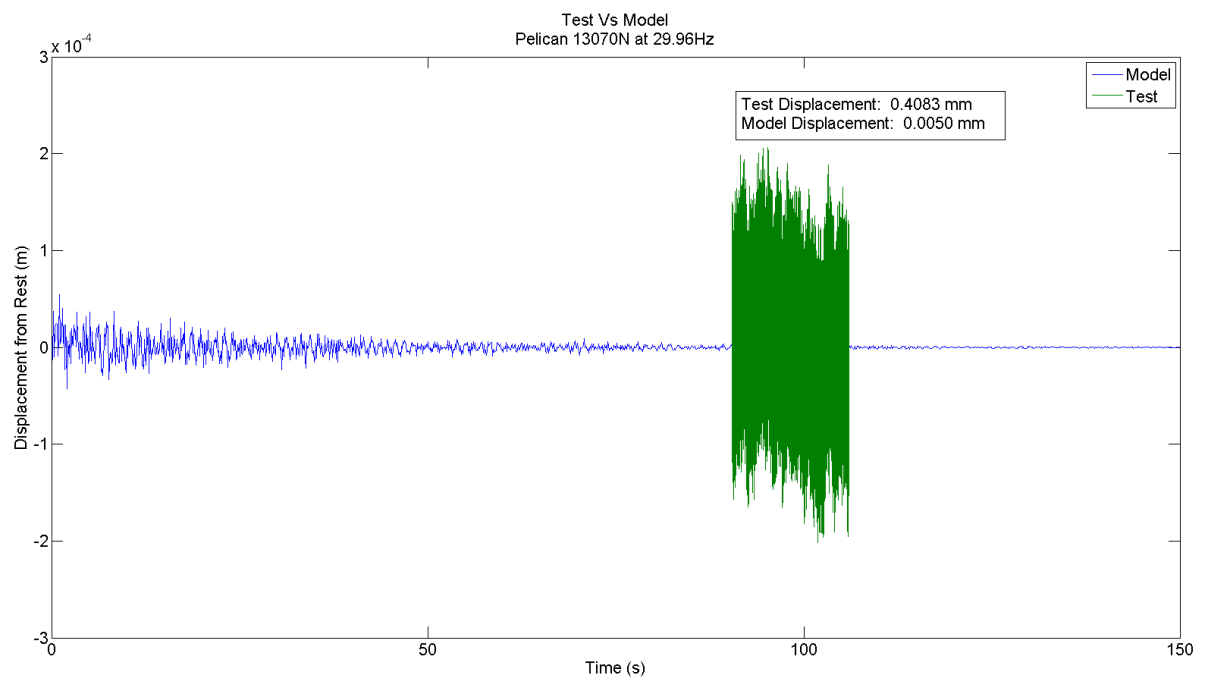
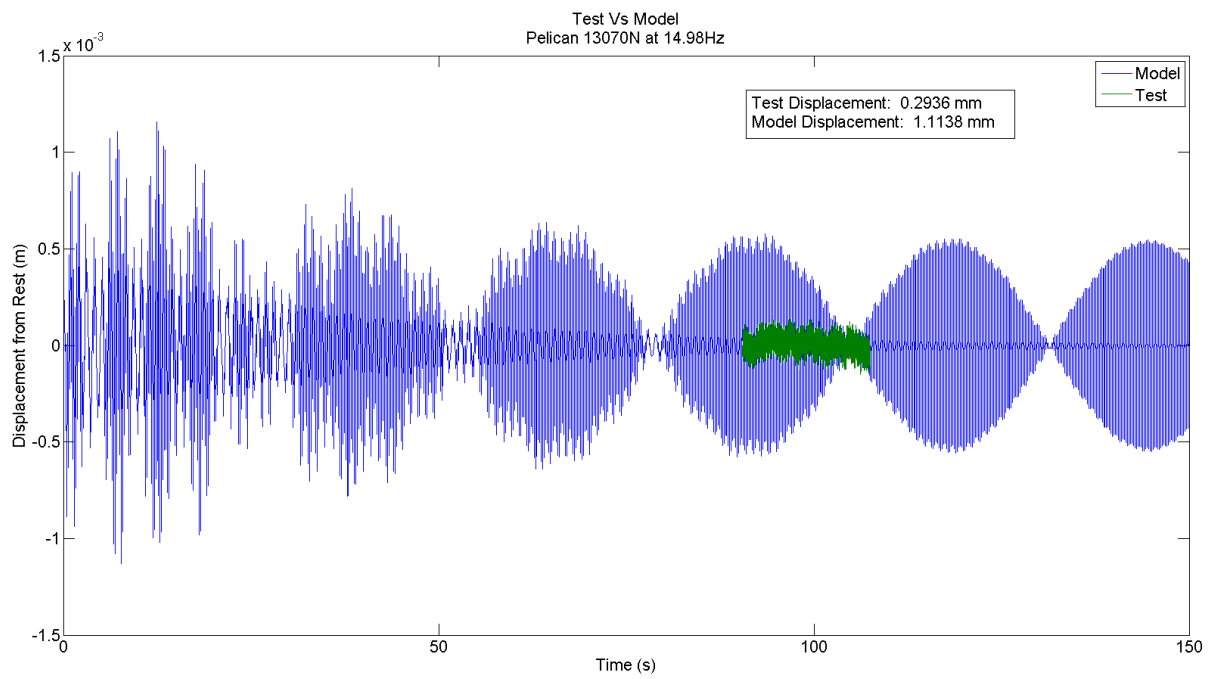


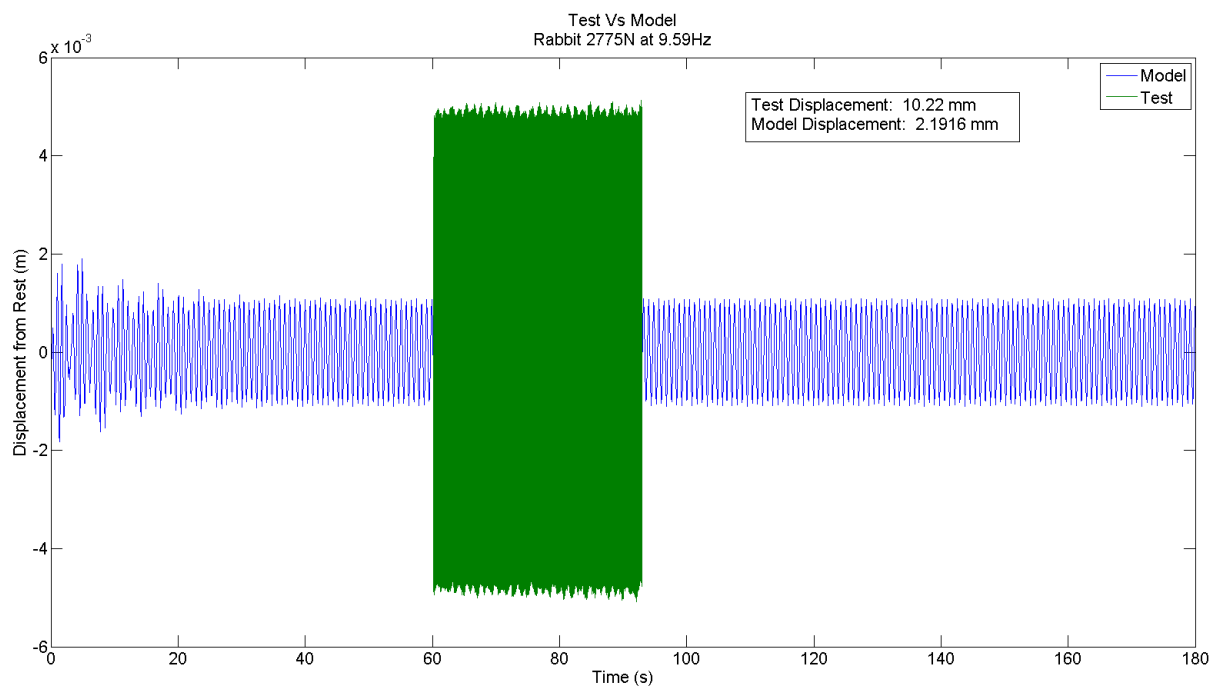
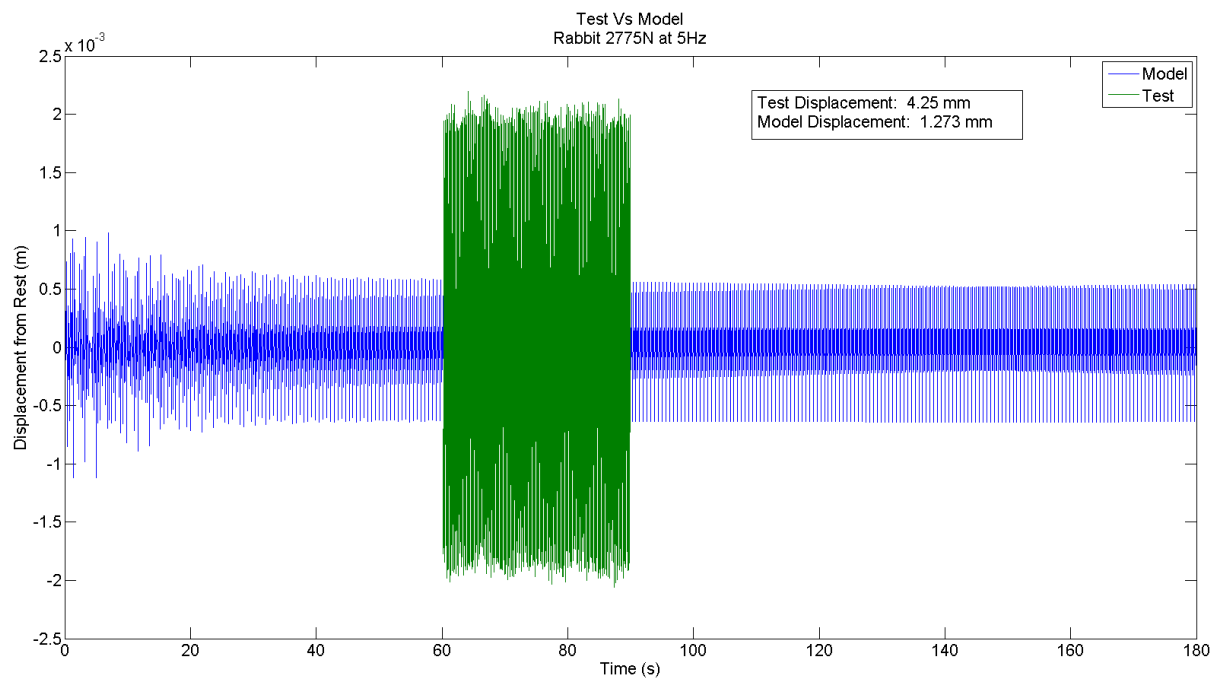


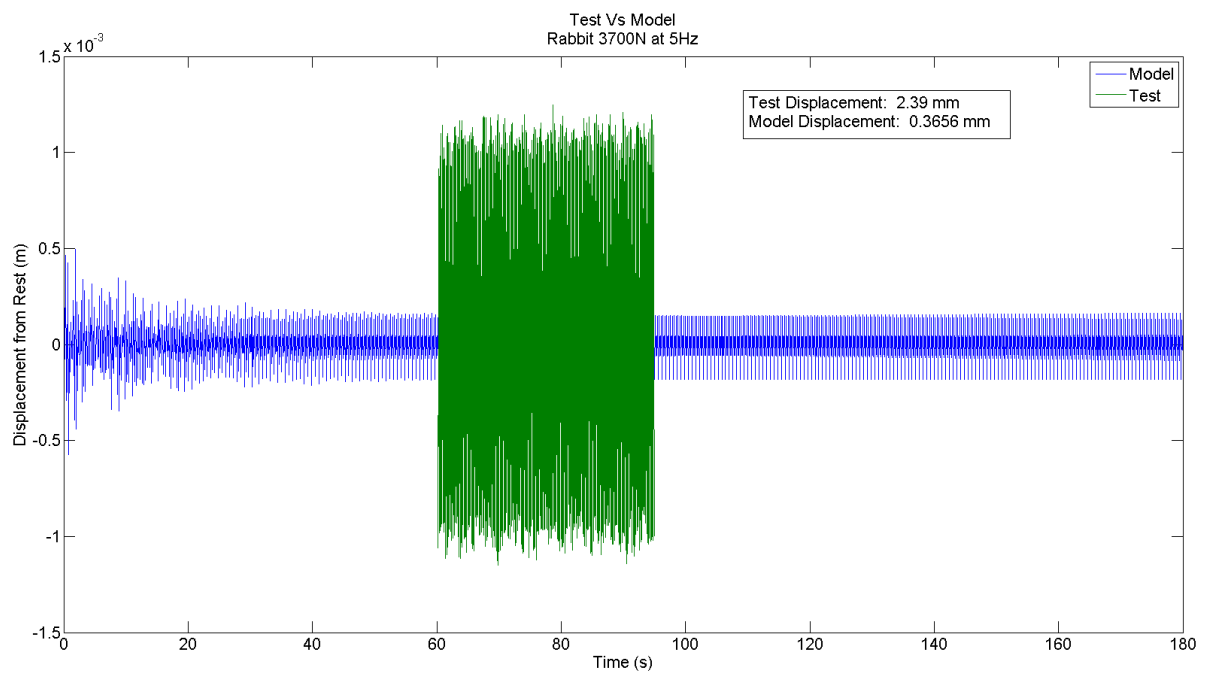
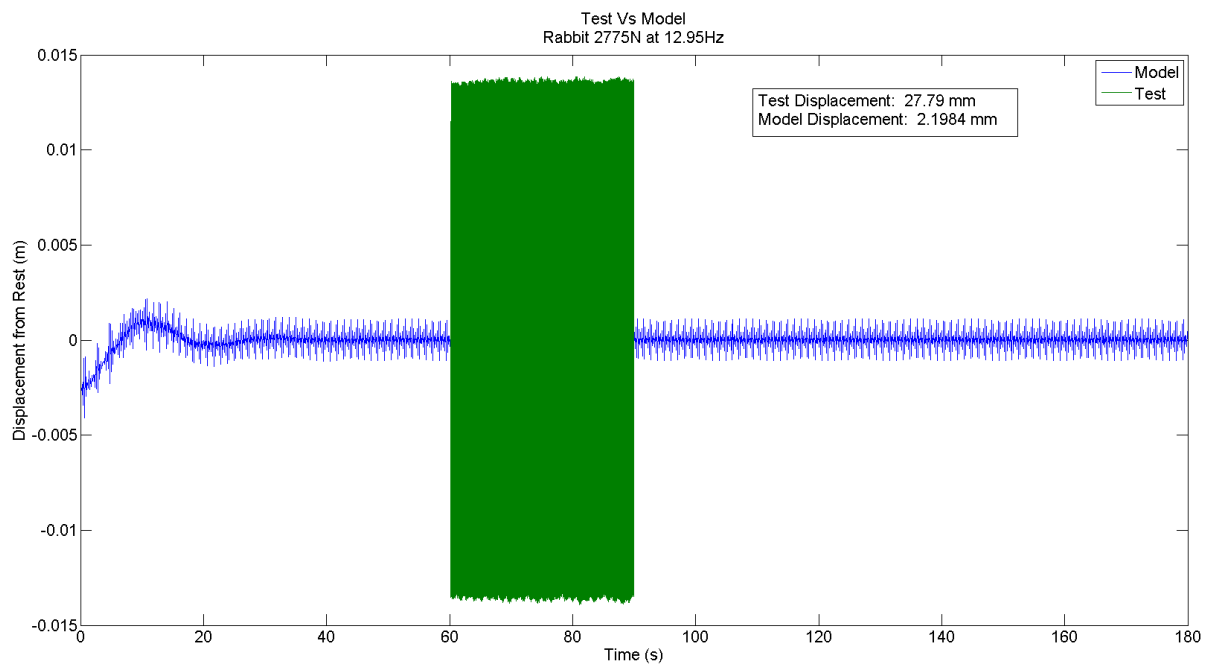


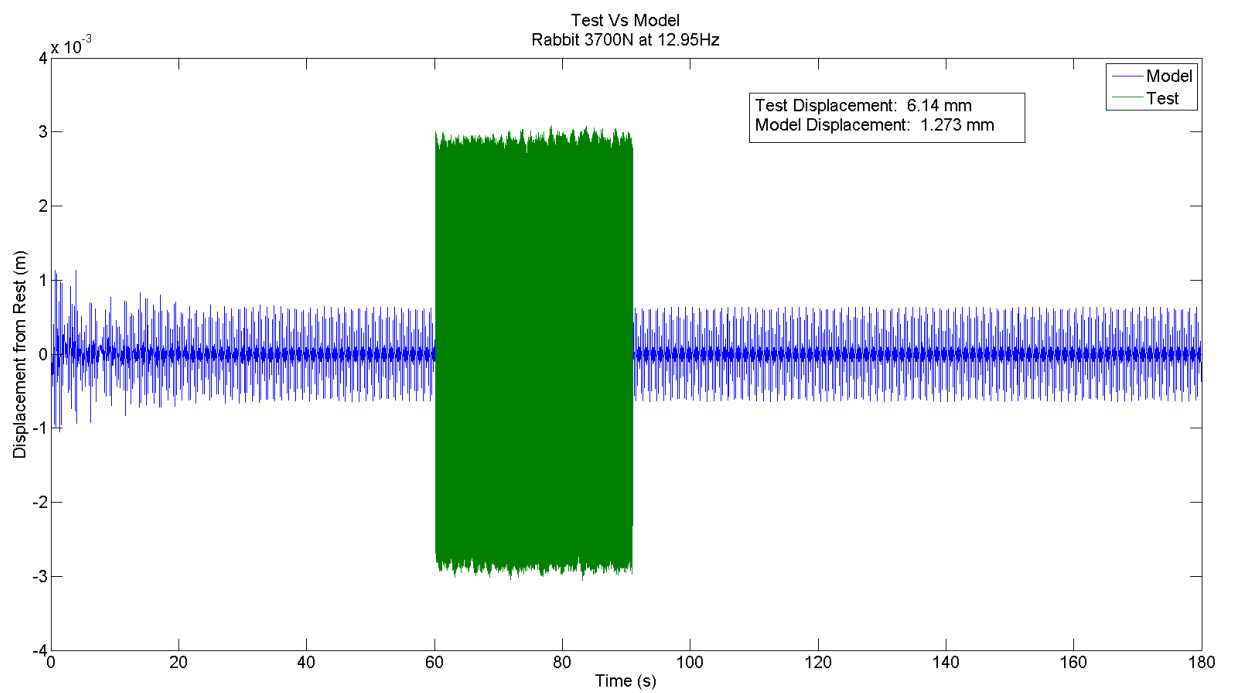
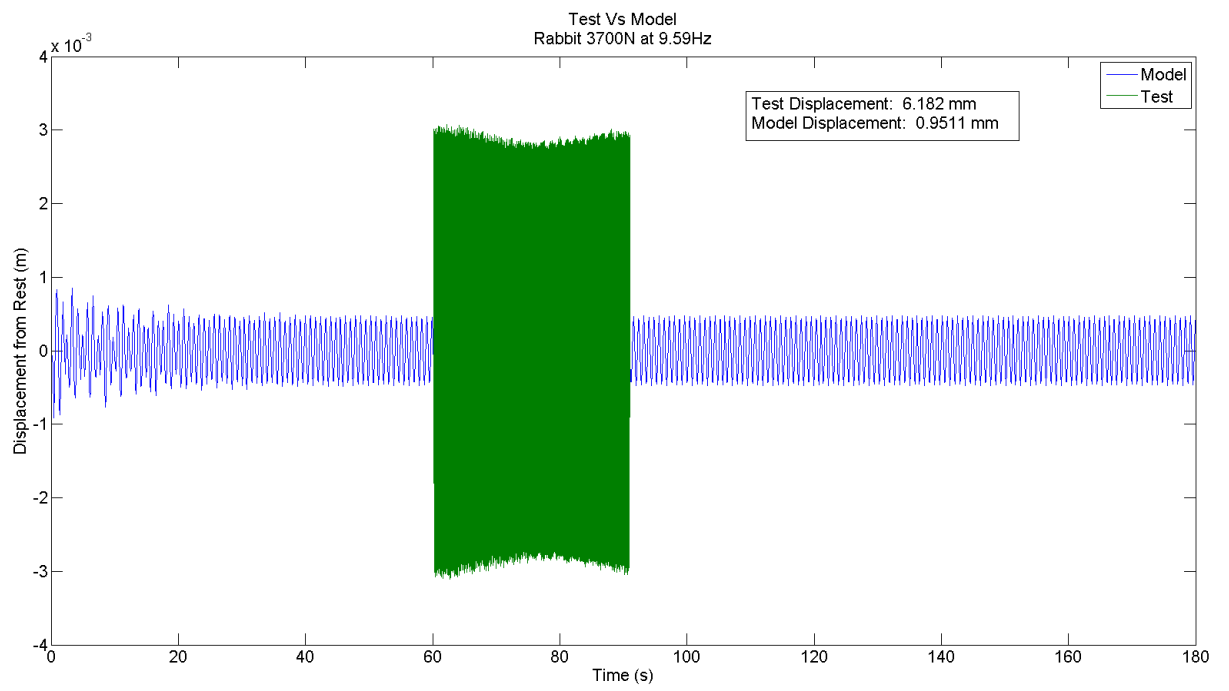


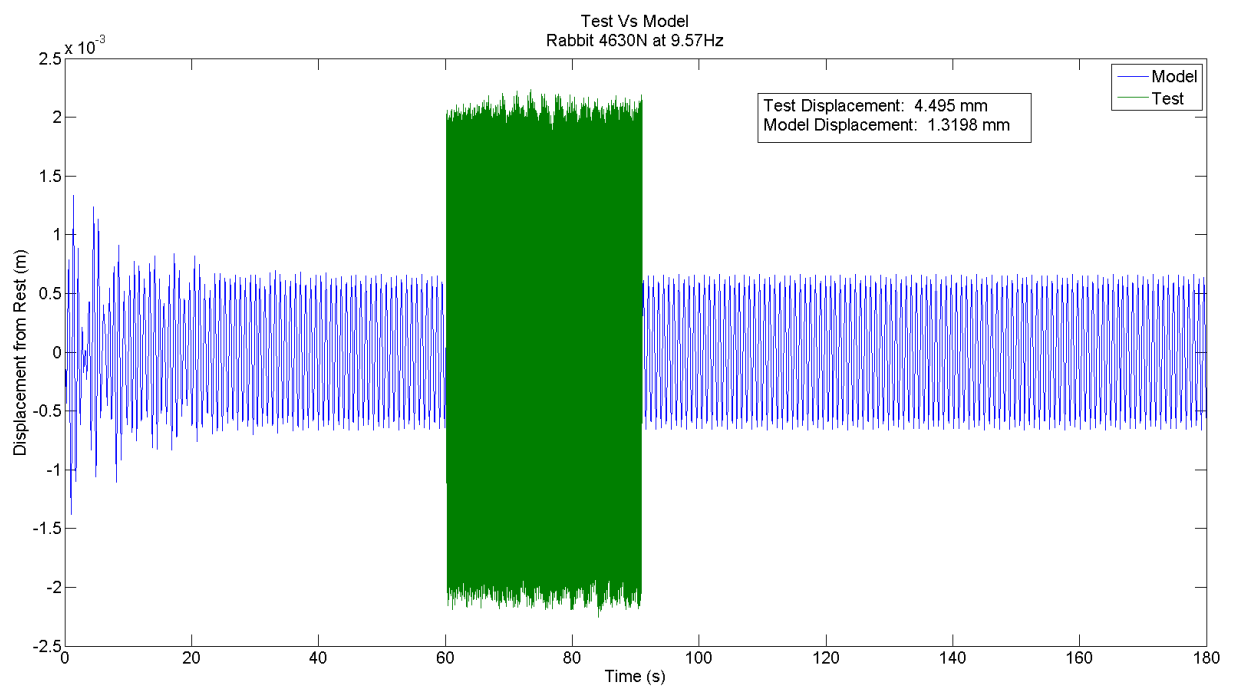
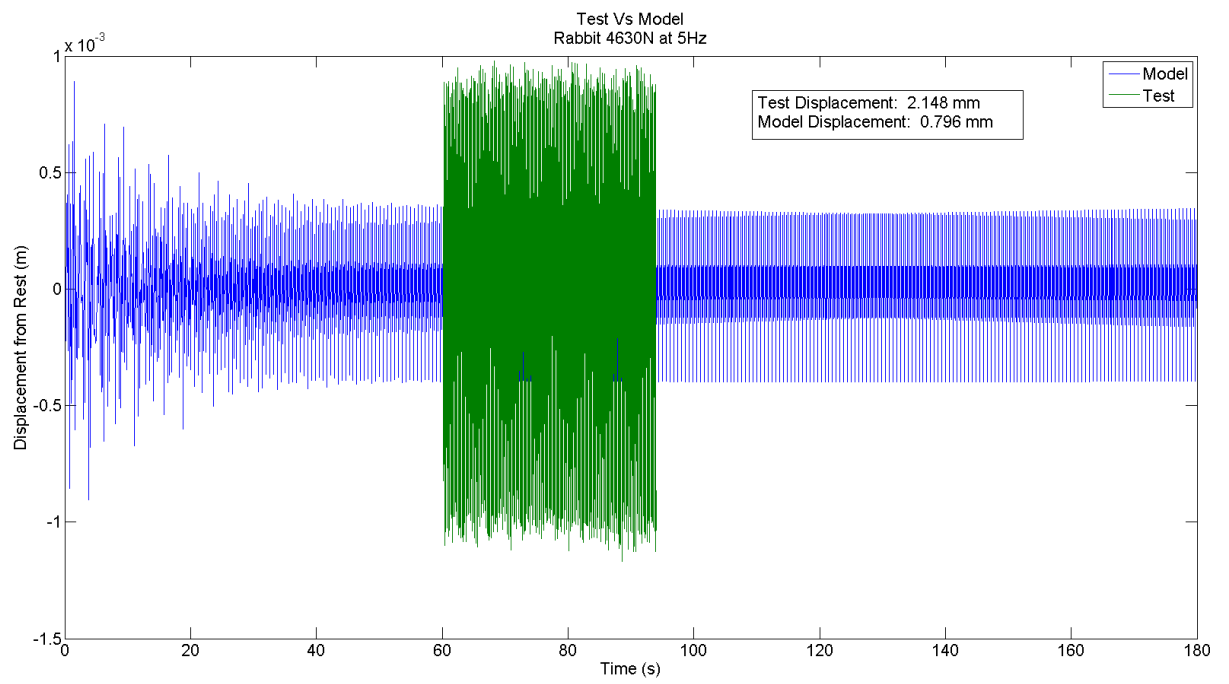


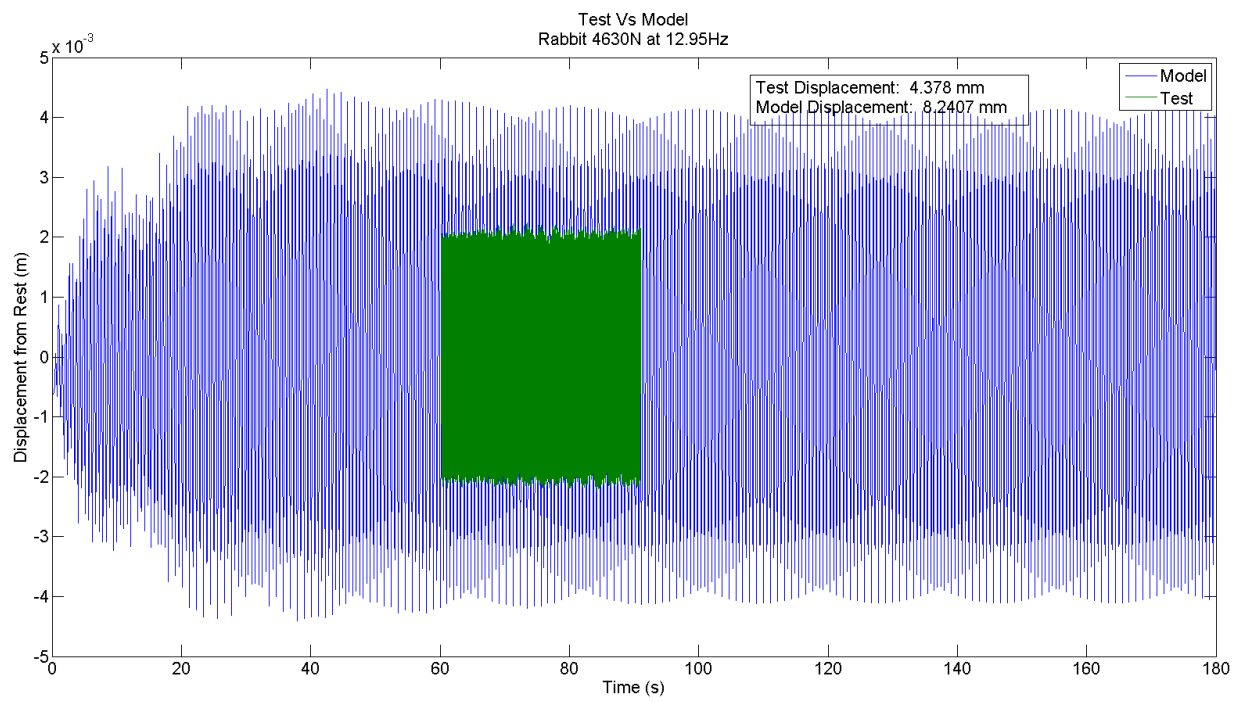












APPENDIX C

MATLAB Coding

C1: Without Shaker

```
%This Code was created by A.M. Athol-Webb
%for an MSc thesis: The Computational Investigation of the Wind-
Induced
%Vibration of Overhead Conductors
%e-mail: 204511710@ukzn.ac.za
%
%This program is a finite element model of a conductor length. It
uses the
%2nd order Lagrangian equations of motion to model the conductor
cable.
%The finite elements used are a series of planar frame elements. They
are
%2-node cubic, 3 degrees of freedom per node (2 translational, 1
rotational)
%
%The global variable 'n' represents the number of elements used to
model the
%conductor.

%Initialise Fixed Parameters

global n rho AE L EI Iter start_t Tension init_pos Om Ramp_Step Step_T
grav D C shaker_start

Om = 2*pi*1.5; %this is the starting ramp frequency, one
ramp step below starting

n=40;
t_final = 3.000; %time that simulation ends
Ramp_Step = t_final/13.63;
Step_T = 0;
cable='Rabbit.dat';
M=dlmread(cable);
rho=1310/1000;%M(9,1); %0.214; %0.775; %in kg/m, linear mass
A= M(10,1); % 0.0000617; %0.00025577; %in m^2, Cross sectional
area
E= M(8,1); %80400e+06;% 66200e+06;
D= M(11,1); %Overall Diameter, m

AE = Axial_Stiffness();
EI = Bending_Stiffness(cable);

Total_Length = 51.950; %Total length of overhead cable, in metres
I=(A^2)/(4*pi); %Derived from Izz = pi*d^4/64 (the second moment
of area of a circle)
Tension = 16481; %This represents the external tension as set by
the clamping method
```

```

EI=0.514e+03;
AE=0.00037414*48800e+06;
% AE=A*E;
% % EI= E*I;
grav = 1;

% a=KeepA;
% y_int=C;

Iter = 0; %Iteration counter for each time the ODE solver
cycles through the Lagrange Function

%Initialise variables
q = zeros(3+3*n,1);
init_pos = q;

j=1;
span=Total_Length;
height=0;
for i=0:n %even spacing of nodes
    init_pos(j) = span*(i/n);
    init_pos(j+1)= height*(i/n);
    j=j+3;
end

init_pos = Initial_pos(init_pos,span,height) %non even
spacing, see method

qdot = zeros(3+3*n,1);

%simple single point initial conditions
% q(3)= 0.1;
% q(6)= 0.2;
% qdot(17) = 100;
% qdot(47) = 1;
% qdot(30*3+2) = 1;

qddot = zeros(3+3*n,1);

L = element_length(q,init_pos)
% q = Catenary(Total_Length,a,y_int,q)

p0 = [q; qdot];
% start_v=zeros(size(start));
% p0 = [start'; start_v']
% p0=start;
% shaker_start=start(5);
% p0(14+3+3*n)=1;
% p0(17+150)= p0(17+150) + 50;

%Solve for displacement with the 2nd order equation using an ODE
solver

%t = [0 t_final]; %total timestep capturing
t = linspace(0,t_final,15000); %forced timestep capturing

```

```

in1 = input('Y/N for eigenvalue: ','s');
if in1 == 'y';
    [Omega,Phi,ModF,C] = Eigenmode(p0);
    format long eng
    disp(Omega)
    disp(ModF)
    return
end
start_t = clock;
% options =
odeset('Stats','on','InitialStep',0.000000000001,'AbsTol',1e-8);
[t,p] = ode45('lagrange', t, p0);
Q=0;
B=0;
B = dataset(p);
    export(B,'XLSfile','sim_results.xlsx');
Q = xlsread('sim_results.xlsx');

disp('Run Complete')

```

```

function EI_final = Bending_Stiffness(cable)

%This function reads it's data from a .dat file in the main directory

M=dlmread(cable);
% A0 = 8.8141*1e-6;      %Centre Core Area, m
% d=3.35e-3;
% E0 = 210000e+06;      %Centre Core Stiffness N/m2
% k=1;      %number of layers apart from single centre core
% i=[6 12];      %cables per layer
%
% A_k=[8.8141*1e-6 8.8141*1e-6];      %Area of each cable in a layer,
m
% alpha_k = [atand(pi*(3.35*3-3.35)/130) atand(pi*(20.7-(4.14*3-
4.14))/231.84)]; %Angle of wrap for a cable in a layer
% r_k = [3.35/1000 (3.35+4.14)/1000];      %radius from core to cable
in individual layer

A0 = M(1,1);      %Centre Core Area, m
d=M(2,:);
k=M(3,1);      %number of layers apart from single centre core
i=M(4,:);      %cables per layer

A_k=M(5,:);      %Area of each cable in a layer, m
alpha_k = [atand(pi*M(6,:))]; %Angle of wrap for a cable in a layer
r_k = M(7,:);      %radius from core to cable in individual layer

E=M(8,1);

% EI_min=0;
% EI_sec=0;
% AE=0;
% for c=1:k
%     EI_min=E_k(c)*i(c)*((A_k(c)^2)/(4*pi))*cosd(alpha_k(c))

```

```

%      EI_sec=E_k(c)*(i(c)/2)*A_k(c)*r_k(c)^2*(cosd(alpha_k(c)))^3
%
%
%      AE=E_k(c)*i(c)*A_k(c)*(cosd(alpha_k(c)))^3;
%
% end

% core=E0*((A0^2)/(4*pi))
% EI_min=(EI_min+E0*((A0^2)/(4*pi)))
% EI_final=(EI_min+EI_sec)
% AE=AE+A0*E0
%
% ((A0^2)/(4*pi))+2*cosd(alpha_k(1))*((A0^2)/(4*pi))+4*(cosd(alpha_k(1))
% *((A0^2)/(4*pi))+A0*(3.35e-3*sin(pi/3))^2)
I0 = (A0.^2)/(4*pi);
I1=0;
Is=0;
for c=1:k
    I1(c) = (A_k(c)^2)/(4*pi)*cosd(alpha_k(c))*i(c);
    Is(c) = i(c)/2*(cosd(alpha_k(c))^3)*A_k(c)*r_k(c)^2;
end

Total_I= sum(I1)+ sum(Is)+ I0;
% Total_I=((A0.^2)/(4*pi))+((A0.^2)/(4*pi))*cosd(alpha_k(1))*i(1) +
i(1)/2*(cosd(alpha_k(1))^3*(A0*(r_k(1))^2))
% Total_EI=Total_I*E
EI_final=Total_I*E;
% (cosd(alpha_k(1)))^3
% I=(A^2)/(4*pi);
% EI_main=E*I
% AE_main=A*E
% ((EI_main-EI_final)/EI_final)*100

function [Force,Stiff,Damp,Mass] = Boundary(Force,Stiff,Damp,Mass,r,v)
% this function assigns the boundary condition of a static axial
% co-ordinate.

global n

m = 3+n*3;
j=1;
for k=1:2 %This double for loop creates the boundary
condition at the ends for zero axial movement
    for i=1:m
        Force(i) = Force(i) - Stiff(i,j)*r(j);% - Damp(i,j)*v(j);
        Stiff(j,i)=0;
        Stiff(i,j)=0;
        Stiff(j,j)=1;

        Damp(j,i)=0;
        Damp(i,j)=0;
        Damp(j,j)=1;

        Mass(j,i)=0;
        Mass(i,j)=0;
        Mass(j,j)=1;
        Force(j)=r(j);
    end
end

```

```

        end
        j=m-2;
    end

    j=2;
    for k=1:2 %This creates the end boundary conditions of no
    movement in the Y direction
        for i=1:m
            Force(i) = Force(i) - Stiff(i,j)*r(j);% - Damp(i,j)*v(j);
            Stiff(j,i)=0;
            Stiff(i,j)=0;
            Stiff(j,j)=1;

            Damp(j,i)=0;
            Damp(i,j)=0;
            Damp(j,j)=1;

            Mass(j,i)=0;
            Mass(i,j)=0;
            Mass(j,j)=1;
            Force(j)=r(j);
        end
        j=m-1;
    end

    j=3;
    for k=1:2 %This creates the end boundary conditions of no
    movement in the theta direction
        for i=1:m
            Force(i) = Force(i) - Stiff(i,j)*r(j);% - Damp(i,j)*v(j);
            Stiff(j,i)=0;
            Stiff(i,j)=0;
            Stiff(j,j)=1;

            Damp(j,i)=0;
            Damp(i,j)=0;
            Damp(j,j)=1;

            Mass(j,i)=0;
            Mass(i,j)=0;
            Mass(j,j)=1;
            Force(j)=r(j);
        end
        j=m;
    end

    % j=5; %This creates the end boundary conditions of no
    movement in any direction for a driver
    % for k=1:1
    % for i=1:m
    % Force(i) = Force(i) - Stiff(i,j)*r(j);% -
    Damp(i,j)*v(j);
    % Stiff(j,i)=0;
    % Stiff(i,j)=0;
    % Stiff(j,j)=1;
    %
    % Damp(j,i)=0;
    % Damp(i,j)=0;
    % Damp(j,j)=1;

```

```

%
%           Mass(j,i)=0;
%           Mass(i,j)=0;
%           Mass(j,j)=1;
%           Force(j)=r(j);
%       end
%       j=j+1;
%   end

```

end

```

% function [q] = Catenary(Total,a,y_int,q)
% %Catenary This function returns the co-ords for a cable as a
catenary
% %   Sag is a positive number
%
% global n L
%
% j=1;
% Current_Length=0;
% for i=1:n
%     Current_Length=Current_Length+L(i);
%     x(i+1)=q(j)+Current_Length;
%     j=j+3;
% end
% x(1)=0;
% x
% j=2;
% for i=1:n+1
%     q(j) = -y_int + a*cosh((Total/2 - x(i))/a) ;
%     j=j+3;
% end
% q(2) = 0;
% q(3*n+2) = 0;
% end

```

```

Sag=0.5555;
ends = 42.3; %Must be half of actual length
x_cat=-ends:0.1:ends;
Prev = 100000;
for a=0.01:0.1:10000
    y_cat = a*cosh(x_cat/a);
    Drop = y_cat(1)-y_cat(length(x_cat)/2+0.5);
    Err = Sag - Drop;
    if abs(Err)<0.001 && (Err<Prev)
        KeepA = a;
        Prev = KeepA;
    end
    Err
end

```

```

C = KeepA+Sag
y_cat= KeepA*cosh(x_cat/KeepA)-C;

```

```
plot(x_cat,y_cat)
```

```
KeepA  
C
```

```
function [Omega,Phi,ModF,C] = Eigenmode(p)  
%This function calculates the undamped eigenmodes of the  
% cable model. It returns the natural frequency and eigenvector  
  
global n L  
  
s = 3 + 3*n; %s is the size of the state vectors  
  
pdot = zeros(size(p)); %initialise pdot  
r = zeros(s,1); %r is the position vector, to be used  
in calculating  
v = r; %the various state matrices  
a = r;  
  
t=0;  
  
for i=1:s %populate local position vector  
    r(i) = p(i);  
end  
  
for i=1:s %populate local and global velocity  
vector  
    j=s+i;  
    v(i) = p(j);  
  
    pdot(i) = p(j);  
end  
  
% L = element_length(r);  
  
Beta = zeros(n,1);  
Beta = transformation_angle_beta(r);  
  
%Calculate initial Matrices  
M = zeros(s);  
M = globalmassmatrix(Beta);  
  
K = zeros(s);  
K = globalstiffnessmatrix(Beta);  
  
F = zeros(size(r));  
F = globalforcevector(Beta,t);  
  
Damp = zeros(s);
```



```

[F,K,~,M] = Boundary(F,K,Damp,M,r);

[V,D] = eig(K,M);

Phi = V;
Al2 = D^2;
Phi_t = transpose(Phi);
Phi_c = conj(Phi);
Al2_c = conj(Al2);
Phi_ct = ctranpose(Phi);

C = M*(Phi*Al2*Phi_t + Phi_c*Al2_c*Phi_ct)*M;
C=0.003598*K;

[lamda,k] = sort(diag(D));
V=V(:,k);
Factor = diag(V'*M*V);
Phi = V*inv(sqrt(diag(Factor)));
Omega = diag(sqrt(Phi'*K*Phi));
ModF = Phi'*F;
W=[ModF Omega]
end

function L = element_length(q,init)
%this returns the linear element length as a function of it's position
in 3
%space

global n

L=zeros(n,1);
q=q+init;
j=1;
k=2;
for i=1:(n)
    L(i) = sqrt((q(j+3)-q(j))^2 + (q(k+3)-q(k))^2);
    k=k+3;
    j=j+3;
end

end

function Damp = globaldampingmatrix(Beta,Stiff,Mass)
%This function returns the global damping matrix
global n C

```

```

m=3+n*3;           %this represents the final size of the matrix i.e. m x
m

local_damp=zeros(6);
gl_matrix = zeros(3+3*n);
%
for i=1:6
    local_damp(i,i) = 10.0;%0.00155;           %this is the local damping
matrix, pretty simple
end

l=0;               %counter to shift starting position of next local
matrix
for k=1:n
    T = transformation(Beta(k));               %calculates the
local transformation matrix
    local_damp = T'*local_damp*T;               %transformed matrix

    gl_matrix = globalmatrix(gl_matrix,local_damp,l);
    l=l+3;
end
%
%
%
Damp = gl_matrix;
% Damp = C;

% Damp = (0.15*Mass + 0.00002*Stiff);    %Low Tension z=0.0135
% Damp = (0.05*Mass + 0.00006*Stiff);    %High Tension z=0.0045
end

function Force = globalforcevector(Beta,t)
%This is a user defined function that returns the vector of non-
conserved external
%forces in the lagrange equation

global n Tension rho Om Ramp_Step Step_T grav L D

Force = zeros(3+3*n,1);
m= 3+3*n;

% Beta = zeros(n,1);
% Beta = transformation_angle_beta(r);

j=4;
for i=1:n-1
    Force(j) = -(Tension*cos(Beta(i))-Tension*cos(Beta(i+1)));
    Force(j+1) = -(Tension*sin(Beta(i))-Tension*sin(Beta(i+1)));
    j=j+3;
end
end

```

```

%This is the external gravity force, acting on each node in the -
Y(global)
%direction

if grav == true
    i=2:3:m;
    Force(i) = Force(i) - rho*84.6*9.81/(n-1);
end

%This is the distributed Wind Force input
Re=1000;
air_rho=1.1839; %kg/m^3 at 25Celsius
mu=1.861597e-5;
V=(Re*mu)/(air_rho*D);
Fs=0.27; %reduced frequency
fs = (Fs*V)/(2*D);
z=0.0135;
phi=atan(2*z);
Cl=0.4;
Cy = Cl*(sin(2*pi*fs*t+phi));
i=2:3:m;
Force(i) = Force(i) + (0.5*Cy*air_rho*D*sum(L)*V^2)/(n-1);
end

function Mass = globalmassmatrix(Beta)
%globalmassmatrix Computes the mass global mass matrix as a function
of Beta, for
% the co-ord transformation

global n L
local_mass=zeros(6); %This is local mass matrix, to be used in
making the global

gl_matrix = zeros(3+3*n);
T = zeros(6);

%populate global matrix with local elements

l=0; %counter to shift starting position of next local
matrix
for k=1:n
    local_mass = local_massmatrixlinear(L(k));

    T = transformation(Beta(k));
    local_mass = (T'*local_mass*T);

    gl_matrix = globalmatrix(gl_matrix,local_mass,l);
    l=l+3;
end

Mass = gl_matrix;
end

```

```

function gl_matrix = globalmatrix(gl_matrix,local_matrix,l)
%This function takes in a local matrix and adds them together
% to create the global matrix. Only need to take in 1 local matrix
as the
% the elements are all identical

for i=1:6
    for j=1:6
        gl_matrix(l+i,l+j) = gl_matrix(l+i,l+j) + local_matrix(i,j);
    end
end

end

```

```

function Stiff = globalstiffnessmatrix(Beta,r,v)
%This function calculates the global stiffness matrix
% using the transformation angle and other various parameters

local_stiff=zeros(6); %This is local mass matrix, to be used in
making the global
global n L

gl_matrix = zeros(3+3*n);

%populate global matrix with local elements
l=0; %counter to shift starting position of next local
matrix
for k=1:n
    local_stiff = local_stiffnesslinear(L(k)) +
local_stiffness_nonlinear(L(k));
    T = transformation(Beta(k));
    local_stiff = T'*local_stiff*T; %transformed matrix
    gl_matrix = globalmatrix(gl_matrix,local_stiff,l);
    l=l+3;
end
m=3+n*3;
Stiff = gl_matrix;
end

```

```

function init_pos = Initial_pos(init_pos,span,height)
%Initial_pos This returns the non-even spacing for the
% initial position of the conductor

```

```

global n

```

```

sfactor = input('Scaling Factor:');
if isempty(sfactor)
    sfactor =1;
end

scale = linspace(sfactor,1,ceil(n/2));
L=sqrt((span^2+height^2)/n);

```

```

j=3*n-2;
for i=1:(n/2)
    init_pos(3*i+1) = init_pos(3*i+1)*scale(i);
    init_pos(3*i+2) = init_pos(3*i+2)*scale(i);
    init_pos(j) = span - init_pos(3*i+1);
    init_pos(j+1) = height - init_pos(3*i+2);
    j=j-3;
end
Beta = transformation_angle_beta(init_pos);

j=3;
for i=1:n+1
    init_pos(j) = Beta(1);
    j=j+3;
end

function pdot = lagrange(t,p)
%this function returns the reduced order vector pdot, which contains
the
%velocity and acceleration vectors

global n Iter L shaker_start

clc
s = 3 + 3*n; %s is the size of the state vectors

Iter = Iter + 1;
disp(['Iteration Count: ' num2str(Iter)])

pdot = zeros(size(p)); %initialise pdot
r = zeros(s,1); %r is the position vector, to be used
in calculating
v = r; %the various state matrices
a = r;

i=1:s; %populate local position vector
r(i) = p(i);

i=1:s; %populate local and global velocity vector
j=s+i;
v(i) = p(j);
pdot(i) = p(j);

Beta = zeros(n,1);
Beta = transformation_angle_beta(r);

angY = Beta;

```

```

i=1:n; % this loop converts the plane angle into
degrees for progress tracking
angY(i) = (angY(i)/pi)*180;

% disp(angY)

%Calculate initial Matrices
Force = globalforcevector(Beta,t);

Mass = zeros(s);
Mass = globalmassmatrix(Beta);

Stiff = zeros(s);
Stiff = globalstiffnessmatrix(Beta,r,v);

Damp = zeros(s);
Damp = globaldampingmatrix(Beta,Stiff,Mass);

% Stiff(10,10)
% Stiff(13,13)

% Force = zeros(size(r));

% shaker_amp=0.001/2;
% node=5;
% w0=2*pi*(5); %The driving conditions of the
shaker base node
% m=2*pi*(2/60);
% % % if t<= (pi/w)
% r(node) = shaker_start + shaker_amp*sin(w0*t + m*t^2);
% p(node) = r(node);
% v(node) = shaker_amp*(w0+2*m*t)*cos(w0*t+m*t^2);
% pdot(node) = v(node);
% p(node+s) = v(node);
% else
% r(node) = 0;
% p(node) = r(node);
% v(node) = 0;
% pdot(node) = 0;
% p(node+s) = v(node);
% end

[Force,Stiff,Damp,Mass] = Boundary(Force,Stiff,Damp,Mass,r,v);
% if t<0.005
% Force(17) = Force(17) + 1000;
% end
% Force'
%sim time
disp(['Simulation Time: ' num2str(t), ' Seconds']);
% DETERMINANT = [det(Mass) det(Damp) det(Stiff)]
% Condition = cond(Stiff)
Damp = sparse(Damp);
Stiff = sparse(Stiff);
Mass = sparse(Mass);
B = (Force - Damp*v - Stiff*r);

```

```

a = mldivide(Mass,B);
% a=tfqmr(Mass,B);

x=zeros(1,n+1);
y=x;
i=1:3:s;
x = r(i)';
i = 2:3:s;
y = r(i)';
vy = v(i)';
i=3:3:s;
theta = r(i);

% disp(x);
disp(y);
% disp(x);

i=1:s;
j=s+i; %Result Parse out
pdot(j) = a(i);

% Force = (Stiff*r)' + (Damp*v)' + (Mass*a)'
timing(); %shows real time used for simulation
end

function local_mass_lin = local_massmatrixlinear(L)
%This creates a local linear mass matrix that is transformed

global rho

local_mass = zeros(6);

F = 13/35;
G = L^2/105;
H = 11*L/210;
I = 9/70;
J = 13*L/420;
K = L^2/140;

local_mass(1,1) = 1/3;
local_mass(4,1) = 1/6;

local_mass(2,2) = F;
local_mass(3,2) = H;
local_mass(5,2) = I;
local_mass(6,2) = -J;

local_mass(2,3) = H;
local_mass(3,3) = G;
local_mass(5,3) = J;
local_mass(6,3) = -K;

local_mass(1,4) = 1/6;
local_mass(4,4) = 1/3;

```

```

local_mass(2,5) = I;
local_mass(3,5) = J;
local_mass(5,5) = F;
local_mass(6,5) = -H;

local_mass(2,6) = -J;
local_mass(3,6) = -K;
local_mass(5,6) = -H;
local_mass(6,6) = G;

local_mass_lin = (rho*L)*local_mass;

end

function local_stiff = local_stiffness_nonlinear(L)
%This creates a local stiffness matrix that is transformed

local_stiff = zeros(6);

local_stiff(2,2) = 6/5;           %Column 2
local_stiff(3,2) = L/10;
local_stiff(5,2) = -6/5;
local_stiff(6,2) = L/10;

local_stiff(2,3) = L/10;         %Column 3
local_stiff(3,3) = (2*L^2)/15;
local_stiff(5,3) = -L/10;
local_stiff(6,3) = (-L^2)/30;

local_stiff(2,5) = -6/5;         %Column 5
local_stiff(3,5) = -L/10;
local_stiff(5,5) = 6/5;
local_stiff(6,5) = -L/10;

local_stiff(2,6) = L/10;         %Column 6
local_stiff(3,6) = (-L^2)/30;
local_stiff(5,6) = -L/10;
local_stiff(6,6) = (2*L^2)/15;

local_stiff = local_stiff/L;

end

```



```

function local_stiff = local_stiffnesslinear(L)
%This creates a local stiffness matrix that is transformed

global AE EI

local_stiff = zeros(6);
F=12*(EI)/L^3;
G=4*EI/L;
P=6*EI/L^2;
H=2*EI/L;
K=AE/L;

local_stiff(1,1) = K;           %Column 1
local_stiff(4,1) = -K;

local_stiff(2,2) = F;           %Column 2
local_stiff(3,2) = P;
local_stiff(5,2) = -F;
local_stiff(6,2) = P;

local_stiff(2,3) = P;           %Column 3
local_stiff(3,3) = G;
local_stiff(5,3) = -P;
local_stiff(6,3) = H;

local_stiff(1,4) = -K;          %Column 4
local_stiff(4,4) = K;

local_stiff(2,5) = -F;          %Column 5
local_stiff(3,5) = -P;
local_stiff(5,5) = F;
local_stiff(6,5) = -P;

local_stiff(2,6) = P;           %Column 6
local_stiff(3,6) = H;
local_stiff(5,6) = -P;
local_stiff(6,6) = G;

end

%This function analyses the result of a test to determine the
logarithmic
%decrement of a result. It works best with tests with a single start
%condition. It produces the decrement and the damping factor zeta.
% clc
x = Y1;
x=abs(x);
[xj,locs] = findpeaks(x, 'MINPEAKDISTANCE',1)

xj=Y1;
locs=X1;
j=length(xj);
ln_xj = log(xj);

```

```

yj=zeros(j,1);
yj2=yj;
ln_xjyj=yj;
ln_xk=yj;

for i=1:j
    yj(i)=i-1;
    yj2(i) = yj(i)*yj(i);
    ln_xjyj(i) = ln_xj(i)*(yj(i));
end

MAT(1,1) = sum(yj2);
MAT(1,2) = sum(yj);
MAT(2,1) = MAT(1,2);
MAT(2,2) = j;

VEC(1) = sum(ln_xjyj);
VEC(2) = sum(ln_xj);

Const = MAT\VEC';

del = -Const(1)
zeta = del/(sqrt((2*pi)^2+del^2))
Quality = 1/(2*zeta)
% plot(X1(locs),xj,X,Y)
plot(X1,xj,X,Y)

w=2*pi/((X1(j)-X1(1))/(j-1))

%This File displays the shape of the conductor and
N = length(Q);

global n init_pos
X = zeros(n+1,1);
Y = X;
TH=Y;

x1 = [0 10.575 21.15 42.3 63.45 74.2 84.6]';
y1 = [0 -0.155 -0.295 -0.460 -0.295 -0.155 0]';

x2 = [0 8.4600 16.9200 25.3800 33.8400 42.3000 50.7600 59.2200 67.6800
76.1400 84.6000]';
y2 = [0 -0.1372 -0.2432 -0.3190 -0.3644 -0.3796 -0.3644 -0.3190 -
0.2432 -0.13720 0]';

x3 = [0 0.645 1.2 3.2 5.2 7.2 9.2 11.2 13.2 15.2 17.2 19.2 21.2 23.2
25.2 27.2 31.2 33.2 35.2 37.2 39.2 41.2 42.5]';
y3 = [1.139 1.125 1.110 1.080 1.050 1.010 0.975 0.952 0.920 0.894
0.865 0.845 0.823 0.806 0.785 0.775 0.756 0.745 0.735 0.722 0.712
0.715 0.717]';

x4 = [0 5.2875 10.575 21.15 31.725 42.3 (84.6-31.725) (84.6-21.15)
(84.6-10.575) (84.6-5.2875) 84.6]';
y4 = [1.129 1.026 0.936 0.787 0.702 0.682 0.702 0.787 0.936 1.026
1.129]';

```

```

y4=y4-1.129;
x5 = [0 5.2875 10.575 21.15 31.725 42.3 (84.6-31.725) (84.6-21.15)
(84.6-10.575) (84.6-5.2875) 84.6]';
y5 = [1.129 1.00 0.888 0.706 0.602 0.577 0.602 0.706 0.888 1.00
1.129]';
y5=y5-1.129;

y6 = [1.129 1.00 0.888 0.706 0.602 0.577 0.602 0.706 0.888 1.00
1.129];
y6=y6-1.129;

y7 = [1.1330 1.021 0.917 0.750 0.653 0.625 0.653 0.750 0.917 1.021
1.1330];
y7=y7-1.1330;

x6=x5;

y9 = [1.1330 1.057 0.984 0.870 0.804 0.79 0.804 0.870 0.984 1.057
1.1330];
y9 = y9-1.1330;

l = length(x3);
for i=1:l-1
    x3(l+i) = x3(l+i-1) + (x3(l-i+1)-x3(l-i));
    y3(l+i) = y3(l-i);
end
y3 = y3 - 1.139;
x3
y3
length(x3)

for i=1:5:N %this loop displays each step of the results and plots
them. This index can be sped up by skipping indices
    j=1;
    for k=1:n+1
        X(k)=Q(i,j) + init_pos(j);
        Y(k)=Q(i,j+1)+ init_pos(j+1);
        TH(k)=Q(i,j+2) + init_pos(j+2);
        j=j+3;
    end

    V=[];
    U=[];

    for k=1:n

        z=0;
        interval=(X(k+1)-X(k))/100;
        l = X(k+1)-X(k);
        for x=0:interval:l
            z=z+1;
%            l=L(k);
            N1 = (1-x/l);
            N2 = (2*(x^3)-3*l*(x^2) + l^3)/l^3;
            N3 = (x^3 - 2*l*x^2 + l^2*x)/l^2;
            N4 = (x/l);
            N5 = -(2*x^3 - 3*l*x^2)/l^3;
            N6 = (x^3 - l*x^2)/l^2;

```

```

        V_ele(z) = [N2 N3 N5 N6]*[Y(k) TH(k) Y(k+1) TH(k+1)]';
        U_ele(z) = [N1 N4]*[X(k) X(k+1)]';
    end
    V=horzcat(V,V_ele);
    U=horzcat(U,U_ele);

end
V;
U;

%     for a=1:length(V)-1
%         Length(a) = sqrt((V(a+1)-V(a))^2+(U(a+1)-U(a))^2);
%     end
%     Total=sum(Length);

%     for a=1:length(x1)-1
%         Length(a) = sqrt((x1(a+1)-x1(a))^2+(y1(a+1)-y1(a))^2);
%     end
%     Total=sum(Length);

figure(1);
plot(U,V,X,Y,x6,y5)

axis manual
axis([0 84.6 -0.07 0.001])
%     pause()

end
for a=1:length(U)-1
    Length(a) = sqrt((U(a+1)-U(a))^2+(V(a+1)-V(a))^2);
end
Total=sum(Length);
disp('Shows Over')

function timing()
%this converts the time measured to perform the calculations into a
%realtime counter for easier monitoring
%start_t is when the program first starts, with the time difference
being
%calculated each time the function is called.

global start_t

elapsed = clock - start_t;

if elapsed(6) < 0
    elapsed(6) = elapsed(6) + 60;
    elapsed(5) = elapsed(5) - 1;
end

if elapsed(5) < 0
    elapsed(5) = elapsed(5) + 60;
    elapsed(4) = elapsed(4) - 1;
end
end

```

```

if elapsed(4) < 0
    elapsed(4) = elapsed(4) + 24;
    elapsed(3) = elapsed(3) - 1;
end

if elapsed(3) < 0
    elapsed(3) = elapsed(3) + 1;
end

day = elapsed(3);
hour = elapsed(4);
min = elapsed(5);
sec = elapsed(6);

disp(' ')
disp('Simulation has been running for:')
disp([num2str(day), ' Days ', num2str(hour), ' Hours ', num2str(min), '
Minutes ', num2str(sec), 's '])
end

function T = transformation(beta)
%This function calculates the local co-ordinate transformation as a
%function of the local to global angle beta
t = [cos(beta) sin(beta) 0;
     -sin(beta) cos(beta) 0;
     0 0 1];
T = [t zeros(3); zeros(3) t];

end

function Beta = transformation_angle_beta(q)
%This calculates the angle between the element and global co-ord
system
% after each time step for the XY plane

j=5;
global n init_pos
Beta = zeros(1,n);
real_pos = init_pos + q;
for i=1:n
    m=((real_pos(j)-real_pos(j-3))/(real_pos(j-1)-real_pos(j-4)));
    Beta(i) = atan(m);
    j=j+3;
end
end

```

C2: With Shaker

```
%This Code was created by A.M. Athol-Webb
%for an MSc thesis: The Computational Investigation of the Wind-
Induced
%Vibration of Overhead Conductors
%e-mail: 204511710@ukzn.ac.za
%
%This program is a finite element model of a conductor length. It
uses the
%2nd order Lagrangian equations of motion to model the conductor
cable.
%The finite elements used are a series of planar frame elements. They
are
%2-node cubic, 3 degrees of freedom per node (2 translational, 1
rotational)
%
%The global variable 'n' represents the number of elements used to
model the
%conductor.

%This version includes the shaker, which changes the size of the
matrix

%Initialise Fixed Parameters

global n rho AE L EI Iter start_t Tension init_pos Om step_t next_t
grav init_pos zz countstep shaker_start rs vs t_interp shaker_amp freq
drivemethod

Om = 450; %this is the starting ramp frequency, one ramp
step below starting

n=40;
t_final = 600 %time that simulation ends
step_t=0.5952;
next_t=step_t;
countstep=1;
cable='Pelican.dat';
M=dlmread(cable);
rho=1310/1000;%M(9,1); %0.214; %0.775; %in kg/m, linear mass
A= M(10,1); % 0.0000617; %0.00025577; %in m^2, Cross sectional
area
E= M(8,1); %80400e+06;% 66200e+06;
D= M(11,1); %Overall Diameter, m

AE = Axial_Stiffness();
EI = Bending_Stiffness(cable);

Total_Length = 51.950; %Total length of overhead cable, in metres
I=(A^2)/(4*pi); %Derived from Izz = pi*d^4/64 (the second moment
of area of a circle)
Tension = 21778; %This represents the external tension as set by
the clamping method

EI=0.7415e+03;
```

```

AE=0.00037414*48800e+06;
% AE=A*E;
% % EI= E*I;

grav = 1;

a=386.0100%2362;%8991.414;
y_int=416.4900%2362+0.3796;%8991.514;

Iter = 0; %Iteration counter for each time the ODE solver
cycles through the Lagrange Function

%Initialise variables
q = zeros(4+3*n,1);
init_pos = q;

j=1;
span=Total_Length;
height=0;
for i=0:n %even spacing of nodes
    init_pos(j) = span*(i/n);
    init_pos(j+1)= height*(i/n);
    j=j+3;
end

init_pos = Initial_pos(init_pos,span,height) %non even
spacing, see method

qdot = zeros(4+3*n,1);

%simple single point initial conditions
% q(92)= q(92) + 0.1;
% qdot(30*3+2)= 1;

qddot = zeros(4+3*n,1);

% q = Catenary(Total_Length,a,y_int,q)
L = element_length(q,init_pos)

% p0(92+184)=1;

% p0(17+150)= p0(17+150) + 50;

%Solve for displacement with the 2nd order equation using an ODE
solver

%t = [0 t_final]; %total timestep capturing
step=0.02;
t = linspace(0,t_final,15000); %forced timestep capturing

drivemethod='logarithmic'; %steady, logarithmic or linear
shaker_amp=0.0015/2;
f_start=5;
f_end=(40/600)*t_final + f_start;
t_interp=[t(1);t(end);length(t)*10];

```

```

switch drivemethod
    case {'steady'}
        freq=29.96;    %hz
        % vs(1)=2*pi*freq*shaker_amp;
        vs(1)=0;
    case {'logarithmic'}

[rs,vs,fr]=shaker_drive(t,f_start,f_end,shaker_amp,'logarithmic');
    case {'linear'}
        [rs,vs,fr]=shaker_drive(t,f_start,f_end,shaker_amp,'linear');
end

p0 = [q; qdot];
start_v=start*0;
start_v(end)=vs(1);
p0 = [start'; start_v']
shaker_start=start(end);
pause()

in1 = input('Y/N for eigenvalue: ','s');
if in1 == 'y';
    [Omega,Phi,ModF] = Eigenmode(p0,span);
    format long eng
    disp(Omega)
    disp(ModF)
    return
end
start_t = clock;
% options =
odeset('Stats','on','InitialStep',0.000000000001,'AbsTol',1e-8);
[t,p] = ode45('lagrange', t, p0);
Q=0;
B=0;
B = dataset(p);
    export(B,'XLSfile','sim_results.xlsx');
Q = xlsread('sim_results.xlsx');

disp('Run Complete')

%Be Careful of raw data units!  Acceleration
%import test data as "data"
%xlsread sim data as "Q"
[b,a]=butter(5,[0.001],'high');

data2=data(11:end,:);
accd=data2(:,2)*9.81; %Converting g to m/s^2
dt=5e-4;
acc=filter(b,a,accd);
vel=cumsum(acc)*dt;
pos=cumsum(vel)*dt;
figure(1)
plot(data2(:,1),pos)
figure(2)
plot(data2(:,1),accd,data2(:,1),acc)

```



```

A=max(pos(10000:end));
B=min(pos(10000:end));
Disp_T=(A-B)*1000

figure(3)
mid=Q(:,30*3+2);
mid=mid-mid(1);
t=linspace(0,180,length(Q));
pos=pos-mean(pos);
mid=mid-mean(mid);
A=max(mid(12000:end));
B=min(mid(12000:end));
Disp_S=(A-B)*1000
Accur=((Disp_T-Disp_S)/Disp_T)*100
mid=mid(1:10:end);
pos=pos(1000:10:end);
t=t(1:1:end);
t_T=data2(:,1);
t_T=t_T(1000:10:end);
t_T=t_T;
%plot(t,mid,t_T,pos)
plot(t_T,pos)
title({'Test Vs Model';'Tern 14800N at 8.14Hz'})
xlabel('Time (s)')
ylabel('Displacement from Rest (m)');
legend('Model','Test')

```

```

function EI_final = Bending_Stiffness(cable)

%This function reads it's data from a .dat file in the main directory

M=dlmread(cable);
% A0 = 8.8141*1e-6; %Centre Core Area, m
% d=3.35e-3;
% E0 = 210000e+06; %Centre Core Stiffness N/m2
% k=1; %number of layers apart from single centre core
% i=[6 12]; %cables per layer
%
% A_k=[8.8141*1e-6 8.8141*1e-6]; %Area of each cable in a layer,
m
% alpha_k = [atand(pi*(3.35*3-3.35)/130) atand(pi*(20.7-(4.14*3-
4.14))/231.84)]; %Angle of wrap for a cable in a layer
% r_k = [3.35/1000 (3.35+4.14)/1000]; %radius from core to cable
in individual layer

A0 = M(1,1); %Centre Core Area, m
d=M(2,:);
k=M(3,1); %number of layers apart from single centre core
i=M(4,:); %cables per layer

A_k=M(5,:); %Area of each cable in a layer, m
alpha_k = [atand(pi*M(6,:))]; %Angle of wrap for a cable in a layer
r_k = M(7,:); %radius from core to cable in individual layer

```

```

E=M(8,1);

% EI_min=0;
% EI_sec=0;
% AE=0;
% for c=1:k
%     EI_min=E_k(c)*i(c)*((A_k(c)^2)/(4*pi))*cosd(alpha_k(c))
%     EI_sec=E_k(c)*(i(c)/2)*A_k(c)*r_k(c)^2*(cosd(alpha_k(c)))^3
%
%
%     AE=E_k(c)*i(c)*A_k(c)*(cosd(alpha_k(c)))^3;
%
% end

% core=E0*((A0^2)/(4*pi))
% EI_min=(EI_min+E0*((A0^2)/(4*pi)))
% EI_final=(EI_min+EI_sec)
% AE=AE+A0*E0
%
% ((A0^2)/(4*pi))+2*cosd(alpha_k(1))*((A0^2)/(4*pi))+4*(cosd(alpha_k(1))
% *(A0^2)/(4*pi))+A0*(3.35e-3*sin(pi/3))^2)
I0 = (A0.^2)/(4*pi);
I1=0;
Is=0;
for c=1:k
    I1(c) = (A_k(c)^2)/(4*pi)*cosd(alpha_k(c))*i(c);
    Is(c) = i(c)/2*(cosd(alpha_k(c))^3)*A_k(c)*r_k(c)^2;
end

Total_I= sum(I1)+ sum(Is)+ I0;
% Total_I=((A0.^2)/(4*pi))+((A0.^2)/(4*pi))*cosd(alpha_k(1))*i(1) +
i(1)/2*(cosd(alpha_k(1))^3*(A0*(r_k(1))^2))
% Total_EI=Total_I*E
EI_final=Total_I*E;
% (cosd(alpha_k(1)))^3
% I=(A^2)/(4*pi);
% EI_main=E*I
% AE_main=A*E
% ((EI_main-EI_final)/EI_final)*100

function
[rs,vs,fr]=shaker_drive(t,f_start,f_end,shaker_amp,methodflag)
%This is the function to drive the shaker displacement
step=0.5;
step_size=0.5;
step_count=0;
w=0;
rs=0;
vs=0;
lt=0;
t=linspace(t(1),t(end),length(t)*10);
delt=t(2)-t(1);

switch methodflag

```

```

case {'linear'}
    gr=((f_end-f_start)/t(end))*step_size*2*pi;
    for i=1:length(t)

        w(i)=f_start*2*pi+gr*(step_count);
        rs(i)=shaker_amp*sin(w(i)*lt);

        if t(i)>step && abs(-shaker_amp-rs(i))<0.001
            step=step+step_size;
            step_count=step_count+1;
            w(i)=f_start*2*pi+gr*(step_count);
            lt=asin(-1)/w(i)+(2*pi/w(i));
        end

        rs(i)=shaker_amp*sin(w(i)*lt);
        vs(i)=w(i)*shaker_amp*cos(w(i)*lt);
        lt=lt+delt;
    end

case {'logarithmic'}

    B=(f_end/f_start)^(1/t(end));
    for i=1:length(t)

        f(i)=f_start*B^(step-step_size);
        w(i)=f(i)*2*pi;
        rs(i)=shaker_amp*sin(w(i)*lt);

        if t(i)>step && abs(-shaker_amp-rs(i))<0.000005
            step=step+step_size;
            step_count=step_count+1;
            f(i)=f_start*B^(step-step_size);
            w(i)=f(i)*2*pi;
            lt=asin(-1)/w(i)+(2*pi/w(i));
        end

        rs(i)=shaker_amp*sin(w(i)*lt);
        vs(i)=w(i)*shaker_amp*cos(w(i)*lt);
        lt=lt+delt;
    end

end

end

rs=rs';
w=w';
vs=vs';
figure(1)
plot(t,rs)
figure(2)
plot(t,vs)
figure(3)
plot(t,w/(2*pi))
fr=w/(2*pi);
end

```

```

function [Force,Stiff,Damp,Mass] =
Boundary(Force,Stiff,Damp,Mass,r,v,t)
% this function assigns the boundary condition of a static axial
% co-ordinate.

global n

m = 4+n*3;
j=1;
for k=1:2           %This double for loop creates the boundary
condition at the ends for zero axial movement
    for i=1:m
        Force(i) = Force(i) - Stiff(i,j)*r(j) - Damp(i,j)*v(j);
        Stiff(j,i)=0;
        Stiff(i,j)=0;
        Stiff(j,j)=1;

        Damp(j,i)=0;
        Damp(i,j)=0;
        Damp(j,j)=1;

        Mass(j,i)=0;
        Mass(i,j)=0;
        Mass(j,j)=1;
        Force(j)=r(j);
    end
    j=m-3;
end

j=2;
for k=1:2           %This creates the end boundary conditions of no
movement in the Y direction
    for i=1:m
        Force(i) = Force(i) - Stiff(i,j)*r(j) - Damp(i,j)*v(j);
        Stiff(j,i)=0;
        Stiff(i,j)=0;
        Stiff(j,j)=1;

        Damp(j,i)=0;
        Damp(i,j)=0;
        Damp(j,j)=1;

        Mass(j,i)=0;
        Mass(i,j)=0;
        Mass(j,j)=1;
        Force(j)=r(j);
    end
    j=m-2;
end

j=3;
for k=1:2           %This creates the end boundary conditions of no
movement in the theta direction
    for i=1:m
        Force(i) = Force(i) - Stiff(i,j)*r(j) - Damp(i,j)*v(j);
        Stiff(j,i)=0;
        Stiff(i,j)=0;
        Stiff(j,j)=1;

```

```

        Damp(j,i)=0;
        Damp(i,j)=0;
        Damp(j,j)=1;

        Mass(j,i)=0;
        Mass(i,j)=0;
        Mass(j,j)=1;
        Force(j)=r(j);
    end
    j=m-1;
end
j=4+n*3;           %This creates the end boundary conditions of no
movement in any direction for the shaker driver
    for i=1:m
        Force(i) = Force(i) - Stiff(i,j)*r(j) -Damp(i,j)*v(j);
        Stiff(j,i)=0;
        Stiff(i,j)=0;
        Stiff(j,j)=1;

        Damp(j,i)=0;
        Damp(i,j)=0;
        Damp(j,j)=1;

        Mass(j,i)=0;
        Mass(i,j)=0;
        Mass(j,j)=1;
        Force(j)=r(j);
    end

    j=6;
    % for k=1:1           %This creates the end boundary conditions of no
movement in the theta direction
    %     for i=1:m-1
    %         Force(i) = Force(i) - Stiff(i,j)*r(j);% - Damp(i,j)*v(j);
    %         Stiff(j,i)=0;
    %         Stiff(i,j)=0;
    %         Stiff(j,j)=1;
    %
    %         Damp(j,i)=0;
    %         Damp(i,j)=0;
    %         Damp(j,j)=1;
    %
    %         Mass(j,i)=0;
    %         Mass(i,j)=0;
    %         Mass(j,j)=1;
    %         Force(j)=r(j);
    %     end
    % end
    %     j=m-1;

end

```

```

function [q] = Catenary(L,a,y_int,q)
%Catenary This function returns the co-ords for a cable as a catenary
%   Sag is a positive number

global n

j=1;
for i=1:n+1
    x(i)=q(j)+(L/n)*(i-1);
    j=j+3;
end
x
j=2;
for i=1:n+1
    q(j) = -y_int + a*cosh((L/2 - x(i))/a );
    j=j+3;
end
q(2) = 0;
q(3*n+2) = 0;
end

function [Omega,Phi,ModF] = Eigenmode(p,span)
%This function calculates the undamped eigenmodes of the
%   cable model. It returns the natural frequency and eigenvector

global n L Tension EI rho W

s = 4 + 3*n; %s is the size of the state vectors

pdot = zeros(size(p)); %initialise pdot
r = zeros(s,1); %r is the position vector, to be used
in calculating %the various state matrices
v = r;
a = r;

t=0;

for i=1:s %populate local position vector
    r(i) = p(i);
end

for i=1:s %populate local and global velocity
    vector
    j=s+i;
    v(i) = p(j);

    pdot(i) = p(j);
end

% L = element_length(r);

Beta = zeros(n,1);
Beta = transformation_angle_beta(r);

%Calculate initial Matrices
M = zeros(s);

```

```

M = globalmassmatrix(Beta);

K = zeros(s);
K = globalstiffnessmatrix(Beta);

F = zeros(size(r));
F = globalforcevector(r,v,t);

Damp = zeros(s);

[F,K,~,M] = Boundary(F,K,Damp,M,r);

[V,D] = eig(K,M);
[lamda,k] = sort(diag(D));
V=V(:,k);
Factor = diag(V'*M*V);
Phi = V*inv(sqrt(diag(Factor)));
Omega = diag(sqrt(Phi'*K*Phi));
ModF = Phi'*F;

for nmode=1:s

fn(nmode)=sqrt(((nmode*pi/span)^2)*(Tension/rho)*(1+(nmode*pi/span)^2*
(EI/Tension)))/(2*pi);
end
fn'
W=[fn' Omega]


function L = element_length(q,init)
%this returns the linear element length as a function of it's position
in 3
%space

global n

L=zeros(n,1);
q=q+init;
j=1;
k=2;
for i=1:(n)
    L(i) = sqrt((q(j+3)-q(j))^2 + (q(k+3)-q(k))^2);
    k=k+3;
    j=j+3;
end

end

```

```

function pdot = lagrange(t,p)
%this function returns the reduced order vector pdot, which contains
the
%velocity and acceleration vectors

global n Iter L zz next_t step_t countstep shaker_start rs vs t_interp
shaker_amp freq drivemethod

clc
s = 4 + 3*n; %s is the size of the state vectors

Iter = Iter + 1;
disp(['Iteration Count: ' num2str(Iter)])

pdot = zeros(size(p)); %initialise pdot
r = zeros(s,1); %r is the position vector, to be used
in calculating
v = r; %the various state matrices
a = r;

i=1:s; %populate local position vector
r(i) = p(i);

i=1:s; %populate local and global velocity vector
j=s+i;
v(i) = p(j);
pdot(i) = p(j);

Beta = zeros(n,1);
Beta = transformation_angle_beta(r);

angY = Beta;

i=1:n; % this loop converts the plane angle into
degrees for progress tracking
angY(i) = (angY(i)/pi)*180;

% disp(angY)

%Calculate initial Matrices
Force = globalforcevector(Beta,r,v,t);

Mass = zeros(s);
Mass = globalmassmatrix(Beta);

Stiff = zeros(s);
Stiff = globalstiffnessmatrix(Beta,r,v);

Damp = zeros(s);
Damp = globaldampingmatrix(Beta,Stiff,Mass);

```



```

[Force,Stiff,~,Mass] = Shaker (Force,Stiff,Damp,Mass,r,v,t,Beta);

% Stiff(10,10)
% Stiff(13,13)

% Force = zeros(size(r));
% if t<0.0001

% if t<next_t
%     %constant frequency
%     shaker_amp=0.001/2; %displacement peak to
peak amplitude of shaker base in m;
%     w=2*pi*(zz(countstep)) %The driving
conditions of the shaker base node
%     r(s) = shaker_start + shaker_amp*sin(w*t);
%     p(s) = r(s);
%     v(s) = shaker_amp*w*cos(w*t);
%     pdot(s) = v(s);
%     p(s+s) = v(s);
% else
%     countstep=countstep+1;
%     next_t=next_t+step_t;
%     shaker_amp=0.001/2; %displacement peak to
peak amplitude of shaker base in m;
%     w=2*pi*(zz(countstep)); %The driving
conditions of the shaker base node
%     r(s) = shaker_start + shaker_amp*sin(w*t);
%     p(s) = r(s);
%     v(s) = shaker_amp*w*cos(w*t);
%     pdot(s) = v(s);
%     p(s+s) = v(s);
% end

switch drivemethod
    case {'steady'}
        %constant frequency
        w=2*pi*(freq); %The driving conditions of
the shaker base node
        r(s) = r(s) + shaker_amp*sin(w*t);
        p(s) = r(s);
        v(s) = v(s) + shaker_amp*w*cos(w*t);
        pdot(s) = v(s);
        p(s+s) = v(s);

        otherwise
            %interpolated ramp
            ti=linspace(t_interp(1),t_interp(2),t_interp(3));
            r(s)=shaker_start + qinterp1(ti,rs,t,0);
            v(s)=qinterp1(ti,vs,t,0);
            p(s) = r(s);
            pdot(s) = v(s);
            p(s+s) = v(s);
end

[Force,Stiff,Damp,Mass] = Boundary (Force,Stiff,Damp,Mass,r,v,t);
%sim time
disp (['Simulation Time: ' num2str(t), ' Seconds']);
% DETERMINANT = [det(Mass) det(Damp) det(Stiff)]

```

```

% Condition = cond(Stiff)
% Damp
% Mass
% Stiff
% pause()

Damp = sparse(Damp);
Stiff = sparse(Stiff);
Mass = sparse(Mass);

B = (Force - Damp*v - Stiff*r);
a = mldivide(Mass,B);

x=zeros(1,n+1);
y=x;
i=1:3:s;
x = r(i)';
i = 2:3:s;
y = r(i)';

% disp(x);
disp(y);
% pause()

i=1:s;
j=s+i; %Result Parse out
pdot(j) = a(i);

% Force = (Stiff*r)'
timing(t); %shows real time used for simulation
end

function Damp = globaldampingmatrix(Beta,Stiff,Mass)
%This function returns the global damping matrix
global n
m=4+n*3; %this represents the final size of the matrix i.e. m x
m

local_damp=zeros(6);
gl_matrix = zeros(4+3*n);

% local_damp(1,1) = 5;
% local_damp(4,4) = local_damp(1,1);
% for i=1:6
%     local_damp(i,i) = 10.0;%0.00155; %this is the local damping
matrix, pretty simple
% end
l=0; %counter to shift starting position of next local
matrix
for k=1:n

```

```

        T = transformation(Beta(k)); %calculates the
local transformation matrix
        local_damp = T'*local_damp*T; %transformed matrix

        gl_matrix = globalmatrix(gl_matrix,local_damp,1);
        l=l+3;
end
Damp = gl_matrix;

% Damp = (0.1*Mass + 7e-6*Stiff);
% Damp = (0.15*Mass + 0.00002*Stiff);
Damp = (0.05*Mass + 0.00006*Stiff);

end

function Force = globalforcevector(Beta,r,v,t)
%This is a user defined function that returns the vector of non-
conserved external
%forces in the lagrange equation

global n Tension rho Om Ramp_Step Step_T grav

Force = zeros(4+3*n,1);
m= 4+3*n;

% Beta = zeros(n,1);
% Beta = transformation_angle_beta(r);

j=4;
for i=1:n-1
    Force(j) = -(Tension*cos(Beta(i))-Tension*cos(Beta(i+1)));
    Force(j+1) = -(Tension*sin(Beta(i))-Tension*sin(Beta(i+1)));
    j=j+3;
end

% Force(2) = Tension*sin(Beta(1));
% Force(1) = Tension*cos(Beta(1));
% % Force(3) = Tension*sin(r(3));
% Force(m-1) = Tension*sin(Beta(n));
% Force(m-2) = Tension*cos(Beta(n));
% % Force(m) = Tension*sin(r(m));

%This is the external gravity force, acting on each node in the -
Y(global)
%direction

if grav == true
    i=2:3:m;
    Force(i) = Force(i) - rho*84.6*9.81/(n-1);
end
% i=2:3:m;
% Force(i) = Force(i) + 5*sin(50*t);

% Force(3*n+2) = Force(3*n+2) + 27;
% Force(3*(n-5)+2) = 270;

```

```

% Force(6) = -500;
% if t<0.01
%     Force(16) = Force(16) + 5000;
%     Force(17) = Force(17) + 5000;
% end
% Force(7) = -50;

% %ramping frequency driver, single point oscillation
% if t>Step_T
%     Om=Om+50;
%     Force(11) = Force(11) + 10*sin(Om*t);
%     Step_T = Step_T + Ramp_Step;
% else
%     Force(11) = Force(11) + 10*sin(Om*t);
% end

%ramping frequency driver, distributed load oscillation
% if t>Step_T
%     Om=Om+50;
%     i=2:3:m;
%     Force(i) = Force(i) + 10*sin(Om*t);
%     Step_T = Step_T + Ramp_Step;
% else
%     i=2:3:m;
%     Force(i) = Force(i) + 10*sin(Om*t);
% end
% Ramp_Step
% Step_T
% Om

end

function Mass = globalmassmatrix(Beta)
%globalmassmatrix Computes the mass global mass matrix as a function
of Beta, for
% the co-ord transformation

global n L
local_mass=zeros(6); %This is local mass matrix, to be used in
making the global

gl_matrix = zeros(4+3*n);
T = zeros(6);

%populate global matrix with local elements

l=0; %counter to shift starting position of next local
matrix
for k=1:n
    local_mass = local_massmatrixlinear(L(k));

    T = transformation(Beta(k));
    local_mass = (T'*local_mass*T);
    gl_matrix = globalmatrix(gl_matrix,local_mass,l);

```

```

        l=l+3;
end

Mass = gl_matrix;
end

function local_stiff = local_stiffnesslinear(L)
%This creates a local stiffness matrix that is transformed

global AE EI

local_stiff = zeros(6);
F=12*(EI)/L^3;
G=4*EI/L;
P=6*EI/L^2;
H=2*EI/L;
K=AE/L;

local_stiff(1,1) = K;           %Column 1
local_stiff(4,1) = -K;

local_stiff(2,2) = F;           %Column 2
local_stiff(3,2) = P;
local_stiff(5,2) = -F;
local_stiff(6,2) = P;

local_stiff(2,3) = P;           %Column 3
local_stiff(3,3) = G;
local_stiff(5,3) = -P;
local_stiff(6,3) = H;

local_stiff(1,4) = -K;          %Column 4
local_stiff(4,4) = K;

local_stiff(2,5) = -F;          %Column 5
local_stiff(3,5) = -P;
local_stiff(5,5) = F;
local_stiff(6,5) = -P;

local_stiff(2,6) = P;           %Column 6
local_stiff(3,6) = H;
local_stiff(5,6) = -P;
local_stiff(6,6) = G;

end

```

```

function Stiff = globalstiffnessmatrix(Beta,r,v)
%This function calculates the global stiffness matrix
%    using the transformation angle and other various parameters

local_stiff=zeros(6);    %This is local mass matrix, to be used in
making the global
global n L Tension

gl_matrix = zeros(4+3*n);

%populate global matrix with local elements
l=0;    %counter to shift starting position of next local
matrix
for k=1:n
    local_stiff = local_stiffnesslinear(L(k)) +
    local_stiffness_nonlinear(L(k));
    T = transformation(Beta(k));    %calculates the
    local transformation matrix
    local_stiff = T'*local_stiff*T;    %transformed matrix
    gl_matrix = globalmatrix(gl_matrix,local_stiff,l);
    l=l+3;
end

m=3+n*3;    %this represents the final size of the matrix i.e. m x
m
Stiff = gl_matrix;

end

```

```

function init_pos = Initial_pos(init_pos,span,height)
%Initial_pos This returns the non-even spacing for the
%    Detailed explanation goes here

```

```

global n

sfactor = input('Scaling Factor:');
if isempty(sfactor)
    sfactor =1;
end

scale = linspace(sfactor,1,ceil(n/2));
L=sqrt(span^2+height^2)/n;
j=3*n-2;
for i=1:(n/2)
    init_pos(3*i+1) = init_pos(3*i+1)*scale(i);
    init_pos(3*i+2) = init_pos(3*i+2)*scale(i);
    init_pos(j) = span - init_pos(3*i+1);
    init_pos(j+1) = height - init_pos(3*i+2);
    j=j-3;
end
Beta = transformation_angle_beta(init_pos);

```

```

j=3;
for i=1:n+1
    init_pos(j) = Beta(1);
    j=j+3
end

function local_mass_lin = local_massmatrixlinear(L)
%This creates a local linear mass matrix that is transformed

global rho A

local_mass = zeros(6);

F = 13/35;
G = L^2/105;
H = 11*L/210;
I = 9/70;
J = 13*L/420;
K = L^2/140;

local_mass(1,1) = 1/3;
local_mass(4,1) = 1/6;

local_mass(2,2) = F;
local_mass(3,2) = H;
local_mass(5,2) = I;
local_mass(6,2) = -J;

local_mass(2,3) = H;
local_mass(3,3) = G;
local_mass(5,3) = J;
local_mass(6,3) = -K;

local_mass(1,4) = 1/6;
local_mass(4,4) = 1/3;

local_mass(2,5) = I;
local_mass(3,5) = J;
local_mass(5,5) = F;
local_mass(6,5) = -H;

local_mass(2,6) = -J;
local_mass(3,6) = -K;
local_mass(5,6) = -H;
local_mass(6,6) = G;

local_mass_lin = (rho*L)*local_mass;

end

```

```

function Yi = qinterp1(x,Y,xi,methodflag)
% Performs fast interpolation compared to interp1
%
% qinterp1 provides a speedup over interp1 but requires an evenly
spaced
% x array. As x and y increase in length, the run-time for interp1
increases
% linearly, but the run-time for
% qinterp1 stays constant. For small-length x, y, and xi, qinterp1
runs about
% 6x faster than interp1.
%
%
% Usage:
%   yi = qinterp1(x,Y,xi) - Same usage as interp1
%   yi = qinterp1(x,Y,xi,flag)
%       flag = 0          - Nearest-neighbor
%       flag = 1          - Linear (default)
%
% Example:
%   x = [-5:0.01:5];    y = exp(-x.^2/2);
%   xi = [-4.23:0.3:4.56];
%   yi = qinterp1(x,y,xi,1);
%
% Usage restrictions
%   x must be monotonically and evenly increasing
%   e.g., x=-36:0.02:123;
%
%   Y may be up to two-dimensional
%
% Using with non-evenly spaced arrays:
%   Frequently the user will wish to make interpolations "on the fly"
from
%   a fixed pair of library (i.e., x and y) vectors. In this case,
the
%   user can generate an equally-spaced set of library data by calling
%   interp1 once, and then storing this library data in a MAT-file or
%   equivalent. Because the speed of qinterp1 is independent of the
length
%   of the library vectors, the author recommends over-sampling this
%   generated set untill memory considerations start limiting program
speed.
%
%   If the user wishes to use two or more spacings (i.e., a closely-
spaced
%   library in the region of fine features, and a loosely-spaced
library in
%   the region of coarse features), just create multiple libraries,
record
%   the switching points, and send the search data to different
qinterp1
%   calls depending on its value.
%
% Example:
%   x1 = [-5:0.01:5];    x2 = [-40:1:-5 5:1:40];
%   y1 = exp(-x1.^2/3); y2 = exp(-x2.^2/3);
%   xi = [-30:0.3:30];
%   in = xi < 5 & xi > -5;
%   yi(in) = qinterp1(x1,y1,xi(in));
%   yi(~in) = qinterp1(x2,y2,xi(~in));

```



```

% Author: N. Brahms
% Copyright 2006

% Forces vectors to be columns
x = x(:); xi = xi(:);
sx = size(x); sY = size(Y);
if sx(1)~=sY(1)
    if sx(1)==sY(2)
        Y = Y';
    else
        error('x and Y must have the same number of rows');
    end
end

if nargin>=4
    method=methodflag;
else
    method = 1;    % choose nearest-lower-neighbor, linear, etc.
                  % uses integer over string for speed
end

% Gets the x spacing
ndx = 1/(x(2)-x(1)); % one over to perform divide only once
xi = xi - x(1);      % subtract minimum of x

% Fills Yi with NaNs
s = size(Y);
if length(s)>2
    error('Y may only be one- or two-dimensional');
end
Yi = NaN*ones(length(xi),s(2));

switch method
case 0 %nearest-neighbor method
    rxi = round(xi*ndx)+1;    % indices of nearest-neighbors
    flag = rxi<1 | rxi>length(x) | isnan(xi);
                                % finds indices out of bounds
                                % finds indices in bounds
    nflag = ~flag;
    Yi(nflag,:) = Y(rxi(nflag),:);
case 1 %linear interpolation method
    fxi = floor(xi*ndx)+1;    % indices of nearest-lower-
neighbors
    flag = fxi<1 | fxi>length(x)-1 | isnan(xi);
                                % finds indices out of bounds
                                % finds indices in bounds
    nflag = ~flag;
    Yi(nflag,:) = (fxi(nflag)-xi(nflag)*ndx).*Y(fxi(nflag),:)+...
        (1-fxi(nflag)+xi(nflag)*ndx).*Y(fxi(nflag)+1,:);
end

function [Force,Stiff,Damp,Mass] =
Shaker(Force,Stiff,Damp,Mass,r,v,t,Beta)
% this function creates the effects of a real shaker with spring mass
% attachment

global n

```

```

m = 4+n*3;
Con_Node = 2;    %This specifies to which cable node the shaker
attaches
A = Con_Node*3-1;
B = m;

K = 3514.2;    %Stiffness of shaker connection N/m
M = 0.53;    %Mass of shaker connection Kg

Stiff(A,A) = Stiff(A,A) + K;
Stiff(A,B) = Stiff(A,B) - K;
Stiff(B,A) = Stiff(B,A) - K;
Stiff(B,B) = Stiff(B,B) + K;

%Shaker has a natural frequency at 12.95hz

% Damp(B,B) = Damp(B,B) + 0.00155;
% Damp(A,A) = Damp(A,A) + 0.00155;
% Damp(A,B) = Damp(A,B) - 0.00155;
% Damp(B,A) = Damp(B,A) - 0.00155;

Mass(A,A) = Mass(A,A) + M;
% Mass(A,B) = Mass(A,B) + M;
% Mass(B,A) = Mass(B,A) + M;
% Mass(B,B) = Mass(B,B) + M/2;
% Mass(A,A) = Mass(A,A) + M;
% Force(B) = 1000*sin(66.41*t);
end

N = length(Q);

global n init_pos
X = zeros(n+1,1);
Y = X;
TH=Y;

% x1 = [0 10.575 21.15 42.3 63.45 74.2 84.6]'
% y1 = [0 -0.155 -0.295 -0.460 -0.295 -0.155 0]'
%
% x2 = [0 8.4600 16.9200 25.3800 33.8400 42.3000 50.7600 59.2200
67.6800 76.1400 84.6000]'
% y2 = [0 -0.1372 -0.2432 -0.3190 -0.3644 -0.3796 -0.3644 -0.3190 -
0.2432 -0.13720 0]'
%
% x3 = [0 0.645 1.2 3.2 5.2 7.2 9.2 11.2 13.2 15.2 17.2 19.2 21.2 23.2
25.2 27.2 31.2 33.2 35.2 37.2 39.2 41.2 42.5]';
% y3 = [1.139 1.125 1.110 1.080 1.050 1.010 0.975 0.952 0.920 0.894
0.865 0.845 0.823 0.806 0.785 0.775 0.756 0.745 0.735 0.722 0.712
0.715 0.717]';
% l = length(x3);
% for i=1:l-1
%     x3(l+i) = x3(l+i-1) + (x3(l-i+1)-x3(l-i));
%     y3(l+i) = y3(l-i);
% end
% y3 = y3 - 1.139;

```

```

% x3
% y3
% length(x3)

%x axis measurments valid for all cable readings
x1 = [0 5.2875 10.575 21.15 31.725 42.3 (84.6-31.725) (84.6-21.15)
(84.6-10.575) (84.6-5.2875) 84.6]';

%Pelican
y2 = [1.129 1.00 0.888 0.706 0.602 0.577 0.602 0.706 0.888 1.00
1.129]'; %Pelican 8120
y2=y2-1.129;

y3 = [1.129 1.026 0.936 0.787 0.702 0.682 0.702 0.787 0.936 1.026
1.129]'; %Pelican 13070
y3=y3-1.129;

y4 = [1.129 1.04 0.899 0.721 0.624 0.597 0.624 0.721 0.899 1.04
1.129]'; %Pelican 11200
y4 = y4-1.129;

%Rabbit
y5 = [1.1330 1.021 0.917 0.750 0.653 0.625 0.653 0.750 0.917 1.021
1.1330]';
y5=y5-1.1330;

y6 = [1.1330 1.057 0.984 0.870 0.804 0.79 0.804 0.870 0.984 1.057
1.1330]';
y6=y6-1.1330;

y7 = [1.330 1.023 0.930 0.779 0.692 0.667 0.692 0.779 0.930 1.023
1.330]';
y7 = y7-1.330;

%Tern
y8 = [1.280 0.955 0.806 0.573 0.432 0.394 0.432 0.573 0.806 0.955
1.280]';
y8 = y8-1.280;

y9 = [1.280 1.002 0.891 0.716 0.610 0.586 0.610 0.716 0.891 1.002
1.280]';
y9 = y9 - 1.280;

y10 = [1.280 1.028 0.940 0.801 0.717 0.697 0.717 0.801 0.940 1.028
1.280]';
y10 = y10 - 1.280;

%Bersford
y11 = [1.131 1.011 0.919 0.763 0.678 0.653 0.678 0.763 0.919 1.011
1.131]';
y11 = y11-1.131;

y12 = [1.131 0.996 0.891 0.713 0.614 0.587 0.614 0.713 0.891 0.996
1.131]';
y12 = y12 - 1.131;

```

```

y13 = [1.131 0.953 0.811 0.588 0.445 0.405 0.445 0.588 0.811 0.953
1.131]';
y13 = y13 - 1.131;

i=1
    j=1;
    for k=1:n+1
        X(k)=Q(i,j) + init_pos(j);
        Y(k)=Q(i,j+1)+ init_pos(j+1);
        TH(k)=Q(i,j+2) + init_pos(j+2);
        % Fx(k)=start(j)+init_pos(j);
        % Fy(k)=start(j+1)+init_pos(j+1);
        j=j+3;
    end

    V=[];
    U=[];

    for k=1:n

        z=0;
        interval=(X(k+1)-X(k))/100;
        l = X(k+1)-X(k);
        for x=0:interval:l
            z=z+1;
            % l=L(k);
            N1 = (1-x/l);
            N2 = (2*(x^3)-3*l*(x^2) + l^3)/l^3;
            N3 = (x^3 - 2*l*x^2 + l^2*x)/l^2;
            N4 = (x/l);
            N5 = -(2*x^3 - 3*l*x^2)/l^3;
            N6 = (x^3 - l*x^2)/l^2;

            V_ele(z) = [N2 N3 N5 N6]*[Y(k) TH(k) Y(k+1) TH(k+1)]';
            U_ele(z) = [N1 N4]*[X(k) X(k+1)]';
        end
        V=horzcat(V,V_ele);
        U=horzcat(U,U_ele);

    end
    V;
    U;

    % for a=1:length(V)-1
    % Length(a) = sqrt((V(a+1)-V(a))^2+(U(a+1)-U(a))^2);
    % end
    % Total=sum(Length);

    %x1,y2,x1,y3,x1,y4,x1,y5,x1,y6,x1,y7,x1,y8,x1,y9,x1,y10,x1,y11,x1,y12,
    x1,y13
    figure(1);
    plot(U,V,x1,y2)
    axis manual
    axis([0 84.6 -0.6505 0.05])
    % pause(0.04)

    disp('Shows Over')
end

```

```

N = length(Q);

global n init_pos
X = zeros(n+1,1);
Y = X;
TH=Y;

% x1 = [0 10.575 21.15 42.3 63.45 74.2 84.6]'
% y1 = [0 -0.155 -0.295 -0.460 -0.295 -0.155 0]'
%
% x2 = [0 8.4600 16.9200 25.3800 33.8400 42.3000 50.7600 59.2200
67.6800 76.1400 84.6000]'
% y2 = [0 -0.1372 -0.2432 -0.3190 -0.3644 -0.3796 -0.3644 -0.3190 -
0.2432 -0.13720 0]'
%
% x3 = [0 0.645 1.2 3.2 5.2 7.2 9.2 11.2 13.2 15.2 17.2 19.2 21.2 23.2
25.2 27.2 31.2 33.2 35.2 37.2 39.2 41.2 42.5]';
% y3 = [1.139 1.125 1.110 1.080 1.050 1.010 0.975 0.952 0.920 0.894
0.865 0.845 0.823 0.806 0.785 0.775 0.756 0.745 0.735 0.722 0.712
0.715 0.717]';
% l = length(x3);
% for i=1:l-1
%     x3(l+i) = x3(l+i-1) + (x3(l-i+1)-x3(l-i));
%     y3(l+i) = y3(l-i);
% end
% y3 = y3 - 1.139;
% x3
% y3
% length(x3)

%x axis measurments valid for all cable readings
x1 = [0 5.2875 10.575 21.15 31.725 42.3 (84.6-31.725) (84.6-21.15)
(84.6-10.575) (84.6-5.2875) 84.6]';

%Pelican
y2 = [1.129 1.00 0.888 0.706 0.602 0.577 0.602 0.706 0.888 1.00
1.129]'; %Pelican 8120
y2=y2-1.129;

y3 = [1.129 1.026 0.936 0.787 0.702 0.682 0.702 0.787 0.936 1.026
1.129]'; %Pelican 13070
y3=y3-1.129;

y4 = [1.129 1.04 0.899 0.721 0.624 0.597 0.624 0.721 0.899 1.04
1.129]'; %Pelican 11200
y4 = y4-1.129;

%Rabbit
y5 = [1.1330 1.021 0.917 0.750 0.653 0.625 0.653 0.750 0.917 1.021
1.1330]';
y5=y5-1.1330;

y6 = [1.1330 1.057 0.984 0.870 0.804 0.79 0.804 0.870 0.984 1.057
1.1330]';
y6=y6-1.1330;

```

```

y7 = [1.330 1.023 0.930 0.779 0.692 0.667 0.692 0.779 0.930 1.023
1.330]';
y7 = y7-1.330;

%Tern
y8 = [1.280 0.955 0.806 0.573 0.432 0.394 0.432 0.573 0.806 0.955
1.280]';
y8 = y8-1.280;

y9 = [1.280 1.002 0.891 0.716 0.610 0.586 0.610 0.716 0.891 1.002
1.280]';
y9 = y9 - 1.280;

y10 = [1.280 1.028 0.940 0.801 0.717 0.697 0.717 0.801 0.940 1.028
1.280]';
y10 = y10 - 1.280;

%Bersford
y11 = [1.131 1.011 0.919 0.763 0.678 0.653 0.678 0.763 0.919 1.011
1.131]';
y11 = y11-1.131;

y12 = [1.131 0.996 0.891 0.713 0.614 0.587 0.614 0.713 0.891 0.996
1.131]';
y12 = y12 - 1.131;

y13 = [1.131 0.953 0.811 0.588 0.445 0.405 0.445 0.588 0.811 0.953
1.131]';
y13 = y13 - 1.131;


for i=1:5:N
    j=1;
    for k=1:n+1
        X(k)=Q(i,j) + init_pos(j);
        Y(k)=Q(i,j+1)+ init_pos(j+1);
        TH(k)=Q(i,j+2) + init_pos(j+2);
        %    Fx(k)=start(j)+init_pos(j);
        %    Fy(k)=start(j+1)+init_pos(j+1);
        j=j+3;
    end

    V=[];
    U=[];

    for k=1:n

        z=0;
        interval=(X(k+1)-X(k))/100;
        l = X(k+1)-X(k);
        for x=0:interval:l
            z=z+1;
            %    l=L(k);
            N1 = (1-x/l);
            N2 = (2*(x^3)-3*l*(x^2) + l^3)/l^3;
            N3 = (x^3 - 2*l*x^2 + l^2*x)/l^2;
            N4 = (x/l);

```

```

N5 = -(2*x^3 - 3*1*x^2)/1^3;
N6 = (x^3 - 1*x^2)/1^2;

V_ele(z) = [N2 N3 N5 N6]*[Y(k) TH(k) Y(k+1) TH(k+1)]';
U_ele(z) = [N1 N4]*[X(k) X(k+1)]';
end
V=horzcat(V,V_ele);
U=horzcat(U,U_ele);

end
V;
U;

% for a=1:length(V)-1
% Length(a) = sqrt((V(a+1)-V(a))^2+(U(a+1)-U(a))^2);
% end
% Total=sum(Length);

%x1,y2,x1,y3,x1,y4,x1,y5,x1,y6,x1,y7,x1,y8,x1,y9,x1,y10,x1,y11,x1,y12,
x1,y13
figure(1);
plot(U,V,x1,y2)
axis manual
axis([0 84.6 -0.6505 0.05])
% pause(0.04)

end

disp('Shows Over')

```

```

function timing(t)
%this converts the time measured to perform the calculations into a
%realtime counter for easier monitoring
%start_t is when the program first starts, with the time difference
being
%calculated each time the function is called.

```

```

global start_t

```

```

elapsed = clock - start_t;

```

```

if elapsed(6) < 0
    elapsed(6) = elapsed(6) + 60;
    elapsed(5) = elapsed(5) - 1;
end

```

```

if elapsed(5) < 0
    elapsed(5) = elapsed(5) + 60;
    elapsed(4) = elapsed(4) - 1;
end

```

```

if elapsed(4) < 0
    elapsed(4) = elapsed(4) + 24;
    elapsed(3) = elapsed(3) - 1;
end

```

```

end

if elapsed(3) < 0
    elapsed(3) = elapsed(3) + 1;
end

day = elapsed(3);
hour = elapsed(4);
min = elapsed(5);
sec = elapsed(6);

disp(' ')
disp('Simulation has been runnning for:')
disp([num2str(day), ' Days ', num2str(hour), ' Hours ', num2str(min), '
Minutes ', num2str(sec), 's '])
end

step_t=0.5952;
next_t=step_t;
countstep=1;
r=0;
v=0;
t=linspace(0,100,15000);
gw=0;
y_hz=0;
i=0;
r1=0;
v1=0;
for s=1:length(t)

if t(s)<next_t
%     %constant frequency
%     shaker_amp=0.0015;
%     i=i+1;%displacement peak to peak amplitude of shaker base in m;
%     w=2*pi*(zz(countstep)); %The driving
conditions of the shaker base node
%     r(s) = shaker_start + shaker_amp*sin(w*t(s));
%     v(s) = shaker_amp*w*cos(w*t(s));
%     gw(s)=w;
y_hz(countstep,i)=y(s);
else
countstep=countstep+1;
next_t=next_t+step_t;
%     shaker_amp=0.0015; %displacement peak to peak
amplitude of shaker base in m;
%     w=2*pi*(zz(countstep)); %The driving
conditions of the shaker base node
%     r(s) = shaker_start + shaker_amp*sin(w*t(s));
%     v(s) = shaker_amp*w*cos(w*t(s));
%     gw(s)=w;
i=1;
y_hz(countstep,i)=y(s);
end

w0=2*pi*(5); %The driving conditions of the
shaker base node
m=2*pi*(2/60); %hz increase per minute

```



```

%      r1(s) = shaker_amp*sin(w0*t(s) + m*t(s)^2);
%      v1(s) = shaker_amp*(w0+2*m*t(s))*cos(w0*t(s)+m*t(s)^2);

end

for i=1:countstep
    Y(i)=max(abs(y_hz(i,:)));
end

shaker_amp=0.001; %displacement peak to peak amplitude of shaker base
in m;

function Beta = transformation_angle_beta(q)
%This calculates the angle between the element and global co-ord
system
% after each time step for the XY plane

j=5;
global n init_pos
Beta = zeros(1,n);
real_pos =init_pos + q;
for i=1:n
    m=((real_pos(j)-real_pos(j-3))/(real_pos(j-1)-real_pos(j-4)));
    Beta(i) = atan(m);
    j=j+3;
end
end

%run mainbody.m first to generate frequency spectrum
%

oldfolder=cd('C:\Users\TopHat\Desktop\MSc\Proper Final Sim
Results\Grosbeak')
Model_in=xlsread('sim_results Grosbeak 16481N 45s 5-17.5hz 2nd
3.xlsx');
cd(oldfolder)
M_mid=Model_in(:,39*3+2);
M_drive=Model_in(:,40*3+4);
M_mid=M_mid-M_mid(1); %shifts graph to start at zero on axis i.e.
relative displacement
M_drive=M_drive-M_drive(1);
sim_t=linspace(0,45,length(M_mid));
frl=fr(1:10:150000);

%Code to generate downsampled data from model results
step_t=0.1;
next_t=step_t;
countstep=1;
gw=0;
y=M_mid;
y_hz=0;
i=0;
y=y-y(1);
Y=0;
I=0;
t_hz=0;

```

```

Thz=0;
fr_hz=0;
for s=1:length(t)

    if sim_t(s)<next_t
        %constant frequency
        i=i+1;
        y_hz(countstep,i)=y(s);
        t_hz(countstep,i)=sim_t(s);
    else
        countstep=countstep+1;
        next_t=next_t+step_t;
        i=1;
        y_hz(countstep,i)=y(s);
        t_hz(countstep,i)=t(s);
    end

end

abs(y_hz);
for i=1:countstep
    [Y(i),I]=max(abs(y_hz(i,:)));
    Thz(i)=t_hz(i,I);
end
fr_hz=qinterpl(t,fr1,Thz,0);
Y=Y*2000;    %scale to mm and peak to peak displacement
%END OF SIMULATED DATA PROCCESSING


% %this section takes the cell data from
% %the puma shaker results and produces a usable matrix.
% %the data must be imported from the excel file as cell data, x.
% q=0;
% x=T_mid;
% z=zeros(length(x),2);
% for i=1:length(x)
%     u=cell2mat(x(i));
%     q=str2num(u);
%     z(i,1)=q(1);
%     z(i,2)=q(2);
% end
%
% q=0;
% x=T_s;
% z1=zeros(length(x),2);
% for i=1:length(x)
%     u=cell2mat(x(i));
%     q=str2num(u);
%     z1(i,1)=q(1);
%     z1(i,2)=q(2);
% end
%
% q=0;
% x=T_d;
% z2=zeros(length(x),2);
% for i=1:length(x)
%     u=cell2mat(x(i));
%     q=str2num(u);

```

```

%      z2(i,1)=q(1);
%      z2(i,2)=q(2);
% end
%
% Test_F=z(:,1);
% Test_Y=z(:,2);
%
% Test_F1=z1(:,1);
% Test_Y1=z1(:,2);
%
% Test_F2=z2(:,1);
% Test_Y2=z2(:,2);

% %STEADY FREQ FROM ACC DATA PROCESS
% G=9806.65 %mm/s^2
% F=9.613 %Hz
% Real_t=Acc_Data(:,1);
% Acc=Acc_Data(:,2);
% Disp = (G*Acc)/(2*(pi^2)*F^2);
% Disp=Disp-mean(Disp);

%Align Axes, scale sim results and plot
% Diff=Y(1)-Test_Y(1);
% Y=Y-Diff; %align starting value

figure(1)
plot(fr_hz,Y)%Test_F,Test_Y,,Test_F1,Test_Y1,Test_F2,Test_Y2,fr_hz,Y)
axis manual
axis([5 17.5 -0.1505 max(Y)+0.5])
% figure(2)
% plot(Real_t,Disp)

```

APPENDIX D

Conductor Specifications

Aluminum Conductor Steel Reinforced - ACSR (Canadian Standard Sizes)																
Code name	Equival. copper area mm ²	Stranding and wire diameter mm	Diameter over steel mm	Overall diameter mm	Aluminium area mm ²	Steel area mm ²	Total area mm ²	Mass kg/km		Ultimate tensile strength Newton	Coefficient of linear expansion /C° x 10 ⁻⁴	Initial modulus of elasticity MPa	Final modulus of elasticity MPa	DC resistance at 20°C Ω/km	Current rating A	Standard drum length m
PELICAN BERSFORD	152,00	18/1/4,14	4,14	20,70	242,31	13,46	255,77	669	106	53800	21,44	40700	66200	0,1189	600	1000
	430,70	48/4,27 +7/3,32	9,96	35,58	687,36	60,60	747,96	1906	480	177650	20,68	43200	68800	0,0420	1132	2000
GROSBEAK	202,70	26/3,97 +7/3,09	9,27	25,15	321,84	52,49	374,34	891	415	111000	19,37	48800	77100	0,0897	730	1000
TERN	253,40	45/3,38	6,75	27,00	403,77	27,83	431,60	1120	220	98700	21,12	42900	66600	0,0718	830	1500
RABBIT	32,26	6/1/3,35	3,35	10,05	52,88	8,81	61,70	145	69,40	18500	19,31	49500	80400	0,5426	240	1500

Data Obtained From Aberdare Cables

APPENDIX E

Shaker Specifications

System		TV 56263/LS	
Shaker		S 561/LS	
Amplifier		A 52312	
Blower		SD 9	
		Metric	American
Rated peak force (N)/(lbf)	Sine/Random	6300/6300/12600	1416/1416/2833
Frequency range (Hz)		DC - 3000	DC - 3000
Max. rated travel (mm)/(inch)	Pk - Pk	50.8	2.0
Max. velocity (m/s)/(inch/sec)	Sine/Random	1.7/1.7/2.5	67/67/98
Max. acceleration (g)	Sine/Random	81/81/161	81/81/161
Rated current (A)		90	90
Nominal impedance (Ohm)		0.5	0.5
Suspension Stiffness (N/mm) (lbf/inch)		50	285.5
Effective moving mass (kg)/(lb)		8	17.6
Max. weight tested (kg)/(lb)		150	330.7
Main resonance frequency (Hz)		>3000	>3000
Weight with trunnion (kg)/(lb)		1000	2204.6
Stray magnetic field (mT)	without/with degauss kit	<8.5/<0.5	<8.5/<0.5
Armature (ø/mm)/(ø/inch)		180	7.09
Cooling (m³/h)/(ft³/min)		500	294
Interlocks		Temperature Overtravel Airflow	