**School of Civil and Mechanical Engineering**

**Department of Mechanical Engineering**

# Automatic Multi-Modal Tuning of Idiophone Bars

**Zhao Mingming**

**This thesis is presented for the Degree of**
**Master of Philosophy**
**of**
**Curtin University**

**October 2011**

This page intentionally left blank.

## Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: ………………………………………….

Date: ………………………………………….

## Acknowledgements

I would like to thank my supervisor Dr. Rodney Entwistle for guiding me throughout this research. My supervisor has been very generous to me with his time, his profession and enthusiasm always encourage me though the good and bad phases during my research.

I would like to thank Mr. Frank Cicoria for the number of hours he spent with me in the laboratory. Frank has always been supportive and friendly.

I also appreciate the help from Mr. Dave Collier for providing the room and the mechanical components to me for this research.

# Abstract

Idiophones generate sound through the vibration of their beam-like “keys”. The musical sound generated depends on the natural bending vibrations of the free-free beams. The tonal quality of the idiophone bar is achieved by tuning the second and third natural bending frequencies in relation to the fundamental natural frequency. Tuning these harmonic overtones becomes one of the primary tasks for making idiophone bars. It is achieved by removing material from the underside of the beams. This thesis focuses on the accurate prediction of the geometry of the beam underside (undercut) shape of marimba bars[1] and the fine tuning process for correcting the unavoidable uncertainties of wood during automated tuning.

The correct underside shape of the marimba bar was predicted using Timoshenko beam receptances. The underside shape predictive model predicts the resulting natural bending frequencies based on the undercut geometry of the bars. A search algorithm was implemented to find the correct geometry of the undercut for the multi-mode frequency requirements. A CNC machine tool was adapted to mill the specified underside shape from a wood blank, and this machine tool was combined with the predictive model and automatically controlled by the hardware controlling program. A fine tuning program was developed to incrementally approach the target natural frequencies from above, thereby correcting for the unknown non-homogeneity and anisotropy of wood.

Manufacture of wooden bars showed that the underside shape predictive model was very accurate when the elastic properties of the test material are accurate. For non-homogeneous and anisotropic material the improvement of the actual results made by the fine tuning program were observed. A physical machining centre, which combines the underside shape predictive model, the fine tuning program, the hardware controlling program, the frequency measuring program and a self-built CNC machine, have been developed to automate the tuning process.

[1] Idiophones can be classified by their materials, for example, marimba bars are made of wood and vibraphone bars are made of aluminium. (Marimba bars were studied in this thesis.)

## Nomenclature

| Symbols | Definitions | Units |
|---|---|---|
| A | Area of cross-section of a beam | $m^2$ |
| $a_1$ | Coefficient in the cubic shape equation | $y = a_1x^3 + a_2x^2 + a_3$ |
| $a_2$ | Coefficient in the cubic shape equation | $y = a_1x^3 + a_2x^2 + a_3$ |
| $a_3$ | Coefficient in the cubic shape equation | $y = a_1x^3 + a_2x^2 + a_3$ |
| b | Width of a beam | m |
| d | Depth of a beam | m |
| E | Young's modulus | $N/m^2$ |
| fn | Natural bending frequency of the $n^{th}$ mode | Hz |
| G | Shear modulus | $N/m^2$ |
| I | Second moment of area | $m^4$ |
| len | Length of a beam | m |
| M | Bending moment | Nm |
| n | Number of elements | - |
| F | Shear force | N |
| T | Applied torque | Nm |
| y | Displacement response | m |
| *v* | Poisson's ratio | - |
| ω | Angular velocity | rad/s |
| α | Receptance of system A | Various |
| β | Receptance of system B | Various |
| γ | Receptance of system C | Various |
| φ | Phase angle | rad |
| ψ | Slope of deflection curve | - |
| ρ | Density | $kg/m^3$ |

# Table of Contents

Declaration ........................................ i

Acknowledgements ........................................ ii

Abstract ........................................ iii

Nomenclature ........................................ iv

List of Figures ........................................ ix

List of Tables ........................................ xii

Chapter 1: Introduction ........................................ 1

Chapter 2: Literature Review ........................................ 5

2.1 Selected Literature ........................................ 5

2.1.1 Non-uniform beams with harmonically related overtones for use in percussion instruments (Bustamante, 1991) ........................................ 5

2.1.2 The receptances of beams, in closed form, including the effects of shear and rotary inertia (Stone, 1992) ........................................ 6

2.1.3 Practical tuning of xylophone bars and resonators (Bork, 1994) .......... 7

2.1.4 Tuning process of xylophone and marimba bars analysed by finite element modelling and experimental measurements (Ostergaard, 1997) ........................................ 7

2.1.5 Finite element analysis and experimental measurements of natural eigenmodes and random responses of wooden bars used in musical instruments (Bretos, Santamaria and Moral, 1998) ........................................ 8

2.1.6 Basic physics of xylophone and marimba bars (Suits, 2001) .............. 9

2.1.7 Designing idiophones with tuned overtones (Legge and Petrolito, 2007) ........................................ 10

2.1.8 Geometric shape identification for multi-mode tuning of percussion instrument bars (Entwistle and McGrechan, 2007) ........................................ 11

2.2 A Summary of the Major Agreements and Disagreements in the Selected Literature ........................................ 12

2.2.1 Agreement on the Timoshenko Model ........................................ 12

2.2.2 Agreement on the Receptance Model ........................................ 12

2.2.3 Disagreement on the Elastic Properties .......... 13
2.3 Implications for Present Work .......... 14

Chapter 3: Marimba Bar Tuning .......... 17
3.1 Support and End Conditions .......... 17
3.2 Influence of the Vibrational Behaviour due to the Undercut Shape .......... 20
3.3 Determination of the Tuning Intervals .......... 23

Chapter 4: Timoshenko-receptance Model .......... 25
4.1 The Receptance Sub-structuring Method .......... 25
4.1.1 Sub-systems Linked by a Single Coordinate .......... 26
4.1.2 Sub-systems Linked by Multiple Coordinates .......... 28
4.2 The Timoshenko-receptance Model .......... 31

Chapter 5: The Underside Shape Predictive Model .......... 37
5.1 Selection of the Mathematical Model .......... 38
5.2 The Preparation of Model Inputs .......... 39
5.3 Possible Cutout Geometries .......... 40

Chapter 6: Fine Tuning Program .......... 43
6.1 The Fine Tuning Procedure .......... 44
6.2 The Determination of the “Relationship Linear Equations” .......... 47
6.3 The Determination of the “Shape Matrix” .......... 50

Chapter 7: CNC Machine Tool and Automatic Control .......... 55
7.1 The Physical Machining Centre .......... 57
7.2 Determination of the Tool Path Coordinates .......... 58
7.3 Automatic Cutter Control .......... 62
7.4 Automatic Spindle Control .......... 67
7.5 Frequency Measurement .......... 68

Chapter 8: Experiments and Results .......... 71
8.1 The Underside Shape Prediction Test .......... 71
8.2 Fine Tuning Program .......... 78

Chapter 9: Discussion ........................................................................................ 89

9.1 Sources of Error ........................................................................................ 89

9.1.1 Errors from the Predictive Model ........................................................ 89

9.1.2 Errors from the Fine Tuning Program and Measurements ................. 90

9.1.3 Errors from Manufacture ...................................................................... 90

9.2 Application.................................................................................................. 91

Chapter 10: Conclusions and Future Work.................................................... 95

Chapter 11: References .................................................................................. 97

Appendix A: Mathematical Models.................................................................. 101

Appendix A.1 Euler-Bernoulli Model .............................................................. 101

Appendix A.2 Timoshenko Model .................................................................... 104

Appendix A.3 Finite Element Model................................................................ 108

Appendix B: Calculations for the Mechanical Properties............................... 110

Appendix C: Newton-Raphson Method............................................................ 112

Appendix D: G-Code Generated by the Auto-tuning Program........................ 115

Appendix E: G-Code Generated by AlphaCam................................................ 116

Appendix F: Fine Tuning Results .................................................................... 117

Appendix F.1: Fine Tuning Test (27/05/2011) .................................................. 117

Appendix F.2: Fine Tuning Test (06/06/2011) .................................................. 119

Appendix G: Programming Code..................................................................... 121

Appendix G.1: main.m....................................................................................... 124

Appendix G.2 Method_Timoshenko.m ............................................................. 126

Appendix G.3 Shape.m ...................................................................................... 127

Appendix G.4 beam_recept_timoshenko.m........................................................ 129

Appendix G.5 bracket_natfreq.m....................................................................... 132

Appendix G.6 find_freq_timo.m........................................................................133

Appendix G.7 shear_factor_timoshenko.m.......................................................135

Appendix G.8 Tuning_Program.m....................................................................136

Appendix G.9 Shape_Matrix.m ........................................................................138

Appendix G.10 Method_Timo_TP.m ...............................................................140

Appendix G.11 find_freq_timo_TP.m ..............................................................141

Appendix G.12 Shape_Reprediction.m.............................................................143

Appendix G.13 desired_vs_experimental.m .....................................................144

Appendix G.14 Toolpath_Generation.m...........................................................146

Appendix G.15 Coordinates_to_steps.m...........................................................150

Appendix G.16 Steps_to_pulses.m ...................................................................152

Appendix G.17 Driving_motors.m ...................................................................155

Appendix G.18 G_Code_Generation.m............................................................156

## List of Figures

Figure 1.1 Marimba (University of Notre Dame, 2011) ............................................ 1
Figure 1.2 A marimba beam.......................................................................... 2

Figure 2.1 A wood blank.............................................................................. 9

Figure 3.1 Spatial distribution of shear force, bending moment and displacement for the first five modes of a prismatic bar (after Bork, 1994) ........................ 19
Figure 3.2 Spatial distribution of shear force, bending moment and displacement for the first five modes of a tuned marimba bar (after Bork, 1994)............... 19
Figure 3.3 Effect of removing material from various locations along a uniform bar (after Bork, 1994) ......................................................................... 21
Figure 3.4 Effect of removing material from various locations along a marimba bar (after Bork, 1994) ......................................................................... 21
Figure 3.5 Effect of removing material from various locations along a marimba bar prepared for tuning the third partial (after Bork, 1994)............................ 22
Figure 3.6 Subjective valuation of xylophone sounds (after Bork, 1994) ................ 24

Figure 4.1: A cross-receptance system...................................................... 26
Figure 4.2: A System with two beam sections B and C by a single coordinate......... 27
Figure 4.3: A system with two beam sections B and C coupled by two coordinates (Entwistle and McGrechan, 2007)............................................................ 29

Figure 5. 1 A flowchart of the underside shape predictive model solution procedure ......................................................................................................... 37
Figure 5.2 A marimba beam with undercut (Inverted) ............................................ 41
Figure 5.3 Tip receptance of a marimba beam......................................................... 42

Figure 6.1 A flowchart of the fine tuning process .................................................... 45
Figure 6.2 Cutting layers......................................................................................... 47
Figure 6.3 Incremental target $f_1$ vs. experimental $f_1$ ................................................. 48
Figure 6.4 Incremental target $f_2$ vs. experimental $f_2$ ................................................. 48
Figure 6.5 Incremental target $f_3$ vs. experimental $f_3$ ................................................. 49

Figure 6.6 Natural frequency changes vs. the change of $a_1$ ........................................ 50
Figure 6.7 Natural frequency changes vs. the change of $a_2$ ........................................ 51
Figure 6.8 Natural frequency changes vs. the change of $a_3$ ........................................ 51
Figure 6.9 Original underside shape and the re-predicted underside shape (Inverted) ........................................ 53

Figure 7.1 The procedure of the fine tuning automation ........................................ 56
Figure 7.2 Selected components of the physical machining centre ........................................ 57
Figure 7.3 The complete physical machining centre ........................................ 58
Figure 7.4 An underside surface of a marimba bar ........................................ 59
Figure 7.5 The determination of cutter locations ........................................ 60
Figure 7.6 Tool path directions (Plane view) ........................................ 61
Figure 7.7 The achieved tool path and the underside surface ........................................ 61
Figure 7.8 A flowchart of the tool path generation ........................................ 62
Figure 7.9 Step signals and step pulses ........................................ 63
Figure 7.10 X and Z Step Signals ........................................ 64
Figure 7.11 Modified X and Z step signals ........................................ 64
Figure 7.12 A Flowchart of the generation of the tool path signals (1) ........................................ 65
Figure 7.13 A Flowchart of the generation of the tool path signals (2) ........................................ 66
Figure 7.14 A PC-controlled spindle board ........................................ 67
Figure 7.15 Photograph of frequency measurement ........................................ 68
Figure 7.16 Measured frequencies for an aluminum beam ........................................ 70

Figure 8.1 The predictive model testing procedure ........................................ 71
Figure 8.2 Edge finder checking angle plate ........................................ 72
Figure 8.3 Machine setup ........................................ 73
Figure 8.4 CNC cutter finding Z origin ........................................ 74
Figure 8.5 The complete setup for the angle plate ........................................ 74
Figure 8.6 Predicted underside shape from the model (Inverted) ........................................ 75
Figure 8.7 The design of the cutting process on AlphaCam ........................................ 76
Figure 8.8 Simulation of the cutting process on AlphaCam ........................................ 76
Figure 8.9 A complete marimba beam manufactured on the Leadwell V30 CNC machine ........................................ 77
Figure 8.10 The fine tuning procedure ........................................ 79

Figure 8.11 The clamping setup for the self-built CNC machine .............................. 80
Figure 8.12 Measurements of the X and Y origin..................................................... 80
Figure 8.13 Measurement of Z origin ........................................................................ 81
Figure 8.14 Measured bending and torsional natural frequencies of the blank marimba bar ............................................................................................ 82
Figure 8.15 Incremental target $f_1$ vs. Experimental $f_1$................................................ 84
Figure 8.16 Incremental target $f_2$ vs. Experimental $f_2$................................................ 85
Figure 8.17 Incremental target $f_3$ vs. Experimental $f_3$................................................ 85
Figure 8.18 Comparison between the original prediction and the re-prediction........ 87
Figure 8.19 Measured natural bending frequencies of manufactured marimba beam87

Figure A.1: A simple Euler-Bernoulli beam in transverse vibration (Bishop and Johnson 1960, 283)................................................................................ 101
Figure A.2: Deflected position of an element of a uniform beam (Stone, 1992)..... 104
Figure A.3: Forces and moments acting on an element (Stone, 1992) .................... 105
Figure A.4: A simple finite element model of a prismatic beam (Inman 2008, 588) ......................................................................................................... 108

Figure C.1: An illustration of one iteration of Newton_Raphson method (Smith, 1998) ......................................................................................................... 112

Figure G.1: Structure of the fine tuning program ................................................... 123

# List of Tables

Table 1: Calculated positions of nodes for a prismatic bar and a tuned marimba bar ($X_k$ = node locations) (Bork, 1994) .......... 18

Table 2: Frequency ratios of the first three transverse modes calculated and measured in a uniform rectangular beam (Bustamante, 1991) .......... 38

Table 3: The first three transverse modes calculated and measured in marimba bars39

Table 4: The incremental target frequencies and the experimental natural frequencies for each cut .......... 48

Table 5: The target frequencies and the re-predicted target natural frequencies .......... 49

Table 6: Inputs and original underside shape prediction .......... 75

Table 7: Underside frequency results for wooden beam .......... 77

Table 8: Underside shape prediction results for wooden beam (Entwistle and McGrechan, 2007) .......... 78

Table 9: Underside shape prediction results for aluminum beam (Entwistle and McGrechan, 2007) .......... 78

Table 10: Inputs of the fine tuning Test .......... 82

Table 11: Detailed fine tuning results .......... 83

Table 12: Summarized incremental target frequencies and the experimental natural frequencies for the fine tuning process .......... 84

Table 13: Results of the fine tuning test .......... 88

Table 14: Fine tuning results obtained from other tests .......... 88

Table F.1 Properties of wood (27/05/2011) .......... 117

Table F.2 Fine tuning results (27/05/2011) .......... 117

Table F.3 Detailed records of experiment (27/05/2011) .......... 118

Table F.4 Final results of the fine tuning process (27/05/2011) .......... 118

Table F.5 Properties of wood (06/06/2011) .......... 119

Table F.6 Fine tuning results (06/06/2011) .......... 119

Table F.7 Detailed records of experiment (06/06/2011) .......... 120

Table F.8 Final results of the fine tuning process (06/06/2011) .......... 120

## Chapter 1: Introduction

Percussion instruments produce musical sounds when excited by the impact of a mallet, and the percussion family can be classified by the method of sound production. For example, idiophones generate sounds through the vibration of their whole body, whereas membranophones (e.g. drums) produce sounds when their membrane is struck. In particular, idiophones can be further classified by their materials; marimbas consist of wooden bars and vibraphones use aluminum bars. For this thesis, marimba bars as shown in Figure 1.1 are studied.

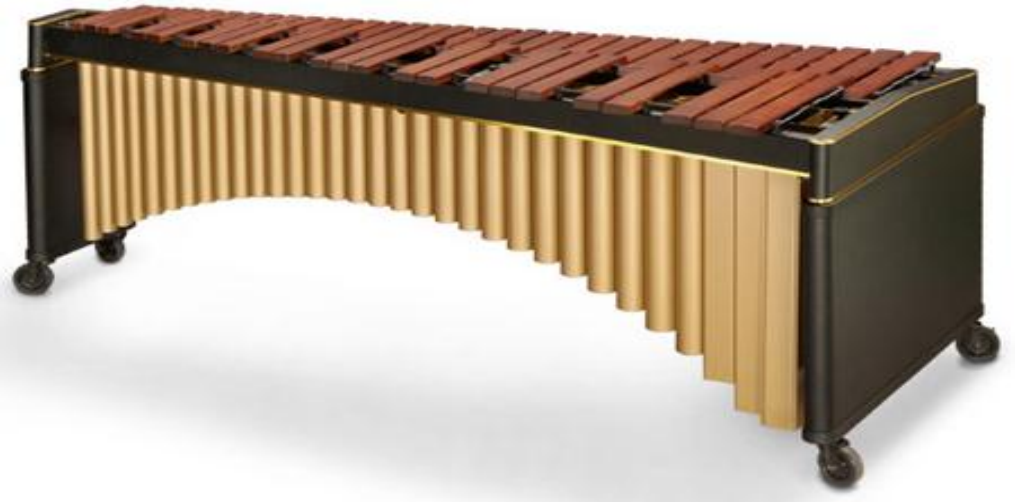


Figure 1.1 Marimba (University of Notre Dame, 2011)

Marimbas use free-free beams to generate musical sounds, which are dependent on the natural bending vibrations of beams. The tonal quality of this instrument is achieved by tuning the second and third natural bending frequencies in relation to the fundamental natural frequency. For example, the frequency ratio for marimba bars is 1:4:9.8. (Bork, 1982) However, free-free beams with uniform cross section produce non-harmonic overtones as the first three natural bending frequencies of the beams have the frequency ratios ($f_1$:$f_2$:$f_3$) approximately 1.00:2.76:5.40, which is not musically pleasing. (Bustamante, 1991) Therefore, tuning harmonic overtones becomes the primary task when making marimba bars.

To achieve the desired frequency ratios the geometry of the beams is modified by making them non-uniformly undercut as shown in Figure 1.2. The upper surface of the beam is left flat as it is used by players to strike with the mallet, and the underside of it has material removed to form an arch shape so that the lowest three transverse vibration modes can be simultaneously tuned to the fundamental and relative (first and second harmonic) pitches, namely the first two overtones give the harmonic relationship with the fundamental natural frequency.

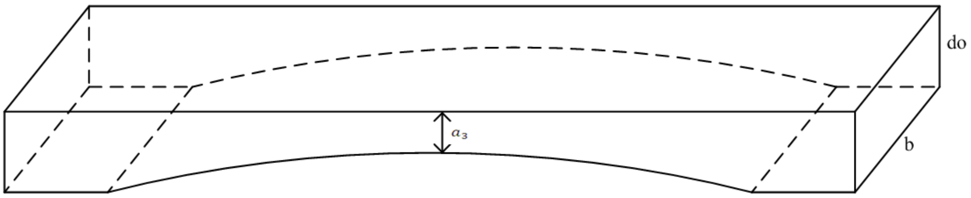


Figure 1.2 A marimba beam

Traditionally the underside material is incrementally removed from different locations of the marimba bar by experienced craftsmen until the correct result is achieved. However, removing the material at any location affects all three bending natural frequencies, and as such the tuning process must be approached from above as once the frequencies become lower than the desired natural frequencies, correction is no longer possible. This gives problems for industrial volume production as large amount of labour time and costs would be required to tune each marimba bar. Although some work has been carried out on predicting the underside shape of marimba bars, success in accurate tuning has been limited due to the complicated nature of wood used in marimba bars.

For this thesis, an automatic multi-modal tuning system has been developed, and by using this system, the underside shape of marimba bars, which result in the desired frequency ratio ($f_1$:$f_2$:$f_3$), is predicted. An accurate underside shape predictive model was firstly developed to predict the natural bending frequencies of marimba bars, and a search algorithm was then implemented to find the correct geometry of the beam that responds with the target natural bending frequencies. The accuracy of the predictive model is mostly dependent on the input data, including the elastic and inertia properties of the material and the dimensions of the wood blank. The assumptions such as homogeneity and isotropy of the material make this model simpler, which is broadly true for metallic beams used in vibraphones. However, the beam material used for marimbas (wood) is non-homogeneous and anisotropic. Thus, this predictive model cannot produce the correct dimensions of the underside shape as it does for vibraphones.

The unknown non-homogeneity and anisotropy of the wood was not included in this prediction. Rather, a fine tuning program was developed to correct the error caused by the unavoidable uncertainties of the test material during marimba bar tuning, which is a major part of this thesis and an improvement on previous work. A CNC machine tool was then adapted to mill the specified underside shape from a wood blank, and a physical machining centre, combining the CNC machine tool, the underside shape predictive model, the fine tuning program, the hardware controlling program and the frequency measuring program, was built as a prototype of the automatic multi-modal tuning system. In more detail, the work included:

1) Develop an accurate predictive model, which is capable of producing the natural bending frequencies of a non-uniform beam.
2) Use the obtained model to devise a scheme to adjust the underside shape so that the frequency ratios ($f_1$, $f_2$ and $f_3$) can be simultaneously tuned to their desired (target) frequencies.
3) Adapt a CNC machine tool to mill the specified underside shape from a wood blank to achieve the goal of automatic tuning.
4) Devise a scheme to incrementally approach the desired frequencies from above, thereby correcting the unknown non-homogeneity and anisotropy of the wood.
5) Assess whether the scheme is sufficiently accurate to be used for production tuning of marimba bars.

This page intentionally left blank.

# Chapter 2: Literature Review

In this chapter, the works that have been previously completed in similar research are presented in Section 2.1. The agreement and disagreement among the selected literature are discussed in Section 2.2, and the implications of the present work are shown in Section 2.3.

## 2.1 Selected Literature

The works that have been previously completed in similar research were selected in this section based on the tasks presented in the previous chapter. Bustamante (1991) addressed the problem of determining the optimal dimensions of the underside shape, and Suits (2001) applied the Timoshenko theory to model non-prismatic beams and compared the experimental results with those based on the Euler-Bernoulli model. Ostergaard (1997) compared the predictive results with the experiments based on three geometries (constant cross section, parabolic undercut and rectangular undercut) by using finite element model. Bretos, Santamaria and Moral (1998) determined these marimba geometries with similar methods that Ostergaard (1997) performed and further analysed the influence of the elastic properties of wood on the resulting natural frequencies. Using Stone's (1992) Timoshenko-receptance model[2], Entwistle and McGrechan (2007) applied receptance sub-structuring methods to produce accurate predictions of undercut shapes.

### 2.1.1 Non-uniform beams with harmonically related overtones for use in percussion instruments (Bustamante, 1991)

This paper addressed the problem of determining the optimal dimensions of the underside shape of marimba beams so that the frequencies of mode two and mode three of transverse motion are harmonically related with that of the fundamental (mode one). The results plotted in this paper were of the relative frequency shift for different modes of vibration as a function of the amount of material removed at different locations along the beam. Thus, the paper developed a numerical approach in which a search for optimal underside shapes was performed to tune the first two overtones harmonically with the fundamental.

[2] See Chapter four for the description of the Timoshenko-receptance model

The calculation of the natural frequencies of non-prismatic beams was carried out in two steps. Firstly, a finite element model was used to obtain accurate approximations to the eigenvalues and eigenvectors of the Euler-Bernuolli equation. In step two, the effect of the Timoshenko equation was included as a perturbation on the eigenvalues of the Euler-Bernoulli equation. The calculations performed with and without taking into account the effects of rotary inertia and shear stress were compared against measurements. The comparison showed that including the effects of rotary inertia and shear stress results in a better prediction of the underside shape of marimba bars.

All the beams that were produced for experiments had their first two overtones tuned harmonically (the largest tuning error is seven cents[3]). The dimensions of the resulting underside shape are close to those predicted by the Timoshenko beam model in the numerical study, which means that rotary inertia and shear stress, which are neglected in the Euler-Bernoulli model, play an important role in the vibration behaviour of beams used in idiophone bars (e.g. marimba bars). This paper showed the need to include rotary inertia and shear effects in the present work.

### 2.1.2 The receptances of beams, in closed form, including the effects of shear and rotary inertia (Stone, 1992)

Receptance, being the harmonic response to an applied harmonic force, is sometimes called “mechanical admittance”. Techniques exist that allow sub-sections of a structure, of which the receptances are known, to be joined and the receptances of the complete structure determined. At frequencies corresponding with natural modes, the total receptance is unbounded and such frequencies can be identified. Receptances of beams derived from the Euler-Bernoulli equations may have limitations on the accuracy that result in significant errors under some circumstances. This paper derived the receptances of beams including the effects of rotary inertia and shear stress and compares the results with those based on the Euler-Bernoulli model.

Experimental results for the first four modes of seven different shafts were obtained and the errors due to omitting the effects of rotary inertia and shear stress are apparent. Close agreement between the predictions obtained using the derived

[3] 100 cents = 1 semi tone ($f_2$:$f_1$= 1.0595:1)

receptances and both the experimental and finite element results indicated the accuracy of those receptances. It was shown that a great improvement had been achieved over those based on the Euler-Bernoulli equations, especially for the high-frequency range. This paper showed again the need to include rotary inertia and shear effects.

### 2.1.3 Practical tuning of xylophone bars and resonators (Bork, 1994)

This paper presents the bending wave differential equation of the Euler-Bernoulli model and the solution by the application of the Runge-Kutta-Nystorm method. The influence on the vibrational behaviour due to the undercut of the bar was studied, and a computer program was developed to predict this effect. It was shown that the influence on a specific mode is greatest where the bending moment is maximum. The influence of a small thinning on the modal frequencies of the bar was investigated with respect to dependence on the location of the thinning. The influence of the thinning is strong where the corresponding bending moment is maximum and it vanishes in the regions of weak moment. In the case of xylophone bars, the influence of the cut is the most pronounced in the thin middle region since here the relative decrease in the thickness is greater than the ends.

### 2.1.4 Tuning process of xylophone and marimba bars analysed by finite element modelling and experimental measurements (Ostergaard, 1997)

This paper calculated the natural frequencies of a constant section bar and two types of undercut bars using a finite element model and compared the results with experimental measurements. The work offered a brief report on the FEA calculations of three cases of bars with different shapes using the ABAQUS software (developed by Hibbit, Karlsson and Sorrenson). Two different tunings were achieved for the two first harmonics, the 1:4 relationship for marimba bars and the 1:3 for xylophone bars. Regarding the mechanical properties required by the FEA (especially the elastic constants and density) the bars' properties were those of a special kind of wood, the *lophera alata azobe,* nine elastic constants being provided to calculate the corresponding frequencies.

For the constant section bars, the resulting differences between the experimental and numerical results were less than 3% for all frequencies. For the case of the parabolic undercut, a double-octave relationship (1:4) for the first and second natural frequencies for marimba bars was produced. The experimental result of the fundamental frequency was 1.2% lower than the prediction, and less than 2.76% differences for all the other frequencies. The rectangular undercut produced a relationship of an octave and fifth between the first two natural frequencies for xylophone bars, with less than 1.1% difference for all frequencies between the predicted and experiment results. The agreement between the predicted and experimental results was good, and the finite element model can be considered as an accurate tool for modelling these musical instruments.

### 2.1.5 Finite element analysis and experimental measurements of natural eigenmodes and random responses of wooden bars used in musical instruments (Bretos, Santamaria and Moral, 1998)

This paper studied the vibrational properties of wooden bars for musical instruments using finite element analysis. The natural frequencies of a constant section and undercut bars were calculated. The influence of each elastic parameter of the wood on natural frequencies was studied, and the influence of an undercut on the vibrational behaviour of wooden bars was described. The achievement of harmonic relationships between frequencies was studied for different undercut shapes. All the theoretical results were compared with experimental results. As the procedure of the second part of this paper was very similar to the procedure and results made by Ostergaard (1997), it will not be described in detail again in this section.

It has been shown that certain of the elastic parameters do not affect any of the natural frequencies. These elastic parameters are the Young's modulus across the grain ($E_2$, $E_3$), the three Poisson ratios ($\nu_{12}$, $\nu_{13}$, $\nu_{23}$) and shear modulus across the grain ($G_{13}$). When their characteristic values in *lophera alata azobe* (a kind of wood) are changed from their characteristic values by 50%, the natural frequencies undergo negligible change. On the other hand, the Young's modulus and shear modulus along the grain $E_1$, $G_{12}$ and $G_{13}$, and the density produce significant changes in natural frequencies.

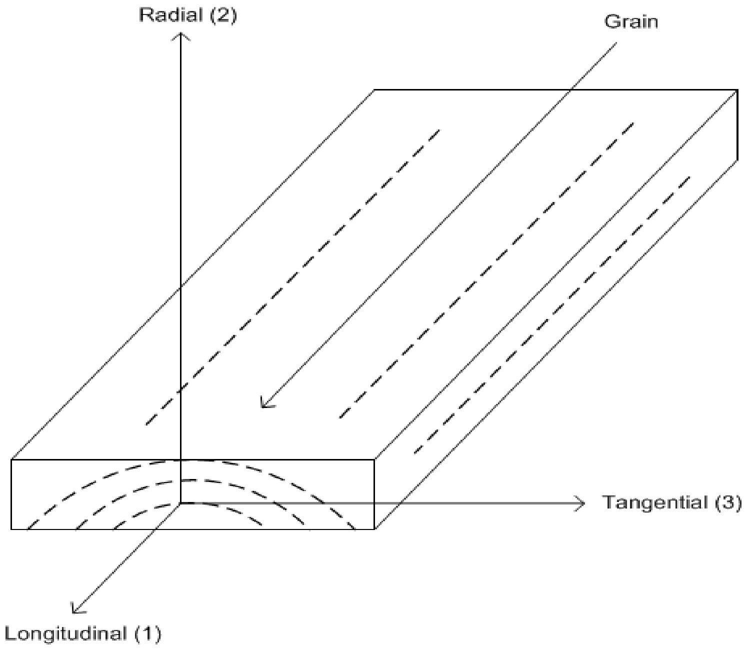


Figure 2.1 A wood blank

Regarding the density and the Young's modulus along the grain, natural frequencies increase when $E_1$ increases or the density decreases. Regarding the shear modulus along the grain, $G_{13}$ does not affect the transverse natural frequencies and only has a clear effect on torsional frequencies. For the other shear modulus, $G_{12}$, the impact of this on the natural frequencies was compared with that related to $E_1$. It was shown that the effect of $E_1$ is stronger in low frequencies (when its values decreases by 20% the calculated natural frequencies change for around 10%), and regarding to $G_{12}$, the opposite situation occurs. Therefore, it can be concluded that the density of wood and the Young's modulus ($E_1$) and the shear modulus ($G_{12}$) along the grain are important for natural frequency determination.

### 2.1.6 Basic physics of xylophone and marimba bars (Suits, 2001)

The equations for transverse vibrations of uniform rectangular bars are presented in this paper using the Euler-Bernoulli model and the Timoshenko model. It could be seen using a comparison between theory and experiment that the Euler-Bernoulli result is not accurate for wood bars of a size comparable to those used musically. The comparison between these theories and experiments with bars made of wood and of a size comparable to those used in marimba bars shows that the Timoshenko model predicts natural frequencies more accurately. The Timoshenko model was developed to model the non-prismatic bars, and the normal mode behaviour in these non-prismatic wooden bars was predicted using a simple finite element model. It was shown that accurate predictions for the non-prismatic bars were achieved when the dynamic elastic constants were appropriate.

The Euler-Bernoulli results for the mode frequencies of a uniform bar are independent of bar width, and the Timoshenko result makes the same prediction. The Euler-Bernoulli result predicts that the normal mode frequencies are proportional to the reciprocal of bar length squared ($f \propto {}^{1}/_{l^2}$), and it is not accurate for shorter bars, namely with a length less than approximately ten times the bar's depth. The Euler-Bernoulli model predicts no length dependence for higher frequency modes, however, a rather significant length dependence is observed and is correctly predicted by the Timoshenko model. Therefore, the Euler-Bernoulli model is not suitable for the determination of the higher frequencies of the overtones, and the Timoshenko model appears to give improved predictions.

### 2.1.7 Designing idiophones with tuned overtones (Legge and Petrolito, 2007)

This paper developed the numerical technique (finite element analysis) to determine the mode shapes arising in musical structures. Where the numerical optimization strategies have been adopted, the temptation has been to emulate the maker in the choice of optimising function to optimise the frequencies of the modes. Such an approach could determine how close to the desired frequency each mode should be and which mode is more important to get right. The design goal was to produce a structure that responds with specified frequencies for specified modes of vibration. Hence, the required frequencies are critical for this design problem and a solution strategy was required, where the frequencies are the constraints. The constrained optimization problem is uniquely specified by three things, namely, the geometry parameters, the optimisation function and the constraint function, and different solutions might be obtained by varying any of these. Such an approach allows significant freedom in the choice of optimising function.

Firstly, the geometry of the structure that responds in the pre-determined manner is determined, and it is described by a number of parameters. These parameters are the primary unknowns for the problem and will be varied to produce the required outcome. Secondly, the optimisation function will optimise the frequency of each critical mode, and this approach leaves a wide range of possible optimising functions. Thirdly, constraint functions describe what must be obtained for an acceptable solution, and the required frequencies in this approach are defined as the constraints. In this paper, a mathematical approach has been outlined to design constraints (the required frequencies), and has other parameters optimised.

### 2.1.8 Geometric shape identification for multi-mode tuning of percussion instrument bars (Entwistle and McGrechan, 2007)

This paper models the beam free vibrations and accurately predicts the non-prismatic shape that results in the simultaneous tuning of the first three frequencies of marimba bars. It presents a model for marimba bar tuning with two features. Firstly, the non-prismatic beam is modelled using the receptance sub-structuring technique, the Timoshenko beam receptances being used. Secondly, the geometrical shape of the undercut is not prescribed but rather a mathematical curve with three adjustable parameters is used. A three dimensional Newton-Raphson solution technique solves the three variable parameters such that the lowest three required natural bending frequencies are achieved. This algorithm locates the geometric shape of the undercut curve that satisfies the multi-mode frequency requirements.

The combination of receptance sub-structuring methods and use of the Timoshenko beam receptances provided accurate predictions of the undercut shapes required to meet the objective transverse vibration frequency ratios of marimba bars, and the fundamental frequencies were very accurate compared with experiment. The sensitivity of the frequencies to dimensional variations is reported in this paper, and manufacture of aluminium and wooden bars shows that the predictions are very accurate and suitable for the basis of automated manufacture. What is more, the experimental results have shown that the technique could be used to provide initial shaping of marimba bars in wood prior to manual fine tuning. Automated initial and final tuning might be possible.

## 2.2 A Summary of the Major Agreements and Disagreements in the Selected Literature

In this section, the agreement and disagreement of the selected literature are further summarized so that the conclusions drawn by the previous papers can be used to guide this thesis in the mathematical model, the fine tuning program and the mechanical property measurement, etc.

### 2.2.1 Agreement on the Timoshenko Model

The Timoshenko model that takes into account the rotary inertia and shear stress has been verified to be more appropriate compared to the Euler-Bernoulli model. Especially, the Euler-Bernoulli model is not suitable for the determination of the higher frequencies of the overtones, and the Timoshenko model appears to give improved predictions.

Bustamante (1991) determined the optimal dimensions of the underside shape of marimba bars by using the Euler-Bernoulli model and the Timoshenko model, and compared the experimental results with the predictions. The experimental results showed that the dimensions of the resulting underside shape are closer to those predicted by the Timoshenko beam model than the ones predicted by the Euler-Bernoulli model.

What is more, Suits (2001) showed that the Euler-Bernoulli model predicted that the normal mode frequencies are proportional to the reciprocal of bar length squared ($f \propto {}^{1}/_{l^2}$), but it is not accurate for shorter bars, namely with a length less than approximately ten times the bar's depth. On the other hand, the Euler-Bernoulli model predicted no length dependence for the higher frequency modes. However, a rather significant length dependence was observed and was correctly predicted by the Timoshenko model.

### 2.2.2 Agreement on the Receptance Model

The combination of the receptance sub-structuring method and the Timoshenko model, which was developed by Stone (1992) and Entwistle and McGrechan (2007), has been verified to be very accurate in relation to the underside shape prediction.

This Timoshenko-receptance model[4] produced 0.8 Hz (0.3%), 28Hz (3.2%) and 40Hz (1.9%) error for the lowest three natural frequencies compared with the experimental results. On the other hand, the experimental results that were obtained using the finite element method by Bustamante (1991), Ostergaard (1997) and Suits (2001) produced the fundamental frequency 1.2% lower than the prediction and less than 2.76% differences for all other frequencies.

It can be seen that both the Timoshenko-receptance model and the finite element model produce good results for marimba bar prediction. However, the experimental result that the receptance sub-structuring model achieved was better than the result that was obtained by the finite element model for the prediction of the fundamental natural frequency. Furthermore, the receptance model is capable of providing accurate mode shapes, whereas the finite element model only gives accurate amplitudes at the nodes.

### 2.2.3 Disagreement on the Elastic Properties

It can be seen that the combination of the receptance sub-structuring method and the Timoshenko model would produce accurate results based on the precondition that the elastic constants were accurately provided to the model. However, the literature selected in this chapter has not agreed on the method of accurately solving the elastic constants of wooden bars.

Ostergaard (1997) has completed the FEA calculations of three cases of bars with different shapes (a constant section bar and two types of undercut bars) using the ABAQUS software (developed by Hibbit, Karlsson and Sorrenson), in which nine elastic constants were provided to the model, and produced good results. Bretos, Santamaria and Moral (1998) followed Ostergaard (1997) by investigating the influence of each elastic constant on the natural frequencies by using the same software, material property data and the mathematical model. It was concluded that the density of wood, the Young's modulus ($E_1$) and the shear modulus ($G_{12}$) along the grain play an important role in predicting the natural bending frequencies of

[4] The underside shape predictive model (described in Chapter 5) was developed based on the Timoshenko-receptance model.

marimba bars. On the other hand, other elastic properties would not affect natural frequencies much. However, the most appropriate method of solving the Young's modulus ($E_1$) and the shear modulus ($G_{12}$) along the grain was not described in any selected literature.

Other authors have simplified their models by basing on Young's modulus ($E_1$) and the shear modulus ($G_{12}$) along the grain as their inputs. Suits (2001) set the values of *E=14.7GPa* and *G=E/10* to achieve acceptable results for oak bars. Entwistle and McGrechan (2007) achieved accurate results for a 220Hz marimba bar with only 0.8 Hz (0.3%), 28Hz (3.2%) and 40Hz (1.9%) error for the lowest three natural frequencies. The elastic properties of the tested blank were inferred by measuring the vibration frequencies of a prismatic beam and adjusting E and G in the Timoshenko-receptance model until they agreed. The values obtained by using this method were the best average values that produced the best predictive results, however, errors cannot be avoided.

### 2.3 Implications for Present Work

This thesis continues to develop an automatic multi-modal tuning system by following the work of Entwistle and McGrechan (2007), in which the Timoshenko-receptance model and the Newton-Raphson search algorithm were applied, as the basis of this tuning system. The design procedure was similar to the work of Legge and Petrolito (2007). The geometry of the undercut of a marimba bar that responds with specified natural bending frequencies was determined, and it was described by a number of parameters (three shape coefficients[5]). These parameters are the primary unknowns and would be varied to produce the required natural frequencies.

It was shown that the underside shape predictive model developed by Entwistle and McGrechan (2007) was accurate for predicting aluminium beams, but it cannot avoid errors when tuning wooden beams. Thus, for this thesis the underside shape predictive model will be further modified so that an automatic multi-modal tuning system can be developed for wooden beams.

[5] See Section 5.3 for the description of the shape coefficients

The non-homogeneity and anisotropy of wood cannot be accurately measured. This thesis develops a fine tuning program, which corrects the error caused by the unavoidable uncertainties of the test material throughout the fine tuning process. A physical machining centre was developed so that the tests could be carried out automatically.

This page intentionally left blank.

# Chapter 3: Marimba Bar Tuning

Marimba bars are traditionally tuned by experienced craftsmen by incrementally removing their underside material from different locations until the correct result is achieved. However, removing the material at any location would affect all three bending natural frequencies, and such tuning method must be approached from above as once the frequencies become lower than the desired natural frequencies, correction will not be possible. This gives problems for industrial volume production as large amounts of labouring time and costs would be required.

Although some work (described in Chapter 2) has been carried out on predicting the underside shape of marimba bars, it has still been concerned with the tuning accuracy due to the unavoidable uncertainties of the test material. Thus, manual fine tuning will still be required. Therefore, the principles of tuning marimba bars are described in this chapter, and they are presented in three sections: the support and end conditions for prismatic and marimba bars, the influence on the vibration behaviour due to the underside shape of marimba bars and the determination of the tuning intervals.

## 3.1 Support and End Conditions

Marimbas use free-free beams to generate musical sounds, which depend on the natural bending vibrations of beams. For a bar aligned in the x-direction, the vibration behaviour can be described by the Euler-Bernoulli differential equation[6]:

$$\rho A \frac{\partial^2 y}{\partial t^2} + EI \frac{d^4 y}{dx^4} = 0$$

where $y$ is the displacement of the bar from the mean position. For a free-free bar of length $l$, both ends of which are freely movable, the following conditions are valid at the ends for the bending moment $M$ and the shearing force F:

$$F(x=0) = F(x=l) = 0;\ M(x=0) = M(x=l) = 0 \qquad \text{(3.1a, 3.1b)}$$

[6] See Appendix A.1 for the derivation of the Euler-Bernoulli differential equation.

For prismatic beams, the natural frequencies can be calculated based on the Euler-Bernoulli equation (as shown in equation B.1):

$$\omega_n = 2\pi f_n = (\beta_n l)^2 \sqrt{\frac{EI}{\rho A l^4}}$$

where $\beta_n l$ is the weighted frequency[7] and $\omega_n$ is the resulting natural frequency. Other mechanical quantities, such as M, F, y and $y'$ (the derivative of y), are represented in Figures 3.1 and 3.2 for five modes for the prismatic bar and the marimba bar. These two figures show both the mode shapes and the points of maximum and minimum bending, which are important for marimba bar tuning. The positions of the maximum values of a curve correspond to those of the zero points of the curve arranged underneath. Thus, to find the points of maximum bending moment, the zero points of the shearing force can be determined. (Bork, 1994)

The free-free bar should be supported on its strings at the nodes of the bar's fundamental mode. This can be achieved at the zero points of the displacements y in Figures 3.1 and 3.2. An example is shown in Table 1, the supporting points of a marimba bar are located at the 19.29% and 80.34% respectively of the bar length.

| n | $(f_n/f_1)_{prismatic}$ | $(f_n/f_1)_{tuned}$ | $(X_k/l)_{prismatic}$ | $(X_k/l)_{tuned}$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 0.2241 | 0.1929 |
| | | | 0.7754 | 0.8034 |
| 2 | 2.7565 | 3.9584 | 0.1322 | 0.1331 |
| | | | 0.5 | 0.4999 |
| | | | 0.8678 | 0.8672 |
| 3 | 5.404 | 10.6726 | 0.0945 | 0.1064 |
| | | | 0.3558 | 0.3731 |
| | | | 0.6441 | 0.6274 |
| | | | 0.9054 | 0.8962 |

Table 1: Calculated positions of nodes for a prismatic bar and a tuned marimba bar ($X_k$ = node locations) (Bork, 1994)

[7] For a free-free prismatic beam, $(\beta_1 l) \cong 4.73, (\beta_2 l) \cong 7.853\ and\ (\beta_3 l) \cong 10.996$ (Inman, 2008)

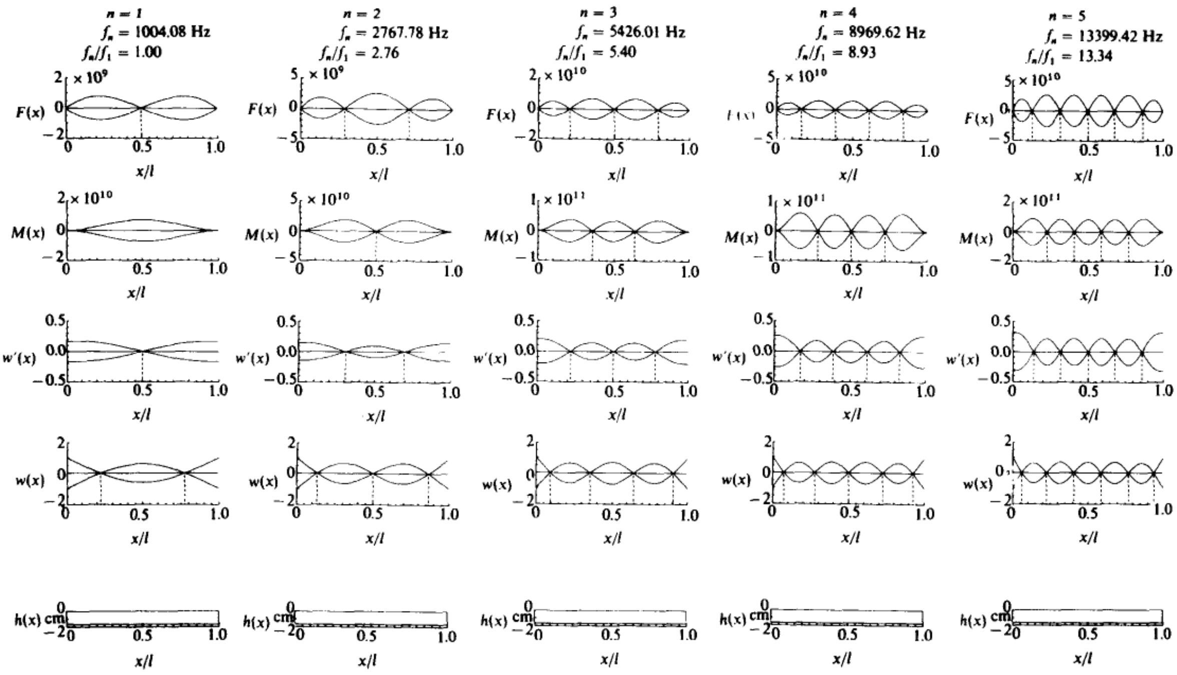


Figure 3.1 Spatial distribution of shear force, bending moment and displacement for the first five modes of a prismatic bar (after Bork, 1994)

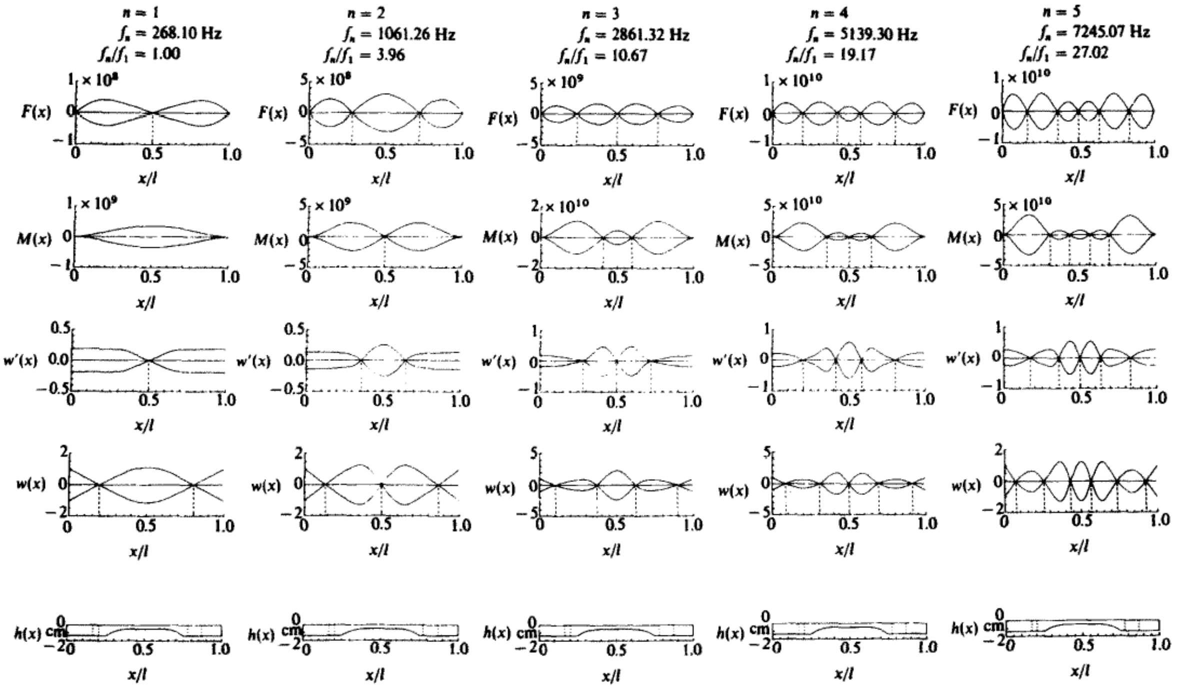


Figure 3.2 Spatial distribution of shear force, bending moment and displacement for the first five modes of a tuned marimba bar (after Bork, 1994)

## 3.2 Influence of the Vibrational Behaviour due to the Undercut Shape

When a section is cut out of the underside of a prismatic bar, both the mass ($m = \rho Al, A = bd$) and the flexural rigidity would be reduced. The second moment of area and the flexural rigidity can be described as:

$$I = \frac{bd^3}{12};\ B = EI \qquad (3.2, 3.3)$$

where $b$ is the width of the beam, $d$ is the depth of the beam and $E$ is the Young's modulus of the material. Thus, according to equations 3.2 and 3.3 the flexural rigidity is proportional to the bar thickness $d^3$, whereas $\rho A$ is proportional to $d$. As a result, a reduction in thickness is connected with a reduction in natural frequencies. Figures 3.1 and 3.2 show the spatial distribution of shear force, bending moment and displacement of the first five modes of a prismatic and a tuned marimba bar. It can be seen that the influence of the cut on a specific mode is greatest where the bending moment becomes maximum. For a free-free beam it is expected that the moment and the shear forces will disappear at bar ends, and a reduction in the bar thickness, at either end, leads to a rise in frequency for all modes as only the decrease in mass is important here. (Bork, 1994)

A computer program was developed by Bork (1994) to predict the effect on the natural frequencies based on the location of the material removal. The influence of a cut on the individual frequencies was investigated with respect to the dependence on the location, and the results for a prismatic beam and a marimba beam are shown in Figures 3.3 and 3.4. The frequency shift of the harmonics is plotted as a function of the location of the cut of constant depth. For a prismatic beam, the regions of negative frequency shift are symmetrically distributed along the bar axis. A comparison with the curves M(x) in Figure 3.2 shows the behaviour anticipated. The influence of the cut is strong where the corresponding bending moment is maximum and it vanishes in the regions of weak moment. At the ends a slight increase in higher frequencies can be observed. In the case of the marimba bar, the influence of the cut is concentrated on the thin middle region, since here the relative decrease in thickness is stronger than at the ends.

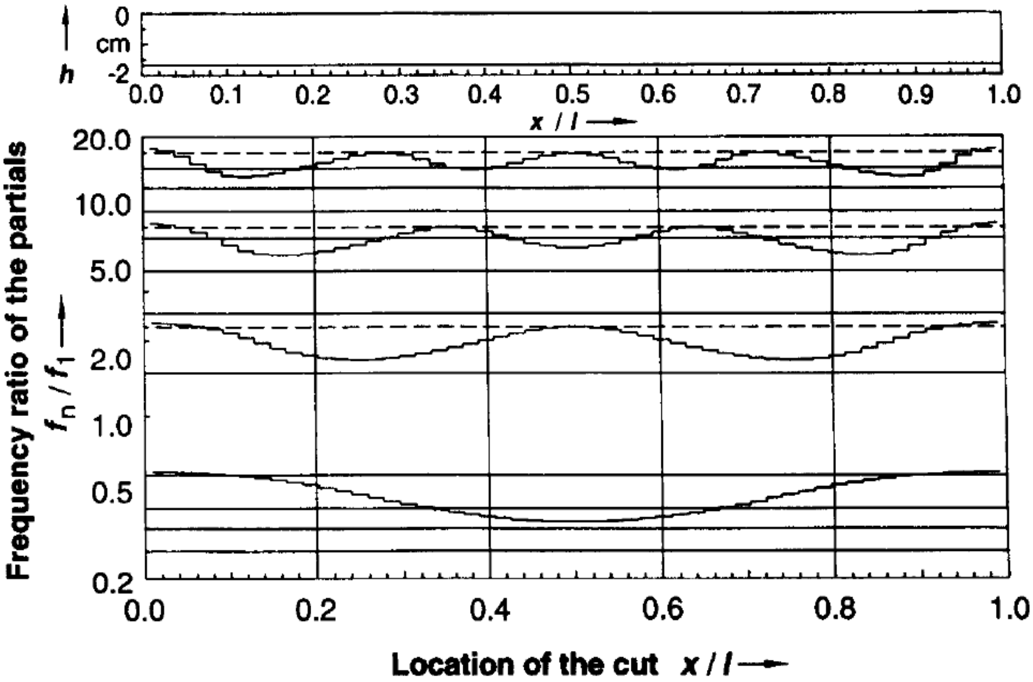


Figure 3.3 Effect of removing material from various locations along a uniform bar (after Bork, 1994)

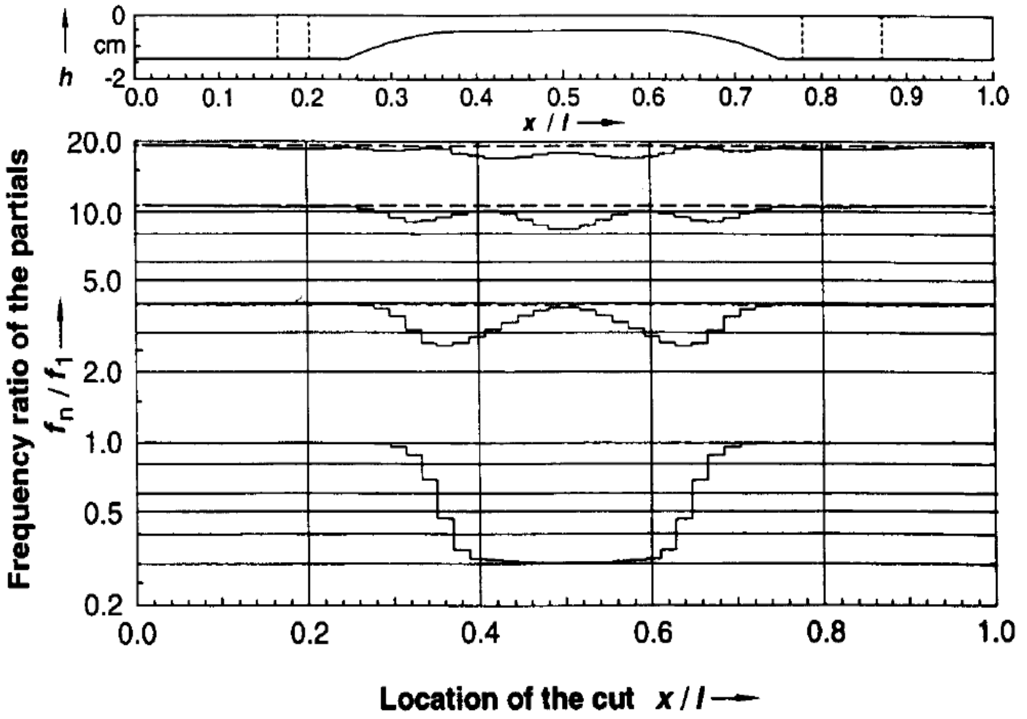


Figure 3.4 Effect of removing material from various locations along a marimba bar (after Bork, 1994)

From the curves in Figure 3.4, a procedure for tuning the second partial can be derived. Starting with a rough shape as shown in the upper part of the figure, the frequencies of the first and second partial should be slightly sharp since only decreases in frequency can be achieved. Next, the second partial is tuned by cutting in regions where the corresponding curve shows its minimum, for example, *x/l* = 0.35 and 0.65. Finally, the fundamental frequency must be tuned by cutting in the centre, where the influence curve for the second shows a minimal frequency deviation. As shown in Figure 3.4 again, there is no region along the bar where only the third partial is influenced, rather, it shows where the bar is thinned in a region (e.g. at the bearing holes). Therefore, a cut was added between the end and the support at a point where little influence appears on the first and second partials in Figure 3.4.

The calculations for this bar were done in the same way as in Figure 3.4 and the results are shown in Figure 3.5, and it can be seen that the expected effect has been achieved. In the region of this additional cut the fundamental frequency is not influenced, the second is slightly influenced, while the third and fourth partials show significant frequency shifts.

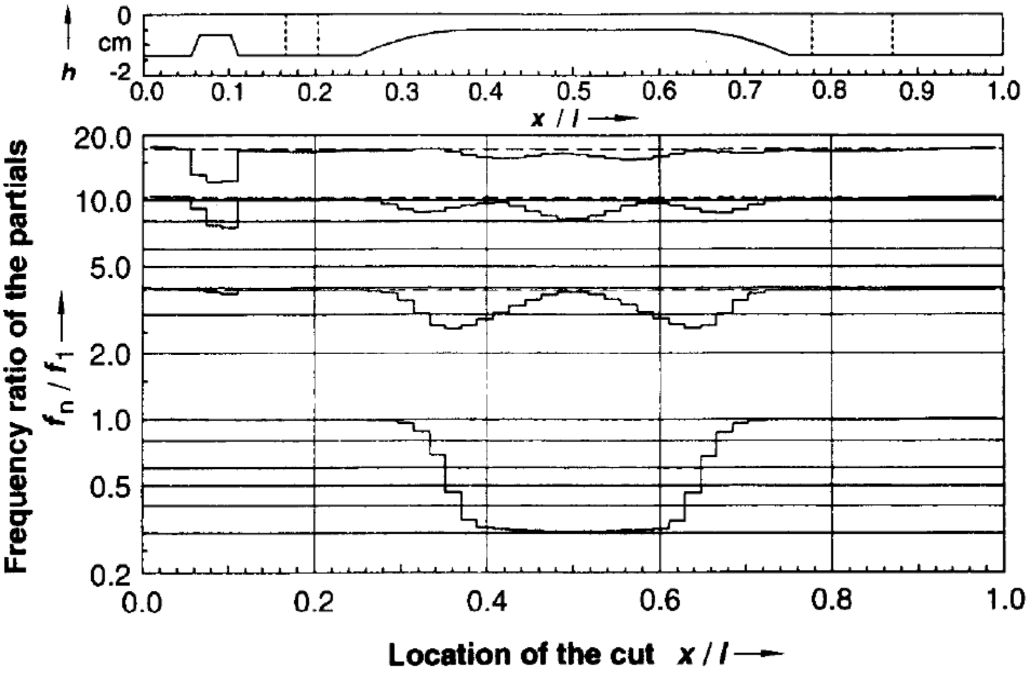


Figure 3.5 Effect of removing material from various locations along a marimba bar prepared for tuning the third partial (after Bork, 1994)

The influence on the vibration behaviour due to the underside shape has been shown in this section, and a computer program that was developed by Bork (1994) has predicted the effect on the natural frequencies based on the material removal as shown in Figures from 3.3 to 3.5. The work of Bork (1994) can be used to guide this thesis in the underside shape prediction for marimba bar tuning as the change of the underside shape of marimba bars would result in the change of the natural frequencies.

In this thesis, a search algorithm that is based on the Newton-Raphson method was developed to find the correct shape coefficients of the underside shape to produce the target natural frequencies. The underside shape was varied to generate resulting natural frequencies during simulation, and the correct results can be achieved when the predicted natural frequencies match the targets.

On the other hand, if the accuracy of the software prediction is limited due to the unavoidable uncertainties of the test material, the fine tuning will still be performed by experienced craftsmen following the principle of marimba bar tuning as described in this section.

## 3.3 Determination of the Tuning Intervals

The next step is to determine which intervals are optimal for tuning marimba bars. For this purpose a synthetic marimba sound produced by a synthesizer was used, containing three partials, where the third partial could be varied in frequency. To make the subjective pitch perception of marimba sounds as clear as possible, tuning of the second partial at a double octave interval to the fundamental (4:1) was used. For the third partial, the selected intervals were two integer ratios 9:1 and 10:1, representing the major second and the major third above the third octave of the harmonic overtone series, the minor third with a ratio of 6:5 relative to the third octave (e.g. 9.6:1 to the fundamental). The three intermediate intervals tuned 50 cent higher than these intervals were also investigated. The test melody used, as shown in the upper part of Figure 3.6, consists of harmonically changing parts in order to put the tones into different musical contexts of major and minor tonality. The chosen intervals all lie in the possible tuning range, and they have been found by experiment. The hearing test with musically well-trained subjects revealed a preference for the interval between the minor and major third as shown in Figure 3.6. A frequency ratio in the middle of the range 9.6 to 10 (e.g. 9.8) appears to be optimal. A tolerance range from 9.715 to 9,881 is acceptable. Thus, for this thesis 9.8 was selected, and the frequency ratios of the first three natural frequencies becomes 1:4:9.8.  (Bork, 1994)

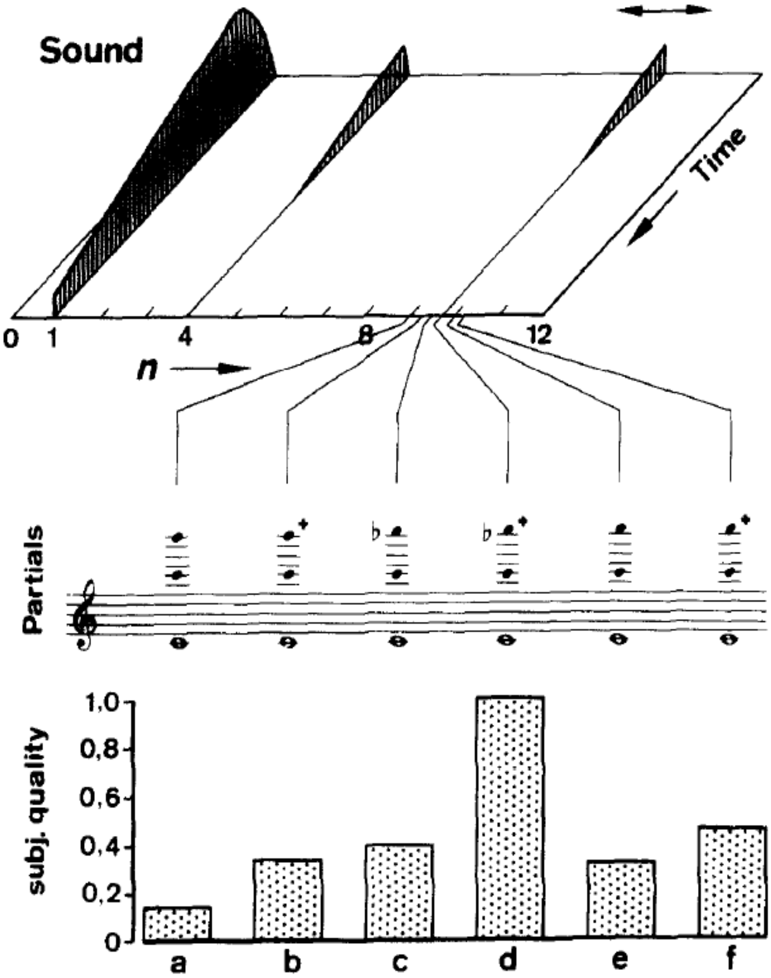


Figure 3.6 Subjective valuation of xylophone sounds
(after Bork, 1994)

# Chapter 4: Timoshenko-receptance Model

The receptance sub-structuring method allows the division of complex systems into sub-systems, and it can be used to add sub-systems together and therefore determine the receptance of the complex system. (Stone, 1992) The combination of the receptance sub-structuring method and the Timoshenko model has provided accurate predictions of the undercut shapes required to meet the objective transverse vibration frequency ratios of marimba bars. The fundamental frequencies were very accurate from the tests that were completed by Entwistle and McGrechan (2007). In this chapter, the receptance sub-structuring method is described in Section 4.1, and the receptance substructuring method is combined with the Timoshenko model to take into account the effects of rotary inertia and shear stress in Section 4.2.

## 4.1 The Receptance Sub-structuring Method

Receptance is a frequency response function given by the ratio of the displacement of a system to a steady state sinusoidal force excitation. (Bishop and Johnson 1960, 17) A complex system (or structure) can be sub-structured into simpler components, whose receptances are known (from predictions or experiments). Thus, the receptances and the natural frequencies of this complex system can be calculated based on the available information.

When a harmonic force $Pe^{j\omega t}$ acts at some point of a dynamical system with some frequency "$\omega$", the point of the application has the displacement $u = Ue^{j\omega t}$. If the equation of motion is linear, it can be written as $u = \alpha Pe^{j\omega t}$, where "α" only depends upon the nature of the system and the frequency "$\omega$" but not upon the amplitude of the force P. The quantity "α" is defined as "the direct receptance at u", and is written as $\alpha_{11}$.

Alternatively, if the displacement $u$ is not at the point where the force is applied, as shown in Figure 4.1, a cross-receptance "α" can then be defined. As the force and the displacement response are described as

$$p = P_2 e^{j\omega t};\ u = U_1 e^{j\omega t} \qquad (4.1.1a, 4.1.1b)$$

Hence,

$$\alpha_{12} = \frac{U_1 e^{j\omega t}}{P_2 e^{j\omega t}} \tag{4.1.2}$$

where the first subscript "1" represents the displacement response at "1" and the second subscript "2" represents the location of force at "2".

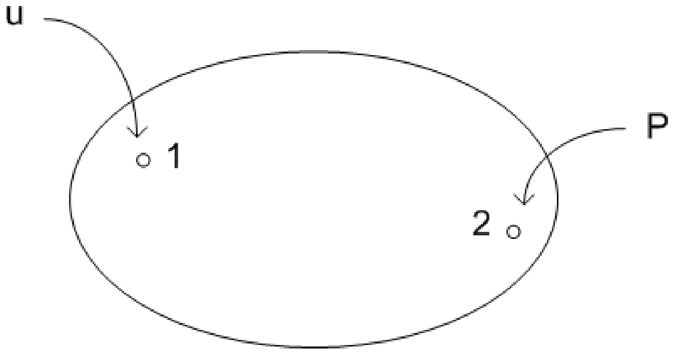


Figure 4.1: A cross-receptance system

The resonant frequencies occur when a displacement is maintained as the applied force tends to zero. (Entwistle and McGrechan, 2007) That is:

$$\alpha_{jk} \rightarrow \infty \tag{4.1.3}$$

where the subscript "*j*" represents the displacement response at "*j*" and the subscript "*k*" represents the location of force at "*k*".

### 4.1.1 Sub-systems Linked by a Single Coordinate

An advantage of the receptance sub-structuring method is that a structure consisting of many sub-sections can be joined (or assembled) together. As shown in Figure 4.2 two sections B and C are joined into section A, and connected by a single coordinate at point 2. The determination of tip receptance of A enables further sections to be joined in the same fashion as C was joined into B. The equations for joining sections at a single coordinate are simple compared to those required for beams and so will be presented first.

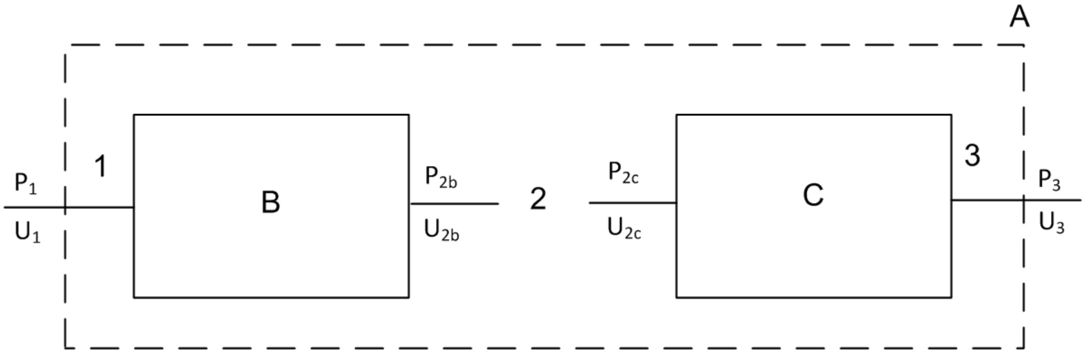


Figure 4.2: A System with two beam sections B and C by a single coordinate

The definition of receptances gives

$$\alpha_{22} = \frac{U_{2a}e^{j\omega t}}{P_{2a}e^{j\omega t}};\ \beta_{22} = \frac{U_{2b}e^{j\omega t}}{P_{2b}e^{j\omega t}};\ \gamma_{22} = \frac{U_{2c}e^{j\omega t}}{P_{2c}e^{j\omega t}} \quad \text{(4.1.4a, 4.1.4b, 4.1.4c)}$$

where α, β and $\gamma$ are the receptances of sub-system A, B and C respectively, and in the steady state the $e^{j\omega t}$ will cancel. Equations of compatibility and equilibrium can be written relating to the internal forces and displacements of the two sub-systems at point 2 in Figure 4.2:

$$P_{2b} + P_{2c} = P_{2a};\ U_{2a} = U_{2b} = U_{2c} \qquad \text{(4.1.5a, 4.1.5b)}$$

Combining equations 4.1.4 and 4.1.5 gives

$$\frac{1}{\alpha_{22}} = \frac{1}{\beta_{22}} + \frac{1}{\gamma_{22}} \qquad \text{(4.1.6)}$$

which gives the direct receptance of system A at point 2 ($\alpha_{22}$). If system A is to be joined to other sub-systems, the tip receptance of system A ($\alpha_{11}$) will be required. Thus, the equation of the equilibrium of forces and the compatibility relationship at point 2 can be written as

$$P_{2b} + P_{2c} = 0;\ U_{2b} = U_{2c} \qquad \text{(4.1.7a, 4.1.7b)}$$

According to the definition of receptances as described in equation 4.1.4 and the linear principle of superposition, the displacement of each sub-system can be re-written in terms of the forces that act on it.

$$U_{2b} = \frac{U_{2b}}{P_{2b}} P_{2b} + \frac{U_{2b}}{P_1} P_1 = \beta_{22} P_{2b} + \beta_{21} P_1 \qquad (4.1.8a)$$

$$U_{2c} = \frac{U_{2c}}{P_{2c}} P_{2c} = \gamma_{22} P_{2c} \text{ (when } P_3 = 0\text{)} \qquad (4.1.8b)$$

The displacement at point 1 can be achieved using the similar manner:

$$U_1 = \frac{U_1}{P_1} P_1 + \frac{U_1}{P_{2b}} P_{2b} = \beta_{11} P_1 + \beta_{12} P_{2b} \qquad (4.1.9)$$

Combining equations 4.1.5, 4.1.6 and 4.1.8 gives

$$P_{2b} = \frac{-\beta_{21} P_1}{\gamma_{22} + \beta_{22}} \qquad (4.1.10)$$

Substituting equation 4.1.10 into 4.1.9 gives the tip receptance of system A

$$\alpha_{11} = \frac{U_1}{P_1} = \beta_{11} - \frac{{\beta_{12}}^2}{\gamma_{22} + \beta_{22}} \qquad (4.1.11)$$

Section A can now become the new section C to which a new section is added to build up the entire structure. The cross receptances $\alpha_{12}$ and $\alpha_{13}$ can similarly be calculated:

$$\alpha_{12} = \frac{\beta_{12}\gamma_{22}}{\beta_{22} + \gamma_{22}};\ \alpha_{13} = \frac{\beta_{12}\gamma_{23}}{\beta_{22} + \gamma_{22}} \qquad (4.1.12a, 4.1.12b)$$

### 4.1.2 Sub-systems Linked by Multiple Coordinates

In this section, the receptance of beams, in which both the deflection and slope are taken into account simultaneously, are modelled. This modelling approach will be used to analyse the vibration of the marimba bars with underside shapes.

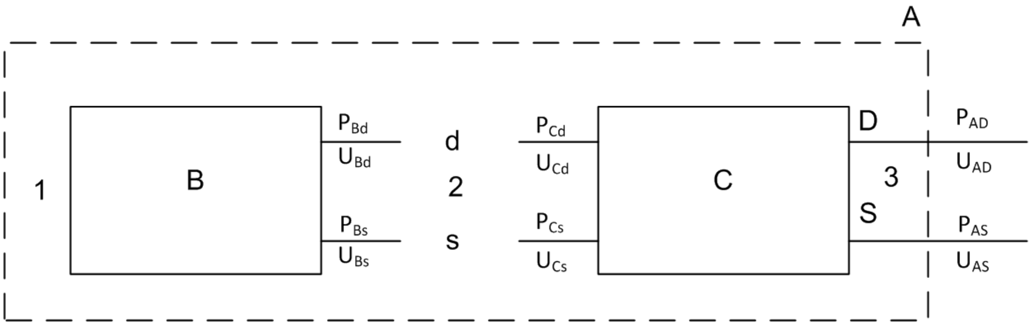


Figure 4.3: A system with two beam sections B and C coupled by two coordinates (Entwistle and McGrechan, 2007)

At the joint in Figure 4.3, four conditions should be fulfilled. Firstly, the internal shear force ($F_B and\ F_C$) on B and C must sum to zero, secondly the internal moment at B and C ($M_B\ and\ M_C$) must sum to zero, thirdly the deflections ($d_B\ and\ d_C$) must be equal, and finally the rotation slopes due to bending ($s_B\ and\ s_C$) must be same:

$$F_B + F_C = 0;\ M_B + M_C = 0 \qquad \text{(4.1.13a, 4.1.13b)}$$

$$d_B = d_C;\ s_B = s_C \qquad \text{(4.1.14a, 4.1.14b)}$$

which gives the following relationships in equations 4.1.15a and 4.1.15b for the system in figure 4.3. (All loads (forces and moments) are P and all displacements (deflection and slope) are U.)

$$\begin{bmatrix} P_{Bd} \\ P_{Bs} \end{bmatrix} + \begin{bmatrix} P_{Cd} \\ P_{Cs} \end{bmatrix} = 0;\ \begin{bmatrix} U_{Bd} \\ U_{Bs} \end{bmatrix} = \begin{bmatrix} U_{Cd} \\ U_{Cs} \end{bmatrix} \qquad \text{(4.1.15a, 4.1.15b)}$$

where the second subscript "$d$" is for displacement and "$s$" is for slope. Based upon the above equations, receptances in the system can be defined as $\alpha_{ij}$ for A, $\beta_{ij}$ for B and $\gamma_{ij}$ for C, of which the first subscript "i" represents the response location while the second subscript "j" represents the location of the excitation. Therefore, further expressed receptance relationships in system A can be obtained as follows

$$\begin{bmatrix} U_{Cd} \\ U_{Cs} \end{bmatrix} = \begin{bmatrix} \gamma_{dd} & \gamma_{ds} \\ \gamma_{sd} & \gamma_{ss} \end{bmatrix} \begin{bmatrix} P_{Cd} \\ P_{Cs} \end{bmatrix} + \begin{bmatrix} \gamma_{dD} & \gamma_{dS} \\ \gamma_{sD} & \gamma_{sS} \end{bmatrix} \begin{bmatrix} P_{CD} \\ P_{CS} \end{bmatrix} \qquad \text{(4.1.16a)}$$

$$\begin{bmatrix} U_{Bd} \\ U_{Bs} \end{bmatrix} = \begin{bmatrix} \beta_{dd} & \beta_{ds} \\ \beta_{sd} & \beta_{ss} \end{bmatrix} \begin{bmatrix} P_{Bd} \\ P_{Bs} \end{bmatrix} \tag{4.1.16b}$$

$$\begin{bmatrix} U_{AD} \\ U_{AS} \end{bmatrix} = \begin{bmatrix} \gamma_{DD} & \gamma_{DS} \\ \gamma_{SD} & \gamma_{SS} \end{bmatrix} \begin{bmatrix} P_{AD} \\ P_{AS} \end{bmatrix} + \begin{bmatrix} \gamma_{Dd} & \gamma_{Ds} \\ \gamma_{Sd} & \gamma_{Ss} \end{bmatrix} \begin{bmatrix} P_{Cd} \\ P_{Cs} \end{bmatrix} \tag{4.1.16c}$$

$$\begin{bmatrix} U_{AD} \\ U_{AS} \end{bmatrix} = \begin{bmatrix} \beta_{DD} & \beta_{DS} \\ \beta_{SD} & \beta_{SS} \end{bmatrix} \begin{bmatrix} P_{AD} \\ P_{AS} \end{bmatrix} \tag{4.1.16d}$$

Abbreviating the equations from 4.1.15a to 4.1.16d in an abbreviated form, where bold font denotes a matrix.

$$\boldsymbol{P_B} + \boldsymbol{P_C} = \boldsymbol{0};\ \boldsymbol{U_B} - \boldsymbol{U_C} = \boldsymbol{0} \tag{4.1.17, 4.1.18}$$

$$\boldsymbol{U_C} = \boldsymbol{\gamma_1 P_C} + \boldsymbol{\gamma_2 P_A};\ \boldsymbol{U_B} = \boldsymbol{\beta P_B} \tag{4.1.19, 4.1.20}$$

$$\boldsymbol{U_A} = \boldsymbol{\gamma_3 P_A} + \boldsymbol{\gamma_4 P_C};\ \boldsymbol{U_A} = \boldsymbol{\alpha P_A} \tag{4.1.21, 4.1.22}$$

To find the tip receptance "α" in terms of the known receptances of B and C, the equations from 4.1.17 to 4.1.22 can be rearranged to form a new receptance equation in terms of receptances of B and C by following the procedure as shown below.

From equations 4.1.17 to 4.1.20:

$$\boldsymbol{P_C} = \frac{\boldsymbol{U_C - \gamma_2 P_A}}{\boldsymbol{\gamma_1}} = \frac{\boldsymbol{U_B - \gamma_2 P_A}}{\boldsymbol{\gamma_1}} = \frac{\boldsymbol{\beta P_B - \gamma_2 P_A}}{\boldsymbol{\gamma_1}} = \frac{\boldsymbol{-\beta P_C - \gamma_2 P_A}}{\boldsymbol{\gamma_1}} \tag{4.1.23a}$$

$$\boldsymbol{P_C \gamma_1} = \boldsymbol{-\beta P_C} - \boldsymbol{\gamma_2 P_A} \tag{4.1.23b}$$

Hence,

$$\boldsymbol{P_C} = \frac{\boldsymbol{-\gamma_2 P_A}}{\boldsymbol{\gamma_1 + \beta}} \tag{4.1.23c}$$

Substituting equation 4.1.23c to 4.1.21:

$$\boldsymbol{U_A} = \boldsymbol{\gamma_3 P_A} + \boldsymbol{\gamma_4}\left(\frac{\boldsymbol{-\gamma_2 P_A}}{\boldsymbol{\gamma_1 + \beta}}\right) = \boldsymbol{\gamma_3 P_A} - \boldsymbol{\gamma_4}\left(\frac{\boldsymbol{\gamma_2 P_A}}{\boldsymbol{\gamma_1 + \beta}}\right) \tag{4.1.24}$$

From equation 4.1.22:

$$\boldsymbol{\alpha} = \frac{\boldsymbol{U_A}}{\boldsymbol{P_A}} = \frac{\boldsymbol{\gamma_3 P_A} - \boldsymbol{\gamma_4}(\frac{\boldsymbol{\gamma_2 P_A}}{\boldsymbol{\gamma_1}+\boldsymbol{\beta}})}{\boldsymbol{P_A}} = \boldsymbol{\gamma_3} - \boldsymbol{\gamma_4}(\frac{\boldsymbol{\gamma_2}}{\boldsymbol{\gamma_1}+\boldsymbol{\beta}}) \qquad (4.1.25)$$

Due to the reciprocity resulting from the assumed linearity, $\gamma_{Dd} = \gamma_{dD}$ , etc. Therefore,

$$\boldsymbol{\gamma_4} = \boldsymbol{\gamma_2}^{\boldsymbol{T}} \qquad (4.1.26)$$

Substituting equation 4.1.26 into equation 4.1.25:

$$\boldsymbol{\alpha} = \boldsymbol{\gamma_3} - \boldsymbol{\gamma_2}^{\boldsymbol{T}}\left(\frac{\boldsymbol{\gamma_2}}{\boldsymbol{\gamma_1}+\boldsymbol{\beta}}\right) = \boldsymbol{\gamma_3} - \boldsymbol{\gamma_2}^{\boldsymbol{T}}(\boldsymbol{\gamma_1} + \boldsymbol{\beta})^{-1}\boldsymbol{\gamma_2} \qquad (4.1.27)$$

## 4.2 The Timoshenko-receptance Model

The receptance approach was firstly developed using the Euler-Bernoulli equation (Bishop and Johnson 1960, 285), and it was then further developed using the Timoshenko equation with the effects of shear and rotary inertia. (Stone, 1992) It is safe to ignore the shear deformation in Euler-Bernoulli theory as long as the height and width of the beam is small compared with the length of the beam. However, if the beam becomes shorter or the node spacing is small, the effect of shear deformation becomes evident. (Inman 2008, 501) Suits (2001) showed that the Euler-Bernoulli model predicts that the normal mode frequencies are proportional to the reciprocal of bar length squared ($f \propto {}^{1}/_{l^2}$), and it is not accurate for shorter bars or higher frequencies (according to the experimental results). In this case bars have a length less than approximately ten times the depth of the bar. It was verified by experiment that the Euler-Bernoulli model predicts no length dependence for higher frequency modes, however, a rather significant length dependence is observed and is correctly predicted by the Timoshenko model[8]. Therefore, the combination of the Timoshenko model and the receptance sub-structuring method was developed to predict the required frequency ratios of marimba bars.

[8] Detailed descriptions of the Timoshenko model are presented in Appendix A.2.

First of all, the Timoshenko model differential equation that was achieved in Appendix A.2 is presented. What follows is based on Stone (1992).

$$EI\frac{\partial^4 y}{\partial x^4} - \rho I\left(1+\frac{E}{k^2G}\right)\frac{\partial^4 y}{\partial x^2\partial t^2} + \frac{\rho^2 I}{k^2 G}\frac{\partial^4 y}{\partial t^4} + \rho A\frac{\partial^2 y}{\partial t^2} = 0 \qquad (4.2.1)$$

The above equation is in the form of displacement *y* independent of the bending slope ψ, and the solution of it is in the form of

$$y = Y(x)e^{j\omega t} \qquad (4.2.2)$$

which is substituted in equation 4.2.1

$$EI\frac{\partial^4 Y(x)}{\partial x^4} + \rho I\left(1+\frac{E}{k^2G}\right)\omega^2\frac{\partial^2 Y(x)}{\partial x^2} + \rho\omega^2\left(\frac{\rho I\omega^2}{k^2G}\right)Y(x) = 0 \qquad (4.2.3)$$

The solution of *Y(x)* is in the form of

$$Y(x) = C_1 cos\xi x + C_2 sin\xi x + C_3 cosh\eta x + C_4 sinh\eta x \qquad (4.2.4a)$$

$$Y'(x) = -\xi C_1 sin\xi x + \xi C_2 cos\xi x + \eta C_3 sinh\eta x + \eta C_4 cosh\eta x \qquad (4.2.4b)$$

Similarly,

$$\psi(x) = D_1 cos\xi x + D_2 sin\xi x + D_3 cosh\eta x + D_4 sinh\eta x \qquad (4.2.5a)$$

$$\psi'(x) = -\xi D_1 sin\xi x + \xi D_2 cos\xi x + \eta D_3 sinh\eta x + \eta D_4 cosh\eta x \qquad (4.2.5b)$$

Combining equations A.1.3 and A.2.3, the shearing force becomes

$$\sum F = \rho A\frac{\partial^2\omega}{\partial t^2} = k^2 AG\left[\frac{d^2\omega}{dx^2} - \frac{d\psi}{dx}\right] \qquad (4.2.6)$$

The relationships between C$_1$, C$_2$, C$_3$, C$_4$, D$_1$, D$_2$, D$_3$ and D$_4$ are obtained by substituting $y = Y(x)e^{j\omega t}$ and $\psi = \psi(x)e^{j\omega t}$ into equation 4.2.7

$$\frac{d^2Y(x)}{dx^2} + \frac{\rho\omega^2}{k^2G}Y(x) - \frac{d\psi(x)}{dx} = 0 \qquad (4.2.7)$$

Substituting “$F = \frac{\rho\omega^2}{k^2G\xi} - \xi$” and “$r = \frac{\rho\omega^2}{k^2G\eta} + \eta$” into equations 4.2.4b and 4.2.5b gives

$$D_1 = -sC_2;\ D_2 = sC_1 \qquad \text{(4.2.8a, 4.2.8b)}$$

$$D_3 = rC_4;\ D_4 = rC_3 \qquad \text{(4.2.9a, 4.2.9b)}$$

By considering the free-free end conditions for solving Cs and Ds, “$\psi'(x) = 0$” and “$Y'(x) - \psi(x) = 0$” are given at the free end, which means that no bending moment and shear force appear at the end.

When an excitation force “$Fe^{j\omega t}$” is applied at the other end, “$\psi'(x) = 0$” and “$Y'(x) - \psi(x) = \frac{F}{k^2GA}$” are obtained. If an excitation moment “$He^{j\omega t}$” is applied at the other end, “$\psi'(x) = \frac{H}{EI}$” and “$Y'(x) - \psi(x) = 0$” are obtained.

Hence, substituting from equations 4.2.8b and 4.2.9b gives:

$$s\xi C_1 + r\eta C_2 = 0 \qquad \text{(4.2.9a)}$$

$$Y' - \psi = \xi C_2 + \eta C_4 - D_1 - D_3 = \frac{-F}{k^2GA} \qquad \text{(4.2.9b)}$$

and substituting from equations 4.2.8a and 4.2.9a gives:

$$(\xi + s)C_2 + (\eta - r)C_4 = \frac{-F}{k^2GA} \qquad \text{(4.2.10)}$$

At x=L,

$$\psi' = s\xi C_2 sin\xi L + s\xi C_1 cos\xi L + r\eta C_4 sinh\eta L + r\eta C_3 cosh\eta L = 0 \qquad \text{(4.2.11a)}$$

$$Y' - \psi = -(\xi + s)C_1 sin\xi L + (\xi + s)C_2 cos\xi L + (\eta + r)C_3 sinh\eta L + (\eta - r)C_4 cosh\eta L = 0 \quad \text{(4.2.11b)}$$

Therefore, combining equations from 4.2.9a to 4.2.11b with the end conditions, C$_1$, C$_2$, C$_3$ and C$_4$ can be obtained:

$$C_1 = \frac{-\xi\{(r\eta^2/s\xi^2)sinh\eta Lcos\xi L+cosh\eta Lsin\xi L\}F}{\rho\omega^2 A\{2-2cos\xi Lcosh\eta L+(r\eta^2/s\xi^2-s\xi^2/r\eta^2)sin\xi Lsinh\eta L\}} \quad (4.2.12a)$$

$$C_2 = \frac{-\xi\left\{\left(\frac{r\eta^2}{s\xi^2}\right)sinh\eta Lcos\xi L(1-cosh\eta Lcos\xi L)\right\}F}{\rho\omega^2 A\{2-2cos\xi Lcosh\eta L+(r\eta^2/s\xi^2-s\xi^2/r\eta^2)sin\xi Lsinh\eta L\}} \quad (4.2.12b)$$

$$C_3 = \frac{\eta\{sinh\eta Lcos\xi L+(s\xi^2/r\eta^2)cosh\eta Lsin\xi L\}F}{\rho\omega^2 A\{2-2cos\xi Lcosh\eta L+(r\eta^2/s\xi^2-s\xi^2/r\eta^2)sin\xi Lsinh\eta L\}} \quad (4.2.12c)$$

$$C_4 = \frac{-\eta\left\{\left(\frac{s\xi^2}{r\eta^2}\right)sinh\eta Lsin\xi L+cosh\eta Lcos\xi L-1\right\}F}{\rho\omega^2 A\{2-2cos\xi Lcosh\eta L+(r\eta^2/s\xi^2-s\xi^2/r\eta^2)sin\xi Lsinh\eta L\}} \quad (4.2.12d)$$

To make the equations simpler, the following substitutions were used

$$F_1 = sin\xi Lsinh\eta L;\ F_2 = cos\xi Lcosh\eta L \quad (4.2.13a, 4.2.13b)$$

$$F_3 = cos\xi Lcosh\eta L - 1;\ F_4 = sin\xi Lcosh\eta L \quad (4.2.13c, 4.2.13d)$$

$$F_5 = cos\xi Lsinh\eta L;\ F_6 = cos\xi L - cosh\eta L \quad (4.2.13e, 4.2.13f)$$

$$\Delta = \rho\omega^2 A\{2 - 2F + (r\eta^2/s\xi^2 - s\xi^2/r\eta^2)F_1\} \quad (4.2.13g)$$

so that substituting in equation 4.2.4 for C$_1$, C$_2$, C$_3$ and C$_4$, the receptance $\alpha_{XO} = Y(x)/F$ could be achieved as shown in equation 4.2.14. The subscript X is the chosen location, whereas O is the origin of the beam and L is the end of the beam. As receptances can be defined as $\alpha_{ij}$ of which the first subscript "i" represents the response location while the second subscript "j" represents the location of the excitation. If the first subscript "i" is in capital, it means the deflection response is required. Otherwise, the slope of the response is required. If the second subscript "j" is in capital, it means force is applied. Otherwise, the moment is applied.

$$\alpha_{XO} = \frac{-[\xi\left\{\left(\frac{r\eta^2}{s\xi^2}\right)F_5+F_4\right\}cos\xi x+\xi\left\{\left(\frac{r\eta^2}{s\xi^2}\right)F_1-F_3\right\}sin\xi x-\eta\left\{\left(\frac{s\xi^2}{r\eta^2}\right)F_4+F_5\right\}cosh\eta x+\eta\left\{\left(\frac{s\xi^2}{r\eta^2}\right)F_1+F_3\right\}sinh\eta x]}{\Delta} \quad (4.2.14)$$

In the similar manner the receptances α$_{XO}$, α$_{XL}$ and α$_{Xl}$ could be achieved, and they are provided by Stone (1992) as shown in equations from 4.2.15 to 4.2.17. [9]

[9] The receptances are valid for the condition $\omega^2 < \frac{k^2GA}{\rho I}$

$$\alpha_{Xo} = \frac{-[\xi\left\{\left(\frac{r\eta}{s\xi}\right)F_3+\left(\frac{s\xi}{\eta}\right)F_1\right\}cos\xi x-\xi\left\{\left(\frac{s\xi}{\eta}\right)F_5-\left(\frac{r\eta}{\xi}\right)F_4\right\}sin\xi x-\eta\left\{\left(\frac{r\eta}{\xi}\right)F_1-\left(\frac{s\xi}{\eta}\right)F_3\right\}cosh\eta x-\eta\left\{\left(\frac{s\xi}{\eta}\right)F_5+\left(\frac{r\eta}{\xi}\right)F_4\right\}sinh\eta x]}{\Delta} \quad (4.2.15)$$

$$\alpha_{XL} = \frac{-[\xi\left\{\left(\frac{r\eta^2}{s\xi^2}\right)sinh\eta L+sin\xi L\right\}cos\xi x-\xi F_6 sin\xi x-\eta\left\{\left(\frac{s\xi^2}{r\eta^2}\right)sin\xi L+sinh\eta L\right\}cosh\eta x-\eta F_6 sinh\eta x]}{\Delta} \quad (4.2.16)$$

$$\alpha_{Xl} = \frac{-[r\eta F_6 cos\xi x-\left\{\left(\frac{s\xi^2}{\eta}\right)sinh\eta L-r\eta sin\xi L\right\}sin\xi x-s\xi F_6 cosh\eta x-\left\{s\xi sinh\eta L-\left(\frac{r\eta^2}{\xi}\right)sin\xi L\right\}sinh\eta x]}{\Delta} \quad (4.2.17)$$

When sub-systems are added together the tip receptances are used, that is the receptances at the ends of each beam. These can be obtained by putting *x=0* and *L* respectively as shown from equations 4.2.18 to 4.2.23.

$$\alpha_{OO} = \alpha_{LL} = \frac{-F_5\left(\frac{r\eta^2}{s\xi}-\eta\right)+F_4\left(\frac{s\xi^2}{r\eta}-\xi\right)}{\Delta} \quad (4.2.18)$$

$$\alpha_{oO} = \alpha_{Oo} = -\alpha_{lL} = -\alpha_{Ll} = \frac{-F_3(s\xi+r\eta)+F_1\left(\frac{r\eta^2}{\xi}-\frac{s\xi^2}{\eta}\right)}{\Delta} \quad (4.2.19)$$

$$\alpha_{LO} = \alpha_{OL} = \frac{-sinh\eta L\left(\frac{r\eta^2}{s\xi}-\eta\right)+sin\xi L\left(\frac{s\xi^2}{r\eta}-\xi\right)}{\Delta} \quad (4.2.20)$$

$$\alpha_{lO} = \alpha_{Ol} = -\alpha_{oL} = -\alpha_{Lo} = \frac{-F_6(r\eta-s\xi)}{\Delta} \quad (4.2.21)$$

$$\alpha_{oo} = \alpha_{ll} = \frac{F_4 r\eta\left(s-\frac{r\eta}{\xi}\right)+F_5 s\xi\left(r-\frac{s\xi}{\eta}\right)}{\Delta} \quad (4.2.22)$$

$$\alpha_{lo} = \alpha_{ol} = \frac{sinh\eta L\left(rs\xi-\frac{s^2\xi^2}{\eta}\right)+sin\xi L\left(rs\eta-\frac{r^2\eta^2}{\xi}\right)}{\Delta} \quad (4.2.23)$$

This page intentionally left blank.

# Chapter 5: The Underside Shape Predictive Model

Non-prismatic beams are used in marimbas as shown in Figure 1.2. They have their lowest three natural bending frequencies tuned to the fundamental and relative (first and second harmonic) pitches by removing the underside material of the beam at different locations. The transverse vibration modes need to be accurately modelled so that the correct dimensions of them can be accurately calculated. The above task has been achieved by an underside shape predictive model that was developed by Entwistle and McGrehan (2007). The Timoshenko beam receptances were applied to the non-prismatic beams, and a search algorithm using Newton-Raphson method was implemented to determine the correct dimensions of the underside shape of the non-prismatic beam as shown in Figure 5.1. It has been verified by experiment that this underside shape predictive model was very accurate for aluminium beams, however it cannot avoid errors when tuning wooden beams. Thus, the existing predictive model was modified in this thesis to develop an automatic tuning for wooden beams.

In this chapter, the modelling procedure of this underside shape predictive model, including the selection of the mathematical model, the method of measuring the mechanical properties of the test material, etc, are described in detail.

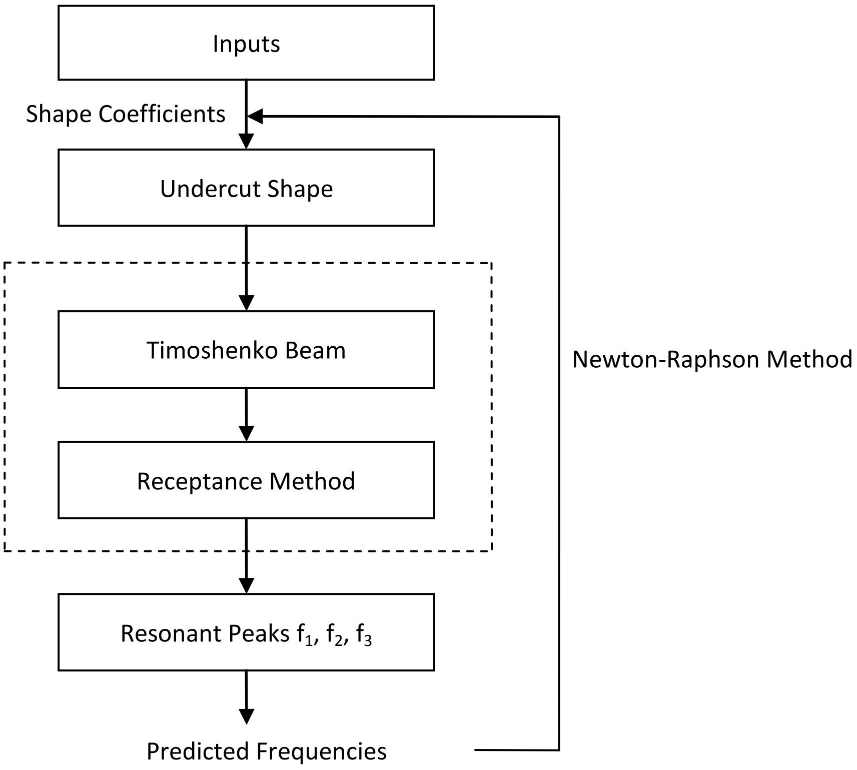


Figure 5.1 A flowchart of the underside shape predictive model solution procedure

## 5.1 Selection of the Mathematical Model

The calculations of the natural frequencies of non-prismatic structures can be achieved from a mathematical model, such as the Euler-Bernoulli model or the Timoshenko model that takes into account the effects of rotary inertia and shear stress. In this section, the selection of the appropriate mathematical model for this thesis is verified.

Bustamante (1991) has provided his experimental results on the frequency ratios of the first three transverse modes predicted and measured in uniform rectangular beams as shown in Table 2. It can be seen that the calculation performed by the Timoshenko model (combined with FEA method) produced the predicted results closer to the experimental results comparing with the ones obtained by the Euler-Bernoulli model, as was described in Section 2.1.1.

| | ${}^{f_2}\!/_{f_1}$ | ${}^{f_3}\!/_{f_1}$ |
|---|---|---|
| Euler-FEA Model | 2.76 | 5.4 |
| Timoshenko-FEA Model | 2.71 | 5.15 |
| Experiment 1 | 2.71 | 5.15 |
| Experiment 2 | 2.7 | 5.16 |

Table 2: Frequency ratios of the first three transverse modes calculated and measured in a uniform rectangular beam (Bustamante, 1991)

To confirm the conclusions of Bustamante (1991), a similar experiment, as shown in Table 3, was carried out for this thesis, in which the predicted results obtained by mathematical models, including the Euler-receptance model, the Timoshenko-receptance model and the finite element model, were compared with the experimental results. The Timoshenko model was firstly applied to calculate the geometry of a marimba bar (using Tasmanian oak) to produce the targeted lowest three natural frequencies (300Hz, 1200Hz, 2940Hz). The Euler-Bernoulli model and the finite element model were then used to predict the natural frequencies based on the same geometry calculated by the Timoshenko model. (Detailed calculations of Euler-Bernoulli theory and finite element method are presented in Appendix A) A wood blank was manufactured based on the geometry that was determined by the Timoshenko model.

| | $^{f_2}/_{f_1}$ | $^{f_3}/_{f_1}$ |
|---|---|---|
| Timoshenko-receptance Model | 4 | 9.8 |
| Euler-receptance Model | 4.035 | 10.014 |
| FEA-Euler Model | 4.035 | 10.013 |
| Experiment | 3.71 | 8.56 |

Table 3: The first three transverse modes calculated and measured in marimba bars

The results show that the experimental results are closer to the ones that the Timoshenko-receptance model predicts, and therefore it is concluded that the Timoshenko model is more suitable for this thesis compared to the Euler-receptance model and the finite element model for the undercut prediction of wooden beams.

### 5.2 The Preparation of Model Inputs

The accuracy of the underside shape predictive model is mostly dependent on the accuracy of the model inputs, such as the dimensions, density, elastic modulus, Poisson's ratio and the Timoshenko shear factor[10] of the wood blank. However, the elastic properties of wood vary along the length of the tested wood and cannot be accurately measured due to its nature of non-homogeneity and anisotropy. Bretos, Santamaria and Moral (1998) followed Ostergaard (1997) to investigate the influence of each elastic constant on the natural frequencies, and it was shown that the density of wood and the Young's modulus ($E_1$) and the shear modulus ($G_{12}$) along the grain are important for natural frequencies. On the other hand, other elastic properties would not affect natural frequencies much. However, the most appropriate method of solving the Young's modulus ($E_1$) and the shear modulus ($G_{12}$) was not provided.

To solve this problem, Entwistle and McGrechan (2007) obtained the elastic properties by measuring the vibration frequencies of a prismatic beam and adjusted

[10] See Appendix B for descriptions

the Young's modulus ($E_1$) and shear modulus ($G_{12}$) in the Timoshenko-receptance model until they agreed. The elastic values obtained by using this method were the best average values that produce the best predictive results. However, it still cannot avoid errors that are caused by the uncertainties of the tested wood.

For this thesis, the elastic properties of wood are not expected to be accurately measured. Instead, some assumptions were made to simplify the underside shape predictive model, e.g. wood is homogeneous and isotropic, etc. It can be seen from equations B.2 and B.3 in Appendix B that the elastic properties of a wood blank can be calculated if the bending and torsional natural frequencies were available. (These were measured on a free-free beam.) When such beam is struck at its centre[11], the natural bending frequencies could be observed, whereas when the beam is struck at the corner, both the bending and torsional frequencies could be observed. The torsional frequencies could then be isolated by comparing the above two results. The Poisson's ratio and the shear factor could be determined using equations B.4 and B.5. This method could produce the predicted results that are close to the targets but with some error, and leave the fine tuning or the correction process in latter stages (described in Chapter 6).

### 5.3 Possible Cutout Geometries

For manufacturing purposes, the undercut of a marimba bar should be formed by a smooth curve. Since there are three target frequency outputs ($f_1$, $f_2$ and $f_3$), the mathematical function describing the curve requires three adjustable coefficients. Many functions can be devised that comply with the requirements, and some of the possibilities are provided by Entwistle and McGrechan (2007) as follows:

Cubic:
$$y(x) = a_1x^3 + a_2x^2 + a_3, x \geq 0 \tag{5.1}$$

Exponential 1:
$$y(x) = a_3[1 + \sum_{i=1}^{\infty} \frac{{a_1}^i - {a_2}^i}{i!}(x)^i] \tag{5.2}$$

Exponential 2:
$$y(x) = a_3[2 + \sum_{i=1}^{\infty} \frac{{a_1}^i + (-1)^i {a_2}^i}{i!}(x)^i] \tag{5.3}$$

[11] Or off-centre when a central node exists

While not reported here in detail, many curves provide solutions, and those used here are not unique solutions. For this thesis, the cubic equation without the linear term (equation 5.1) was chosen so that the central slope at x=0 is zero.

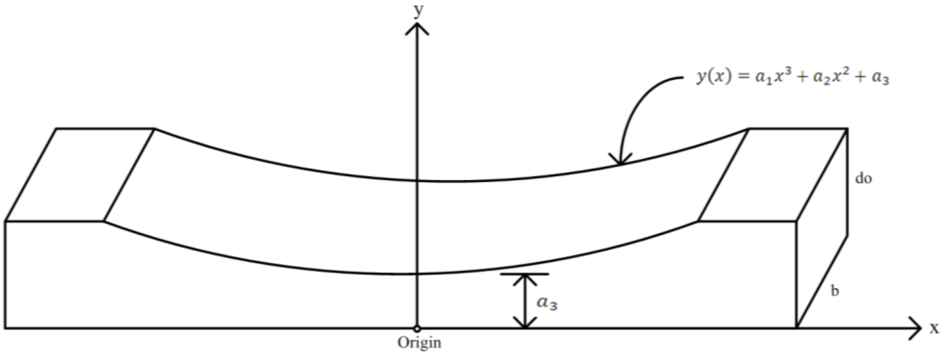


Figure 5.2 A marimba beam with undercut (Inverted)

An initial guess for the three coefficients in the chosen function $y(x) = a_1x^3 + a_2x^2 + a_3$ enables the receptance modelling to proceed and extract estimates for the three lowest bending modes. The smooth curve is approximated, to any required accuracy, with a "staircase" consisting of prismatic sections. Increasing the number of sections improves the approximation to the cubic equations. (It was found that the results converged when the staircase approximation was given 300 to 600 steps.)

The tip receptances of such geometry can be obtained using equations from 4.2.16 to 4.2.21. The natural frequencies would occur when any element of α becomes infinite[12], and due to the zero damping assumption, a sign change occurs when a resonant frequency is traversed. Alternatively, the natural frequencies can be calculated as the zeros of $1/\alpha$ of the non-prismatic beams. (Entwistle and McGrehan, 2007)

In order to apply the Newton-Raphson method to find the correct values of $a_1$, $a_2$ and $a_3$, an estimate of $\frac{\partial f_i}{\partial a_j}$ is needed as described in Appendix C. This is achieved by slight changes in $a_j$ ($da_j$), recalculations of $f_i$ and hence an estimate of $\frac{\partial f_i}{\partial a_j}$. The three

[12] Recall that for a beam α is a 2x2 matrix of tip receptances. See equation 4.1.3.

variable Newton-Raphson method is then applied to find the correct values of $a_1$, $a_2$ and $a_3$ that achieve the objective $f_1$, $f_2$ and $f_3$.

The following figure shows the calculated tip receptance magnitude that was obtained from the predicted 442Hz marimba bar.

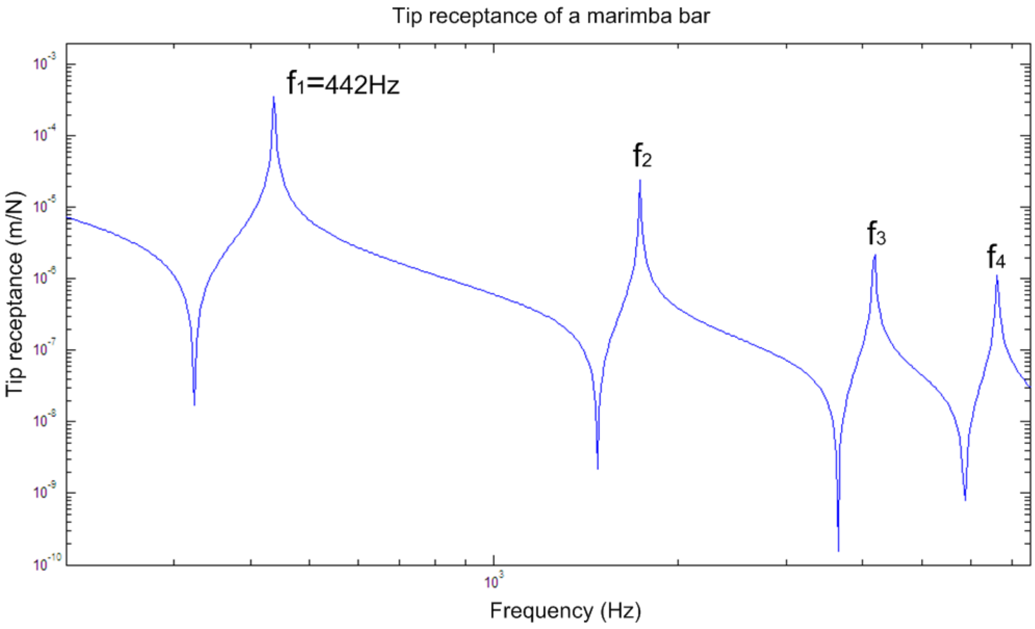


Figure 5.3 Tip receptance of a marimba beam

## Chapter 6: Fine Tuning Program

The underside shape predictive model (described in Chapter 5) was developed based upon many assumptions, such as whether the nature of the test material (wood) is homogeneous and isotropic, etc. However, these assumptions are not true. Hence, to improve the accuracy of the underside shape prediction, a fine tuning program was developed to correct the error that is caused by the unavoidable uncertainties of the test material. The idea of the solution was to take into account the experimental results (experimental natural frequencies), namely, if the error between the predicted natural frequencies and the measured natural frequencies can be predicted, it can be cancelled out in advance in the prediction phase.

As the underside shape predictive model has primarily determined the undercut geometry that produces the target natural frequencies of a marimba bar, the wood blank will be incrementally cut from a rectangular shape with relatively high natural frequencies to an undercut shape with the target frequencies. The resulting natural frequencies decrease along with the reduction of the central depth during manufacture. If the predicted natural frequencies for a particular cutting layer are defined as the "incremental target frequencies", the difference between the incremental target frequencies and the experimental natural frequencies of that cutting layer always exist due to the complicated nature of wood. If these differences can be analysed, the relationships between them may possibly be described by linear equations.

The wood blank will be incrementally cut to the depth above the original predicted underside shape, leaving extra material at the central location "uncut"[13]. The final underside shape (as the final cut of a marimba bar) can then be re-predicted by extrapolation based on the achieved linear equations. It is expected that the re-predicted underside shape will be different from the original shape prediction.

The fine tuning program is a major part of this thesis; it manages the fine tuning process, such as how the wood blank is manufactured during fine tuning, the analysis of the relationships between the incremental target natural frequencies and the

[13] See explanations in Section 6.2.

experimental natural frequencies, and the underside shape re-prediction. It also communicates with the underside shape predictive model and the hardware controlling program[14] during the fine tuning process as the predictive model provides the incremental natural frequencies for each cutting layer and the hardware controlling program drives the CNC machine to cut the wood blank to produce the experimental natural frequencies. As the programming code is provided in Appendix G, the detail of the communication between these models and programs is not included in this chapter. Rather, the fine tuning process, the analysis of the relationship between the incremental target natural frequencies and the experimental natural frequencies, and the underside shape re-prediction are explained in this chapter.

The fine tuning program is described in three sections: Section 6.1 presents the fine tuning procedure and the definitions of the terms that are used in the fine tuning program. Section 6.2 describes the "shape" matrix, which finds the relationship between the resulting natural frequencies and the shape coefficients ($a_1$, $a_2$ and $a_3$). Such "shape" matrix makes the underside shape re-prediction easier when the changes of the frequencies are known. Section 6.3 explains the determination of the "relationship linear equations". (Examples are used in Sections 6.2 and 6.3.)

### 6.1 The Fine Tuning Procedure

The fine tuning process starts with the underside shape prediction. It is expected that the experimental natural frequencies will be different from the user-desired (target) natural frequencies if the wood blank is cut directly to the determined undercut geometry as the assumptions for the underside shape predictive model are not true. To correct this error, the fine tuning program (as shown in Figure 6.1) will manage the iteration steps so that the wood blank will be incrementally cut by a CNC machine from above to reduce the resulting natural frequencies.

[14] See Chapter seven for the descriptions of the hardware controlling program

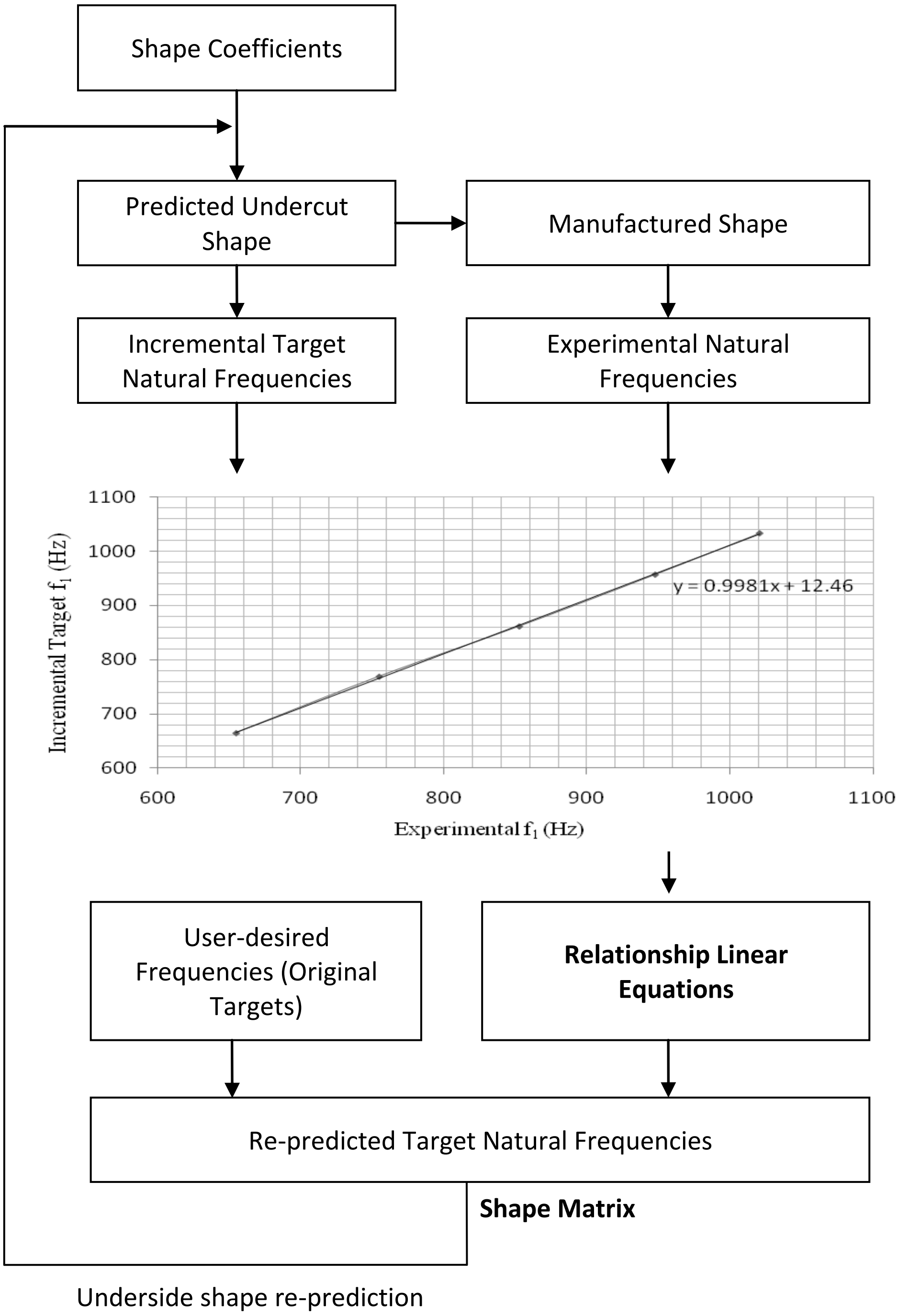

Shape Coefficients
Predicted Undercut Shape
Manufactured Shape
Incremental Target Natural Frequencies
Experimental Natural Frequencies
1100
1000
900
800
700
600
Incremental Target $f_1$ (Hz)
600
700
800
900
1000
1100
Experimental $f_1$ (Hz)
y = 0.9981x + 12.46
User-desired Frequencies (Original Targets)
Relationship Linear Equations
Re-predicted Target Natural Frequencies
Shape Matrix
Underside shape re-prediction

Figure 6.1 A flowchart of the fine tuning process

During manufacture, the fine tuning program records both the incremental target natural frequencies and the experimental natural frequencies for each cut, and by the completion of not less than five cuts the “relationship linear equations” (equations from 6.1a to 6.1c) can be generated.

$$\mathrm{Predicted\_f1} = \mathrm{A(experimental\_f1)} + \mathrm{B} \tag{6.1a}$$

$$\mathrm{Predicted\_f2} = \mathrm{C(experimental\_f2)} + \mathrm{D} \tag{6.1b}$$

$$\mathrm{Predicted\_f3} = \mathrm{E(experimental\_f3)} + \mathrm{F} \tag{6.1c}$$

Once the “relationship linear equations” are available and ready to be used for the final cut, the user-desired (target) natural frequencies of a marimba bar is considered as the experimental results for the final cut, and the re-predicted target frequencies of the final cut can be determined according to the obtained linear equations by extrapolation. Once the difference between the original target frequencies and the re-predicted target frequencies natural frequencies are calculated, the change of the underside shape can be determined, described by the shape coefficients. This brings another problem to this thesis. How to find the changes of the shape coefficients when given the changes of the resulting natural frequencies?

For this thesis, another term was introduced to solve the above problem; a “shape matrix” gives the theoretical relationship between the change of the underside shape and the change of the resulting natural frequencies. It can be described as shown in equation 6.2, and any change of the shape coefficients results changes in the natural frequencies.

$$\text{shape matrix} = \begin{bmatrix} \frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} & \frac{\partial f_1}{\partial a_3} \\ \frac{\partial f_2}{\partial a_1} & \frac{\partial f_2}{\partial a_2} & \frac{\partial f_2}{\partial a_3} \\ \frac{\partial f_3}{\partial a_1} & \frac{\partial f_3}{\partial a_2} & \frac{\partial f_3}{\partial a_3} \end{bmatrix} \tag{6.2}$$

A successful tuning process will cancel out the error that is caused by the uncertainties of the test material, and re-produce the correct final undercut geometry for the target natural frequencies.

## 6.2 The Determination of the “Relationship Linear Equations”

As introduced in the previous section, the “relationship linear equations” are defined as the equations that describe the relationships between the incremental target frequencies and the experimental natural frequencies, while the incremental target frequencies are the predicted natural frequencies for different cutting layers during manufacture. The fine tuning process requires the wood blank to be incrementally cut to the predicted underside shape from above, and it normally needed approximately not less than five cuts as shown in Figure 6.2 so that the required incremental target frequencies and the experimental natural frequencies of each cutting layer can be supplied to the fine tuning program to solve the “relationship linear equations”. These equations will match the experimental natural frequencies with the incremental target frequencies.

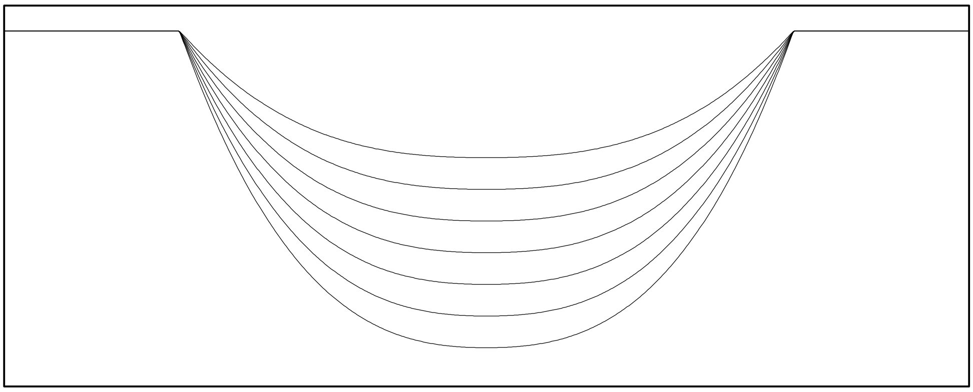


Figure 6.2 Cutting layers

For this section, a wood blank (Tasmanian oak) was used as the test blank as an example, the user-desired (target) natural frequencies were set to be 380Hz, 1520Hz and 3724Hz. The incremental target frequencies and the experimental natural frequencies of each cut were recorded in Table 4, and plotted in figures from 6.3 to 6.5. As shown in Table 4 the resulting natural frequencies of the last cut prior to the final cut (Fifth cut) were higher than the target natural frequencies, in order to leave extra depth at the central location “uncut”. The “relationship linear equations” can be solved based on the obtained information. Therefore, if the re-predicted underside shape varies largely from the original prediction, the material that has already been cut will not affect the shape.

| | First Cut | Second Cut | Third Cut | Fourth Cut | Fifth Cut |
|---|---|---|---|---|---|
| Incremental target $f_1$ | 1033.1 | 957 | 861.6 | 769.2 | 665.2 |
| Experimental $f_1$ | 1021 | 948 | 853 | 755 | 655 |
| Incremental target $f_2$ | 2920.9 | 2821.5 | 2682.5 | 2467.2 | 2318.7 |
| Experimental $f_2$ | 2621 | 2543 | 2413 | 2197 | 2046 |
| Incremental target $f_3$ | 4508 | 5522.7 | 5344.1 | 5164 | 4931.7 |
| Experimental $f_3$ | 5646.4 | 4448 | 4335 | 4210 | 4052 |

Table 4: The incremental target frequencies and the experimental natural frequencies for each cut

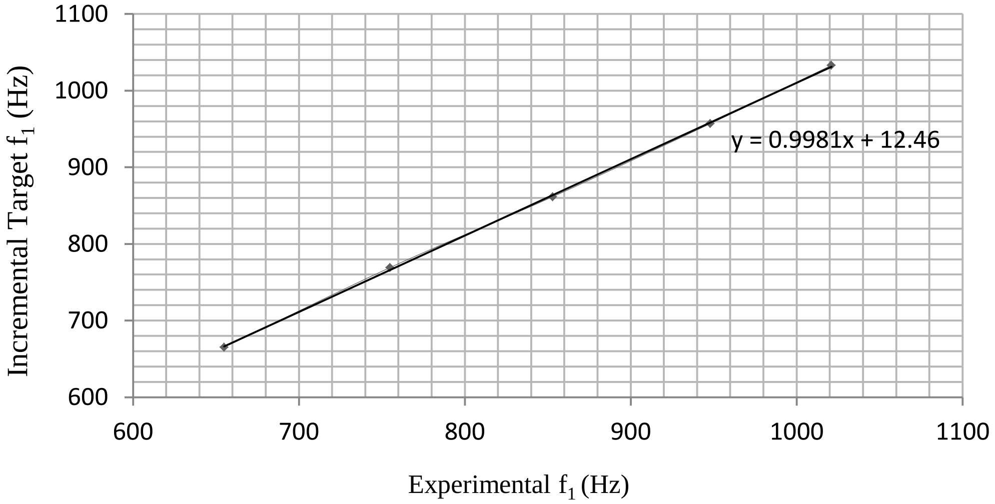


Figure 6.3 Incremental target $f_1$ vs. experimental $f_1$

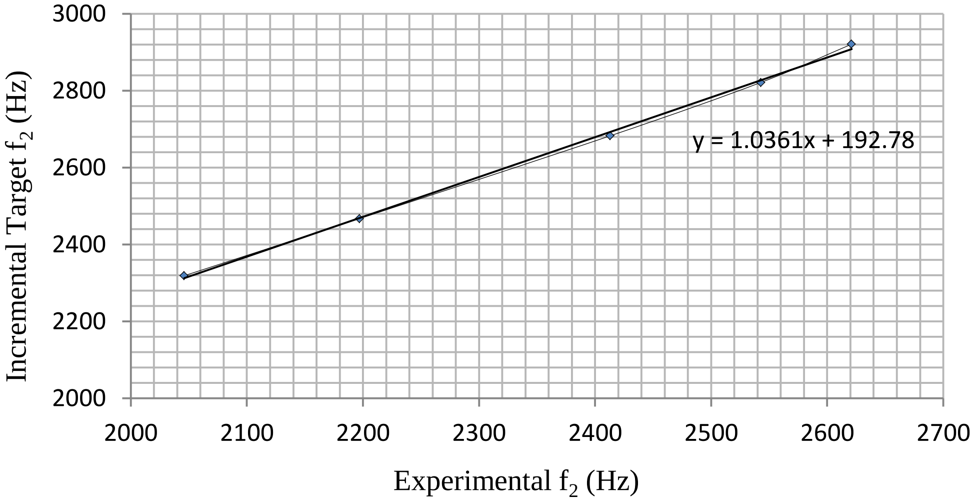


Figure 6.4 Incremental target $f_2$ vs. experimental $f_2$

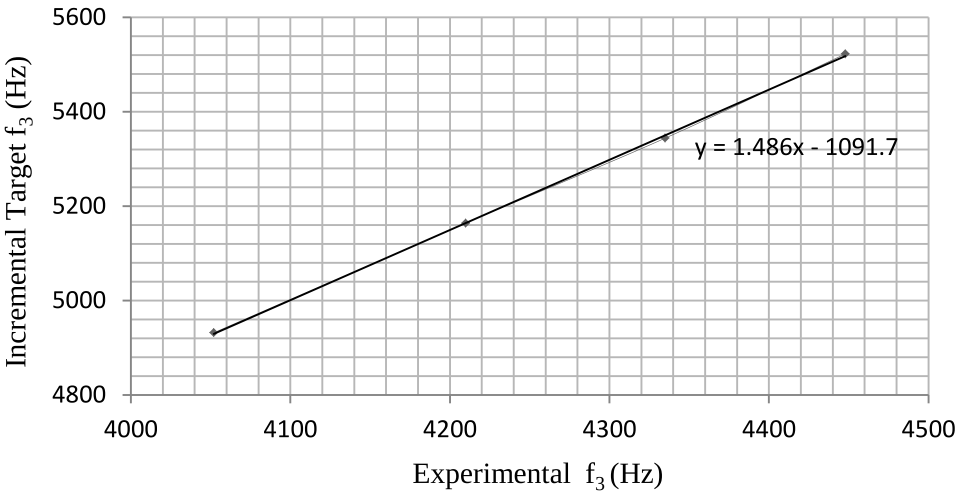


Figure 6.5 Incremental target $f_3$ vs. experimental $f_3$

By using the above figures, equations from 6.1a to 6.1c can be solved as

$$\mathrm{f1}_{\text{predicted}} = 0.9981\mathrm{f1}_{\text{experimental}} + 12.46 \tag{6.3a}$$

$$\mathrm{f2}_{\text{predicted}} = 1.0361\mathrm{f2}_{\text{experimental}} + 192.78 \tag{6.3b}$$

$$\mathrm{f3}_{\text{predicted}} = 1.486\mathrm{f3}_{\text{experimental}} - 1091.7 \tag{6.3c}$$

The user-desired (target) natural frequencies were set to be 380Hz, 1520Hz and 3724Hz, they can be considered as the experimental natural frequencies in equations from 6.3a to 6.3c. Therefore, the relative “predicted” natural frequencies in the equations can be determined by extrapolation, and the solutions became the re-predicted target natural frequencies for the final cut, which are 391.738Hz, 1808.182Hz and 4436.764Hz as shown in Table 5.

| Target f1(Hz) | Target f2(Hz) | Target f3(Hz) |
|---|---|---|
| 380 | 1520 | 3724 |
| Re-predicted Target f1(Hz) | Re-predicted Target f2(Hz) | Re-predicted Target f3(Hz) |
| 391.738 | 1808.182 | 4436.764 |

Table 5: The target frequencies and the re-predicted target natural frequencies

### 6.3 The Determination of the "Shape Matrix"

The "shape matrix" describes the relationships between the changes of the lowest three natural bending frequencies of a marimba bar and the changes of shape coefficients ($a_1$, $a_2$, $a_3$). As shown in Section 5.3 that the underside shape of a marimba beam is described by equation 5.1, the shape coefficients of this cubic equation alter the shape so that the natural frequencies of the beam can be tuned. If the target natural frequencies are re-predicted as described in the previous section, the approximate underside shape should be re-predicted as well, in this case the change of the resulting natural frequencies requires the change of three shape coefficients.

In this section, the method of determining such "shape matrix" is presented in detail. As shown in Figures from 6.6 to 6.8, when the shape coefficients ($a_1$, $a_2$, $a_3$) vary in the predictive model (as the inputs), the resulting lowest three natural frequencies vary in an approximately linear fashion. (The same wood blank was chosen as the example for this section.)

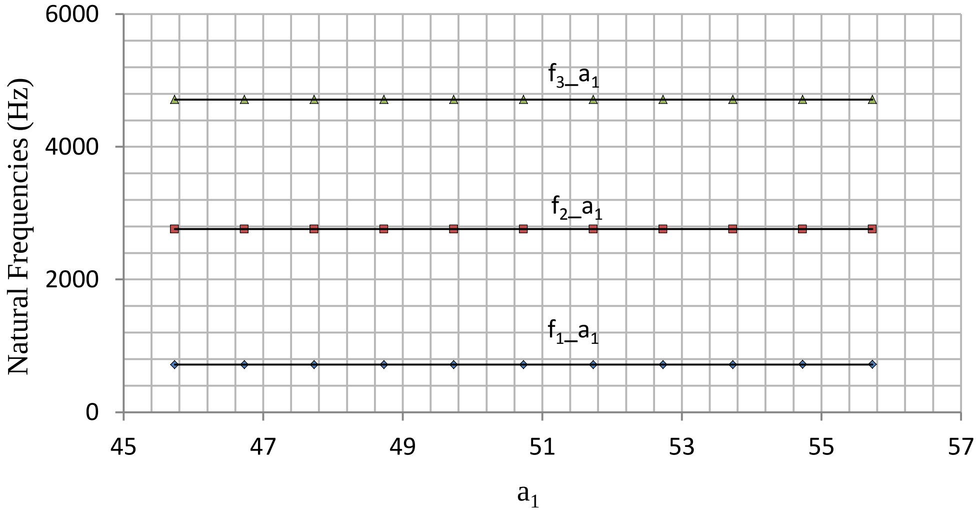


Figure 6.6 Natural frequency changes vs. the change of $a_1$

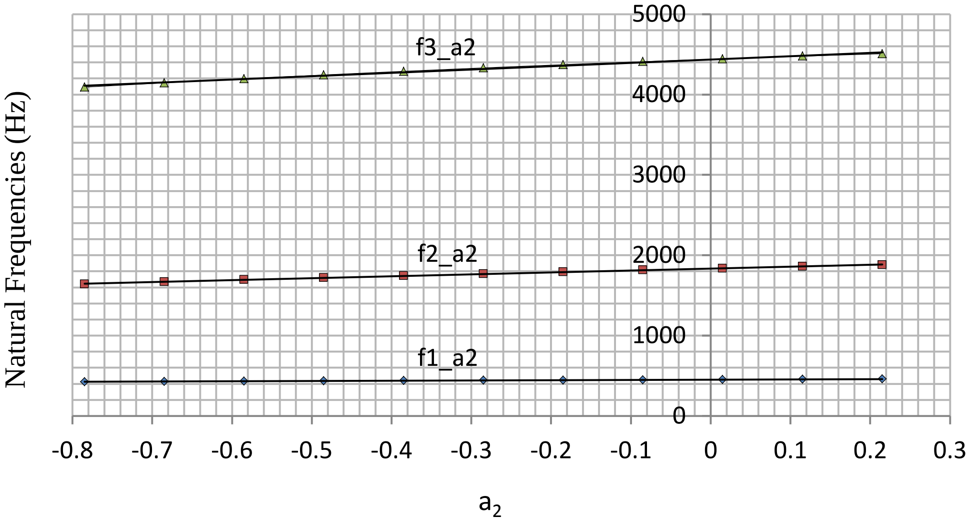


Figure 6.7 Natural frequency changes vs. the change of $a_2$

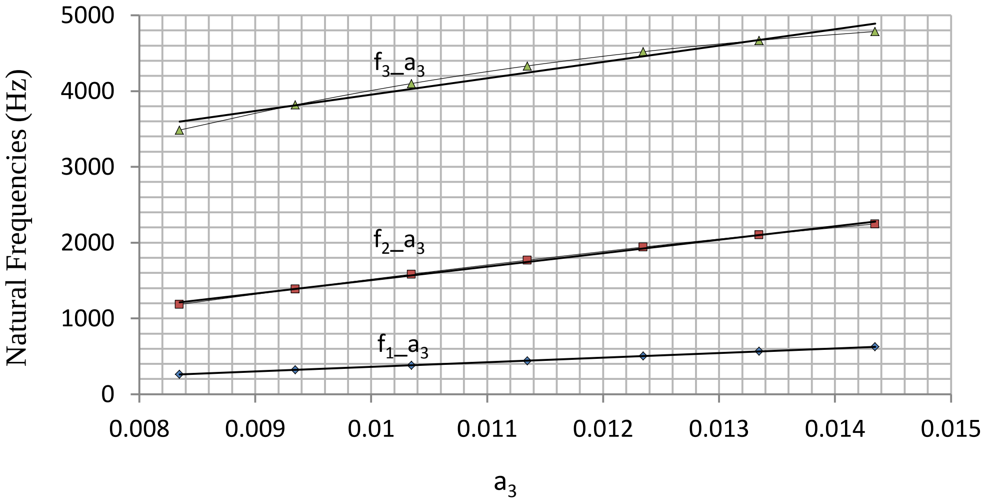


Figure 6.8 Natural frequency changes vs. the change of $a_3$

The relationship between the changes in the cubic equation's coefficients and the changes of the natural frequencies can be described by equation 6.2. Any change of the shape coefficients will change all three natural frequencies. Thus, the lowest three natural frequencies can be tuned by adjusting these shape coefficients as shown in equation 6.4.

$$\begin{bmatrix}\Delta f_1\\ \Delta f_2\\ \Delta f_3\end{bmatrix}=\begin{bmatrix}f1_{new}\\ f2_{new}\\ f3_{new}\end{bmatrix}-\begin{bmatrix}f1_{old}\\ f2_{old}\\ f3_{old}\end{bmatrix}=\begin{bmatrix}\frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} & \frac{\partial f_1}{\partial a_3}\\ \frac{\partial f_2}{\partial a_1} & \frac{\partial f_2}{\partial a_2} & \frac{\partial f_2}{\partial a_3}\\ \frac{\partial f_3}{\partial a_1} & \frac{\partial f_3}{\partial a_2} & \frac{\partial f_3}{\partial a_3}\end{bmatrix}\begin{bmatrix}\Delta a_1\\ \Delta a_2\\ \Delta a_3\end{bmatrix} \tag{6.4}$$

The shape matrix in the example can be obtained by fitting the slope of each line from Figures 6.6 to 6.8 as:

$$\begin{bmatrix}\frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} & \frac{\partial f_1}{\partial a_3}\\ \frac{\partial f_2}{\partial a_1} & \frac{\partial f_2}{\partial a_2} & \frac{\partial f_2}{\partial a_3}\\ \frac{\partial f_3}{\partial a_1} & \frac{\partial f_3}{\partial a_2} & \frac{\partial f_3}{\partial a_3}\end{bmatrix}=\begin{bmatrix}1.8536 & 43.2814 & 64228\\ 18.8939 & 357.1171 & 202530\\ 73.0139 & 1079.6 & 372350\end{bmatrix} \tag{6.5}$$

The desired frequencies were set to be 380Hz, 1520Hz and 3724Hz, which are the same as the target natural frequencies in Section 6.2. The re-predicted natural frequencies are 391.738Hz, 1808.182Hz and 4436.764Hz. They can be written in the matrix format as shown in equations 6.6a and 6.6b.

$$\begin{bmatrix}f1_{old}\\ f2_{old}\\ f3_{old}\end{bmatrix}=\begin{bmatrix}380\\ 1520\\ 3724\end{bmatrix};\ \begin{bmatrix}f1_{new}\\ f2_{new}\\ f3_{new}\end{bmatrix}=\begin{bmatrix}391.738\\ 1808.182\\ 4436.764\end{bmatrix} \tag{6.6a, 6.6b}$$

The differences between the original predicted natural frequencies in equation 6.6a and the re-predicted natural frequencies in equation 6.6b can be calculated as:

$$\begin{bmatrix}\Delta f_1\\ \Delta f_2\\ \Delta f_3\end{bmatrix}=\begin{bmatrix}f1_{new}\\ f2_{new}\\ f3_{new}\end{bmatrix}-\begin{bmatrix}f1_{old}\\ f2_{old}\\ f3_{old}\end{bmatrix}=\begin{bmatrix}+11.738\\ +288.182\\ +712.764\end{bmatrix} \tag{6.7}$$

Submitting equation 6.7 into equation 6.4, the changes of the shape coefficients can be calculated as

$$\begin{bmatrix}\Delta a_1\\ \Delta a_2\\ \Delta a_3\end{bmatrix}=\begin{bmatrix}-4.3158\\ +2.5258\\ -0.006\end{bmatrix} \tag{6.8}$$

When the new shape coefficients are determined using equation 6.8, the final underside shape can be re-predicted. The following figure shows the re-predicted underside shape and the original shape, and it can be seen that the new generated underside shape is slightly different from the original prediction.

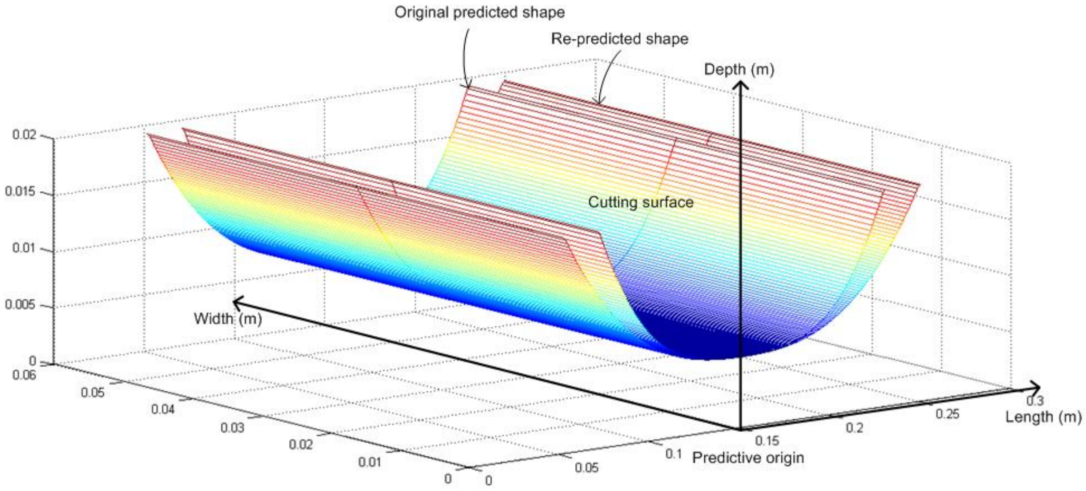


Figure 6.9 Original underside shape and the re-predicted underside shape (Inverted)

While the equations of the above examples relationships were manually solved in Excel, in the simulation the above equations will be solved by the fine tuning program automatically.  To achieve so, a “curve fitting” method is used in Matlab instead of using Excel, the program will calculate the parameters of these equations to make the curves match both the incremental natural frequencies and the experimental natural frequencies.

This page intentionally left blank.

# Chapter 7: CNC Machine Tool and Automatic Control

In this chapter, the physical machining centre and the method of tuning automation are presented. A CNC machine tool was adapted to mill the specified underside shape from a wood blank. The electronics and the hardware controlling program were combined with the milling machine so that the physical machining centre could operate under the control of the predictive model and the fine tuning program.

It can be seen from Figure 7.1 that the underside shape is generated by the predictive model, and along with the fine tuning program, rearranges the tuning process for both the software prediction and manufacture (described in Chapter 6). The underside shape predicted in each iteration was further edited to produce the tool path coordinates, namely the cutter locations (CL). This takes into account the selected components of the milling machine such as the type of stepper motor and lead screws. The coordinates are converted to the signals that the stepper motor driver could read, and therefore drive the lead screws of the milling machine for manufacture. All the signals were pre-calculated and sent to the driver as required, and hence the milling machine has no wait time. Following the manufacture, the resulting bar natural frequencies were measured. The signal was digitised by a computer sound card and the frequency spectrum was generated via FFT. The well separated peak responses were taken to be the natural bending frequencies.

The whole process starts with the underside shape prediction and ends with the frequency measurement. The program runs automatically except for the exciting action that was performed by the user. The automatic tuning process was controlled by Matlab, including the processes of the underside shape prediction, the fine tuning, control of the milling machine and the frequency measurements.

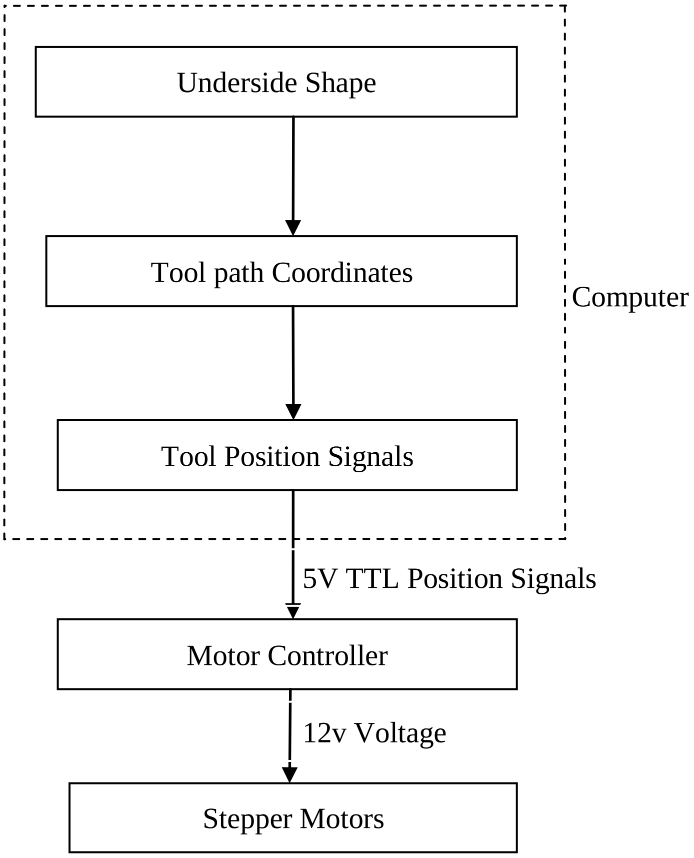


Figure 7.1 The procedure of the fine tuning automation

In this chapter, the above tasks are described in five sections. Section 7.1 presents the self-built CNC machine, which was converted from a SIEG X2 milling machine, to manufacture marimba bars. Section 7.2 describes the determination of the tool path coordinates. The method was simple and the accuracy was dependent on the accuracy of the mechanical components of the machine. Section 7.3 explains how the tool path coordinates that are calculated in Section 7.2 were converted into the CNC machine readable signals. This process connects the software prediction and the hardware manufacture. In section 7.4, the method of the automatic control of the machine spindle is described. This allows the program to decide when to turn the spindle on or off so that the spindle will not run throughout the whole tuning processes. Finally, Section 7.5 presents the frequency measuring program, which allows the program to analyse the sound that is picked up from the vibrating beam and estimate the natural vibration frequencies.

### 7.1 The Physical Machining Centre

The physical machining centre was built as shown in Figure 7.2. The components of the system include a computer that runs the software, a stepper motor driver that receives the signals from the software and drives the stepper motors of the CNC machine, three NEMA23 stepper motors, a power supply, and a SIEG X2 milling machine. The combination of these components upgraded the original milling machine (SIEG X2 milling machine) into a PC-controlled CNC machine so that the manufacture of marimba bars could be undertaken automatically.

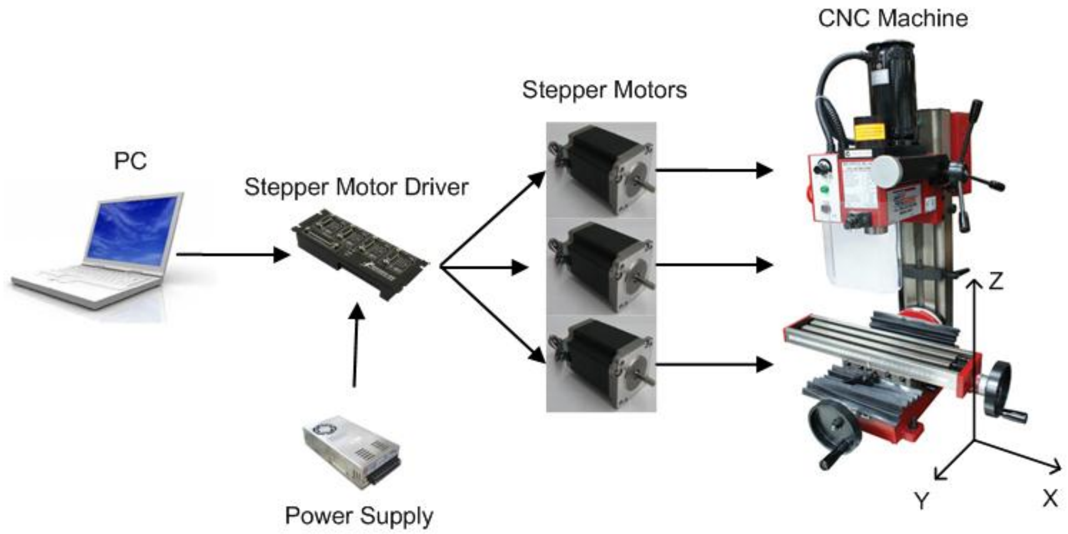


Figure 7.2 Selected components of the physical machining centre

The SIEG X2 milling machine has longitudinal (X axis) travel of 228 mm, cross (Y axis) travel of 100mm and vertical (Z axis) travel of 190mm. These limit the maximum dimensions of the predicted underside shape of marimba bars. The spindle speeds available are 100 to 2500 rpm. To upgrade such milling machine to a PC-controlled CNC machine, new lead screws were installed and “16 TPI 2-start ACME” lead screws were selected, which give eight motor turns for one inch travel on the linear axis. NEMA 23 stepper motors were selected to drive the CNC machine table, they have two hundreds motor steps (or two thousands micro-steps) for each turn. The following photograph Figure 7.3 shows the complete machining centre.

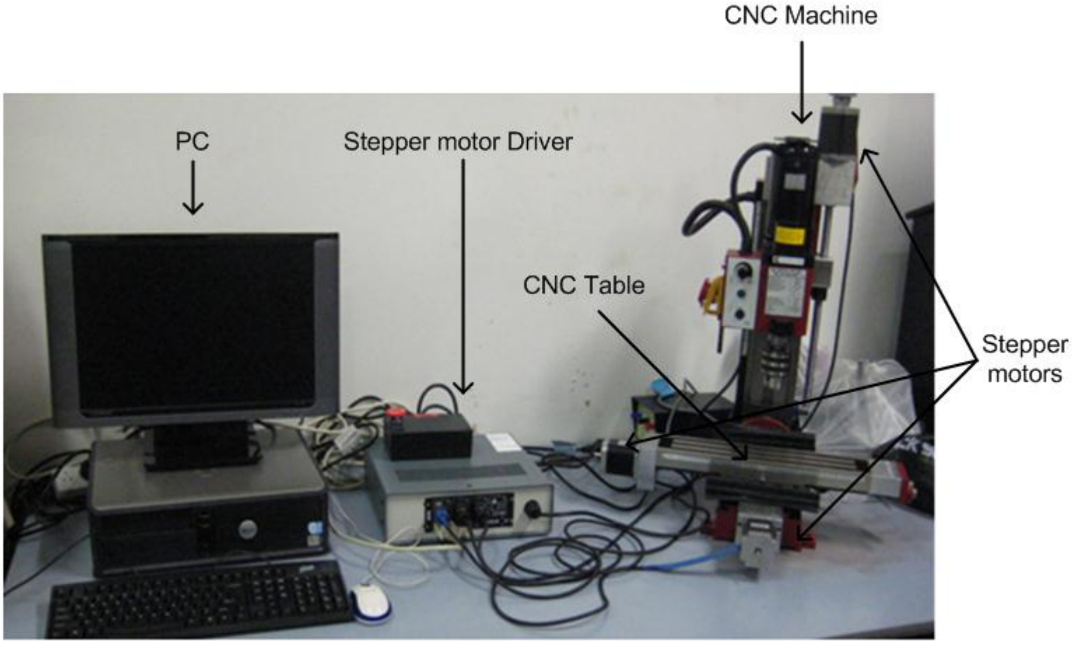


Figure 7.3 The complete physical machining centre

The accuracy of the completed milling machine was tested. The machine table was controlled to move forwards and backwards many times and it was found that the repeatability error was less than 0.05 mm. (The possible source of error might be the specification information of the lead screw that is provided by the manufacturer as the stepper motor is rather accurate and rarely lose steps.) Therefore, some adjustments were made to define the exact value of the linear travel distance per motor step by manually varying this value and measuring the true moving distance by experiment, and an error of less than 0.01 mm was achieved for the required linear distance of 300 mm. This proved the accuracy of the mechanical parts of this machine. However, it cannot prove that the distance that the CNC table travels will produce the predicted thickness of the marimba bar as the calculations of the tool path coordinates cannot yet be tested in this stage, and the details of the tool path coordinates will be described in the following two sections.

### 7.2 Determination of the Tool Path Coordinates

The undercut surface of a marimba bar is the surface where the CNC cutter tip travels. The coordinates of the points on this surface can be presented in the Cartesian coordinate system as shown in Figure 7.4. The cutter contact (CC) points would lie on this surface, whereas the cutter locations (CL) are defined as the locations where the CNC machine is instructed to hold the cutter. For this thesis, the cutter locations are considered as the locations where the centre of the cutter, a ball nose slot drill’s tip, travels.

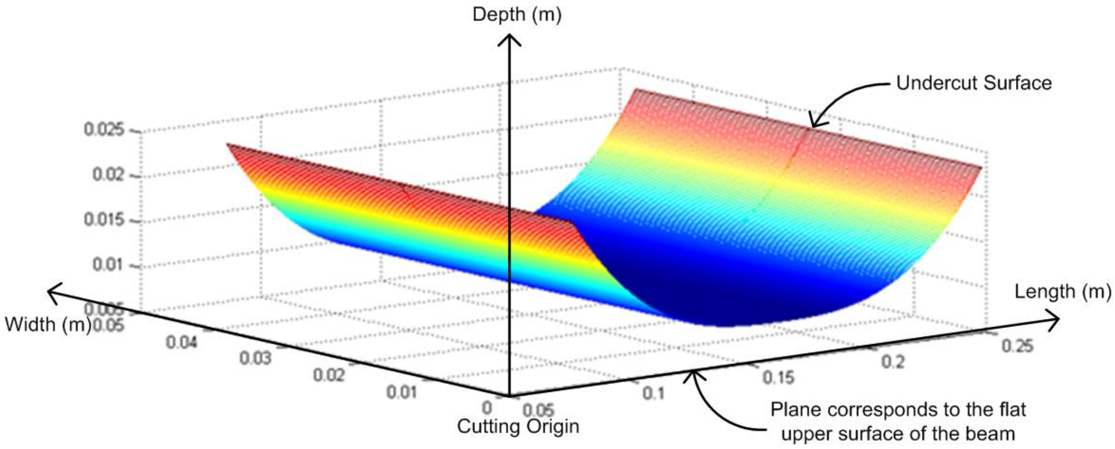


Figure 7.4 An underside surface of a marimba bar

The determination of the tool path coordinates included two steps. The first step was the determination of the cutting increments. The cutting increment is considered as the linear distance that the cutter travels for each motor movement. (It can be a single motor-step or many motor-steps.) The CNC stepper motors produce fixed linear travelling length for each motor-step, which limits the minimum distance between two cutter locations. The cutting increment depends on the specifications of the selected stepper motors and the lead screws. The second step was the determination of the tool path (cutter location). The tool path coordinates that are produced in this section are used to drive the self-built CNC machine. The G-code can then be generated based on the tool path coordinates for other commercial CNC machines (e.g. Leadwell V30).

For this thesis, the NEMA23 stepper motors and the “16 TPI 2-start ACME” lead screws were selected for the self-built CNC machine, giving a minimum travelling distance of 0.0015875 mm per micro-step as calculated in equations 7.1 and 7.2. It can be seen that to obtain one inch linear travelling distance the stepper motor needs to rotate for eight turns, and each turn of the motor step has 200 motor-steps (or 2000 micro-motor steps), which gives 16000 micro-steps for one inch of the linear travelling length as shown in equation 7.1. This is converted to metres per micro-step in equation 7.2, which simplifies the program as all the units are same.

$$\frac{\text{Steps}}{\text{Inch}} = \frac{\text{Steps}}{\text{Turn}} * \frac{\text{Turns}}{\text{Inch}} = 2000 * 8 = 16000 \text{ (Microsteps/inch)} \tag{7.1}$$

$$\frac{\text{Linear Distance}}{\text{Motor Step}} = \frac{\text{Lead screw Inch}}{\text{Motor Step/inch}} = \frac{1}{16000}\left(\frac{\text{inch}}{\text{step}}\right) = 0.0000015875\ (\frac{\text{m}}{\text{step}}) \tag{7.2}$$

When the CNC cutter moves with smaller steps, it produces smoother surface but takes longer. Therefore, a compromise design of the cutting increment makes the cut surface as smooth as possible within an acceptable length of time. The value obtained in equation 7.2 gives the minimum length that the selected motor could achieve, and the cutting increment can then be adjusted in equation 7.3.

$$\text{Cutting increment} = (\text{Motor steps}) * (\frac{\text{Linear Distance}}{\text{Motor Step}}) \tag{7.3}$$

The cut surface is achieved from the cutter contact surface and the cutter's radius as shown in Figure 7.5. The centre of the ball-nose radius is used as the reference for the tool path calculations. The actual surface is defined by the geometry as shown in Figure 7.5 and the local slope of the undercut curve.

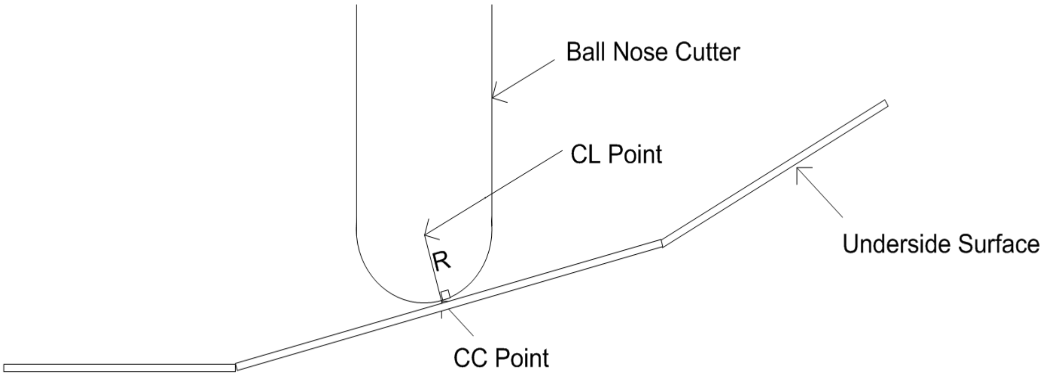


Figure 7.5 The determination of cutter locations

The tool path curves can be achieved by linking all the cutter location points together. However, the cutter location points that are obtained at this stage might not be suitable for the machine as some cutter location points are too close and some are not, depending on the curvature of the underside shape. To solve this problem, these cutter location points were linked together to form a smooth underside curve using the "curve fitting" function in Matlab. The cutter location points that are expected to lie on this curve were re-calculated based on the minimum cutting increments, which were calculated from equation 7.3.

Furthermore, the tool path generation can be designed in different ways, and it takes different amount of time in simulation. As shown in Figure 7.6, the Y axis is set to be the direction of the cutter's forward steps, and X and Z axes are set to be direction of the cutter's side steps, which is along the length of the marimba bar.

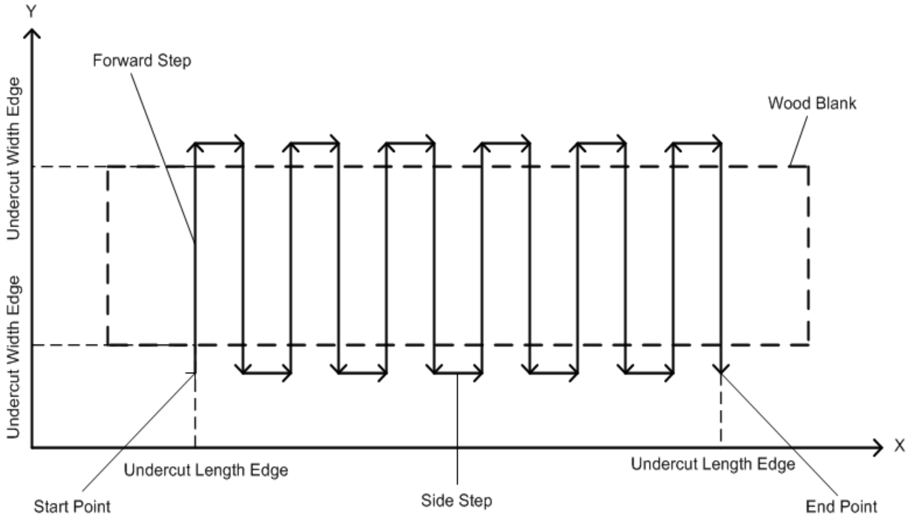


Figure 7.6 Tool path directions (Plane view)

Once the direction of the tool path is decided, the tool path coordinates can be rearranged and stored separately for X, Y and Z axes. The following figure shows the resulting tool path and the resulting underside surface using a ball-nose cutter with 10mm diameter.

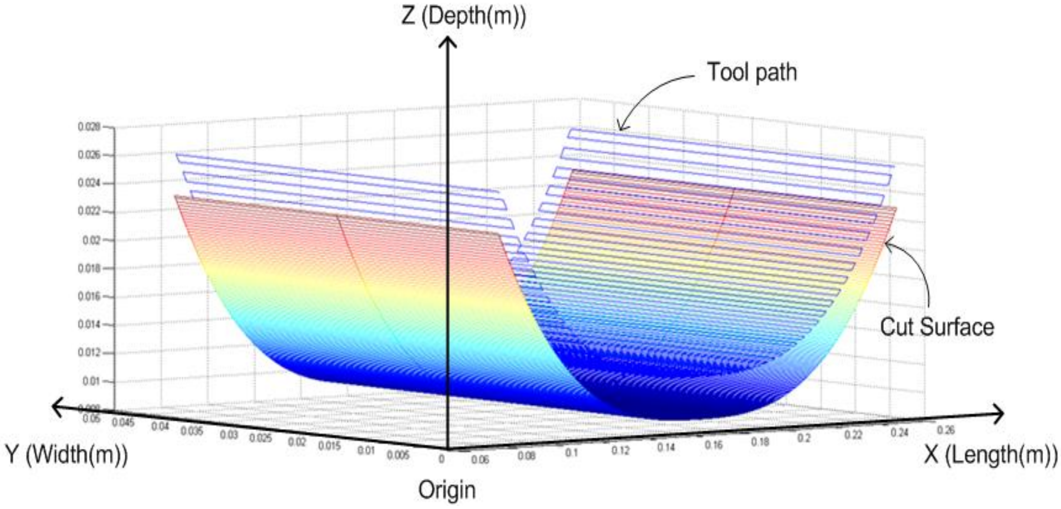


Figure 7.7 The achieved tool path and the underside surface

With the tool path determined, commercial G-code (e.g. for Leadwell V30) was generated. The differences between the above two codes are the coordinate systems that they are based on and the printed format. The G-code in Appendix D that are generated by Matlab from Appendix G.18 are the same as the one produced by AlphaCam (a CAM software) as shown in Appendix E. This allows Matlab to complete similar tasks for other applications as CAM software does, such as being a CAM program so that Matlab files can be used directly by CNC machines. The following flowchart shows the process of the tool path generation.

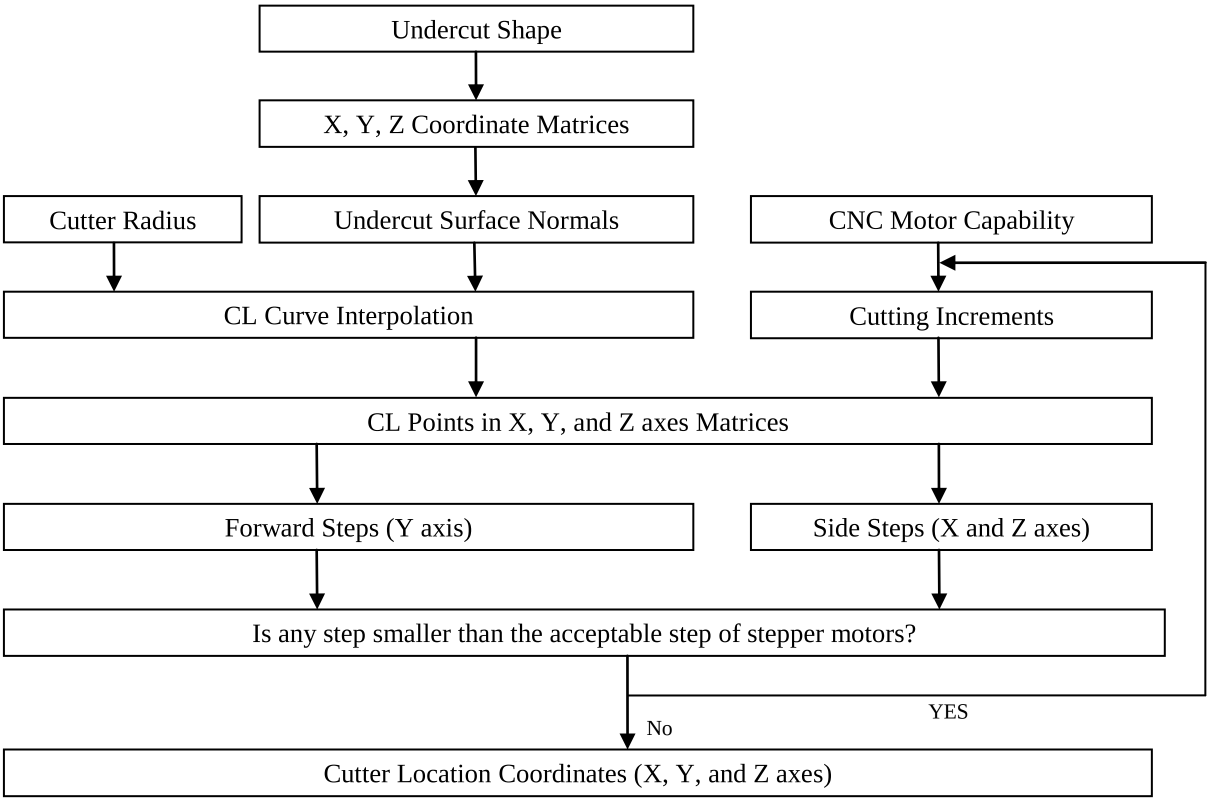

Undercut Shape
X, Y, Z Coordinate Matrices
Cutter Radius
Undercut Surface Normals
CNC Motor Capability
CL Curve Interpolation
Cutting Increments
CL Points in X, Y, and Z axes Matrices
Forward Steps (Y axis)
Side Steps (X and Z axes)
Is any step smaller than the acceptable step of stepper motors?
YES
No
Cutter Location Coordinates (X, Y, and Z axes)

Figure 7.8 A flowchart of the tool path generation

### 7.3 Automatic Cutter Control

As shown in Figure 7.2, the stepper motors cannot be controlled by the tool path coordinates directly as the driver cannot read digital signals. Thus, the tool path coordinates that are achieved in Section 7.2 were translated into step and direction signals. (The movement of stepper motors requires both of them.) One motor step requires one step signal at high level (“1”), and zero step on stepper motor also requires one step signal but at low level (“0”). Similarly, if the stepper motor operates at the positive direction, the direction signal is at high level (“1”), otherwise, it is at low level (“0”). To convert tool path coordinates into the step and direction signals, the directions of these movements were firstly determined. Thus, for different cutter movement, if the number of steps of that movement is non-zero a high-level step signal (“1”) will be replaced into the array, otherwise, a low-level step signal (“0”) will be stored, and stepper motors operate when the signal pulse changes.

Figure 7.9 shows the fundamental principle of the stepper motor control. The step signals are presented at the upper place and the pulse signals are presented at the lower place. It can be seen that the stepper motor moves two steps. Thus, the step signals keep at the high level for two steps. To make two steps, the pulse signals need to switch twice between low and high levels.

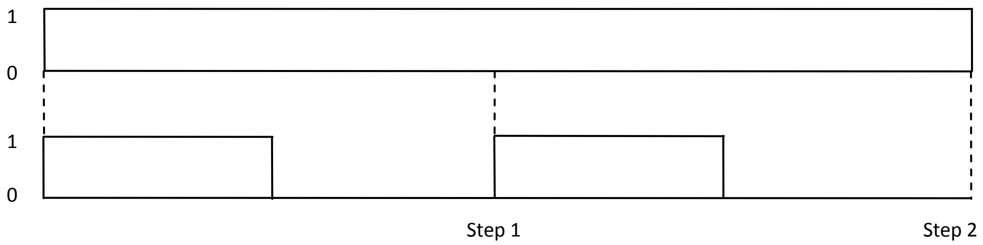


Figure 7.9 Step signals and step pulses

However, if two stepper motors operate at the same time but with different speed (or different travelling distance), the controlling pulses are in different sequence although the numbers of steps for both motors are same. In this thesis, the “side-step” direction of the tool path is set on X axis. Thus, the amount of steps for each cutter’s movement on X axis is fixed, and the cutter moves with the varying Z axis tool path coordinates. The operating speed of the stepper motor on Z axis varies for different cutter movements, and the pulse changes occur less than the ones on X axis when it operates slower, vice versa. Therefore, the stepper motors on X and Z axes would be required to operate at the same time, but the travelling length of them are not the same.

The following figure shows one of the examples of how the pulse signals on X and Z axes work. The stepper motor on the X axis gives five steps while the stepper motor on Z axis only gives two steps. For a “five-step” movement on X axis, ten pulses are required (“1”s and “0”s), and a “two-step” move on Z axis needs four signals. Therefore, the duration of pulses on Z axis is set to be the same as the time on X axis.

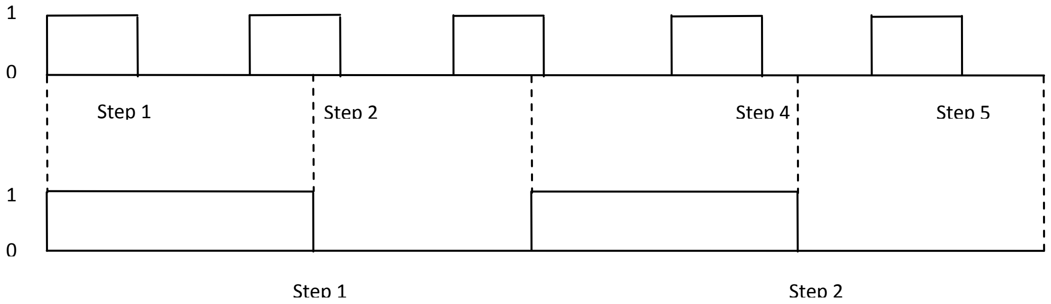


Figure 7.10 X and Z Step Signals

It can be seen in Figure 7.10 that the first pulse change on the Z axis occurs after the third pulse and before the fourth pulse on Z axis, which is unable to achieve as pulses are sent out discretely and there is no position between two discrete points on X axis. Thus, the pulses on Z axis should be adjusted so that they can be sent out at the same positions as the ones on X axis. As shown in Figure 7.11, the positions of the pulses Z axis have been rearranged. The second Z pulse is closer to the fourth pulse on the X axis, etc. Once the new positions of pulses on Z axis are obtained, the step signals (high-level “1” and low-level “0”) are placed into these positions.

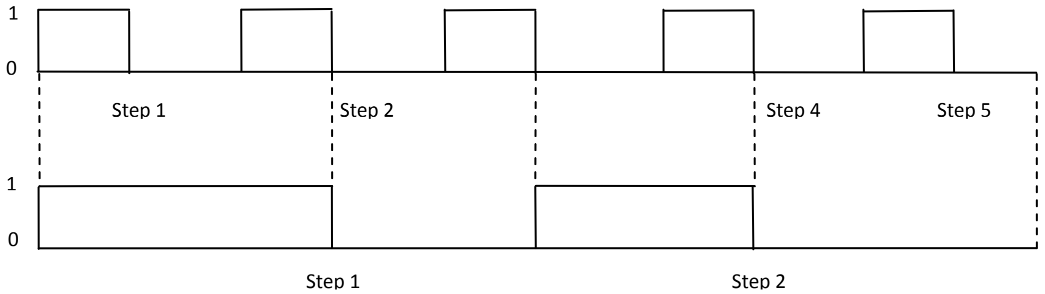


Figure 7.11 Modified X and Z step signals

According to the fact that the travelling length on Y axis is much larger than the ones on the other two axes, it requires much more PC memory. To avoid this problem the travelling length on Y axis is made to be only one step. Thus, only two pulses are required for the movements on the Y axis. During the operation, the program repeats the process of moving the stepper motor forwards and backwards for the number of the steps as required, saving the memory and time in simulation. The CNC machine that is used in this thesis is controlled by Matlab through the parallel port on the PC, and six pins of it were used to drive three stepper motors. The following two flowcharts show the controlling processes.

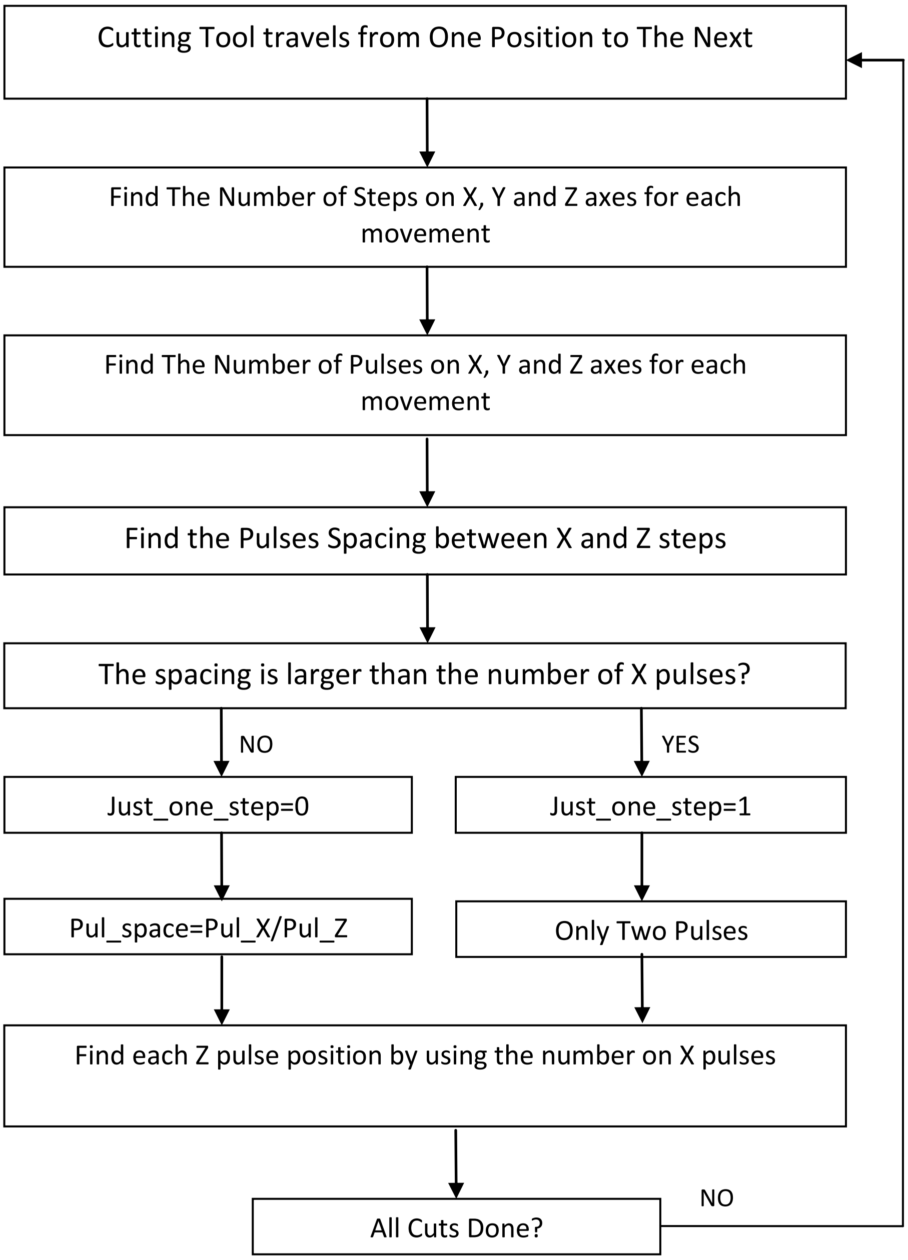

Cutting Tool travels from One Position to The Next
Find The Number of Steps on X, Y and Z axes for each movement
Find The Number of Pulses on X, Y and Z axes for each movement
Find the Pulses Spacing between X and Z steps
The spacing is larger than the number of X pulses?
NO
YES
Just_one_step=0
Just_one_step=1
Pul_space=Pul_X/Pul_Z
Only Two Pulses
Find each Z pulse position by using the number on X pulses
All Cuts Done?
NO

Figure 7.12 A Flowchart of the generation of the tool path signals (1)

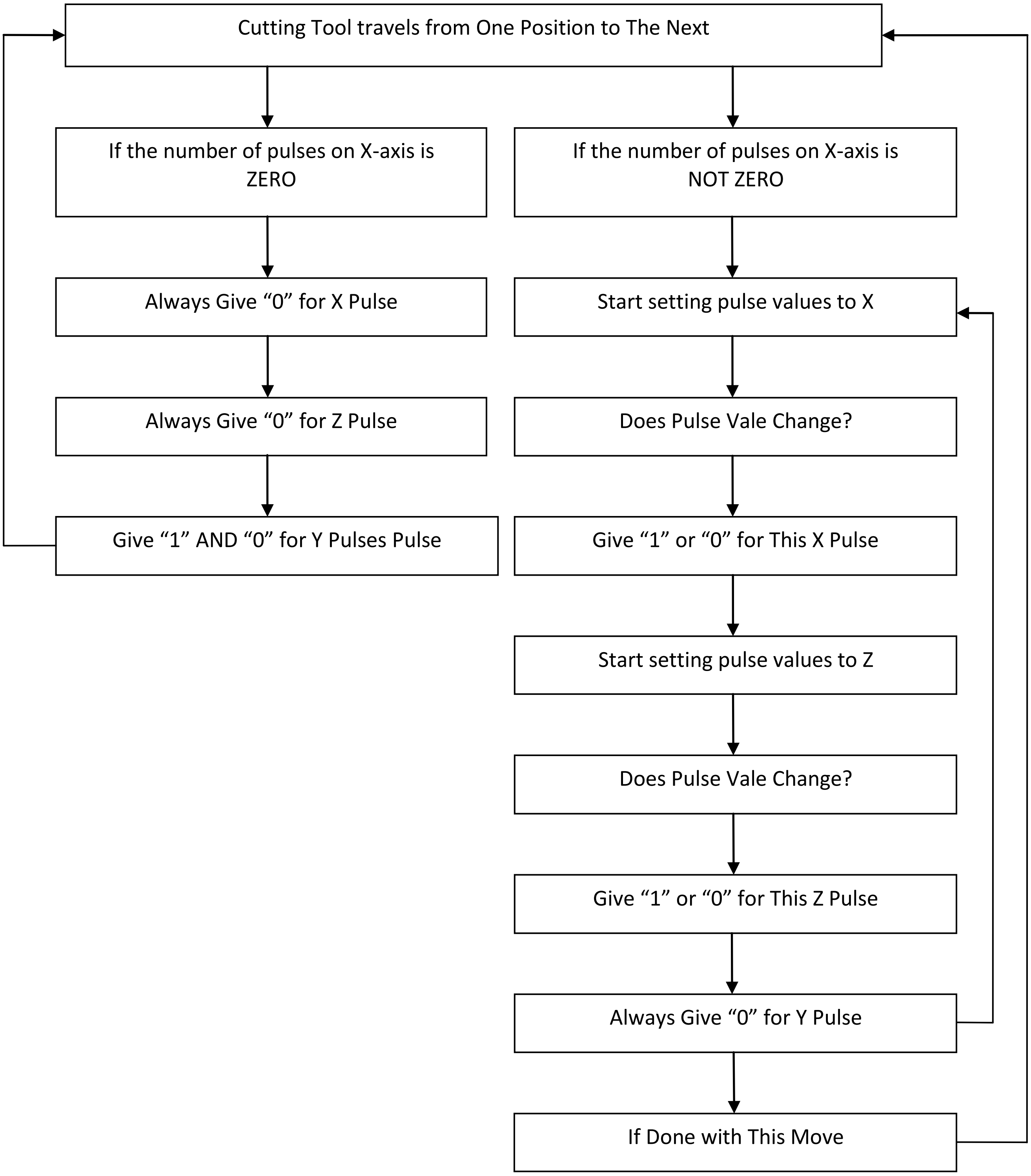

Cutting Tool travels from One Position to The Next
If the number of pulses on X-axis is ZERO
Always Give “0” for X Pulse
Always Give “0” for Z Pulse
Give “1” AND “0” for Y Pulses Pulse
If the number of pulses on X-axis is NOT ZERO
Start setting pulse values to X
Does Pulse Vale Change?
Give “1” or “0” for This X Pulse
Start setting pulse values to Z
Does Pulse Vale Change?
Give “1” or “0” for This Z Pulse
Always Give “0” for Y Pulse
If Done with This Move

Figure 7.13 A Flowchart of the generation of the tool path signals (2)

### 7.4 Automatic Spindle Control

The spindle speed on the SIEG X2 milling machine was originally controlled by a potentiometer with a switch on the control panel. There are five wires connected to the control board at the back of the control box. Two wires connect the switch ends of the potentiometer. As it is a “normally open” switch the spindle is activated and the voltage drops across the spindle motor once it is “closed”. The voltage can be varied by changing the resistance on the potentiometer. The potentiometer had to be at its “off” position before the spindle is switched on, which was designed as the self-protection function for this control board. To control the spindle automatically from the controlling program the potentiometer was no longer used. The switch on the potentiometer was replaced by a solid state relay, controlled by a 12 voltage DC power supply and the PC signal. Thus, the controlling program sends out logic signals to turn the relay on and off. A 9 volt battery supplies voltage to a time-delay circuit (achieved by an R-C charging circuit). It charges the capacitor from 0 volts to 7 volts, and this voltage is used to supply the spindle. The potentiometer is always on its “off” position so that the self-protection function of control board is always ensured. The modified control board is shown below.

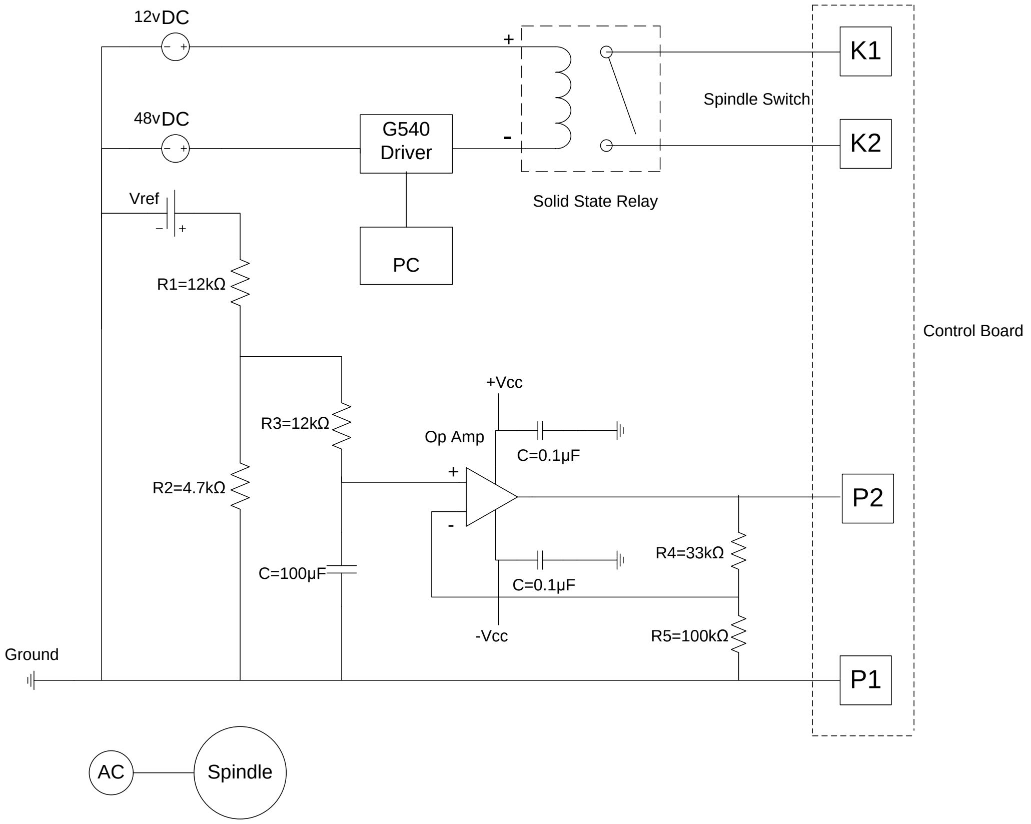


Figure 7.14 A PC-controlled spindle board

## 7.5 Frequency Measurement

The frequency that marimba bars vibrate at was measured and analyzed by a frequency measuring program. The sound produced by the marimba bar is picked up by a microphone and sent to a sound card as shown in Figure 7.15. This frequency measuring program analyzed the sound signal, converting the time-based signal into a frequency-based signal. In this section, the method of measuring and analyzing the resulted natural frequencies is presented.

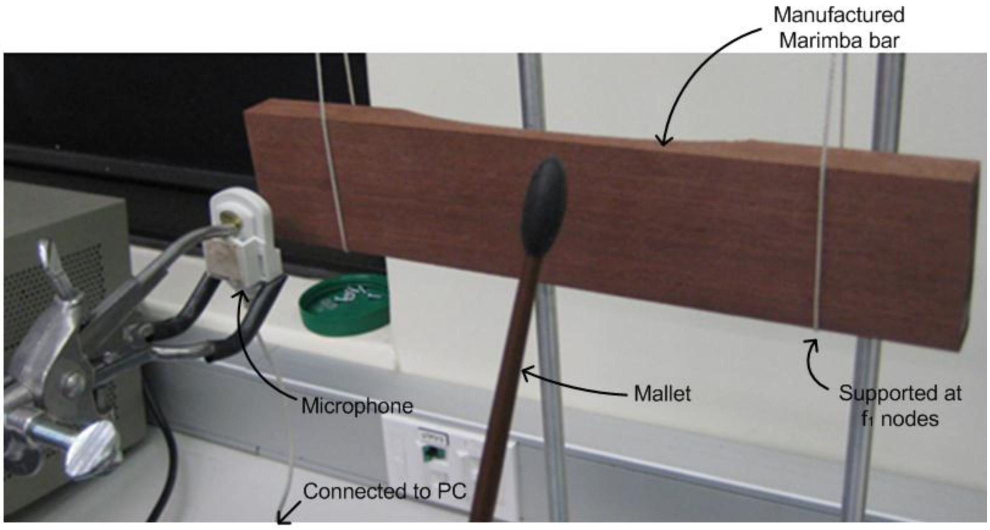


Figure 7.15 Photograph of frequency measurement

First of all, the “Data Acquisition toolbox” of Matlab was used to pick up the sound. The hardware device and channels were firstly initialized. An analog input device object was created and the channels for data input were opened.

```
AI=analoginput('winsound',0);
chan=addchannel(AI,1);%AI=analogue input
```

The above commands define the sound card to be the analog input device and name it as “0”. The program then opens channel 1 for the data input channel. The AI object properties would be set following users requirements such as the sample rate (Fs), duration, trigger type, time out, etc. “Manual” means that the sound card is triggered manually after command “trigger”. The sample rate is the number of samples that are recorded per second, and once the sample rate is available, the timing for each sample can be calculated as “1/Fs”. The total time required can be calculated as:

$$\text{Time} = \text{Samples} * (\text{timing per sample}) \tag{7.4}$$

“Samples per trigger” can be calculated by

$$\text{Samples per Triger} = \text{Duration} * (\text{Sample Rate}) \tag{7.5}$$

The following shows an example of the initialization information for this program.

```
set(AI,'TriggerChannel',chan)
duration = 1;
SampleRate = 44100;
set(AI,'SampleRate',SampleRate);
set(AI,'SamplesPerTrigger',duration*SampleRate);
set(AI,'TriggerType','Software');
set(AI,'TriggerCondition','Rising');
set(AI,'TriggerConditionValue',0.2);
```

The hanning windowing function and the FFT analysis method were used to analyze the sound signal. The sampled signal was multiplied by the hanning function, and the ends of the time records were forced to zero. The program was designed to read out the measured frequencies automatically, the peaks of the measured frequencies were found in the regions around the target frequencies. The results were also plotted and show the uncalibrated magnitude of the signal.

An aluminum beam that was manufactured by Entwistle and McGrechan (2007) was used to test the accuracy of this frequency measuring program. The target fundamental natural frequency is 220Hz as the known result. If the fundamental frequency that is obtained by this frequency measuring program matches with this value, the program could then be used with confidence.

The sound card records 44100 samples each second (or 0.0000226 second per sample). The recording time was set at two seconds, which gives 88200 samples.

Thus, $\Delta t$ becomes "$\frac{2}{88200}$" and $\Delta f$ gives $\frac{1}{N\Delta t}$ in the frequency domain, which is 0.5 Hz.

Figure 7.16 shows the measured natural bending frequencies of this 220Hz aluminium beam. It can be seen that the measured results are the same as the target values, which validates the accuracy of this frequency measuring program.

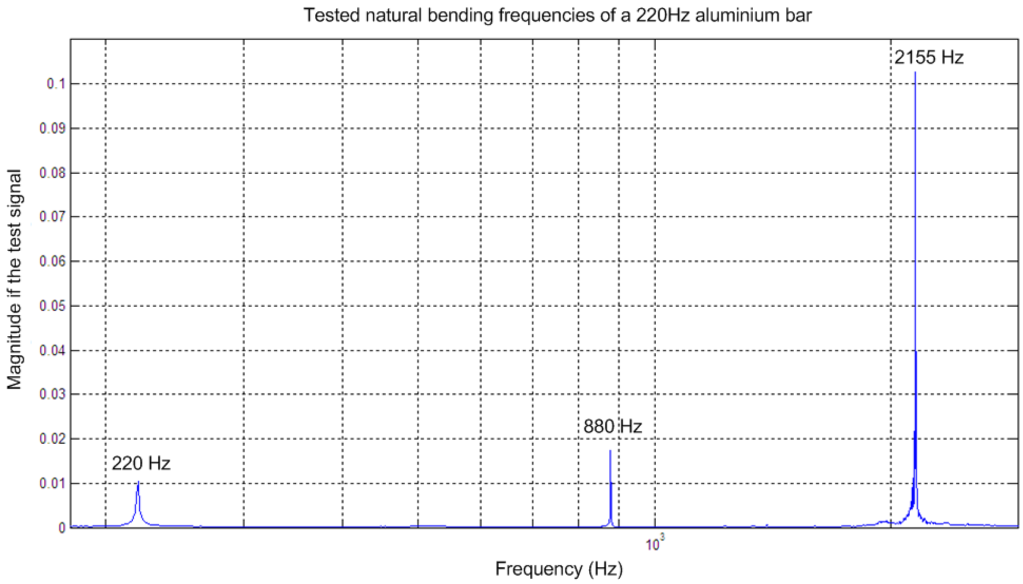


Figure 7.16 Measured frequencies for an aluminum beam

# Chapter 8: Experiments and Results

In this chapter, the underside shape-predictive model is tested for wooden beams using the "Leadwell V30" CNC machine (Section 8.1). Entwistle and McGreham (2007) verified that the predictive model is accurate for aluminium bars. This section shows how accurate this predictive model is for tuning wooden beams, and it will further show whether the method of solving the mechanical properties of wooden beams is appropriate for the fine tuning process. In Section 8.2, the fine tuning program (described in Chapter 6) is tested using the physical machining centre that was developed for this thesis (described in Chapter 7). The method for the measurement of the predictive model's inputs, which were obtained from Section 8.1, were used in the preparation of the fine tuning process. The procedure and the results obtained are presented and a comparison between results available in the literature and those obtained in this thesis is made.

## 8.1 The Underside Shape Prediction Test

In this section, the underside shape predictive model is tested following the procedure in Figure 8.1. The "Leadwell V30" CNC machine was used for this test to mill the wood blank (made of Jarrah) using the predicted dimensions converted into the CNC readable code (described in Chapter 7).

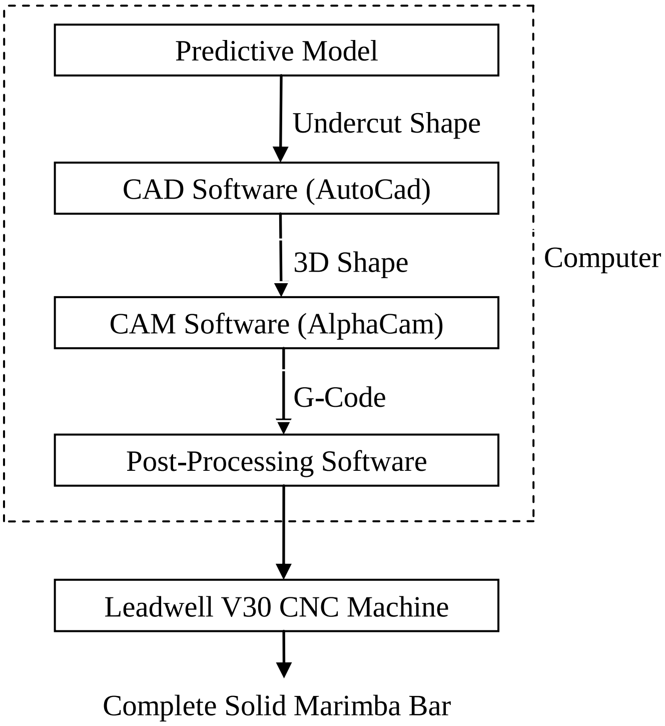


Figure 8.1 The predictive model testing procedure

As shown in Figure 8.1 the underside shape predictive model generates the correct dimensions of the non-prismatic underside shape of a marimba bar when given the target natural frequencies. The CAD software (AutoCAD) makes a 3D shape based on the predicted geometry, and the CAM software (AlphaCam) is then used to convert the 3D shape so that the manufacture can be simulated. Once the simulation is successful the tool path will be converted into CNC machine readable code (G-code) in AlphaCam. The Leadware V30 CNC machine will then manufacture the wood blank based on those G-code.

For this section, it is important to ensure that the coordinates on the “Leadwell V30” CNC machine are the same as the coordinates used in the AlphaCam software. The wood blank was expected to be clamped perfectly on the CNC machine table so that the error can be minimized. As shown in Figure 8.2, an angle plate was placed on the table of this CNC machine and aligned with the machine axis. The edge finder, which has the diameter of 1.25 mm, was then placed at the upper right corner of the angle plate to determine the X and Y coordinates of that point, where the wood blank would be placed (This position is considered as the origin of cutting).

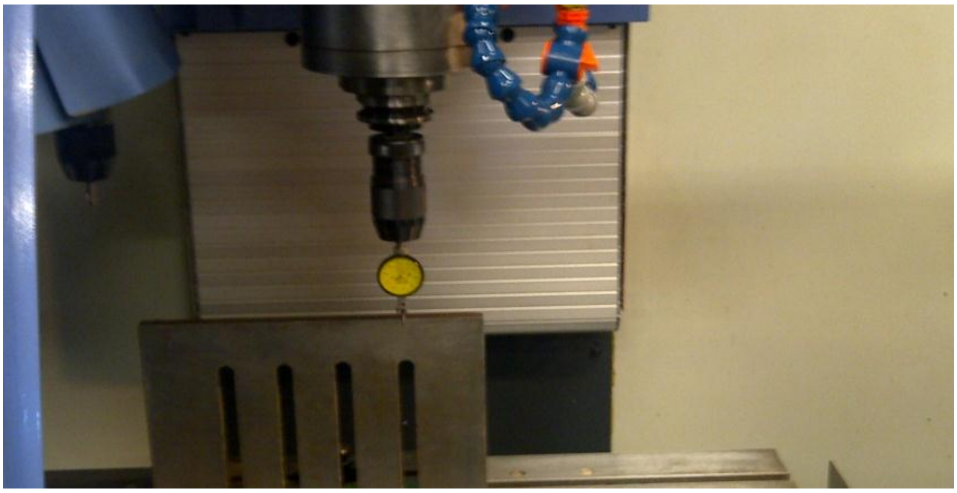


Figure 8.2 Edge finder checking angle plate

The coordinates of the origin on the X and Y axes were found to be:

$$X = 499.16 \text{ (mm)}$$
$$Y = -293.48 \text{ (mm)}$$

The coordinates on the X and Y axes are especially important for this test as the thickness coordinates of the predicted marimba bar lie on the Y axis of this Leadwell machine, and the length of the bar lies on the X axis as shown in Figure 8.3. Any difference of the X and Y coordinates between the simulation and the actual setup results large error on the resulting natural frequencies.

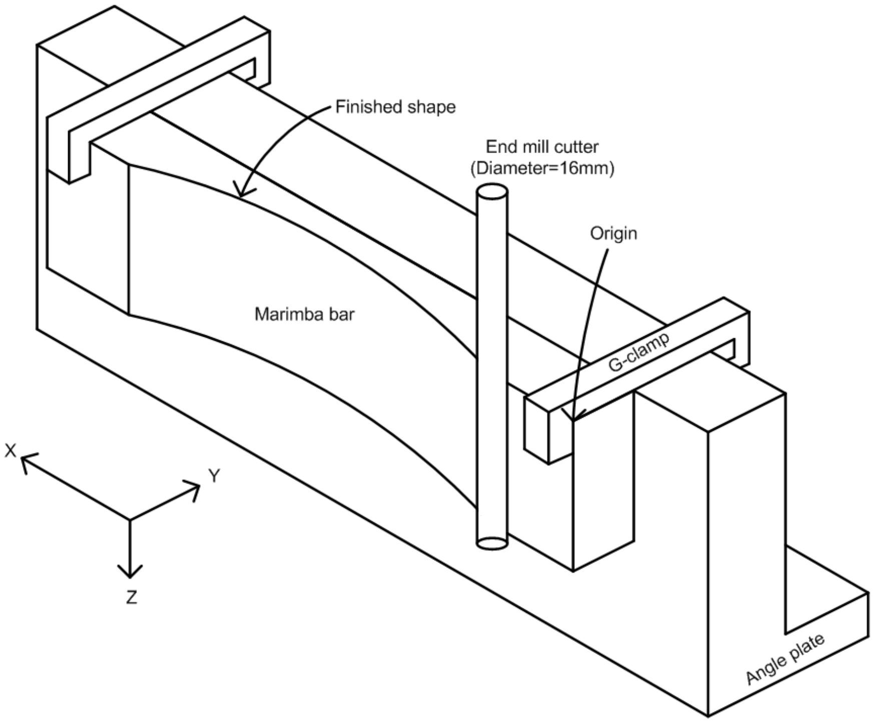


Figure 8.3 Machine setup

The Z coordinate of the origin was measured with the help of a cigarette paper to detect proximity. When the cutter was turned on and slowly moved downwards (close enough) to the wood blank, the coordinate was recorded when the cigarette paper was just removed by the cutter. This ensured that the cutter was within 0.05 mm of the wood blank. Figure 8.4 shows the Z origin, and it gives the Z coordinate of the cutting origin to be

$$Z = -255.11\ (mm)$$

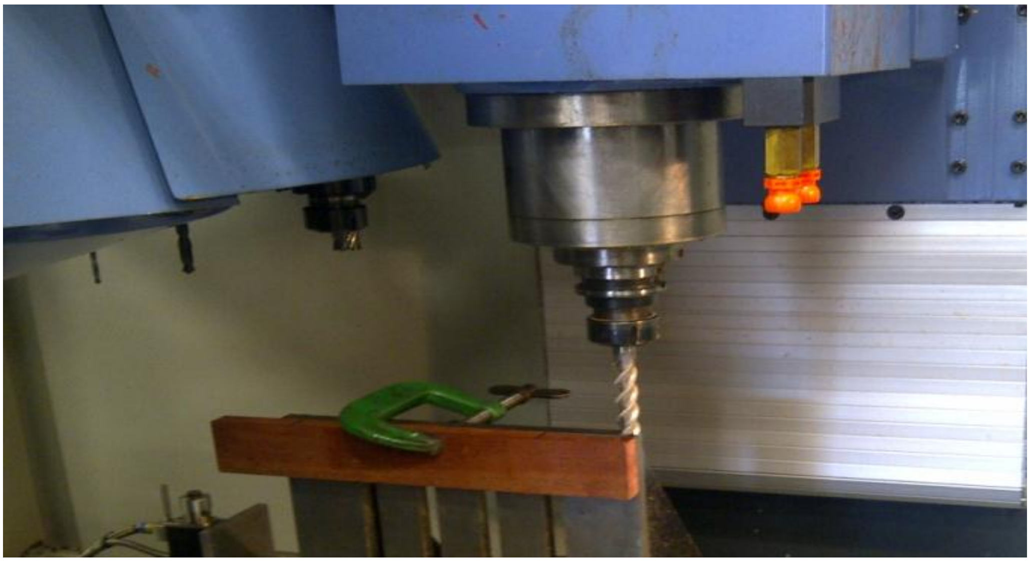


Figure 8.4 CNC cutter finding Z origin

Figure 8.5 shows the wooden test blank mounted on the angle plate, with a G-clamp placed at the corner of the blank clear of the cutter.

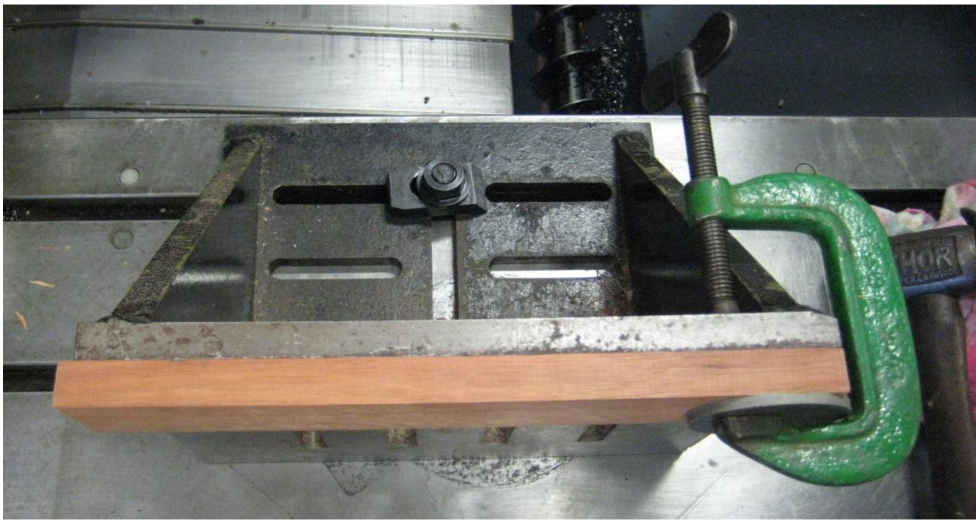


Figure 8.5 The complete setup for the angle plate

The underside shape prediction started with inputting the mechanical properties of the tested wood into the predictive model, and setting the target frequencies for the lowest three natural frequencies to 220 Hz, 880 Hz and 2156 Hz. The following table shows the inputs and the predicted results.

| Properties | |
|---|---|
| $\rho$ (kg/m$^3$) | 823.22 |
| E (GPa) | 22.36 |
| G (GPa) | 9.97 |
| L (mm) | 299.0 |
| b (mm) | 52.0 |
| $d_o$ (mm) | 19.11 |
| Desired frequencies | |
| $f_1$ (Hz) | 220 |
| $f_2$ (Hz) | 880 |
| $f_3$ (Hz) | 2156 |
| Predicted coefficients | |
| $a_1$ | 6.7343 |
| $a_2$ | 0.0808 |
| $a_3$ | 0.0044 |

Table 6: Inputs and original underside shape prediction

The predicted underside shape of the marimba bar is shown in Figure 8.6. It was then input to a CAM software (AlphaCam) so that the predicted dimensions of this underside shape could be converted into G-code, which are readable for the Leadwell V30 CNC machine.

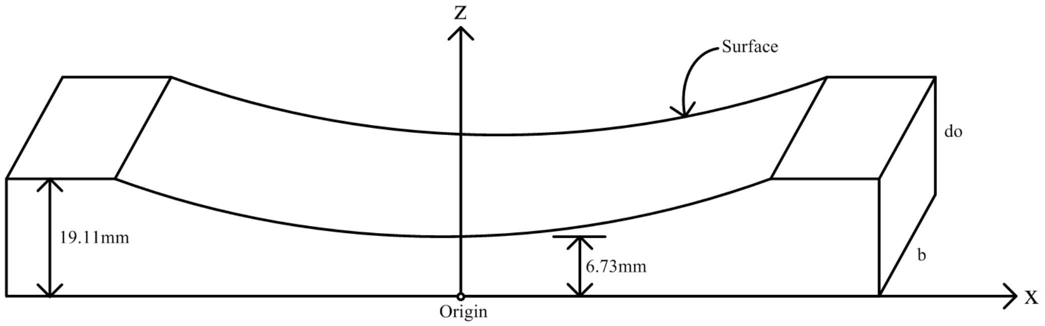
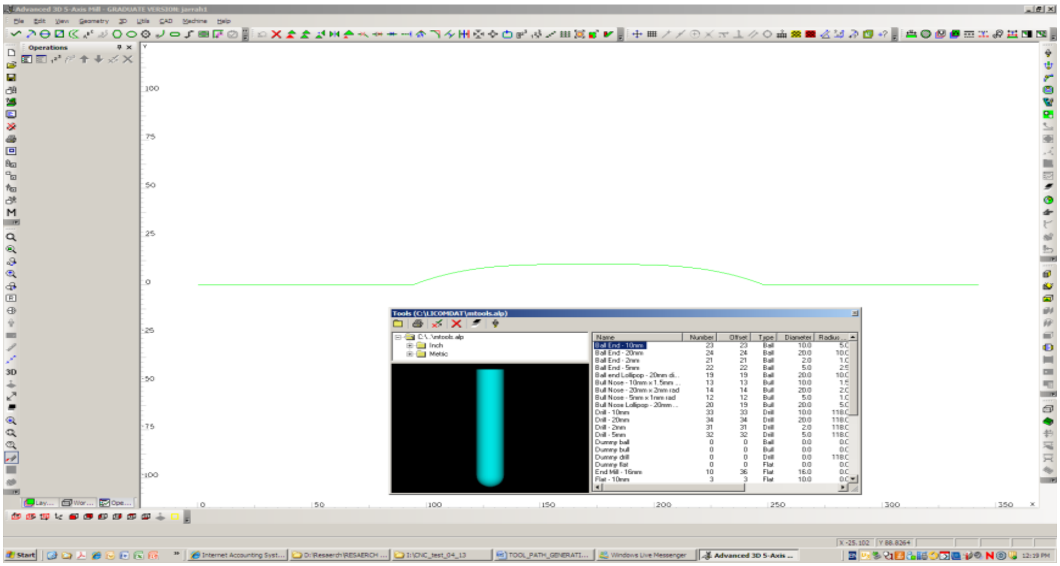


Figure 8.6 Predicted underside shape from the model (Inverted)

Figure 8.7 shows the manufacture simulation on AlphaCam. The material was removed in three passes, each with 8 mm depth of cut. The final pass required a small depth of cut, improving accuracy.



| Name | Number | Offset | Type | Diameter | Radius ... |
|---|---|---|---|---|---|
| Ball End - 10mm | 23 | 23 | Ball | 10.0 | 5.0 |
| Ball End - 20mm | 24 | 24 | Ball | 20.0 | 10.0 |
| Ball End - 2mm | 21 | 21 | Ball | 2.0 | 1.0 |
| Ball End - 5mm | 22 | 22 | Ball | 5.0 | 2.5 |
| Ball end Lollipop - 20mm di... | 19 | 19 | Ball | 20.0 | 10.0 |
| Bull Nose - 10mm x 1.5mm ... | 13 | 13 | Bull | 10.0 | 1.5 |
| Bull Nose - 20mm x 2mm rad | 14 | 14 | Bull | 20.0 | 2.0 |
| Bull Nose - 5mm x 1mm rad | 12 | 12 | Bull | 5.0 | 1.0 |
| Bull Nose Lollipop - 20mm ... | 20 | 19 | Bull | 20.0 | 5.0 |
| Drill - 10mm | 33 | 33 | Drill | 10.0 | 118.0 |
| Drill - 20mm | 34 | 34 | Drill | 20.0 | 118.0 |
| Drill - 2mm | 31 | 31 | Drill | 2.0 | 118.0 |
| Drill - 5mm | 32 | 32 | Drill | 5.0 | 118.0 |
| Dummy ball | 0 | 0 | Ball | 0.0 | 0.0 |
| Dummy bull | 0 | 0 | Bull | 0.0 | 0.0 |
| Dummy drill | 0 | 0 | Drill | 0.0 | 118.0 |
| Dummy flat | 0 | 0 | Flat | 0.0 | 0.0 |
| End Mill - 16mm | 10 | 36 | Flat | 16.0 | 0.0 |
| Flat - 10mm | 3 | 3 | Flat | 10.0 | 0.0 |

Figure 8.7 The design of the cutting process on AlphaCam

Figure 8.8 shows the simulated result after the final cut, which allowed checking for mistakes.

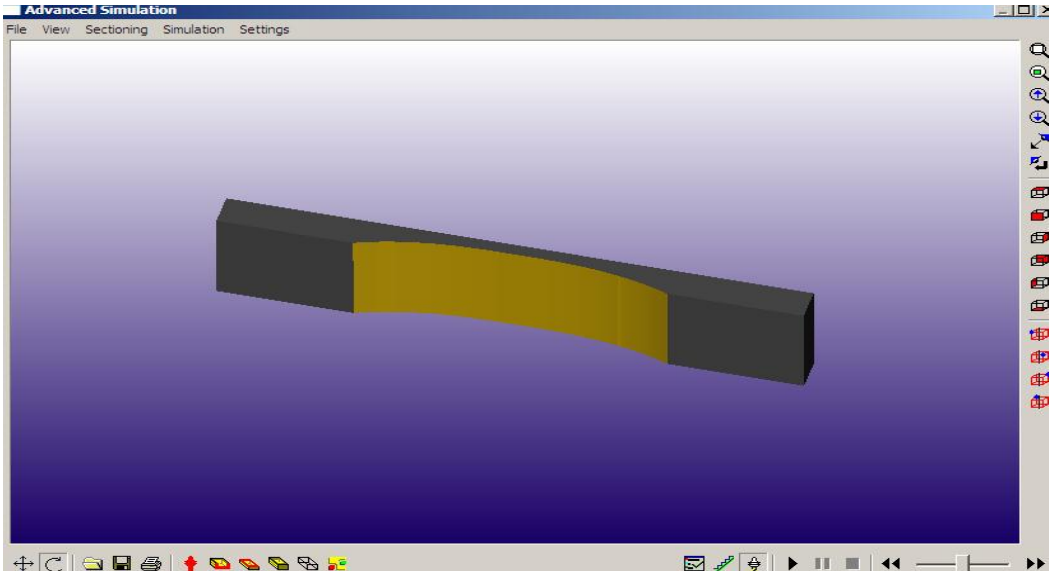


Figure 8.8 Simulation of the cutting process on AlphaCam

Figure 8.9 shows the complete marimba bar that was cut by the Leadwell V30 CNC machine. The actual dimensions were checked against the user-desired (target) natural frequencies.

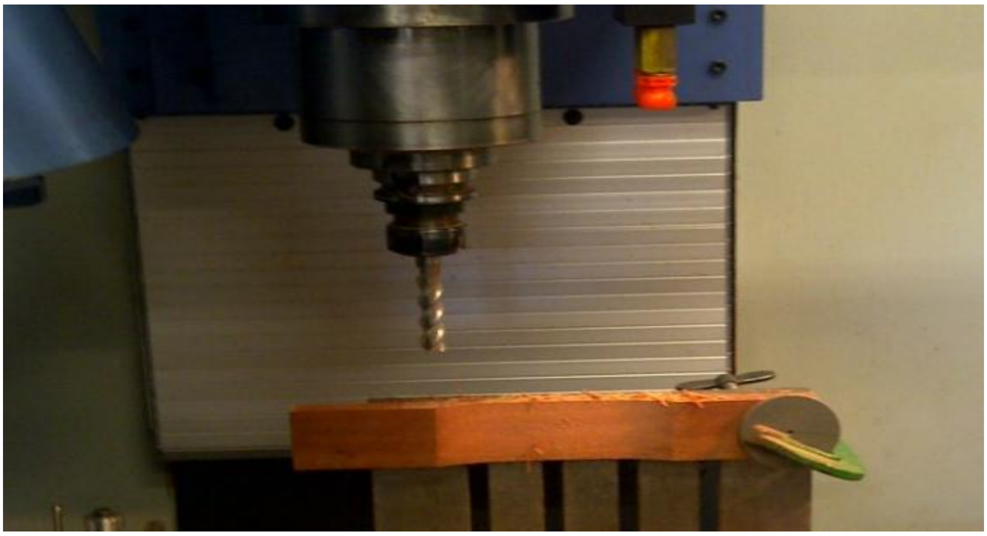


Figure 8.9 A complete marimba beam manufactured on the Leadwell V30 CNC machine

The mid-section dimension of this manufactured marimba bar was measured and found to have an error of +0.12 mm. The predicted frequencies were re-calculated based on this actual dimension. The predicted natural frequencies became 222.67 Hz, 882 Hz and 2149 Hz. The measured frequencies are recorded in Table 7, and it shows that the errors are 5.33Hz, 6Hz and 90Hz for the lowest three natural bending frequencies. Note that, apart from re-prediction based on the actual $a_3$ dimension (d at centre location), these frequencies have not undergone “fine tuning”.

| Modes | Target frequencies (Hz) | Experimental frequencies (Hz) | Error (Hz) |
|---|---|---|---|
| f1 | 222.67 | 228 | 5.33 |
| f2 | 882 | 888 | 6 |
| f3 | 2149 | 2059 | -90 |

Table 7: Underside frequency results for wooden beam

Comparing the above results with the ones obtained by Entwistle and McGrechan (2007) in Table 8 and Table 9, it can be seen that the results which were predicted by that underside shape predictive model were very close to the measured results (when accounting for the error that was caused by manufacture), as shown by the results in the aluminum test in Table 9. On the other hand, for the wooden beam, both tests (in Tables 7 and 8) show that the non-homogeneity and anisotropy of wood affect the accuracy of the shape prediction. Therefore, some correction or fine tuning is required to improve the marimba tuning.

| Modes (Jarrah) | Target frequencies (Hz) | Experimental frequencies (Hz) | Error (Hz) |
|---|---|---|---|
| f1 | 220 | 220.8 | 0.8 |
| f2 | 880 | 852 | -28 |
| f3 | 2156 | 2096 | -60 |

Table 8: Underside shape prediction results for wooden beam (Entwistle and McGrechan, 2007)

| Modes (Aluminium) | Target frequencies (Hz) | Experimental frequencies (Hz) | Error (Hz) |
|---|---|---|---|
| f1 | 220 | 222.3 | 2.3 |
| f2 | 880 | 885.3 | 5.3 |
| f3 | 2156 | 2164.3 | 8 |

Table 9: Underside shape prediction results for aluminum beam (Entwistle and McGrechan, 2007)

### 8.2 Fine Tuning Program

The results of the underside shape predictive model that were obtained in Section 8.1 have shown that the non-homogeneity and anisotropy of wood affect the accuracy of the shape prediction. Thus, the fine tuning process (described in Chapter 6) is necessary for improving marimba bar tuning, and Figure 8.10 shows the procedure of this fine tuning process.

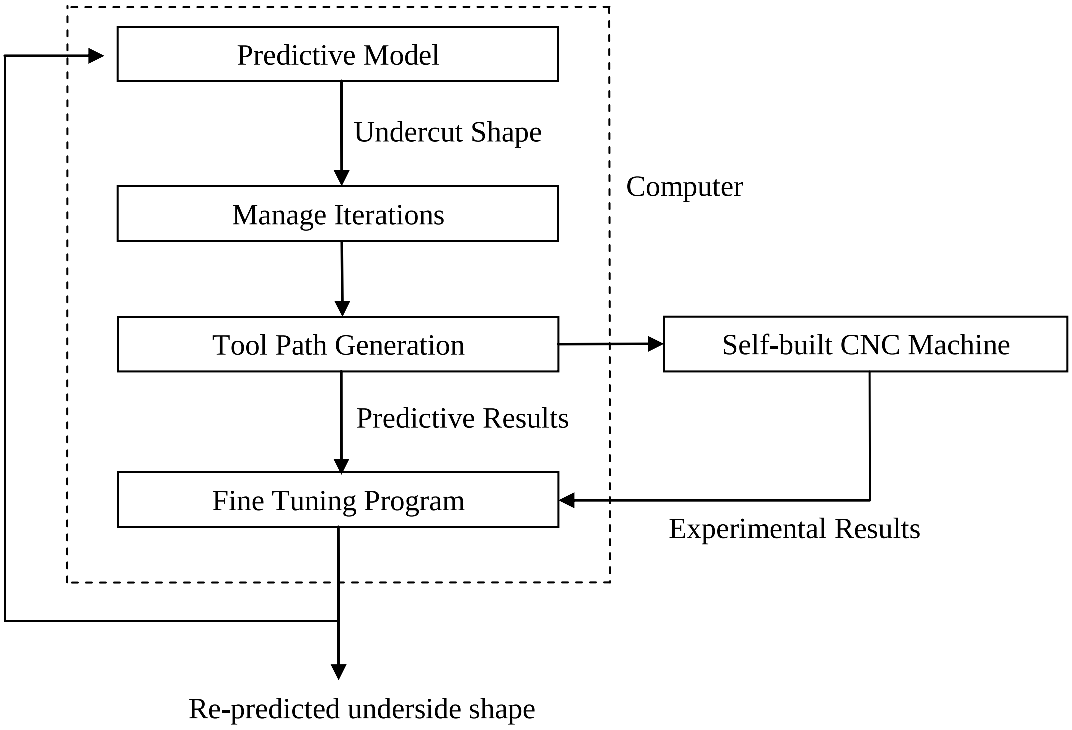


Figure 8.10 The fine tuning procedure

As shown in Figure 8.10 the underside shape predictive model firstly generates the predicted dimensions of the marimba bar (assuming the tested wood is homogeneous and isotropic), and the fine tuning program generates the tool path coordinates for each iteration. The wood blank is clamped to the CNC machine table, and once the software receives the signal that the wood has been placed on the machine from the user it will turn the spindle on to manufacture the wood blank. The required incremental target frequencies and experimental natural frequencies for each iteration are stored (described in Chapter 6). The underside shape of this marimba bar is re-determined for the corrected prediction, which gives a closer result to the target frequencies.

The physical machining centre, which combines a CNC machine tool, the underside shape predictive model, the fine tuning program and the frequency measuring program, was used in this test for the goal of tuning automation. The software completes the shape prediction and manages the fine tuning process based on the predicted undercut geometry, and then controls the CNC machine tool during manufacture.

The CNC machining tool (SIEG X2) that was selected for this task used a 10 mm ball nose slot drill as its cutter. Due to the motor's power limitation it cannot be used in the same way as Section 8.1. Thus, the wood blank was placed horizontally on the CNC machine table as shown in Figure 8.11 so that the cutter's tip faces the underside surface of the wood blank.

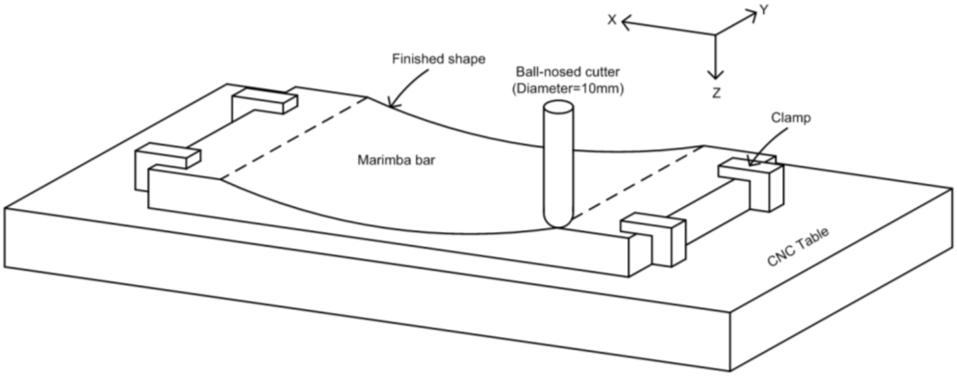


Figure 8.11 The clamping setup for the self-built CNC machine

The origin of the coordinate system was set at the edge of the predicted underside shape as shown in Figure 8.12. The tool path generation and the CNC machine controlling were designed based on this coordinate system. To achieve the accurate coordinates of the cutting origin, measurements were carefully performed by hand. Firstly, the X and Y coordinates of the origin were measured using a pencil with a sharp tip on it, and by manually moving this pencil to the first cutting point on the predicted underside shape the X and Y coordinates of the origin could be established.

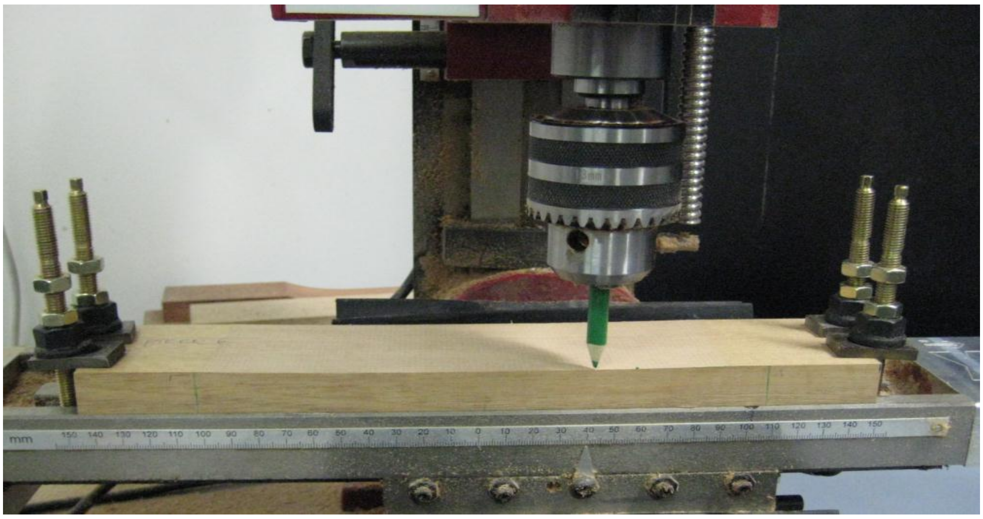


Figure 8.12 Measurements of the X and Y origin

For the coordinate of the original point on Z axis, papers were placed on the CNC table surface as shown in Figure 8.13. The spindle was switched on and slowly moved towards to the papers. The coordinate was recorded when the paper was just removed by the cutter, and the uncut papers were measured with a micrometre and added back to the coordinate.

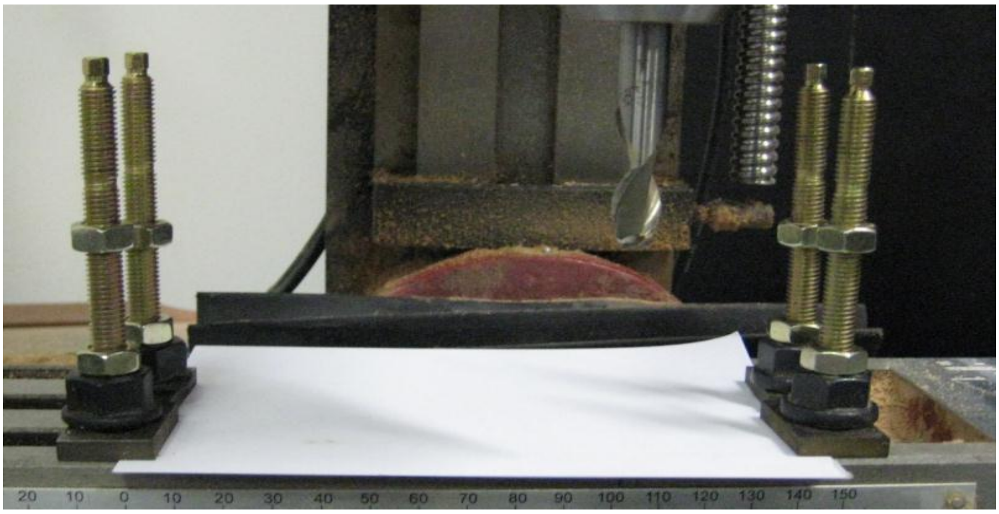


Figure 8.13 Measurement of Z origin

The mechanical properties of the test wood blank, including the dimensions, mass and the elastic properties, were measured. For example, the Young's modulus and shear modulus were inferred from measuring the natural bending and torsional frequencies of the prismatic blank as shown Figure 8.14. (Details are described in Section 5.2)

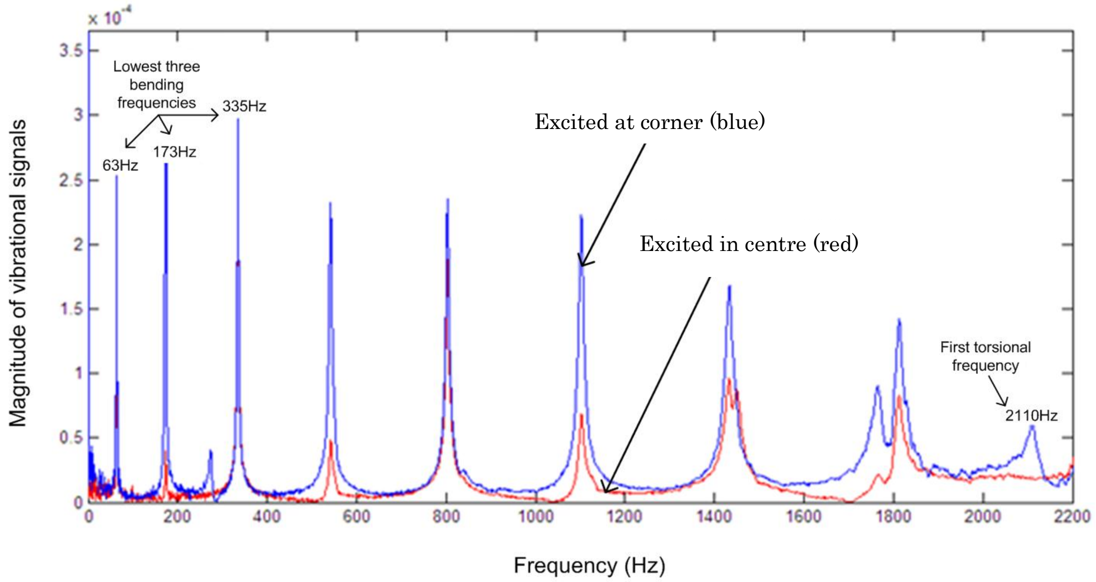


Figure 8.14 Measured bending and torsional natural frequencies of the blank marimba bar

The properties of the tested wood are provided in the table below.

| Properties | Jarrah (19/07/2011) |
|---|---:|
| m (kg) | 1.157 |
| L (m) | 1.203 |
| b (m) | 0.06 |
| do(m) | 0.01915 |
| A =bdo(m$^2$) | 0.00115 |
| V=Lbdo (m$^3$) | 0.00138 |
| ρ (kg/m$^3$) | 837.187 |
| I (m$^4$) | 3.51E-08 |
| $\beta_1 l$ (See equation B.1) | 4.73 |
| $\beta_2 l$ (See equation B.1) | 7.853 |
| $\beta_3 l$ (See equation B.1) | 10.996 |
| $f_1$ (Hz) | 63 |
| $f_2$ (Hz) | 173 |
| $f_3$ (Hz) | 335 |
| $E_1$ (GPa) | 17.96 |
| $f_{t1}$ (Hz) | 2110 |
| G (GPa) | 6.75 |

Table 10: Inputs of the fine tuning Test

Using these properties and setting the arbitrary target frequencies to 300Hz, 1200Hz and 2940Hz, the manufacturing process commenced.

| First Cut | Incremental targets | Experiment | Repredicted targets | Error |
|---|---|---|---|---|
| f1 (Hz) | 884.9 | 840 | 885.3 | -45.3 |
| f2 (Hz) | 2536.8 | 2201 | 2537.6 | -336.6 |
| f3 (Hz) | 4901.6 | 4190 | 4902.5 | -712.5 |
| do(mm) | 16.89 | 16.897 | 0.007 | |
| Second Cut | Incremental targets | Experiment | Repredicted targets | Error |
| f1 (Hz) | 767.6 | 730 | 769.6 | -39.6 |
| f2 (Hz) | 2363.5 | 2081 | 2367.7 | -286.7 |
| f3 (Hz) | 4717.6 | 4049 | 4722.3 | -673.3 |
| do(mm) | 14.64 | 14.674 | 0.034 | |
| Third Cut | Incremental targets | Experiment | Repredicted targets | Error |
| f1 (Hz) | 647.1 | 617 | 648.6 | -31.6 |
| f2 (Hz) | 2154.1 | 1921 | 2157.5 | -236.5 |
| f3 (Hz) | 4506.3 | 3873 | 4510.3 | -637.3 |
| do(mm) | 12.38 | 12.405 | 0.025 | |
| Fourth Cut | Incremental targets | Experiment | Repredicted targets | Error |
| f1 (Hz) | 524.2 | 499 | 526.1 | -27.1 |
| f2 (Hz) | 1902 | 1702 | 1907 | -205 |
| f3 (Hz) | 4249.8 | 3639 | 4256.1 | -617.1 |
| do(mm) | 10.12 | 10.152 | 0.032 | |
| Fifth Cut | Incremental targets | Experiment | Repredicted targets | Error |
| f1 (Hz) | 400 | 380 | 400.2 | -20.2 |
| f2 (Hz) | 1600 | 1435 | 1600.7 | -165.7 |
| f3 (Hz) | 3920 | 3349 | 3921 | -572 |
| do(mm) | 7.868 | 7.872 | 0.004 | |

Table 11: Detailed fine tuning results

The obtained incremental target frequencies and the measured natural frequencies are summarized below:

| | First Cut | Second Cut | Third Cut | Fourth Cut | Fifth Cut |
|---|---|---|---|---|---|
| Incremental target f1 (Hz) | 885.3 | 769.6 | 648.6 | 526.1 | 400.2 |
| Experimental f1 (Hz) | 840 | 730 | 617 | 499 | 380 |
| Incremental target f2 (Hz) | 2537.6 | 2367.7 | 2157.5 | 1907 | 1600.7 |
| Experimental f2 (Hz) | 2201 | 2081 | 1921 | 1702 | 1435 |
| Incremental target f3 (Hz) | 4902.5 | 4722.3 | 4510.3 | 4256.1 | 3921 |
| Experimental f3 (Hz) | 4190 | 4049 | 3873 | 3639 | 3349 |

Table 12: Summarized incremental target frequencies and the experimental natural frequencies for the fine tuning process

The relationships between the incremental target frequencies and the experimental frequencies are represented by lines as shown from Figure 8.15 to Figure 8.17.

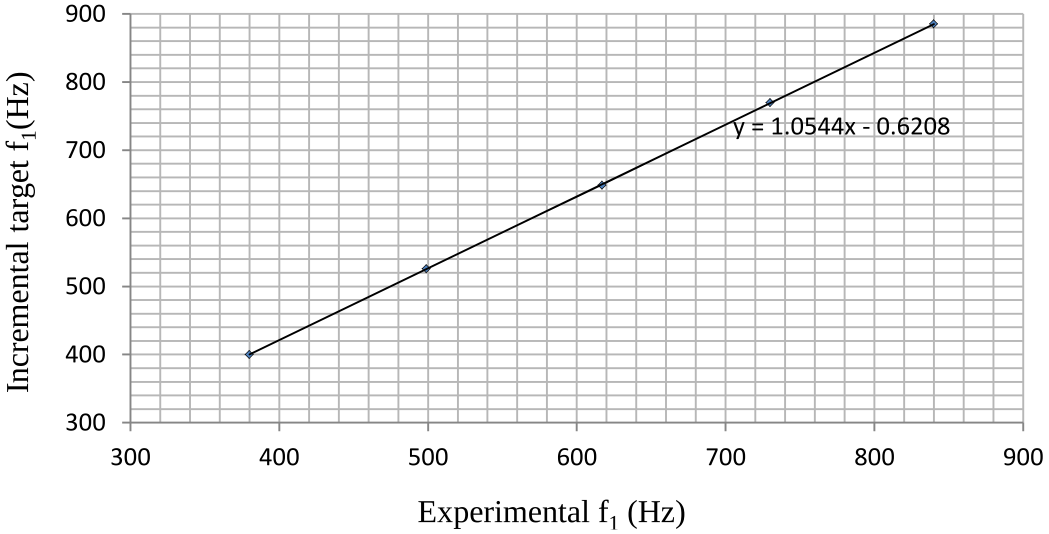


Figure 8.15 Incremental target $f_1$ vs. Experimental $f_1$

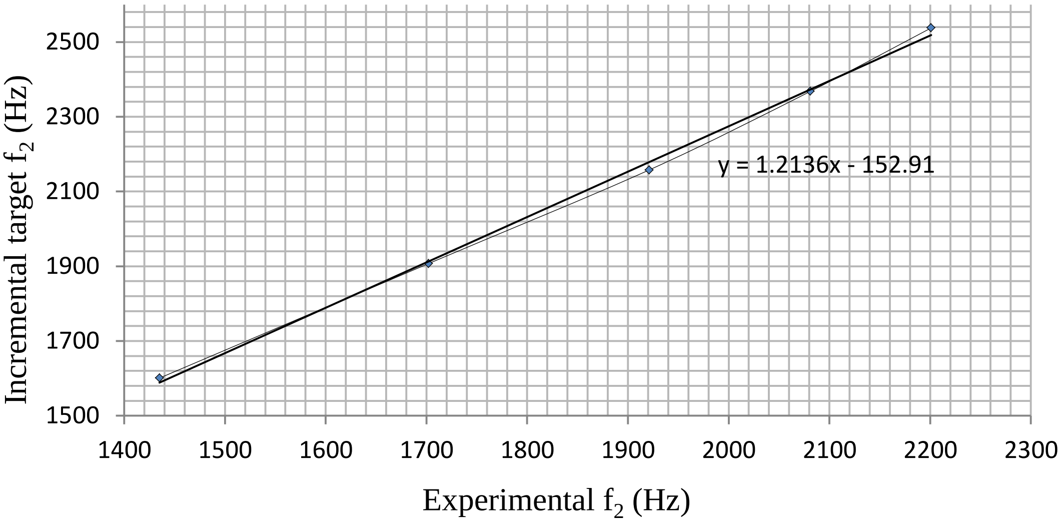


Figure 8.16 Incremental target $f_2$ vs. Experimental $f_2$

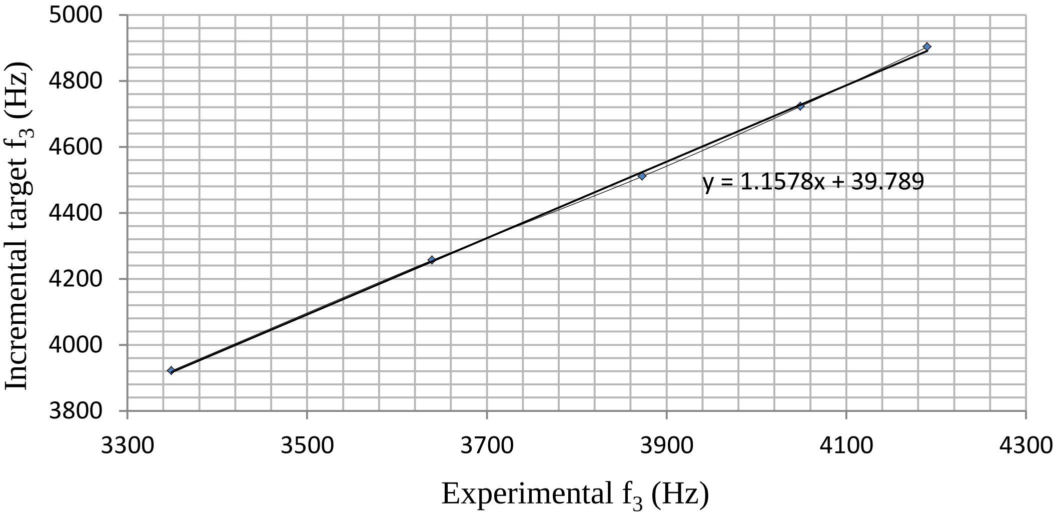


Figure 8.17 Incremental target $f_3$ vs. Experimental $f_3$

The fitted relationships for the lowest three natural bending frequencies were:

$$\mathrm{f1}_{\mathrm{predicted}} = 1.0544\mathrm{f1}_{\mathrm{measured}} - 0.6208 \tag{8.1a}$$

$$\mathrm{f2}_{\mathrm{predicted}} = 1.2173\mathrm{f2}_{\mathrm{measured}} - 154.62 \tag{8.1b}$$

$$\mathrm{f3}_{\mathrm{predicted}} = 1.1598\mathrm{f3}_{\mathrm{measured}} + 35.482 \tag{8.1c}$$

The user-desired (target) natural frequencies were set to be 300Hz, 1200Hz and 2940Hz. If these values are considered as the measured frequencies of the final cut, the re-predicted natural frequencies can be calculated using equations from 8.1a to 8.1c.

$$\text{Repredicted natural frequencies} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 315.6992 \\ 1306.140 \\ 3445.294 \end{bmatrix} \tag{8.2}$$

The differences between the original target natural frequencies and the re-predicted target natural frequencies can be calculated as:

$$\begin{bmatrix} \Delta \mathrm{f}_1 \\ \Delta \mathrm{f}_2 \\ \Delta \mathrm{f}_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} - \begin{bmatrix} f_{1\_old} \\ f_{2\_old} \\ f_{3\_old} \end{bmatrix} = \begin{bmatrix} +15.6992 \\ +106.14 \\ +505.294 \end{bmatrix} \tag{8.3}$$

The required shape matrix (equation 6.2) used in the fine tuning process of this test were determined as

$$\text{shape matrix} = \begin{bmatrix} 1.4906 & 37.3413 & 52494 \\ 15.7444 & 314.3455 & 166040 \\ 63.2721 & 997.2794 & 318740 \end{bmatrix} \tag{8.4}$$

Substituting equations 8.3 and 8.4 into equation 6.4, the change of the shape were calculated as

$$\begin{bmatrix} \Delta a_1 \\ \Delta a_2 \\ \Delta a_3 \end{bmatrix} = \begin{bmatrix} -0.2344 \\ +0.01881 \\ +0.0000419 \end{bmatrix} \tag{8.5}$$

The original prediction of the underside shape has three coefficients ($a_1$, $a_2$ and $a_3$), presenting the shape for the target natural frequencies. As the fine tuning program has adjusted the target natural frequencies for the final cut (equation 8.3), the underside shape of the final cut was adjusted as well (equation 8.5), which gives the changed shape coefficients:

$$\text{new shape coefficients} = \begin{bmatrix} a_{1\_new} \\ a_{2\_new} \\ a_{3\_new} \end{bmatrix} = \begin{bmatrix} 31.97 \\ -0.48779 \\ 0.0065419 \end{bmatrix} \tag{8.6}$$

The new generated shape coefficients give a new predicted underside shape as shown in Figure 8.18. It can be seen that the re-predicted underside shape was slightly different from the original prediction. By using the new prediction it was expected that the measured frequencies would be close enough to the target natural frequencies.

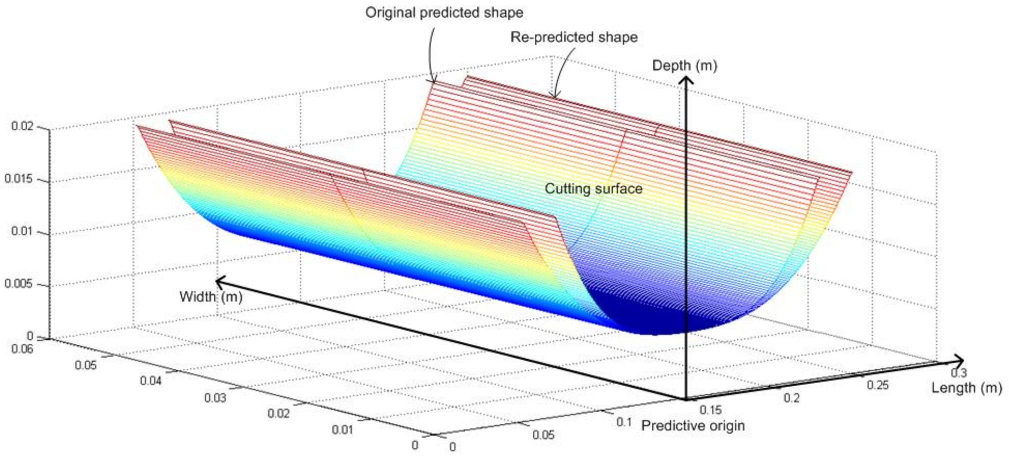


Figure 8.18 Comparison between the original prediction and the re-prediction

The final natural frequency results are shown in Figure 8.19.

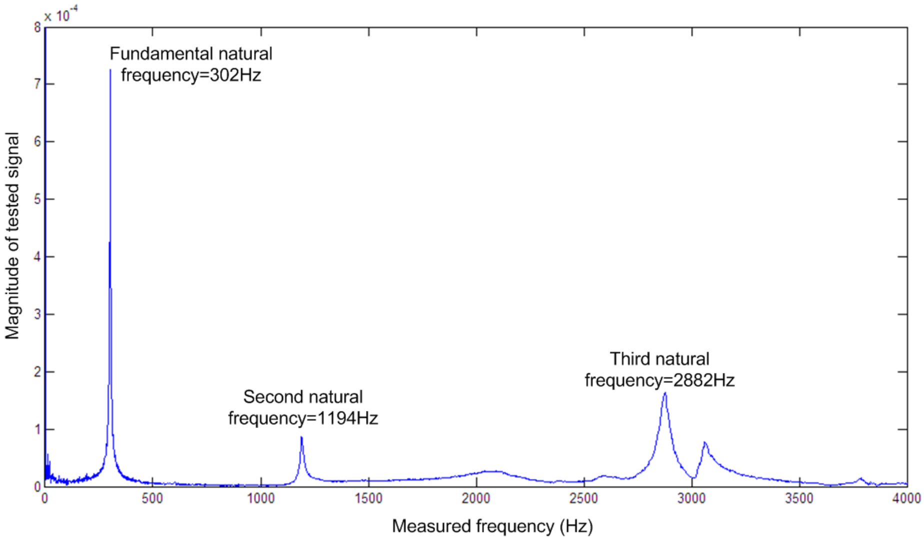


Figure 8.19 Measured natural bending frequencies of manufactured marimba beam

The target natural frequencies were 300 Hz, 1200 Hz and 2940 Hz for the lowest three bending frequencies. The manufactured marimba bar gives 302 Hz, 1194 Hz and 2882 Hz. However, the obtained marimba bar had +0.025 mm manufacture error in thickness, which affected the accuracy of tuning. To cancel out the manufacture errors that were caused by the inaccurate machining, this thickness error (0.025 mm) was returned back to the program to determine the predicted frequencies that are produced by the geometry of this manufactured marimba bar. Combining the re-predicted target frequencies (301.329Hz, 1202.103Hz and 2939.75Hz) and the measured frequencies, it gives 0.671 Hz, 8.103 Hz and 57.75 Hz being the error of tuning.

| Modes | Re-predicted Targets (Hz) | Experimental frequencies (Hz) | Error (Hz) |
|---|---|---|---|
| f1 | 301.329 | 302 | 0.671 |
| f2 | 1202.103 | 1194 | -8.103 |
| f3 | 2939.75 | 2882 | -57.75 |

Table 13: Results of the fine tuning test

Similar tests were performed using Tasmanian oak, and they are presented in Table 14. (Details are recorded in Appendix F.)

| Modes (Oak 1) | Targets (Hz) | Experimental f (Hz) | Re-predicted Targets (Hz) | Error (Hz) |
|---|---|---|---|---|
| f1 | 380 | 372 | 373.12 | -1.12 |
| f2 | 1520 | 1496 | 1500.55 | -4.55 |
| f3 | 3724 | 3654 | 3700 | -46 |
| Modes (Oak 2) | Targets (Hz) | Experimental f (Hz) | Re-predicted Targets (Hz) | Error (Hz) |
| f1 | 300 | 301 | 300.5 | 0.5 |
| f2 | 1200 | 1216 | 1204 | 12 |
| f3 | 2940 | 3020 | 2966 | 54 |

Table 14: Fine tuning results obtained from other tests

It can be seen that the resulting natural frequencies in Tables 13 and 14 are closer to the targets than the results in Tables from 7 to 9, especially the second and third natural frequencies. The improvements are due to the application of the fine tuning technique that counteracts the unknown wood characteristics.

# Chapter 9: Discussion

In this chapter, the sources of error are discussed in Section 9.1. The application of the present work is presented in Section 9.2.

## 9.1 Sources of Error

The results of the predictive model and the fine tuning program have shown the improvements made by this automatic multi-modal tuning system (described in Chapter 8). However, inaccuracies due to many reasons still cannot be avoided. This section explores the possible sources of error in this thesis, such as the uncertainties of the material used in tests, the limitations of the fine tuning program, the hardware limitations and some possible human errors. The following three sections explain these in detail.

### 9.1.1 Errors from the Predictive Model

The surfaces of wood blanks were expected to be perfectly flat, and therefore sand paper was used to smooth the surface of the wood blank as machining the wood blanks resulted in rough surfaces. However, a perfect smooth surface was hard to achieve as 0.05mm manufacture error in the thickness still could not be avoided. This affected the accuracy of the prediction as the surfaces of the wood blanks were assumed to be perfectly flat. For example, the density of the wood blank was calculated based on the rectangular shape, the calculation might have error if the actual dimensions are different (same for the calculation of the elastic properties). What is more, as the wood is a complicated material, some regions of the wood blank may have higher densities than other regions.

The elastic moduli, including the Young’s modulus and the shear modulus, were determined from the measured natural frequencies of a blank. To make the modeling simpler it was firstly assumed that the test material (wood) was homogeneous and isotropic. As the test wood blank possesses mechanical resonant frequencies that are determined by the elastic modulus, dimensions and mass of the tested wood, the elastic modulus can be calculated when the mechanical resonant frequencies, mass and dimensions were measured. However, wood is a non-homogeneous and anisotropic material, there are three Young’s moduli and three shear moduli in wood, and some elastic properties of wood cannot be measured using this method. Although the density of wood, the Young's modulus ($E_1$) and the shear modulus ($G_{12}$) along the grain are more important for the resulting natural frequencies, the lack of measurements of the rest of the elastic properties may still affect the model's accuracy.

### 9.1.2 Errors from the Fine Tuning Program and Measurements

The underside shape was re-predicted based on the discovered "relationship linear equations", and the underside shape was then modified when the shape coefficients varied. The relationship between the resulting natural frequencies and the underside shape coefficients gave the shape matrix as explained in Section 6.2. The value was determined by the slope of the lines, found by the curve-fitting method. While the curves were approximately linear, a higher order equation may have produced better predictions.

The major task of the fine tuning process was to generate "relationship linear equations" which could best describe the relationship between the theoretical prediction and the actual experiments. As finite cutting depth increments were required, the accuracy of this fitted relationship was limited. The accuracy of these linear equations also depends on the nature of the wood. If the mechanical properties of the wood piece change rapidly along the length of the test piece, the accuracy of the linear equation would be limited. (Mostly the wood has normal conditions as over 30 tests were completed in this thesis, and only two tests failed because of this problem.)

### 9.1.3 Errors from Manufacture

The commercial Leadwell V30 CNC machine has an average accuracy of (up to) 0.05 mm. The self-built "SIEG X2" CNC machine has an average accuracy of 0.04 – 0.07 mm, but could achieve the best accuracy of up to 4 μm. The coordinate setup of the original position on the CNC machine is one of the major sources of error during manufacture. For both the underside shape predictive model test using the commercial Leadwell V30 CNC machine in Section 8.1 and the fine tuning program test using the self-built CNC machine in Section 8.2, the coordinates of the cutting origin were manually determined by the user, and it has been shown that the coordinates of the cutting origin were not perfectly accurate. This affected the accuracy of the machining. (The sensitivity of, for example, the fundamental frequency on the central bar thickness was 64 Hz/mm.)

Furthermore, when using the self-built CNC machine, the linear travelling distance per motor step was calculated using equations 7.1 and 7.2 based on the information provided by the manufacturers of the components, such as the type of lead screw and the type of the stepper motor used. The specifications of these components were directly used in calculations to generate the tool path coordinates, which were further converted to the CNC readable signals to drive the stepper motors on the machine. However, the results in Section 7.1 have shown that the linear travelling distance per motor step was not accurate, which affected the accuracy of the wood manufacture. An error of less than 0.01 mm was achieved for the required linear distance of 300 mm. This error could still cause 0.6 Hz error on the fundamental natural frequency, 1.9 Hz on the second natural frequency and 3.6 Hz on the third natural frequency.

Because a ball-nosed cutter was used, undulations were left between two cutting lines as shown in Figure 7.6, which left extra material that ideally should have been removed. This affected the results as marimba bars are very sensitive to such material, and any material left on the underside shape of it could affect the accuracy of tuning. To improve this, the setting in the program has been modified so that the gaps between cutting paths were reduced and limited to 0.2mm, and the surface of the underside of the marimba bar would then be smoothed with sand paper to ensure that the shape was not affected.

## 9.2 Application

The work presented in this thesis shows that three major problems of tuning wooden marimba bars have been solved or improved; the underside shape prediction of marimba beams using the predictive Timoshenko-receptance model, automatic tuning for correcting the nature of non-homogeneity and anisotropy of wood and the manufacturing automation of marimba beams. It has shown that the techniques of automatic tuning of marimba bars can largely reduce the time and costs in the industrial volume production as the major processes can be performed by a system such as described. The final accurate fine tuning should still be performed by experienced craftsmen due to the accuracy limitation of this system. Besides the direct application of this technique, the methods and algorithm which were developed in this thesis can also be used in other areas.

Firstly, the predictive (Timoshenko-receptance) model can be used in the prediction of the dynamic behaviour of a complicated mechanical structure as well as the determination of the natural frequencies and mode shapes of such structures. The sub-structuring (receptance) method can be used to analyze the vibration characteristics of complicated structures as the original (uniform or non-uniform) structure can be sub-divided into a number of small components, which are considered as simple prismatic beam-like structures. The small components, which are broken up from the original structure, can be small enough so that the model's accuracy will be adequate. An actual example (Schmitz and Burns, 2003) of this receptance model in other applications was to predict the high-speed machining dynamics, where the tool-point frequency response for high-speed machining was predicted, and the receptance model was then used to predict the changes in the tool-point receptance for the setup variations such as tool length. The resulting tool-point frequency response was then used to determine the stable cutting zones as a function of chip width and spindle speed so that the appropriate machining parameters could be selected.

Secondly, the fine tuning program, which combined the predictive and experimental results in the marimba bar tuning process, was developed to correct the error, which is caused by the complicated nature (non-homogeneity and anisotropy) of wood. During the process, three components are involved: the elastic moduli (Young's modulus and shear modulus), the predicted natural frequencies and the experimental natural frequencies. If the elastic modulus is used as the fixed parameter, the relationships between the predicted natural frequencies and the experimental natural frequencies can be determined. Because wood is non-homogeneous and anisotropic, the highly accurate "relationship linear equations" between the predicted and experimental frequencies was difficult to achieve. However, if the material is homogeneous for other applications, the prediction of the natural frequencies will be more accurate. The fine tuning program can then be modified to determine the elastic modulus for homogeneous materials. When the elastic modulus is set as the varying parameter, the frequency prediction of the homogeneous material will be accurate. Therefore, changing the elastic modulus to match the experimental results and the prediction will give accurate elastic modulus. (Start the iterative process using the elastic values assuming the material is isotropic)

Thirdly, the goal of automatic tuning marimba bars has been achieved in this thesis as the underside shape predictive model, fine tuning program, frequency measuring program and the control of physical machining centre, including the stepper motors and the spindle of CNC machine, were coded in the same software (Matlab). Some of the techniques that were used in the above tasks were only developed in this thesis. For example, the stepper motors and spindle of the CNC machine are controlled directly by Matlab, which allows users to not only calculate and simulate an engineering problem, but also to control an external device from Matlab. What is more, one of the major roles that Matlab played in this thesis was to act as a CAM software as the tool path was generated once the dimensions of the correct geometry is available, and the CNC readable G-code were generated by Matlab as shown in Section 7.2. Therefore, it might be possible to directly link Matlab with a commercial CNC machine in the future.

This page intentionally left blank.

# Chapter 10: Conclusions and Future Work

This thesis has developed an automatic multi-modal tuning system for tuning marimba bars. It combined the underside shape predictive model, the fine tuning program, the hardware controlling program, the frequency measuring program and a CNC machine tool to automatically tune marimba bars and correct the unavoidable uncertainties of wood. The work presented in this thesis showed that three major problems of tuning wooden marimba bars have been solved: the underside shape prediction of marimba bars using the underside shape predictive model; the fine tuning process of correcting the nature of non-homogeneity and anisotropy of wood; and the fine tuning automation. The lowest three natural bending frequencies of marimba bars (as the tuning results) were simultaneously tuned to the target ratios, and it has shown the improvements achieved by the fine tuning process.

This automatic multi-modal tuning system is a combination of many existing techniques to solve the tuning problem of marimba bars. It provides convenience to users during marimba bar manufacture and has made accurate tuning automation possible for the future. However, the error caused by the complicated natural of wood still cannot be completely avoided. Therefore, the improvements to the current work can be made by making the underside shape predictive model suitable for wood (or for one particular type of wood). This may be achieved by adding an “input measuring program” to the current tuning system so that all elastic properties of wood can be taken into account. Some modifications in the current programs will be required. What is more, the accuracy of marimba bar manufacture can be improved as a small error in the size of beam will result a relatively large error in the resulting natural bending frequencies. A better milling machine may be chosen for this task.

This page intentionally left blank.

## Chapter 11: References

ASTM International. 2008. *Standard Test Method for Dynamic Young's Mudulus, Shear Modulus, and Poisson's Ratio by Sonic Resonance*. PA 19428-2959.

Ballarini, R. 2003. *The Da Vinci-Euler-Bernoulli Beam Theory*. http://www.memagazine.org/contents/current/webonly/webex418.html (accessed September 16, 2009)

Bishop, R., and D. Johnson. 1960. *The Mechanics of Vibration*. London: Cambridge University Press. pp. 282 - 284

Bork, I. 1994. Practical Tuning of Xylophone Bars and Resonators. *Applied Acoustics* 46 (1):103-127. http://www.sciencedirect.com (accessed October 19, 2009)

Bretos, J., C. Santamanria, and J. Moral. 1997. Tuning process of xylophone and marimba bars analyzed by finite element modeling and experimental measurements. *Journal of the Acoustical Society of America* 102 (6): 3815-3816. http://www.sciencedirect.com (accessed October 29, 2009)

Bretos, J., C. Santamanria, and J. Moral. 1998. Finite element analysis and experimental measurements of natural eigenmodes and random responses of wooden bars used in musical instruments. *Applied Acoustics* 56 (3): 141-156. http://www.sciencedirect.com (accessed October 20, 2009)

Bustamante, F. 1991. Nonuniform beams with harmonically related overtones for use in percussion instruments. *Journal of the Acoustical Society of America* 90 (6): 2935-2941. http://adsabs.harvard.edu (accessed October 20, 2009)

Choi, Y. K. 2005. *Tool path generation and 3D tolerance analysis for free-form surfaces*. PhD Thesis, Texas A&M University, Texas, USA

Entwistle, R.D. and M$^{c}$Grechan, S.R. 2007. *Geometric Shape Identification for Multi-Mode Tuning of Percussion Instrument Bars*. *Proceedings of the 14th*

*International Congress on Sound and Vibration.* Cairns, Australia, 9-12 July 2007. ISBN 978 0 7334 2516 5. CD-ROM

Henrique, L. and J. Antunes. 2003. Optimal Design and Physical Modelling of Mallet Percussion Instruments. *Act Acustica* 89 (6): 948-963.

Hutchinson, J. R. 2001. Shear coefficients for Timoshenko beam theory. *Journal of Applied Mechanics*. 68 (1): 87-92.

Inman, D. J. 2008. *Engineering Vibration Third Edition*. New Jersey, USA. Pp. 494-497.

McGrechan, S.R. 2006. *Percussive Beam Tuning Using Receptance Modelling Techniques*. B.Eng. Project Report, Department of Mechanical Engineering, Curtin University of Technology, Perth, Western Australia.

Petrolito, J. and K. Legge. 1997. Optimal undercuts for the tuning of percussive beams. *Journal of the Acoustical Society of America* 102 (4): 2432-2437.

Petrolito, J. and K. Legge. 2004. Designing musical structures using a constrained optimization approach. *Journal of the Acoustical Society of America* 117 (1): 384-390.

Petrolito, J. and K. Legge. 2007. Designing idiophones with tuned overtones. *Acoustics Australia* 35 (2): 47-50.

Schmitz, T. and T. Burns. 2003. *Receptance coupling for high speed machining dynamics prediction.* http://sem-proceedings.com/21i/sem.org-IMAC-XXI-Conf-s28p06-Receptance-Coupling-High-Speed-Machining-Dynamics-Prediction.pdf (accessed May 18, 2011)

Smith, M.D. 1998. *Newton-Raphson Method*, http://web.mit.edu/10.001/Web/Course_Notes/NLAE/node6.html (accessed September 26, 2009)

Stone, B. 1992. The receptances of beams, in closed form, including the effects of shear and rotary inertia. *Journal of Mechanical Engineering Science* 206 (2): 87-94.

Suits, B. H. 2001. Basic physics of xylophone and marimba bars. *American Journal of Physics* 69 (7): 743

Witmer, A. (1992). *Elementary Euler-Bernoulli Beam Theory*. MIT Unified Engineering Course Notes. pp. 5-115 to 5-164

This page intentionally left blank.

# Appendix A: Mathematical Models

## Appendix A.1 Euler-Bernoulli Model[15]

The Euler-Bernoulli theory is a simplification of the linear isotropic theory of elasticity, it provides deflection characteristics of prismatic beams. (Witmer, 1992) The following figure is a model of a transverse vibrating beam element, which has a length "$\delta x$" and are bounded by plane faces labelled as "1" and "2". The shearing forces and bending moments of this beam element are labeled as "$F_1$", "$F_2$", "$M_1$" and "$M_2$".

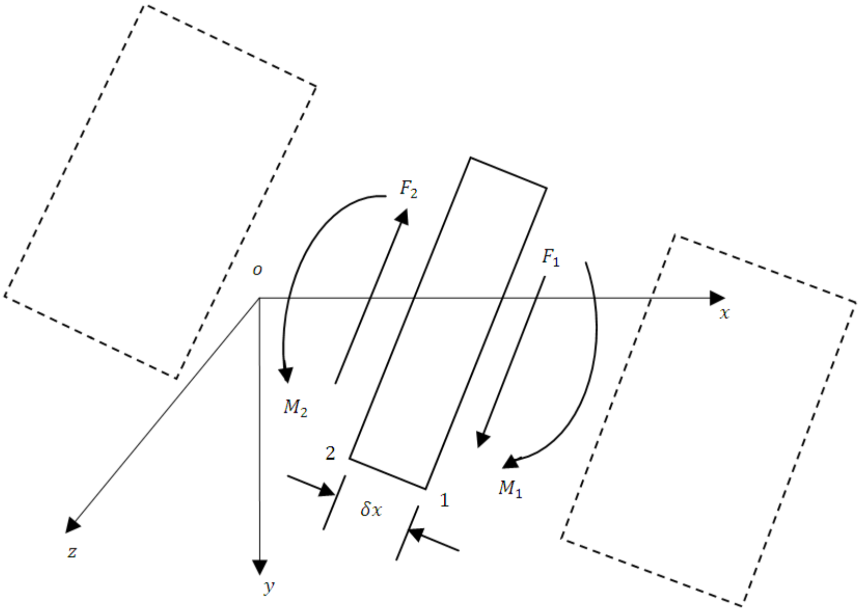


Figure A.1: A simple Euler-Bernoulli beam in transverse vibration (Bishop and Johnson 1960, 283)

It is assumed that the prismatic beam as shown in Figure A.1 vibrates freely and the variation of the shearing forces and bending moment across this element dimension "$\delta x$" is linear. Thus, by denoting the shearing force "$F_2$" and bending moment "$M_1$" as "F" and "M", "$F_1$" and "$M_2$" become "$F + \frac{\partial F}{\partial x} dx$" and "$M + \frac{\partial M}{\partial x} dx$". Hence,

$$\sum F = F_1 - F_2 = \left(F + \frac{\partial F}{\partial x} dx\right) - F = \frac{\partial F}{\partial x} dx \tag{A.1.1}$$

$$\sum M = (\sum F) dx + M_1 - M_2 = \left(\frac{\partial F}{\partial x} dx\right) dx + \frac{\partial M}{\partial x} dx \tag{A.1.2}$$

[15] Derivation based on Witmer (1992).

Applying the Newton's second law in the y direction

$$\sum F_y = dm\frac{\partial^2 y}{\partial t^2} = \rho A dx\frac{\partial^2 y}{\partial t^2} \qquad \text{(A.1.3)}$$

where "$y$" represents the displacement in the y direction, and hence combining equations A.1.1 and A.1.3 gives

$$\frac{\partial S}{\partial x}dx = \rho dx A\frac{\partial^2 y}{\partial t^2} \qquad \text{(A.1.4)}$$

It is again assumed in the Euler-Bernoulli theory that the plane faces remain plane during vibration, and the radius of the curvature of a bent beam is large compared with the beam's depth, such that the rotary inertia and shear deformation could be neglected. (Bishop and Johnson 1960, 285) Thus, the small deflection makes the equation A.1.2 as

$$\sum M = (\sum F)dx + M_1 - M_2 = \left(\frac{\partial F}{\partial x}dx\right)dx + \frac{\partial M}{\partial x}dx = 0 \qquad \text{(A.1.5)}$$

When the length of the element is made small, equation A.1.5 becomes

$$F = -\frac{\partial M}{\partial x} \qquad \text{(A.1.6)}$$

Combining equation A.1.4 and A.1.6 gives

$$\rho A\frac{\partial^2 y}{\partial t^2} = -\frac{\partial^2 M}{\partial x^2} \qquad \text{(A.1.7)}$$

The normal stress and moment-curvature formula for a linear elastic beam is given as

$$\frac{M}{I} = \frac{\sigma}{y} = \frac{E}{R} \qquad \text{(A.1.8)}$$

$$\frac{1}{R} = \frac{\left|\frac{d^2y}{dx^2}\right|}{\left(1+\left[\frac{dy}{dx}\right]^2\right)^{\frac{3}{2}}}$$

where “R” is the radius of curvature. (Ballarini, 2003) As stated in the assumption that the deflection is small, the slope of the deflected term “$\frac{dy}{dx}$” would be small and much smaller than 1. Thus, the radius of curvature has an approximation of

$$R^{-1} = \frac{d^2y}{dx^2} \tag{A.1.9}$$

Combining equations A.1.7 and A.1.8 gives

$$M = EI\frac{d^2y}{dx^2} \tag{A.1.10}$$

Combining equations A.1.6 and A.1.9 gives the final Euler-Bernoulli beam differential equation in equation A.1.11. (Inman 2008, 494)

$$\rho A\frac{\partial^2 y}{\partial t^2} + EI\frac{d^4y}{dx^4} = 0 \tag{A.1.11}$$

This equation was used to determine the Young’s modulus, as the input of the predictive model (described in Section 5.2 and Appendix B). It was also combined with the receptance sub-structuring method to produce the underside shape predictive model. The result of this model was shown and compared with other models in Section 5.1.

## Appendix A.2 Timoshenko Model[16]

It is safe to ignore the shear deformation in Euler-Bernoulli theory as long as the height and width of the beam are small compared with the length of the beam. However, if the beam becomes shorter, the effect of shear deformation becomes evident. (Inman 2008, 501) To solve the above problem, the Timoshenko model was then developed and the Timoshenko model has made an improvement over the Euler-Bernoulli theory, in which it incorporates both the effect of rotary inertia and shear deformation.

Consider a small element if a beam at some time t in its motion as shown in Figure A.2, it has some displacement $y$ from its equilibrium position and the slope of its external surface is made up of two components γ and ψ.

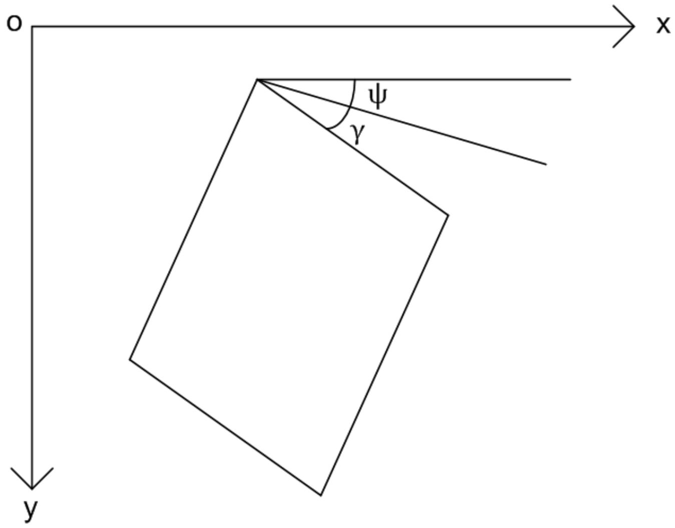


Figure A.2: Deflected position of an element of a uniform beam (Stone, 1992)

By definition, ψ is the slope resulting from bending and γ is the slope resulting from shear. Thus,

$$\frac{dy}{dx} = \psi + \gamma \tag{A.2.1}$$

[16] Derivation based on Inman (2008).

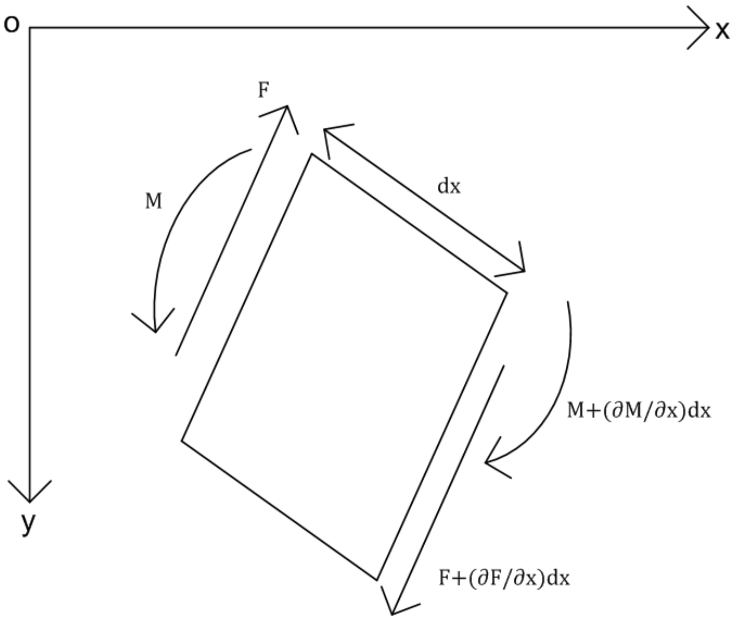


Figure A.3: Forces and moments acting on an element (Stone, 1992)

Considering the forces and moments acting on this element as shown in Figure A.3, the bending moment "M" is against the internal elastic inertia in the form below, which is obtained from the Euler-Bernoulli theory in equation A.1.10.

$$M = EJ\frac{d^2\omega}{dx^2} = EJ\frac{d\psi}{dx} \qquad \text{(A.2.1)}$$

The shearing force becomes

$$F = k^2AG\gamma = \ k^2AG\left[\frac{dy}{dx} - \psi\right] \qquad \text{(A.2.3)}$$

Compared to the Euler-Bernoulli theory, the moment of inertia used in the Timoshenko beam theory is the mass moment of inertia (I), which is a measure of an object's resistance to changes in its rotation rate, whereas the second moment of area (J) in the Euler-Bernoulli theory is a property of a shape to predict the resistance of beams to bending and deflection.

$$I = \rho J \qquad \text{(A.2.4)}$$

$$\sum M = \ I_{element}\frac{\partial^2\psi}{\partial t^2} = (\rho J dx)\frac{\partial^2\psi}{\partial t^2} \qquad \text{(A.2.5)}$$

Similar to the Euler-Bernoulli theory, the moment equation can be described as

$$\sum M = (\sum F)dx + M_1 - M_2 = (F)dx + \frac{\partial M}{\partial x}dx \tag{A.2.6}$$

Combining equation A.2.3, A.2.5 and A.2.6 gives

$$(\rho J)\frac{\partial^2\psi}{\partial t^2} = k^2AG\left[\frac{dy}{dx} - \psi\right] + \frac{\partial M}{\partial x} \tag{A.2.7}$$

Combining equation A.2.2 and A.2.7 gives

$$(\rho J)\frac{\partial^2\psi}{\partial t^2} = k^2AG\left[\frac{dy}{dx} - \psi\right] + EJ\frac{\partial^2\psi}{\partial x^2} \tag{A.2.8}$$

For transverse motion, the following equation shows an equilibrium relationship of translational force per unit length against the internal shear force gradient.

$$\sum F = F_1 - F_2 = \left(F + \frac{\partial F}{\partial x}dx\right) - F = \frac{\partial F}{\partial x}dx \tag{A.2.9}$$

Combining equation A.1.3 and A.2.9 gives

$$\rho A\frac{\partial^2 y}{\partial t^2} - k^2AG\left(\frac{\partial^2 y}{\partial x^2} - \frac{\partial\psi}{\partial x}\right) = 0 \tag{A.2.10}$$

It was assumed that the system is linear and the inertia effects resulting from the shear of the element were not included. The method that will be developed eliminates γ and working with ψ and *y*. when two beams are joined together, the values of ψ and *y* are the same at the interface and not the total slope, hence, it would be necessary to have the tip receptances in terms of ψ and *y* for adding beams. (Stone, 1992)

Consider equations A.2.8 and A.2.10. When differentiate equation A.2.8 with respect to x:

$$(\rho J)\frac{\partial^3\psi}{\partial x\partial t^2} = k^2AG(\frac{\partial^2 y}{\partial x^2} - \frac{\partial\psi}{\partial x}) + EJ\frac{\partial^3\psi}{\partial x^3} \tag{A.2.11}$$

Adding equations A.2.10 and A.2.11:

$$\rho A\frac{\partial^2 y}{\partial t^2}+EJ\frac{\partial^3\psi}{\partial x^3}-(\rho J)\frac{\partial^3\psi}{\partial x\partial t^2}=0 \tag{A.2.12}$$

From equation A.2.10,

$$\frac{\partial\psi}{\partial x}=-\frac{\rho}{k^2G}\frac{\partial^2 y}{\partial t^2}+\frac{\partial^2 y}{\partial x^2} \tag{A.2.13}$$

Therefore,

$$\frac{\partial^3\psi}{\partial x^3}=-\frac{\rho}{k^2G}\frac{\partial^4 y}{\partial x^2\partial t^2}+\frac{\partial^4 y}{\partial x^4} \tag{A.2.14}$$

$$\frac{\partial^3\psi}{\partial x\partial t^2}=-\frac{\rho}{k^2G}\frac{\partial^4 y}{\partial t^4}+\frac{\partial^4 y}{\partial x^2\partial t^2} \tag{A.2.15}$$

Substituting equations A.2.14 and A.2.15 into equation A.2.12 gives an equation in *y* independent of ψ:

$$EJ\frac{\partial^4 y}{\partial x^4}-\rho J\left(1+\frac{E}{k^2G}\right)\frac{\partial^4 y}{\partial x^2\partial t^2}+\frac{\rho^2 J}{k^2G}\frac{\partial^4 y}{\partial t^4}+\rho A\frac{\partial^2 y}{\partial t^2}=0 \tag{A.2.16}$$

If equation A.2.10 is differentiated with respect to x, and $\partial\psi/\partial x$ is obtained from equation A.2.8, using a procedure similar to that above, an equation in ψ independent of *y* could then be achieved. (Inman 2008, 501)

$$EJ\frac{\partial^4 \psi}{\partial x^4}-\rho J\left(1+\frac{E}{k^2G}\right)\frac{\partial^4 \psi}{\partial x^2\partial t^2}+\frac{\rho^2 J}{k^2G}\frac{\partial^4 \psi}{\partial t^4}+\rho A\frac{\partial^2 \psi}{\partial t^2}=0 \tag{A.2.17}$$

This solution of equation was combined with the receptance sub-structuring method to produce the underside shape predictive model (described in Section 4.2), which was used as the mathematical model for this thesis.

## Appendix A.3 Finite Element Model

The finite element method is a numerical technique for modelling and solving the problems of complicated mechanical structures from the motion of joints. It approximates a structure in two steps. The first step is to divide the mechanical model into many small elements with many connected joints. Each of these simple elements has an equation of motion, which can be easily solved, the solution of each element is joined together to form the "global mass matrix" and "global stiffness matrix", and this is the second step, which is used to express the structure as a whole. (Inman 2008, 575)

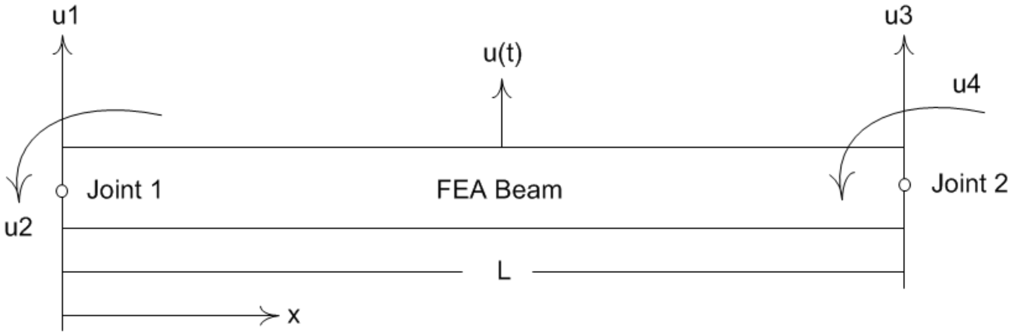


Figure A.4: A simple finite element model of a prismatic beam
(Inman 2008, 588)

Inman (2008, 588) introduced a finite element beam model as shown in Figure A.4, where two linear coordinates "$u_1$" and "$u_3$" and two rotational coordinates "$u_2$" and "$u_4$" are used to show the motion conditions. Once each element is solved in terms of these coordinates, the global mass and stiffness matrices can be obtained as follows.

$$[m] = \frac{\rho AL}{420}\begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{bmatrix} \tag{A.3.1}$$

$$[K] = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{A.3.2}$$

If there is no damping and applied loading, the equation of motion becomes

$$[M]\{\ddot{u}\} + [K]\{u\} = \{0\} \qquad \text{(A.3.3)}$$

The inverse of the mass matrix becomes

$$M^{-1} = \frac{Adj(M)}{\det(M)} \qquad \text{(A.3.4)}$$

Combine equations A.3.3 and A.3.4

$$\{\ddot{u}\} + [M^{-1}][K]\{u\} = \{0\} \qquad \text{(A.3.5)}$$

$$D = [M^{-1}][K] \qquad \text{(A.3.6)}$$

where D is the dynamic matrix , and the eigenvalues can be determined from

$$\det[D - I\lambda] = 0 \qquad \text{(A.3.7)}$$

where λ is the eigenvalue. Therefore, the natural frequencies can be calculated as

$$w = 2\pi f_n = \sqrt{\lambda} \qquad \text{(A.3.8)}$$

This model was used to produce the underside shape predictive model. The result of this model was shown and compared with other models in Section 5.1.

## Appendix B: Calculations for the Mechanical Properties

For isotropic materials Young's modulus is considered as a measure of stiffness, it is defined as the ratio of the uniaxial stress to the uniaxial strain. Thus, the following equation can be determined from the Euler-Bernoulli equation A.1.11.

$$\omega_n = 2\pi f_n = (\beta_n l)^2 \sqrt{\frac{EI}{\rho A l^4}} \tag{B.1}$$

Where $\beta_n l$ is the weighted frequencies and $\omega_n$ is the resulted nature frequencies. The Young's modulus can then be determined from equation B.1, where $\beta_1 l = 4.73, \beta_2 l = 7.853, \beta_3 l = 10.996$ (Inman 2007, 500)

$$E = \frac{4\pi^2 \rho A L^4}{I}\left(\frac{{f_n}^2}{{\beta_n}^4}\right) \tag{B.2}$$

The shear modulus is defined as the ratio of shear stress to the shear strain, and it can be calculated from the measured fundamental torsional frequency. (ASTM, 2002)

$$G = \frac{4Lm{f_t}^2}{bt}\left(\frac{B}{1+A}\right) \tag{B.3}$$

$$A = \frac{0.5062 - 0.8776(b/t) + 0.3504(b/t)^2 - 0.0078(b/t)^3}{12.03\left(\frac{b}{t}\right) + 9.892(b/t)^2}$$

$$B = \frac{(b/t) + (t/b)}{4(t/b) - 2.52(t/b)^2 + 0.21(t/b)^6}$$

The Poisson's ratio is the ratio of the transverse strain to the axial strain when an object is stretched, and once the Young's modulus and shear modulus are avaiable the Poisson's ratio could then be calculated as

$$v = \left({}^{E}/_{2G}\right) - 1 \tag{B.4}$$

If the Timoshenko model was selected for the beam structure prediction, where the shear deformation and rotary inertia are included, a shear coefficient to account for the variation of the shear stress across the cross section is required. Therefore, an equation that is introduced by Hutchinson (2001, 87) could be used to calculate the shear factor for rectangular beams.

$$K = -\frac{2(1+\upsilon)}{[\frac{9}{4a^5b}C_4+\upsilon(1-\frac{b^2}{a^2})]} \tag{B.5}$$

$$C_4 = \frac{4}{45}a^3b(-12a^2 - 15\upsilon a^2 + 5\upsilon b^2) + \sum_{n=1}^{\infty}\frac{16\upsilon^2 b^5(n\pi a - btanh(\frac{n\pi a}{b}))}{(n\pi)^5(1+\upsilon)}$$

In summary, the equations obtained in this section (from B.1 to B.5) were used to find the inputs of the underside shape predictive model (described in Section 5.2). It was assumed that wood was homogeneous and isotropic to make the calculation and measurement simpler. The inaccuracy was corrected by the fine tuning program (described in Chapter 6).

# Appendix C: Newton-Raphson Method

The Newton-Raphson method is one of the most widely used mathematical methods for the determination of roots. The idea of this method is that one starts with an initial guess which is close to the true root, and the function could then be approximated by its tangent line, and one computes the x-intercept of this tangent line. This x-intercept becomes a better approximation to the function's root than the original guess, and hence the same procedure is iterated until obtaining the final root. (Smith, 1998)

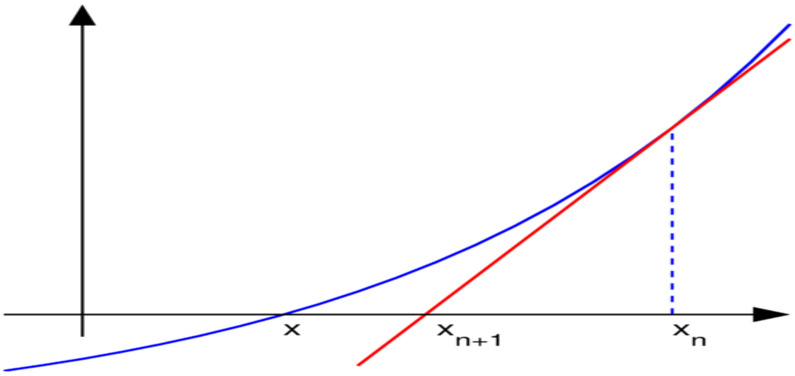


Figure C.1: An illustration of one iteration of Newton_Raphson method (Smith, 1998)

As shown in Figure C.1 an initial guessed value "$x_n$" is chosen for the true value "x", and then the difference between the true value "x" and the guessed value "$x_n$" could be denoted as "h".

$$x = x_n + h \tag{C.1}$$

The function f can be expressed as

$$f(x) = f(x_n + h) = f(x_n) + h * \dot{f}(x_n) \tag{C.2}$$

From equation C.2

$$h = \frac{f(x) - f(x_n)}{\dot{f}(x_n)} \tag{C.3}$$

Combine equations C.1 and C.3

$$x = x_n + \frac{f(x)-f(x_n)}{\dot{f}(x_n)} \tag{C.4}$$

In mathematics field, the Newton-Raphson method is normally used to solve equations of the form $f(x) = 0$, thus, equation C.4 could be rewritten to equation C.5. Rather, some engineering problems such as automatic tuning in this thesis require nonzero target values for $f(x)$, which makes this term a non-zero value.

$$x_{n+1} = x_n - \frac{f(x_n)}{\dot{f}(x_n)} \tag{C.5}$$

As the Newton-Raphson method was used to tune the resulted natural frequencies by a few iterations for this thesis, some changes were made to it. The curve coefficients of the underside shape are initially given to start the iteration process. For each iterative step the frequency error between the desired frequency and the predicted frequency becomes

$$df = f\_req - f(x_o) \tag{C.6}$$

where $f\_req$ is the desired frequency and $f(x_o)$ is the predicted frequency obtained from the initial geometry. According to equation C.4 the change of shape coefficients can be expressed in terms of the user-desired and predicted frequencies as

$$\partial a = \frac{f_{req}-f(x_o)}{\dot{f}(x_o)} \tag{C.7}$$

$$\dot{f}(x_o) = \frac{\partial f}{\partial a} = \frac{\Delta f}{\Delta x} \tag{C.8}$$

Assume “x” moves slightly to the second point, a new “y” value can then be obtained. Thus, the slope of these two points can be calculated by making $\Delta x = 0.00001$.

$$\Delta f = f\_new - f\_old \tag{C.9}$$

Substituting equations C.8 and C.10 into equation C.9:

$$\partial a = \frac{f\_req - f\_old}{\partial f / \partial a} \tag{C.10}$$

The new “x” value can be obtained as

$$x\_new = x\_old + \partial a \tag{C.11}$$

where $x\_new$ is the new point that moves forwards to the result, and the best shape coefficients can be achieved once the iterative process finishes.

The equations obtained in this section show the determination of one target natural frequency. In this thesis, the lowest three natural frequencies of marimba bars were tuned to the target frequency ratios. The relationships between the resulting natural frequencies and the shape coefficients are described in equation C.12. (Detailed code is presented in Appendix G.1)

$$\begin{bmatrix} \Delta a_1 \\ \Delta a_2 \\ \Delta a_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} & \frac{\partial f_1}{\partial a_3} \\ \frac{\partial f_2}{\partial a_1} & \frac{\partial f_2}{\partial a_2} & \frac{\partial f_2}{\partial a_3} \\ \frac{\partial f_3}{\partial a_1} & \frac{\partial f_3}{\partial a_2} & \frac{\partial f_3}{\partial a_3} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} f1_{new} \\ f2_{new} \\ f3_{new} \end{bmatrix} - \begin{bmatrix} f1_{old} \\ f2_{old} \\ f3_{old} \end{bmatrix} \right\} \tag{C.12}$$

## Appendix D: G-Code Generated by the Auto-tuning Program

The following code were generated by the G-code program written in Appendix G.18.

```
START

N10
N20 G90
N30 G40 G80
N40 M06 T10
N50 M03 S1000
N60 G0  X403.908 Y-317.748
N70 G43 Z-200. H10 M09
N80  Z-220
N90 G1 X403.737 Y-301.628 Z-310. F320
N100 X402.571 Y-301.43 F200
N110 G2 X401.11 Y-301.038 R8.
N120 G1 X399.944 Y-300.602
N130 G2 X399.763 Y-300.533 R8.
N140 G1 X398.638 Y-300.081
…
…
…
N1610 X244.072 Y-317.748 Z-220.
N1620 G0  Z-200.
N1630 M09
N1640 M30
```

## Appendix E: G-Code Generated by AlphaCam

The following code was generated by AlphaCam.

```
Operation List    POST: Fanuc M21 vertical Mill
--------------------------------------------------------------------------------
OP 1   FINISH PASS   TOOL 10   END MILL - 16MM
EFFECTIVE DIAMETER 16
Feed Distance: 284.3   Time for OP 1: 1m 20s
--------------------------------------------------------------------------------
Total Feed Distance ............................................................... 284.3
Tool Change Time ................................................................ 0m 10s
Total Time ............................................................................ 1m 30s
--------------------------------------------------------------------------------
Material: Mild Steel Roughing
Use Emulsion Coolant
--------------------------------------------------------------------------------
START

N10 (PROGRAM PRODUCED  - 13 APR 11)
N20 G90
N30 G40 G80
'(OP 1   FINISH PASS   TOOL 10   END MILL - 16MM)
'(EFFECTIVE DIAMETER 16)
N40 M06 T10         'Select tool and offset
N50 M03 S1000      'Next tool is 00, Next XY is 403.908, -317.748
N60 G0  X403.908 Y-317.748
N70 G43 Z-200. H10 M09
N80  Z-220.
N90 G1 X403.737 Y-301.628 Z-310. F320
N100 X402.571 Y-301.43 F200
N110 G2 X401.11 Y-301.038 R8.
N120 G1 X399.944 Y-300.602
N130 G2 X399.763 Y-300.533 R8.
N140 G1 X398.638 Y-300.081
…
…
…
N1610 X244.072 Y-317.748 Z-220.
N1620 G0  Z-200.
N1630 M09
N1640 M30
%
```

# Appendix F: Fine Tuning Results

Another two tests of the fine tuning program (and tuning automation) are presented in this section. The inputs of the fine tuning program, details of the testing records and the results are presented.

## Appendix F.1: Fine Tuning Test (27/05/2011)

The properties of the tested Tasmanian oak blank are presented in Table F.1, the primary prediction and the experimental results are shown in Table F.2. Detailed results of each iteration were recorded in Table F.3. It can be seen that the error of the first three natural frequencies were 1.12Hz, 4.55Hz and 46Hz, which have largely improved tuning results of this marimba bar.

| 27/5/2011 | Tasmanian Oak |
|---|---|
| Properties | |
| rho (kg/m$^3$) | 550.995 |
| E (GPa) | 15.81 |
| G (GPa) | 6.598 |
| L (mm) | 302.0 |
| b (mm) | 65.0 |
| do(mm) | 19.05 |
| Target Freqs | |
| $f_1$ (Hz) | 380 |
| $f_2$ (Hz) | 1520 |
| $f_3$ (Hz) | 3724 |

Table F.1 Properties of wood (27/05/2011)

| | First Cut | Second Cut | Third Cut | Fourth Cut | Fifth Cut |
|---|---|---|---|---|---|
| Incremental target f1 (Hz) | 1033.1 | 957 | 861.6 | 769.2 | 665.2 |
| Experimental f1 (Hz) | 1021 | 948 | 853 | 755 | 655 |
| Incremental target f2 (Hz) | 2920.9 | 2821.5 | 2682.5 | 2467.2 | 2318.7 |
| Experimental f2 (Hz) | 2621 | 2543 | 2413 | 2197 | 2046 |
| Incremental target f3 (Hz) | 4508 | 5522.7 | 5344.1 | 5164 | 4931.7 |
| Experimental f3 (Hz) | 5646.4 | 4448 | 4335 | 4210 | 4052 |

Table F.2 Fine tuning results (27/05/2011)

| First Cut | Incremental targets | Experiment | Repredicted targets | Error |
|---|---|---|---|---|
| $f_1$ (Hz) | 1045.8 | 1021 | 1033.1 | -12.1 |
| $f_2$ (Hz) | 2945.8 | 2621 | 2920.9 | -299.9 |
| $f_3$ (Hz) | 5677.2 | 4508 | 5646.4 | -1138.4 |
| do(mm) | 17.45 | 17.395 | -0.055 | |
| Second Cut | Incremental targets | Experiment | Repredicted targets | Error |
| $f_1$ (Hz) | 954.4 | 948 | 957 | -9 |
| $f_2$ (Hz) | 2816 | 2543 | 2821.5 | -278.5 |
| $f_3$ (Hz) | 5515.8 | 4448 | 5522.7 | -1074.7 |
| do(mm) | 15.84 | 15.878 | 0.038 | |
| Third Cut | Incremental targets | Experiment | Repredicted targets | Error |
| $f_1$ (Hz) | 861.1 | 853 | 861.6 | -8.6 |
| $f_2$ (Hz) | 2671.5 | 2413 | 2672.5 | -259.5 |
| $f_3$ (Hz) | 5342.8 | 4335 | 5344.1 | -1009.1 |
| do(mm) | 14.24 | 14.233 | -0.007 | |
| Fourth Cut | Incremental targets | Experiment | Repredicted targets | Error |
| $f_1$ (Hz) | 766.3 | 755 | 769.2 | -14.2 |
| $f_2$ (Hz) | 2510.4 | 2197 | 2467.2 | -270.2 |
| $f_3$ (Hz) | 5155 | 4210 | 5164 | -954 |
| do(mm) | 12.63 | 12.588 | -0.042 | |
| Fifth Cut | Incremental targets | Experiment | Repredicted targets | Error |
| $f_1$ (Hz) | 670 | 655 | 665.2 | -10.2 |
| $f_2$ (Hz) | 2330.6 | 2046 | 2318.7 | -272.7 |
| $f_3$ (Hz) | 4948.3 | 4052 | 4931.7 | -879.7 |
| do(mm) | 11.03 | 10.99 | -0.04 | |

Table F.3 Detailed records of experiment (27/05/2011)

| Input Freqs | | |
|---|---|---|
| f1 (Hz) | 391.738 | |
| f2 (Hz) | 1808.182 | |
| f3 (Hz) | 4436.764 | |
| Measured Freqs | | |
| f1 (Hz) | 372 | |
| f2 (Hz) | 1496 | |
| f3 (Hz) | 3654 | Error |
| Measured d0 | 6.046 | 0.14 |
| Re-predicted Freqs | | Error |
| f1 (Hz) | 373.12 | 1.12 |
| f2 (Hz) | 1500.55 | 4.55 |
| f3 (Hz) | 3700 | 46 |

Table F.4 Final results of the fine tuning process (27/05/2011)

**Appendix F.2: Fine Tuning Test (06/06/2011)**

The properties of the tested Tasmanian oak blank are presented in Table F.4, the primary prediction and the experimental results are shown in Table F.5. Detailed results of each iteration were recorded in Table F.6. It can be seen that the error of the first three natural frequencies were 0.5Hz, 12Hz and 54Hz, which have largely improved tuning results of this marimba bar.

| 6/6/2011 | Tasmanian Oak |
|---|---|
| Properties | |
| rho (kg/m$^3$) | 611.0152 |
| E (GPa) | 16.0303 |
| G (GPa) | 6.076 |
| L (mm) | 300.0 |
| b (mm) | 65.0 |
| t (mm) | 19.2 |
| Target Freqs | |
| $f_1$ (Hz) | 300 |
| $f_2$ (Hz) | 1200 |
| $f_3$ (Hz) | 2940 |

Table F.5 Properties of wood (06/06/2011)

| | First Cut | Second Cut | Third Cut | Fourth Cut | Fifth Cut |
|---|---|---|---|---|---|
| Incremental target f1 (Hz) | 988.1 | 852.7 | 719.6 | 577.6 | 439 |
| Experimental f1 (Hz) | 951 | 831 | 702 | 565 | 431 |
| Incremental target f2 (Hz) | 2824.2 | 2621.4 | 2389.9 | 2095.5 | 1757.1 |
| Experimental f2 (Hz) | 2625 | 2434 | 2101 | 2064 | 1611 |
| Incremental target f3 (Hz) | 5447.1 | 5230.1 | 4993.7 | 4685.9 | 4307.8 |
| Experimental f3 (Hz) | 4586 | 4410 | 4200 | 3964 | 3673 |

Table F.6 Fine tuning results (06/06/2011)

| First Cut | Incremental targets | Experiment | Repredicted targets | Error |
|---|---|---|---|---|
| $f_1$ (Hz) | 980.1 | 951 | 988.1 | -37.1 |
| $f_2$ (Hz) | 2809.4 | 2625 | 2824.2 | -199.2 |
| $f_3$ (Hz) | 5430.9 | 4586 | 5447.1 | -861.1 |
| do(mm) | 16.93 | 17.05 | 0.12 | |
| Second Cut | Incremental targets | Experiment | Repredicted targets | Error |
| $f_1$ (Hz) | 849.2 | 831 | 852.7 | -21.7 |
| $f_2$ (Hz) | 2614 | 2434 | 2621.4 | -187.4 |
| $f_3$ (Hz) | 5221.7 | 4410 | 5230.1 | -820.1 |
| do(mm) | 14.65 | 14.704 | 0.054 | |
| Third Cut | Incremental targets | Experiment | Repredicted targets | Error |
| $f_1$ (Hz) | 714.9 | 702 | 719.6 | -17.6 |
| $f_2$ (Hz) | 2378.8 | 2101 | 2389.9 | -288.9 |
| $f_3$ (Hz) | 4980.5 | 4200 | 4993.7 | -793.7 |
| do(mm) | 12.38 | 12.452 | 0.072 | |
| Fourth Cut | Incremental targets | Experiment | Repredicted targets | Error |
| $f_1$ (Hz) | 578.1 | 566 | 577.6 | -11.6 |
| $f_2$ (Hz) | 2096.8 | 2076 | 2095.5 | -19.5 |
| $f_3$ (Hz) | 4687.6 | 3976 | 4685.9 | -709.9 |
| do(mm) | 10.11 | 10.102 | -0.008 | |
| Fifth Cut | Incremental targets | Experiment | Repredicted targets | Error |
| $f_1$ (Hz) | 440 | 431 | 439 | -8 |
| $f_2$ (Hz) | 1760 | 1611 | 1757.1 | -146.1 |
| $f_3$ (Hz) | 4312 | 3673 | 4307.8 | -634.8 |
| do(mm) | 7.833 | 7.818 | -0.015 | |

Table F.7 Detailed records of experiment (06/06/2011)

| Input Freqs | | |
|---|---|---|
| $f_1$ (Hz) | 309 | |
| $f_2$ (Hz) | 1324.6 | |
| $f_3$ (Hz) | 3405.7 | |
| Measured Freqs | | |
| $f_1$ (Hz) | 301 | |
| $f_2$ (Hz) | 1216 | |
| $f_3$ (Hz) | 3020 | |
| Measured do | 5.69 | |
| Re-predicted Freqs | | Errors |
| $f_1$ (Hz) | 300.5 | 0.5 |
| $f_2$ (Hz) | 1204 | 12 |
| $f_3$ (Hz) | 2966 | 54 |

Table F.8 Final results of the fine tuning process (06/06/2011)

## Appendix G: Programming Code

Tuning automation is a key feature of this thesis. Firstly, the process starts with the underside shape prediction under a few assumptions. Secondly, the fine tuning program re-arrange the fine tuning process and corrects the error that is caused by the unavoidable uncertainties of test material. Finally, the physical machining centre was automatically controlled for the tuning automation. All the above processes needed to be well managed, and the structure of the program is shown in Figure G.1.

Three major components are included: underside shape predictive model, the fine tuning program and the hardware controlling program. The underside shape predictive model developed by Entwistle and McGrehan (2007) has successfully achieved the task of predicting the correct dimensions of non-prismatic beams. This model is accurate for aluminium beams as the mechanical properties of the tested material could be accurately measured and provided to this model. The functions of the program include:

1) Main function
2) Method_Timoshenko function
3) Shape function
4) Beam_recept_Timo function
5) Bracket_natfreq function
6) Find_freq_timo function
7) Shear_factor_timo function

However, this model would give error for wooden beams. Thus, as described in Chapter 6 (the fine tuning program) marimba bars were incrementally manufactured from above so that the required incremental target frequencies and the experimental frequencies for each iteration can be achieved, and therefore to generate the “relationship linear equations” (as described in Chapter 6). The incremental target frequencies are the resulting natural frequencies, which are determined by the predictive model based on the underside shape of iteration. However, the predictive model was developed to generate the shape coefficients rather than the discrete points of the thickness along the length of marimba bars as described in Section 5.1.

This brings the problem that when the wood blank is manufactured from above the underside surface is cut vertically down as shown in Figure 6.2. The thickness at different locations of the marimba bar is calculated. Thus, the existed predictive model was expected to be modified so that it could read the discrete thickness coordinates as the shape inputs rather than the shape coefficients. This can avoid the predictive error as the shape coefficients, which are calculated based on the underside shape at each iteration, might not produce the correct curve that match all the thickness points. Therefore, once the predictive model was capable of taking both the shape coefficients and the thickness coordinates of the shape as the shape inputs, the accuracy of this predictive model could be ensured. Therefore, the functions that the fine tuning program included are:

8) Tuning_Program function
9) Shape_matrix function
10) Method_Timo_TP function
11) Find_freq_timo_TP function
12) Shape_reprediction function
13) Desired_vs_experimetnal function

Such modification can apply on the structures for other applications, which is described in the Cartesian coordinate system as the shape inputs to the predictive model, to predict the resulting natural frequencies. Such structures might not necessarily be cubic, parabolic or the shapes that are described by mathematic equations.

Finally, the hardware controlling program has converted the calculated underside shape into the coordinates of the CNC tool path, and therefore the signals that the CNC machine reads. The functions of the program included:

14) Tool_path_generation function
15) Coordinates_to_steps function
16) Steps_to_signals function
17) Driving_motors function
18) G_code_generation function

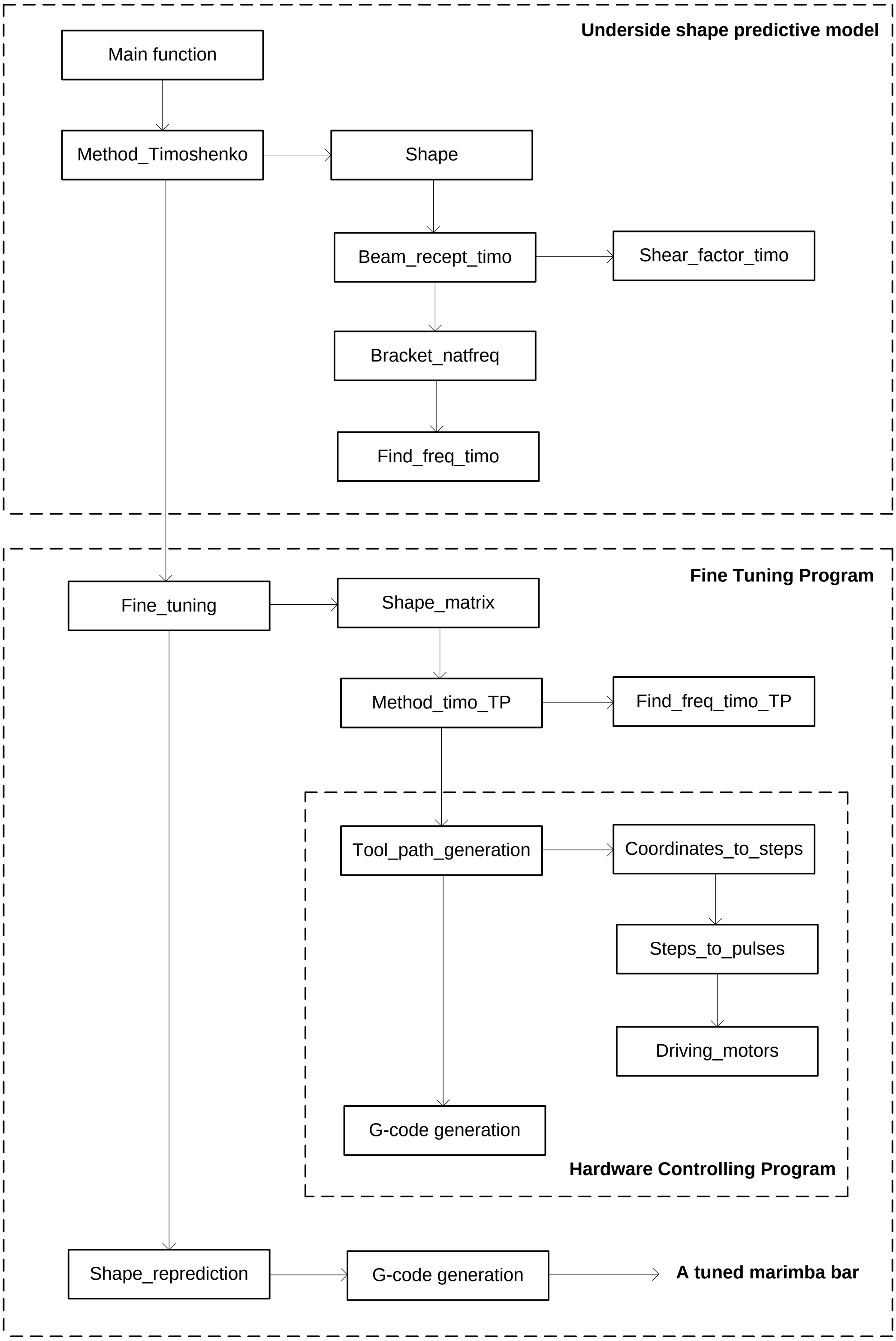


Figure G.1: Structure of the fine tuning program

### Appendix G.1: main.m

```
%Function main.m
%This is the main file of the program, and it will call other files
%Written in 18/09/09 by Zhao Mingming
%This program is a modified code from Dr.Rodney Entwistle

clear all;
global E rho G x_section nu d0 width bar_length b d len;

%Give parameter values
% w = frequency rad/s
% E = Elastic Modulus Pa
% rho = density kg/m^3
% b = width (or diameter) of beam segments in sequence
% d = depth (or diameter) of beam segments in sequence
% len = length of beam segments in sequence
% shape = 'rectangular' or 'circular' cross-section
%------------------------------------------------------------------------------------------------------------
selected_frequency=300
frequency_required=[selected_frequency;4;9.8];
rho=837.187565;
E=17.96e9;
G=6.75e9;
nu=(E/(2*G))-1;

d0=0.01915;
width=0.06;
bar_length=0.3;

a1=6; a2=0.3; a3=0.003;
 coeffs=[a1;a2;a3];
tol=[0.0001;0.001;0.001];
number_fn=3; shape='rectangular';

profile='cubic';
analysis='timoshenko';
if strcmp(analysis,'timoshenko')
    method='Method_Timoshenko';
    method_NN='Method_Timoshenko_NN';
elseif strcmp(analysis,'euler')
    method='Method_Euler';
        method_NN='Method_Timoshenko_NN';
        Euler_NN='method_Euler_NN';
else
    error('Unknown analysis method');
end;
x_section='rectangular';
number_of_elements=300;
ploton=false;
%------------------------------------------------------------------------------------------------------------

delta_a=1.000001; df=1.1.*tol;

while any(abs(df)>tol)
```

```
%displys the resulted natural frequencies
 frequency_old=eval([method
'(profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,number_of_elements,tol,number_f
n,ploton)']);
   slope=[];

%for the first three natural frequencies
for k=1:3
        coeffs_new=coeffs;
        coeffs_new(k)=coeffs_new(k).*delta_a;
        frequency_new=eval([method
'(profile,coeffs_new,E,G,rho,x_section,nu,d0,width,bar_length,number_of_elements,tol,num
ber_fn,ploton)'])
        slope=[slope (frequency_new-frequency_old)./(coeffs_new(k)-coeffs(k))];
   end;
   df=frequency_required-frequency_old;
   da=(slope)\df;
   coeffs=coeffs+da;
end;

%Get the correct coefficients and check natural frequencies
coeffs=coeffs-da;
Natural_Frequencies_timo=eval([method
'(profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,number_of_elements,tol,number_f
n,ploton)'])

%Calculate the positions of the bearing points
Hole_Positions=Drilled_Hole_Predictions(E,rho,b,d,len,shape)

%Determine the G-code so that other CNC machines could cut this marimba bar directly
G_Code_Generation(d, len, bar_length);
%----------------------------------------------------------------------------------------------------------
%Call the fine tuning program (correct the uncertainties of the tested material)
Tuning_Program(method,method_NN,profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_leng
th,number_of_elements,tol,number_fn,ploton,d)
%----------------------------------------------------------------------------------------------------------
%Re-predict the shape geometry based on the predicted and measured frequencies
Shape_Reprediction(method,method_NN,profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_l
ength,number_of_elements,tol,number_fn,ploton,d)
```

## Appendix G.2 Method_Timoshenko.m

```
Function
out=Method_Timoshenko(profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,number_
of_elements,tol,number_fn,ploton)

global d b len;
a1=coeffs(1); a2=coeffs(2); middle_d=coeffs(3);
[b,d,len]=shape(width,d0,bar_length,profile,a1,a2,middle_d,number_of_elements,ploton);

%check size of steps in profile
delta_d_max=0;
for k=1:length(d)-1
  delta_d_max=max([delta_d_max abs((d(k)-d(k+1))/d(k))]);
end;

% Find the natural frequencies to this tolerance (rad/s)
tolerance=tol(1);
how_many=number_fn;
%place to store results by using Timoshenko
recept_all_timoshenko=[];
freq=[]; % place to store freqs

   for w=logspace(2,5,600); % nice spread of frequencies
  recept_timoshenko=beam_recept_timoshenko(w,E,rho,b,d,len,x_section,G,nu);
  recept_all_timoshenko=[recept_all_timoshenko recept_timoshenko(1,1)];
     freq=[freq w]; % record the freq in rad/s
end;

% Find the resonances - Timoshenko
bracket=bracket_natfreq(recept_all_timoshenko,freq,how_many);
if isempty(bracket)
  error('No natural frequencies found. Perhaps alter frequency span?');
end;

nat_freqs=find_freq_timo(bracket,tolerance);
fundamental_Timoshenko=nat_freqs(1)./(2*pi);
ratios_Timoshenko=nat_freqs./nat_freqs(1);

if ploton
  out=d;
else
  out=[fundamental_Timoshenko;ratios_Timoshenko(2:how_many)];
end;
```

## Appendix G.3 Shape.m

```
function
[b,d,len]=shape(width,d0,bar_length,profile,a1,a2,middle_d,number_of_elements,ploton)
%Generates the sizes of all of the beam elements based on the equation of the cutout.
%Shape currently a cubic (no linear term) y=a1*|x|^3 + a2*|x|^2 + a3

%to make sure the element decided to be break into are even numbers
if(mod(number_of_elements,2))~=0
   error('Choose even number for number of elements');
end;
size_of_elements=bar_length./number_of_elements;
    a3=middle_d; % depth at centre of bar

%on x-axis, x varies starting from 0 to half of the bar length in an increments of the size of
each element
x=0:size_of_elements:bar_length*0.5;

if a3>=d0
   error('complete sever!');
elseif a3<=0
   error('No cutout!');
end;

%Calculate the shape of the cutout.
switch profile
   case 'cubic'
      y=a1.*x.^3 + a2.*x.*x + a3;
   case 'parabola'
           for k=1:length(x) % flat with parabolic ends
         if x(k)<=a1;
            y(k)=a3;
         end;
         if x(k)>a1
                %parabola with origin shifted to end of parallel portion
                x_temp=x(k)-a1;
            y(k)=a2.*x_temp.*x_temp+a3;
         end;
      end;
   case 'cosine'
      y=a3+a1.*(1-cos(a2*pi.*x./bar_length));
   case 'exp_plus'
      y=(a3./2).*(exp(a1.*x)+exp(a2.*x));
   case 'exp_minus'
      y=a3.*(1+exp(a1.*x)-exp(a2.*x));
   case 'prismatic'
      y=d0.*(x+1)./(x+1);
   otherwise
      error('Unknown cutout profile');
end;



for k=1:length(y);
```

```
   if y(k) > d0;
      y(k)=d0;
   elseif y(k) < 0;
      error('Part of beam thickness gone to zero');
   end;
end;

%to make the cutout shape more smooth, the resulted cutout values above at
%two points close together can be joined by calculating the mean value of
%those two vlaues
for k=1:number_of_elements/2
       d_temp(k)=(y(k)+y(k+1))./2; % mean thickness of section
end;
d=[]; len=[]; k=0;

% Merge contiguous elements of same depth into one.
% This removes unnecessary divisions and minimises the number of elements.
while k<number_of_elements/2
   k=k+1;
   d=[d d_temp(k)];
   len=[len size_of_elements];

   %while((k<number_of_elements/2) && (d_temp(k+1)==d_temp(k)))
   %    len(length(len))=len(length(len))+size_of_elements;
   %    k=k+1;
   %end;

end;
b=width*ones(1,2*length(d));
% mirror to build final vectors
d=[fliplr(d) d];
len=[fliplr(len) len];

if ploton
   H1=figure(1);
   subplot(2,1,1);
   line(1000.*[0 -bar_length*0.5 fliplr(-x) x bar_length*0.5 0],1000.*[0 0  fliplr(y) y 0 0]);
   axis equal;
   xlabel('To scale');
   subplot(2,1,2);
   line(1000.*[0 -bar_length*0.5 fliplr(-x) x bar_length*0.5 0],1000.*[0 0  fliplr(y) y 0 0]);
   xlabel('Thickness stretched');
end;
```

## Appendix G.4 beam_recept_timoshenko.m

```
% Timoshenko Beam - based on BJ Stone paper P I Mech Eng: PtC J Mech Eng Sc Vol 206, 1992. pp87-94.
%----------------------------------------------------------------------------------------------------
function recept_timoshenko=beam_recept_timoshenko(w,E,rho,b,d,len,shape,G,nu)

% recept_euler = 2x2 tip receptance of the whole beam at freq w.
 %Here the number of sections is set to be the number of depth values
sections=length(d);

%the number of values in width and length should be the same as depth, otherwise, cannot calculate in different values
if(length(b)~=sections)||(length(len)~=sections)
   error('Vectors for breadth, depth and length must be same size.');
end;

%if x_section is set to be retangular ot circular, the flexural rigidity and cross sectional area can be calculated, and the type of x_section is passed from main function
if strcmp(shape,'rectangular')
   I=b.*d.^3./12;
   EI=E.*I;  % Flexural rigidity for each section
   area=b.*d;  % Cross section area of each section

   % Calculates Hutchinson's factor for each element
   %lambda=shear_factor_timoshenko(nu,b,d);
   lambda=shear_factor_timoshenko(nu,b,d);
elseif strcmp(shape,'circular')
   I=pi.*d.^4./64;
   EI=E.*I;
   area=pi.*d.*d./4;
   %Shear shape factor.  Ref.S.S. Rao, Mechanical Vibrations, 4th Ed,p624.
   lambda=9/10;
else
   error('Unknown cross-sectional shape');
end;

for k=1:sections
aplus(k)=1+E./(lambda(k).*G);
aminus(k)=1-E./(lambda(k).*G);
end;

%there are two sets of equations to calculate tip receptances, and it depends on whetehr w*w is larger or smaller than
%lambda(k).*G.*area(k)./(rho.*I(k))
signs_ref=[-1 1 1 1 -1 -1 -1 -1 -1 1 1 -1 -1 1];

% Need to do this since each section may have different  lambda(k).*G.*area./(rho.*I)
for k=1:sections

   if (w.*w < lambda(k).*G.*area(k)./(rho.*I(k)))
      signs=signs_ref;
   else
      signs=-signs_ref;
```

```
    end;

    m(k)=rho*I(k)*w*w;
    n(k)=sqrt(m(k)*(m(k).*aminus(k)^2+4*E*area(k)));
    eta(k)=sqrt(((signs(1)*m(k)*aplus(k)) + signs(2)*n(k))/(2*EI(k)));
    xi(k)=sqrt(((m(k)*aplus(k)) + n(k))/(2*EI(k)));
    s(k)=rho*w*w/(lambda(k)*G*xi(k))-xi(k);
    r(k)=rho*w*w/(lambda(k)*G*eta(k))+signs(3)*eta(k);
    e(k)=r(k)*eta(k)^2/(s(k)*xi(k)^2);
    xi_len(k)=xi(k)*len(k);
    eta_len(k)=eta(k)*len(k);
    c(k)=cos(xi_len(k)); ch(k)=cosh(eta_len(k));
    si(k)=sin(xi_len(k)); sh(k)=sinh(eta_len(k));
    f1(k)=si(k)*sh(k);
    f2(k)=c(k)*ch(k);
    f3(k)=f2(k)-1;
    f4(k)=si(k)*ch(k);
    f5(k)=c(k)*sh(k);
    f6(k)=c(k)-ch(k);
    recip_delta(k)=1/(rho*w*w*area(k)*(2-2*f2(k)+signs(4)*(e(k)+signs(5)/e(k))*f1(k)));
end;

% tip receptance of first section
if (w*w < lambda(k)*G*area(1)/(rho*I(1)))
  signs=signs_ref;
else
  signs=-signs_ref;
end;
beta=recip_delta(1)*[signs(6)*f5(1)*(r(1)*eta(1)^2/(s(1)*xi(1))-
eta(1))+f4(1)*(s(1)*xi(1)^2/(r(1).*eta(1))-xi(1)),...

signs(7)*f3(1)*(s(1)*xi(1)+r(1)*eta(1))+signs(8)*f1(1)*(r(1)*eta(1)^2/xi(1)+signs(9)*s(1).*
xi(1).^2./eta(1));...

signs(7)*f3(1)*(s(1)*xi(1)+r(1)*eta(1))+signs(8)*f1(1)*(r(1)*eta(1)^2/xi(1)+signs(9)*s(1).*
xi(1).^2./eta(1)),...
     f4(1)*r(1)*eta(1)*(s(1)-r(1)*eta(1)/xi(1))+f5(1)*s(1)*xi(1)*(r(1)-s(1)*xi(1)/eta(1))];

% add all of the remaining sections in section C
for k=2:sections
  if (w*w < lambda(k).*G.*area(k)/(rho.*I(k)))
     signs=signs_ref;
  else
     signs=-signs_ref;
  end;

  gamma1=recip_delta(k).*[signs(6).*f5(k).*(r(k).*eta(k).^2./(s(k).*xi(k))-
eta(k))+f4(k).*(s(k).*xi(k).^2./(r(k).*eta(k))-xi(k)),...

signs(10).*f3(k).*(s(k).*xi(k)+r(k).*eta(k))+signs(11).*f1(k).*(r(k).*eta(k).^2./xi(k)+signs(9
).*s(k).*xi(k).^2./eta(k));...

signs(10).*f3(k).*(s(k).*xi(k)+r(k).*eta(k))+signs(11).*f1(k).*(r(k).*eta(k).^2./xi(k)+signs(9
).*s(k).*xi(k).^2./eta(k)),...
```

```
        f4(k).*r(k).*eta(k).*(s(k)-r(k).*eta(k)./xi(k))+f5(k).*s(k).*xi(k).*(r(k)-s(k).*xi(k)./eta(k))];
  gamma2=recip_delta(k).*[signs(12).*sh(k).*(r(k).*eta(k).^2./(s(k).*xi(k))-eta(k))+si(k).*(s(k).*xi(k).^2./(r(k).*eta(k))-xi(k)),...
        signs(13).*f6(k).*(r(k).*eta(k)-s(k).*xi(k));...
        signs(14).*f6(k).*(r(k).*eta(k)-s(k).*xi(k)),...
        sh(k).*(r(k).*s(k).*xi(k)-s(k).^2.*xi(k).^2./eta(k))+si(k).*(r(k).*s(k).*eta(k)-r(k).^2.*eta(k).^2./xi(k))];
  gamma3=gamma1.*[1 -1; -1 1];


  alpha=gamma3-gamma2'*((gamma1+beta)\gamma2); % build the structure
  beta=alpha;  % new tip receptance ready for next section to be added
end;

recept_timoshenko=alpha; % This is a 2x2 matrix of tip receptances.
```

### Appendix G.5 bracket_natfreq.m

```
function bracket=bracket_natfreq(x,freq,max_roots)

% Extract the frequencies between which the resonances lie.
% For free-free beams, the resonances have a +1 to -1 phase shift as w rises.  Hence need to look for + to - changes in alpha11.

bracket=[];
how_many=0;  % Number of singularities found at first

for k=1:length(x)-1

   %when the current receptance is equal to or larger than zero and the next one is negative, then record this bracket
   if (x(k)>=0)&&(x(k+1)<0)
      % we have a positive to negative singularity
      bracket=[bracket; freq(k) freq(k+1)];

      % record how many brackets are found out
      how_many=how_many+1;

      %if the bracket found out are already the number of the max roosts
      %set in main function, then stop
      if how_many==max_roots;
         break;
      end;
   end;
end;
```

## Appendix G.6 find_freq_timo.m

```
function nat_freqs=find_freq_timo(bracket,tol)

% Given a span of frequencies known to contain the natural frequency,  use the bisection method to locate it within tolerance tol.
% Note: bisection finds a root or a singularity.

global b len d E rho x_section G nu;

jmax=40;  % Maximum allowed number of bisections
nat_freqs=[];
[rows]=size(bracket);

if(rows==0)
   error('No frequencies bracketed?');
end;

for k=1:rows

   w_low=bracket(k,1);  %load the lower frequencies at the brackets found
   w_high=bracket(k,2); %load the higher frequencies at the brackets found

   %ymid is the receptance at the brackets with higher freuencies
   ymid=beam_recept_timoshenko(w_high,E,rho,b,d,len,x_section,G,nu);
   %y is the receptance at the brackets with lower frequencies
   y=beam_recept_timoshenko(w_low,E,rho,b,d,len,x_section,G,nu);

   %if the signs of the above two receptances are same, then error
   if(y.*ymid>=0)
      error('Roots must be bracketed for find_freq to work.');
   end;

   %if the receptance at the brackets with lower frequencies is negative,
   %then store the lower frequency in "estimate". on the other hand, if
   %the receptance at the brackets with lower frequencies is positive or
   %zero, then store the higher frequency in "estimate".
   if(y<0)
      estimate=w_low;
      dw=w_high-w_low; %delta frequency
   else
      estimate=w_high;
      dw=w_low-w_high; %delta frequency
   end;

   for j=1:jmax
      dw=dw.*0.5;

      % "wmid" is the frequency in the middle of the lower frequency
      %and the higher frequency
      wmid=estimate+dw;

      %the receptance at the freuency of "wmid" can be found by sending
      %the frequency "wmid" to the "beam_recept_timoshenko" again.
```

```
        ymid=beam_recept_timoshenko(wmid,E,rho,b,d,len,x_section,G,nu);

        %if the receptance at the frequency of "wmid" is equal or smaller
        %than zero, then the frequency of "wmid" can be stored into
        %"estimate" insted of "w_low"
        if(ymid<=0);
           estimate=wmid;
        end;

        %if the magnitude of "dw" is smaller than the tolerance set in the
        %main function, then the frequency in "estimate" is recorded and
        %stored
        if(abs(dw)<tol)
           nat_freqs(k)=estimate;
        end;

        if(ymid==0)
           nat_freqs(k)=estimate;
        end;

    end;
end;

nat_freqs=nat_freqs';
```

## Appendix G.7 shear_factor_timoshenko.m

```
function lambda=shear_factor_timoshenko(nu,b,d)

% Shear factor calcs using Hutchinson (2001) eqn 47
 num_terms=15;   % Number of terms for summation lambda=zeros(1,length(b));

% Calculate truncated approximation to infinite series
for k=1:length(b)
  sum_s=0;
  for n=1:num_terms
      sum_s=sum_s+((16*nu*nu*b(k)^5)*(n*pi*d(k)                                  -
b(k)*tanh(n*pi*d(k)/b(k))))/(n^5*pi^5*(1+nu));
  end;

  % Calculate first term in C4 expression
  first_term=(d(k)^3*b(k)*(-12*d(k)*d(k)-15*nu*d(k)*d(k)+5*nu*b(k)*b(k))*4/45);
  c4=first_term+sum_s;

  %lambda=(-(2*(1+nu))/(9*c4/(4*d(k)^5*b(k)) + nu*(1-(b(k)/d(k))^2)))*ones(1,length(b));
  lambda(k)=5/6;  % Overrides with a constant kappa:0.888
end;
```

## Appendix G.8 Tuning_Program.m

```
function
Tuning_Program(method,method_NN,profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_leng
th,number_of_elements,tol,number_fn,ploton,d)

nodeon=false;
b=[];
for bbb=1:number_of_elements
   b(bbb)=width;
end
len=[];
for lll=1:number_of_elements
   len(lll)=bar_length./number_of_elements;
end
%-----------------------------------------------------------------------------------------------------------
%These three parameters should be passed from the previous function, which determines the
shape of the beam
d1=coeffs(1); d2=coeffs(2); d3=coeffs(3);
coeffs=[d1;d2;d3];
% a matrix, which gives a relationship between possible frequencies and the parameters (a1,
a2, a3
fn_matrix=shape_matrix(method,profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,nu
mber_of_elements,tol,number_fn,ploton)
%-----------------------------------------------------------------------------------------------------------
%Get the current shape for 442Hz
yaya=0;
for c=1:(number_of_elements-1)
   if (d(c) ~= d(c+1))
      yaya=yaya+1;
      where(yaya)=c;
   end;
end;
fixed_position=where(1);
last_fixed_position=(number_of_elements-fixed_position)+1;

% the new shape
New_d=[];
for k=1:fixed_position
   New_d(k)=d(k);
end;
for k=last_fixed_position:number_of_elements
   New_d(k)=d(k);
end;
%-----------------------------------------------------------------------------------------------------------
lalala=0;
f1=[];f2=[];f3=[];

%Here "1" represents the depth as same as the final shape, and the prediction is backwards,
so "0.4" means the depth is "0.4 times" of the difference between the uncutted depth and the
final depth higher than the final depth. Also I want each cut is "0.005 meters" deeper, then
the increment is set to be "0.005"
cut_perc=1
lala=0;
```

```
lalala=lalala+1;
%length_all_the_time=((fixed_position)*(bar_length/600):(bar_length/600):(last_fixed_posi
tion)*(bar_length/600));
   %Here I moved "d" curve upper by "1/10"
   for k=(fixed_position+1):(last_fixed_position-1)
      lala=lala+1;
      New_ddd(lala)=((d0-d(k))*(1-cut_perc))+d(k);
   end;
lala=0;
   for k=(fixed_position+1):(last_fixed_position-1)
      lala=lala+1;
      New_d(k)=New_ddd(lala);
   end;

% check the frequencies of this new shape, and I send the depth curve to the frequency file
directly
ploton=false;
new_Natural_Frequencies=eval([method_NN
'(New_d,profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,number_of_elements,tol,nu
mber_fn,ploton)'])

TOOLPATH_GENERATION(profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,num
ber_of_elements,tol,number_fn,ploton,d)
```

## Appendix G.9 Shape_Matrix.m

```
function
fn_matrix=shape_matrix(method,profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,nu
mber_of_elements,tol,number_fn,ploton)

%this matrix is designed for the shapes around 442Hz
%-----------------------------------------------------------------------------------------------------
%find the relationships between f1 Vs.d1, f2 Vs. d1, f3 Vs. d1:
d2=coeffs(2); d3=coeffs(3); k=1;
for d1=(coeffs(1)-1):0.1:(coeffs(1)+1)
   coeffs=[d1 d2 d3];
   Natural_Frequencies=eval([method
'(profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,number_of_elements,tol,number_f
n,ploton)']);
   f1(k)=Natural_Frequencies(1);
   f2(k)=Natural_Frequencies(2)*f1(k);
   f3(k)=Natural_Frequencies(3)*f1(k);
   k=k+1;
end;
x1=(coeffs(1)-1):0.1:(coeffs(1)+1);
p=polyfit(x1,f1,1); p2=polyfit(x1,f2,1); p3=polyfit(x1,f3,1);
a1_1=p(1);
a2_1=p2(1);
a3_1=p3(1);
%-----------------------------------------------------------------------------------------------------
coeffs=[9.162910623796002;0.262241312871511;0.005752111284758];
d1=coeffs(1); d3=coeffs(3); k=1;
for d2=(coeffs(2)-0.1):0.01:(coeffs(2)+0.1)
   coeffs=[d1 d2 d3];
   Natural_Frequencies=eval([method
'(profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,number_of_elements,tol,number_f
n,ploton)']);
   f1_2(k)=Natural_Frequencies(1);
   f2_2(k)=Natural_Frequencies(2)*f1_2(k);
   f3_2(k)=Natural_Frequencies(3)*f1_2(k);
   k=k+1;
end;
x2=(coeffs(2)-0.1):0.01:(coeffs(2)+0.1);
p2_1=polyfit(x2,f1_2,1); p2_2=polyfit(x2,f2_2,1); p2_3=polyfit(x2,f3_2,1);
a2_1_1=p2_1(1);
a2_2_1=p2_2(1);
a2_3_1=p2_3(1);
%-----------------------------------------------------------------------------------------------------
coeffs=[9.162910623796002;0.262241312871511;0.005752111284758]
d1=coeffs(1); d2=coeffs(2); k=1;
for d3=(coeffs(3)-0.001):0.0001:(coeffs(3)+0.001)
   coeffs=[d1 d2 d3];
   Natural_Frequencies=eval([method
'(profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,number_of_elements,tol,number_f
n,ploton)']);
   f1_3(k)=Natural_Frequencies(1);
   f2_3(k)=Natural_Frequencies(2)*f1_3(k);
   f3_3(k)=Natural_Frequencies(3)*f1_3(k);
   k=k+1;
end;
x3=(coeffs(3)-0.001):0.0001:(coeffs(3)+0.001);
p3_1=polyfit(x3,f1_3,1); p3_2=polyfit(x3,f2_3,1); p3_3=polyfit(x3,f3_3,1);
a3_1_1=p3_1(1);
a3_2_1=p3_2(1);
a3_3_1=p3_3(1);
%------------------------------------------------------------------------------------------------------------
fn_matrix=[a1_1 a2_1_1 a3_1_1; a2_1 a2_2_1 a3_2_1; a3_1 a2_3_1 a3_3_1];
%------------------------------------------------------------------------------------------------------------
```

## Appendix G.10 Method_Timo_TP.m

```
function
out=Method_Timo_TP(new_d,profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,num
ber_of_elements,tol,number_fn,ploton)

global b len ddd;
ddd=new_d;

%check size of steps in profile
delta_d_max=0;
for k=1:length(ddd)-1
   delta_d_max=max([delta_d_max abs((ddd(k)-ddd(k+1))/ddd(k))]);
end;

% Find the natural frequencies to this tolerance (rad/s)
tolerance=tol(1);
% Find up to only this number of frequencies
how_many=number_fn;

recept_all_timoshenko=[];  % place to store results by using Timoshenko
freq=[]; % place to store freqs

for w=logspace(2,5,600);  % nice spread of frequencies

   recept_timoshenko=beam_recept_timoshenko(w,E,rho,b,ddd,len,x_section,G,nu);
   recept_all_timoshenko=[recept_all_timoshenko recept_timoshenko(1,1)];

   freq=[freq w];  % record the freq in rad/s
end;

% Find the resonances - Timoshenko
bracket=bracket_natfreq(recept_all_timoshenko,freq,how_many);
if isempty(bracket)
   error('No natural frequencies found.  Perhaps alter frequency span?');
end;

nat_freqs=find_freq_timo_NN(ddd,bracket,tolerance);
fundamental_Timoshenko=nat_freqs./(2*pi);
%fundamental_Timoshenko=nat_freqs(1)./(2*pi);
%ratios_Timoshenko=nat_freqs./nat_freqs(1);

if ploton
   out=d;
else
   out=fundamental_Timoshenko;
   %out=[fundamental_Timoshenko;ratios_Timoshenko(2:how_many)];

end;
```

## Appendix G.11 find_freq_timo_TP.m

```
function nat_freqs=find_freq_timo_TP(newnewddd,bracket,tol)

% Given a span of frequencies known to contain the natural frequency, use the bisection
method to locate it within tolerance tol. % Note: bisection finds a root or a singularity.
 global b len E rho x_section G nu;

 ddd=newnewddd;

jmax=40;  % Maximum allowed number of bisections
nat_freqs=[];
[rows]=size(bracket);

if(rows==0)
    error('No frequencies bracketed?');
end;

for k=1:rows

    w_low=bracket(k,1);  %load the lower frequencies at the brackets found
    w_high=bracket(k,2); %load the higher frequencies at the brackets found

    %ymid is the receptance at the brackets with higher freuencies
    ymid=beam_recept_timoshenko(w_high,E,rho,b,ddd,len,x_section,G,nu);
    %y is the receptance at the brackets with lower frequencies
    y=beam_recept_timoshenko(w_low,E,rho,b,ddd,len,x_section,G,nu);

    %if the signs of the above two receptances are same, then error
    if(y.*ymid>=0)
        error('Roots must be bracketed for find_freq to work.');
    end;

    %if the receptance at the brackets with lower frequencies is negative,
    %then store the lower frequency in "estimate". on the other hand, if
    %the receptance at the brackets with lower frequencies is positive or
    %zero, then store the higher frequency in "estimate".
    if(y<0)
        estimate=w_low;
        dw=w_high-w_low; %delta frequency
    else
        estimate=w_high;
        dw=w_low-w_high; %delta frequency
    end;

    for j=1:jmax
        dw=dw.*0.5;

        %here "wmid" is the frequency in the middle of the lower frequency
        %and the higher frequency
        wmid=estimate+dw;

        %the receptance at the freuency of "wmid" can be found by sending
        %the frequency "wmid" to the "beam_recept_timoshenko" again.
        ymid=beam_recept_timoshenko(wmid,E,rho,b,ddd,len,x_section,G,nu);

        %if the receptance at the frequency of "wmid" is equal or smaller
        %than zero, then the frequency of "wmid" can be stored into
        %"estimate" insted of "w_low"
        if(ymid<=0);
           estimate=wmid;
        end;

        %if the magnitude of "dw" is smaller than the tolerance set in the
        %main function, then the frequency in "estimate" is recorded and
        %stored
        if(abs(dw)<tol)
           nat_freqs(k)=estimate;
        end;

        if(ymid==0)
           nat_freqs(k)=estimate;
        end;

    end;
end;

nat_freqs=nat_freqs';
```

## Appendix G.12 Shape_Reprediction.m

```
function
Shape_Reprediction(method,method_NN,profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_l
ength,number_of_elements,tol,number_fn,ploton,d)

% for the final cut, get the desired_fn
final_f1=330;final_f2=1320;final_f3=3234;
experimental=[final_f1; final_f2; final_f3];

%The following "desired_fn" is a parameter this program uses in the middle, users dont need
to know it, and wont know it during simulation
desired_fn=desired_vs_experimental(experimental)
%-------------------------------------------------------------------------------------------------------
ff1=desired_fn(1); ff2=desired_fn(2); ff3=desired_fn(3);
delta_f1=ff1-final_f1; delta_f2=ff2-final_f2; delta_f3=ff3-final_f3;
delta_ffn=[delta_f1; delta_f2; delta_f3];

delta_aa=[];
delta_aa=fn_matrix\delta_ffn;

%Here "new_aa" is the new generated shape parameters, and these parameters will give the
right shape of the beam relative to the user-wanted frequencies in reality
new_aa=coeffs+delta_aa

%now checking (ONly for testing)
ploton=false;
fffn=Method_Timoshenko(profile,new_aa,E,G,rho,x_section,nu,d0,width,bar_length,number
_of_elements,tol,number_fn,ploton)
ploton=true;
new_ddddd=Method_Timoshenko(profile,new_aa,E,G,rho,x_section,nu,d0,width,bar_length
,number_of_elements,tol,number_fn,ploton);

final_f1=330;final_f2=1320;final_f3=3234;
experimental=[final_f1; final_f2; final_f3];
desired_fn=desired_vs_experimental(experimental)
%-------------------------------------------------------------------------------------------------------
ff1=desired_fn(1); ff2=desired_fn(2); ff3=desired_fn(3);
delta_f1=ff1-final_f1; delta_f2=ff2-final_f2; delta_f3=ff3-final_f3;
delta_ffn=[delta_f1; delta_f2; delta_f3];

delta_aa=[];
delta_aa=fn_matrix\delta_ffn;
new_aa=coeffs+delta_aa

ploton=false;
fffn=Method_Timoshenko(profile,new_aa,E,G,rho,x_section,nu,d0,width,bar_length,number
_of_elements,tol,number_fn,ploton)
ploton=true;
new_ddddd=Method_Timoshenko(profile,new_aa,E,G,rho,x_section,nu,d0,width,bar_length
,number_of_elements,tol,number_fn,ploton);
```

## Appendix G.13 desired_vs_experimental.m

```
function desired_fn=desired_vs_experimental(experimental_fn)
experiments_1=experimental_fn(1);
experiments_2=experimental_fn(2);
experiments_3=experimental_fn(3);
%----------------------------------------------------------------------------------------------------------------
desired_fundamental=[1055 927.9 778.7 674.1];
experiment_fundamental=[1034 915 772 659];
desired_second=[2958.8 2719.3 2406.7 2187];
experiment_second=[2663 2478 2128 1969];
desired_third=[5742.6 5407 4937.4 4617.4];
experiment_third=[4580 4377 4104 3850];
%----------------------------------------------------------------------------------------------------------------
number_row=length(experiment_fundamental);
if number_row==2
    p_1=polyfit(experiment_fundamental,desired_fundamental,1);
    p_2=polyfit(experiment_second,desired_second,1);
    p_3=polyfit(experiment_third,desired_third,1);
    A=0;B=0;C=0;D=0;E=0;F=p_1(1);G=p_1(2);

elseif number_row==3
    p_1=polyfit(experiment_fundamental,desired_fundamental,2);
    p_2=polyfit(experiment_second,desired_second,2);
    p_3=polyfit(experiment_third,desired_third,2);
    A=0;B=0;C=0;D=0;E=p_1(1);F=p_1(2);G=p_1(3);

elseif number_row==4
    p_1=polyfit(experiment_fundamental,desired_fundamental,3);
    p_2=polyfit(experiment_second,desired_second,3);
    p_3=polyfit(experiment_third,desired_third,3);
    A_1=p_1(1);B_1=p_1(2);C_1=p_1(3);D_1=p_1(4);
    A_2=p_2(1);B_2=p_2(2);C_2=p_2(3);D_2=p_2(4);
    A_3=p_3(1);B_3=p_3(2);C_3=p_3(3);D_3=p_3(4);

elseif number_row==5
    p_1=polyfit(experiment_fundamental,desired_fundamental,4);
    p_2=polyfit(experiment_second,desired_second,4);
    p_3=polyfit(experiment_third,desired_third,4);
    A=0;B=0;C=p_1(1);D=p_1(2);E=p_1(3);F=p_1(4);G=p_1(5);

elseif number_row==6
    p_1=polyfit(experiment_fundamental,desired_fundamental,5);
    p_2=polyfit(experiment_second,desired_second,5);
    p_3=polyfit(experiment_third,desired_third,5);
    A=0;B=p_1(1);C=p_1(2);D=p_1(3);E=p_1(4);F=p_1(5);G=p_1(6);

elseif number_row==7
    p_1=polyfit(experiment_fundamental,desired_fundamental,6);
    p_2=polyfit(experiment_second,desired_second,6);
    p_3=polyfit(experiment_third,desired_third,6);
    A_1=p_1(1);B_1=p_1(2);C_1=p_1(3);D_1=p_1(4);E_1=p_1(5);F_1=p_1(6);G_1=p_1(7);
    A_2=p_2(1);B_2=p_2(2);C_2=p_2(3);D_2=p_2(4);E_2=p_2(5);F_2=p_2(6);G_2=p_2(7);
    A_3=p_3(1);B_3=p_3(2);C_3=p_3(3);D_3=p_3(4);E_3=p_3(5);F_3=p_3(6);G_3=p_3(7);
```

```
end;

desired_fundamental=A_1.*experiments_1.^3+B_1.*experiments_1.^2+C_1.*experiments_1+D_1;
desired_second=A_2.*experiments_2.^3+B_2.*experiments_2.^2+C_2.*experiments_2+D_2;
desired_third=A_3.*experiments_3.^3+B_3.*experiments_3.^2+C_3.*experiments_3+D_3;
desired_fn=[desired_fundamental;desired_second;desired_third];
```

## Appendix G.14 Toolpath_Generation.m

```
function
TOOLPATH_GENERATION(profile,coeffs,E,G,rho,x_section,nu,d0,width,bar_length,num
ber_of_elements,tol,number_fn,ploton,d)
%Here determine where the points start to change to find the undercut area
yaya=0;
for c=1:(number_of_elements-1)
   if (d(c) ~= d(c+1))
      yaya=yaya+1;
      where(yaya)=c;
   end;
end;
fixed_position=where(1);%??????????????????????????????????????????????????
last_fixed_position=(number_of_elements-fixed_position)+1;%????????????????

%Now I will get the new shape
New_d=[];
for k=1:fixed_position-1%???????????????????????????????????????????????????
   New_d(k)=d(k);
end;
for k=(last_fixed_position+1):number_of_elements%??????????????????????????
   New_d(k)=d(k);
end;
lala=0;
%The following gives the undercut points, and then saved into "New_ddd"
%for k=(fixed_position):(last_fixed_position)%??????????????????????????????
% lala=lala+1;
% New_ddd(lala)=d(k);
%end;
lalala=0;
cut_perc=1
   lala=0;
   lalala=lalala+1;
   %length_all_the_time=((fixed_position)*(bar_length/600):(bar_length/600):(last_fixed_p
osition)*(bar_length/600));
   %Here moved "d" curve upper by "1/10"
   for k=(fixed_position):(last_fixed_position)
      lala=lala+1;
      New_ddd(lala)=((d0-d(k))*(1-cut_perc))+d(k)+0.000025;
   end;

   lala=0;
   for k=(fixed_position):(last_fixed_position)
      lala=lala+1;
      New_d(k)=New_ddd(lala);
   end;

%----------------------------------------------------------------------------------------------------
%the number of points should be changed after the cutting beam length is
%decided, "points" is the number of points for undercut area
points=length(New_ddd);

line_number=3;
```

```
for line=1:line_number
   for i=1:points
      %Here the value for X acis CC points should be updated after
      %getting the CNC machine working.(The coordinates should be different)
      CC_X(i,line)=((fixed_position+i-1).*bar_length./number_of_elements);
      CC_Y(i,line)=(width/(line_number-1)).*(line-1);
      CC_Z(i,line)=New_ddd(i);
   end;
end;
figure(3)
mesh(CC_X,CC_Y,CC_Z);
%---------------------------------------------------------------------------------------------------------------
%CC to CL for side step
[surf_x_normal,surf_y_normal,surf_z_normal]=surfnorm(CC_X,CC_Y,CC_Z);
%toolpath CL points %still need interpolation
R=0.005;%???????????????????????????????????????????????????????????????
PCL_X=CC_X-(R.*surf_x_normal);
PCL_Y=CC_Y-(R.*surf_y_normal);
PCL_Z=CC_Z-(R.*surf_z_normal);

%now input length_per_motor_step, this is the property of stepper
%motor, and should be checked latter, it shows the minimum length the
%stepper motor is capable of moving
length_per_motor_step=0.000002532;
%length_per_motor_step=0.0000015875;%?????????????????????????????????????
% if the stepper motor only moves at its minimum length, it
%requires too much of movements, then set each move to be 100 stepper
%motor minimum length. Here the program should keep trying different values
%so that it can ensure the accuracy is good AND does not take too much time
s=400*length_per_motor_step;%????????????????????????????????????????????
%The following shows how many cuts aer required for the undercut area based
%on the motor increments set above
segment_double=(length(PCL_X(:,1)).*(bar_length./number_of_elements))/s;
% make the obtained segment number into integer
NN=0;
for nn=1:(number_of_elements*10)
   if nn>segment_double
      NN=NN+1;
      segment_int(NN)=nn;
   end;
end;
HOW_MANY_SEGMENT=segment_int(1);
%---------------------------------------------------------------------------------------------------------------
%Once the segment number is changed to integer, the value is slightly
%larger than the original "double" value, thus, the motor increment is no
%longer "s", it is slightly smaller
tt_single_length=(PCL_X(length(PCL_X(:,1)),1)-PCL_X(1,1))/HOW_MANY_SEGMENT;
iii=0;
for x_direct=PCL_X(1,1):tt_single_length:PCL_X(length(PCL_X(:,1)),1)
   iii=iii+1;
   x_direction(iii,1)=x_direct;
   Z_direction(iii,1)=interp1(PCL_X(:,1),PCL_Z(:,1),x_direct);
end;
%---------------------------------------------------------------------------------------------------------------
%forward Step
```

```
column_number=length(x_direction(iii,1));
%Now I will form all CL points into three matrices
row_number=length(x_direction(:,1));
%-------------------------------------------------------------------------------------------------------------
CL_X=zeros(row_number,column_number);
CL_Y=zeros(row_number,column_number);
CL_Z=zeros(row_number,column_number);
%Here x and z axes CL points are saved into relative array
for column=1:column_number
   for row=1:row_number
      CL_X(row,column)=x_direction(row,1);
      CL_Z(row,column)=Z_direction(row,1);
   end;
end;
%-------------------------------------------------------------------------------------------------------------
side_cut=0;
side_cut_number=row_number;
sideside=side_cut_number/2;
shuangshu=false;
for nn=1:100
   if nn==sideside
      shuangshu=true;
   end;
end;

if shuangshu %Can be divided by "2"
   for side_CO=1:2:side_cut_number-1
      side_cut_X(1,side_CO)=CL_X(side_CO,1);
      side_cut_Y(1,side_CO)=0;
      side_cut_Z(1,side_CO)=CL_Z(side_CO,1);
      side_cut_X(2,side_CO)=CL_X(side_CO,1);
      side_cut_Y(2,side_CO)=width;
      side_cut_Z(2,side_CO)=CL_Z(side_CO,1);
   end;
   for side_CO=2:2:side_cut_number
      side_cut_X(1,side_CO)=CL_X(side_CO,1);
      side_cut_Y(1,side_CO)=width;
      side_cut_Z(1,side_CO)=CL_Z(side_CO,1);
      side_cut_X(2,side_CO)=CL_X(side_CO,1);
      side_cut_Y(2,side_CO)=0;
      side_cut_Z(2,side_CO)=CL_Z(side_CO,1);
   end;
else %Cant be divided by "2"
   for side_CO=1:2:side_cut_number
      side_cut_X(1,side_CO)=CL_X(side_CO,1);
      side_cut_Y(1,side_CO)=0;
      side_cut_Z(1,side_CO)=CL_Z(side_CO,1);
      side_cut_X(2,side_CO)=CL_X(side_CO,1);
      side_cut_Y(2,side_CO)=width;
      side_cut_Z(2,side_CO)=CL_Z(side_CO,1);
   end;
   for side_CO=2:2:side_cut_number-1
      side_cut_X(1,side_CO)=CL_X(side_CO,1);
      side_cut_Y(1,side_CO)=width;
      side_cut_Z(1,side_CO)=CL_Z(side_CO,1);
      side_cut_X(2,side_CO)=CL_X(side_CO,1);
      side_cut_Y(2,side_CO)=0;
      side_cut_Z(2,side_CO)=CL_Z(side_CO,1);
   end;
end;
%----------------------------------------------------------------------------------------------------
forward_cut_number=length(side_cut_X(1,:));%?????????????????????????????
forward=forward_cut_number/2; forward_shuangshu=false;
for nnn=1:100
   if nnn==forward
      forward_shuangshu=true;
   end;
end;

SIDE_CUT_NUM=length(side_cut_X(1,:));
FORWARD_CUT_NUM=length(side_cut_Z(1,:));
TOTAL_CUT_NUM=0; TOTAL_SIDE=0;
for FORWARD_STEP=1:SIDE_CUT_NUM
      TOTAL_CUT_NUM=TOTAL_CUT_NUM+1;
      TOOL_CUT_X_TEMP(TOTAL_CUT_NUM)=side_cut_X(1,FORWARD_STEP);
      TOOL_CUT_Y_TEMP(TOTAL_CUT_NUM)=side_cut_Y(1,FORWARD_STEP);
      TOOL_CUT_Z_TEMP(TOTAL_CUT_NUM)=side_cut_Z(1,FORWARD_STEP);
      TOTAL_CUT_NUM=TOTAL_CUT_NUM+1;
      TOOL_CUT_X_TEMP(TOTAL_CUT_NUM)=side_cut_X(2,FORWARD_STEP);
      TOOL_CUT_Y_TEMP(TOTAL_CUT_NUM)=side_cut_Y(2,FORWARD_STEP);
      TOOL_CUT_Z_TEMP(TOTAL_CUT_NUM)=side_cut_Z(2,FORWARD_STEP);
end;
for FORWARD_STEP=1:TOTAL_CUT_NUM+1
   if FORWARD_STEP==TOTAL_CUT_NUM+1
      TOOL_CUT_X(FORWARD_STEP)=TOOL_CUT_X_TEMP(1);
      TOOL_CUT_Y(FORWARD_STEP)=TOOL_CUT_Y_TEMP(1);
      TOOL_CUT_Z(FORWARD_STEP)=TOOL_CUT_Z_TEMP(1);
   else
      TOOL_CUT_X(FORWARD_STEP)=TOOL_CUT_X_TEMP(FORWARD_STEP);
      TOOL_CUT_Y(FORWARD_STEP)=TOOL_CUT_Y_TEMP(FORWARD_STEP);
      TOOL_CUT_Z(FORWARD_STEP)=TOOL_CUT_Z_TEMP(FORWARD_STEP);
   end;
end;

figure(5)
plot3(TOOL_CUT_X,TOOL_CUT_Y,TOOL_CUT_Z)
%----------------------------------------------------------------------------------------------------
%Coordinates_to_steps(TOOL_CUT_X,TOOL_CUT_Y,TOOL_CUT_Z,TOTAL_CUT_NUM,length_per_motor_step)
%----------------------------------------------------------------------------------------------------
%G_Code_Generation(TOOL_CUT_X,TOOL_CUT_Y,TOOL_CUT_Z,TOTAL_CUT_NUM)
%----------------------------------------------------------------------------------------------------
```

## Appendix G.15 Coordinates_to_steps.m

```
function
Coordinates_to_steps(TOOL_CUT_X,TOOL_CUT_Y,TOOL_CUT_Z,TOTAL_CUT_NUM
,meter_per_microstep)
%X Axis
%Imagine the cutter has already been placed at the edge of the undercut area, which is the
first point of undercut shape, and the first move will go towards to the second point of the
undercut shape
for cutting=1:TOTAL_CUT_NUM-1
  %The following gives the different between two undercut points in length (coordinates
difference)
  step_signal_X=abs(TOOL_CUT_X(cutting+1)-TOOL_CUT_X(cutting));
  %The following gives the number of motor steps for the above difference, it wont be
integer, but motor can move "double" type of steps, then I will make the steps integers after
this
  step_signal_X_double=step_signal_X./meter_per_microstep;
  www=0;
  for iiiii=0:50000
     if iiiii >= step_signal_X_double
        www=www+1;
        step_signal_X_temp(www)=iiiii;
     end;
  end;
  step_signal_X_int(cutting)=step_signal_X_temp(1); %Now integer steps

  %the step number for this movies known, but the direction is not known yet, thus, make
positive direction if the next point is larger than the previous one, vise versa.
  if TOOL_CUT_X(cutting+1)>=TOOL_CUT_X(cutting)
    direction_X(cutting)=1;
  else
    direction_X(cutting)=0;
  end;

  %The above process will go on until all the moves are translated into
  %steps and directions.
end;
%-------------------------------------------------------------------------------------------------------
%Y Axis
%For Y axis, the cutter only moves linearly with fixed travelling length, so I only need to
find one travelling length and use it for all the moves, but the direction of travelling will be
different
step_signal_Y_2_1=(abs(TOOL_CUT_Y(2)-TOOL_CUT_Y(1)))./meter_per_microstep;

uuu=0;
for iiiii=1:50000
  if iiiii >= step_signal_Y_2_1
     uuu=uuu+1;
     step_signal_Y_second(uuu)=iiiii;
  end;
end;
step_signal_Y_2_1_int=step_signal_Y_second(1);

for cutting=1:TOTAL_CUT_NUM-1
```

```
    %The magnitude is always same, so put "absolute" here
    step_signal_Y_next=abs(TOOL_CUT_Y(cutting+1)-TOOL_CUT_Y(cutting));
    if step_signal_Y_next ~=0
        step_signal_Y_int(cutting)=step_signal_Y_2_1_int;
    else
        step_signal_Y_int(cutting)=0;
    end;

    if TOOL_CUT_Y(cutting+1)>=TOOL_CUT_Y(cutting)
      direction_Y(cutting)=1;
    else
      direction_Y(cutting)=0;
    end;
end;
%-----------------------------------------------------------------------------------------------------------
%Z Axis
for cutting=1:TOTAL_CUT_NUM-1
    www=0;
    step_signal_Z_next=abs(TOOL_CUT_Z(cutting+1)-TOOL_CUT_Z(cutting));
    step_signal_Z_double=step_signal_Z_next./meter_per_microstep;
    for iiiii=0:10000
        if iiiii >= step_signal_Z_double
            www=www+1;
            step_signal_Z_temp(www)=iiiii;
        end;
    end;
    step_signal_Z_int(cutting)=step_signal_Z_temp(1);

    if TOOL_CUT_Z(cutting+1)>=TOOL_CUT_Z(cutting)
      direction_Z(cutting)=1;
    else
      direction_Z(cutting)=0;
    end;
end;
%-----------------------------------------------------------------------------------------------------------
%Imagine the cutter is already at the first point on the undercut area, thus, the number of
moves is one less than the number of undercut points
for cutting=1:TOTAL_CUT_NUM-1
     STEP_SIGNAL_X(cutting)=step_signal_X_int(cutting);
     DIRECTION_SIGNAL_X(cutting)=direction_X(cutting);
     STEP_SIGNAL_Y(cutting)=step_signal_Y_int(cutting);
     DIRECTION_SIGNAL_Y(cutting)=direction_Y(cutting);
     STEP_SIGNAL_Z(cutting)=step_signal_Z_int(cutting);
     DIRECTION_SIGNAL_Z(cutting)=direction_Z(cutting);
end;

steps_to_pulses(TOTAL_CUT_NUM,STEP_SIGNAL_X,STEP_SIGNAL_Y,STEP_SIGNA
L_Z,DIRECTION_SIGNAL_X,DIRECTION_SIGNAL_Y,DIRECTION_SIGNAL_Z)
```

### Appendix G.16 Steps_to_pulses.m

```
function
steps_to_pulses(TOTAL_CUT_NUM,STEP_SIGNAL_X,STEP_SIGNAL_Y,STEP_SIGNA
L_Z,DIRECTION_SIGNAL_X,DIRECTION_SIGNAL_Y,DIRECTION_SIGNAL_Z)

for cutting=1:TOTAL_CUT_NUM-1 %each cut has different motor steps
   %The number of pulses is always twice of the number of the steps for each move, thus,
simply multiply "2" for x and z axes
   pul_numnum_x=2*STEP_SIGNAL_X(cutting);
   pul_numnum_z=2*STEP_SIGNAL_Z(cutting);

   %The following is to make x and z pulses placed on the right positions
   just_one_step=0;
   %If there is no pulses on x axis, it means there is also no pulses on z axis, thus, the space
between x and z axes is zero
   if pul_numnum_x==0
      pul_space=0;
   %If the number of the pulses on x axis is non-zero, it means both the x  and z axes move
for this move, and then the space between these two pulse trains should be determined
   else
      %the following shows the times between the number of x pulses and the z pulses
      pul_space=(pul_numnum_x)./(pul_numnum_z);
%If the "times" is even larger than or equal to the number of x pulses, it means z axis hasn’t
moved yet when the x-axis has finished moving. In this case I will manually make z axis
move one step instead
      if pul_space>=pul_numnum_x
         just_one_step=1;
      end;
   end;%End of "if pul_numnum_x==0"

   %Here for this move, check whether the z-axis only moves one step or not
   if just_one_step==0
      for tim=1:pul_numnum_z
         %The first z pulse is at the same position as the first x pulse
         if tim==1
            zzzzz(tim)=tim;
         %the last z pulse is at the same position as the last x pulse
         elseif tim==pul_numnum_z
            zzzzz(tim)=pul_numnum_z;
         %For the pulses in the middle, they should multiply the space, and saved into the
"integer" positions
         else
            z_on_x(tim)=tim.*pul_space;
            uuu=0;
            for iiiiii=0:300
               if iiiiii >= z_on_x(tim)
                  uuu=uuu+1;
                  pul_space_temp(uuu)=iiiiii;
               end;
            end;
            pul_space_2=pul_space_temp(1);
            if pul_space_2>pul_numnum_x
               pul_space_2=pul_space_2-1;
            end;
            zzzzz(tim)=pul_space_2;
          end;
        end;%End of "for tim=1:pul_numnum_z"

        for real_pul=1:pul_numnum_z
          if real_pul==1
            zzzzzzzz(cutting,real_pul)=1;
          elseif real_pul==pul_numnum_z
            zzzzzzzz(cutting,real_pul)=pul_numnum_x;
          else
            zzzzzzzz(cutting,real_pul)=zzzzz(real_pul);
          end;
        end;%End of "for real_pul=1:pul_numnum_z"

      else %here is for "only one step on Z axis"
        zzzzzzzz(cutting,1)=1;
        zzzzzzzz(cutting,2)=pul_numnum_x;
      end;%End of "if just_one_step==0"

    end;%End of "for cutting=1:TOTAL_CUT_NUM-1"
    %-----------------------------------------------------------------------------------------------------------
    for cutting=1:TOTAL_CUT_NUM-1 %each cut has different motor steps
      pul_numnum_x(cutting)=2*STEP_SIGNAL_X(cutting);
      pul_numnum_z(cutting)=2*STEP_SIGNAL_Z(cutting);
      pul_numnum_y(cutting)=2*STEP_SIGNAL_Y(cutting);%for Y axis

      % make the first pulse to be "1"
      step_one=1;step_two=0;
      step_one_z=1;step_two_z=0;
      step_one_y=1;step_two_y=0;

      step_todd=step_two;
      step_todd_z=step_two_z;
      step_todd_y=step_two_y;

      pulse_x_change=1;
      cris=1;

      if pul_numnum_x(cutting)~=0 %Here X and Z axes move
        %-------------------------------------------------------------------------------------------------------
        for pul_pul=1:pul_numnum_x(cutting)%for example:198 pulses
          %Here is for X axis pulses
          if pulse_x_change==1
            if step_todd==step_one
              step_todd=step_two;
            else
              step_todd=step_one;
            end;%End of "if step_todd==step_one"
            x_pulse(cutting,pul_pul)=step_todd;
          else
            if step_todd==step_one
              step_todd=step_one;
            else
              step_todd=step_two;
            end;
            x_pulse(cutting,pul_pul)=step_todd;
         end;%End of "pulse_x_change==1"
         %----------------------------------------------------
         %Here is for Z axis pulses
         if pul_pul==zzzzzzzz(cutting,cris)
            cris=cris+1;
            pulse_z_change=1;
         else
            pulse_z_change=0;
         end;%End of "if pul_pul==zzzzzzzz(cutting,cris)"
         if pulse_z_change==1
            if step_todd_z==step_one_z
               step_todd_z=step_two_z;
            else
               step_todd_z=step_one_z;
            end;
            z_pulse(cutting,pul_pul)=step_todd_z;
         else
            if step_todd_z==step_one_z
               step_todd_z=step_one_z;
            else
               step_todd_z=step_two_z;
            end;
            z_pulse(cutting,pul_pul)=step_todd_z;
         end;%End of "if pulse_z_change==1"
         %----------------------------------------------------
         %Here is for Y axis pulses
         y_pulse(cutting,pul_pul)=0;
         %----------------------------------------------------

      end;%End of "pul_pul=1:pul_numnum_x(cutting)"
      %-------------------------------------------------------

   else %Here only Y axis moves
      %-------------------------------------------------------
      %Here is for X axis pulses
      x_pulse(cutting,1)=0; x_pulse(cutting,2)=0;
      %-------------------------------------------------------
      %Here is for Z axis pulses
      z_pulse(cutting,1)=0; z_pulse(cutting,2)=0;
      %-------------------------------------------------------
      %Here is for Y axis pulses
      y_pulse(cutting,1)=1; y_pulse(cutting,2)=0;
      %-------------------------------------------------------
   end;

end;%End of "for cutting=1:TOTAL_CUT_NUM-1"
%------------------------------------------------------------------------------------------------------------
Driving_motors(STEP_SIGNAL_X,STEP_SIGNAL_Y,STEP_SIGNAL_Z,x_pulse,z_pulse,y_pulse,TOTAL_CUT_NUM,DIRECTION_SIGNAL_X,DIRECTION_SIGNAL_Z,DIRECTION_SIGNAL_Y)
```

## Appendix G.17 Driving_motors.m

```
function
Driving_motors(STEP_SIGNAL_X,STEP_SIGNAL_Y,STEP_SIGNAL_Z,x_pulse,z_pulse,
y_pulse,TOTAL_CUT_NUM,DIRECTION_SIGNAL_X,DIRECTION_SIGNAL_Z,DIREC
TION_SIGNAL_Y)

parport=digitalio('parallel','LPT1');
line=addline(parport,0:5,0,'out');
for cutting=1:TOTAL_CUT_NUM-1
   pul_numnum_x(cutting)=2*STEP_SIGNAL_X(cutting);
   pul_numnum_z(cutting)=2*STEP_SIGNAL_Z(cutting);
   pul_numnum_y(cutting)=2*STEP_SIGNAL_Y(cutting);
end;
for cutting=1:TOTAL_CUT_NUM-1
   if STEP_SIGNAL_X(cutting)~=0
      for g=1:pul_numnum_x(cutting)
         putvalue(parport,[x_pulse(cutting,g)    DIRECTION_SIGNAL_X(cutting)    0    0
z_pulse(cutting,g) DIRECTION_SIGNAL_Z(cutting)]);
      end;
   else
      for g=1:STEP_SIGNAL_Y(cutting)
         putvalue(parport,[0 0 y_pulse(cutting,1) DIRECTION_SIGNAL_Y(cutting) 0 0]);
         putvalue(parport,[0 0 y_pulse(cutting,2) DIRECTION_SIGNAL_Y(cutting) 0 0]);
      end;
   end;
end;
gval = getvalue(parport);
delete(parport);
clear parport;
```

## Appendix G.18 G_Code_Generation.m

```
function G_Code_Generation(d, len, bar_length)

fid=fopen('piece2part1.txt','w');
fprintf(fid,'START\n');
fprintf(fid,'N10 \n');
fprintf(fid,'N20 G90;\n');
fprintf(fid,'N30 G40 G80;\n');
fprintf(fid,'N40 M06 T10;\n');
fprintf(fid,'N50 M03 S1800;\n');

fprintf(fid,'N60 G01 X%8.3f.Y%7.3f;\n',0,((0.019-d(1))*1000));
ffirst_step=len(1)/2;
fprintf(fid,'N70 G01 X%8.3f.Y%7.3f;\n',(ffirst_step*1000),((0.019-d(1))*1000));
for Go_X=2:length(d)
        fprintf(fid,'N%u                         G01                         X%7.3f.Y%8.3f;\n',(70+10*(Go_X-1)),((ffirst_step+(len(Go_X)*(Go_X-1)))*1000),((0.019-d(Go_X))*1000));
end;
fprintf(fid,'N%u       G01       X%8.3f.Y%7.3f;\n',(70+10*(Go_X)),(bar_length*1000),((0.019-d(Go_X))*1000));

 fclose(fid);
```