ENHANCED MEASUREMENTS IN FOURIER ANALYSIS OF MEMS DYNAMICS

by

Mehrdad Mottaghi

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

Dalhousie University
Halifax, Nova Scotia
June 2012

© Copyright by Mehrdad Mottaghi, 2012

DALHOUSIE UNIVERSITY

DEPARTMENT OF MECHANICAL ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled “ENHANCED MEASUREMENTS IN FOURIER ANALYSIS OF MEMS DYNAMICS” by Mehrdad Mottaghi in partial fulfilment of the requirements for the degree of Master of Applied Science.

Dated: June 14, 2012

Supervisors: _____

_____

Readers: _____

_____

DALHOUSIE UNIVERSITY

DATE: June 14, 2012

AUTHOR: Mehrdad Mottaghi

TITLE: ENHANCED MEASUREMENTS IN FOURIER ANALYSIS OF MEMS DYNAMICS

DEPARTMENT OR SCHOOL: Department of Mechanical Engineering

DEGREE: MASc CONVOCATION: October YEAR: 2012

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions. I understand that my thesis will be electronically available to the public.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the thesis (other than the brief excerpts requiring only proper acknowledgement in scholarly writing), and that all such use is clearly acknowledged.

_____

Signature of Author

Table of Contents

List of Tables ..... **vii**

List of Figures ..... **viii**

Abstract ..... **xii**

List of Abbereviatios and Symbols Used ..... **xiii**

Acknowledgements ..... **xv**

Chapter 1: Introduction **1**

1.1: Objective 1

1.2: Thesis Scope 1

1.3: Author Contribution 1

1.4: Introduction to MEMS 2

1.4.1: PolyMUMPs 3

1.5: Dynamic Measurement of MEMS Devices 5

1.5.1: Optical Methods 5

1.5.2: Blur Methods 6

Chapter 2: Fourier Analysis **7**

2.1: Introduction 7

2.2: Fourier Transform 7

2.3: Fourier Series 7

2.4: Discrete Fourier Transform 10

2.5: Fast Fourier Transform 11

Chapter 3: FFT Analysis of Blurred Curve **13**

3.1: Overview 13

3.2: Definitions 13

3.2.1: FFT Harmonics 13

3.2.2: Bessel Function 16

3.3: Mathematical Modeling of Blur 17

3.4: Numerical Simulation 19

Chapter 4: 2D Image FFT Blur Extraction 22

4.1: Overview 22

4.2: Row Averaging of 2D Image 25

4.3: Calculation of FFT Harmonics 26

4.4: Detection of Resonance Frequency 27

4.5: Bessel Curve Fitting of Numerical Values 29

4.6: Conversion to Displacement 30

Chapter 5: Macro Scale Tests 31

5.1: Experimental Setup 31

5.2: Changing Parameters 33

5.2.1: Number of Columns 33

5.2.2: Columns Height 36

5.2.3: Columns Width 37

5.2.4: Reference Comb 39

5.2.5: Duty Cycle 41

5.3: Summary, Recommendations and Analyses 44

5.3.1: Summary 44

5.3.2: Recommendations and Analyses 45

Chapter 6: Micro Scale Tests 46

6.1: Experimental Setup....46
6.2: FIB Instrument....50
6.2.1: Overview....50
6.2.2: Creating Etching Patterns....55
6.3: Etching Extra Patterns....58
6.4: Software Modifications....64
6.5: Summary, Recommendations and Analyses....66
Chapter 7: Simualtion Tests....67
7.1: Adding Noise....67
7.2: Adding Darkness....69
7.3: Summary, Recommendations and Analyses....71
7.3.1: Summary....71
7.3.2: Recommendations and Analyses....72
Chapter 8: Color FFT....73
8.1: Summary, Recommendations and Analyses....78
Chapter 9: Summary and Conclusion....79
References....82
Appendix A: LabVIEW Program....86
Appendix B: MATLAB Codes....88
B.1: Adding Noise....88
B.2: Changing Brightness....92
B.3: Original FFT Analysis Program....96
B.4: Averaging CSV of Multiple Runs....107

List of Tables

TABLE 1: MECHANICAL AND ELECTRICAL PARAMETERS OF POLYMUMPS LAYERS [4] 4

TABLE 2: SUMMARY OF MACRO SCALE MODEL TEST RESULTS 44

TABLE 3: DEFAULT BEAM SETTINGS ON FIB SYSTEM 52

TABLE 4: APPROXIMATE ETCHING SPEED (μM/HR) FOR DIFFERENT BEAMS AND AREAS 53

TABLE 5: SCANNING AREA, ZOOM LEVEL AND CORRESPONDING RESOLUTIONS 56

TABLE 6: SUMMARY OF MICRO SCALE MODEL TEST RESULTS 66

TABLE 7: SUMMARY OF SIMULATION TEST RESULTS FOR 25X2X1 MEMS PAD 71

List of Figures

Figure 1: Illustration of PolyMUMPs Layers and Their Thickness [4] 4

Figure 2: Approximation of Square Wave by Odd Components of Its Fourier Series [12]: (a) 1, (b) 1-3-5, (c) 1-3-5-7-9, (d) 1-3-5-7-9-13 9

Figure 3: Plot of Discrete Fourier Transform for a Sine Function with Frequency of *20 Hz* 10

Figure 4: First Four Harmonics of a Sine Wave with Frequencies of 1/π (blue), 2/π (red), 4/π (black) and 8/π (green). All Harmonics have Same Amplitude 14

Figure 5: Trapezoid Curve with Different Number of Fourier Series Components (Right) and Associated Normalized FFT Harmonics (Left). (a) 1, (b) 1-3, (c) 1-3-5, (d) 1-3-5-7 15

Figure 6: Plot of Bessel Function of First Kind Zero Order 17

Figure 7: Discrete Fourier Approximation of Original and Blurred Trapezoid Wave Form. Fourier Approximation is done with 13 Components (N=13) and 256 Sample Points (n=256) 20

Figure 8: Magnitude of Normalized FFT Harmonics for Blurred and Original Trapezoid Wave Form Indicated with Lines and Squares Respectively 21

Figure 9: Fit Between Numerical Values of Harmonics (Squares) and |Bessel Curve| . The Pairs of Numbers Represent Amount of the Blur Phase and the Attenuation. 21

Figure 10: Static (Left), Slightly Blurred (Middle) and Severely Blurred (Right) Combs. Blur was simulated with MATLAB 22

Figure 11: FFT Blur Image Analysis Process (I) 23

Figure 12: FFT Blur Image Analysis process (II) 24

Figure 13: Image of 2D Static Comb and Plot of Its Row Average 25

Figure 14: Image of 2D Blurred Comb and Plot of Its Row Average 25

Figure 15: Magnitudes of FFT Harmonics for Static Comb 26

Figure 16: Magnitudes of FFT Harmonics for Blurred Comb 26

Figure 17: Magnitude of FFT Harmonics vs. Frequency 27

Figure 18: Normalized Attenuation of FFT Harmonics vs. Frequency 28

FIGURE 19: PLOT OF ATTENUATION VS. HARMONICS FOR THREE FREQUENCIES: 93.8 KHZ, 94.2 KHZ AND 91.8 KHZ. DOTS ARE EXPERIMENTAL DATA AND LINES SHOW BEST FIT FOR BESSEL CURVE ABSOLUTE VALUE 29
FIGURE 20: PLOT OF DISPLACEMENT VS. FREQUENCY (SNR=32) 30
FIGURE 21: FLOWCHART SHOWING CONFIGURATION OF MACRO SCALE TEST SETUP 31
FIGURE 22: PART OF MACRO SCALE SETUP INCLUDING: PAPER COMB AND ITS HOLDER, VIBRATOR, AMPLIFIER AND CCD CAMERA CONNECTED TO PC VIA FIREWIRE CABLE 32
FIGURE 23: SCREEN SHOT OF VI PROGRAM CREATED WITH LABVIEW FOR SIMULTANEOUS CONTROL OF FUNCTION GENERATOR AND CCD CAMERA 33
FIGURE 24: REGION OF INTERESTS CONTAINING 2, 4, 6 AND 8 COLUMNS 34
FIGURE 25: SNR VALUES BASED ON NUMBER OF SELECTED COLUMNS FROM 2-8C (COLUMNS) 34
FIGURE 26: PLOT OF DISPLACEMENT VS. FREQUENCY FOR SELECTION OF TWO COLUMNS (SNR=11.8) 35
FIGURE 27: PLOT OF DISPLACEMENT VS. FREQUENCY FOR SELECTION OF EIGHT COLUMNS (SNR=21.1) 35
FIGURE 28: REGION OF INTERESTS WITH DIFFERENT COMB LENGTHS AS PERCENTAGE OF MAXIMUM SELECTED LENGTH: 5%, 10%, 25%, 50%, 75% AND 100% 36
FIGURE 29: SNR READINGS BASED ON PERCENTAGE OF COLUMNS MAXIMUM SELECTED LENGTH 37
FIGURE 30: PAPER COMBS WITH DIFFERENT COLUMN WIDTHS: 1, 2, 4 AND 6 MILLIMETERS 38
FIGURE 31: SNR VALUES FOR EACH COLUMN WIDTH 39
FIGURE 32: CLOSE VIEW OF REFERENCE COMB SETUP ON VIBRATION ISOLATION TABLE 40
FIGURE 33: MOVING COMB (RIGHT) ALONG WITH STATIC REFERENCE COMB (LEFT) BOTH WITH 4 MM COLUMNS 40
FIGURE 34: PLOT OF DISPLACEMENT VS. FREQUENCY FOR REFERENCE COMB SETUP 41
FIGURE 35: FOUR PAPER COMBS WITH 4MM COLUMN WIDTH AND 25%, 37.5%, 62.5%, 75% DUTY CYCLE 42
FIGURE 36: ROW AVERAGE OF PAPER COMB WITH 25% DUTY CYCLE 42
FIGURE 37: ROW AVERAGE OF PAPER COMB WITH 75% DUTY CYCLE 43
FIGURE 38: SNR VALUES ASSOCIATED WITH DIFFERENT DUTY CYCLES 43
FIGURE 39: FLOWCHART SHOWING CONFIGURATION OF MICRO SCALE TEST SETUP 46
FIGURE 40: PART OF MICRO SCALE SETUP INCLUDING: WENTWORTH PROBE STATION, ZIF SOCKET WITH INSTALLED MEMS CHIP, OPTICAL MICROSCOPE, AND FIREWIRE CCD CAMERA 47

Figure 41: Part of Micro Scale Setup Including: Oscilloscope, Function Generator, Amplifier with its Power Supply and Distribution Box 48
Figure 42: Micrograph of MEMS XY Pad Including: Central Mass, Paper Clip and S Type Springs, Anchors and Chevron Thermal Actuators 49
Figure 43: Micrograph of Central Mass in Static Mode (Left) and at Resonance at 94 kHz (Right) 49
Figure 44: Plot of Displacement vs. Frequency for Micro Scale Model. Resonance Frequency occurs at 94 kHz 50
Figure 45: Front View of Focused Ion Beam (FIB) Instrument Used for Etching [25] 51
Figure 46: FIB Instrument Software User Interface: (a) Fabrication Window, (b) Viewing Window, (c) Fabrication Setup, (d) Column Adjustment 54
Figure 47: Side and Top View of MEMS Chip FIB Holder Made of ZIF Socket 55
Figure 48: Sample of Created Etching Pattern. Boundaries Represent 512×512 Pixels 57
Figure 49: Resultant MEMS Pad Using above Created Etching Pattern (Figure 48) 57
Figure 50: Micrograph of Etched MEMS Structures with Four Different Configuration: (a) 25x1x1, (b) 25x2x1, (c) 25x2x2 and (d) 25x4x1. All Dimensions are in Microns 58
Figure 51: Illustration of Upper and Lower Selection Area in Etched MEMS Pads 59
Figure 52: Row Average of MEMS Pad Upper Section (Plain) 60
Figure 53: Row Average of MEMS Pad Lower Section (25x1x1) 60
Figure 54: Row Average of MEMS Pad Lower Section (25x2x1) 61
Figure 55: Row Average of MEMS Pad Lower Section (25x2x2) 61
Figure 56: Row Average of MEMS Pad Lower Section (25x4x1) 62
Figure 57: Plot of Displacement vs. Frequency for Upper Section of 25×2×2 Pattern (SNR=8.3) 62
Figure 58: Plot of Displacement vs. Frequency for Lower Section of 25×2×2 Patten (SNR=15.4) 63
Figure 59: SNR Values for Four Different Patterns Based on Selected Area: Plain Section or Upper Area (U), Etched Section or Lower Area (L) 63
Figure 60: Original (Left) and Enhanced (Right) Images of MEMS Pads 64

FIGURE 61: COMPARISON OF SNR VALUES FOR DIFFERENT ETCHING PATTERNS BEFORE AND AFTER IMAGE ADJUSTMENT WHEN UPPER AREA (PLAIN) IS SELECTED 65
FIGURE 62: COMPARISON OF SNR VALUES FOR DIFFERENT ETCHING PATTERNS BEFORE AND AFTER IMAGE ADJUSTMENT WHEN LOWER AREA (ETCHED) IS SELECTED 65
FIGURE 63: MEMS PAD WITH ADDED ARTIFICIAL NOISE BY (A) 20%, (B) 30%, (C) 40% AND (D) 50% 67
FIGURE 64: SNR VALUES BASED ON NOISE LEVEL AND SELECTION AREA: LOWER (ETCHED) OR UPPER (PLAIN) 68
FIGURE 65: MEMS PAD WITH ARTIFICIAL REDUCTION IN BRIGHTNESS BY (A) 0%, (B) 50%, (C) 70%, (D) 90% 69
FIGURE 66: SNR VALUES BASED ON LEVEL OF DARKNESS AND SELECTION AREA 70
FIGURE 67: A MEMS PAD WITH DIFFERENT COLOR FILTRATION: (A) RGB, (B) RED, (C) GREEN AND (D) BLUE 73
FIGURE 68: MICROGRAPH OF PAD IN COLOR (LEFT) AND CONVERTED TO GRAYSCALE (RIGHT) USING MATLAB 75
FIGURE 69: SIDE WALL PROFILE OF MEMS PAD (GRAY SCALE) 75
FIGURE 70: SIDE WALL PROFILE OF MEMS PAD (RED FILTER) 76
FIGURE 71: SIDE WALL PROFILE OF MEMS PAD (GREEN FILTER) 76
FIGURE 72: SIDE WALL PROFILE OF MEMS PAD (BLUE FILTER) 77
FIGURE 73: SNR VALUES FOR COLOR FFT TESTS 77
FIGURE 74: BLOCK DIAGRAM OF LABVIEW PROGRAM USED FOR MACRO AND MICRO SCALE TESTS 86
FIGURE 75: SUBVI-1 PROGRAM INTERACTS WITH FLYCAP PROGRAM FOR CAPTURING IMAGES WITH CAMERA 87
FIGURE 76: SUBVI-2 PROGRAM INTERACTS WITH FUNCTION GENERATOR FOR FREQUENCY SWEEPING 87
FIGURE 77: SUBVI-3 PROGRAM SENDS CAPTURED IMAGES TO "RUN" FOLDERS 87

Abstract

This thesis presents a method for dynamic characterization of MEMS structures and discuses parameters that affect its measurements and techniques to improve them. Current methods of non-contact, laser based vibration measurement require special and expensive instruments. The method used in this thesis on the other hand, relies on Fast Fourier Transform analysis of blurred images captured using conventional cameras.

The Fourier series analysis and transformation are introduced. Basic concepts of blur image analysis and associated technical terms are described. Step by step data extraction process for Fourier analysis of blurred images and results such as amplitude, attenuation, signal to noise ratio and Bessel curve are explained.

Macro and micro scale experiments are designed and used to determine the effect and significance of different parameters on signal-to-noise ratio of extracted results. For this purpose geometrical parameters of macro scale combs such as length, width and duty cycle are varied across a considerable range and tests results are examined.

In addition to the experiments, MATLAB code is used to model environmental effects such as addition of noise or changes of brightness. In micro scale experiments, extra patterns are created using Focused Ion Beam and etching process. Test and comparison of modified micro structures with unpatterned structures show improvement in signal to noise ratio especially in environments with high level of noise.

List of Abbereviatios and Symbols Used

2D	2 Dimensional
ATA	AT Attachment (Interface)
b	Oscillatory Value of Blur
b_0	Static Value of Blur
CAD	Computer Aided Design
CCD	Charge-Coupled Device
CMC	Canadian Microsystems Corporation
CSV	Comma-Separated-Values
DFT	Discrete Fourier Transform
DLP	Digital Light Processing
DOF	Degrees Of Freedom
f	Temporal Frequency
f_s	Sampling Frequency
FFT	Fast Fourier Transform
FIB	Focused Ion Beam
IDFT	Inverse Discrete Fourier Transform
Log	Logarithm
MEMS	MicroElectroMechanical Systems
MUMPs	Multi-User MEMS Process
n	Number of Samples Taken from Function
N	Number of Fourier Components
NI	National Instruments
PDE	Partial Differential Equation
PGA	Pin Grid Array

PM	PolyMUMPs
PolyMUMPs	A multi-layer polysilicon based MEMS process
ROI	Region of Interest
SEM	Scanning Electron Microscope
STA	Standard Thermal Actuator, Two-arm Thermal Actuator
t	time
T	Oscillations Period
TA	Thermal Actuator
UI	User Interface
V	Voltage
VI	LabVIEW Virtual Instrument
VPP	Peak to Peak Voltage
ZIF	Zero Insertion Force (Socket)

Greek Symbols

v	Spatial Frequency
ϕ	Phase
λ	Wavelength
μ	Micro

Acknowledgements

First and for most, I thank my supervisors Dr. Ted Hubbard and Dr. Marek Kujath who patiently guided me through my studies and research. It has been a privilege to work with them: their passion and enthusiasm for research and knowledge have been role models for me, their financial support allowed me to pursue advanced studies and above all, they taught me how to learn and master research skills. I also would like to thank my examining committee, Dr. Dominic Groulx and Dr. Yuan Ma for reviewing and commenting on my thesis.

Special thanks to my fellow colleagues and members of research group who I met at Dalhousie University: Rachael Schwartz, Neil Ellerington, Rene d'Entremont, Craig Arthur and Ben Turnbull. Not only they have always been friendly but also helped me both in the MEMS Lab and outside of the university in the new society for me. Many thanks to Peter Jones and Jonathan MacDonald who kindly provided me required mechanical and electrical tools to do experiments in MEMS lab.

Last but not least, thanks to my dear parents and family members. While being far away from Nova Scotia, they have always been supportive and inspiring. I have no doubts that without their assistance, I wouldn't be able to come and study here in Canada. Working with great people and visiting new places have been a valuable experience for me.

Chapter 1: Introduction

1.1: Objective

To investigate both environmental and geometrical parameters that affect the precision of dynamic measurements of MEMS devices based FFT blur method.

1.2: Thesis Scope

This thesis describes Fourier analysis and Fourier transform of blurred waveforms as the basic principle of the measurement method based on blur images. It explains the steps for the detection of the resonance frequency and vibration amplitude. It evaluates parameters that effect measurement results and seeks methods to improve them when it is used for characterization of MEMS devices. This thesis does not consider effect of changing the vibration amplitude and mass/stiffness that shift resonance frequency. They are remained constant throughout the tests for both macro and micro scale models. Damping phenomena and additional modal shapes of structures are also not discussed.

1.3: Author Contribution

While the application of Fourier analysis in dynamic measurements of MEMS devices has been investigated before by Ellerington, et al. [1] this thesis focuses on Signal-to-Noise Ratio (SNR) of measurements and seeks methods to improve it. By developing a macro scale model, the effect of different geometrical parameters on measurement readings was investigated. Based on the macro model test results, micro structure and image processing program were modified to achieve better results. Details of the modification process as well as the final results are discussed. This work also includes the addition of artificial noise and darkness in captured images as well as different color filtrations and examines their effect on the final results.

1.4: **Introduction to MEMS**

MEMS or Micro-Electro-Mechanical Systems refer to devices in the micro scale containing both electrical and mechanical components. They are fabricated with the same techniques used in semiconductor industry including deposition, photolithography and etching. Because of their small size (on the orders of microns), they offer many advantages in comparison with their larger counterparts. They are portable because of their light weight, reliable with proper design and packaging; require little space which means they can be integrated into larger systems, consume little power and cost little if fabricated in mass volumes.

Based on their primary functionality, MEMS devices fall into two main categories: sensors and actuators. Examples of sensors include chemical and pressure sensors in addition to accelerometers which are used extensively in automobile industry as well as consumer electronics. These devices can also be divided into sub categories based on their specific design and application: (1) RF MEMS such as resonators which are used in communication industries, (2) Bio-MEMS: examples include micro fluidic and biomedical micro devices and (3) MOEMS or Optical-MEMS: Digital Light Processing (DLP) used in projectors and micro mirrors used in optical switches with applications in fibre optic communications all fall into this category [2]. There exist two main methods of fabrication of MEMS devices: bulk micromachining and surface micromachining. In the first method, the MEMS structure is formed by etching the bulk silicon. In the second method, the device is constructed by deposition and etching of different layers on top of silicon wafer.

1.4.1: **PolyMUMPs**

One of most commonly used fabrication process for MEMS devices is Multi-User MEMS Processes or MUMPs developed by MEMSCAP Inc. As a standard process, it enables users to access low cost MEMS prototyping and volume production. MUMPs offers three unique multi-mask MEMS processes: PolyMUMPs, SOIMUMPs and MetalMUMPs [3]. The MEMS chips used in this study were fabricated based on PolyMUMPs. Figure 1 shows the layers of PolyMUMPs and their associated thicknesses.

This surface micromachining processes consist of three layers of polysilicon and one layer of metal. The PolyMUMPs layers that are deposited on top of the silicon wafer are as follows: (1) Nitride (0.6 μm): silicon nitride is deposited on the surface of wafer and acts as an electrical isolation layer between polysilicon and the substrate, (2) Poly0 (0.5 μm): first polysilicon layer. Since it is attached to nitride layer, moving mechanical components cannot be made of this layer (3) 1st Oxide (2.0 μm): the first sacrificial layer made of phosphosilicate glass (PSG) which will be removed at the end of fabrication process, (4) Poly1 (2.0 μm): the second polysilicon layer as a main structural layer. It is separated from poly0 by an oxide layer which makes fabrication of free standing structures possible with anchoring. (5) 2nd Oxide (0.75 μm): similar to first one, it separates poly1 and poly2 layers from each other. (6) Poly2 (1.5 μm): third polysilicon layer with functionality similar to Poly1, (7) Metal (0.5 μm): last layer, usually made of Gold or Chromium. It can be used as conductor or for reflecting light sources (micro mirrors). Complete details about this process can be obtained from the handbook provided by MEMSCAP [4]. Table 1 shows mechanical and electrical properties of PolyMUMPs layers [4].

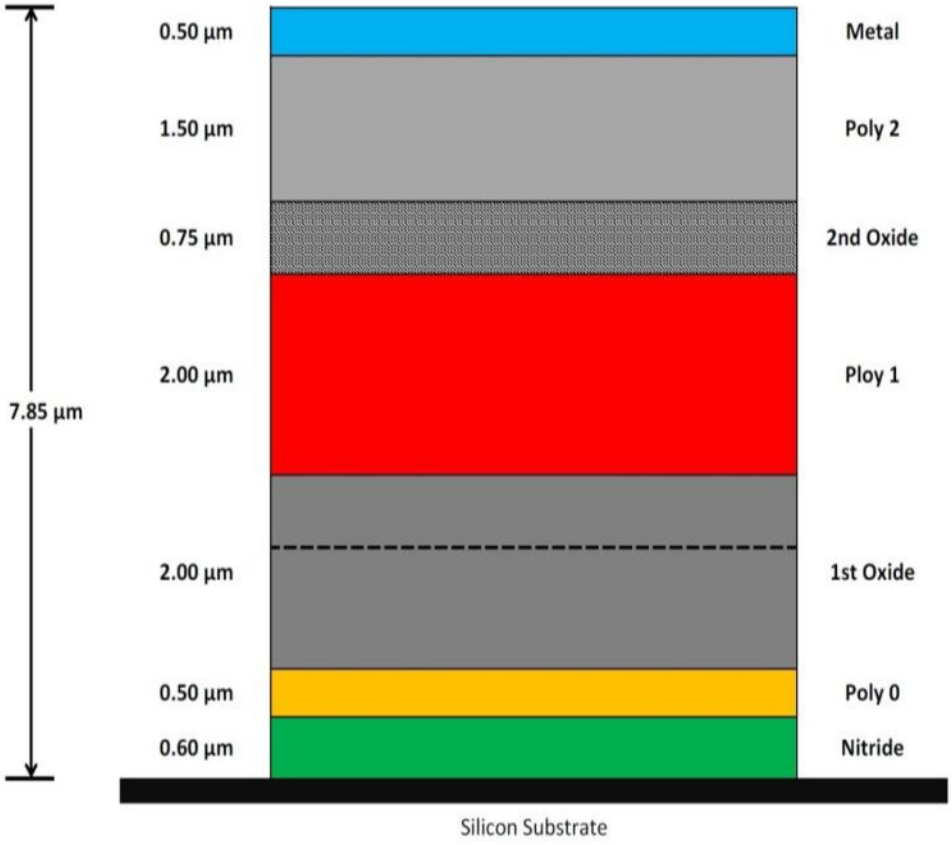


Figure 1: Illustration of PolyMUMPs Layers and Their Thickness [4]

Table 1: Mechanical and Electrical Parameters of PolyMUMPs Layers [4]

Layer	Residual Stress (Mpa)			Resistance (ohm/sq)		
	Min.	Typ.	Max.	Min.	Typ.	Max.
Nitride	0	90	180	N/A		
Poly0	0	-25	-50	15	30	45
Oxide1	N/A			N/A		
Poly1	0	-10	-20	1	10	20
Oxide2	N/A			N/A		
Poly2	0	-10	-20	10	20	30
Metal	0	50	100	0.05	0.06	0.07

1.5: Dynamic Measurement of MEMS Devices

Since the feature sizes of MEMS devices are on the order of microns, they possess very small mass and have high stiffness to mass ratio. Hence, the resonance frequency of such devices occurs at high frequencies in orders of kHz or MHz. Resonance measurement of MEMS devices and especially MEMS resonators is important for their characterization and design optimization. Because of their small size, fragile structure and operation at high frequency, no direct contact measurement method can be used [1].

1.5.1: Optical Methods

Several noncontact optical methods are available for both in-plane and out-of-plane dynamic measurement of micro devices. Laser Doppler vibrometry, laser deflection and interferometric systems are used for high resolution measurement out-of-plane motion [5]. For in-plane motion measurements, three techniques are usually employed: high-speed photography, stroboscopic methods and blur methods. Blur phenomenon occurs when a camera captures an image of an object moving with high speed relative to camera.

Two in-plane measurement methods attempt to remove the blur effect. High-speed photography utilizes specific cameras with short shuttle speed to eliminate blur. This method is expensive and has limited functionality in high resonance frequencies associated with MEMS devices. The second method or stroboscopic uses flashing or pulsing light at specific frequency to make fast moving objects appear to be stationary. However this method has limited functionality at high frequencies similar to high-speed photography. As the frequency increases, the illumination time as well as image intensity decreases [1, 6]

1.5.2: **Blur Methods**

Measurement methods based on blur do not attempt to remove it but they extract properties of dynamic motion by measuring the blur. In contrast to the two methods described above, this method does not require specific and expensive equipment and is not limited in high frequency measurements. The blurred image is widened in the direction of motion and the highest amount of blur occurs at the resonance frequency, therefore the resonance frequency can be detected by measuring the blur width. Several researchers have used blur techniques to measure MEMS actuators frequency response. By using the PSF (point spread function) of micrographs, Le measured the motion of an electrostatic comb drive [7]. Krylov measured the blur width of microphotographs of an electrostatic oscillator to identify its resonance frequency [8]. By synthesizing blurred images from a static image and matching them to measured blurred images, Burns and Helbig determined dynamic motion amplitudes [9].

In addition to measurement methods in the time domain, some work in MEMS research has also been performed in the frequency domain. Yamahata used fast Fourier transform to measure in-plane static displacement of MEMS structures with subnanometer resolution [10]. Ellerington combined FFT method with blur width techniques for high frequency in-plane dynamic measurement of MEMS devices [1]. The aim of this thesis is to further investigate the last method and find the parameters that have the most significant effect on the measurement results and attempts to improve the final results.

Chapter 2: Fourier Analysis

2.1: Introduction

Fourier analysis enables the decomposing of a function into simpler parts. This decomposition makes the study of complex functions much easier. It is named after Joseph Fourier because of his significant contribution to study of trigonometric series. The reverse process, rebuilding of original function from its components, is called Fourier synthesis [11].

2.2: Fourier Transform

The Fourier transform is an integral transform that reflects any function of *f(t)* to another function *F(ω)*, therefore *F(ω)* is Fourier transform of *f(t)* and it is calculated as follows:

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad \textbf{(1)}$$

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} dt \quad \textbf{(2)}$$

Fourier transforms and Fourier analysis have many applications in different disciplines such as: electronic, telecommunications, acoustics, image processing, etc.

2.3: Fourier Series

The Fourier series is special case of Fourier transform when function *f(t)* is periodic with period of T: *f(t+T)=f(t).* It enables the decomposition of a periodic function into a sum of simple harmonic functions (sines and cosines). For example a square wave can be represented by series of sine waves of decreasing amplitude [11]. A Fourier polynomial is defined by [12]:

$$F_n(x) = a_0 + \sum_{k=1}^{k=n} (a_k \cos(kx) + b_k \sin(kx)) \quad (3)$$

The constants a_0, a_k and b_k are called coefficients of $F_n(x)$. Based on trigonometric identities and as 2π periodic functions, they will be equal to:

$$\begin{cases} a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_n(x)dx \\ a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} F_n(x)\cos(kx)dx\,, \qquad 1 \le k \le n \\ b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} F_n(x)\sin(kx)dx\,, \qquad 1 \le k \le n \end{cases} \quad (4)$$

The following trigonometric series is called the Fourier series and associated to the function $f(x)$:

$$F(x) \sim a_0 + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx)) \quad (5)$$

For example, let's see what would be Fourier series of square function with amplitude of $\pi/4$ and 2π period:

$$f(x) = \begin{cases} -\frac{\pi}{4}, -\pi \le x < 0 \\ \frac{\pi}{4}, 0 \le x \le \pi \end{cases} \quad (6)$$

Since $f(x)$ is odd, for $n \ge 0$ then $a_n=0$. For $n \ge 1$, coefficient b_n will be:

$$b_n = \frac{1}{\pi}\int_{-\pi}^{0} -\frac{\pi}{4}\sin(nx)dx + \frac{1}{\pi}\int_{0}^{\pi} \frac{\pi}{4}\sin(nx)\,dx \quad (7)$$

$$b_n = \frac{1-\cos(n\pi)}{2n} = \frac{1-(-1)^n}{2n} \quad (8)$$

Therefore, $b_{2n}=0$. The resultant $F(x)$ is given by equation (9). Figure 2 shows plot of $F(x)$.

$$F(x) \sim \sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5} + \cdots \quad (9)$$

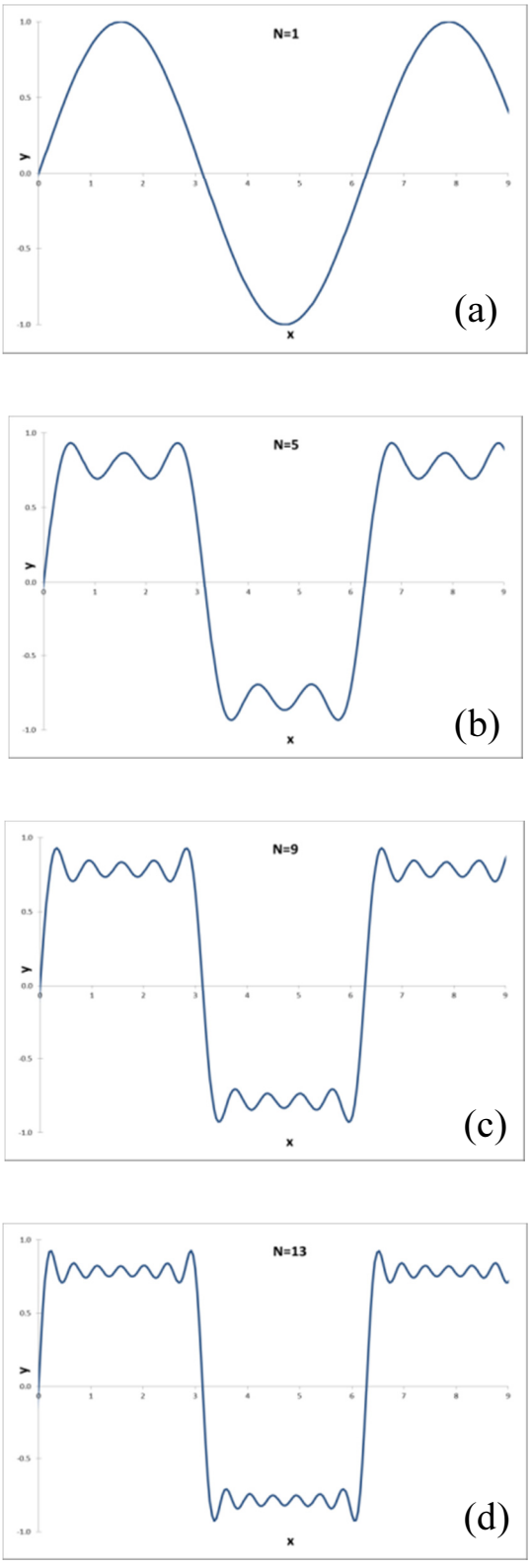


Figure 2: Approximation of Square Wave by Odd Components of Its Fourier Series [12]: (a) 1, (b) 1-3-5, (c) 1-3-5-7-9, (d) 1-3-5-7-9-13

2.4: **Discrete Fourier Transform**

Discrete Fourier transform (DFT) enables the transformation of finite domain discrete functions and signals from the time domain to the frequency domain. The discrete function is usually created by the sampling of a continuous function; finite domain commonly indicates a finite segment of a function that is analyzed instead of the function itself being finite. DFT is defined by the following function where k and n represent frequency and samples taken in time domain respectively [13]:

$$X_k = \sum_{n=0}^{N-1} x_n . e^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1 \tag{10}$$

The "n" is used to convert continuous function into discrete form. Figure 3 shows a plot of Fourier transform for a sine function (f = 20 Hz). Sampling was performed within 0-100 seconds with 1s steps. It is plotted by 500 defining points (N=500) with peak at 20 Hz as expected. In Figure 3, another peak occurs at 80 Hz (mirror of first one with respect to the symmetry line-50 Hz) due to the DFT symmetric nature. The plotted peak is not ideal and includes width and side lobes due to the discrete nature of samples.

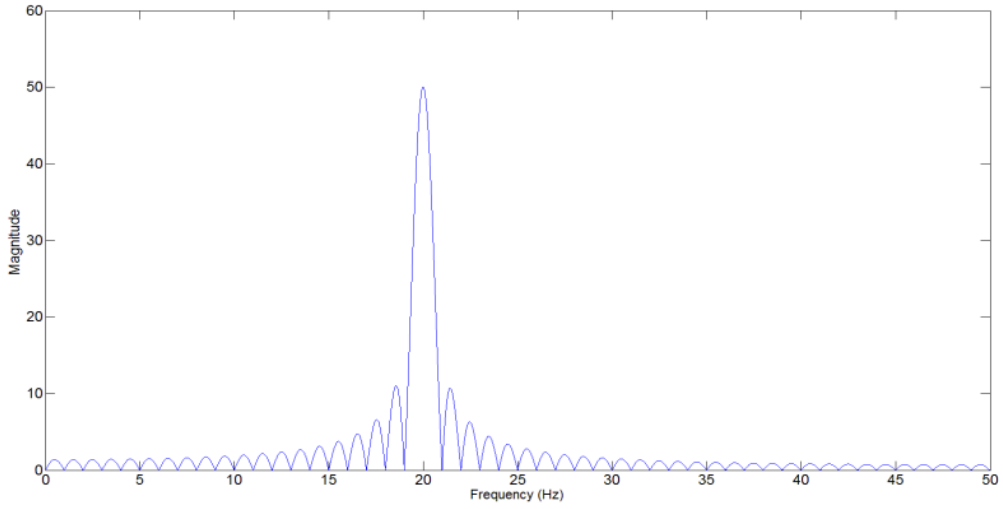


Figure 3: Plot of Discrete Fourier Transform for a Sine Function with Frequency of *20 Hz*

DFT is a linear and complete transform: no data will be lost during transformation and it is invertible. Therefore the initial signal can be reconstructed from transformed form. Based on DFT definition, Inverse Discrete Fourier Transform (IDFT) will be:

$$x_n = \frac{1}{N}\sum_{n=0}^{N-1} X_k . e^{\frac{2\pi i}{N}kn} \quad k = 0, \dots, N-1 \qquad \textbf{(11)}$$

DFT is heavily used in digital signal processing (DSP). It enables spectral analysis of signals and can be used for determination of their distortion, main frequency, harmonics, bandwidth, etc. These parameters are not easily identifiable in the time domain [14]. It is also used for solving partial differential equations (PDE): in this method the signal is expanded into complex exponentials which are eigenfunctions of differentiations, this makes differentiation simple. Another application of DFT is the fast multiplication of polynomials. First, a polynomial is converted to another form so the mathematical operations of multiplication and division can be performed faster, then final answer can be retrieved in from of polynomial with the help of IDFT [13, 15].

2.5: **Fast Fourier Transform**

The Fast Fourier transform (FFT) is an algorithm for fast and efficient computation of direct and inverse DFT. While DFT itself has many useful applications in different disciplines, its direct computation is very slow and time consuming. FFT method on the other hand, provides same results of DFT in much shorter time: Calculation of DFT for N points requires $O\ (N^2)$ mathematical operations; while FFT requires $O\ (N \times Log\ N)$ operations. For example, if N=1000 then DFT requires 10^6 (N^2) operations while FFT only needs $3*10^3$ ($N \times Log\ N$). Therefore, the difference in speed can be quite significant especially for big data sets in which the calculation time will be

reduced multiple times. There is no doubt that the efficiency of FFT makes the use of DFT practical, the improvement in required time for calculations can be approximated as *N/Log N.* In the above example (*N*=1000) improvement would be 1000/3, therefore with FFT only about 1/300 of initial time is required to achieve final result [16].

There exist a large number of FFT algorithms such as Rader, Bruun, Prime-factor, Bluestetin, etc. but the most commonly used algorithm is Cooley-Tukey. It recursively breaks down a composite DFT ($N=N_1.N_2$) and converts it into DFTs with dimensions of N_1 and N_2. FFT algorithms cover wide range of mathematics from calculations of complex numbers to theory of numbers and despite some differences almost all of them require *O (N Log N)* number of operations. They all calculate the approximate value of DFT and contain some amount of error. These errors are small and prevent large and extensive calculations required by DFT itself, therefore such algorithms trade fast calculations and results with very small amount of relative error. In most cases these errors are so small that are quite negligible. For example, the maximum relative error for Cooley-Tukey algorithm is *O (ε log N)* where ε is the precision of calculations [13, 17, 18].

Chapter 3: FFT Analysis of Blurred Curve

3.1: Overview

This chapter discusses Fourier analysis of a 1D profile. For example, the 1D profile can be the cross-section of a MEMS device side wall profile. A mathematical model for the simulation of a blurred profile, its Fourier expression and attenuation of harmonics was presented. The developed model shows that the blur effect in the frequency domain acts as a low pass filter. This conclusion forms the basis for in-plane dynamic measurement of MEMS devices and its methods of improvements presented in this thesis. Fourier series of a trapezoid waveform and its blur image were also numerically simulated and plotted as proof of concept. The magnitudes of FFT harmonics for blurred wave were fitted well with a Bessel curve absolute value.

3.2: Definitions

3.2.1: FFT Harmonics

For a periodic signal with main frequency of *f*, the component frequencies of the original signal are called harmonics and are integer multiples of the main signal. For a signal with frequency of *f*, harmonics are *f*, *2f*, *3f*, *4f,* etc. and are called first harmonic, second harmonic and so forth. The first frequency (*f*) is also called fundamental frequency. Figure 4 shows first four harmonics of a sine wave with fundamental frequency of $1/\pi$. Since all of harmonics are periodic at the fundamental frequency, their sum is periodic at that frequency as well. The mathematical study of overlapping waves and how they can be used to represent main function is called harmonic analysis. It has applications in diverse fields including signal processing [19].

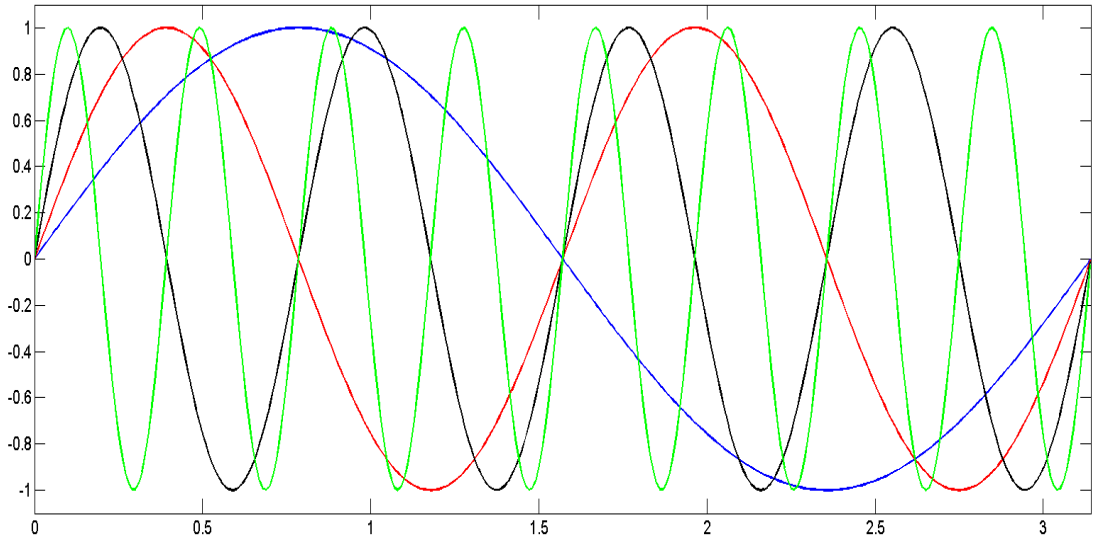


Figure 4: First Four Harmonics of a Sine Wave with Frequencies of 1/π (blue), 2/π (red), 4/π (black) and 8/π (green). All Harmonics have Same Amplitude

As demonstrated in chapter 2, the Fourier series of a function contains different components each being a periodic function. Fourier transformation of each Fourier series components into the frequency domain will result in FFT harmonics. If a function is symmetric about the x axis, it only contains odd FFT components and therefore odd FFT harmonics (even components are zero). Equation (12) shows Fourier series of the trapezoid waveform.

$$F(x)_{Trapezoid} = \sum_{n=1,3,5,..}^{\infty} \frac{a_n \sin nx}{n^2} = \sin(x) + \frac{1}{9}\sin(3x) - \frac{1}{25}\sin(5x) - \cdots \qquad \textbf{(12)}$$

Figure 5 shows FFT harmonics for a trapezoid waveform: At top image (a) only the first component of Fourier series (sine wave with wavelength of 2π) was transformed into frequency domain. As more components of Fourier series included, the trapezoid shape of original wave becomes more evident. Numbers on vertical axis are in logarithmic scale: magnitude of third harmonic is 0.1 of first one.

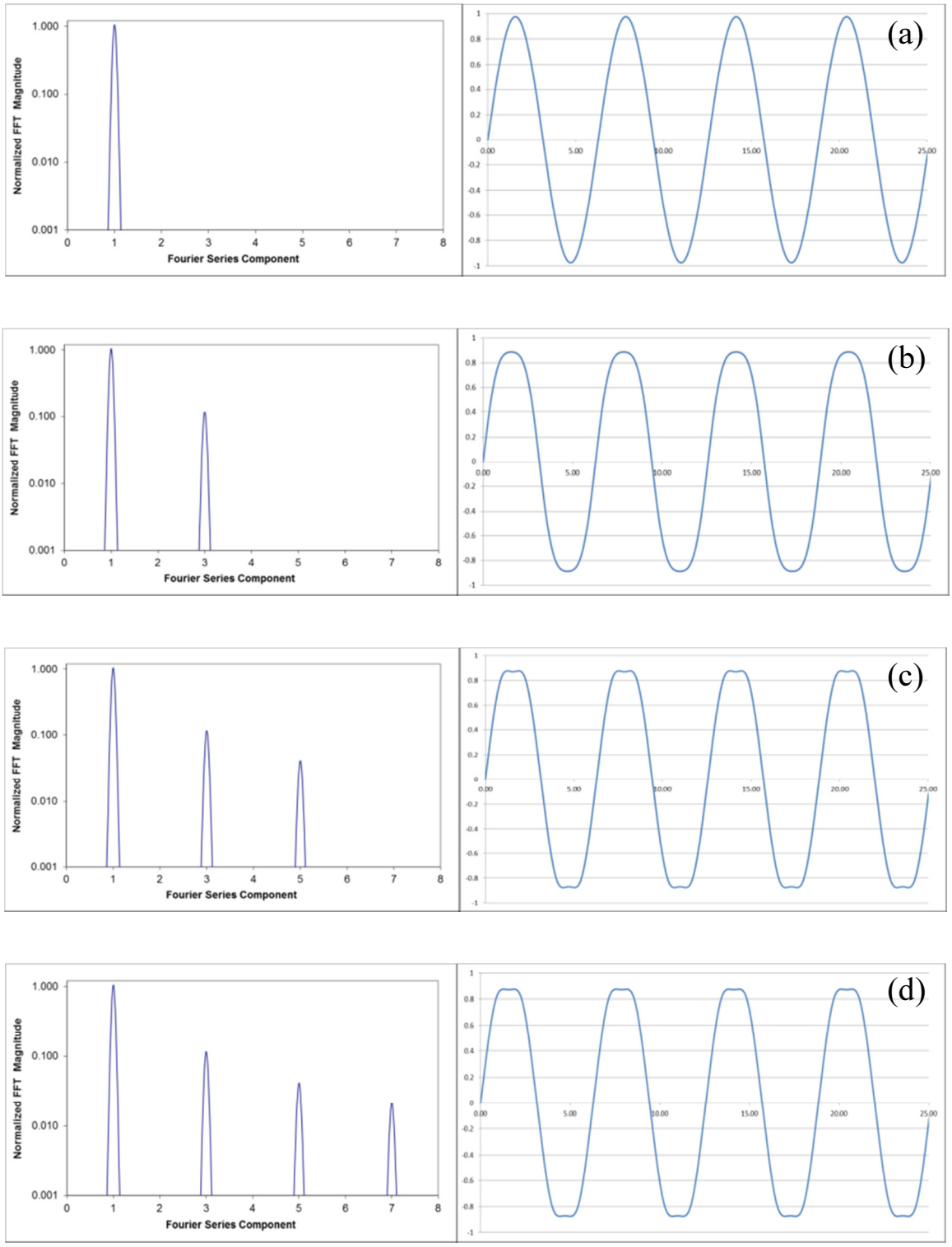


Figure 5: Trapezoid Curve with Different Number of Fourier Series Components (Right) and Associated Normalized FFT Harmonics (Left). (a) 1, (b) 1-3, (c) 1-3-5, (d) 1-3-5-7

3.2.2: **Bessel Function**

The Bessel function is the solution for Bessel second-order differential equation:

$$x^2 \frac{d^2y}{dx^2} + x \frac{dy}{dx} + (x^2 - \alpha^2)y = 0 \quad \textbf{(13)}$$

Where α is a constant. It is named after German astronomer *Friedrich Wilhelm Bessel* (1784 - 1846). When α is an integer, the notation changes to n:

$$x^2 \frac{d^2y}{dx^2} + x \frac{dy}{dx} + (x^2 - n^2)y = 0 \quad \textbf{(14)}$$

Being a second order deferential equation, it has two independent solutions: (1) $J_n(x)$ and (2) $Y_n(x)$. Combination of these solutions is also a solution. The first solution $J_n(x)$ is called a Bessel function of first kind with "n" the order and "x" the argument. It can be written as an infinite polynomial as shown in equation (15). A plot of Bessel function of first kind zero order is shown in Figure 6.

$$J_n(x) = \sum_{i=0}^{\infty} \frac{(-1)^i}{i\,!\;\Gamma\,(n+i+1)} \left(\frac{x}{2}\right)^{2i+n} \quad \textbf{(15)}$$

Bessel functions have useful applications in problems involving static potentials and wave propagation such as electromagnetic waves or heat conduction. They also have useful properties in signal processing. Bessel function can be used to represent frequency modulation (FM) synthesis: the harmonic distribution of a simple sine wave signal modulated by another sine wave signal. Bessel functions are introduced here because they will appear later in this study [20, 21].

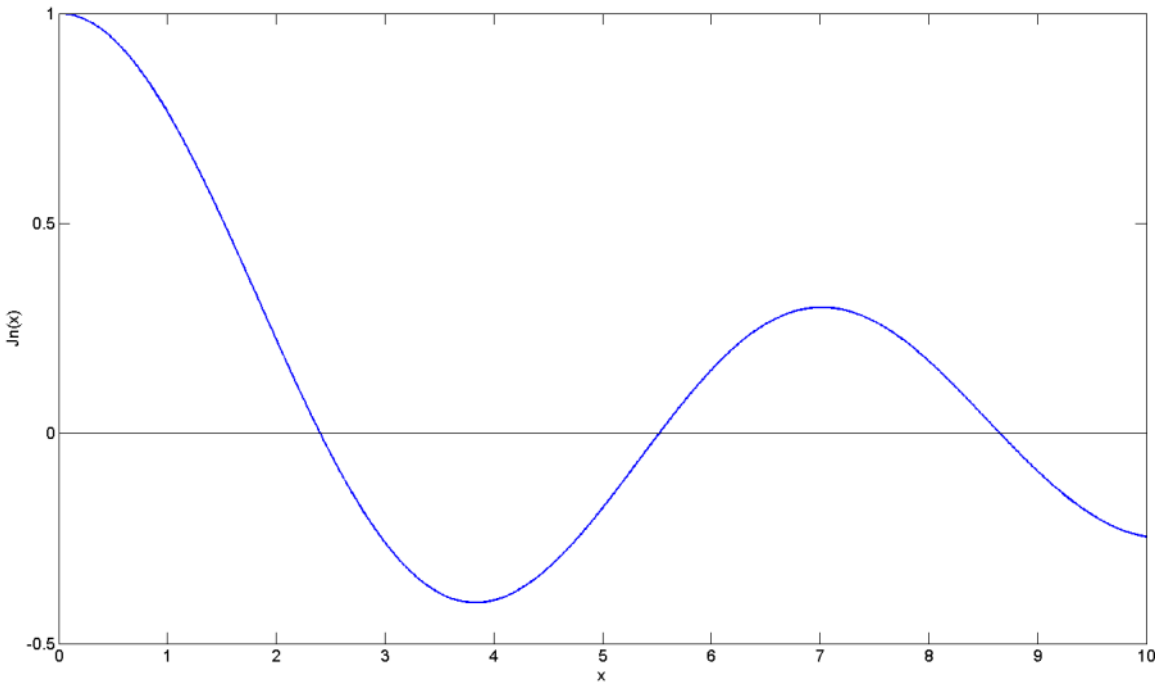


Figure 6: Plot of Bessel Function of First Kind Zero Order

3.3: **Mathematical Modeling of Blur**

The following sections are adapted from Ellerington, et al. [1]: If *p(x)* represents side wall profile of static curve, its Fourier series would be:

$$p(x) = \sum_{i=1}^{N} p_i \sin(v_i x + \phi_i) \quad \textbf{(16)}$$

In the above equation p_i represents the magnitude of i^{th} Fourier component. The spatial frequency is represented by v_i and the phase of i^{th} component by ϕ_i. If λ_i represents the spatial wavelength of i^{th} component, then $v_i = 2\pi/\lambda_i$ [1].

In order to model blurring phenomena in the time domain, the spatial amplitude of blur as a function of time (t) with a periodic sinusoidal function was represented [1]:

$$blur\ (t) = b_0 + b * \sin(2\pi * \frac{t}{T}) \quad \textbf{(17)}$$

In equation (17) b and b_0 represent oscillatory and static value of blur respectively. Temporal period of oscillations is shown with T where $T=2\pi/\omega$ in time domain.

Let's consider spatial blurring of a single sine component: $\sin(v_i x + \phi_i)$. Since the shutter speed of capturing camera is much slower than the oscillation frequency, the resultant blurred image is the temporal average of the curve profile. The resultant blurred image of a single sine wave component $p(x)$ can be given as a function of position [1]:

$$sin_{blur}(v_i x) = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \sin[v_i x + \phi_i + blur(t)]dt \quad \textbf{(18)}$$

In equation (18), the term *blur (t)* is defined in equation (17) and the temporal period of oscillations is shown with *T.*

Equation (18) can be turned into a simpler form by assigning components that do not vary with time $[\omega_i x + \phi_i + b_0]$ to "*A*", those vary with time $[b * \sin(2\pi * t/T)\ ]$ to "*B*" and using trigonometric identity: $[\sin(A + B) = \sin(A)\cos(B) + \cos(A)\sin(B)]$. "*A*" components can be moved outside of the integral as they are constant and the term $\sin(B)$ and hence $\cos(A) * \sin(B)$ disappear since the integral for an odd function over the period is equal to zero. Therefore equation (18) can be rewritten as [1]:

$$sin_{blur}(v_i x) = \sin[v_i x + \phi_i + b_0][\frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \cos[v_i * b * \sin(2\pi * \frac{t}{T})]dt] \quad \textbf{(19)}$$

In equation (19), the first part represents the original wave with b_0 amount shifted; the second part is the attenuation factor. For even functions such as *cos,* the integral from *-T/2* to *T/2* is equal to twice of same integral from 0 to *T/2.* A substitution was made for

the time variable of *sin* function in equation (19) by $\gamma = (2\pi * t/T)$ and attenuation was defined as the ratio of the blurred to unblurred magnitudes, therefore 2nd part becomes:

$$Attenuation_i = \frac{1}{\pi}\int_0^{\pi} \cos[v_i * b * \sin(\gamma)]d\gamma = J_0(2\pi\frac{b}{\lambda_i}) \quad \textbf{(20)}$$

Equation (20) is Bessel function of first kind 0 order: J_0 [22]. The absolute value of Bessel function in above equation can be determined based on magnitude of Fourier components calculated by Fourier transform. In equation (20) and for the same amount of spatial blur amplitude b, value of b/λ_i increases with higher order harmonics as wavelength of i^{th} component λ_i is i times shorter than fundamental wavelength λ_1 ($\lambda_i = \lambda_1/i$). The blur was indicated in radians with $b_i^* = 2\pi * (b/\lambda_i)$, therefore it becomes evident that blur in frequency domain acts as low pass filter: it attenuates all frequency components but higher order harmonics will be attenuated more.

3.4: **Numerical Simulation**

A trapezoid waveform was numerically blurred to confirm the developed mathematical model and the derivation of the Bessel curve attenuation. For modeling purpose, first 13 components of waveform Fourier series was used to construct its ideal side wall profile:

$$Trapeziod(x) = \sum_{n=1}^{13} \frac{a_n}{n^2}\sin(nx) \quad \textbf{(21)}$$

In equation (21) a_n is the alternating sequence 1, 1, -1, -1, 1, 1, -1, -1, etc. and since the trapezoid waveform is symmetrical about x-axis, it contains only odd harmonics, therefore n= 1, 3, 5…

Blur phenomena in the waveform was numerically simulated by phase shifting of the original profile about its x-axis: -λ/20 to the left and +λ/20 to the right with 50 steps in between resulting 51 shifted waveforms. In order to simulate sinusoidal blurring, the spacing between shifts was is not uniform but they were spaced sinusoidally with more curves near the extremes and fewer curves near the origin:

$$Trapeziod_{\,Blur}(x) = \frac{1}{51}\sum_{t=-25}^{t=25}\sum_{n=1}^{13}\frac{a_n}{n^2}\times \sin\left[n\left(x+\frac{2\pi}{20}*\sin\left(\pi*\frac{t}{25}\right)\right)\right] \qquad \textbf{(22)}$$

The Fourier transform on both original and blurred waveforms was performed with 256 sample points. Their plot is shown in Figure 7: the blur (deformed) waveform is very similar to sinusoidal waveform because it only contains first few components of Fourier series as the rest are severely attenuated by blur effect. Figure 8 shows the normalized magnitudes of FFT harmonics for both original and blurred waveform. Figure 9 shows plot of the attenuation (ratio of blurred to unblurred FFT magnitudes) versus blur phase in radians. Calculated numerical values match well with absolute value of Bessel curve as a low pass filter. It was numerically proved that blur in spatial domain acts as low pass filter. See Ellerington et al. paper for more details about this simulation [1].

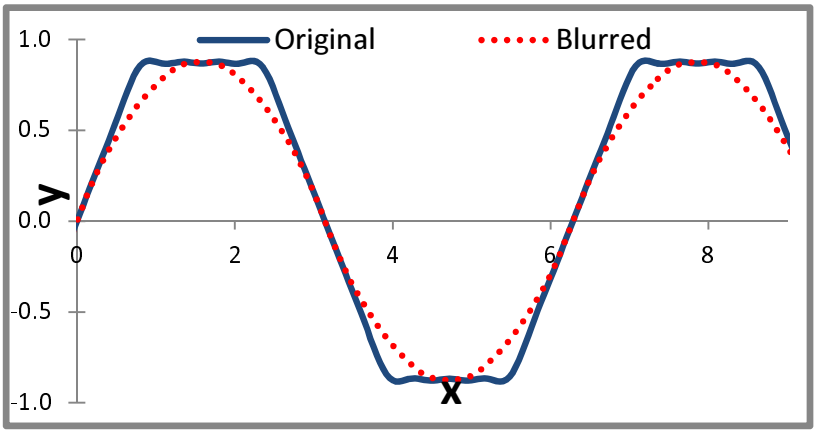


Figure 7: Discrete Fourier Approximation of Original and Blurred Trapezoid Wave Form. Fourier Approximation is done with 13 Components (N=13) and 256 Sample Points (n=256)

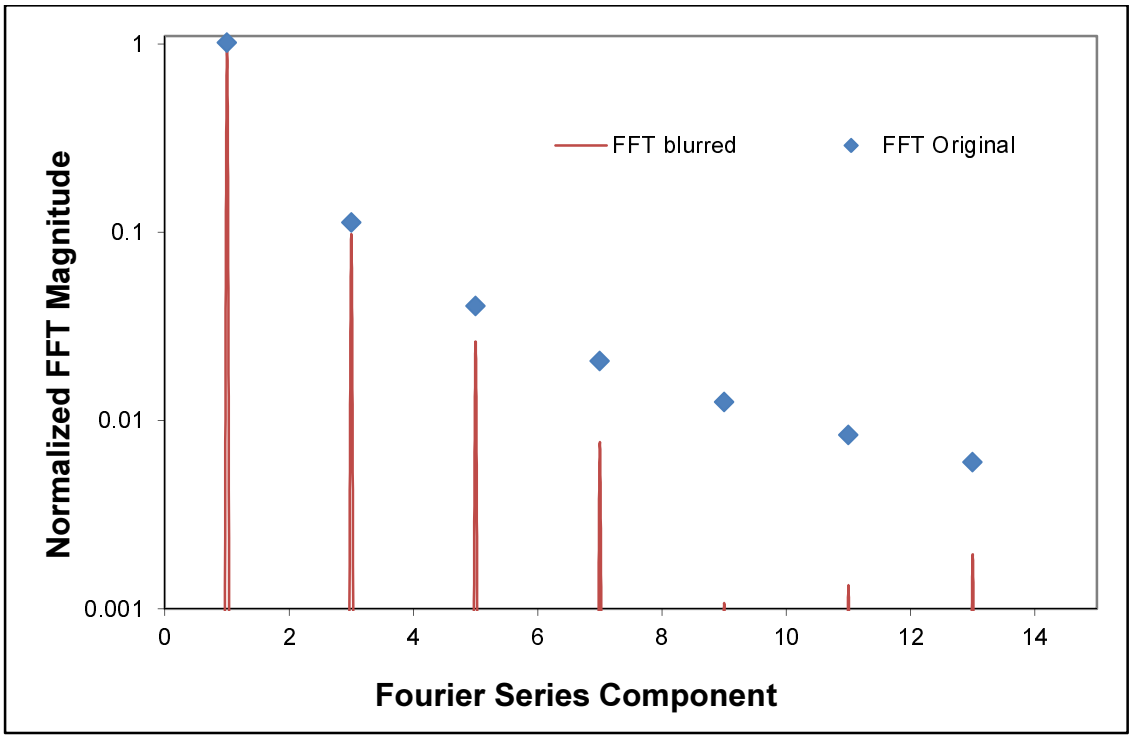


Figure 8: Magnitude of Normalized FFT Harmonics for Blurred and Original Trapezoid Wave Form Indicated with Lines and Squares Respectively

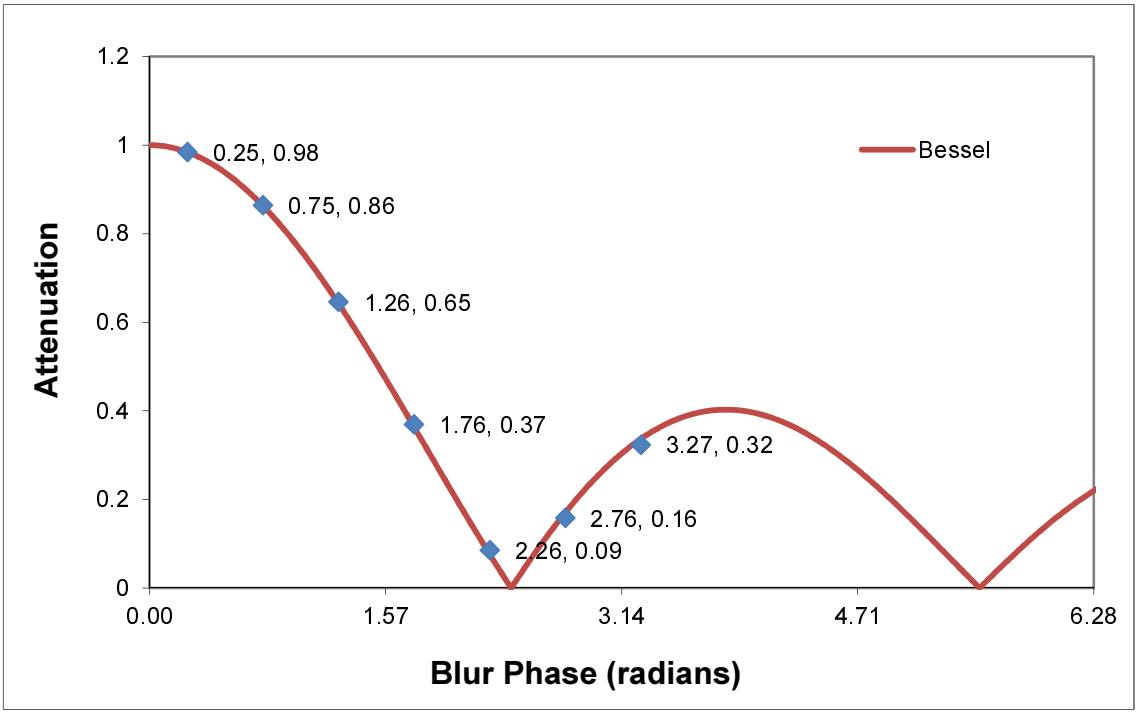


Figure 9: Fit Between Numerical Values of Harmonics (Squares) and |Bessel Curve|. The Pairs of Numbers Represent Amount of the Blur Phase and the Attenuation.

Chapter 4: 2D Image FFT Blur Extraction

4.1: Overview

Consider a comb with periodic patterns as shown in Figure 10 attached to a vibrating structure. The camera takes a series of pictures at vibration frequencies below resonance of the structure, gradually increasing frequencies and passing resonance and finally stops past resonance. At frequencies away from resonance, the image appears almost static because of the small amplitude of vibration. At frequencies closer to resonance however, the blur becomes more visible and at resonance, the highest amount of blur occurs. Figure 10 shows sample images for each case. This chapter describes the process and analysis performed on series of images taken in above situation to determine resonance frequency and amplitude of vibration. It is developed by N. Ellerington, et al. and explained here because it is used extensively throughout our experiments explained in next chapters [1]. Summary of the required steps for this process are shown in Figures 11 and 12. It can be summarized into five steps: (I) Row averaging of 2D images (II) Calculation of FFT harmonics (III) Detection of resonance frequency (IV) Bessel curve fitting of numerical values (V) Conversion to displacement.

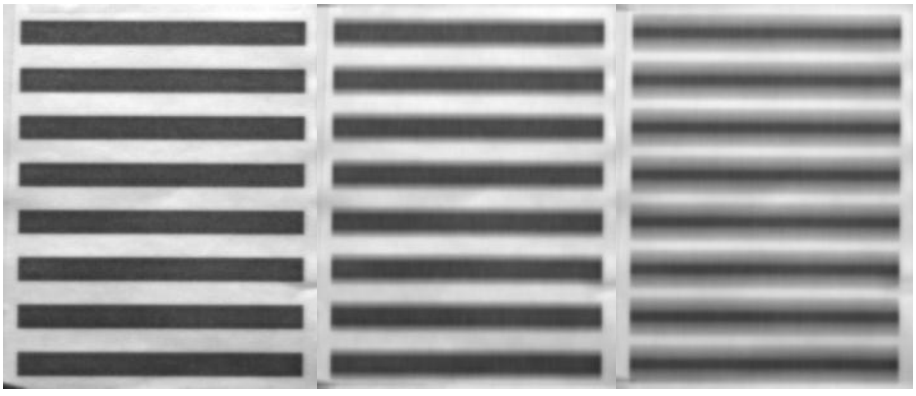


Figure 10: Static (Left), Slightly Blurred (Middle) and Severely Blurred (Right) Combs. Blur was simulated with MATLAB

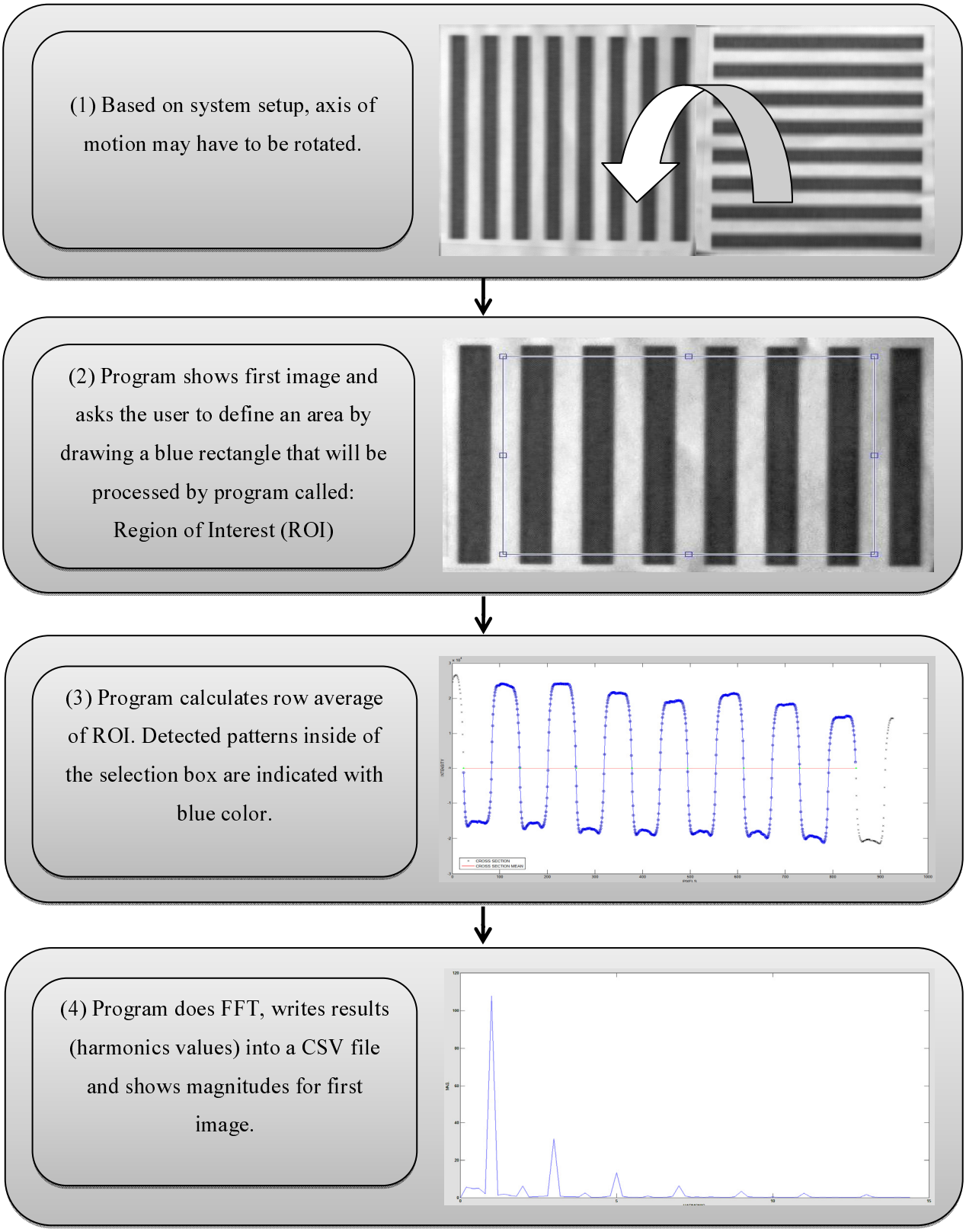
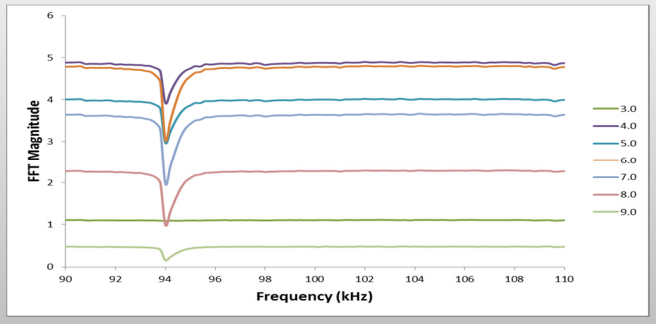
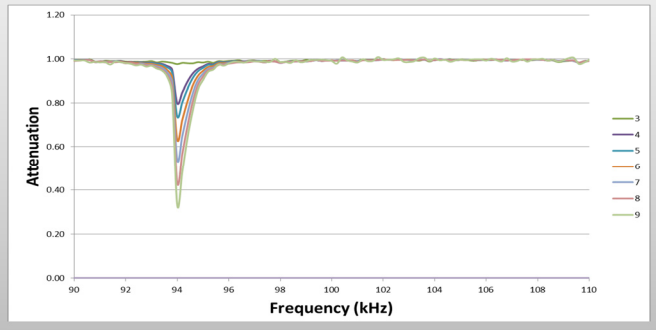


Figure 11: FFT Blur Image Analysis Process (I)

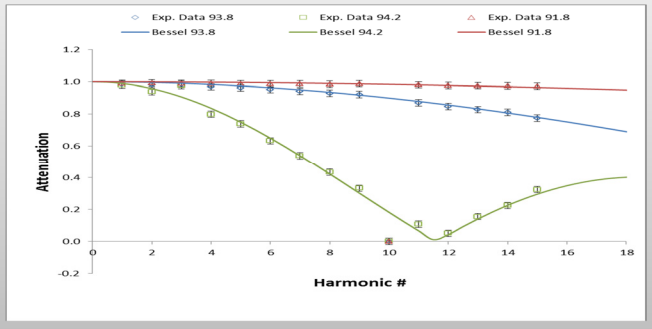
(5) Graph of FFT magnitude vs. freq. for each harmonic. Higher harmonics show higher attenuation at resonance freq.



(6) Normalized magnitude of attenuation for harmonics. (Each plot is normalized by division to its high frequency value)



(7) Attenuation vs. harmonics for three frequencies. Points represent experimental data and lines are best fit curves.



(8) Plot of displacement vs. frequency. Bessel curve is used to convert attenuation to displacement.

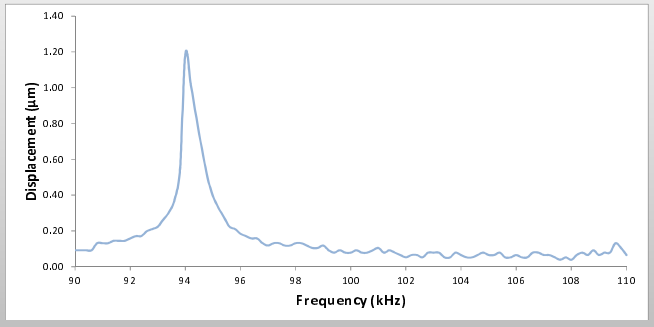


Figure 12: FFT Blur Image Analysis process (II)

4.2: **Row Averaging of 2D Image**

A MATLAB program was developed by Ellerington, et al. to perform the Fourier analysis of captured images [1]. Firstly, the program needs to know direction of vibration (X or Y). Y is default value and if the user enters X, the program rotates each image 90 degrees before proceeding (Fig. 11-1). In the next step, the first captured image will be shown and user defines region of interest (ROI) by plotting a rectangle on appeared image. The selected area determines where the image processing will be performed (Fig. 11-2). The program calculates the row average and shows the 1D side wall profile and the number of detected waves based on black and white patterns in selected ROI (Fig. 11-3). Figure 13 and 14 show the resultant row averages for static and blurred combs.

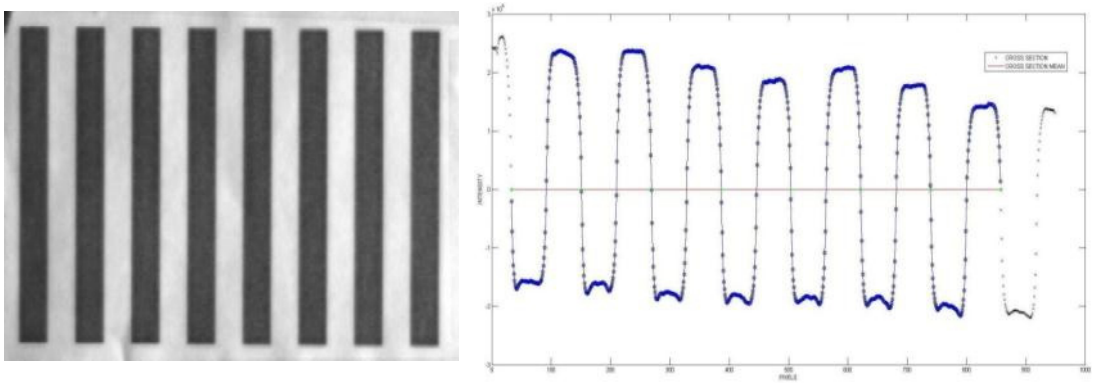


Figure 13: Image of 2D Static Comb and Plot of Its Row Average

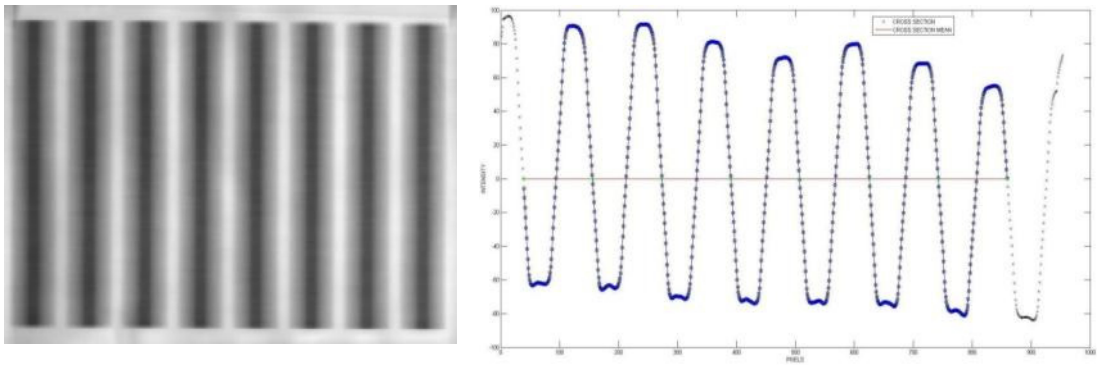


Figure 14: Image of 2D Blurred Comb and Plot of Its Row Average

4.3: **Calculation of FFT Harmonics**

A Fast Fourier transform (FFT) is then performed on the resultant periodic waves (side wall profiles). The program performs the previous step and FFT calculations on each and every captured image and writes the results into a CSV file. It must be noted that the selected ROI for first image will be used for processing of the rest of images as well (Fig. 11-4). Figure 15 and 16 show resultant FFT harmonics for static and blurred combs.

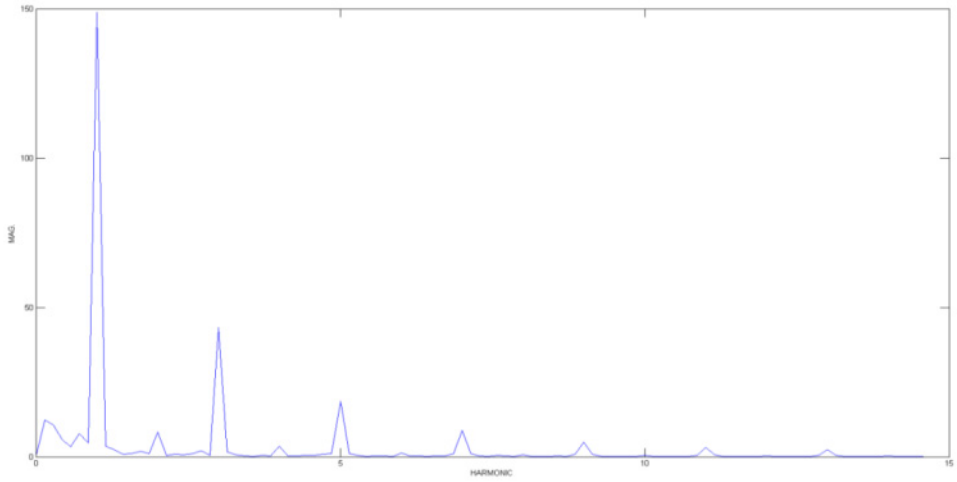


Figure 15: Magnitudes of FFT Harmonics for Static Comb

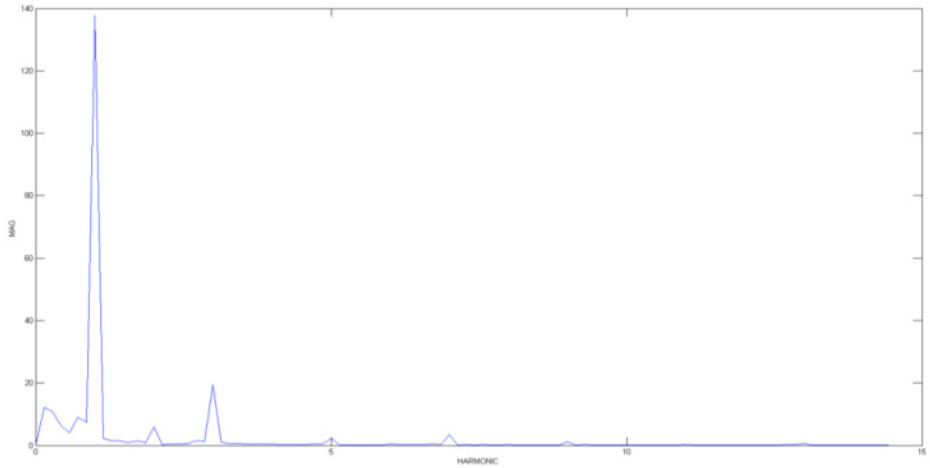


Figure 16: Magnitudes of FFT Harmonics for Blurred Comb

4.4: **Detection of Resonance Frequency**

The second part of the FFT blur extraction process involves extracting raw data from the CSV file and importing it into specifically designed spread sheet (excel file). By doing this, user can obtain the magnitudes of FFT for 15 different harmonics (1^{th}-15^{th}) within the defined frequency range. Figure 17 shows a larger picture of plot used in flowchart step five (Fig. 12-5), for clarity only seven harmonics are displayed. The normalized magnitude of attenuation for selected harmonics can be obtained by dividing the points of the plot to their high frequency average. For this propose the user has to pick a high frequency value after resonance when plot becomes flat. The resultant plot as shown in Figure 18 enables user to compare the attenuation of harmonics with each other. As expected, higher order harmonics show larger amounts of attenuation (Fig. 12-6).

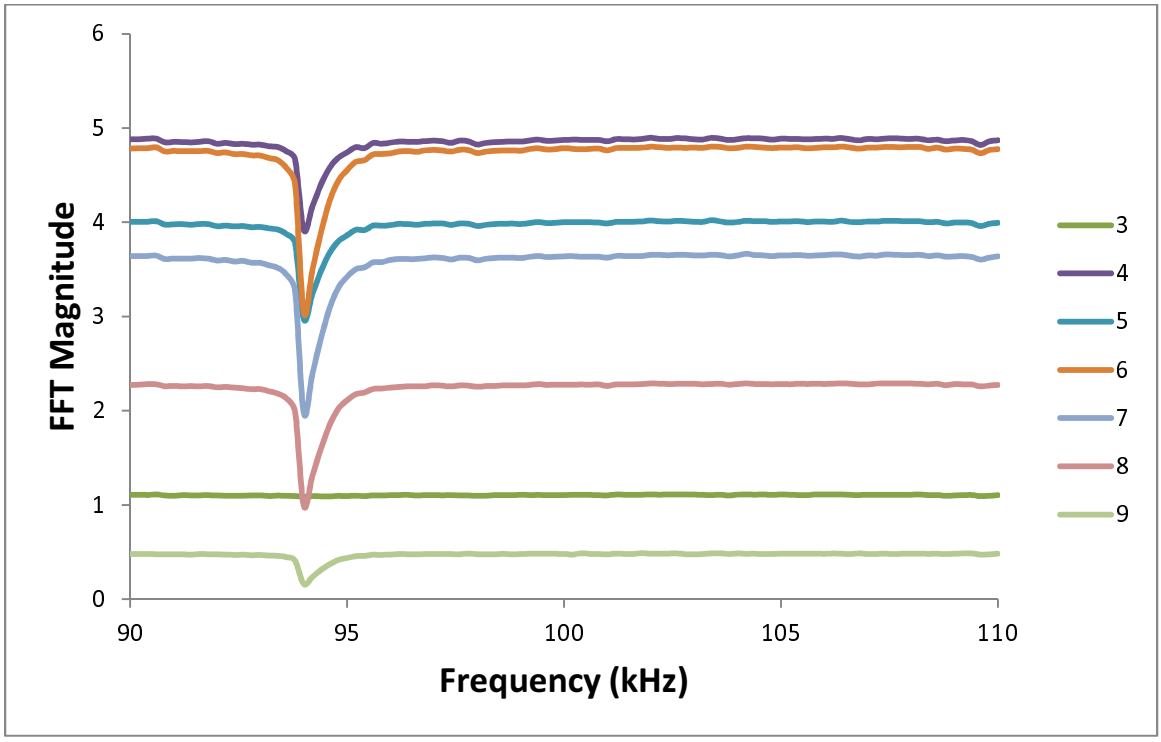


Figure 17: Magnitude of FFT Harmonics vs. Frequency

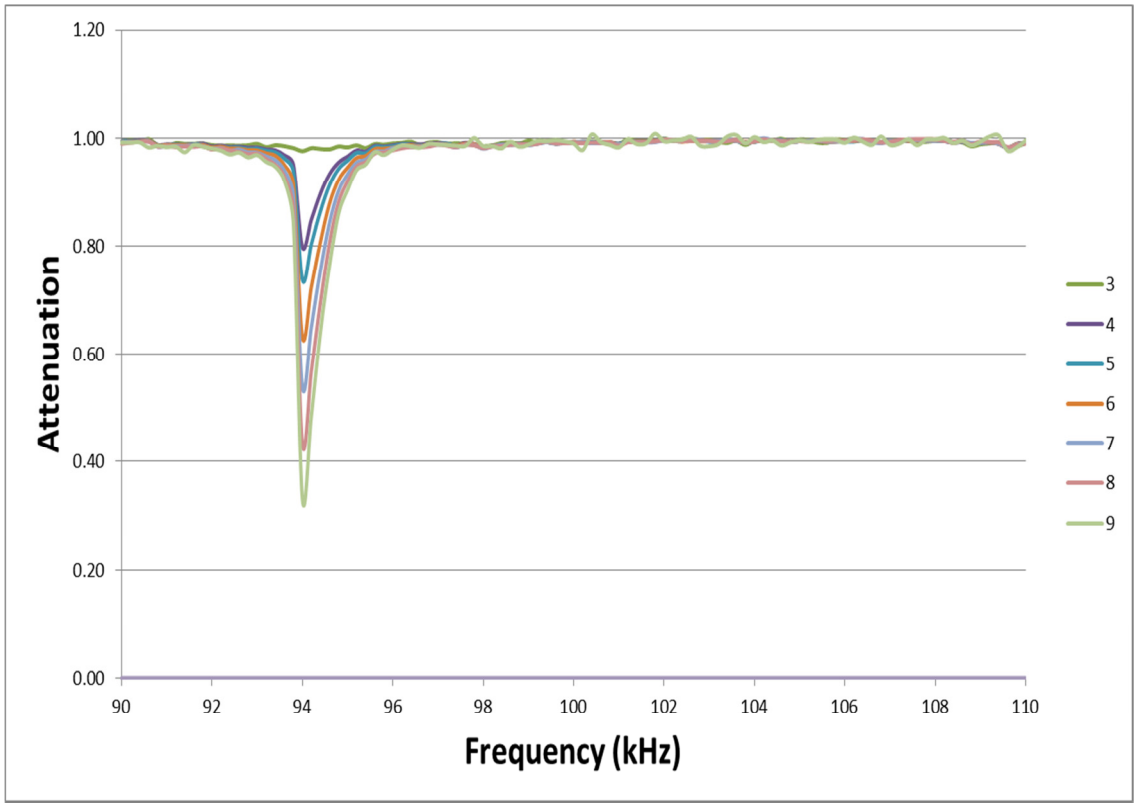


Figure 18: Normalized Attenuation of FFT Harmonics vs. Frequency

Closer look at the plot of FFT magnitudes vs. frequency for different harmonics (Fig. 17) shows that attenuation peaks are asymmetric: harmonic lines fall very sharp at the resonance frequency but their rise occurs at slower rate. These plots are obtained from micro scales tests discussed in Chapter 6 and this phenomenon can be explained by nonlinear resonance of MEMS devices. The description for this nonlinear behaviour is beyond the scope of this thesis but it can be modeled by the Duffing equation and discussed in Ellerington, et al. paper [1].

4.5: **Bessel Curve Fitting of Numerical Values**

Another section of the excel file, gives user the opportunity to pick three frequencies and compare the amount of associated attention for different harmonics. The plot which represents the resonance frequency shows highest amount of attenuation at higher order harmonics while others show little or no decline (Fig. 12-7). A plot of attenuation vs. harmonics for three different harmonics is shown in Figure 19. As expected, the highest amount of attenuation at higher order harmonics occurs for the resonance frequency.

Attenuations can be converted into displacement amplitude by using a Bessel curve. The excel program fits the harmonics to a Bessel curve. It finds the least squares fit Bessel function to the harmonic attenuations for each frequency blur phase.

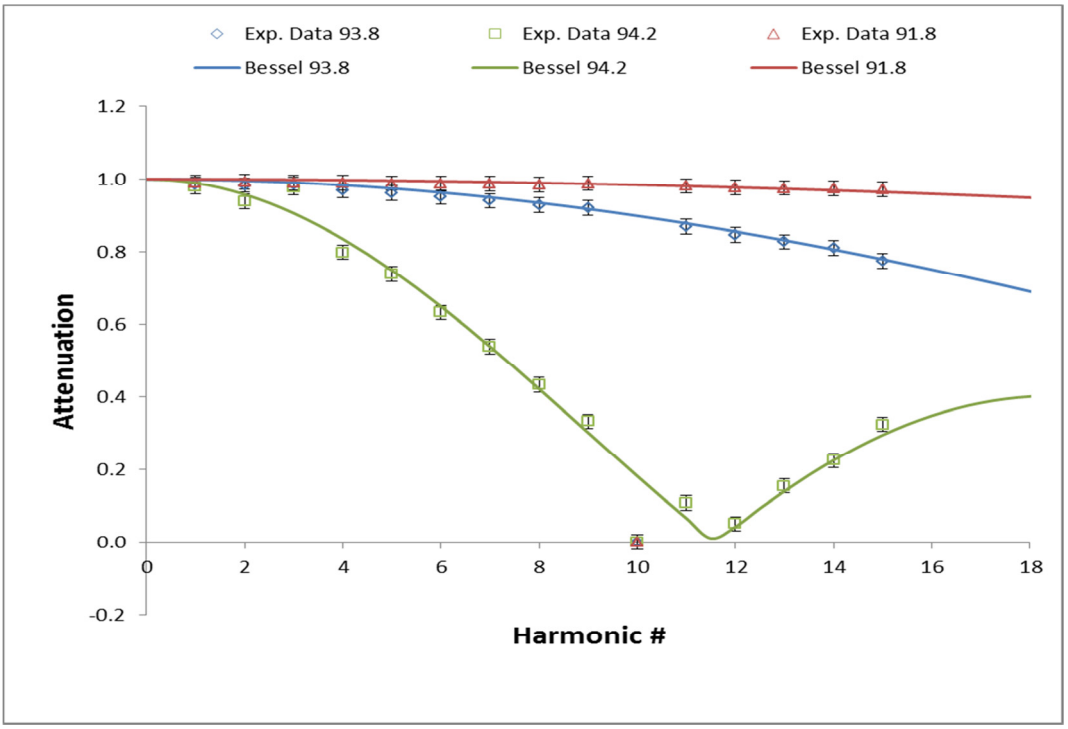


Figure 19: Plot of Attenuation vs. Harmonics for Three Frequencies: 93.8 kHz, 94.2 kHz and 91.8 kHz. Dots are Experimental Data and Lines Show Best Fit for Bessel Curve Absolute Value

4.6: **Conversion to Displacement**

Based on the Bessel curve fitting, the blur phase is then converted from radians to a multiple of the primary spatial wavelength (λ_1) as shown in Figure 20 [1]. In displacement measurements, Signal to Noise Ratio (SNR) as defined by equation (23) is the height of the resonance peak above the higher frequency range divided by twice the standard deviation or noise in the same range. Figure 20 provides a good illustration for definition of SNR (in this sample SNR=32 and high frequency value starts at 105 kHz).

$$SNR = \frac{Resonance\ Peak}{2 * \sigma_{High\ Freq.\ Noise}} \tag{23}$$

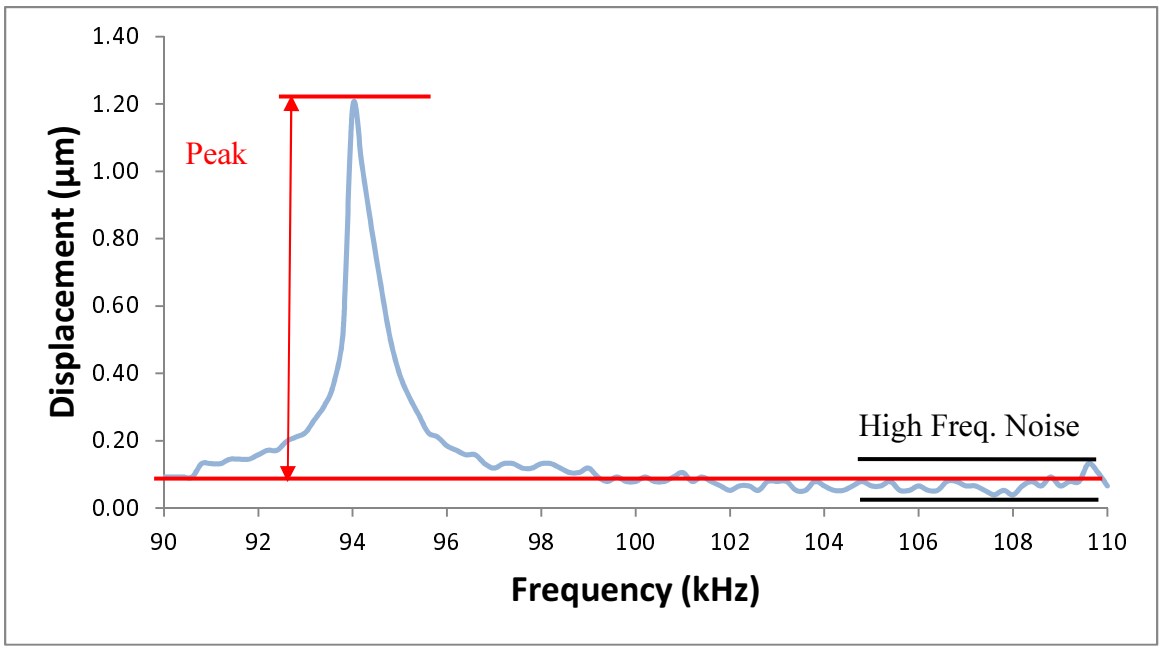


Figure 20: Plot of Displacement vs. Frequency (SNR=32)

Chapter 5: Macro Scale Tests

5.1: Experimental Setup

After describing theory, mathematics and extraction process for the FFT based measurement method, main part of research starts from this chapter. It includes developing a macro scale model to investigate effect of different parameters on measurement results and then using different techniques based on previous findings to improve results in micro scale model. Figure 21 shows the configuration of the macro scale setup. It contains: (1) A Function generator, (2) An amplifier that amplifies the signals from the function generator before sending them to the vibrator, (3) A vibrator as a source of vibrations with an aluminum structure and paper comb attached to it, (4) A Point Grey Grasshopper CCD camera with a resolution of 1280×960 pixels connected to PC via FireWire (IEEE 1394) cable [23], (5) A LabVIEW program that controls the function generator for frequency sweeping with specified frequency range and steps.

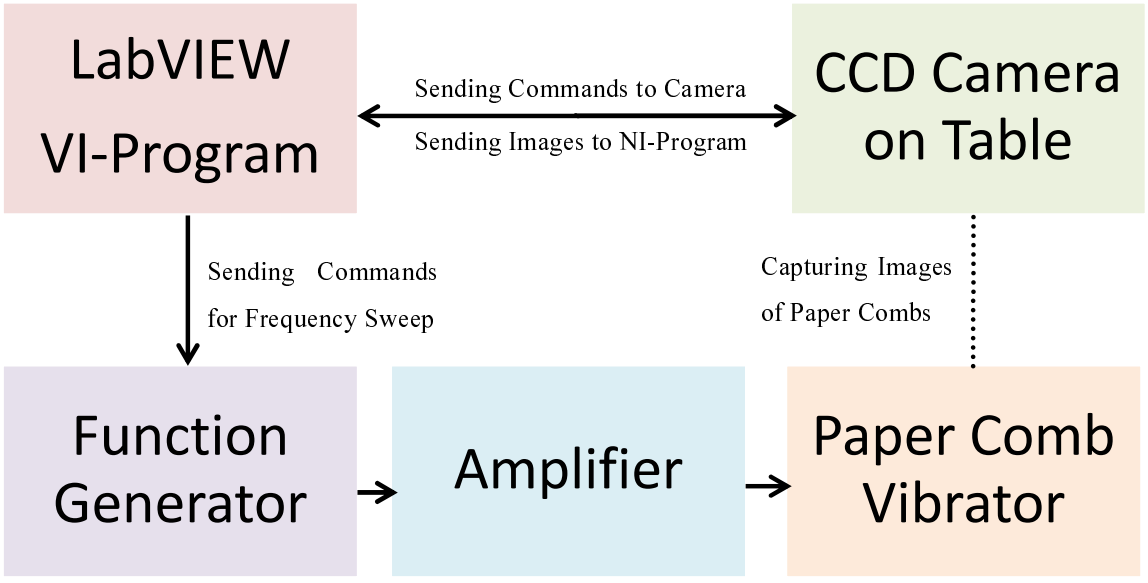


Figure 21: Flowchart Showing Configuration of Macro Scale Test Setup

The camera was mounted at 93 cm distance from the paper comb on a vibration isolation table to achieve more precise measurements. Figure 22 shows the main components of macro scale setup. The function generator (not shown here) connected to amplifier, produced sinusoidal waveform with 3.5 V peak to peak Voltage (VPP). The frequency sweep was performed within 250-350 Hz range with 1 Hz steps. Figure 23 shows screen Shot of LabVIEW program.

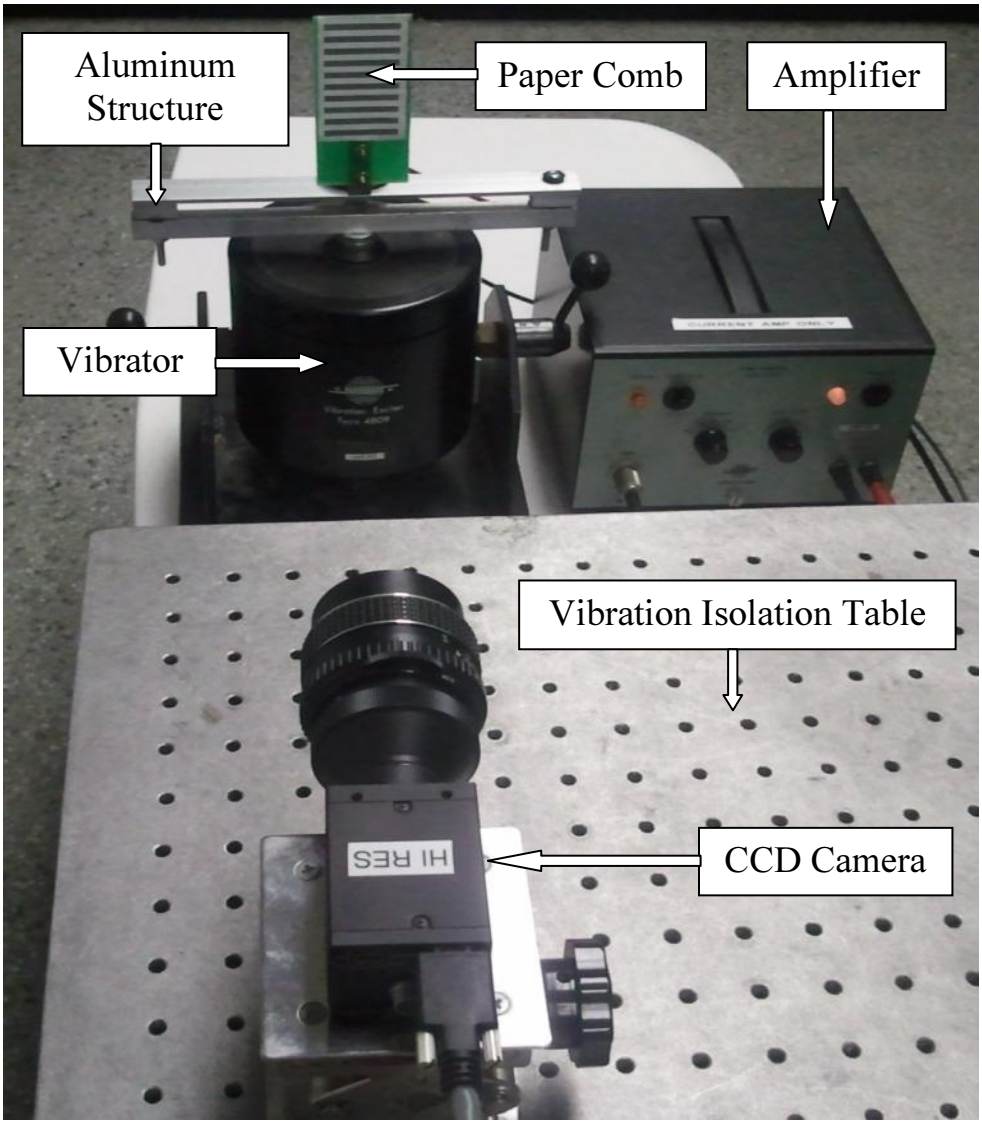


Figure 22: Part of Macro Scale Setup Including: Paper Comb and its Holder, Vibrator, Amplifier and CCD Camera Connected to PC via FireWire Cable

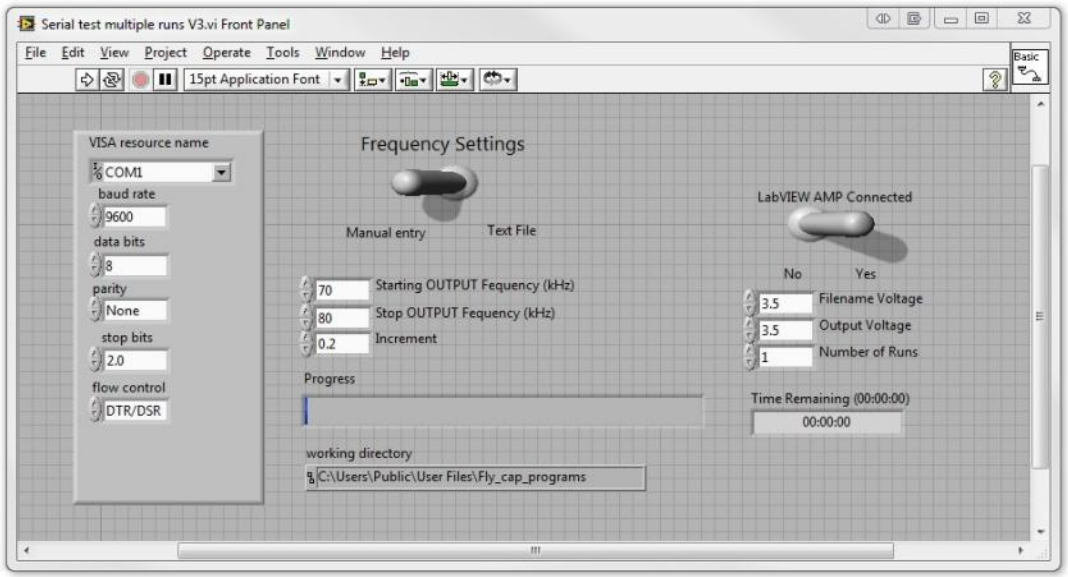


Figure 23: Screen Shot of VI Program Created with LabVIEW for Simultaneous Control of Function Generator and CCD Camera

5.2: Changing Parameters

A series of tests were performed to identify the effect of paper comb geometrical parameters on SNR. To achieve more consistent results and minimize the impact of environmental parameters, each test was performed three times and averaged the numerical results (stored in CSV files) before further analysis and calculation of SNR.

5.2.1: Number of Columns

In the first test, the number of selected columns was changed from 2-8 in a paper comb with 4.0 mm width columns as shown in Figure 24. Our prediction was that adding extra columns would improve SNR readings by reducing the noise and increasing sensitivity in two ways: (1) Improving the column averaging of captured images (2) Increasing the number of spatial wavelengths which means more accurate FFT results. Figure 25 shows the results. While the predicted trend was true, the increase in SNR at

the beginning was higher than later. For example, increase from 2-4 columns improved SNR by 35% while increase from 6-8 had only 9% positive effect. Therefore after a certain numbers (6-8), addition of extra columns did not contribute much to improving column averaging and producing more accurate FFT results.

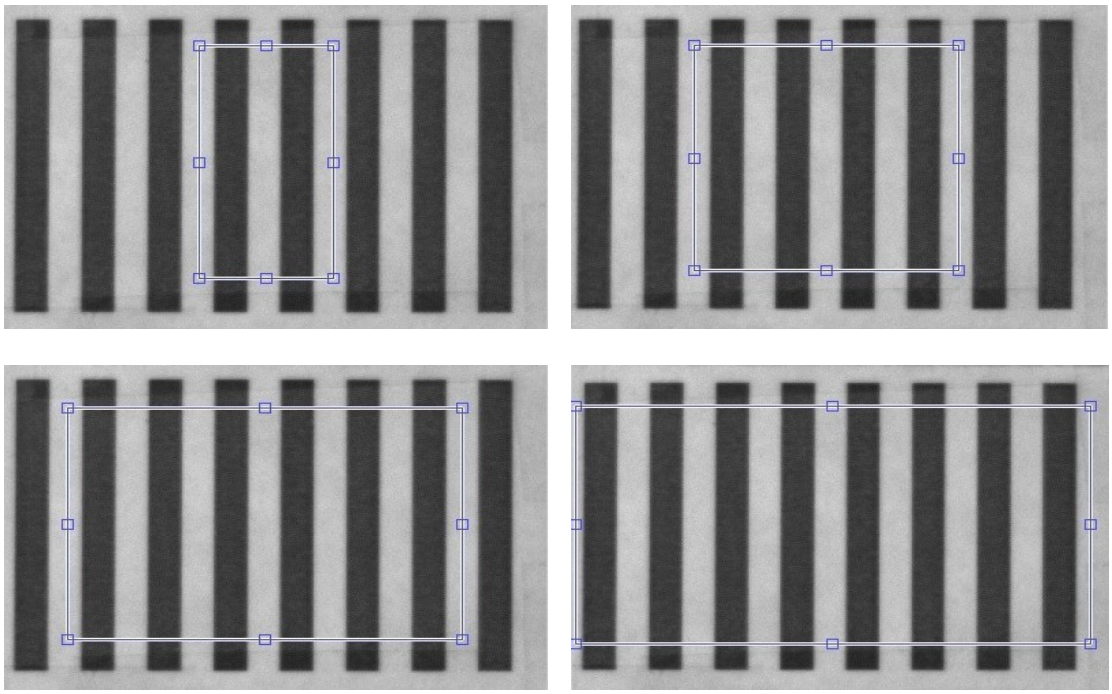


Figure 24: Region of Interests Containing 2, 4, 6 and 8 Columns

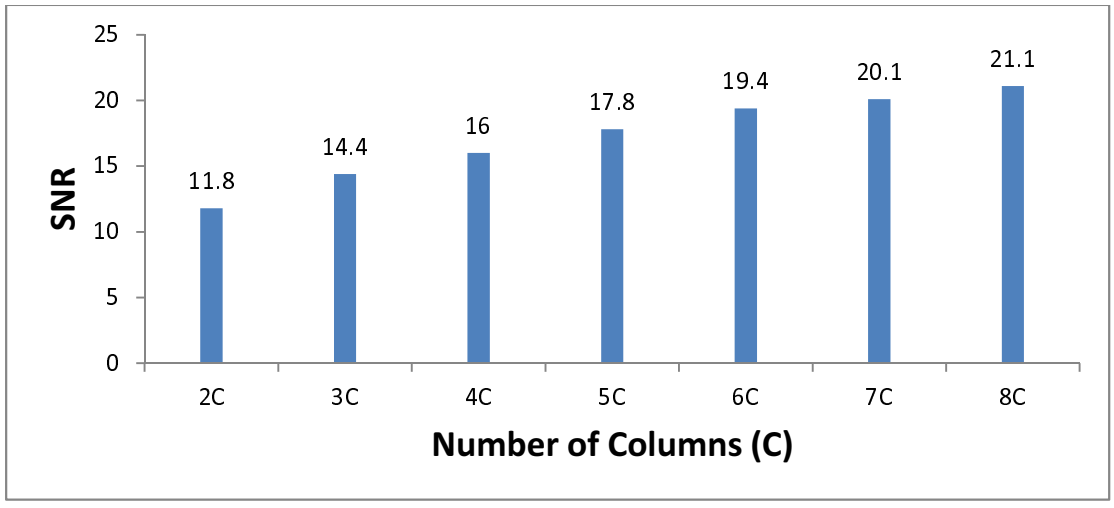


Figure 25: SNR Values based on Number of Selected Columns from 2-8C (Columns)

Figure 26 and 27 show plots of displacement vs. frequency for selection of two and eight columns. The lower plot which was created by selection of eight columns has sharper peak and less noise in high frequency values therefore it has higher SNR value.

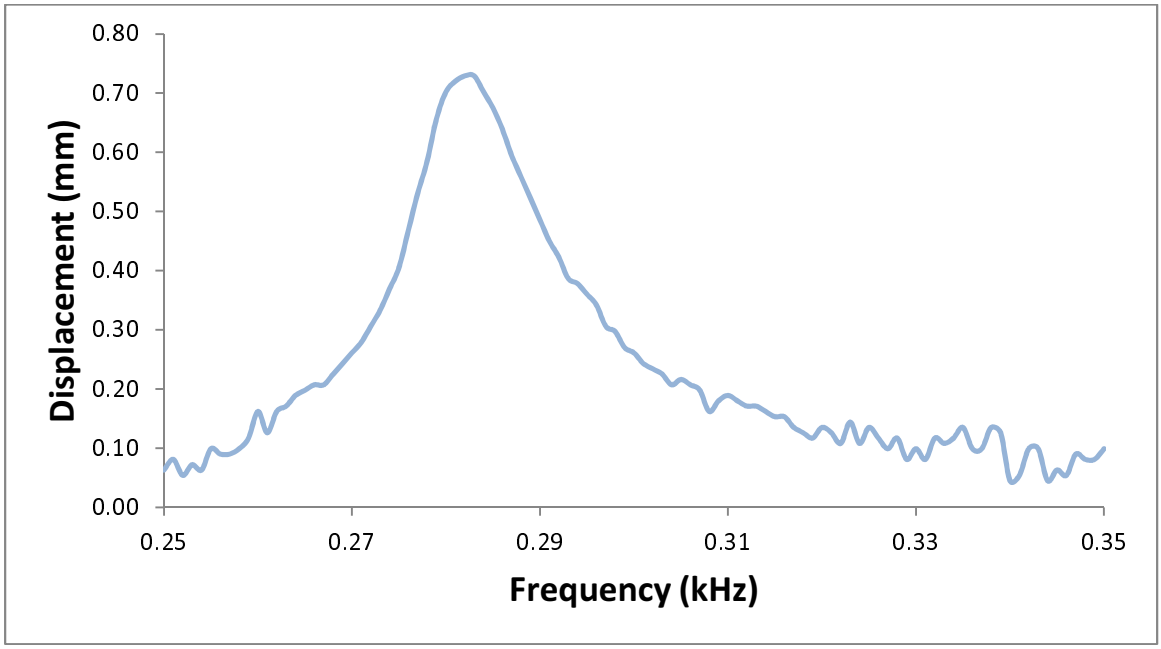


Figure 26: Plot of Displacement vs. Frequency for Selection of Two Columns (SNR=11.8)

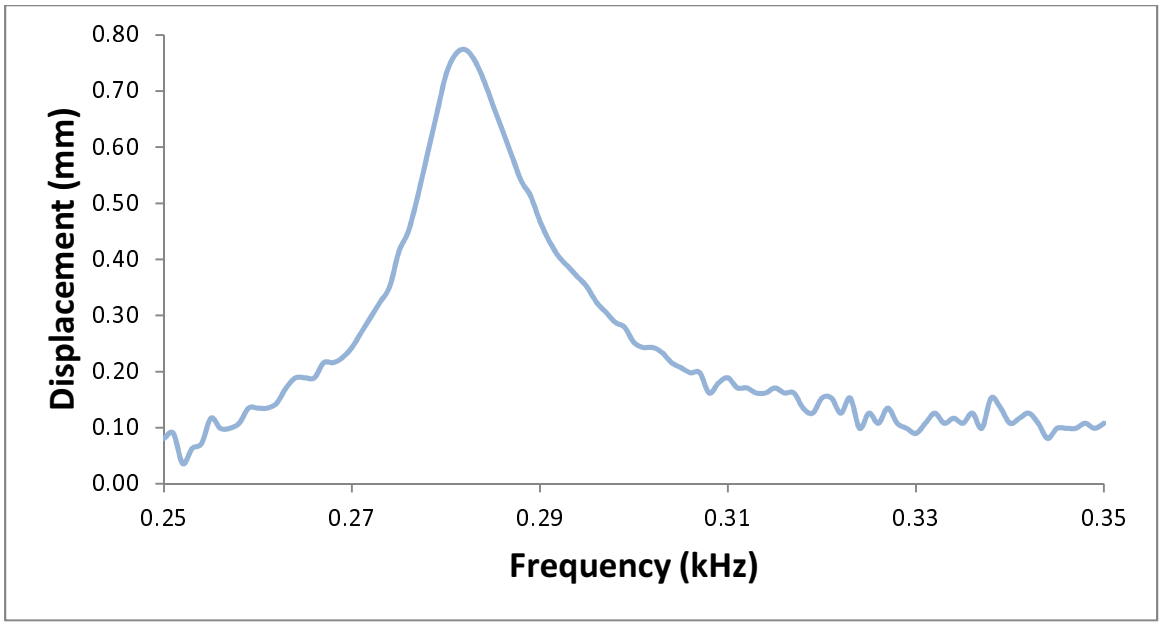


Figure 27: Plot of Displacement vs. Frequency for Selection of Eight Columns (SNR=21.1)

5.2.2: **Columns Height**

The height of ROI was reduced from the highest possible selectable area (100%) to 5, 10, 25, 50 and 75 percent of original amount as shown in Figure 28. Even at the maximum height, some space was left at the very top and bottom of papers as those areas were covered with transparent but glossy tape which could affect final results.

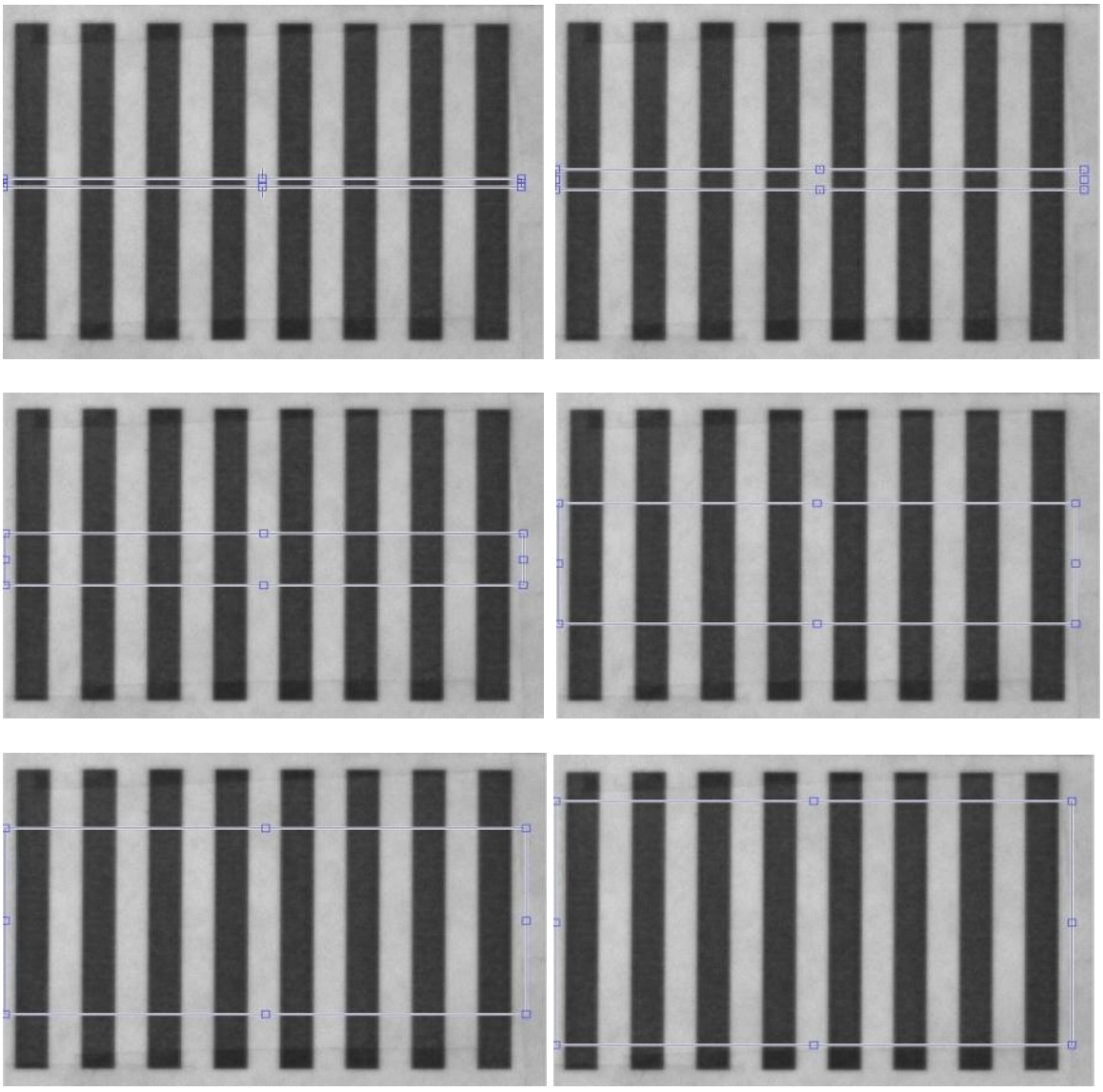


Figure 28: Region of Interests with Different Comb Lengths as Percentage of Maximum Selected Length: 5%, 10%, 25%, 50%, 75% and 100%

Figure 29 shows the obtained results. Selected heights showed 65%, 51%, 32%, 30% and 8% reduction in SNR readings in comparison with full length: narrower selection reduces SNR values. The amount of SNR was higher in wider selection areas because it produced better row average and subsequently more precise FFT calculations.

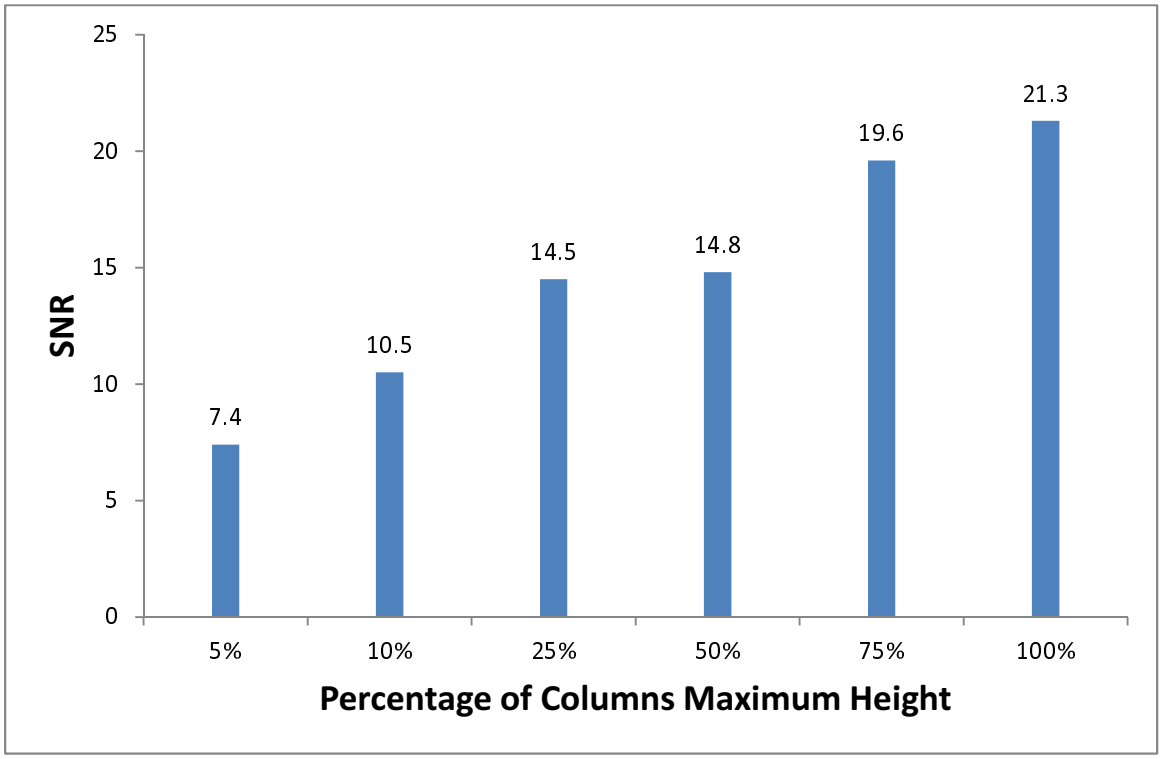


Figure 29: SNR Readings Based on Percentage of Columns Maximum Selected Length

5.2.3: **Columns Width**

In this test, the column width was changed from 1.0-10.0 mm as shown in Figure 30. By decreasing the width, more columns could fit into selected area as ROI in each test was the same. Therefore based on results from the first test, smaller widths should show better results. Figure 31 shows the plot of SNR readings versus columns width.

Decreasing the width from 10 mm to 6-4 mm increased SNR, however further reduction in width did not follow the same trend. This behaviour can be explained by detection of fundamental wavelengths with program. As widths reduced in size, the difference between black and white regions in paper comb became less apparent and their detection became more difficult which added noise to readings. Considering the above effect and results from the first test that addition of extra columns beyond 6-8 does not improve SNR readings, gradual decrease in SNR was observed as width decreased. It can be concluded that increase in number of columns only improves the overall results as long as the difference between dark and bright regions remain clear.

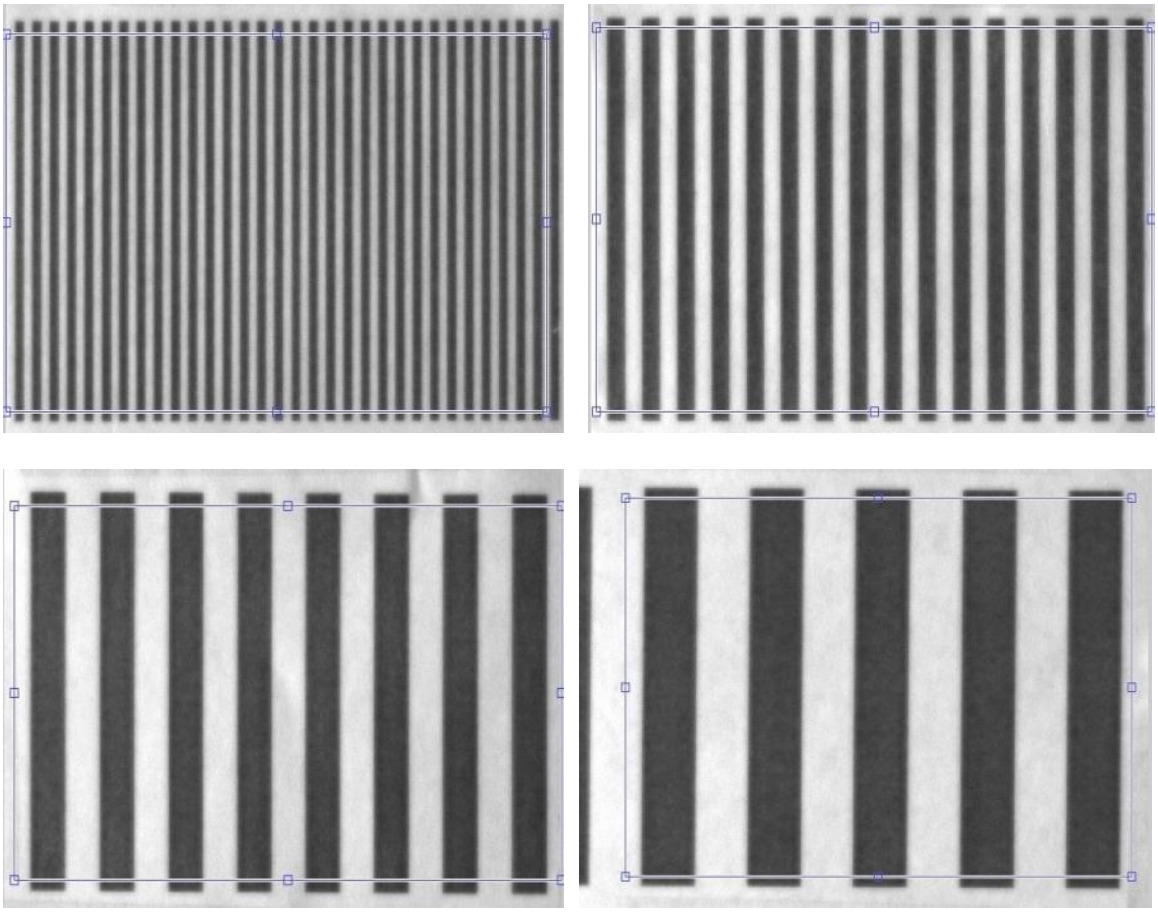


Figure 30: Paper Combs with Different Column Widths: 1, 2, 4 and 6 millimeters

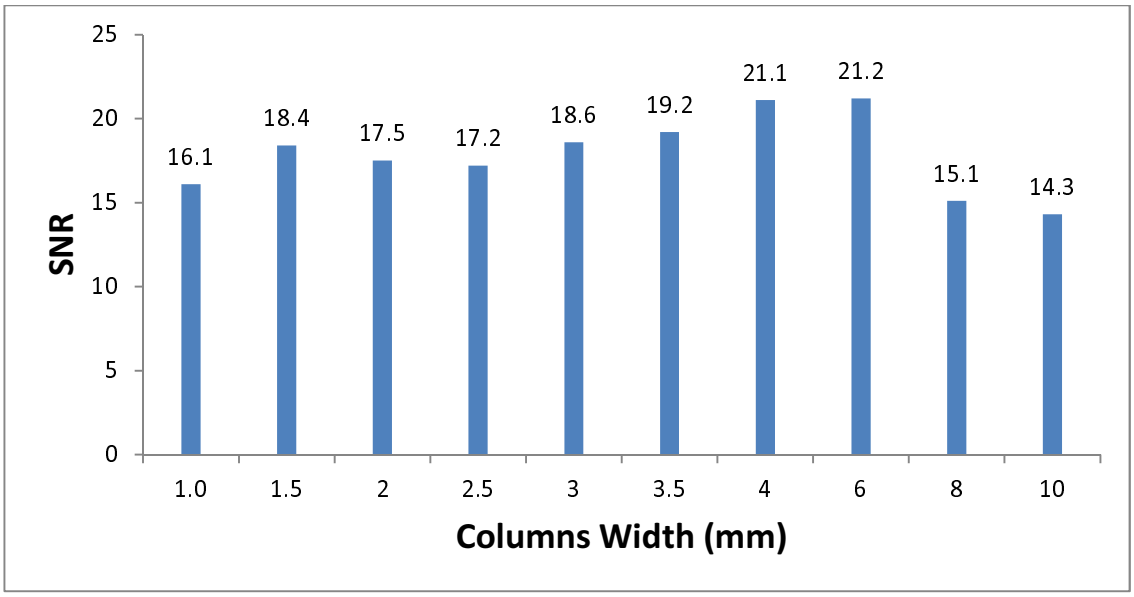


Figure 31: SNR Values for Each Column Width

5.2.4: **Reference Comb**

In order to detect and remove ambient and translation noise from readings, a static reference comb was set up next to the original vibrating paper comp with a small gap between them. Figure 32 shows an overview of the setup. The holder of reference comb was attached to the vibration isolation table similar to the camera. The camera was also adjusted so it could take pictures of both static and moving combs simultaneously as shown in Figure 33. By doing this, only one series of images would be processed by the program for FFT analysis and SNR calculation of static and moving combs. In one run the static comb was selected and in another run, the moving comb. SNR values for moving comb and static one found to be 19.7 and 1.6 respectively. The very low value for the second one indicated that it represents ambient noise. The slightly lower value for moving comb in comparison with other tests was related to narrower selected area and effect by the nearby static structure.

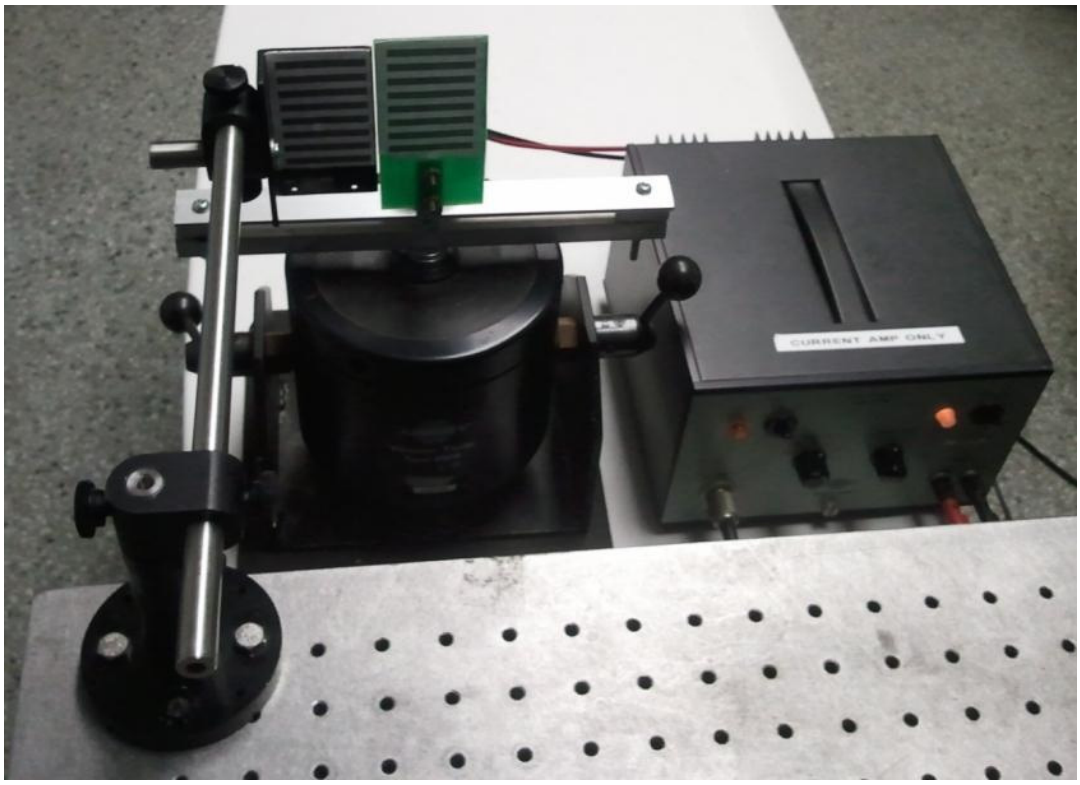


Figure 32: Close View of Reference Comb Setup on Vibration Isolation Table

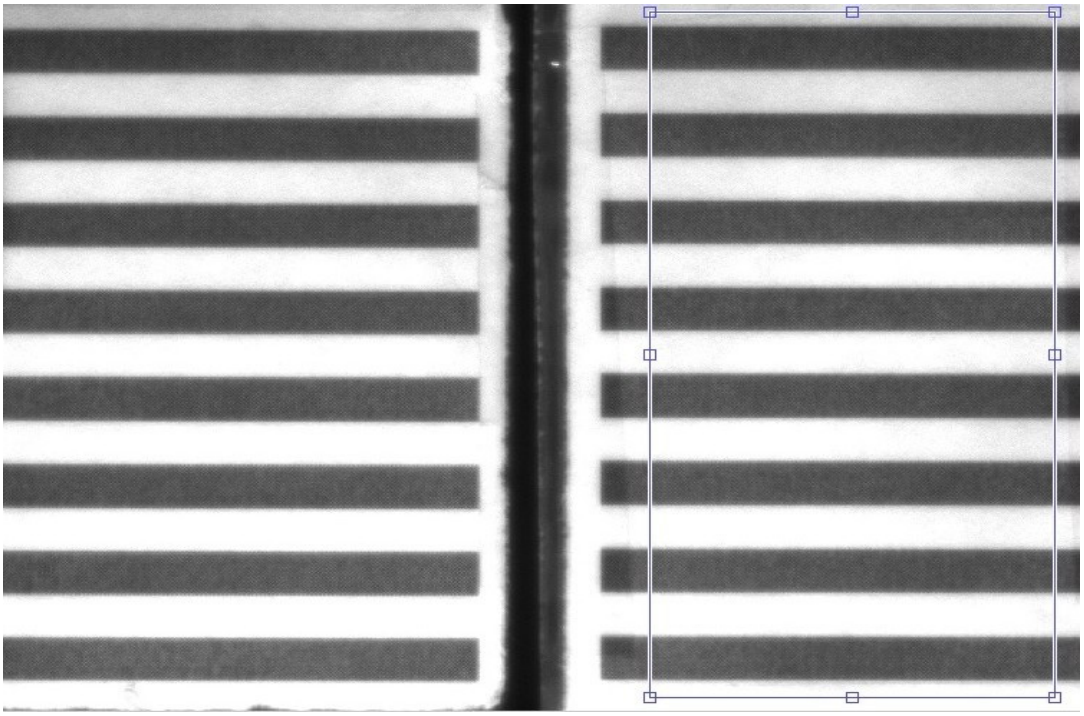


Figure 33: Moving Comb (Right) along with Static Reference Comb (left) both with 4 mm Columns

Figure 34 shows the plots of displacement versus frequency for moving comb, static comb and the difference along with their associated SNR values. The green plot "The Difference" had 21% higher SNR value than the blue plot "Moving Comb", this test showed positive effect of reference comb as expected.

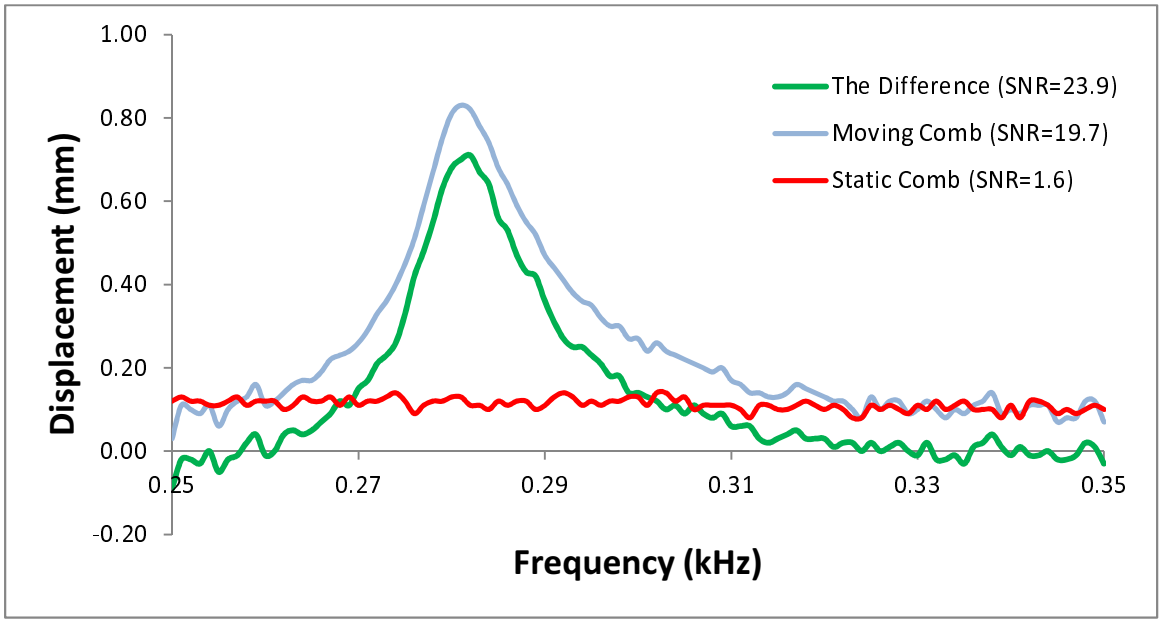


Figure 34: Plot of Displacement vs. Frequency for Reference Comb Setup

5.2.5: **Duty Cycle**

In the last test, the effect of columns duty cycle was investigated. Figure 35 shows created paper combs with column width of 4.0 mm and duty cycle of 25%, 37.5%, 62.5% and 75% in addition to original one (50%). Figures 36 and 37 show the resultant row average of paper combs with 25% and 75% duty cycle. Figure 38 shows the test final results: duty cycle of 50% showed highest SNR. This could be attributed to the balance between widths of black and white regions and the best possible configuration in resultant row average of 2D images which was 50% up and 50% down.

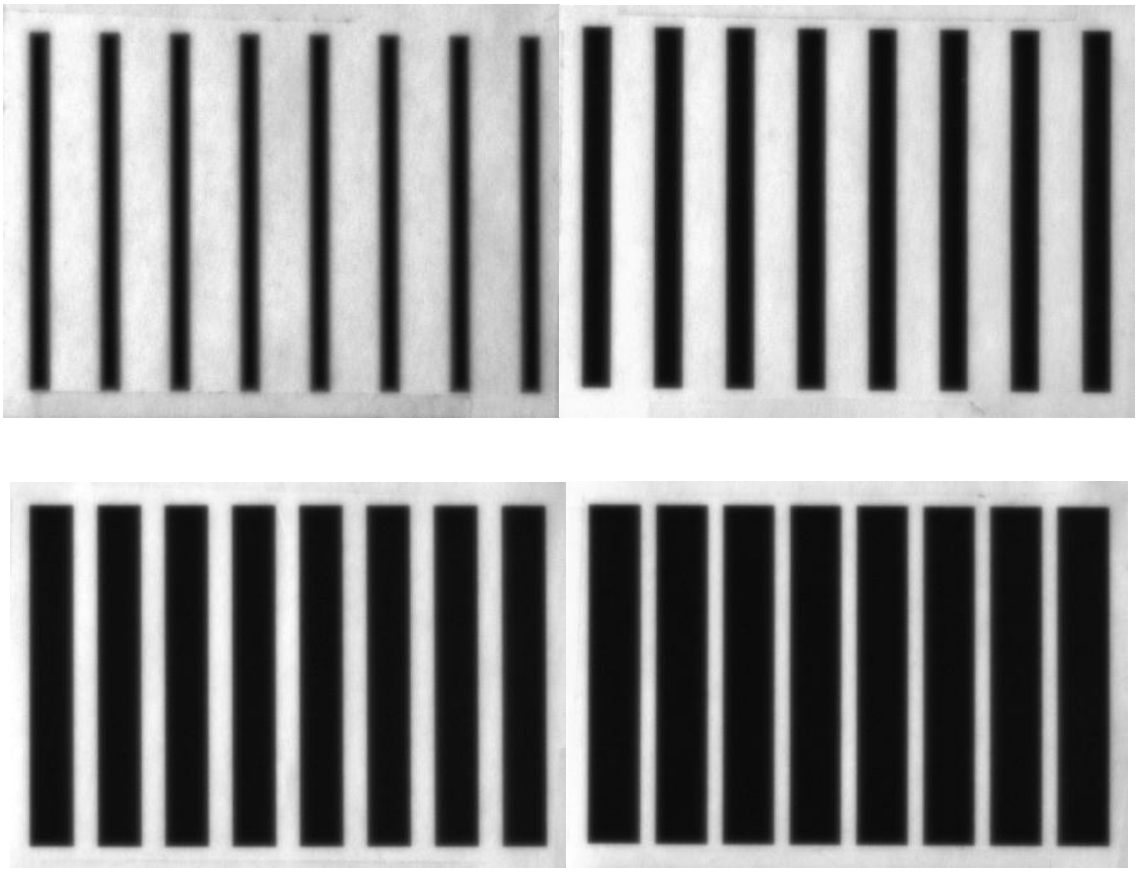


Figure 35: Four Paper Combs with 4mm Column width and 25%, 37.5%, 62.5%, 75% Duty Cycle

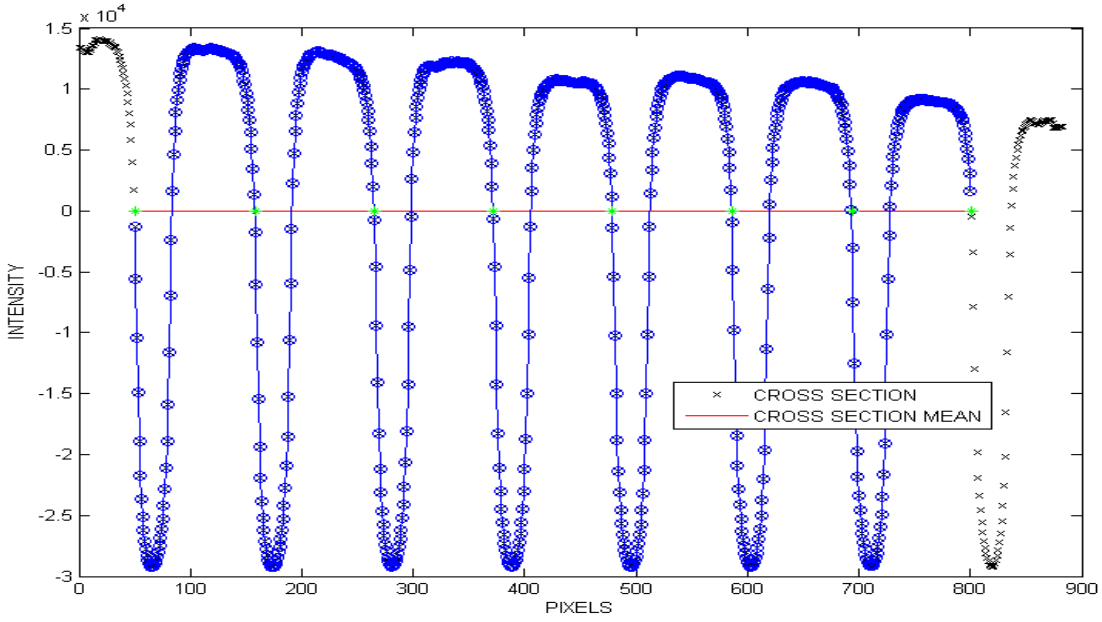


Figure 36: Row Average of Paper Comb with 25% Duty Cycle

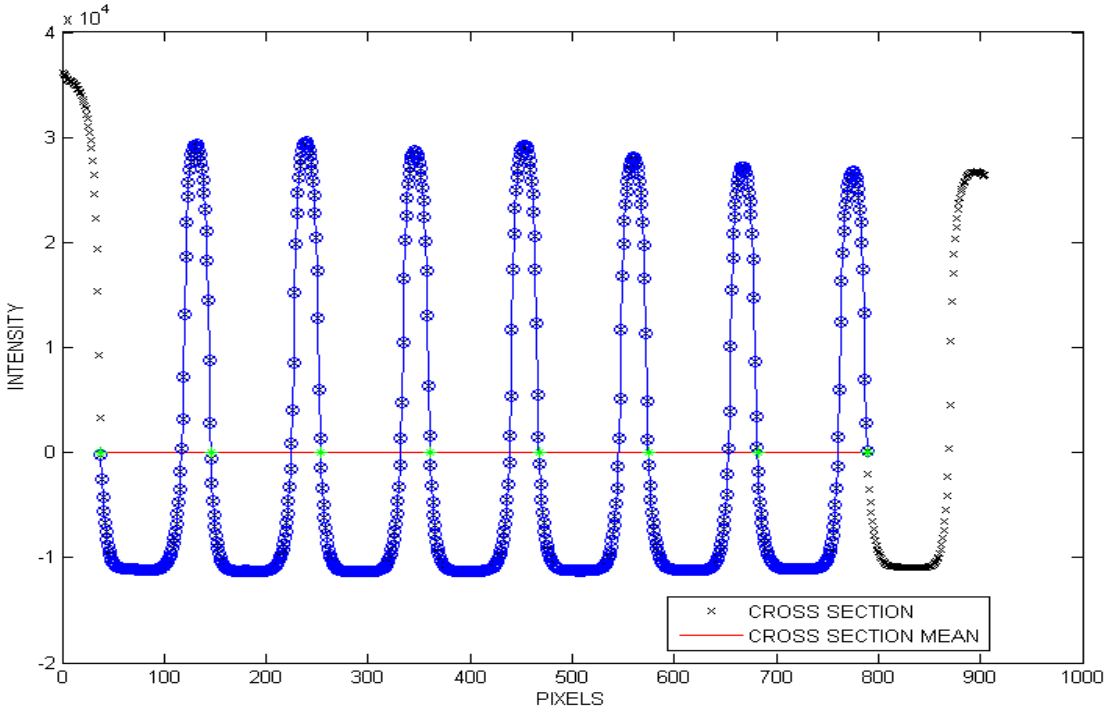


Figure 37: Row Average of Paper Comb with 75% Duty Cycle

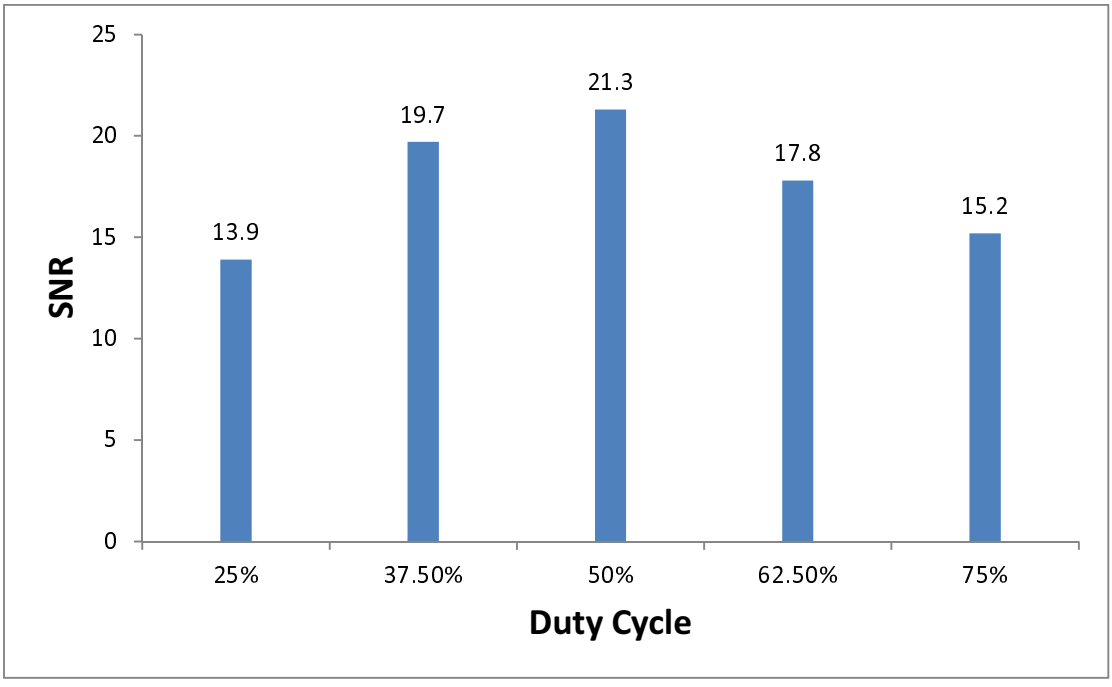


Figure 38: SNR Values Associated with Different Duty Cycles

5.3: Summary, Recommendations and Analyses

5.3.1: Summary

Table 2 shows summary of macro scale tests results. Each experiment was identified with rows of specific color showing parameters that was changed in each test.

Table 2: Summary of Macro Scale Model Test Results

Test #	# Columns	Height (%)	Ref Comb	Width (mm)	Duty Cycle (%)	SNR
1	2	100	N	4	50	11.8
2	3	100	N	4	50	14.4
3	4	100	N	4	50	16.0
4	5	100	N	4	50	17.8
5	6	100	N	4	50	19.4
6	7	100	N	4	50	20.1
7	8	100	N	4	50	21.1
8	8	5	N	4	50	7.4
9	8	10	N	4	50	10.5
10	8	25	N	4	50	14.5
11	8	50	N	4	50	14.8
12	8	75	N	4	50	19.6
13	8	100	N	4	50	21.3
14	8	100	Y (M)	4	50	19.7
15	8	100	Y (S)	4	50	1.6
16	8	100	Y (M-S)	4	50	23.9
17	32	100	N	1	50	16.1
18	22	100	N	1.5	50	18.4
19	16	100	N	2	50	17.5
20	13	100	N	2.5	50	17.2
21	11	100	N	3	50	18.6
22	10	100	N	3.5	50	19.2
23	8	100	N	4	50	21.1
24	6	100	N	6	50	21.2
25	4	100	N	8	50	15.1
26	3	100	N	10	50	14.3
27	8	100	N	4	25	13.9
28	8	100	N	4	37.5	19.7
29	8	100	N	4	50	21.3
30	8	100	N	4	62.5	17.8
31	8	100	N	4	75	15.2

5.3.2: **Recommendations and Analyses**

The macro scale test results showed increase in number of columns generally improves results therefore it is recommended to add extra columns. This trend however, was true as long as the increase in columns number didn't affect the difference between black and white regions. Fitting more columns into limited area lowered the contrast between two regions, made detection of wavelengths difficult for image processing program and ultimately added noise to readings.

Higher heights in ROI were always desirable as they resulted more precise row averaging of profile. An addition of a reference comb helped to cancel background noise and improved results noticeably. Duty cycle of 50% provided highest SNR among other values. This can be explained by the most harmonic resultant row average.

Chapter 6: Micro Scale Tests

6.1: Experimental Setup

A testing method similar to macro scale model was also used for micro scale tests. In this model however, the CCD camera was mounted on a probe station equipped with an optical microscope to capture images of MEMS vibrating pad. Figure 39 shows overview of the setup. The output signal from amplifier was sent to an oscilloscope in addition to distribution box so VPP as well as the frequency could be measured.

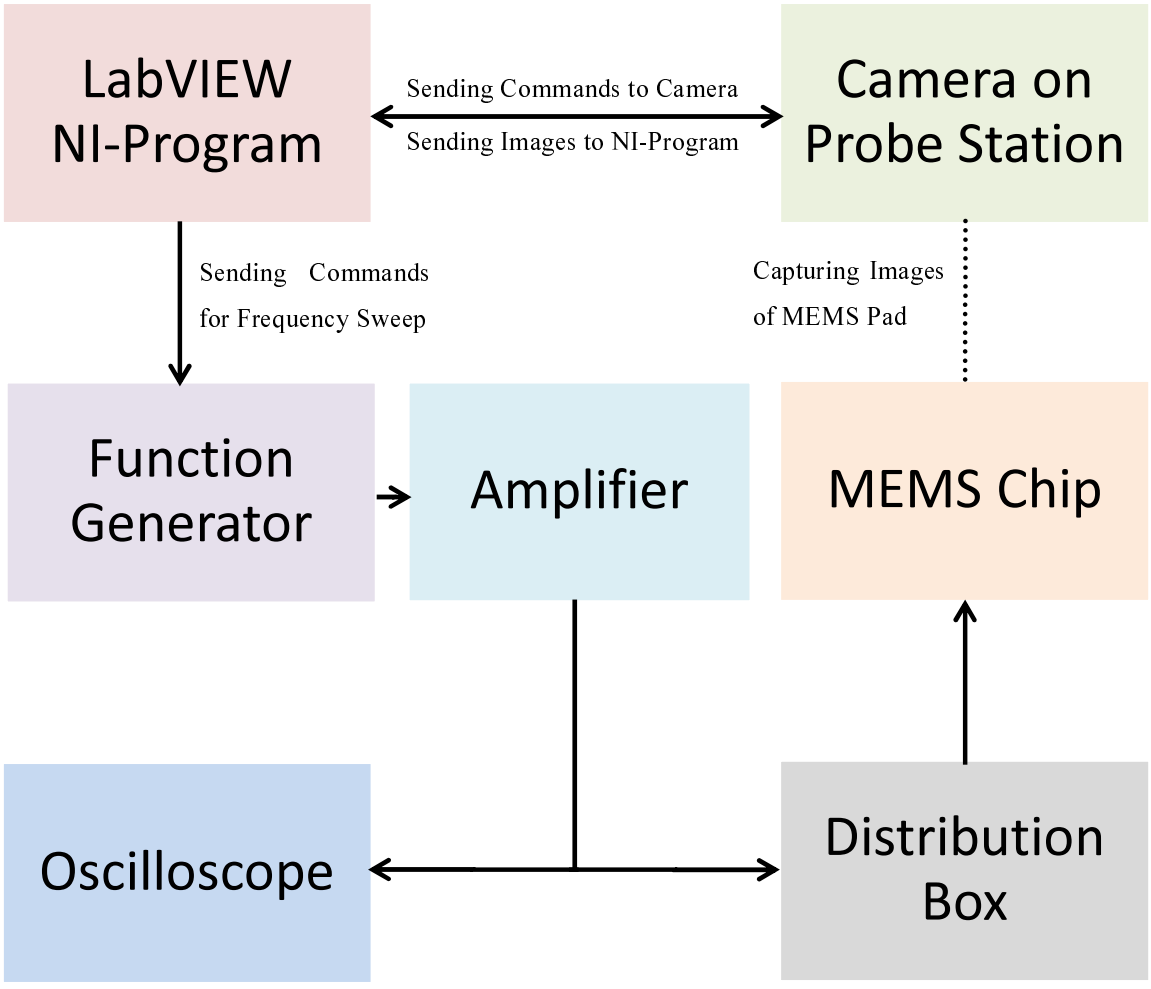


Figure 39: Flowchart Showing Configuration of Micro Scale Test Setup

Figure 40 shows the front view of probe station and its components. The MEMS chip with PGA packaging was inserted into ZIF socket. 68 pins of MEMS chip were connected to distribution box by two ATA cables. Probe station zoom level was set to 50X to take close up images of MEMS pad. The same camera used in the macro scale tests was mounted on top of probe station and connected to PC with FireWire cable. Figure 41 shows an overview of oscilloscope, function generator and amplifier setup.

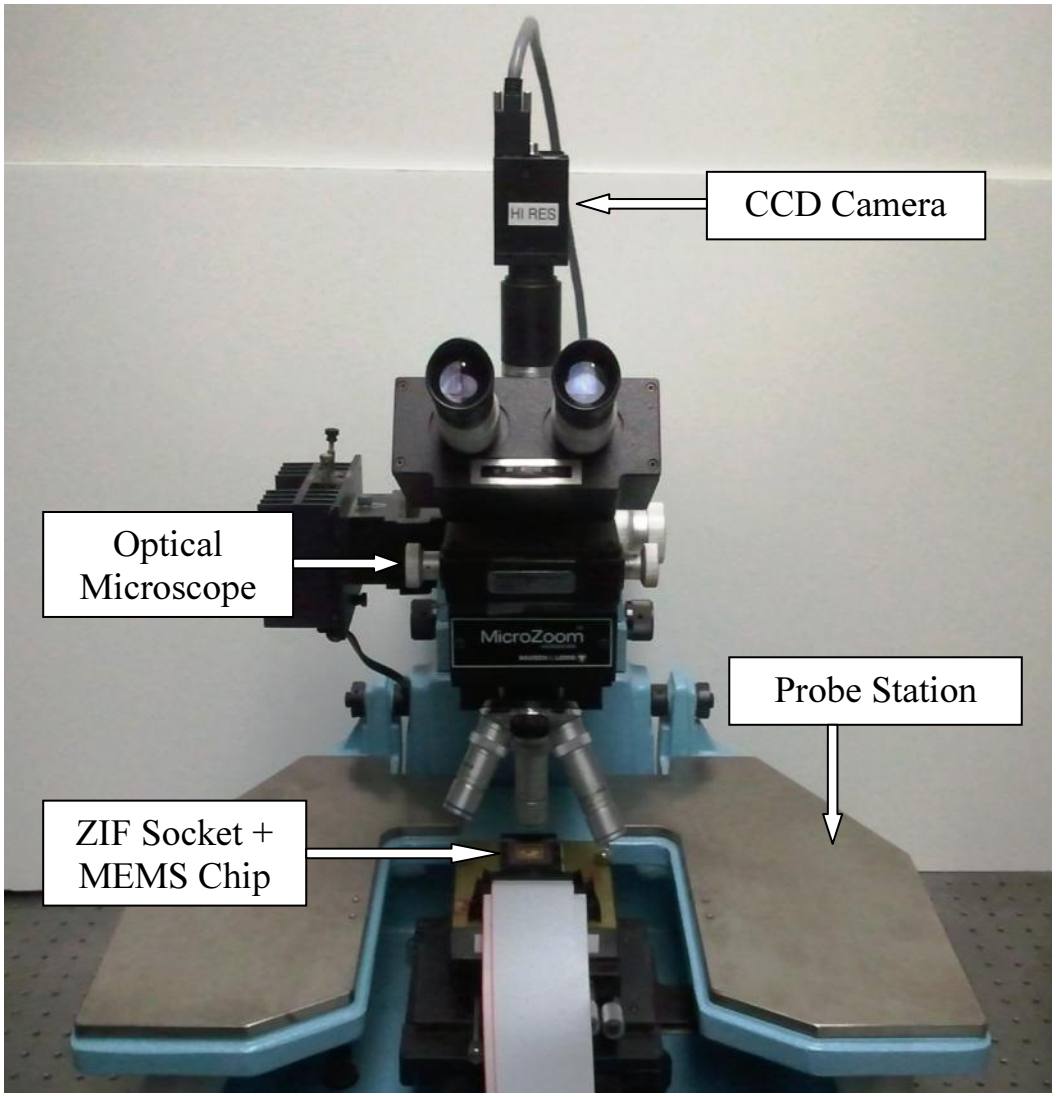


Figure 40: Part of Micro Scale Setup Including: Wentworth Probe Station, ZIF Socket with Installed MEMS Chip, Optical Microscope, and FireWire CCD Camera

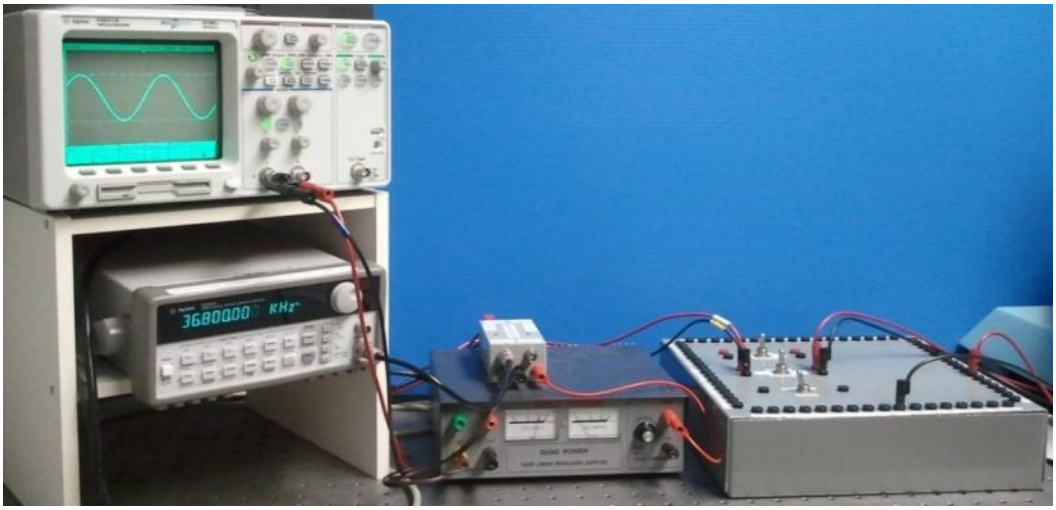


Figure 41: Part of Micro Scale Setup Including: Oscilloscope, Function Generator, Amplifier with its Power Supply and Distribution Box

The used MEMS device for micro scale tests was designed based on PolyMUMPs and fabricated by CMC [24]. Details about structure and its components are provided by Ellerington, et al. [1]. Figure 42 shows a micrograph of MEMS pad and its different parts. It is made of three main sections: (1) a central mass (2) a pair of chevron electrothermal actuators as source of vibration (3) two pairs of springs (S-type and paperclip) connected to anchors, actuators and central mass. The springs and chevrons are made of Poly1 layer while Poly1, Poly2 and Metal layers are used for centre pad to add mass. Both types of springs have high stiffness for out of plane deflection which keeps the central mass suspended. Similar to the macro scale model, each test was performed three times and then the numerical results were averaged. The frequency sweeping was performed within 90-110 kHz range and with 0.2 kHz steps. Relatively high VPP of 22V was applied to MEMS chip to compensate for high resistivity of its connections. Figure 43 shows a micrograph of central mass in static and at resonance. Figure 44 shows plot of displacement vs. frequency for MEMS pad.

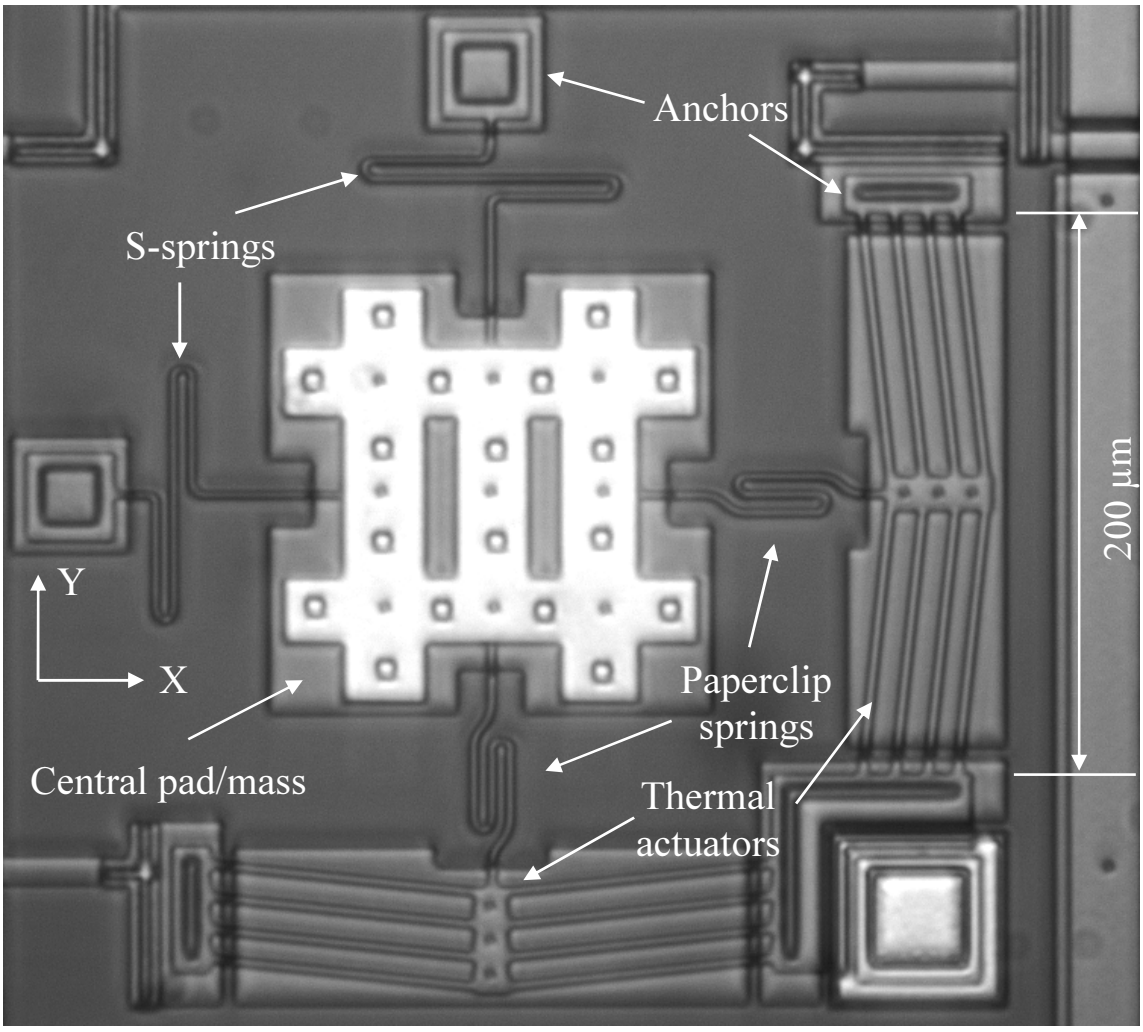


Figure 42: Micrograph of MEMS XY Pad Including: Central Mass, Paper Clip and S Type Springs, Anchors and Chevron Thermal Actuators

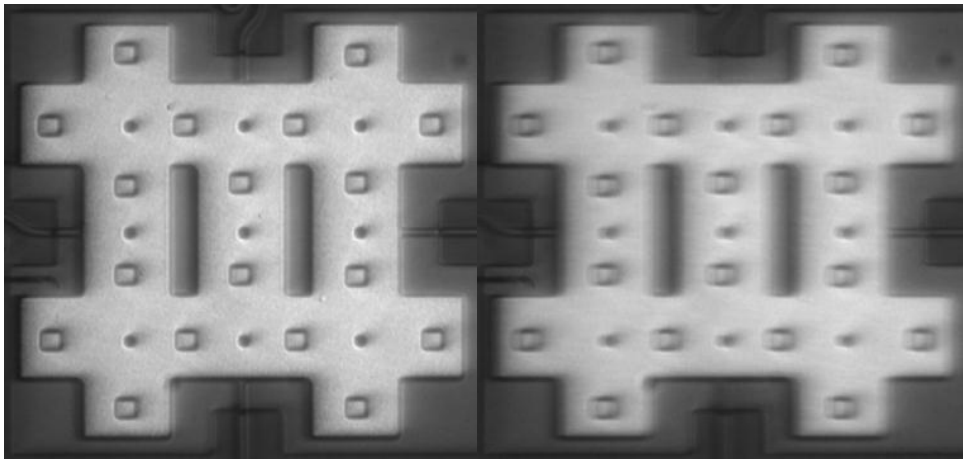


Figure 43: Micrograph of Central Mass in Static Mode (Left) and at Resonance at 94 kHz (Right)

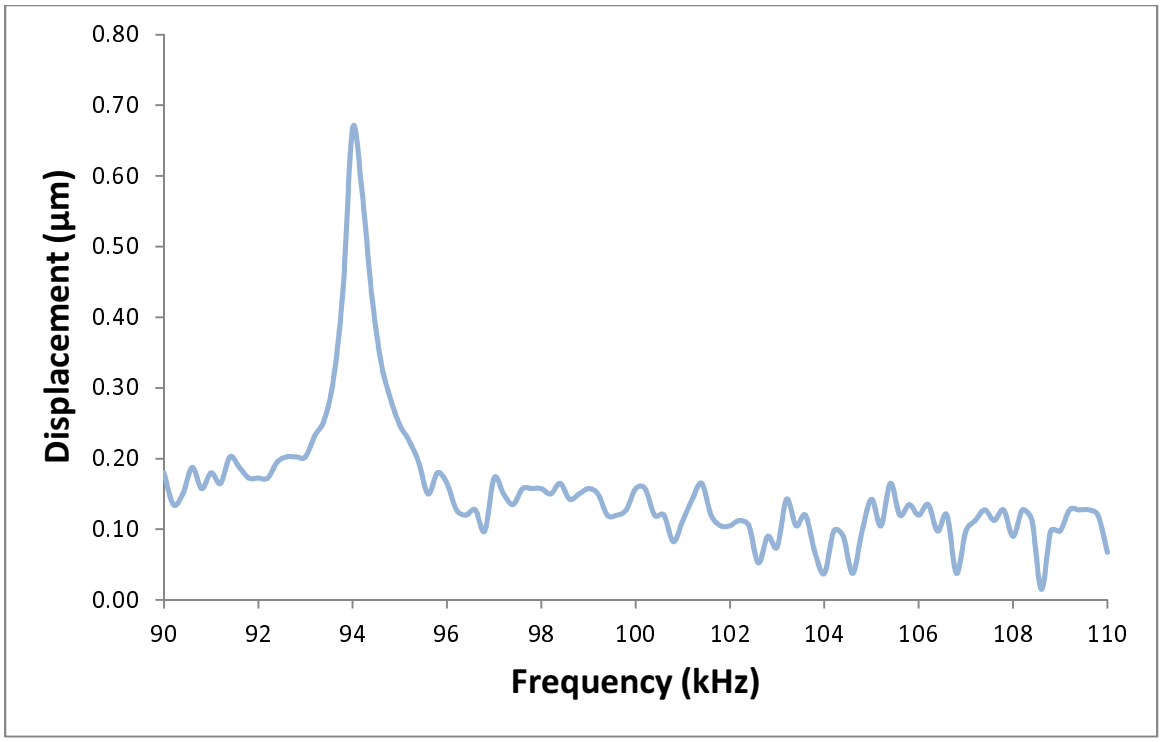


Figure 44: Plot of Displacement vs. Frequency for Micro Scale Model. Resonance Frequency occurs at 94 kHz

6.2: FIB Instrument

6.2.1: Overview

Based on findings in macro scale model, the creation of extra patterns in micro scale model should yield to better results in SNR readings. To examine this, Focused Ion Beam (FIB) instrument was used for etching and removal of material on surface of MEMS pads. The device model used for this experiment was *HITACHI FB-2000A* located in Institute for Research in Materials (IRM) [25]. Figure 45 shows front view of this instrument.

Figure 45: Front View of Focused Ion Beam (FIB) Instrument Used for Etching [25]

The FIB instrument uses highly focused beam of ions (usually gallium) for etching and removal of material on surface of specimen. Available beams in the device fall into two main categories: observation beams (M0) and etching beams (M1). The primary application of the first beams group is for scanning and imaging of specimen but even in this mode there is still some etching present. However, because of the very weak beams used for observation, the removal process is very slow and can be neglected.

Table 3 shows properties of beams with associated current and aperture diameter [26]. Etching beams with higher numbers have higher operational speed but their resolutions are low and therefore are suitable for coarse processing. Table 4 provides approximate etching speed of different beams for some sample areas [27]. Obviously for a specific type of beam and fixed amount of time, larger areas will result less etching depth.

Table 3: Default Beam Settings on FIB System

Beam Name	Lens Mode	Aperture Diameter (μm)	Beam Current (nA)	Beam Diameter (nm)
M1-500	Mode 1	500	11.0-15.0	1000
M1-300		300	4.0-8.0	250
M1-200		200	2.0-3.5	120
M1-100		100	0.400-0.800	60
M1-50		50	0.100-0.300	40
M1-20		20	0.015-0.040	35
M1-6		6	0.001-0.005	35
M0-50	Mode 0	50	0.020-0.050	60
M0-20		20	0.004-0.010	20
M0-6		6	0.000-0.002	10

Table 4: Approximate Etching Speed (µm/hr) for Different Beams and Areas

Area (µm^2)	Beam Type		
	M0-50	**M1-100**	**M1-500**
4×4	2.6	47	1500
64×64	0.01	0.18	5.8
128×128	0.002	0.05	1.5
256×256	0.0006	0.01	0.4

Figure 46 shows the user interface (UI) of FIB software. It must be noted, it shows part of chevron thermal actuator beams not relevant to MEMS pad discussed in this study. Software UI can be divided into four sections: (a) Fabrication window: in this section the user determines whether to import previously created etching patterns or to create geometry with device software and available tools. The areas that will be etched are indicated with orange color. (b) Viewing window: shows current view of specimen by device. The user can select continuous scanning which causes slow etching or single scan for quick view. Observation beam type, zoom level and scanning area are selected in this section. (c) Fabrication setup: fabrication (etching) settings and status are located in this section. The user determines dwelling time, frame number and etching beam. During the process, remaining time is displayed. (d) Column adjustment: user adjusts settings such as focal point, contrast and brightness ratio with sliders. Additional info related to beam status such as current and voltage are also displayed in this section.

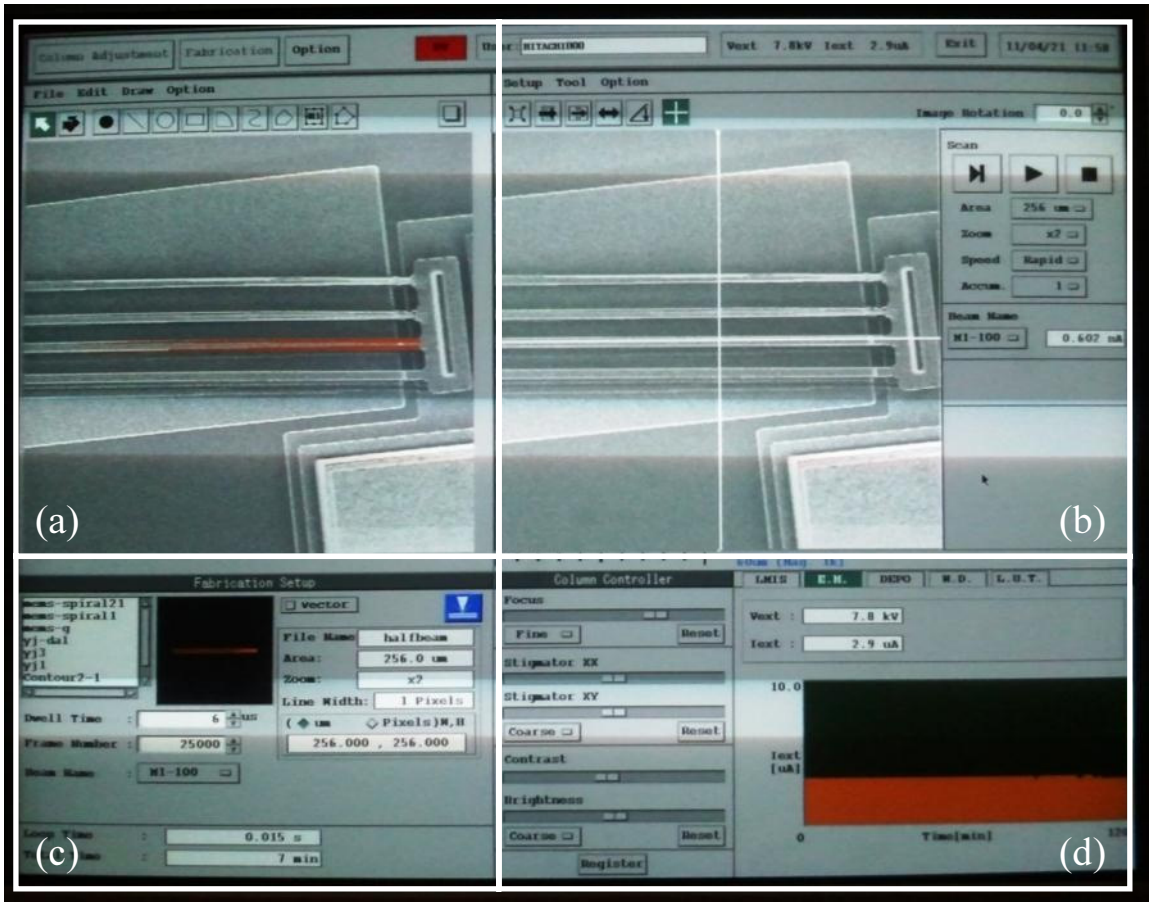


Figure 46: FIB Instrument Software User Interface: (a) Fabrication Window, (b) Viewing Window, (c) Fabrication Setup, (d) Column Adjustment

Figure 47 shows the top and side view of MEMS chip holder for FIB processing along with its dimensions. It is essentially a ZIF socket with metal plate and bolt of specific size glued to it. To secure the MEMS chip during the processing, the ZIF socket should be attached and locked to the FIB sample holder before sending it inside of the FIB vacuum chamber. The sample holder has five degrees of freedom: linear movements in X, Y and Z direction as well as rotation about X and Z axis. This manoeuvrability allows scanning and etching of different parts of sample.

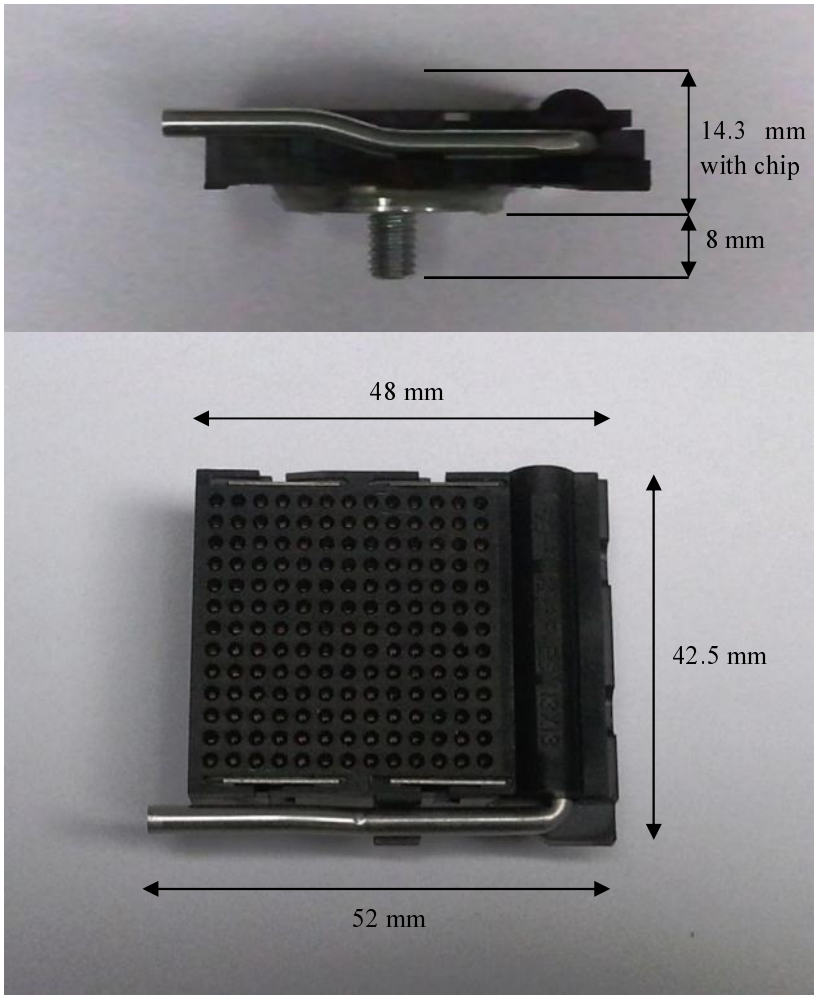


Figure 47: Side and Top View of MEMS Chip FIB Holder Made of ZIF Socket

6.2.2: **Creating Etching Patterns**

The built in drawing feature of FIB software enables the user to create desired etching patterns while using it, however it increases the total usage time of instrument (which is expensive) and lacks precise control over dimension and location of created geometry. Therefore, it is preferable to create and prepare etching patterns beforehand and load them into program immediately. For this purpose, the created layout should have specific properties in order to be readable by software.

The created image must be in BMP format, have resolution of 512×512 pixels and color depth of 1 bit (black and white) in which black areas indicate sections that has to be etched. Table 5 shows corresponding resolutions based on selected area and zoom level [28]. For example with 256×256 μm^2 scanning area and zoom level of ×1, each pixel on layout pattern represents approximately 0.5 μm in specimen. Therefore based on the mentioned settings, black rectangular area in layout image with dimension of 10×20 pixels will be mapped as etching area with dimension of 5×10 microns on specimen.

Figures 48 and 49 show a sample of such etching pattern and its result on a MEMS pad respectively. The aim in this example was to create 12 columns with dimension of 25×2 μm. All etching processes were performed with the largest possible scanning area (256×256 μm^2) and zoom level of 2×. Therefore each column in the layout had dimension of 100×8 pixels.

Table 5: Scanning Area, Zoom Level and Corresponding Resolutions

Area (μm^2)	Zoom Level	Resolutions (nm/pixel)
256×256	1	496
256×256	2	248
256×256	4	124
256×256	8	62
32×32	1	62
32×32	2	31
32×32	4	15.5
32×32	8	7.75
4×4	1	7.75

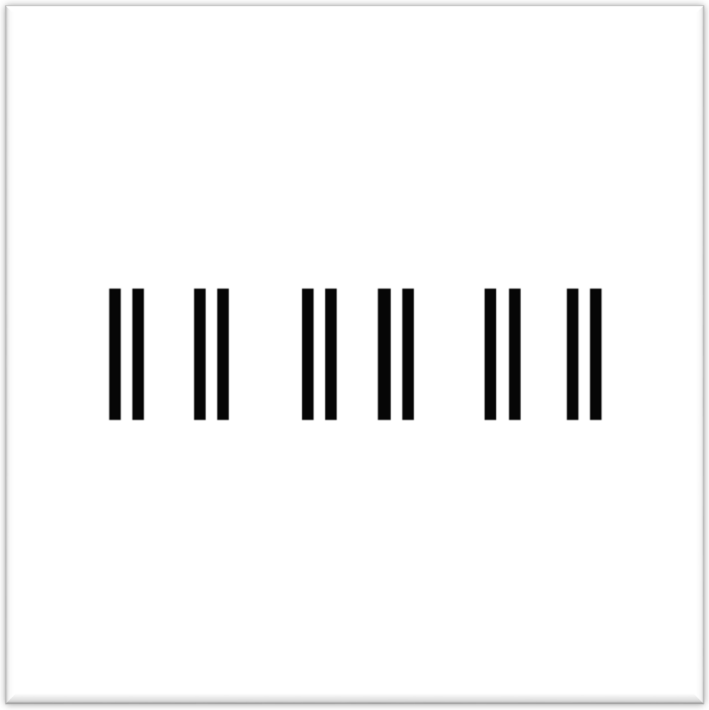


Figure 48: Sample of Created Etching Pattern. Boundaries Represent 512×512 Pixels

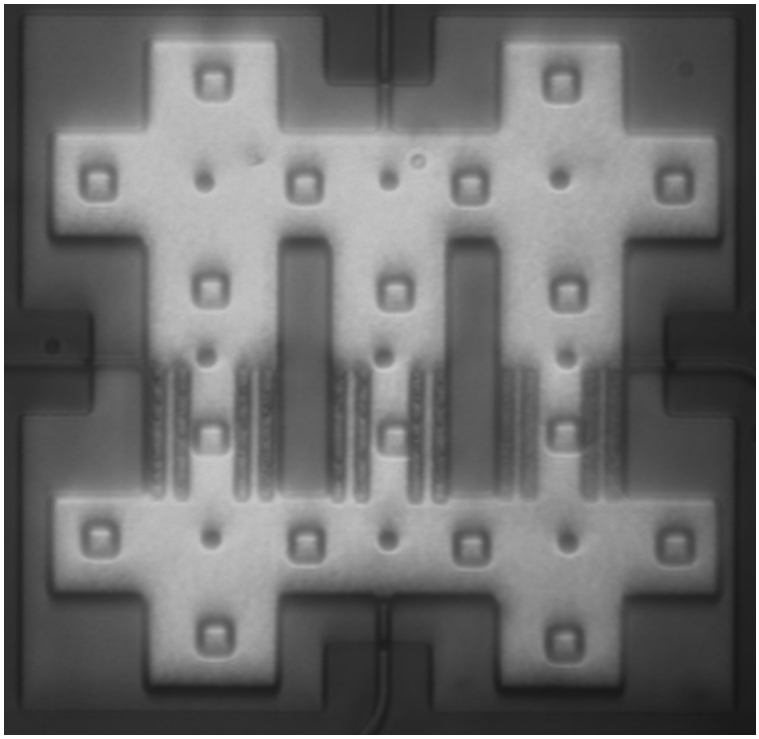


Figure 49: Resultant MEMS Pad Using above Created Etching Pattern (Figure 48)

6.3: **Etching Extra Patterns**

The FIB was used to create dark rectangular shapes in MEMS pads equivalent of dark columns in paper combs. Four different patterns were created on surface of identical MEMS pads to compare results with each other as shown in Figure 50. All rectangles had length of 25 μm but their width varied from 1-4 μm.

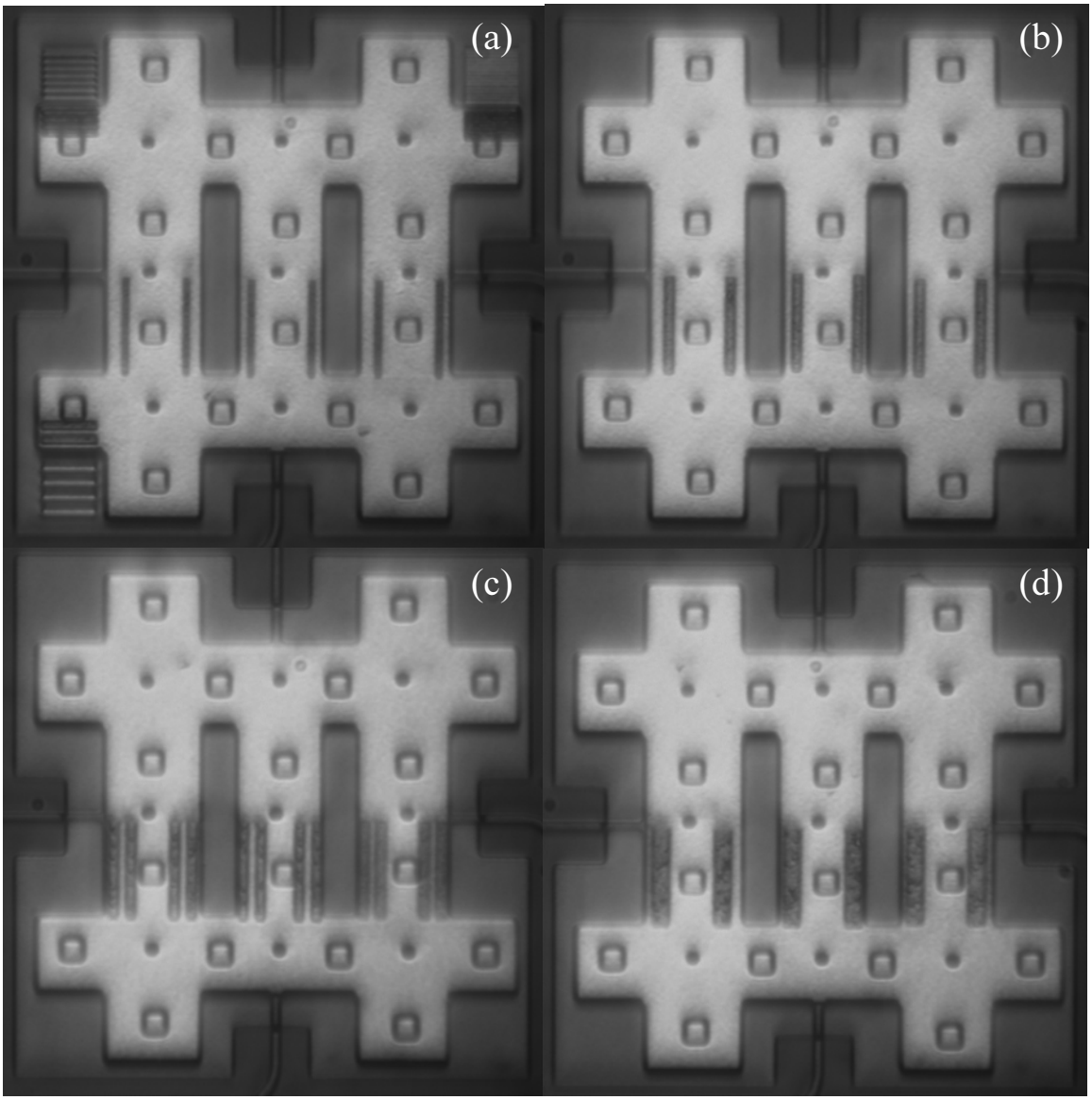


Figure 50: Micrograph of Etched MEMS Structures with Four Different Configuration: (a) 25x1x1, (b) 25x2x1, (c) 25x2x2 and (d) 25x4x1. All Dimensions are in Microns

Only half length of pads columns were etched in order to eliminate effect of environmental parameters such as light intensity, unwanted vibrations, camera focal point, etc. on tests readings. In this configuration, two series of FFT analysis were performed on the exact same series of captured images: in one test upper parts were selected (plain structure) and in another one, lower parts (etched structure). Figure 51 shows this type of selection. Row averages of the MEMS pad for upper section (plain) as well as lower section and for different etchings sizes are shown in Figures 52-56. Plots of displacement vs. frequency for upper and lower selected area of 25×2×2 pattern are shown in Figures 57 and 58 respectively. Figure 59 shows measurement results for each case. In all four patterns, the etched sections (lower area) showed higher SNR than plain ones (upper area). The amounts of improvements in measured SNR were 37% for 25x1x1, 60% for 25x2x1, 30% for 25x4x1 and 85% for 25x2x2 configurations. The highest amount of improvement belonged to the 25x2x2 pattern as expected because it had double number of patterns in comparison with others. 25x2x1 showed the best results between the other three models because it produced most harmonic row average. Our SNR readings in micro scale model were lower than macro model due to lower vibration amplitude and contrast ratio of captured images.

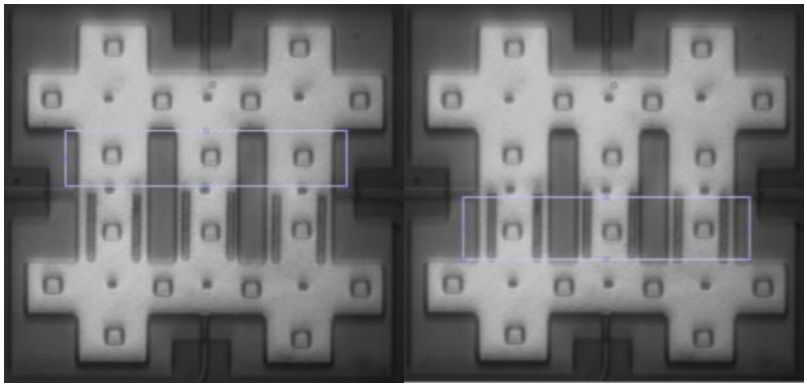


Figure 51: Illustration of Upper and Lower Selection Area in Etched MEMS Pads

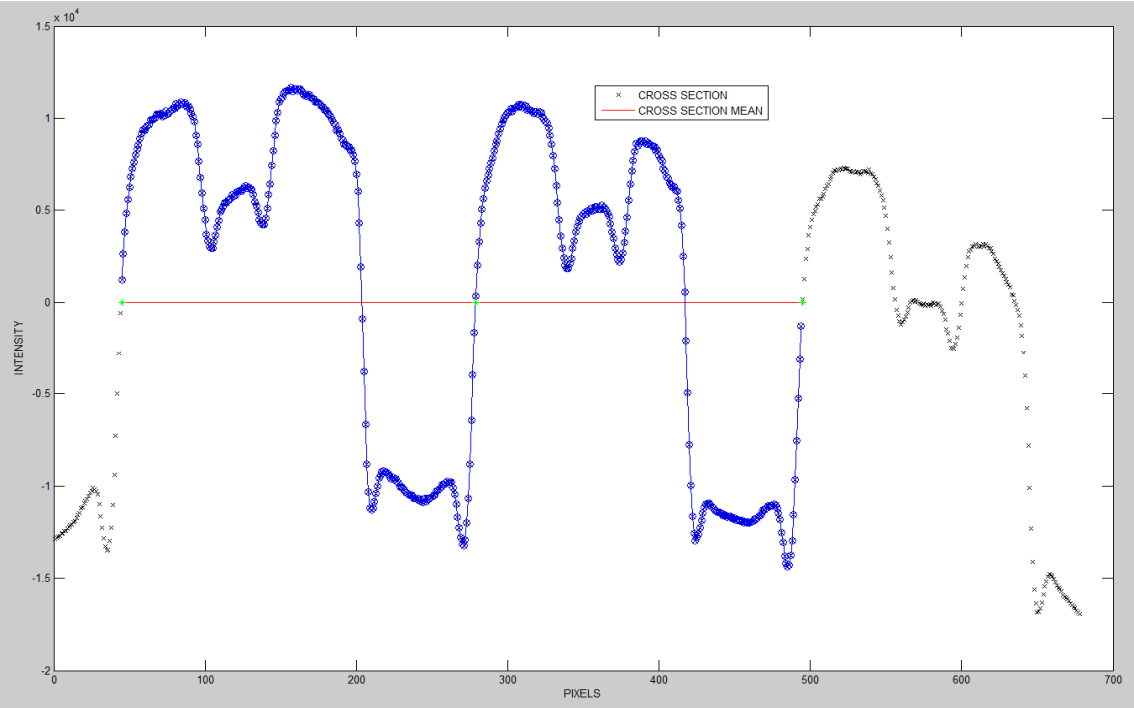


Figure 52: Row Average of MEMS Pad Upper Section (Plain)

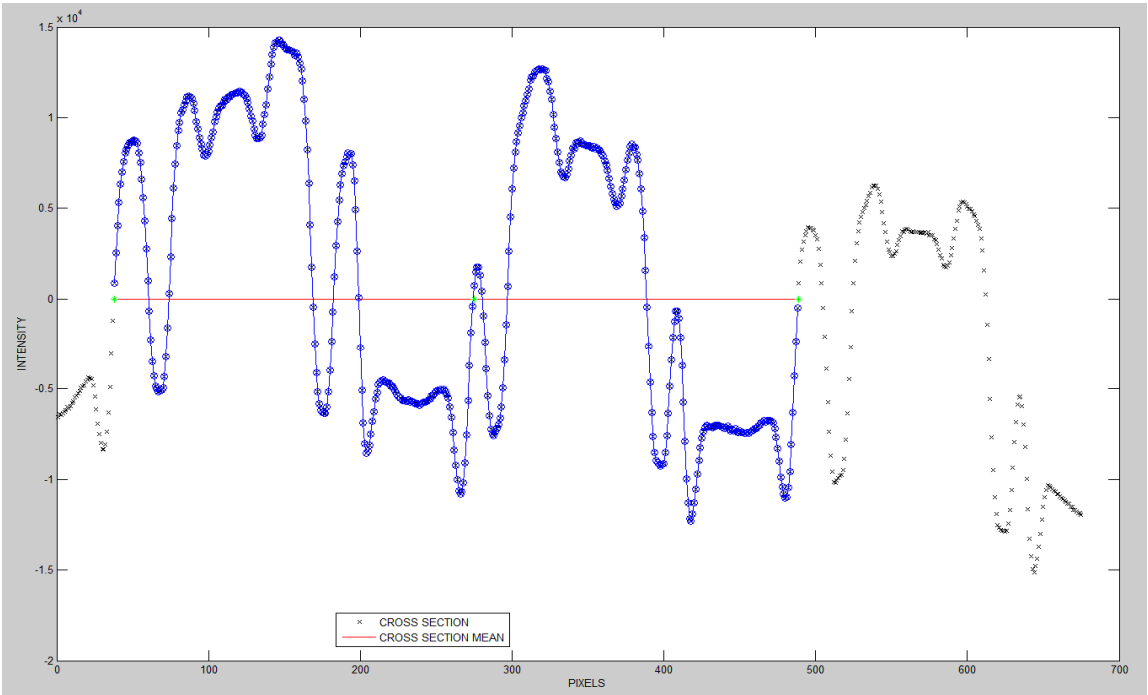


Figure 53: Row Average of MEMS Pad Lower Section (25x1x1)

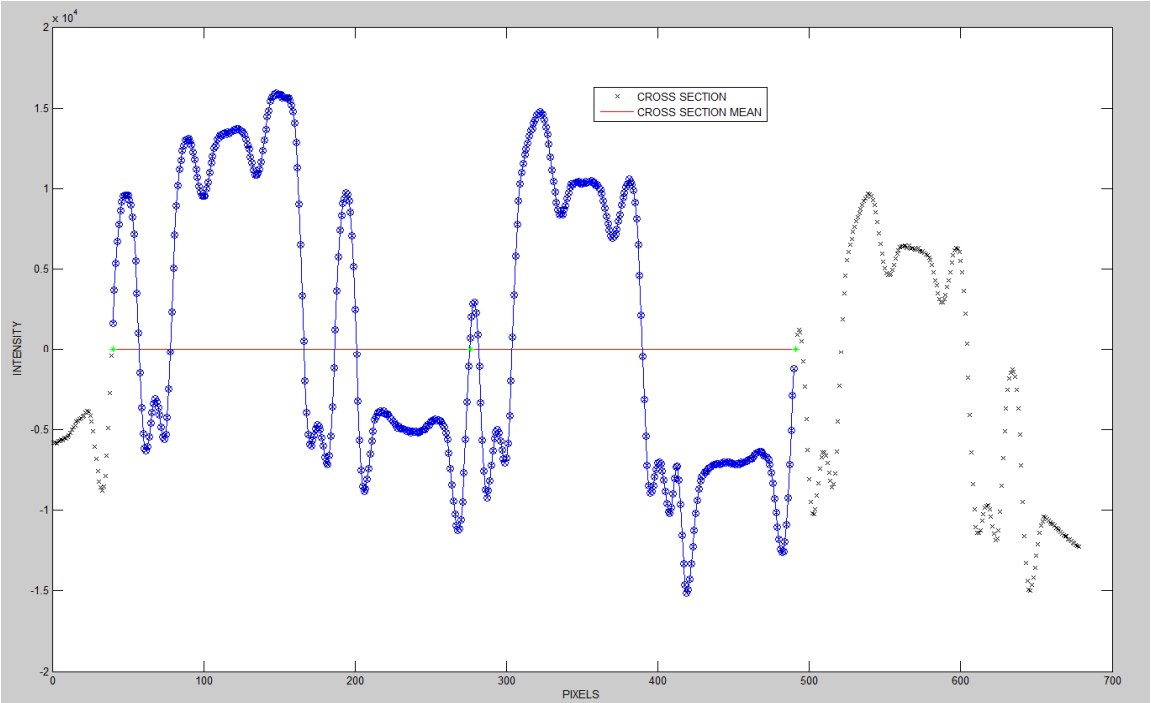


Figure 54: Row Average of MEMS Pad Lower Section (25x2x1)

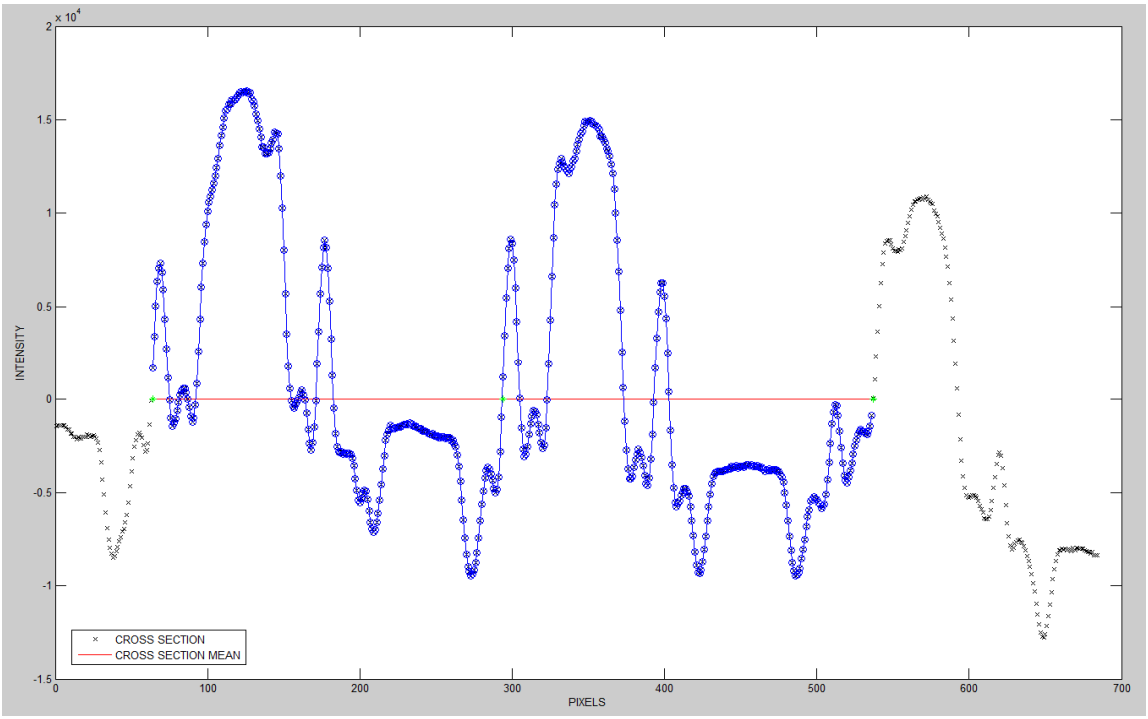


Figure 55: Row Average of MEMS Pad Lower Section (25x2x2)

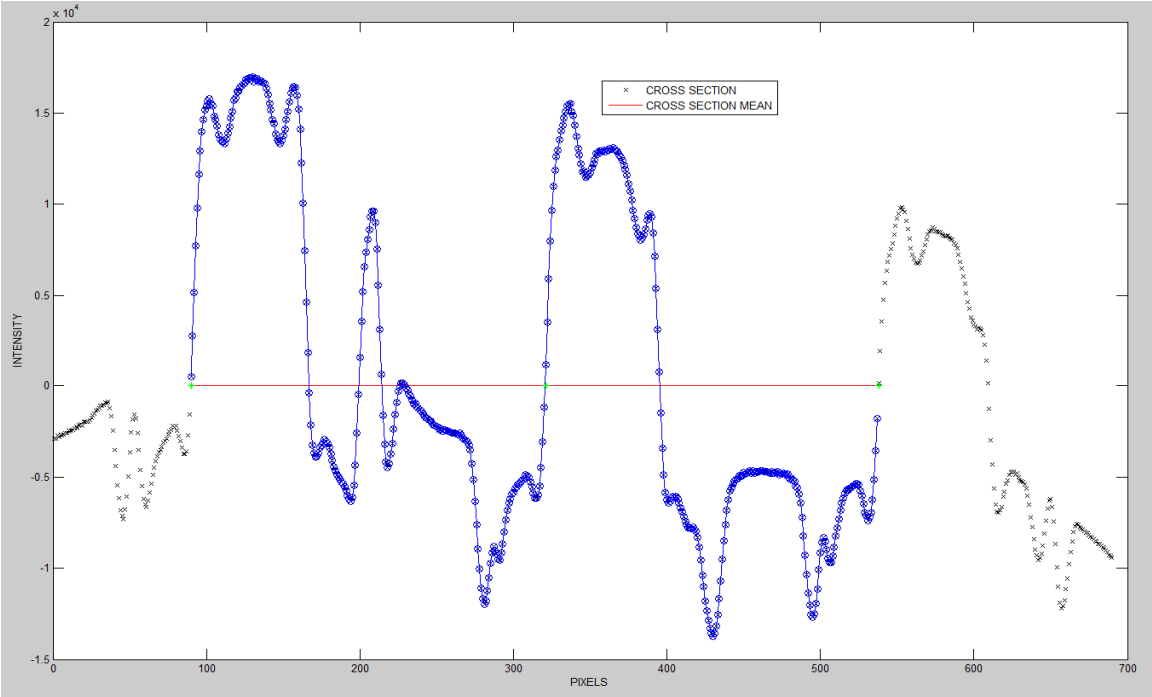


Figure 56: Row Average of MEMS Pad Lower Section (25x4x1)

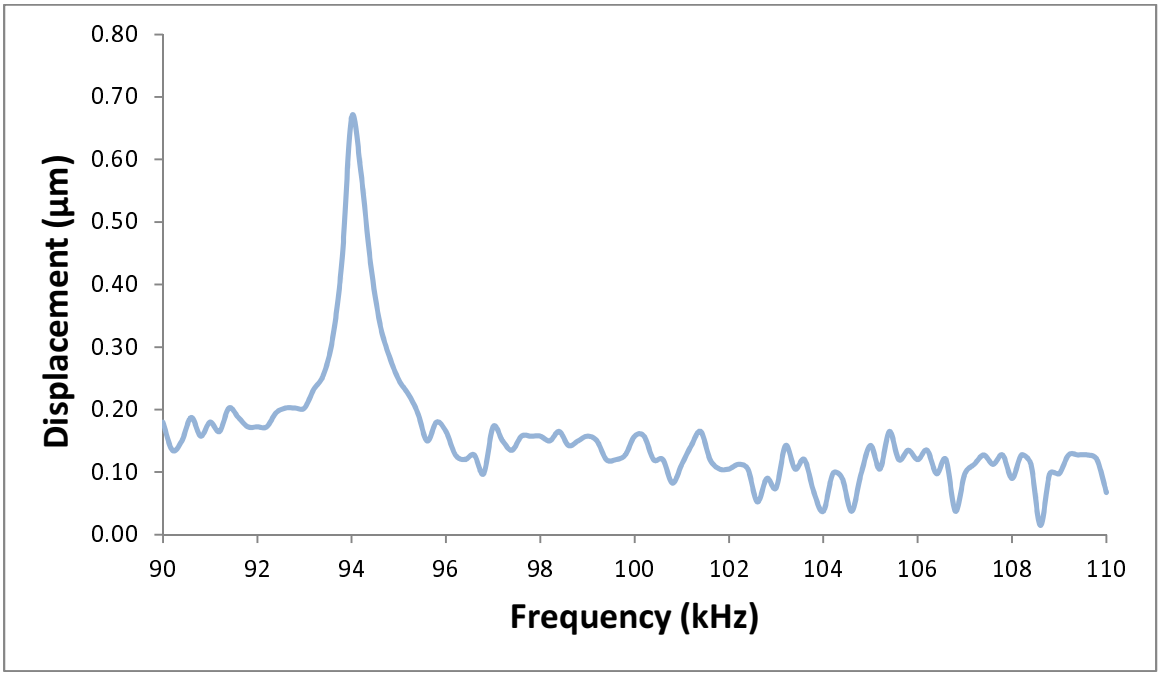


Figure 57: Plot of Displacement vs. Frequency for Upper Section of 25×2×2 Pattern (SNR=8.3)

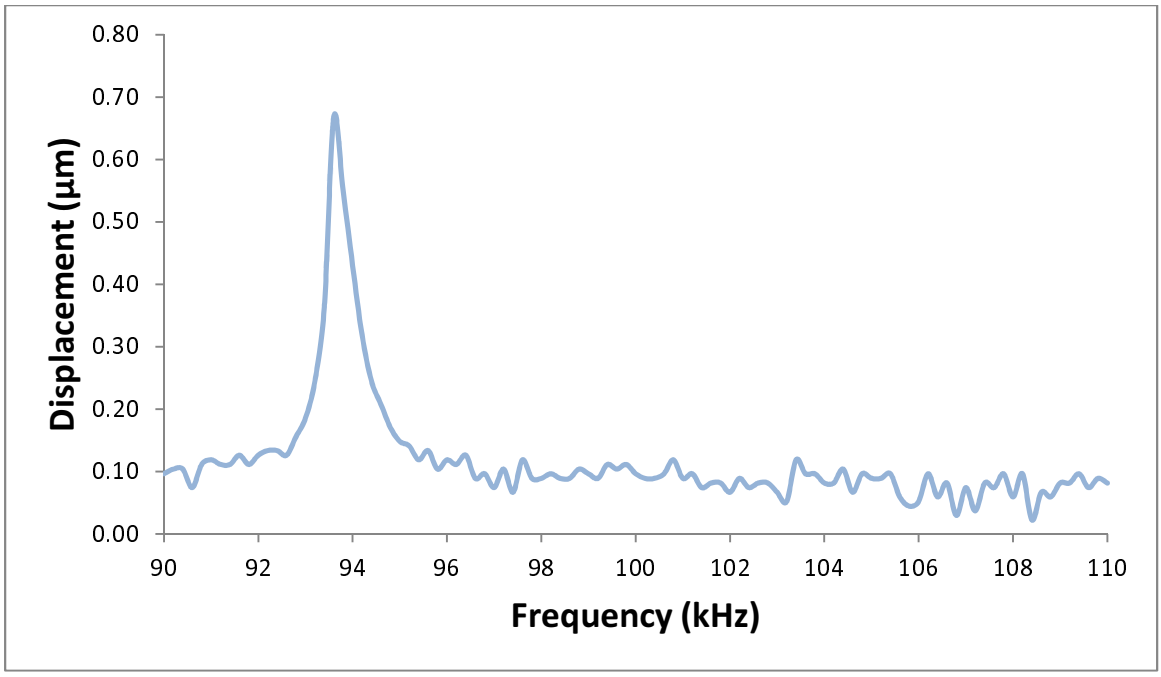


Figure 58: Plot of Displacement vs. Frequency for Lower Section of 25×2×2 Patten (SNR=15.4)

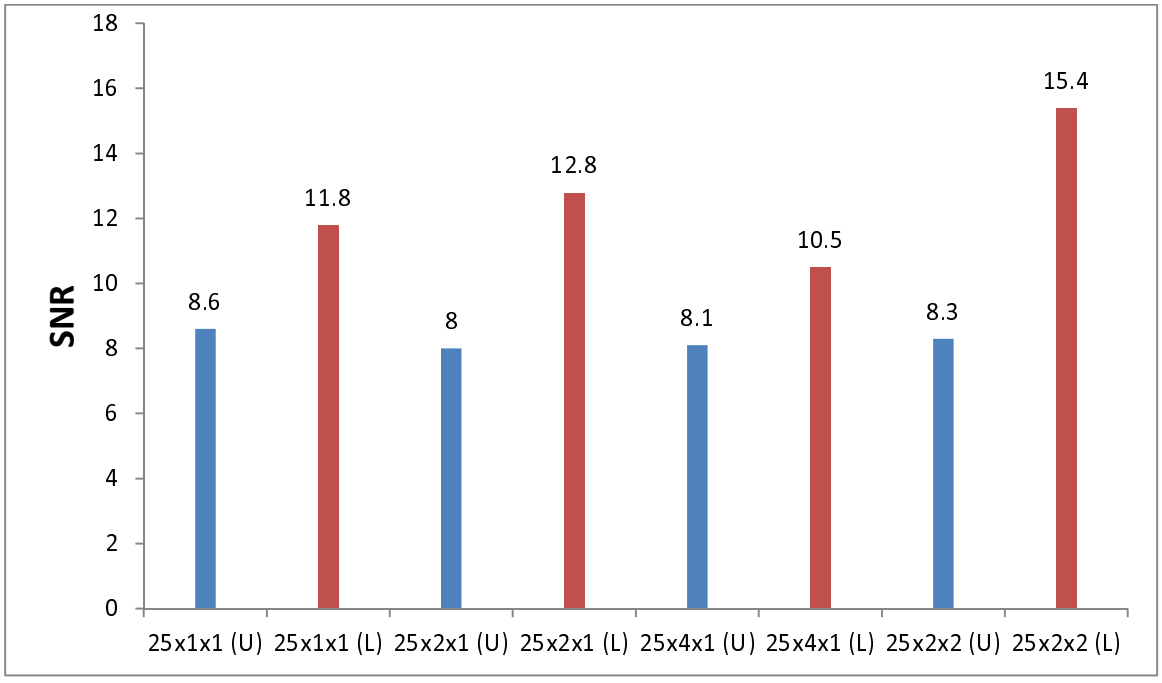


Figure 59: SNR Values for Four Different Patterns Based on Selected Area: Plain Section or Upper Area (U), Etched Section or Lower Area (L)

6.4: **Software Modifications**

While the addition of extra patterns in micro model improved results, it required complex and expensive equipment. Simple yet effective methods such as modification of image processing program can also be used for further enhancement of readings. For example, by embedding MATLAB code “imadjust” from “Image Processing Toolbox” into main program, the intensity of each image was increased before further processing which improved the results [29]. Figure 60 shows sample images of MEMS pad before and after adjustment. Figure 61 and 62 show the test results and effect of this software modification.

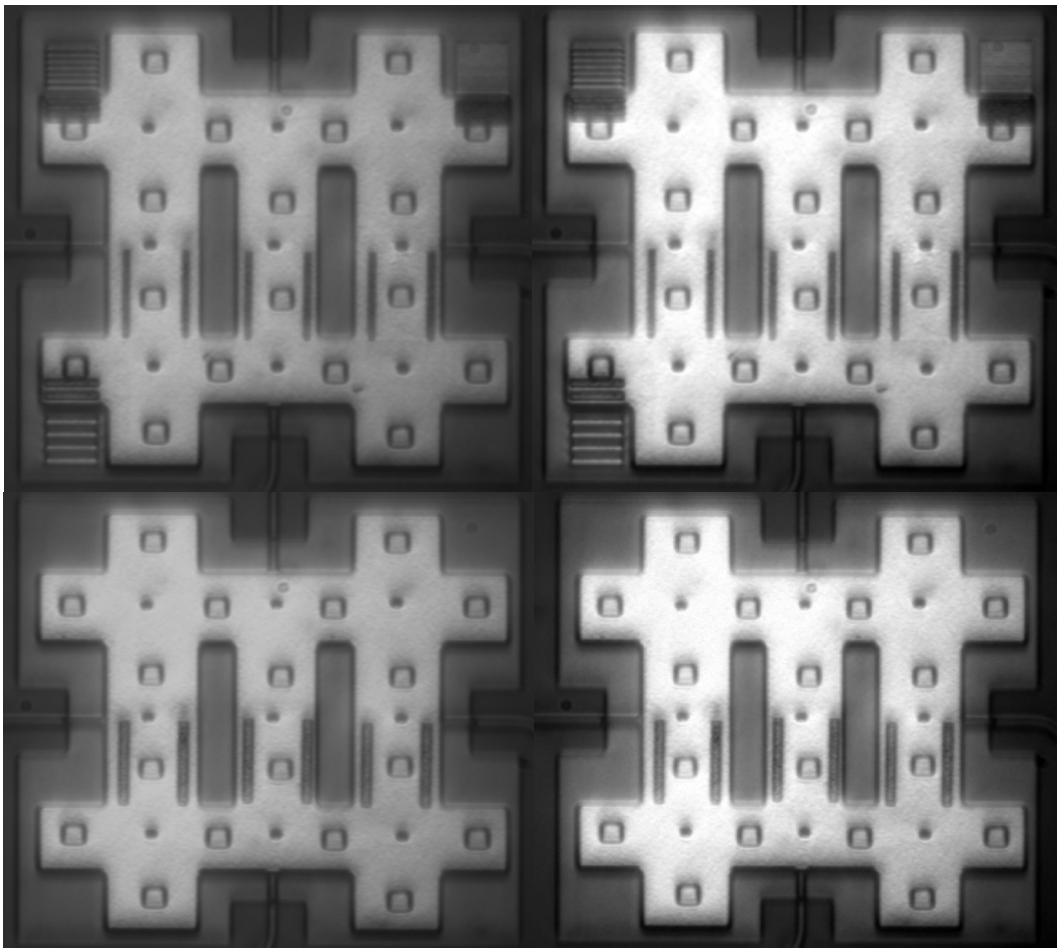


Figure 60: Original (Left) and Enhanced (Right) Images of MEMS Pads

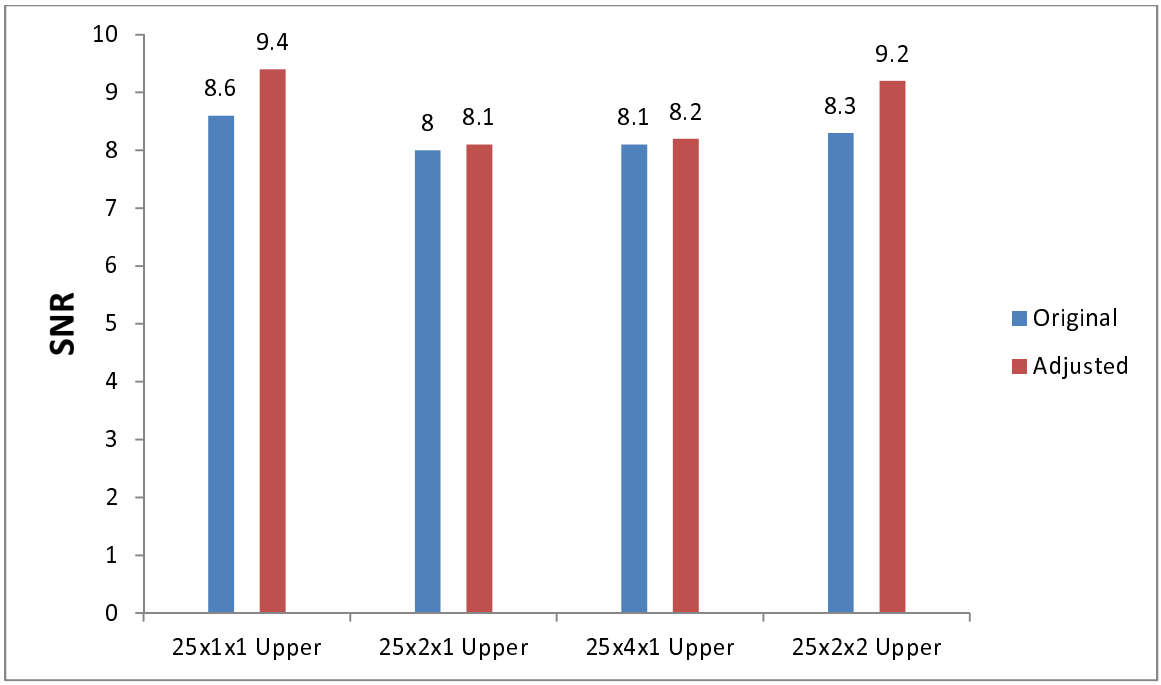


Figure 61: Comparison of SNR Values for Different Etching Patterns before and after Image Adjustment When Upper Area (Plain) is Selected

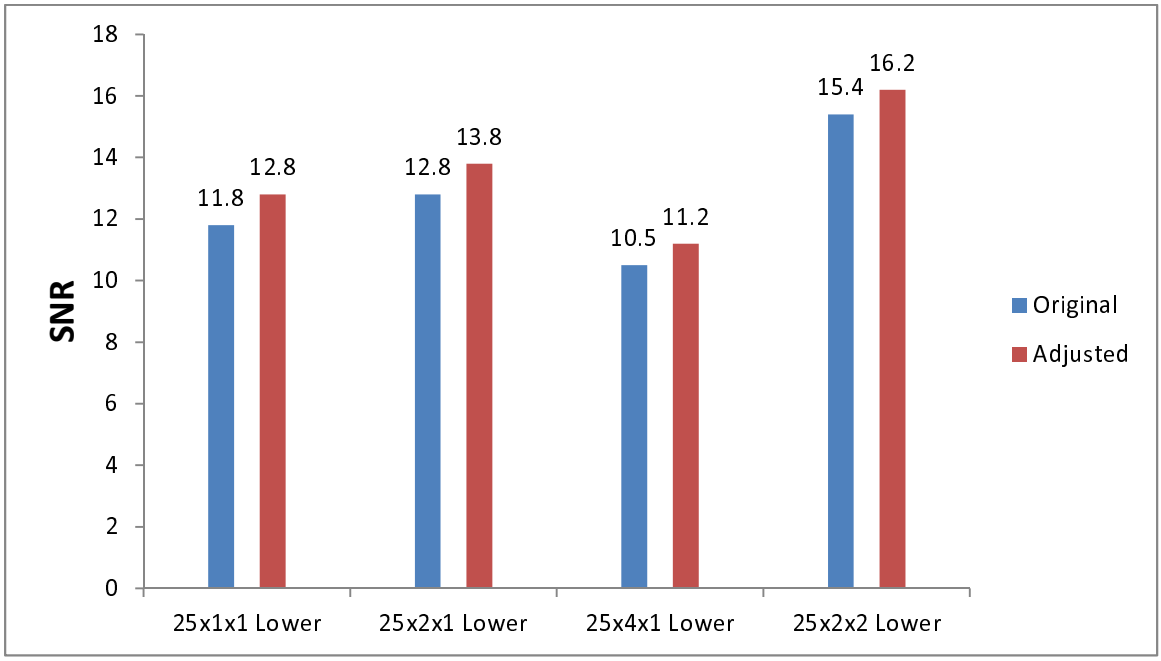


Figure 62: Comparison of SNR Values for Different Etching Patterns before and after Image Adjustment When Lower Area (Etched) is Selected

6.5: Summary, Recommendations and Analyses

Table 6 shows summary of the test results for micro scale model with each color representing different etching pattern. Overall, the FIB created patterns increased SNR values between 30-85% which is significant. Implementation of the MATLAB code into the main program improved readings by 5-12%. Therefore embedding this code into main processing program is recommended. The amount of improvement by software modification was higher in tests with low SNRs in comparison with those having high SNRs. The slight difference in SNRs for upper section of different patterns was related to different testing conditions and the fact that they were four different (although geometrically identical) structures.

Table 6: Summary of Micro Scale Model Test Results

Test #	FIB Pattern (µm)	Selection Area	Mode	SNR
1	25x1x1	Upper	Original	8.6
2	25x1x1	Upper	Adjusted	9.4
3	25x1x1	Lower	Original	11.8
4	25x1x1	Lower	Adjusted	12.8
5	25x2x1	Upper	Original	8.0
6	25x2x1	Upper	Adjusted	8.1
7	25x2x1	Lower	Original	12.8
8	25x2x1	Lower	Adjusted	13.8
9	25x4x1	Upper	Original	8.1
10	25x4x1	Upper	Adjusted	8.2
11	25x4x1	Lower	Original	10.5
12	25x4x1	Lower	Adjusted	11.2
13	25x2x2	Upper	Original	8.3
14	25x2x2	Upper	Adjusted	9.2
15	25x2x2	Lower	Original	15.4
16	25x2x2	Lower	Adjusted	16.2

Chapter 7: Simualtion Tests

In additional to macro and micro scale experimental tests, two sets of simulations with MATLAB program was performed by adding noise and darkness to images.

7.1: Adding Noise

In the first test, different percentile of artificial noise was added to original images of MEMS pad with 25×2×1 etching patterns. Figure 63 shows final results.

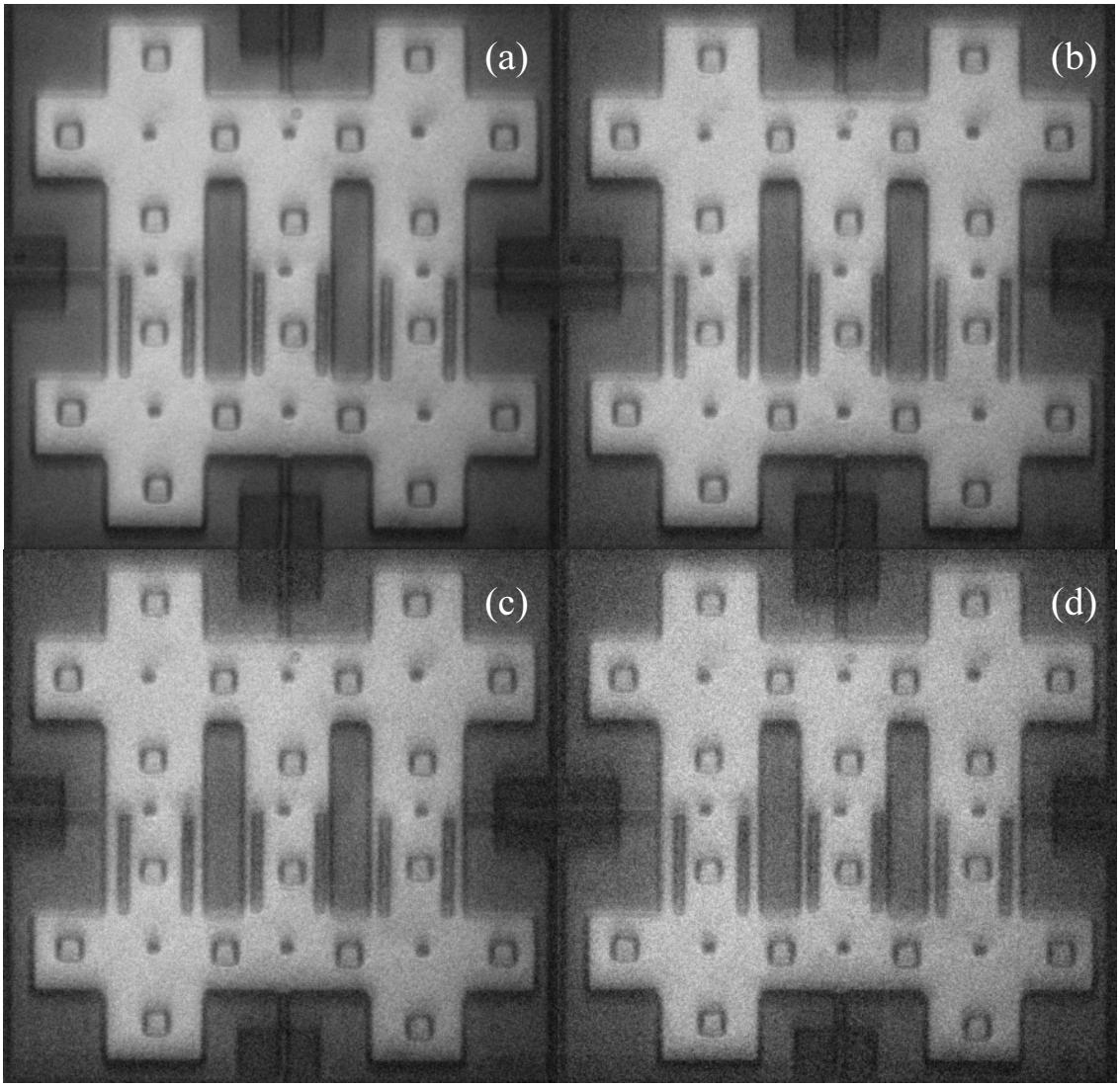


Figure 63: MEMS Pad with Added Artificial Noise by (a) 20%, (b) 30%, (c) 40% and (d) 50%

The purpose of both tests was to simulate environmental factors and evaluate effect of created patterns on improving results. The programs was created such that enabled the user to create artificial blur (in pixels) on images in addition to adding noise or darkness. To create artificial noise, a random matrix with dimension of original image was created and its values multiplied by percentage of noise level (noise level/100) then the resultant matrix was added to original image matrix to form the final result. The processing program then reads the noisy images for further analysis.

Micro scale tests was performed with the same conditions described in chapter six to obtain a series of images before adding effects to them. In this tests however, VPP of 10V was applied directly to the chevrons connection pads instead of through the distribution box. This method greatly reduced resistivity and increased amplitude of vibration as well as SNR values. Figure 64 shows the results for different levels of noise and selection area: even a small amount of noise, severely affected readings. Etching patterns were very beneficial in this case: they increased SNR values by multiple times.

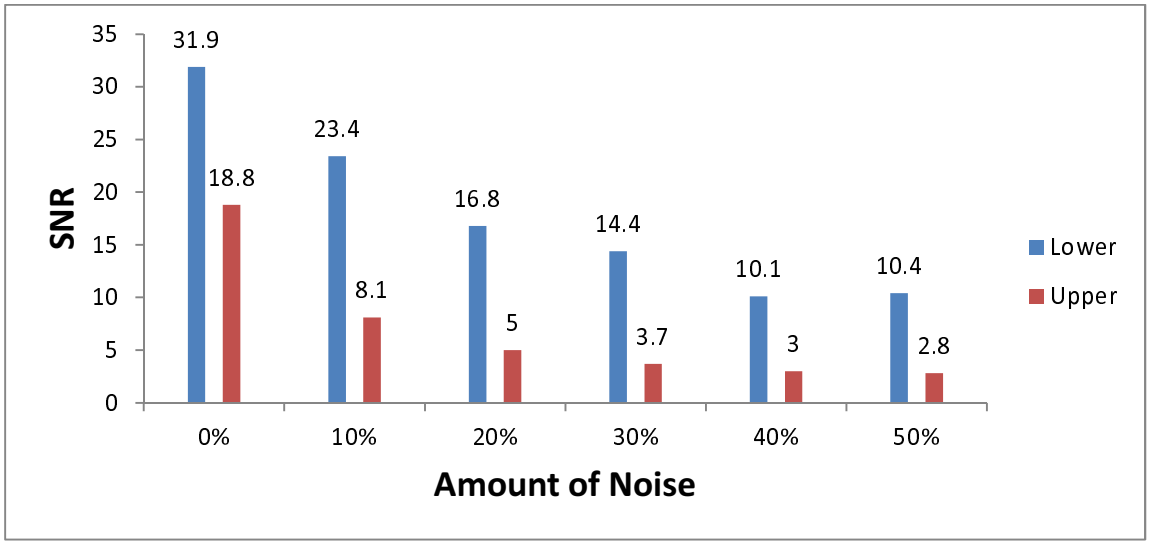


Figure 64: SNR Values Based on Noise Level and Selection Area: Lower (Etched) or Upper (Plain)

7.2: **Adding Darkness**

In addition to noise, the effect of adding darkness or reducing brightness was also investigated. To add darkness, user determines a variable between 0-1 in the program. “Brightness” is defined as 1-variable and initial image matrix is multiplied by “Brightness” to form the final image. Therefore the higher defined variable results the lower numerical values of image matrix and the darker image. Figure 65 shows four samples of one MEMS pad images with different levels of darkness.

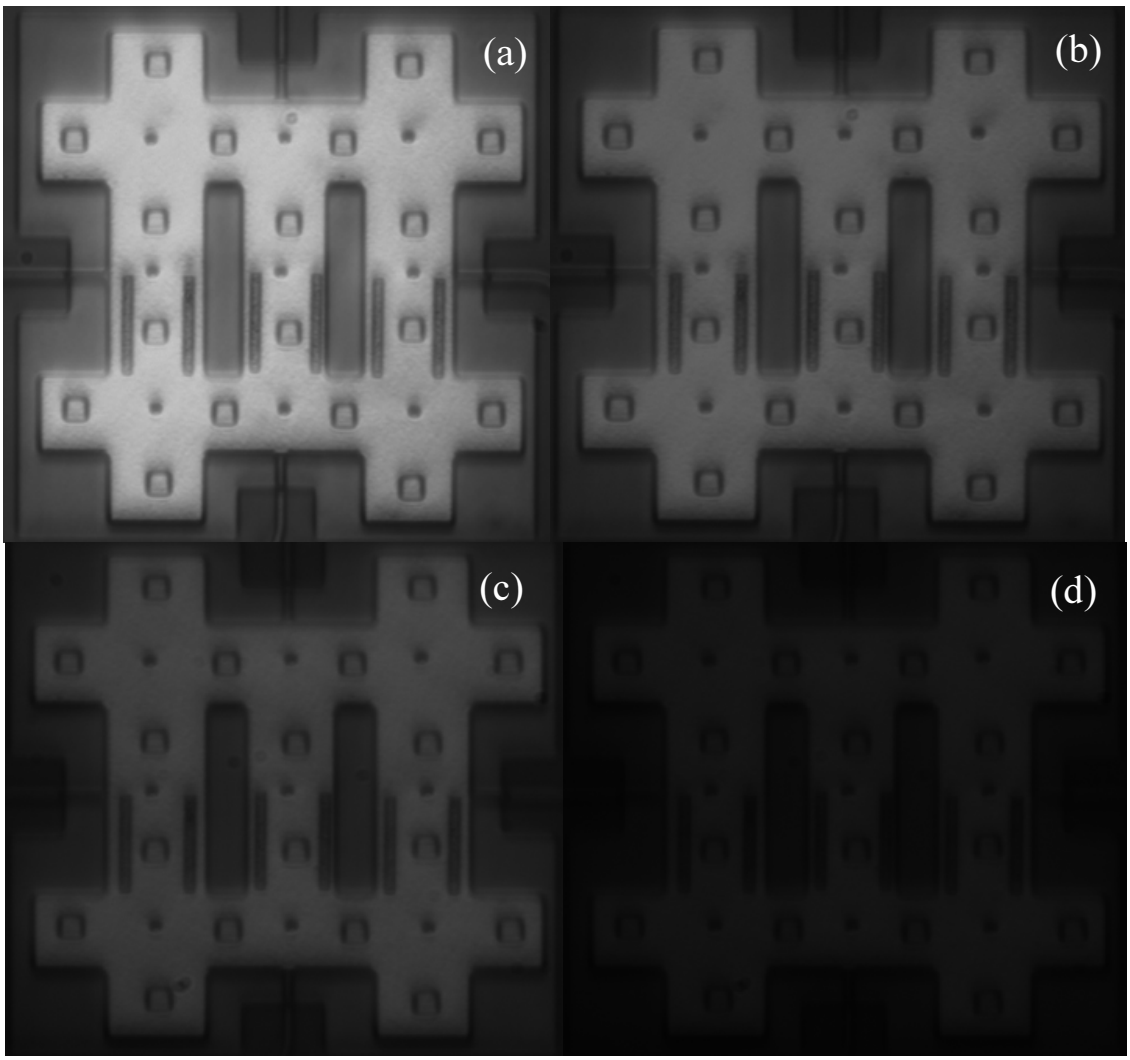


Figure 65: MEMS Pad with Artificial Reduction in Brightness by (a) 0%, (b) 50%, (c) 70%, (d) 90%

The darkness tests were performed with variable (darkness) value of 0.50, 0.70, 0.90, 0.92 and 0.94. Beyond 0.94 values, the resultant image becomes so dark that the user is unable to identify different regions and select suitable ROI (upper and lower area). Figure 66 shows the darkness test results. The image processing program showed much less sensitivity to brightness than human eye: the results looked almost identical even when brightness was reduced by as much as 50%. Only at very high level of darkness the results started to decline and even then, the etched pattern helped to increase SNR by as much as 2 times.

It must be noted, changing the brightness has a different effect on results than applying "imadjust" code into program discussed in previous chapter. The code does not increase or decrease the brightness of images but adjust their intensity to increase their contrast ratio and proved to improve results by 5-12%.

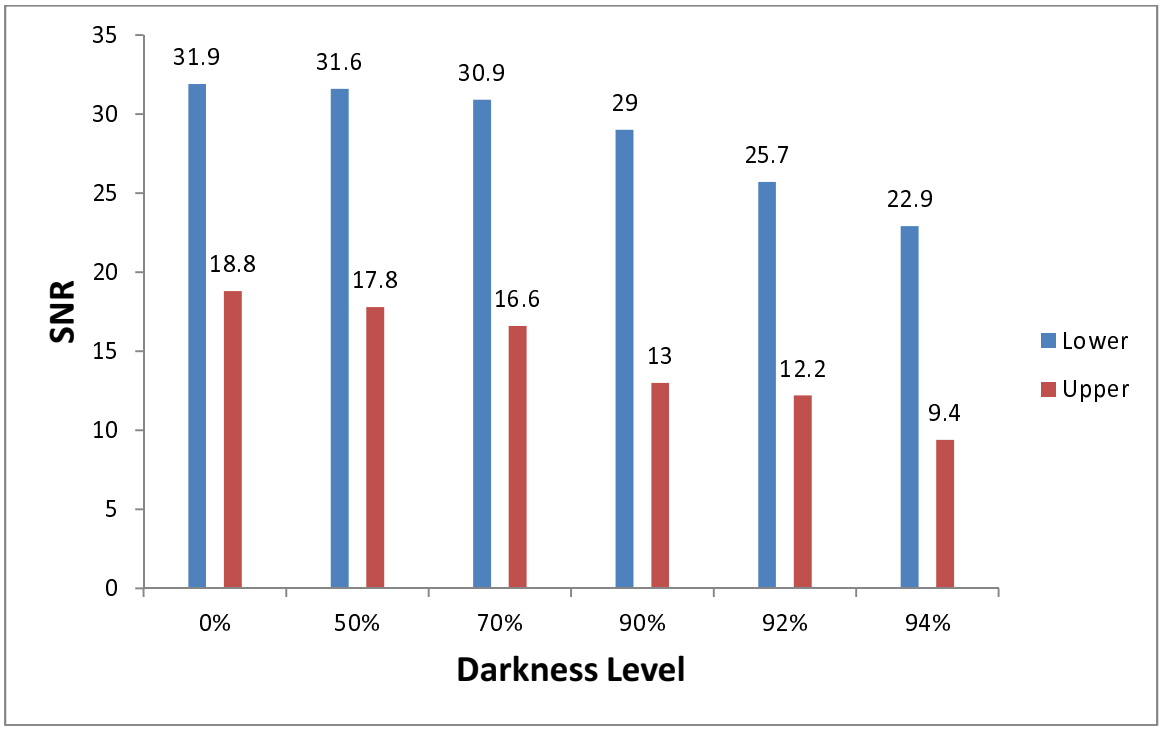


Figure 66: SNR Values Based on Level of Darkness and Selection Area

7.3: Summary, Recommendations and Analyses

7.3.1: Summary

Table 7 shows the SNR values for simulation tests. Each test was identified with a different color indicating parameter (darkness or noise) that has changed.

Table 7: Summary of Simulation Test Results for 25x2x1 MEMS Pad

Test #	Noise (%)	Darkness (%)	Selection Area	SNR
1	0	0	Lower	31.9
2	0	0	Upper	18.8
3	10	0	Lower	23.4
4	10	0	Upper	8.1
5	20	0	Lower	16.8
6	20	0	Upper	5.0
7	30	0	Lower	14.4
8	30	0	Upper	3.7
9	40	0	Lower	10.1
10	40	0	Upper	3.0
11	50	0	Lower	10.4
12	50	0	Upper	2.8
13	0	50	Lower	31.6
14	0	50	Upper	17.8
15	0	70	Lower	30.9
16	0	70	Upper	16.6
17	0	90	Lower	29.0
18	0	90	Upper	13.0
19	0	92	Lower	25.7
20	0	92	Upper	12.2
21	0	94	Lower	22.9
22	0	94	Upper	9.4

7.3.2: **Recommendations and Analyses**

Tests showed adding even small amount of noise severely reduces SNR values. This phenomenon can be explained by the fact that adding noise makes clear detection of edges very difficult for the program. In this situation, extra etched columns helped a lot and boosted SNR readings significantly. In the darkness test, processing program showed to be much less sensitive to darkness than human eye: at 50% darkness level while resultant image looked clearly darker than the original one, there was only 6% reduction in SNR value. Nevertheless and in all cases, etched patterns improved results. It can be concluded that adding noise has much more adverse effect on results than reducing brightness. Therefore it is very important to reduce the amount of noise as much as possible when capturing images to achieve the best possible results.

Chapter 8: Color FFT

So far, a monochrome camera was used for all conducted tests. In this experiment however, a color camera model SANYO with resolution of 640×480 pixels was utilized to capture color images of MEMS pad. Figure 67 shows samples of captured image with different color filtrations. The purpose of changing camera was to investigate effect of color images as well as different color filtration on micro scale experiments. The initial LabVIEW program was modified to work with analog color camera: the old program sent commands to "FlyCapture" (bundled software that came with Grasshopper camera) for capturing images. The new one however, used "NI-IMAQ for USB Cameras" driver software to communicate with IMAQ card [30]. The card was connected to an adapter via USB cable, the adapter itself was connected to the color camera with BNC cable.

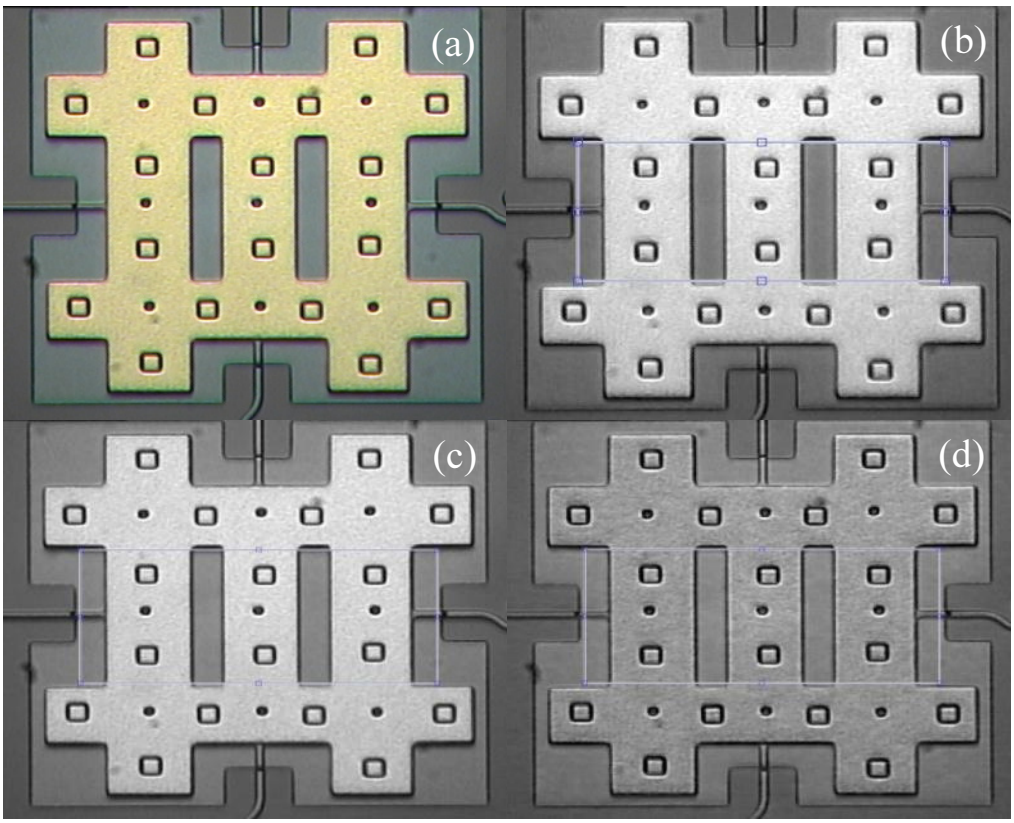


Figure 67: A MEMS Pad with Different Color Filtration: (a) RGB, (b) Red, (c) Green and (d) Blue

The original MATLAB program was also modified to accept and process the color images. The initial program had no problem with reading VGA sized images; however it gave an error massages at later steps about matrix dimensions as it had three side wall profiles associated with three colors (RGB) to process. The modified program however, at the initial steps asks the user which color spectrum to process: 1-Red, 2-Green and 3-Blue (Red being default). The micro scale tests were conducted using a different chip with similar structure but without etchings. The frequency sweep was performed for five runs with VPP of 22V and frequency range of 90-110 kHz with 0.1 kHz steps resulting 200 camera shots. The numerical values of FFT harmonics were different in each color spectrum but they all showed the same resonance frequency (101.2 kHz).

In addition to the analysis of the primary colors (R, G and B), the FFT of gray scale image was also obtained. Two methods can be used to convert an RGB image to gray scale. In first method, “rgb2gray” command from MATLAB image processing toolbox can be used and embed into main program [31]. Hence each image would be converted to gray scale before further processing.

Another method is extraction of Red, Green and Blue matrices of each image and then added together 30% of the red value, 59% of the green value and 11% of the blue value to obtain gray scale image [32]. Both methods were used and the resultant image matrices were compared. The comparison showed they are essentially the same and therefore these two methods are interchangeable. Figure 68 shows comparison of a captured MEMS micrograph in color and gray scale mode. Figures 69-72 show the resultant side wall profile of MEMS pads with different color filtration.

Figure 73 shows the final results: the highest SNR belonged to gray scale image (20.8) followed by green (17.2), red (15.9) and finally blue (2.3). It must be noted, obtained results in this test are not comparable with previously performed tests using high resolution camera because in addition to using a different chip and performing more tests (5 runs vs. 3 runs); the captured color images had different resolution (VGA).

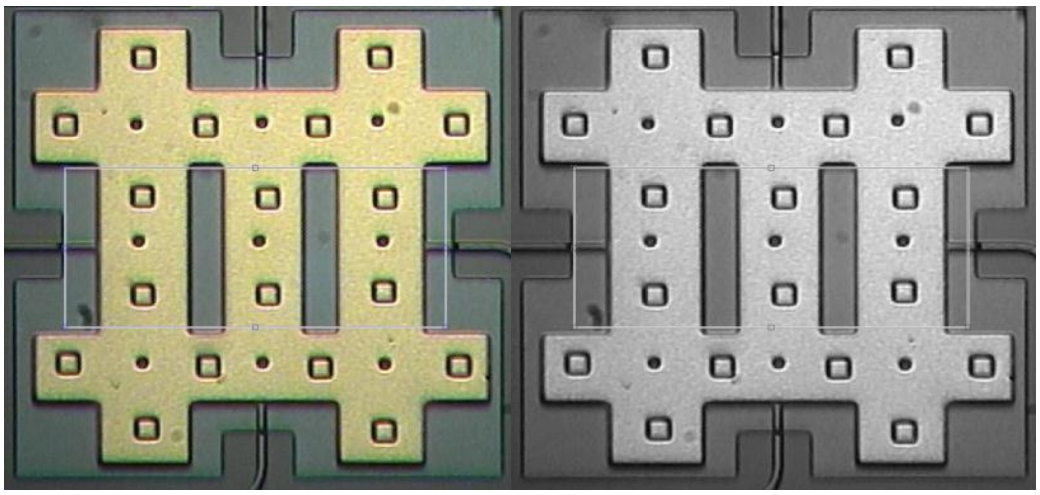


Figure 68: Micrograph of Pad in Color (left) and Converted to Grayscale (right) Using MATLAB

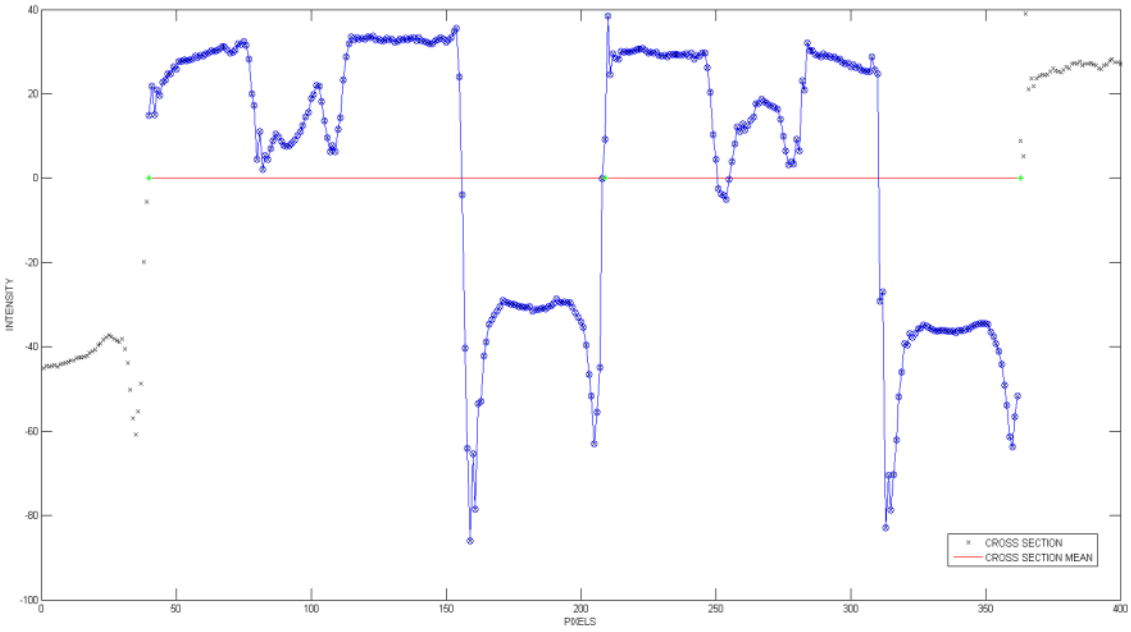


Figure 69: Side Wall Profile of MEMS Pad (Gray Scale)

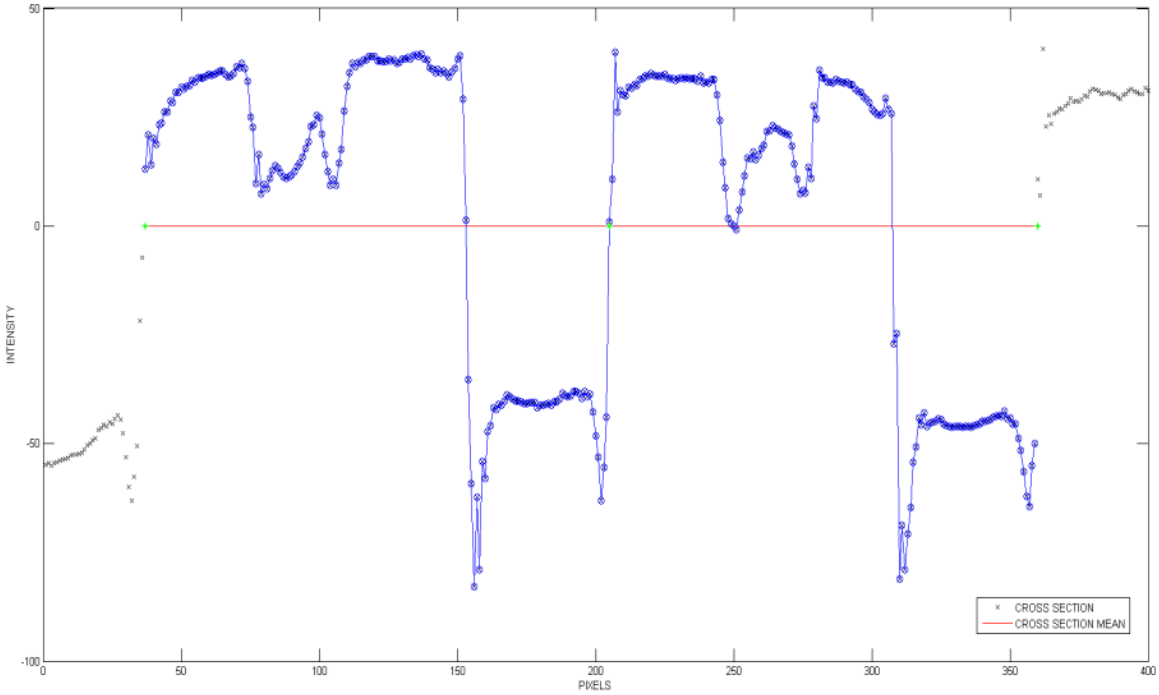


Figure 70: Side Wall Profile of MEMS Pad (Red Filter)

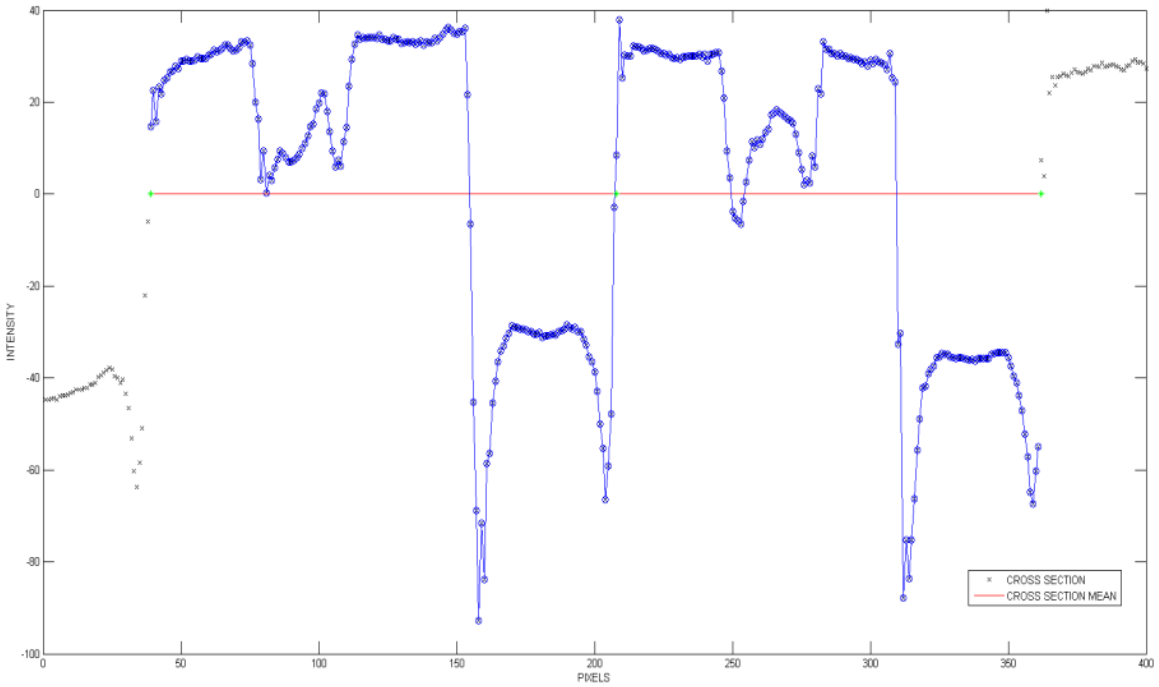


Figure 71: Side Wall Profile of MEMS Pad (Green Filter)

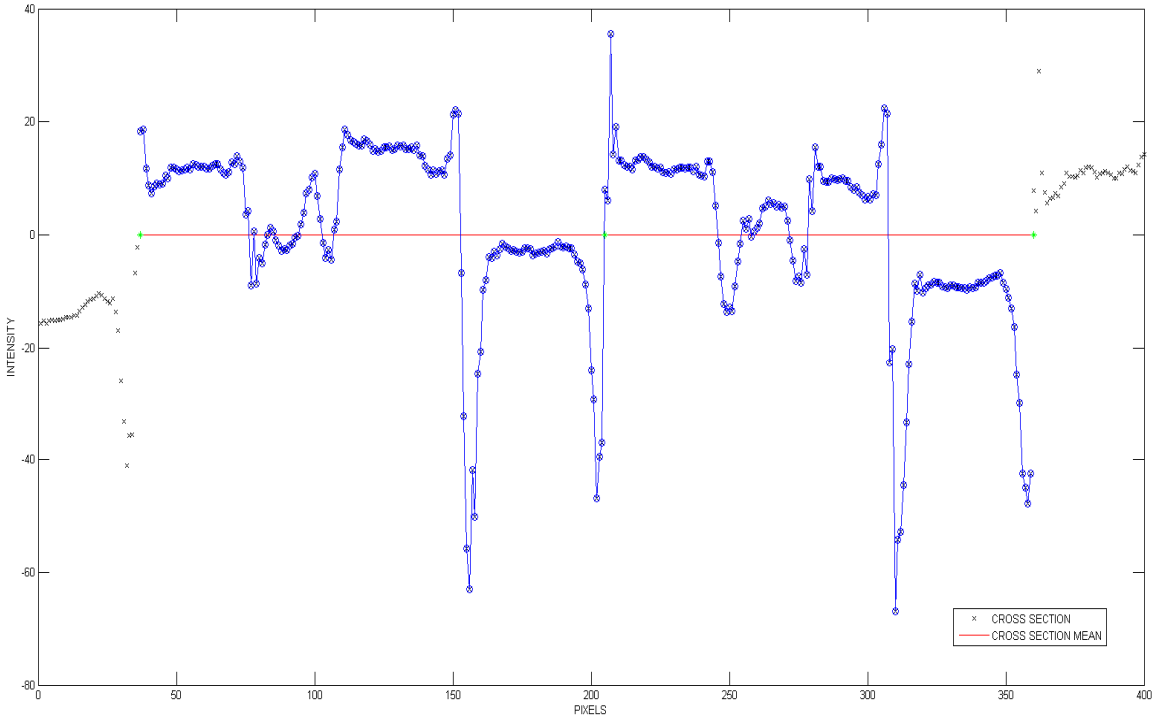


Figure 72: Side Wall Profile of MEMS Pad (Blue Filter)

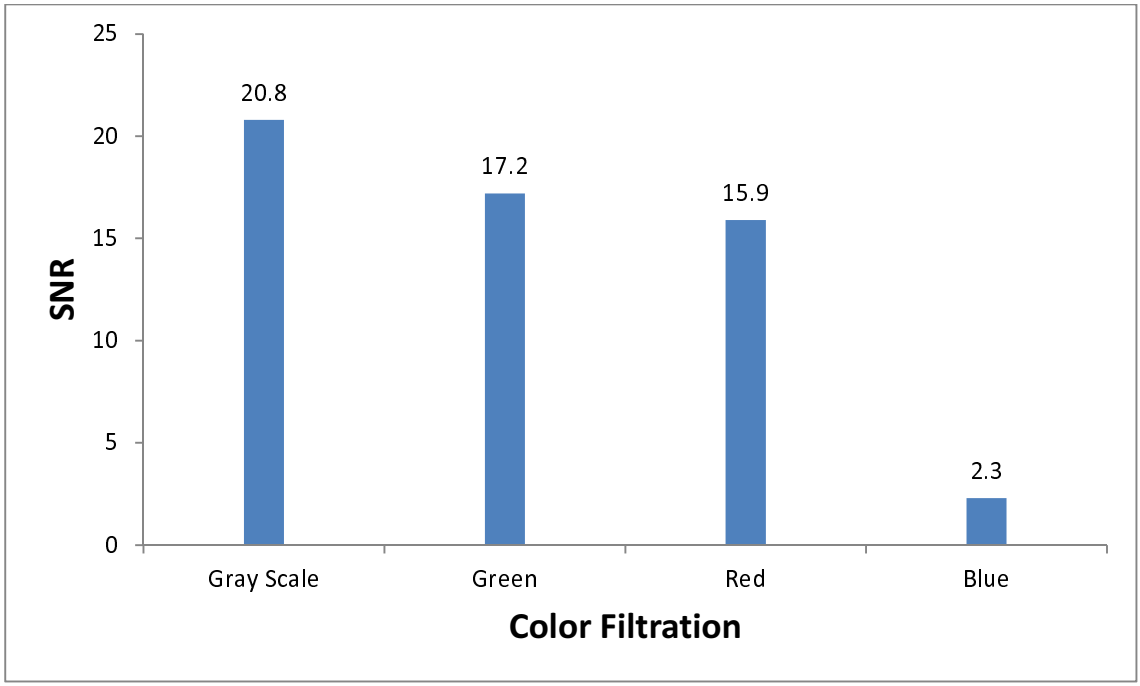


Figure 73: SNR Values for Color FFT Tests

8.1: **Summary, Recommendations and Analyses**

In the final results, gray scale image showed the highest SNR because it produced images with the highest contrast ratio. In case of blue filtration, the substrate and MEMS pad had very small difference in brightness (low contrast ratio) resulting very low SNR value. It can be concluded that using color camera for dynamic measurement of MEMS structures based on FFT analysis offers no advantage over gray scale camera. Therefore if a color camera is used for capturing images, it is recommended to convert them to gray scale images before further processing. This can be done by using and embedding "rgb2gray" command from MATLAB image processing tool box.

Chapter 9: Summary and Conclusion

Fourier series, transform and analysis were introduced in chapter two; discrete as well as fast Fourier transforms were also explained there. Fundamental concepts for in-plane dynamic measurement of MEMS devices based on blur images were introduced in chapter three. It was shown both analytically as well as numerically that blur in the frequency domain acted as a low pass filter for FFT harmonics. Chapter four described the step by step processes for dynamic motion information extraction of 2D images based on FFT analysis: by row averaging of the captured images and calculation of FFT harmonics, it is possible to detect the resonance frequency as well as measure the amplitude of vibration.

Chapter five provided detailed description for macro scale model setup. Different geometrical parameters of paper combs such as column width and height as well as duty cycle were changed to investigate their effect on SNR readings. It was shown that increase in the number of columns generally improves the results. However, if this increase causes decrease in space between them; the higher numbers gradually deteriorate results due to decrease in contrast between dark and bright regions. Selection of regions of interest (ROI) with the highest possible height, using reference comb and duty cycle of 50%, all proved to have positive impact on final results.

In chapter six, micro scale prototype and used methods to enhance readings were discussed. At first the experimental setup and micro structure were explained, later two proposed enhancement methods were examined. The addition of extra patterns in the micro scale prototype was accomplished by using Focused Ion Beam instrument. Comparison of results obtained from four differently etched patterns on MEMS pads with

plain section showed 30-85% improvement. Also, embedding “imadjust” software code into main program enhanced images contrast ratio and improved SNR by 5-12%. This chapter showed that modifying the microstructure by adding extra patterns as well as editing the image processing program by embedding the software code improves the measurement readings. Physical modifications of microstructure required specific and expensive equipment however they proved to be more effective than software modifications.

In chapter seven, artificial environmental effects such as noise or darkness applied to the captured images of micro scale prototype. Comparison of modified micro structures with plain ones showed improvement in SNR especially in environments with high level of noise. Finally in chapter eight, same FFT analysis was performed for each components of a full RGB color image (Red, Green, Blue) by using a color camera. Comparison of results with gray scale model showed Red filtration has the highest SNR value among the other two color components.

In order to achieve the highest amount of SNR in dynamic measurement of MEMS devices based on blur images, the author recommends the following based on the findings in previous chapters:

- Selection or creation of at least 6-8 dark-bright regions similar to paper combs patterns which results sufficient wavelengths for accurate FFT calculations.
- Higher number of patterns improves overall results as long as the difference between dark and bright regions remains clear.
- Created patterns should have duty cycle of 50% or close to it as much as possible. That produces best harmonic row average.

- Selection of highest possible height for region of interest (ROI).
- Creation of an adjacent reference pattern if that is possible to eliminate effect of ambient vibration noise.
- Using the image processing program with “image adjustment” code embedded.
- Captured images should have sufficient brightness since too much darkens negatively impact results.
- Removing optical noise as much as possible since images with small amount noise lower the SNR values dramatically.

References

[1] N. Ellerington, B. Bshaden, T. Hubbard and M. Kujath, "Fourier Analysis of Blurred Images for the Measurment of the In-Plane Dynamics of MEMS," *Journal of Micromechanics and Microengineering,* vol. 22, no. 3, 2012.

[2] M. Gad-el-Hak, The MEMS Handbook, Second Edition, CRC Press, 2005.

[3] "MEMSCAP| MUMPs," MEMSCAP Inc., [Online]. Available: http://www.memscap.com/products/mumps.

[4] "PolyMUMPs Design Handbook," MEMSCAP Inc., [Online]. Available: http://www.memscap.com/__data/assets/pdf_file/0019/1729/PolyMUMPs-DR-13-0.pdf.

[5] A. Bosseboeuf and S. Petitgrand, "Characterization of the static and dynamic behaviour of M(O)EMS by optical techniques: status and trends," *Journal of Micromechanics and Microengineering,* vol. 13, no. 4, pp. S23-S33, 2003.

[6] S. Wang, B. Guan, G. Wang and Q. Li, "Measurement of sinusoidal vibration from motion blurred images," *Pattern Recognition Letters,* vol. 28, no. 9, pp. 1029-1040, 2007.

[7] H. Le, M. Gouiffes, F. Parrain, A. Bosseboeuf and B. Zavidovique, "Image blur analysis for the subpixel-level measurement of in-plane vibration parameters of MEMS resonators," *Proceedings of SPIE,* pp. 6696-66962D, 2007.

[8] S. Krylov, Y. Gerson, T. Nachmias and U. Keren, "Excitation of large-amplitude parametric resonance by the mechanical stiffness modulation of a microstructure," *Journal of Micromechanics and Microengineering,* vol. 20, no. 1, 2010.

[9] D. J. Burns and H. F. Helbig, "System for Automatic Electrical and Optical Characterization of Microelectromechanical Devices," *Journal of Microelectromechanical Systems,* vol. 8, no. 4, pp. 473-482, 1999.

[10] C. Yamahata, E. Sarajlic, G. J. M. Krijnen and M. A. M. Gijs, "Subnanometer Translation of Microelectromechanical Systems Measured by Discrete Fourier Analysis of CCD Images," *Journal of Microelectromechanical Systems,* vol. 19, no. 5, pp. 1273-1275, 2010.

[11] M. Pinsky, Introduction of Fourier Series and Wavelets, Thomson Brooks, 2001.

[12] M. Khamsi, "Fourier Series: Basic Results.," MathMedics, LLC., 12 January 2012. [Online]. Available: http://www.sosmath.com/fourier/fourier1/fourier1.html.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithms, MIT Press and McGraw-Hill, 2001.

[14] S. W. Smith, "Chapter 8: The Discrete Fourier Transform," in *The Scientists and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, 1999.

[15] E. O. Brigham, The fast Fourier transform and its applications, Englewood Cliffs, N.J.: Prentice Hall, 1988.

[16] E. O. Brigham, The Fast Fourier Transform, New York: Prentice-Hall, 2002.

[17] D. Elliott and K. Rao, Fast transforms: Algorithms, analyses, applications, New York: Academic Press, 1982.

[18] S. Haynal and H. Haynal, "Generating and Searching Families of FFT Algorithms," *Journal on Satisfiability, Boolean Modeling and Computation,* vol. 7, pp. 145-187, 2011.

[19] E. M. Stein, Harmonic Analysis, Princeton University Press, 1993.

[20] G. Watson, A Treatise on the Theory of Bessel Functions, Second Edition, Cambridge University Press, 1995.

[21] B. Korenev, Bessel Functions and Their Applications, Taylor & Francis, 2007.

[22] Y. Yitzhaky and N. Kopeika, "Identification of Blur Parameters from Motion Blurred Images," *Graphical Models and Image Processing,* vol. 59, no. 5, pp. 310-320, 1997.

[23] "Point Grey-Imaging Products-Grasshopper CCD Camera," [Online]. Available: http://www.ptgrey.com/products/grasshopper/grasshopper_firewire_camera.asp.

[24] Canadian Microsystems Corporation (CMC), [Online]. Available: http://www.cmc.ca/.

[25] Institute for Research In Materials (IRM), [Online]. Available: http://sem-fib.instituteforresearchinmaterials.dal.ca/index.htm.

[26] "Instruction Manual, FB-2000A Focused Ion Beam System," Hitachi High-Technologies Corporation, 2003.

[27] Y. Lai, "Investigation on the capability of a Focused Ion Beam System (FIB)," Technical Report, Dalhousie University, 2004.

[28] M. Mottaghi, "Fabrication and Change of Microstructures Using Focused Ion Beam," Technical Report, Dalhousie University, 2011.

[29] "MATLAB Documentation - Image Processing Toolbox," MathWorks, [Online]. Available: http://www.mathworks.com/help/toolbox/images/ref/imadjust.html.

[30] "NI-IMAQ for USB Cameras," National Instruments-Developer Zone, [Online]. Available: http://zone.ni.com/devzone/cda/epd/p/id/5030.

[31] "MATLAB Documentation - Image Processing Toolbox," MathWorks, [Online]. Available: http://www.mathworks.com/help/toolbox/images/ref/rgb2gray.html.

[32] W. k. Pratt, Digital Image Processing, New York: John Wiley & Sons, 1991.

Appendix A: LabVIEW Program

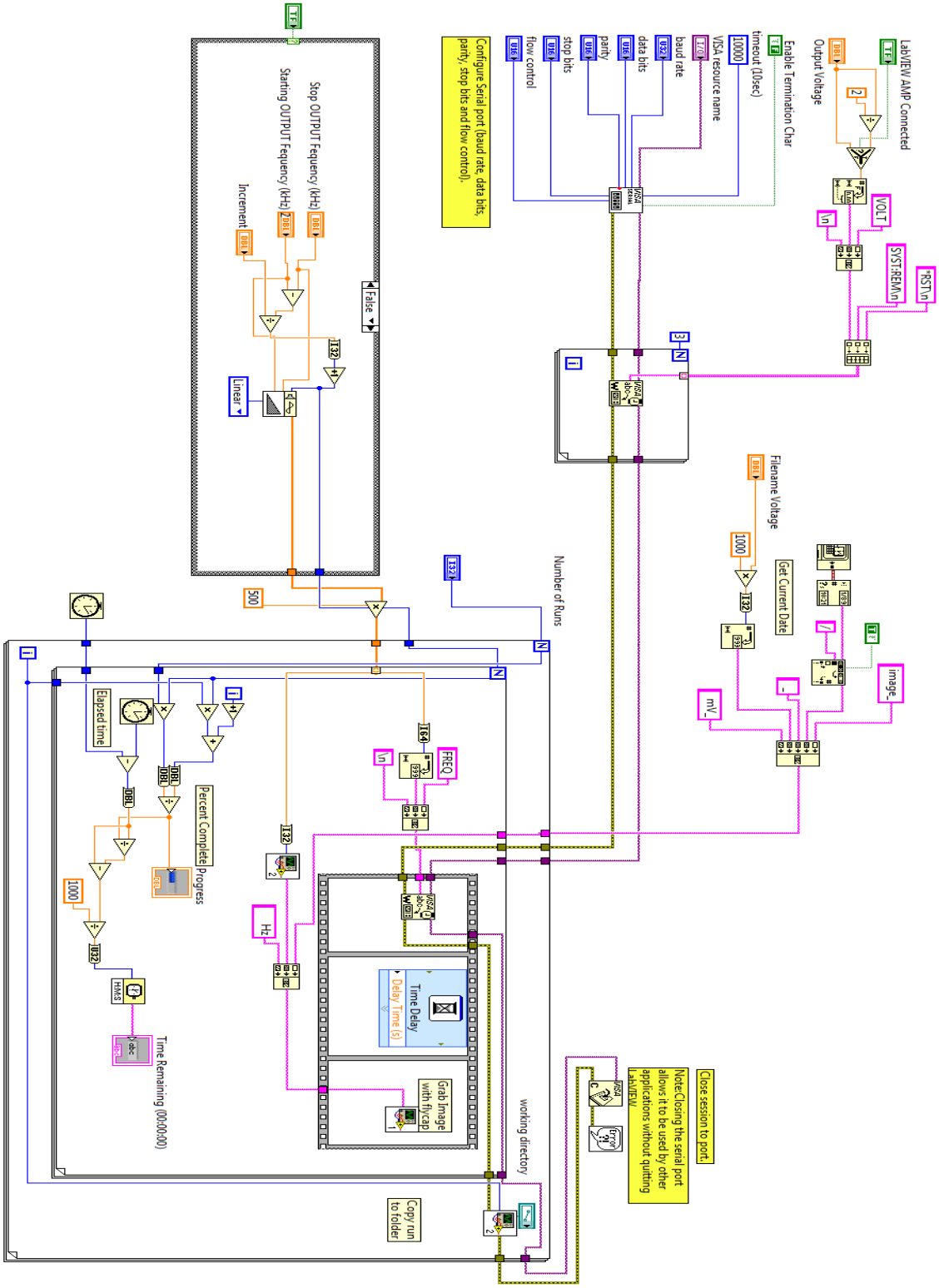


Figure 74: Block Diagram of LabVIEW Program Used for Macro and Micro Scale Tests

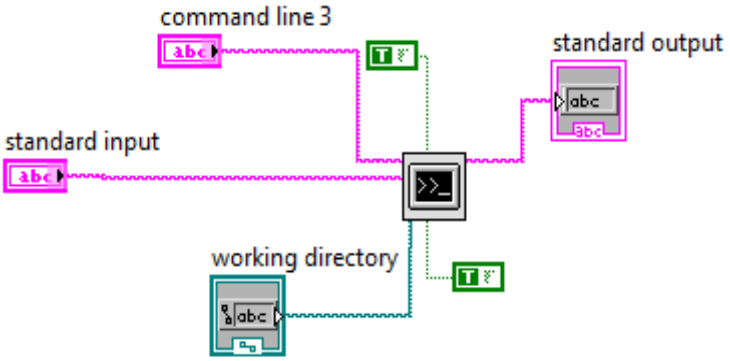


Figure 75: SubVI-1 Program Interacts with FlyCap Program for Capturing Images with Camera

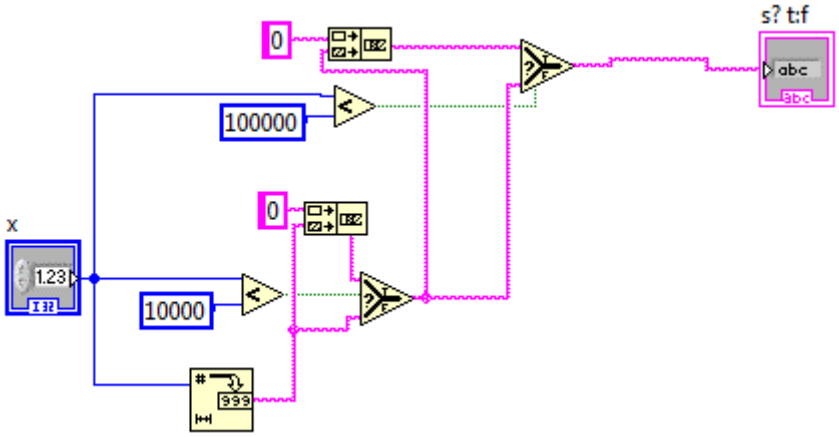


Figure 76: SubVI-2 Program Interacts with Function Generator for Frequency Sweeping

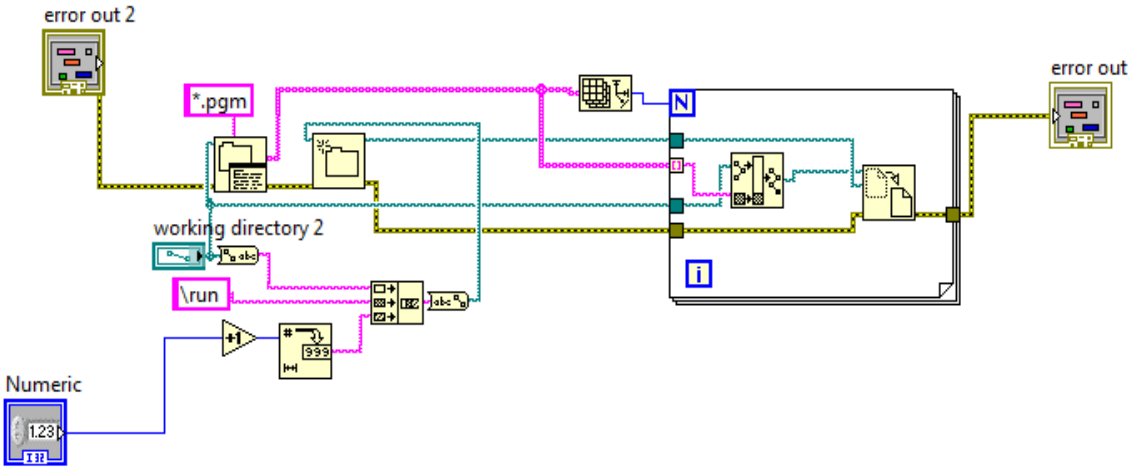


Figure 77: SubVI-3 Program Sends Captured Images to "Run" Folders

Appendix B: MATLAB Codes

B.1: Adding Noise

```
% PUPROSE OF PROGRAM:
%
% reads an image
% perform desired modifications (blur, noise) on image
% re-save image with "modified" appended in front of
% original file name
% modified files are saved in a new folder called "modified" created
% inside folder with images
%
% USER INPUTS:
%               file extension of images (ex: pgm, jpg...)
%               AXIS of blur motion (X or Y)
%               number of blur pixels (ex: 0, 1, 2, 3,...)
%               amount of noise level (0-50%)
%
% OUTPUTS:
%               modified image files saved in original format


clc
close all
clear all

%
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000
%
% File setup -----------------------------------------------------------
-----------------------------------------------------
%
%
000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000

path = uigetdir;
% user selects file location containing runs
cd(path)
% change working directory to selected file

% User enters file extension (example: pgm, jpg)
file_extension = input('WHAT IS THE FILE EXTENSION? (without .) (press
enter for default = pgm)  ','s');
if isempty(file_extension) == 1
    file_extension = 'pgm';
% allows user to press enter instead of default
end

list_of_images=dir(strcat(path,'\*.',file_extension));
num_pics = size(list_of_images,1);
% number of pictures in folder
```

```
[SUCCESS,MESSAGE,MESSAGEID] = mkdir('modified');
% create new folder inside root folder called "modified"


%
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooo
%
% IMAGE MODIFIATION LOOP ------------------------------------------------
-----------------------------------------------------
%
%
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooo

% User enters axis of blur motion
axis_r = input('WHAT IS the AXIS of blur motion? ( x or y press enter
for default = y)  ','s');
if isempty(axis_r) == 1
    angle = 90;
% allows user to press enter instead of default
    axis_r = 'y';
elseif strcmp(axis_r,'x') == 1
    angle = 0;
elseif strcmp(axis_r,'X') == 1
    angle = 0;
end


% User enters number of blur pixels
pixel_blur = input('ENTER number of blur pixels (press enter for
default = 10)  ');
if isempty(pixel_blur) == 1
    pixel_blur = 10;
% allows user to press enter instead of default
end

pixel_step = 0.1;

noise_level = input('WHAT IS Noise level?(0-50%) (press enter for
default = 0%)   ');     % user inters pitch for moving comb

if isempty(noise_level) == 1
    noise_level = 0;
end


waitbar_handle = waitbar(0,(['Performing image alterations and re-
saving in "modified" folder...']));  % opens progress bar
for i = 1:num_pics

    file = list_of_images(i).name;
% file name of images
    file_name = file(1:end-4);
% remove extension from file name
```

```
    image = imread(file);
% read ith image
    image = im2double(image);
% convert image from unit8 to double (for more accurate operations)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % IMAGE MODIFICATION

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    dims=size(image);
    height=dims(1);
    width=dims(2);
    brightest_pixel=max(max(image))

    if pixel_blur>0

        imagesum=zeros(size(image));

        noise_image=(ones(height,width).*-
0.5+rand(height,width)).*noise_level/100;

        for t = -pixel_blur:pixel_step:pixel_blur
            j = sin(t/pixel_blur*pi()/2)*pixel_blur;
            floor_j= floor(j);
            ceil_j = ceil(j);
            if angle == 0
                shift_floor = circshift(image,[0,floor_j]);
                shift_ceil = circshift(image,[0,ceil_j]);
            else
                shift_floor = circshift(image,[floor_j,0]);
                shift_ceil = circshift(image,[ceil_j,0]);
            end
            if floor_j == ceil_j
                imagesum = imagesum+shift_floor+noise_image;
            else
                imagesum = imagesum+(shift_floor*(ceil_j-j)/(ceil_j-
floor_j)+shift_ceil*(j-floor_j)/(ceil_j-floor_j))+noise_image;
            end
        end

        image = imagesum./(2*pixel_blur/pixel_step+1);

    else

         noise_image=(ones(height,width).*-
0.5+rand(height,width)).*noise_level/100;
         image = image+noise_image;

    end

%     for i=-numblurs:numblurs
```

```
%              if angle == 0
%                  imagetemp=circshift(image,[0,i]);
%              else
%                  imagetemp=circshift(image,[i,0]);
%              end
%              imagesum=imagesum+imagetemp/(2*numblurs+1);
%          end
%          image=imagesum;


    %image(95:110,100:400) = image(95:110,100:400) / 2;




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % RE-WRITE DOUBLE TO ORIGINAL IMAGE FORMAT

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    write_file = ([path,'\modified\','modified_',file]);
% new file name with directory to new folder and "modified" added to
filename
    imwrite(image,write_file,file_extension)
% write image to appropriate format

    waitbar(i/num_pics(1),waitbar_handle)
% increment the progress bar based on loop counter
end
close(waitbar_handle)
% close progress bar - image processing complete

display(['Program completed'])
```

B.2: Changing Brightness

```
% PUPROSE OF PROGRAM:
%
% reads an image
% perform desired modifications (blur, brightness) on image
% re-save image with "modified" appended in front of original file name
% modified files are saved in a new folder called "modified" created
% inside folder with images
%
% USER INPUTS:
%               file extension of images (ex: pgm, jpg...)
%               AXIS of blur motion (X or Y)
%               number of blur pixels (ex: 0, 1, 2, 3,...)
%               brightness variation (0-1)
%
% OUTPUTS:
%               modified image files saved in original format


clc
close all
clear all

%
00000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
%
% File setup ----------------------------------------------------------
----------------------------------------------------
%
%
00000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

path = uigetdir;
% user selects file location containing runs
cd(path)
% change working directory to selected file

% User enters file extension (example: pgm, jpg)
file_extension = input('WHAT IS THE FILE EXTENSION? (without .) (press
enter for default = pgm)  ','s');
if isempty(file_extension) == 1
    file_extension = 'pgm';
% allows user to press enter instead of default
end

list_of_images=dir(strcat(path,'\*.',file_extension));
num_pics = size(list_of_images,1);
% number of pictures in folder

[SUCCESS,MESSAGE,MESSAGEID] = mkdir('modified');
% create new folder inside root folder called "modified"
```

```
 %
0000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
%
% IMAGE MODIFIATION LOOP --------------------------------------------
----------------------------------------------------
%
%
0000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

% User enters axis of blur motion
axis_r = input('WHAT IS the AXIS of blur motion? ( x or y press enter
for default = y)  ','s');
if isempty(axis_r) == 1
    angle = 90;
% allows user to press enter instead of default
    axis_r = 'y';
elseif strcmp(axis_r,'x') == 1
    angle = 0;
elseif strcmp(axis_r,'X') == 1
    angle = 0;
end


% User enters number of blur pixels
pixel_blur = input('ENTER number of blur pixels (press enter for
default = 10)  ');
if isempty(pixel_blur) == 1
    pixel_blur = 10;
% allows user to press enter instead of default
end

pixel_step = 0.1

bright_var = input('WHAT IS Brightness variation?(0-1) (press enter for
default = 0)   ');     % user inters pitch for moving comb

if isempty(bright_var) == 1
    bright_var = 0;
end


waitbar_handle = waitbar(0,(['Performing image alterations and re-
saving in "modified" folder...']));  % opens progress bar
for i = 1:num_pics

    file = list_of_images(i).name;
% file name of images
    file_name = file(1:end-4);
% remove extension from file name

    image = imread(file);
% read ith image
```

```
    image = im2double(image);
% convert image from unit8 to double (for more accurate operations)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % IMAGE MODIFICATION

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    dims=size(image);
    height=dims(1);
    width=dims(2);

    if pixel_blur>0

        imagesum=zeros(size(image));

        brightness=(1-bright_var)+2*bright_var*rand

        for t = -pixel_blur:pixel_step:pixel_blur
            j = sin(t/pixel_blur*pi()/2)*pixel_blur;
            floor_j= floor(j);
            ceil_j = ceil(j);
            if angle == 0
                shift_floor = circshift(image,[0,floor_j]);
                shift_ceil = circshift(image,[0,ceil_j]);
            else
                shift_floor = circshift(image,[floor_j,0]);
                shift_ceil = circshift(image,[ceil_j,0]);
            end
            if floor_j == ceil_j
                imagesum = imagesum+shift_floor.*brightness;
            else
                imagesum = imagesum+(shift_floor*(ceil_j-j)/(ceil_j-
floor_j)+shift_ceil*(j-floor_j)/(ceil_j-floor_j)).*brightness;
            end
        end

        image = imagesum./(2*pixel_blur/pixel_step+1);

    else

        brightness=(1-bright_var);

        image = image.*brightness
    end

%     for i=-numblurs:numblurs
%             if angle == 0
%                 imagetemp=circshift(image,[0,i]);
%             else
%                 imagetemp=circshift(image,[i,0]);
%             end
%             imagesum=imagesum+imagetemp/(2*numblurs+1);
```

```
%           end
%           image=imagesum;


    %image(95:110,100:400) = image(95:110,100:400) / 2;




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % RE-WRITE DOUBLE TO ORIGINAL IMAGE FORMAT

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    write_file = ([path,'\modified\','modified_',file]);
% new file name with directory to new folder and "modified" added to
filename
    imwrite(image,write_file,file_extension)
% write image to appropriate format

    waitbar(i/num_pics(1),waitbar_handle)
% increment the progress bar based on loop counter
end
close(waitbar_handle)
% close progress bar - image processing complete

display(['Program completed'])
```

B.3: Original FFT Analysis Program

```
% Dalhousie University
% MEMS Lab
% Ben Bschaden
% fft_blur_V8_0.m
% August 9, 2011
%
% PUPROSE OF PROGRAM:
%               Crop a set of images
%               row average each to produce line profile
%               calculate corresponding fft
%               repeat for all runs in folder
%
% USER INPUTS:
%               number of runs in folder, file location, file type
%               skip images (ex: 1,5,9,13), skip initial images
%               micron/pixel, axis of image rotation
%
% DISPLAYS:
%               1st fft vs harmonic, 3 middle harmonics vs frequency
%
% OUTPUTS:
%               csv table:
%               header 1 - file directory and file name
%               header 2 - entetered micron/pixel
%               header 3- motion of image (x,y,d)
%               header 4 - frequency and harmonic number
%               rows = image num/frequency,
%               columns = harmonics
%



clc
close all
clear all

%
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
%
% File setup ------------------------------------------------------------
--------------------------------------------------
%
%
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000

error_scale = 0.8;
% used for zero crossing, can be adujusted if waves are missed or noise
included


path = uigetdir;
% user selects file location containing runs
```

```
cd(path)
% change working directory to selected file

% User enters number of runs in folder
runs = input('HOW MANY RUNS ARE IN THE FOLDER? (press enter for default
= 5)  ');
if isempty(runs) == 1
    runs = 5;
% allows user to press enter instead of default
end


% User enters file extension (example: pgm, jpg)
file_extension = input('WHAT IS THE FILE EXTENSION? (without .) (press
enter for default = pgm)  ','s');
if isempty(file_extension) == 1
    file_extension = 'pgm';
% allows user to press enter instead of default
end


display(' ')

% User enters image step (allows images to be skipped for faster
processing)
pic_step = input('ANALYSE EVERY Nth IMAGE: WHAT IS N? (press enter for
default = 1)  ');
if isempty(pic_step) == 1
    pic_step = 1;
% allows user to press enter instead of default
end

% User enters image start number (allows images to be skipped for
faster processing)
im_start = input('START AT WHICH IMAGE NUMBER? (press enter for default
= 1 (first))  ');
if isempty(im_start) == 1
   im_start = 1;
% allows user to press enter instead of default
end

display(' ')

% User enters micron/pixel
micron_per_pixel = input('WHAT IS THE MICRON/PIXEL? (press enter for
default = 0.163 um/pix)   ');
if isempty(micron_per_pixel) == 1
   micron_per_pixel = 0.163;
% allows user to press enter instead of default
end


display(' ')
display('Program expects x axis motion, image may need to be rotated:')
```

```
% User enters axis of motion
axis_r = input('WHAT IS the AXIS of motion? ( x, y or d for diagonal,
press enter for default = y)  ','s');
if isempty(axis_r) == 1
    angle = 90;
% allows user to press enter instead of default
    axis_r = 'y';
elseif strcmp(axis_r,'x') == 1
    angle = 0;
elseif strcmp(axis_r,'X') == 1
    angle = 0;
elseif strcmp(axis_r,'d') == 1
    angle = 45;
elseif strcmp(axis_r,'D') == 1
    angle = 45;
end



%
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
%
% Loop through runs in folder ---------------------------------------
--------------------------------------------------
%
%
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000

for loop = 1:runs
    close all
    clearvars -except rect loop error_scale path runs file_extension
voltage pic_step im_start micron_per_pixel angle axis_r
    path_images = ([path,'\run ',num2str(loop)]);
    cd(path_images)
% change directory to folder containing images


% Creates list of all images in folder with desired filename and
extension
list_of_images=dir(strcat(path_images,'\*.',file_extension));
% Select only images with correct file extension
num_pics = size(list_of_images);
% number of pictures in folder
file = list_of_images(1).name;
% file name of images
start_f = file(end-11:end-6);
% extract frequency from file name
start_freq = str2num(start_f)/1000;
% confert string to double and then to kHz

file2 = list_of_images(2).name;
% read second frequency
second_f = file2(end-11:end-6);
second_f = str2num(second_f)/1000;
```

```
step_freq = second_f - start_freq;
% calculate frequency step

voltage = file(end-18:end-15);
voltage = str2num(voltage)/1000;
% Variables for rotation function
rotation_method = 'bilinear';
% used in image rotate function
bbox = 'loose';
% used in image rotate function


% Initiates table to be saved as a csv file: creates first column in
fft table
fft_data = im_start:pic_step:((num_pics(1)));
% vector in picture number
fft_data = ((fft_data-1)*step_freq) + start_freq;
% convert picture number to frequency
fft_data = (fft_data')*2;
% transpose to column vector



%
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
%
% EXTRACT FIRST PROFILE ----------------------------------------------
--------------------------------------------------
%
%
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000




first_image = imread(list_of_images(1).name);
% read in first image

if strcmp(file_extension,'jpg') == 1
    first_image = rgb2gray(first_image);
% if needed, convert jpeg to grayscale
end

first_image = imrotate(first_image,angle,rotation_method,bbox);
% rotate image


% Allows to enter crop box or use previous rectangle
if loop ==  1
    title('WAITING FOR USER TO SELECT MOVING COMB CROP BOX (double
click to finalize)'),hold on
    [roi, rect]=imcrop(first_image);
% crop image by user input
```

```
else
    roi = imcrop(first_image,rect);
end
cross_section = mean(roi);
% row average to get line profile

cross_section_mean = (mean(cross_section));
% cross section mean
cross_section = cross_section - cross_section_mean;
% subtract mean from cross section
crop_width = size(cross_section,2);
% sample size in pixels
samples = crop_width;
hold off
close all


% Perform FFT calculations to find approximate pitch
freq = (0:(samples-1))/(samples*micron_per_pixel);
% create frequency vector

f_f_t = fft(cross_section);
% compute fast Fourier transform of profile
f_f_t_mag = abs(f_f_t);
% compute magnitude of fft result
f_f_t_mag(1) = 0;
% set first value of magnitude to 0 (error due to DC bias)
[peak_approx,pitch_freq_approx] = max(f_f_t_mag);
% untruncated fft estimate
pitch_approx = 1/(freq(pitch_freq_approx));
pixel_per_wavelength_approx = pitch_approx/micron_per_pixel;
% calculated pixels/wavelength


%
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
%
% Zero crossing TO CALCULATE UM/PIXEL ---------------------------------
------------------------------------------------------
%
%
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000

% NOTE: This section is required to prevent error in fft for non
integer number of waves

cross_section_mean = 0;
crossing_count = 0;
% initialize crossing counter
cross_section_location = 0;
crossing_check_old = cross_section(1) - cross_section_mean;
% first loop iteration (to prevent i-1 = 0local_peak = 0;
```

```
for i = 2: size(cross_section,2)
% used to check if crossing has occurred
    if cross_section(i)*cross_section(i-1) <= 0
% if true, crossing has occurred
        if crossing_count == 0
            crossing_count = crossing_count + 1;
% crossing has occurred
            cross_section_location(crossing_count) = i;
% save location of crossing (i = pixel number)
        else
            wave_length_check = i -
cross_section_location(crossing_count);
            if (wave_length_check) > (pixel_per_wavelength_approx *
error_scale)                % if true, crossing has occurred
                crossing_count = crossing_count + 1;
% crossing has occurred
                cross_section_location(crossing_count) = i;
% save location of crossing (i = pixel number)
            end
       end
    end

end

num_waves = crossing_count -1;
% number of integer waves possible


figure('Name','FIRST IMAGE  CROSS
SECTION','NumberTitle','off'),plot(cross_section,'xk')
% plot the first image cross section
hold on


samples = cross_section_location(end) - cross_section_location(1);
cross_section_mean_x =
cross_section_location(1):1:cross_section_location(end);
% length of line to plot (truncated waves)


cross_section =
cross_section(cross_section_location(1):(cross_section_location(1) +
samples-1));% crop to determined size for integer waves
cross_section_mean_vect = ones(1,size(cross_section_mean_x,2)) *
cross_section_mean;           % make vector for mean line plot

% Plot results
plot(cross_section_mean_x,cross_section_mean_vect,'-
r'),xlabel('PIXELS'),ylabel('INTENSITY')
legend('CROSS SECTION ','CROSS SECTION MEAN','Location','Best')
cross_section_x = cross_section_location(1): (cross_section_location(1)
+ samples -1);
plot(cross_section_x,cross_section,'o-b')
plot(cross_section_location,0,'*g')
pause(1)
```

```
pitch = micron_per_pixel*samples/num_waves;
% calculated picth pased on resolution

pixel_per_wavelength = pitch/micron_per_pixel;
% calculated pixels/wavelength

clc
%  If in the first run, display zero crossing results and ask for
confimation
if loop == 1
    display(['APPROXIMATED PIXEL/WAVELENGTH = '
num2str(pixel_per_wavelength_approx)])
    display(['CALCULATED PIXEL/WAVELENGTH WITH TRUNCATION = '
num2str(pixel_per_wavelength)])
    display(['CALCULATED NUMBER OF WAVES = ' num2str(num_waves)])
    display(' ')
    display('GREEN DOTS INDIACATE CALCULATED COMPLETE WAVE SECTIONS')
    display('CHECK ZERO CROSSING FIGURE TO ENSURE CORRECT CALCULATIONS:
PRESS ENTER TO CONTINUE, Control + C TO RESTART AND SELECT NEW CROP
BOX')
    pause
    clc
end


%
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
%
% Find peaks ------------------------------------------------------------
-------------------------------------------------
%
%
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000

% Perform FFT calculations
freq = (0:(samples-1))/(samples*micron_per_pixel);
% create frequency vector
wave_length = 1./freq;
% wavelength = 1/frequency

f_f_t = fft(cross_section);
% compute fast Fourier transform of profile
f_f_t_mag = abs(f_f_t);
% compute magnitude of fft result
f_f_t_mag(1) = 0;
% set first value of magnitude to 0 (error due to DC bias)


%
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
%
```

```
% FFT ANALYSIS -------------------------------------------------------
-----------------------------------------------------
%
%
0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000


waitbar_handle = waitbar(0,(['Performing FFT calculations on images for
run ',num2str(loop),' of ' , num2str(runs),'...']));                %
opens progress bar
for i = im_start:pic_step:num_pics
    if i == im_start + pic_step
        close(waitbar_handle)
        waitbar_handle = waitbar(0,(['Performing FFT calculations on
images for run ',num2str(loop), ' of ' , num2str(runs),'...']));
% opens progress bar
    end
    first_image = imread(list_of_images(i).name);
% read in first image

    if strcmp(file_extension,'jpg') == 1
        first_image = rgb2gray(first_image);
% if needed, convert jpeg to grayscale
    end

    first_image =im2double(first_image);
% convert unit8 to double
    first_image = imrotate(first_image,angle,rotation_method,bbox);
% rotate image to user input angle

    roi=imcrop(first_image,rect);
% crop each image to previously user defined window

    cross_section = mean(roi);
% row average to get line profile
    cross_section = cross_section - mean(cross_section);

    cross_section =
cross_section(cross_section_location(1):(cross_section_location(1) +
samples-1));% crop each profile to integer wavelength



    % Perform FFT calculations
    freq = (0:(samples-1))/(samples*micron_per_pixel);
% create frequency vector

    f_f_t = fft(cross_section);
% compute fast Fourier transform of profile
    f_f_t_mag = abs(f_f_t);
% compute magnitude of fft result
    f_f_t_mag(1) = 0;
% set first value of magnitude to 0 (error due to DC bias)
    [peak_value,first_harmonic] = max(f_f_t_mag);
% find the location of the first peak
```

```
    freq_h = freq/freq(first_harmonic);

    % Plot first fft results
    if i == im_start
% plots only first loop image
        size_p = floor(size(freq,2)/8);
% eight of  the total size of fft (mirror image)
        f_plot = freq_h(1:size_p);
% truncate frequency vector to this size
        fft_plot = f_f_t_mag(1:size_p);
% truncate fft to this size
        figure('Name','FIRST IMAGE
FFT','NumberTitle','off'),plot(f_plot,fft_plot),xlabel('HARMONIC'),ylab
el('MAG.')
    end

    num_peaks = 15;
    % Find location of harmonics based on defined pitch
    for z = 1:num_peaks
        freq_h_temp = (round(freq_h*1000))/1000;
        peak_point = find(freq_h_temp == z);
        fft_data(((i+pic_step-im_start)/pic_step),z+1) =
f_f_t_mag(peak_point);     % stores value of fft at specific harmonic
and stores in a table
    end
      waitbar(i/num_pics(1),waitbar_handle)
% increment the progress bar based on loop counter
end
close(waitbar_handle)
% close progress bar - image processing complete

%
00000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
%
% Plot results
% ---------------------------------------------------------------------
-------------------------------------------------------
%
%
00000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000
%
% Plot normalized values for 3 middle harmonics (example: 15 harmonics
- plot 6 8 10)

display_harm = ceil(num_peaks/2);
% display the ~ middle harmonic

%
(add 1 to index due to first row of image #)
normalized_harm_a = (fft_data(:,display_harm -
2+1))./(max(fft_data(:,display_harm -2+1)));% middle harmonic -2 data
to plot
```

```
normalized_harm_b = (fft_data(:,display_harm
+1))./(max(fft_data(:,display_harm +1)));    % middle harmonic data to
plot
normalized_harm_c = (fft_data(:,display_harm
+2+1))./(max(fft_data(:,display_harm +2+1)));% middle harmonic +2 data
to plot


figure('Name','MID HARMONICS','NumberTitle','off'),hold on
% open figure
plot(fft_data(:,1),normalized_harm_a,'r')
% plot mid -2 harmonic
plot(fft_data(:,1),normalized_harm_b,'g')
% plot mid harmonic
plot(fft_data(:,1),normalized_harm_c)
% plot mid +2 harmonic

% Create strings for header and add to plot
name_1 = strcat('H ', num2str(display_harm -2));
name_2 = strcat('H ', num2str(display_harm ));
name_3 = strcat('H ', num2str(display_harm +2));
legend(name_1, name_2, name_3,'Location','Best')
hold off



%
00000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000
%
% Print data to csv file ----------------------------------------------
-----------------------------------------------------
%
%
00000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000

% Header names
list{1} = strcat(path);

list{2} = strcat(list_of_images(1).name);

list{3} =
strcat('um/pixel',',',num2str(micron_per_pixel),',','voltage',',',num2s
tr(voltage));

list{4} = strcat('motion',',',axis_r);

list{5} = strcat('First harmonic wavelength in
microns',',',num2str(1/freq(find(freq_h == 1))));

list{6} = strcat(' ');

list{7} =
strcat('Harmonic',',','1',',','2',',','3',',','4',',','5',',','6',',','
```

```
7','','8','','9','','10','','11','','12','','13','','14','','15
');



% Print header names
cd(path)
print_file = strcat('fft_peak_data', num2str(loop), '.csv');
% string for file name to be used
fid = fopen(print_file,'w');
% open file and write over any current file with same name

for i = 1:7
    list_print = list{i};
% turn header cell into string
    fprintf(fid,'%s\r\n',list_print);
% print header
end

fclose(fid);
% close file

% Print fft data to csv file below headers
dlmwrite(print_file, fft_data, '-append')
% append data to header file
pause(3)
end

display(' ')
display('PROGRAM COMPLETED, FILE SAVED TO ROOT FOLDER CONTAINING IMAGE
RUNS AS fft_peak_data*.csv')  % user prompt that program is complete
```

B.4: Averaging CSV of Multiple Runs

```
% Dalhousie University
% MEMS Lab
% Ben Bschaden
% averaged_fft_blur_V1_2_multiple_n.m
% August 6, 2011
%
% PUPROSE OF PROGRAM:
%               read csv file from fft_blur.m program output
%               average fft data for 1, 2... n runs
%
% USER INPUTS:
%               pin of pad used, direction of frequency sweep
%               skip images (ex: 1,5,9,13), skip initial images
%
% DISPLAYS:
%               if harmonics in all runs do not match, error message is
%               displayed and no file is saved
%
% OUTPUTS:
%               n csv tables:
%               harmonics, averaged fft data
%


clc
clear all
close all



path = uigetdir;
% user selects file location containing images
cd(path)
% change working directory to selected file

% User enters pin number for pad used
pin = input('WHAT PIN IS POWERED (press enter for default = 57)    ');
if isempty(pin) == 1
    pin = 57;
% allows user to press enter instead of default
end



% User enters direction of scan
direction = input('WHAT DIRECTION WAS THE SCAN (press enter for default
= f, for forward, b for backward)    ' ,'s');
if isempty(direction) == 1
    direction = 'f';
% allows user to press enter instead of default
end


% Creates list of all images in folder with desired filename and
extension
```

```
list_of_files=dir(['fft_peak_data*.csv']);
% Select only images with correct file extension
num_files = size(list_of_files,1);
% number of pictures in folder

rng = [2 3 2 3];
voltage = csvread('fft_peak_data1.csv',2,3,rng);
% extract voltage from csv file

for j = 1:num_files
    data(:,:,j) =
csvread(strcat('fft_peak_data',num2str(j),'.csv'),7,1);
% read all fft data and save as variable data
end



rng = ([6 1 6 (size(data,2)) ]);
sum_data = 0;

for i = 1:size(data,3)
sum_data = sum_data + data(:,:,i);
% add all fft data
harmonics(:,i) =
csvread(strcat('fft_peak_data',num2str(i),'.csv'),6,1,rng);
% read harmonics
average_data(:,:,i) = sum_data/i;
% divide fft data sum by number of runs used
end




rng = ([7 0 (size(data,1)+6) 0]);
print_data = csvread('fft_peak_data1.csv',7,0,rng);
% read frequency data from csv file
start_f = print_data(1);
% extract start frequency
end_f = print_data(end);
% extract end frequency
step_f = print_data(2) - print_data(1);
% calculate frequency step

for k = 1:size(data,3)
    print_data(:,1,k) = csvread('fft_peak_data1.csv',7,0,rng);
% store frequency data for n number of runs
    for i = 1:size(data,2)
        print_data(:,i+1,k) = average_data(:,i,k);
% append fft data to frequency array

        for j = 1:size(data,3)-1
            if harmonics(i,j) == harmonics (i,j+1)
% check that all harmonics are equal
            else
```

```
                display('HARMONICS DO NOT MATCH!!!')
% display error message if harmonics are not equal
                return
% break out of program and do not save file if harmoincs are not equal
            end
        end
    end

   % PRINT TO CSV FILE AFTER EVERY LOOP
    print_file =
strcat('averaged_fft_data_pin_',num2str(pin),'_',num2str(voltage),'V_',
num2str(start_f),'_to_',num2str(end_f),'_step_',num2str(step_f),'_',num
2str(direction),'_',num2str(k),'.csv');

    fid = fopen(print_file,'w');
% open file and write over any current file with same name
    fprintf(fid,'%s\r\n','kHz');
% print header
    fclose(fid);
% close file

    harmonic = (harmonics(:,1))';

    dlmwrite(print_file, harmonic,'roffset',0,'coffset',1, '-append')
% append data to header file
    dlmwrite(print_file, print_data(:,:,k), '-append')
% append data to header file

end


display(['PROGRAM COMPLETED, FILE SAVED TO ROOT FOLDER CONTAINING IMAGE
RUNS AS
"averaged_fft_data_pin_',num2str(pin),'_',num2str(voltage),'V_',num2str
(direction),'.csv"'])    % user prompt that program is complete
```