

INSTRUMENTATION, TECHNIQUES, AND EVALUATION OF ePTV FOR
PARTICLE MANIPULATION STUDIES USING MICRO-SCALE OSCILLATORS

by

Sajad Kafashi

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Mechanical Engineering

Charlotte

2017

Approved by:

Dr. Stuart T. Smith

Dr. Scott D. Kelly

Dr. Tony L. Schmitz

Dr. James M. Conrad

Dr. Susan Sell

ABSTRACT

SAJAD KAFASHI. Instrumentation, techniques, and evaluation of ePTV for particle manipulation studies using micro-scale oscillators. (Under direction of DR. STUART T. SMITH)

A need for dynamic micro-particle manipulation is the ability to position fragile particles without damaging them, for instance biological particles like blood cells, stem cells, neurons, pancreatic β cells, DNA, chromosomes, for repeated measurement without altering their behavior. An oscillating fiber will induce vortices in a slurry of particles, subsequently the vortex force created by this oscillation attracts and traps the particles located at steady streaming micro-eddies. If multiple oscillatory fibers are placed inside the slurry, depending on frequency and timing of oscillation this method can be used for contact-free particle shepherding and sorting and for transporting particles from one location to another. Due to the complicated dynamics of particles traveling in the fluid and the presence of noise, and significant number of particles, attempts to use commercial PIV softwares to track individual particle paths could not discriminate real particles from noise interference. To enhance identification and tracking of individual particles a novel encoded-particle tracking velocimetry (ePTV) technique is developed in this dissertation work and used in the experiments to track the particle trajectories. An analytic model is developed to determine the number of lost particles due to the finite image size based on a calculation of the probability that imaged particles of a specific mean velocity or having a uniform velocity distribution and encoding pattern will exit the field of view. The encoded pulse technique has been implemented in experiments for which images containing 100-

200 objects including encoded trajectories have been measured. Using the developed ePTV algorithm approximately 30 % of the identified objects were classified as an encoded particle trajectory. Two types of oscillation mechanism are used in the experimental component of this study, a PZT flexure-based macro-probe driven at frequencies around 250 Hz and higher frequency dynamic-absorber, quartz-based, micro-probes driven at frequencies around 32 kHz. Two models for predicting the frequency response of micro-scale oscillatory probes are developed in this dissertation. In these studies, the attached fibers were either 75 μm diameter tungsten or 7 μm diameter carbon with lengths ranging from around 1 to 15 mm. The oscillators used in these experiments were commercial 32.768 kHz quartz tuning forks. Theoretical predictions of the values of the natural frequencies for different vibration modes show an asymptotic relationship with the length and a linear relationship with the diameter of the attached fiber. Similar results are observed from experiment, one with a tungsten probe having an initial fiber length of 14.11 mm incrementally etched down to 0.83 mm, and another tungsten probe of length 8.16 mm incrementally etched in diameter, in both cases using chronocoulometry to determine incremental volumetric material removal. Of particular relevance is that, when a 'zero' is observed in the response of the tine, one mode of the fiber is matched to the tine frequency and is acting as an absorber. This represents an optimal condition for contact sensing and for transferring energy to the fiber for fluid mixing, touch sensing and surface modification applications. Consequently the parametric models developed in this dissertation can be utilized for designing probes of arbitrary sizes thereby eliminating the empirical trial and error previously used.

ACKNOWLEDGMENTS

I would like to take the opportunity to thank: Dr. Stuart T. Smith for sharing his abundant knowledge and wisdom, constant help and novel ideas with me on day to day basis; Dr. Scott Kelly for being supportive and passing on his knowledge in vortex dynamics models and for serving on the thesis committee; Dr. Jeff Eldredge (UCLA) for contributing in bi-weekly meetings, and his expertise in theoretical modeling and simulation; Dr. Susan Sell for serving on the thesis committee; John Brien for electronics advices; Dr. Shane Woody (InsituTec) for his help and insight in micro-probes paper. Dr. Bartosz Nowakowski (NIST) for previous development of the etching apparatus; Ralph Strayhorn (Newell Rubbermaid) for contributing in obtaining experimental frequency response of micro-probes; Jeffery Thousands for discussions during the project and programming advices; Stephen Howard for assisting in experimental testing, Kate Laffitte for development of image acquisition programs; The National Science Foundation (NSF) for project funding (award # DMI 1000656 and DMI 1538814). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

DEDICATION

To my mother and father ...

TABLE OF CONTENTS

LIST OF FIGURES	x
LIST OF TABLES	xvii
LIST OF ABBREVIATIONS	xviii
CHAPTER 1: INTRODUCTION – PROJECT OVERVIEW AND PURPOSE.....	1
CHAPTER 2: LITERATURE REVIEW	4
2.1 Particle micromanipulation, sorting and shepherding	4
2.2 Encoded PIV/PTV	11
2.3 Dynamic absorber-based micro-scale oscillatory probes	15
CHAPTER 3: PARTICLE MANIPULATION EXPERIMENTAL FACILITY.....	17
3.1 Design of instruments	17
3.1.1 Experimental apparatus.....	17
3.1.2 Oscillation mechanism.....	20
3.1.3 Illumination and imaging of the flow fluid.....	22
3.1.4 Camera positioning	26
3.1.5 SAJI waveform generator (STQ2)	33
3.2 Experimental Procedures	38
3.2.1 Volume of particles.....	38
3.2.2 Number of particles.....	39
3.2.3 Solution concentration	40
3.3 Results.....	41
3.3.1 Digital image filtering.....	42
3.3.2 Trajectory tracking.....	45
3.3.3 Velocity measurement	46

3.3.4 Single probe experiment	48
CHAPTER 4: ENCODED-PARTICLE TRACKING VELOCIMETRY (EPTV).....	51
4.1 Mathematical modeling	52
4.1.1 Probability analysis	57
4.1.2 Probability of encoded particle trajectories going outside of the boundary of the image.....	59
4.1.3 Physical interpretation	66
4.2 Experimental Apparatus	69
4.2.1 Particle trajectory tracking.....	73
4.3 Results.....	76
4.3.1 Comparison between theoretical model and simulation studies	80
4.3.2 Statistical analysis	85
4.3.3 Probability results versus velocity distribution.....	89
4.3.4 Implementation of pulse coding in experimental trajectory tracking	92
CHAPTER 5: DYNAMIC ABSORBER BASED, MICRO-SCALE OSCILLATORY PROBES FOR MICRO-VORTEX EXPERIMENTS	98
5.1 Theory.....	99
5.1.1 Lateral vibration of two beams serially connected	100
5.1.2 Lumped absorber model	104
5.2 Overview of experimental approach.....	113
5.3 Results.....	115
5.3.1 Varying length	116
5.3.2 Varying diameter	121
5.3.3 Lumped system model	123
5.3.4 Finite element analysis.....	126

5.4 Multi-oscillator, micro-vortex experimental apparatus	128
5.4.1 Design	129
5.4.2 Experimental setup.....	132
CHAPTER 6: CONCLUSIONS	134
6.1 Summary	134
6.1.1 Micro-particle manipulation	134
6.1.2 ePTV	134
6.1.3 Micro probes and vortex studies	135
6.2 Future work.....	137
6.2.1 Micro-particle manipulation and PTV/PIV	137
6.2.2 Micro probes and vortex studies	138
REFERENCES	140
APPENDIX A: EQUATIONS FOR GENERATING THE MATRIX GOVERNING MOTION OF MICRO-SCALE OSCILLATORY PROBES	149
APPENDIX B: CODES AND PROGRAMMS.....	152
APPENDIX C: CIRCUITS AND NOTEBOOK	246
APPENDIX D: DRAWINGS AND MODELS	280
APPENDIX E: LIST OF COMPONENTS	281
APPENDIX F: FOLDERS STRUCTURE	284

LIST OF FIGURES

Figure 2-1. The toroidal feeding vortex of Vorticella [27,28].	6
Figure 2-2. Sustained vortex generation with a single micro-scale oscillatory probe	9
Figure 2-3. (Left) Trajectory of an inertial particle of size $a=R = 0.175$ and density $\rho_p/\rho_f = 1$ transported between four cylinders via sequential cylinder oscillations. Motion of each cylinder at $Re = 40$. Transit times between cylinders are on the order of 105 oscillation cycles. (Right) Cylinder motion (indicated by arrow), associated mean streamlines (gray), and particle trajectory during one transit (in blue) [40].	10
Figure 2-4. Time-averaged streamlines within systems of probes modeled in a simplified way that builds upon the analysis in [41].	10
Figure 2-5. Pulse coding techniques for particle Image/tracking velocimetry.	14
Figure 3-1. Current particle manipulation experimental apparatus.	18
Figure 3-2. Photograph of Particle manipulation experimental apparatus with three legs of the bridge-type frame numbered. a) Side view, b) Back view.	19
Figure 3-3. Particle manipulation system control flowchart.	20
Figure 3-4. a) Flexure model and b) Oscillator drive mechanism showing the oscillating rod that has its axis perpendicular to the motion of the piezo electrically actuated flexure stage.	21
Figure 3-5. Diagram of the illumination system. The blue line indicates LED illumination and red is the reflected image to the camera.	23
Figure 3-6. a) An image of the imaging system implemented for the PTV studies, b) Optical setup.	24
Figure 3-7. Circuit for measuring the intensity of light.	25
Figure 3-8. Sensor voltage (V), which is also shown in Figure 3-7, versus time for LED pulsing of 1 KHz, and 50% duty cycle.	26
Figure 3-9. Double pulsing for each camera exposure results in particle pairing	26
Figure 3-10. Automatic 3-axes motion control apparatus assembled	27
Figure 3-11. Top view of base plate with three circles A, B, and C indicating the location of the three actuators of this system. Linear motion at these points enables geometrical calculation of the tilt and axial motions of the center point O, motion is perpendicular to the page.	28

Figure 3-12. Base plate.	30
Figure 3-13. Mounting plate.	30
Figure 3-14. a) Lead screw holder. Dowel pin inserts into the radial hole shown on the left. b) Dowel pin.	32
Figure 3-15. Holding block for preventing rotation of the actuator lead-screws.....	32
Figure 3-16. Camera positioning stage with microprocessor control.	33
Figure 3-17. WaveforGenerate Board with description of its main features.....	34
Figure 3-18. WaveGenerate pin connections to Arduino-UNO.	35
Figure 3-19. Maximum voltage generated at different frequencies.....	36
Figure 3-20. WaveformGenerate.ino for generating a simple signal using the board when connected to an Arduino™ micro-computer.....	37
Figure 3-21. WaveGenerateLabviewControl.VI (Frequency Sweep Mode) that can be operated from any windows based operating system.....	38
Figure 3-22. a) Image of the master spoon, and b) Geometry of rectangular cube	39
Figure 3-23. Photograph showing four bottles of slurry with concentrations of particles ranging from 1 /400 G·mL ⁻¹ to 25/3 /400 G·mL ⁻¹ (see definition in text) used in experiments.....	41
Figure 3-24. Filtering by thresholding algorithm.	42
Figure 3-25. Filtering by thresholding sequence.	43
Figure 3-26. a) Original image, b) Filtered image.....	43
Figure 3-27. Filtering by subtracting two images. a, b and c show different images. d, e and f show filtering the noise and stationary objects by subtracting images from each other.	44
Figure 3-28. Flow fields image with 270 Hz rod frequency, 10.6 mm reservoir inner diameter, 30 ms LEDon, 30 ms LEDoff, 30ms camera off, 120-150 μm diameter spheres, and 25/3/400GmL ⁻¹ concentration. a) 2 blinks, b) 3 blinks, c) 10 blinks, d) 20 blinks.....	45
Figure 3-29. a) example of a particle in two different position and b) Image of a ruler for calibration.....	46

Figure 3-30. a) An image of three-pulsed video experiment, and b) Trajectories of particles shown in red box after image processing.	47
Figure 3-31. Contour of a) streamline and b) vorticity from Boundary Element Methods [74].	49
Figure 3-32. a) Raw video frame of particle motion around the probe with circular cross section b) filtered image showing only the moving particles c) Velocity field data reconstructed through comparisons of individual particle positions in successive frames. The blue ellipse shows the position of the probe.	50
Figure 4-1. Definition of the geometric factors of the mathematical model for encoded-PTV for which the particle located at coordinates Q_1 is being evaluated. In this diagram the angular orientation of the θ segment is arbitrary and drawn to visualize the angle limit for the condition in equation (2). All parameters are geometrically determined while θ and γ (thereby enabling calculation of ΔR) are set by the user.	54
Figure 4-2. Image of particle traveling in 3-D environment. \mathbf{R}_1 and \mathbf{R}_2 represent the vector distances between consecutive particle in a trajectory. ψ and $\Delta \mathbf{R}$ which are angle (steradians) and maximum spacing. Any particle found in this volume would be accepted to be part of the trajectory.	57
Figure 4-3. Image showing particles having different limiting angle in a pattern.	58
Figure 4-4. Diagram of the Image when particles are traveling at maximum velocity $V_{\max} = T$	61
Figure 4-5. Section view of the circle in Figure 4-4 for probability calculation.	62
Figure 4-6. Diagram of the Image when particles are traveling with uniformly distributed velocity $V = \tau$	64
Figure 4-7. Image of particles in a trajectory coded in pattern $\{1\ c\}$	67
Figure 4-8. Elements of optical particle image/tracking velocimetry system.	70
Figure 4-9. Particle tracking velocimetry a) design of the optical setup and imaging system b) experimental apparatus.	72
Figure 4-10. Flowchart of the encoding algorithm for encoding $\{Q_1, Q_2\}$ and maximum angle θ . Sequence shows how to identify encoded trajectory in an image.	73

Figure 4-11. Encoding with $\{1\ 2\}$ pattern, a) image containing particles in a coded patterns b) identified coded trajectories after implementing the encoded-PTV algorithm.....	74
Figure 4-12. Encoding with $\{1\ 2\ 3\}$ pattern, a) image containing particles in a coded patterns b) identified coded trajectories after implementing the encoded-PTV algorithm.....	75
Figure 4-13. Photograph of an image with randomly generated particles density of 100. These images has been used in the developed ePTV software to find the number of (false) found patterns for specific coding strategies.....	76
Figure 4-14. Simulation results for $\rho=30$, $\theta=10-90$, $\gamma=0.01-0.6$, and decoding pattern $\{1\ 2\}$. \bar{N}_s represents the average simulated number of found patterns.....	77
Figure 4-15. Simulation results for $\rho=100$, $\theta=10-360$, $\gamma=0.01-1$, and decoding pattern $\{1\ 2\}$	78
Figure 4-16. Simulation results for $\rho=10-100$, $\theta=10-360$, $\gamma=0.01-1$, and decoding pattern $\{1\ 2\}$	79
Figure 4-17. Simulation results for $\rho=100$, $\theta=10-360$, $\gamma=0.01-1$, decoding pattern $\{1\ 2\}$ vs decoding pattern $\{2\ 1\}$ showing the same rejection capability. Blue color represents $\bar{N}_{s\{2,1\}}$ and cyan color represents $2\bar{N}_{s\{1,2\}}$	80
Figure 4-18. Maximum spacing constraint for coding pattern $\{1\ 2\}$ in an image size of 1×1	82
Figure 4-19. Representation of edge effect problem limiting the searching angle for particles at different locations in an image.	83
Figure 4-20. Edge effect relationship with angle for different γ in coding pattern $\{1\ 2\}$, $\rho=100$	84
Figure 4-21. Theoretical and simulated results for $\rho=100$, $\theta=10-360$, $\gamma=0.01-1$, decoding pattern $\{1\ 2\}$, Blue color represents the simulated \bar{N}_s and cyan color represents the \bar{N}_s obtained from theoretical model.	85
Figure 4-22. Normal variate, Z , and Chi-squared, χ^2 for $\rho=50$ and decoding pattern $\{1\ 2\}$	87

Figure 4-23. Normal variate, Z , and Chi-squared, χ^2 for $\rho = 100$ and decoding pattern {1 2}.....	88
Figure 4-24. Normal variate, Z , and Chi-squared, χ^2 for $\rho = 100$ and decoding pattern {3 1}.....	88
Figure 4-25. Image showing relationship of probability, P_T , of finding good pattern versus h/T when particles are all traveling with the same velocity V_{\max} whereby all particles travel distance T over the duration of a pulse train corresponding to a given coding strategy.....	90
Figure 4-26. Image showing relationship of probability of finding good pattern P_τ versus h/T when particles are traveling with a uniformly distributed velocity between 0 and V during which particles travel distance $0 < \tau \leq T$	91
Figure 4-27. Flow fields image with 270 Hz rod frequency, 10.6 mm reservoir inner diameter, 30 ms LEDon, 30 ms LEDoff, 30ms camera off, 120-150 μm diameter spheres, 25/3/400GmL ⁻¹ concentration. a) 2 blinks, b) 3 blinks, c) 10 blinks, d) 20 blinks.....	92
Figure 4-28. Flow fields image with 1750 Hz rod frequency, 15.8 mm reservoir inner diameter, 30 ms LED on, 50ms camera off, 10 μm diameter spheres, 1/400GmL ⁻¹ concentration. a) {180 284} encoding- filtered image, b) {230 330} encoding- filtered image, c) {132 180, 232} encoding- filtered image, d) {132 180 232} coding- original image. (The colors in the image are inverted for clarity).....	93
Figure 4-29. ePTV algorithm results for experimental image of encoding {180 284} for $\theta = 40$ degrees and $\gamma = 0.2$	95
Figure 4-30. ePTV algorithm results for experimental image of encoding {132 180 232} for $\theta = 40$ degrees and $\gamma = 0.3$	96
Figure 5-1. Model of tuning fork with a tungsten fiber attached to its upper tine showing the critical dimensional parameters used for vibration analysis of the probe system.	100
Figure 5-2. Lumped absorber model of the fiber probe.....	105
Figure 5-3. Input-output linear system model for the probe demonstration.	112
Figure 5-4. Photograph of 75 μm diameter tungsten fiber attached to the upper tine of tuning fork along with the tine axis.....	114

Figure 5-5. An image of etching apparatus used for mass removal of fibers. A voltage applied through carbon rods as cathodes and tungsten (or carbon) fiber as anode creates flow of current in the solution that is monitored as indication of mass removal.	115
Figure 5-6. Theoretical calculation of frequency as a function of length of fiber for first ten resonant modes obtained from theoretical model for two serially connected beams. (Carbon probe).....	116
Figure 5-7. Three visible modes of carbon fiber with 7 μm diameter, a) third mode with length 2.4 mm, b) fourth mode with length 3.3 mm and c) fifth mode with length 4.3 mm. The theory model predicts these length when the natural frequencies of fiber collide with the tune frequency.....	117
Figure 5-8. Theoretical calculation of frequency as a function of length of fiber for first ten resonant modes obtained from theoretical model for two serially connected beams. (Tungsten probe).....	119
Figure 5-9. Experimentally measured frequency response of a tungsten probe sequentially etched to different lengths	121
Figure 5-10. Theoretical calculation of eigenvalues for an 8.16 mm long Tungsten fiber as a function of varying diameter.	122
Figure 5-11. a) Frequency response of tungsten fiber obtained from experimental testing b) subplot of 14-a demonstrating the linear relationship of modal frequencies with fiber diameter.	123
Figure 5-12. Amplitude frequency response of the tuning fork tine, H_{oy} as a function of tungsten fiber length based on lumped model approximation (tungsten probe).	124
Figure 5-13. Phase frequency response of the tuning fork tine as a function of attached tungsten fiber length based on lumped model approximation.....	125
Figure 5-14. Amplitude frequency response of the fiber, H_{iy} for $i = 1, 2, 3$, and 4 respectively as a function of fiber length based on the lumped model approximation (tungsten probe).	126
Figure 5-15. Fifth oscillating mode shape of carbon probe of 4.3 mm fiber length obtained from a) FEA, b) experimental testing.	127
Figure 5-16. FEA result showing modes shapes of tungsten probe of 5.5 mm fiber length oscillating in a) third mode, and b) fourth mode.	127
Figure 5-17. Solid model of multi-oscillator apparatus. Four stepper motors provide motion in X and Y for two moving probes.	129

Figure 5-18. Close up view of the solid model for the micro-vortex multi-oscillator probing arrangement. Bottom right hand image illustrate the image of the micro probes containing a 7 μm diameter carbon fiber glued to a quartz tuning fork tine..... 131

Figure 5-19. Schematic diagram of the probing system in the fluid container. The two upper moving probes are able to move anywhere inside the container along X and Y axes. 132

Figure 5-20. Experimental multi-oscillator apparatus. a) multi-oscillator apparatus attached to the structural frame of the particle manipulation framework [4,5]. b) Close shot of the micro-probes system. 133

LIST OF TABLES

Table 3-1. Pin function description and the typical pin numbers for connecting to Arduino or similar micro-processors	34
Table 3-2 . Achieving solutions with different concentration in multiple steps	41
Table 3-3. Velocities of particle trajectories shown in figure 22.....	48
Table 3-4. Technical parameters for single probe experiment	49
Table 5-1. Descriptions of mathematical parameters shown in Figure 5-2 above and their units.	106
Table 5-2. Description of tuning fork parameters used in equations to determine values for the parameters of the lumped model and serially connected beams analysis, and their values used for images represented in the result section.	108
Table 5-3. Description of carbon and tungsten fiber parameters used in equations to determine values for the parameters of the lumped model and serially connected beams analysis, and their values used for calculation represented in the result section.	110
Table 5-4. The first ten roots for lateral natural frequencies of a free cantilever beam.	111
Table 5-5. Natural frequencies of tine obtained from theoretical calculation demonstrating its behavior acting as clamped-clamped beam.....	120
Table 5-6. First five natural frequencies of the tungsten probe (fiber length= 5.5 mm) obtained from theoretical calculation and FEA.	128
Table B-1. Programs' name and their purpose used in this thesis.	152

LIST OF ABBREVIATIONS

PIV	Particle image velocimetry
PTV	Particle tracking velocimetry
ePTV	Encoded- particle tracking velocimetry
DDS	Direct digital synthesizer
FEA	Finite element analysis
LED	Light emitting diode
CMM	Coordinate measuring machine
AFM	Atomic force microscopy
CCD	Charged-coupled devices
FPGA	Field-programmable gate array
MEMS	Micro-electro mechanical systems
Op-Amp	Operational amplifier
FCC	Face centered cubic
OUT	Output
GND	Ground
AC	Alternative current
DAQ	Data acquisition
PZT	Lead Zirconate Titanate

CHAPTER 1: INTRODUCTION – PROJECT OVERVIEW AND PURPOSE

In this research, investigations of a novel, micro-particle manipulation method by creating vortices inside fluids induced by oscillatory probes has been undertaken. While some micromanipulation technologies currently exist, each has specific attributes and shortcomings (explained in detail in chapter 2). Many of the processes apply mechanical or thermal stresses on the particles while others require controlled environments. The method created in this thesis study is non-contact, and can be used for interdisciplinary applications for instance gently positioning cells or chemicals [1–3], assisted fertilization of human ova [4,5] and micro-polishing process for fragile components like optical lenses, called vortex machining [6–8].

Machine design implementation and mechatronics development, particle tracking, measurement analysis, fluid flow theory, image processing, frequency response and vibration analysis, and mathematical modeling and statistical simulations have been just a few of the areas studied and tested in depth during the course of this research project. In this thesis, a review of related micromanipulation technologies and vortex oscillation research will be presented; followed by detail of the experimental design, ePTV, and micro-oscillatory probe characterization. In conclusion, results from the analysis will be presented along with future work.

This technology will provide *in situ* microfabrication and cell patterning, separation of biological cells and contact-free particle trapping, shepherding and sorting. The trapping

force of the vortex is controlled by oscillation amplitude, while the trapping location is controlled by the oscillation frequency [3,9]

To track the particles moving in the fluid and understand their dynamics, image velocimetry techniques are commonly used. The common method in particle image velocimetry, PIV, and particle tracking velocimetry, PTV, is that tracer particles are added to the flow, assumed to move with the flow velocity dynamics, and the flow with the illuminated particles is recorded with the camera. The difference between PIV and PTV is the method of post analysis of recorded data. In PIV, the images frames are divided into sub-areas called “interrogation areas”. It is assumed that particles in the same interrogation areas move quite homogeneously, and the displacement of the areas in two subsequent frames are determined by cross-correlation technique. On the contrary, PTV identifies each particle separately and identifies the displacement of each individual particle in the next recorded frame. This means that PIV provides velocity vectors in the interrogation areas, whereas PTV yields a displacement vector of each particle. The measurements analyzed in this research were steady stream eddies around the oscillating fibers, where the patterns of particle path lines could not be determined with acceptable uncertainty, one of the reasons to use a PTV algorithm. On the other hand, the vectors obtained by PTV are not regularly placed, so wrong vectors can be more difficult to eliminate, however, coding the particles patterns in specific strategies will eliminate false reading, which is a different reason to implement the developed new technique; encoded-particle tracking velocimetry (ePTV).

A fundamental method of PTV is to search objects in frame 2 that can correspond to the same particle in frame 1 [10]. With the encoded PTV, searching happens in one frame, and it will only look for the ePTV parameters i.e. spacing tolerance, maximum angle

and coding strategy. In practice, depending on pulse times and dynamics of the particle stream, the next particle will follow a relatively shallow arc and it may therefore be reasonable to restrict the search to a limited angle, θ , chosen to encompass the largest expected angle deviations. The ratiometric spacing tolerance, γ , expressed as a ratio of variation in distance relative to that computed based on an ideal distance for a particle travelling in a straight line with uniform velocity. Any particle found in this region defined by γ (will be described in chapter 4 in detail), makes it acceptable to be part of the potential coded pattern. All these parameters are controllable and depending on the desired outcome can be adjusted. The effect of these parameters on the false reading of the ePTV have been studied in full detail which is presented in chapter 4.

CHAPTER 2: LITERATURE REVIEW

This project advances a novel, biologically inspired method for the manipulation of microscopic-scale solid particles suspended in fluids. A background and literature survey is provided on each of the three main initiatives carried out in this thesis. These are particle micromanipulation sorting and shepherding, encoded-particle tracking velocimetry (ePTV) and performance characterization of micro-oscillatory probes.

2.1 Particle micromanipulation, sorting and shepherding

Separating and manipulating discrete objects on a microscopic scale has variety of applications ranging from the mechanical testing of macromolecules like DNA [1] to the assisted fertilization of human ova with immotile sperm [4,5] to vortex machining using abrasive particles for the precision polishing of surfaces [11]. Several technologies have been developed for micromanipulation purposes, each of them having its own shortcomings. Direct-contact systems containing robotic grippers or fine capillaries, for example, can introduced undesirable mechanical stresses on fragile objects like biological cells. Laser-based systems like optical tweezers [12–15] require high intensity illumination and closed-loop control for practical utilization. Ultrasonic systems using vibrating piezoelectrics [16,17] require manipulated particles to be situated within fluid environments engineered to associate distributed boundary control. Other techniques include dielectrophoresis in which a force is exerted on a dielectric particle when it is

subjected to a non-uniform electric field, the interaction between the induced dipole and the varying electric field can generate motion in the particle. This manipulation/separation process is slow, and in order to be effective the particles must ideally be monodisperse. Clustered and densely populated particles may cause significant problem with the attraction of unlike particles [18,19]. In inertial hydrodynamic effects [20–23] individual particles are propelled by a flowing fluid and are separated due to interactions with channel microscale topography and patterns. This inertial particle drift is associated with the interaction of two hydrodynamic forces, a wall-induced lift force that repels particles away from the wall, and a force due to a non-uniform shear in channel flow that makes particles to migrate away from the channel centerline. Depending on the particle size and channel geometry, a balance between these forces sets the particle equilibrium position [24,25]. None of these is ideal for the sorting of fragile objects in an uncontrolled fluid environment due to limited throughput, requiring complex sample preparation and sophisticated external control.

Figure 2-1 illustrates a member of the protozoan genus *Vorticella*, common to freshwater ponds and lakes. Distributed along the lip of *Vorticella*'s bell are cilia which vibrate to create vortices in the surrounding fluid. The mixing properties of this dynamics have recently been studied experimentally [26], however *Vorticella*'s purpose is not merely to mix the fluid nearby. The panel on the left shows a still frame borrowed from a movie available online [27]. Close inspection of this image reveals the clear distribution of particles to the left of the organism along nested circular paths in space. In the video itself, it is apparent that the net result of *Vorticella*'s ciliary motion is the establishment of sustained vortical flows on either side of the organism's body. These vortices are illustrated schematically in the picture on the right, borrowed from [28], which represent opposite

cross sections of a toroidal vortex structure. As nutrients circulate within this structure, they are brought gradually into Vorticella's mouth. This is how the organism feeds on various other microorganisms mostly bacteria [29].

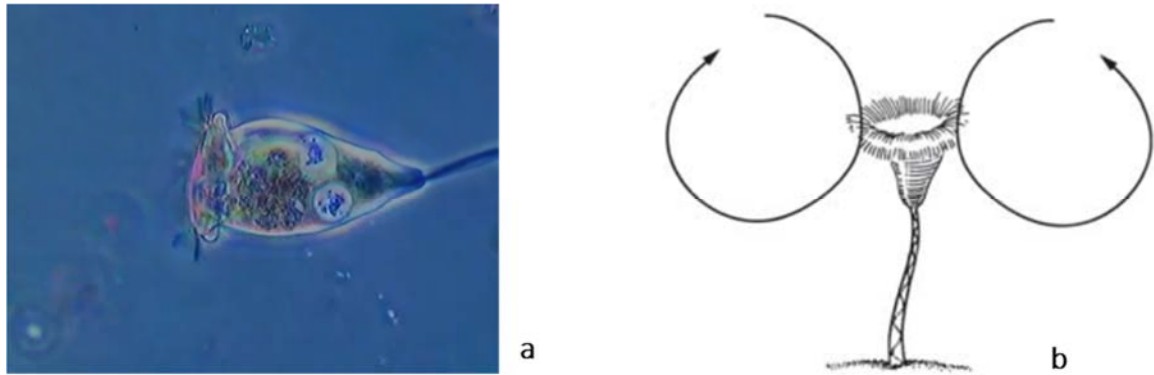


Figure 2-1. The toroidal feeding vortex of Vorticella [27,28].

The details of Vorticella's oscillation strategy—the distribution of amplitudes and frequencies across its body and cilia and the optimality are not known, but the apparently passive nature of this strategy, unmodified by the presence or absence of food particles at particular spatial locations nearby, is consistent with the fact that Vorticella lacks the ability to create or process an active sensory image of its surrounding. The seemingly large dimension of Vorticella's control is actually diminished significantly by tight coupling between the motions of individual cilia [30]. A solitary Vorticella generates a primitive flow structure that manipulates nearby particles to move in a single specific manner.

When several such organisms combine, however, more sophisticated flow structures emerge from their competition for food. Individual particles may be captured and traded repeatedly among distinct feeding vortices, tracing complex, three-dimensional trajectories under the influence of superposed oscillatory stimuli of varying strength, frequency, and location.

The earliest demonstration of steady streaming is given by Andrade et al. in 1931 [31]. They vibrated the air around a cylinder tube at diameters between 1 mm and 5 mm, using smoke particles as tracers. They observed a vortex motion which takes place around any obstacle, provided that the amplitude of oscillation exceeds a certain value. The streaming flow was induced by the interaction between the main fluid body and the non-slip boundary. The outer vortex is clear while the inner vortex is a little bit obscure.

Holtmark et al. [32] performed a designed experiment based on his developed numerical analysis. In his experiment, a cylindrical obstacle with a diameter of 0.11 cm was placed at the antinode of a Kundt's tube, while MgO was used as tracer particles. All images were taken through an objective placed above the Kundt's tube. Holtmark observed that the thickness of the boundary layer is a function of $D\left(\frac{\omega}{\eta}\right)^{1/2}$ and decreases as $D\left(\frac{\omega}{\eta}\right)^{1/2}$ increases. Where D is the diameter of the cylinder, ω is the oscillation frequency and η is the kinematic viscosity.

Sritharan et al. [33] have described a different procedure to induce mixing in a microfluidic channel, in which they used high frequency surface acoustic waves which are coupled through the channel material into the fluid. Acoustic streaming then induces complex material folding lines which significantly enhance the mixing performance in those microfluidic devices. The piezoelectric chip which creates the surface acoustic waves is acoustically coupled to the microfluidic chip which does not necessarily need to be part of the fluidic system. Yang et al. [34] implemented piezoelectric acoustic oscillation to actively mix water and fluorescent dye in a 6 mm×6 mm×0.06 mm chamber.

Dr. Daniel Schwartz group at University of Washington used steady streaming micro-eddies as hydrodynamic tweezers for trapping cells and chemical reagents. They came to the conclusion that the oscillation frequency controls the trapping location while the oscillation amplitude controls the trapping force [35,36]. Later on, they conducted experiments that were able to capture streamlines for 3D steady streaming eddies created around short cylinders in channel [2].

Figure 2-2 depicts a micro-scale oscillatory probe with a 7 μm diameter carbon fiber tip immersed into a millimeter-sized droplet of water in which a large number of 40 μm diameter polymer spheres are suspended. When the probe is induced to vibrate, nearby particles are entrained in a toroidal vortex analogous to that depicted in Figure 2-1, the strength of this vortex is dependent upon the frequency of excitation of the probe. Other efforts to develop artificial cilia are described the recent MEMS literature [37], but the hydrodynamics of ciliary systems — natural or artificial — have yet to be fully comprehended, particularly from the standpoint of control. A thorough elucidation of the dynamics illustrated in Figure 2-2 promises both (1) to lay the groundwork for an inexpensive, scalable technology for the three-dimensional contact-free manipulation of particle systems in fluids and (2) to illuminate a fundamental biophysical mechanism pervading the natural world.

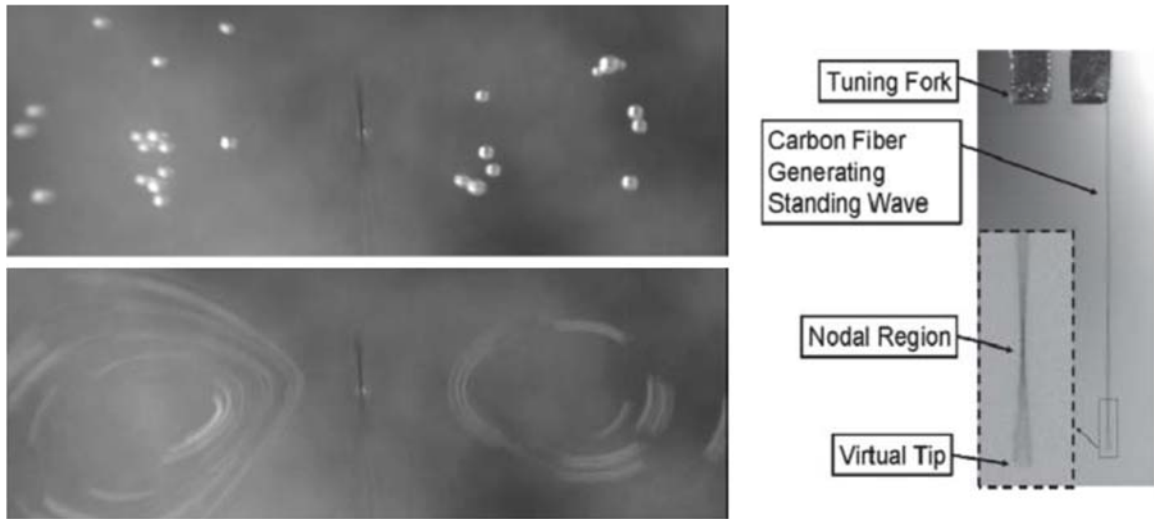


Figure 2-2. Sustained vortex generation with a single micro-scale oscillatory probe

Theoretical understanding of contact-free particle manipulation has been obtained from a detailed study of inertial particle transport and trapping in a viscous streaming flow generated by a single oscillating cylinder [38]. The particles in question might be cells from a biological sample, to be separated and tallied according to cell type for cancer diagnosis [39], for instance, or they might be abrasive particles to be circulated in proximity to a brittle surface for precision machining [6,11].

The left panel of Figure 2-3 depicts a four-step application of the controlled physics described above in a multi-probe experiment. An inertial particle is first captured by the probe on the left and then transferred to the next probe; this is followed by transport to two more probes further to the right.

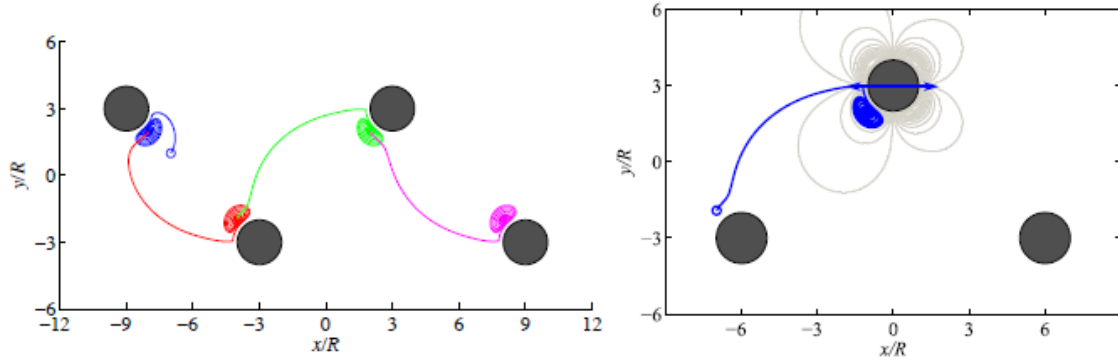


Figure 2-3. (Left) Trajectory of an inertial particle of size $a=R = 0.175$ and density $\rho_p = \rho_f = 1$ transported between four cylinders via sequential cylinder oscillations. Motion of each cylinder at $Re = 40$. Transit times between cylinders are on the order of 105 oscillation cycles. (Right) Cylinder motion (indicated by arrow), associated mean streamlines (gray), and particle trajectory during one transit (in blue) [40].

Figure 2-4 depicts streamlines within systems of probes obtained in the simplified manner just described.

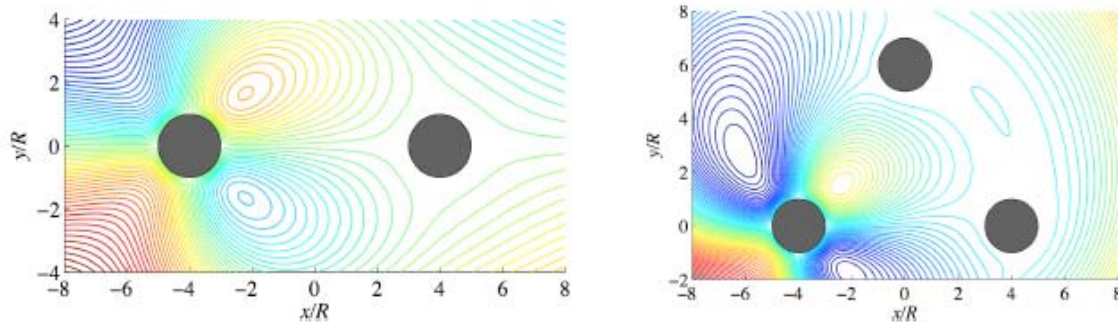


Figure 2-4. Time-averaged streamlines within systems of probes modeled in a simplified way that builds upon the analysis in [41].

The preliminary results provided in [42,43] and this paper promises to lay the groundwork for an inexpensive, scalable technology for the two and three-dimensional contact-free manipulation of particle systems in fluids and to illuminate a fundamental biophysical mechanism pervading the natural world.

2.2 Encoded PIV/PTV

Particle image velocimetry and particle tracking velocimetry have advanced rapidly providing the ability to measure gas and liquid flow fields with low and high velocity ranges [44,45]. Micro PIV (μ PIV) is a useful tool for detailed characterization of micro-fluids for the application of near-wall flow, electro-kinetic flow, biological flow, mixing and turbulence transition [46]. There is now an extensive literature demonstrating a broad range of applications of PIV covering, for example, applications in aerodynamics [47,48], detecting sound sources and distributions [49], biological cell manipulation [15] and many others too numerous to list [44,50]. Typically, a camera system will open a shutter for a fixed time period (the exposure time) during which the flow is illuminated by a bright source. Illuminating a flow using pulsed lasers or LEDs provides illumination method for locating particles in a flow during the time of the pulse. Kreizer and et al. developed a FPGA-based real time image processing system for PTV which makes it suitable for remote PTV or industrial applications [51]. In 2D PIV, The optics generates a light-sheet with adjustable thickness, enabling the user to produce light-sheets for different flow illumination application. The optics in this application consist of a spherical lens and cylindrical lens combination. The cylindrical lens enlarges the laser into a plane, and then the plane gets compressed into a thin sheet by the spherical lens. A 3 dimensional particle tracking system was developed by determining the three dimensional location of particles using three separate CCDs placed at the vertices of an equilateral triangle. Each camera view the same test volume from different angles, the final image is formed by overlaying the CCD images with correct offsets. Particle tracking is done using a hybrid technique. Once all of the three-dimensional positions of the particle are calculated, the three components of the velocity of the particles is determined by finding corresponding particles

in two successive frames separated by a specific time interval.[52]. Li and et al. presented a one-computer-six-camera unit for large-scale particle tracking velocimetry (LSPTV) for measuring surface velocity in vast unsteady flows [53]. Kitzhofer et al. developed a new tomographic reconstruction technique based on telecentric imaging with the epipolar geometry for 3D PTV [54]. A multi-camera set-up is used (in this case 3 cameras) to record particles in the object space. A mathematical model for a stereoscopic camera set-up given by the epipolar geometry describes that a point found in the image of one camera, has to lie on a specific line in the image plane of the second camera. The main feature of tomographic reconstruction in this research is the reconstruction of ghost particles and ghost trajectories.

PIV is a cross correlation based technique used to identify average flows from frame to frame and also regions within a frame. PTV, on the other hand seeks to identify individual particles and their trajectories over a large number of frames. Typically, PTV is used when lower density of particles are involved due to the difficulty of discriminating particles from each other. Both PIV and PTV utilize correlation and statistical analysis techniques to optimize extraction of information. PTV that seeks to identify individual particle trajectories has a higher resolution compared to PIV techniques. In this paper, it is proposed that pulse encoding be used to increase the likelihood that a measured trajectory belongs to a single particle. This will be particularly important when trying to measure particle trajectory in more densely populated particle fields. This preliminary study has evaluated 2 dimensional flow fields. For the purpose of evaluating this technique, only 2 dimensional flows are considered in this paper. In principle this can be, with some effort extended to 3 dimensional flows.

Technical restrictions impact the temporal and spatial resolution of image velocimetry measurements. The temporal resolution is mainly effected by the camera frame rate (fps), while the resolution of the camera sensor (pixel number) and the tracer size determines the spatial resolution in these type of measurements. Additionally, the selection of a proper tracer particle requires careful consideration as they ought to move together with the ambient flow.

Various pulse techniques used in PIV/PTV are illustrated in Figure 2-5, where the image of two consecutive frames are shown for each implemented method. Most simply the particle velocity can be measured by illuminating the image for a fixed period at which point the particle will show up as a streak. For very short pulses, it is possible to record consecutive images and at a fixed frame rate and super-pose images to generate particle-pairs. Equivalently, the illumination source can be pulsed multiple times during an exposure to create an image revealing motion of particles during the frame exposure [55,56]. In reviews of this measurement process, the concept of coded timing of light pulses is mentioned as a method for identifying the direction of the fluid motion [57,58]. Similarly, a multi-pulse and multi frame techniques have been used by [59] to enhance the dynamic range of PIV and velocity measurement. However, pulse light encoding implementation for individual particle identification and tracking does not appear to have been explored in the literature.

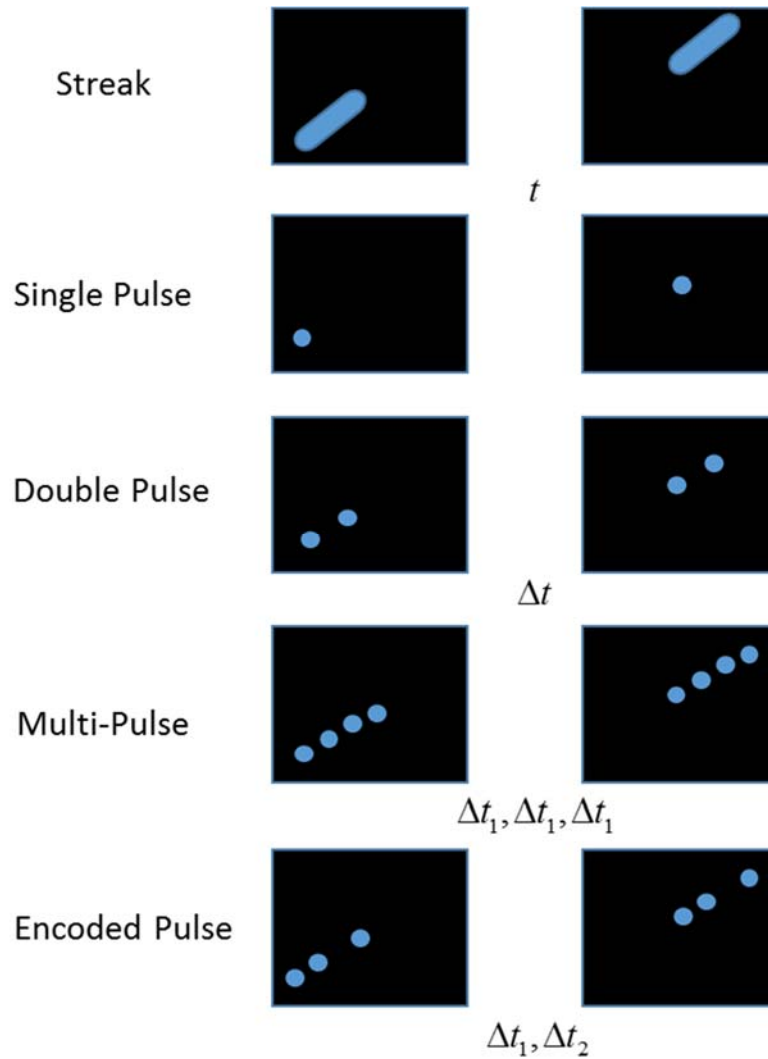


Figure 2-5. Pulse coding techniques for particle Image/tracking velocimetry.

For experiments involving a large density of moving particles and for complicated flow patterns, the proximal location of other particles makes it increasingly difficult to ensure that particle-traces identified by the software belong to the same particle. The technique outlined in this paper addresses this problem. In this article, the concept of encoded-PTV and algorithms developed for evaluating this technique are described. A mathematical model has been developed to determine the probability of finding a wrong particle within a flow comprising varying numbers of particles (i.e. particle densities).

Results of these simulations provide a statistical measure of the ability to extract individual particle trajectories and reject false particle readings as a function of the pulse timing patterns and number of encoder pulses. Finally, experimental double and triple pulse encoding for measurement of particles in the presence of a small vortex flow is presented, followed by statistical analysis for demonstrating the consistency of the theoretical model with the simulated data.

2.3 Dynamic absorber-based micro-scale oscillatory probes

Tuning fork and other quartz-based oscillators are commonly used as timers in clocks, AFM probes [60–63], touch sensitive sensors, micro-robotics fingers [64–66], and micro balances [67]. Micro-scale oscillatory probes made by attaching carbon, glass or tungsten fibers to a quartz tuning fork tine provide a capability to both oscillate the fiber and monitor changes in its response when the fiber comes into proximity with external objects or is immersed in fluids. These probes have been used for a broad range of applications including for surface modification studies in vortex machining process [6,8,11], touch probes in coordinate measuring machines (CMM) [65], touch-sensitive micro-robotic fingers[64], and in high speed fluid-flow in microfluidics studies[68]. Another study plans to use these tuning fork-based micro probes for investigation on micro-particles dynamics around dynamically vibrating objects for non-contact manipulation of particles immersed in fluids, which can be used for assembly and sorting/shepherding particles [38,40,69,70].

To better control these processes, it is necessary to interpret the measured responses of the probes in terms of the fiber and tuning fork dynamics. Hence a mathematical model of oscillating fibers attached to a tuning fork tine is provided in this article. This will be

represented in two main sections, the theory behind the two serially connected beams model and the lumped absorber model, results from which are compared with experimental data and FEA.

CHAPTER 3: PARTICLE MANIPULATION EXPERIMENTAL FACILITY

This chapter contains detailed descriptions of the design and fabrication of the experimental set-up of particle manipulation apparatus, system control, experimental procedures, and preliminary results of particle velocity measurements. The results demonstrate that the developed opto-electro-mechanical system is able to track particle trajectories for studying dynamical behavior under controlled experimental conditions. Also an encoded pulsed LED technique for particle tracking velocimetry (PTV) is explored for tracking complex multiple particle flow fields, which will be discussed in full detail in chapter 4. This chapter will be presented in three main sections; design of instruments, experimental procedures, and results.

3.1 Design of instruments

3.1.1 Experimental apparatus

Figure 3-1 shows a schematic representation of the apparatus of this development. The fluid in a container (a petri dish in this illustration) is placed onto an adjustable platform to set the height of the oscillating rod relative to the bottom of the transparent container using a manual micrometer. Adjustment of the platform enables the height of the fluid container to be positioned so that the tip of the rod nearly touches the bottom of the fluid container. Imaging of the flow zone is achieved using a long range objective, high magnification camera (SONY XCD-SX910). For imaging the induced streaming flows, it

is important that the particles be near neutral density and sufficiently small to follow the flow of the fluid (particles between $10\text{ }\mu\text{m}$ to $150\text{ }\mu\text{m}$ diameter with density of around $1.01\text{ g}\cdot\text{cm}^{-3}$ are used for most of the experiments and are further specified in the results section). Because of the small depth of field of the imaging system, two lasers attached to lockable Heim Joint mounts have been set to coincide at the focal plane of the lens to provide visual feedback during camera adjustment.

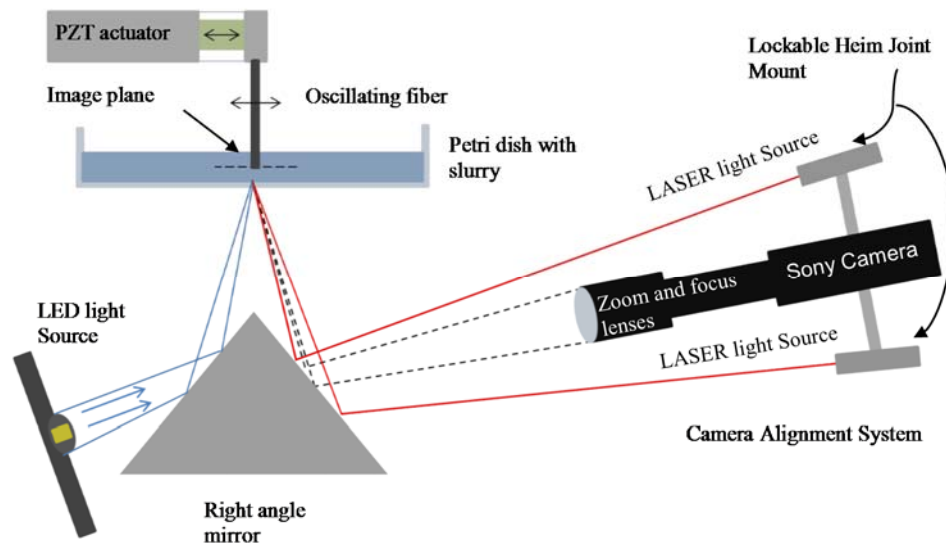


Figure 3-1. Current particle manipulation experimental apparatus.

Figure 3-2 shows a photograph of the experimental facility. The three legged bridge-type frame is constructed from aluminum and used as a rigid mount for the oscillator drive mechanism (described in section 3.1.2). Experiments require the oscillating rod to be submerged in a solution with depth control. This is accomplished with an adjustable height platform using a manual micrometer. An LED and right angle mirror are attached to an adjustable leg which enables alignment of the image plane on the mirror as well as directing the light from the LED to the flow region, see section 3.1.3. The camera positioning stage,

the fluid adjustment platform and the LED and mirror height adjustment move independently relative to each other to provide maximum alignment flexibility.

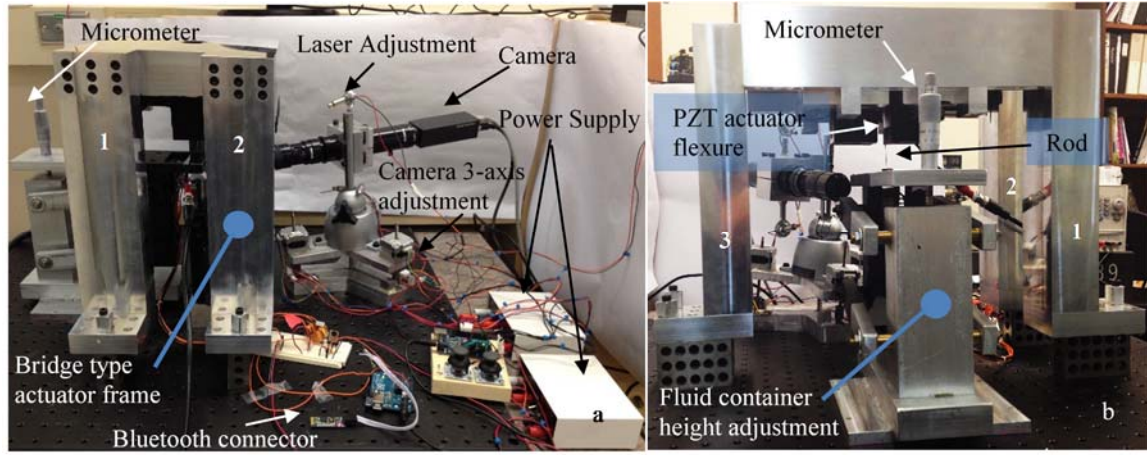


Figure 3-2. Photograph of Particle manipulation experimental apparatus with three legs of the bridge-type frame numbered. a) Side view, b) Back view.

The control flowchart of the particle manipulation system is shown in Figure 3-3. Both triggering of the camera and collection of image frames is controlled by a real-time system using LabView programming. To generate the drive signal at a defined frequency, a DDS-based SAJI Waveform Generator (STQ2) [71] programmed with a microprocessor through Serial Peripheral Interface (SPI) is used. Using a 10 MHz clock this provides a frequency resolution of 0.1 Hz for frequencies ranging from 1 milliHertz up to 2 MHz. The digital signal (trigger-signal) is generated with a microprocessor controlled by the Host computer via Bluetooth/serial communication. This trigger signal transmits to the LED Driver Circuit for LED pulsing and to the camera trigger port through a BNC connector. Similarly, this signal goes to the LabView imaging program in the Host computer via a DAQ Board and Real Time system to synchronize the camera shutter time with camera trigger and LED pulses. A monitor is connected to the Real Time computer for real time display of captured images.

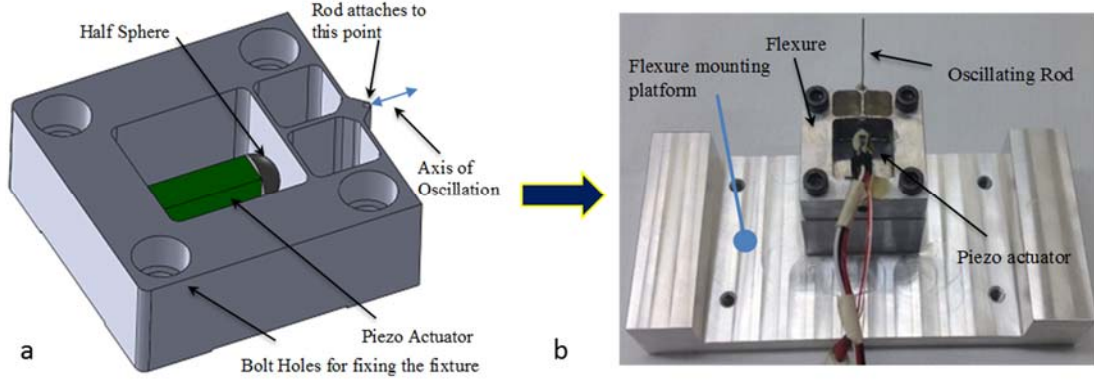


Figure 3-4. a) Flexure model and b) Oscillator drive mechanism showing the oscillating rod that has its axis perpendicular to the motion of the piezo electrically actuated flexure stage.

A high bandwidth class A amplifier (Thorlabs model MDT690, 0-150 V) is used to drive the piezoelectric actuator at frequencies up to 5 kHz. In most experiments, the amplitude of oscillation at the fiber tip is further increased by operating near to a resonant frequency of the rod to obtain efficient energy transfer. The n^{th} natural frequency (f_n) can be obtained using equation 1:

$$f_n = \alpha_n^2 \frac{1}{2\pi} \sqrt{\frac{EI}{ML^3}}, M = \rho\pi r^2 L, I = \frac{\pi r^4}{4} \quad (1)$$

Where E is the Young's modulus, L is the length of the rod and α_n is a constant equal to 1.875 for the first mode. M , the mass of the rod, and I , the second moment of area, ρ is the density of the rod, and r is its radius. A molybdenum wire ($E = 329$ GPa, $\rho = 10188 \text{ kg.m}^{-3}$) of diameter 0.5 mm ($r = 0.00025$ m) and $L = 38$ mm has been used as the oscillating rod. Using the equation 1, the first mode frequency for this fiber is calculated as 275 Hz. The flexure and mounting platform are made out of aluminum. Information

regarding the signal type and frequency are transmitted from WaveGenerate library to a microprocessor.

3.1.3 Illumination and imaging of the flow fluid

Illumination and imaging of fluid flow zone plays an important role in recording the high quality movies with the camera. For the experiments reported in this paper, illumination uses combinations of 40W RGBW LED high brightness light sources (LED Engin, model: LZC-B3MD07). Generally, it is desired to image the illuminated particles with light scattered from a high intensity incident beam. Any other light getting back to the camera will obscure the particles thereby reducing the measurement performance.

3.1.3.1 Optical setup

A right-angle prism with mirror-coated surfaces and the LED are attached to a height adjustment stage as shown in Figure 3-5. The LED shines on one side of the prism mirror and reflects towards the fluid flow zone. The camera is located on the opposite side of the prism which captures the images while viewing on the mirror. With this method, only the light scattered back from the flow zone is captured. On the other hand, to further decrease the stray light, the entire setup surrounding the flow zone is painted a matte (chalkboard) black. As shown in Figure 3-5, the LED is attached on an inclined plate facing towards the mirror. To direct as much light as possible to the flow zone, the optimum angle was calculated for the LED which is shown as α in this figure. The adjustable stage leg is utilized because the distance, h , between the rod and prism is critical for the image of the rod to appear on the center of the mirror face. By placing the reflection of the rod on the center, the flow zone area captured by the camera can be maximized through orientation

of the camera with respect to the prism. The optimal angle, α , for LED illumination can be calculated using the parameters given by

$$\begin{aligned} a &= \frac{\pi}{4} + b \\ b &= \tan^{-1}\left(\frac{4h}{x}\right) - \frac{\pi}{4} \\ c &= \frac{\pi}{2} - b \end{aligned} \quad (2)$$

By substituting the dimensions of the setup with these parameters, angle α is approximately 70 degrees.

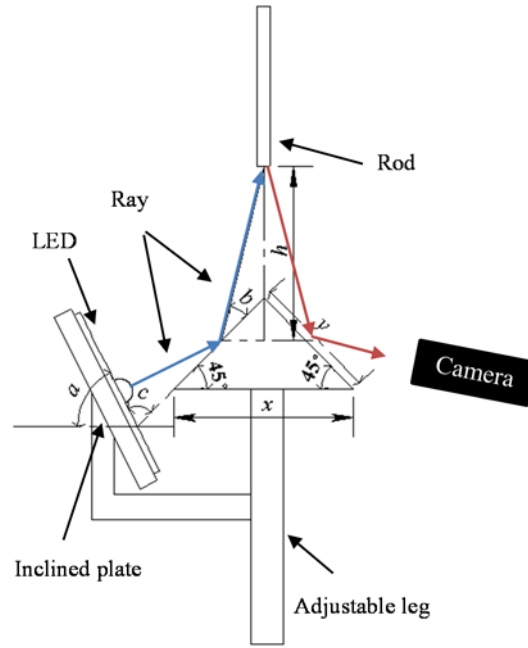


Figure 3-5. Diagram of the illumination system. The blue line indicates LED illumination and red is the reflected image to the camera.

Figure 3-6 show photographs of the experimental setup used for implementing the LED pulsing technique. A fluid reservoir for these experiments has been created by gluing a black rubber cylinder onto the upper surface of a glass microscope slide. Light scattered by particles and other objects in the field of view is then reflected off of the other side of

the right angle mirror, as shown in Figure 3-6(a), after which it is collected and imaged by the camera system.

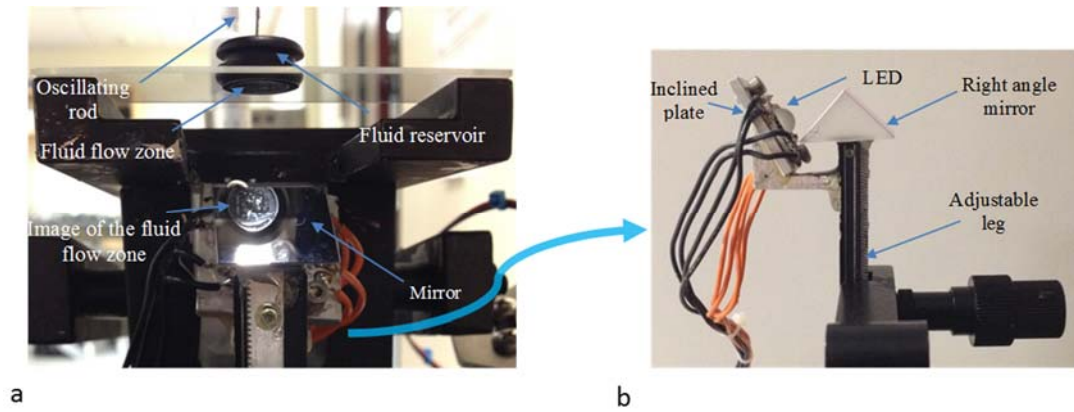


Figure 3-6. a) An image of the imaging system implemented for the PTV studies, b) Optical setup.

3.1.3.2 LED pulsing for encoded-PTV

To obtain images for identifying individual particles in dense particle image velocimetry studies, the use of multiple pulse illumination of an LED source has been implemented and tested. In this technique triggering of each image frame is synchronized with a pulse LED circuit that can be used to produce multiple, timed pulses within the shutter time of each frame. Timing of the pulses is controlled by a microcomputer with micro-second resolution. By providing more than one illumination pulse per frame, a moving particle will be imaged at multiple locations, the separation of which indicates the vector velocity in two dimensions, or, more precisely, within the focal plane of the image. More detailed study about encoded-PTV technique and the effect of various parameters in its performance is presented in [72].

A boost converter is used for the LED driver circuit. This can be driven by a TTL signal from the micro-controller to provide a current of up to 1.5 A for a short time-period.

The LED on/off time, the delay between them and the frequency per camera exposure is being controlled by micro-controller. To evaluate the performance of the boost convertor, the intensity of the light is measured by an optical sensor. The circuit for measuring the light intensity is shown in Figure 3-7 below. An FET operational amplifier (LT1252) is used for amplification.

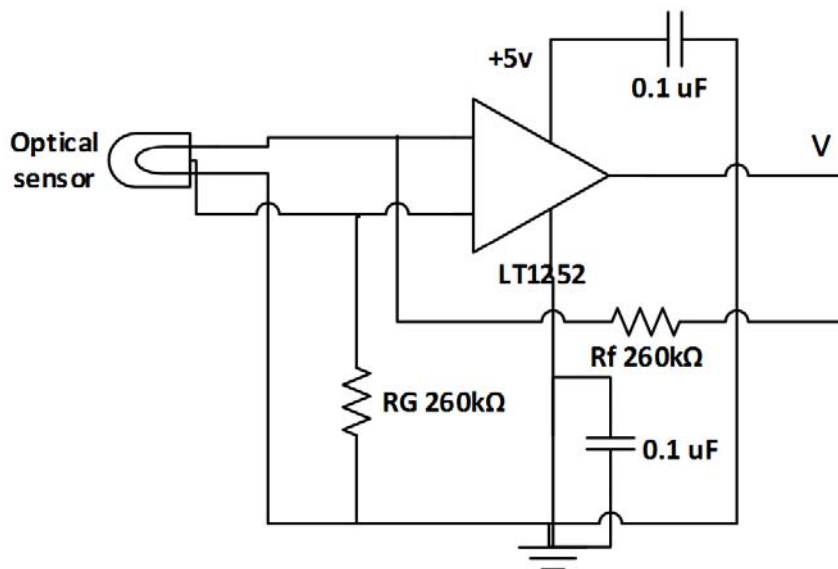


Figure 3-7. Circuit for measuring the intensity of light.

Figure 3-8 shows the sensor voltage (V_s) at 1 kHz with 50% duty cycle respectively. In the charts shown below, the LED is on when the voltage goes down.

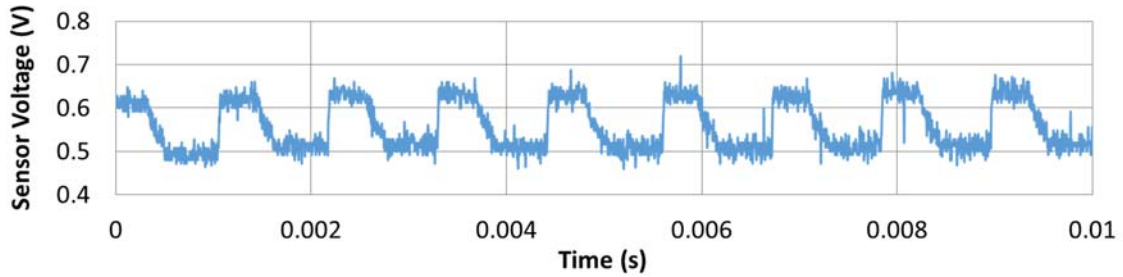


Figure 3-8. Sensor voltage (V), which is also shown in Figure 3-7, versus time for LED pulsing of 1 KHz, and 50% duty cycle.

Figure 3-9 illustrates of its important to mention that LED and camera need to be synchronized. To do this, the signal generated in microcontroller program is connected to the camera trigger port. In this example, the LED blinks twice for a camera exposure, for multiple pulse the LED blinks multiple times for one exposure.

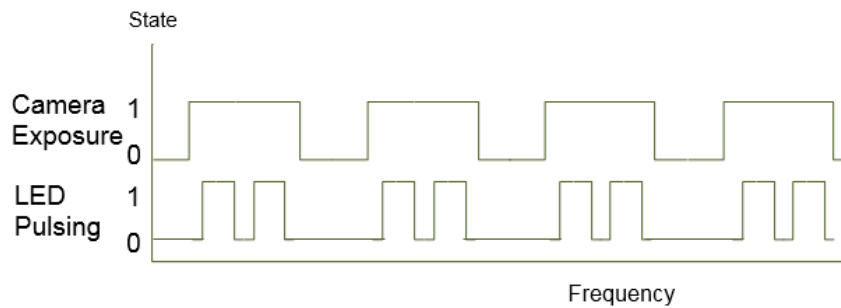


Figure 3-9. Double pulsing for each camera exposure results in particle pairing

3.1.4 Camera positioning

To facilitate the positioning of the camera to view at the fluid flow zone, an automatic 3-axis motion control apparatus has been designed and fabricated. Figure 3-10 shows the schematic of the apparatus for positioning the camera which is also controlled by microprocessor. Three stepper motors with intrinsic leadscrews are attached to a platform with 120 degree pitch about its center. The combination of leadscrew translations provides control of pitch, roll, and height adjustments. A universal vise (Panavise™) is

also attached to the top of this platform and is used for initial coarse alignment of the camera about the same three axes. Two joystick controls are linked to the microprocessor to control the stepper motors. With the analog joysticks, the variable speed and direction control is achieved.

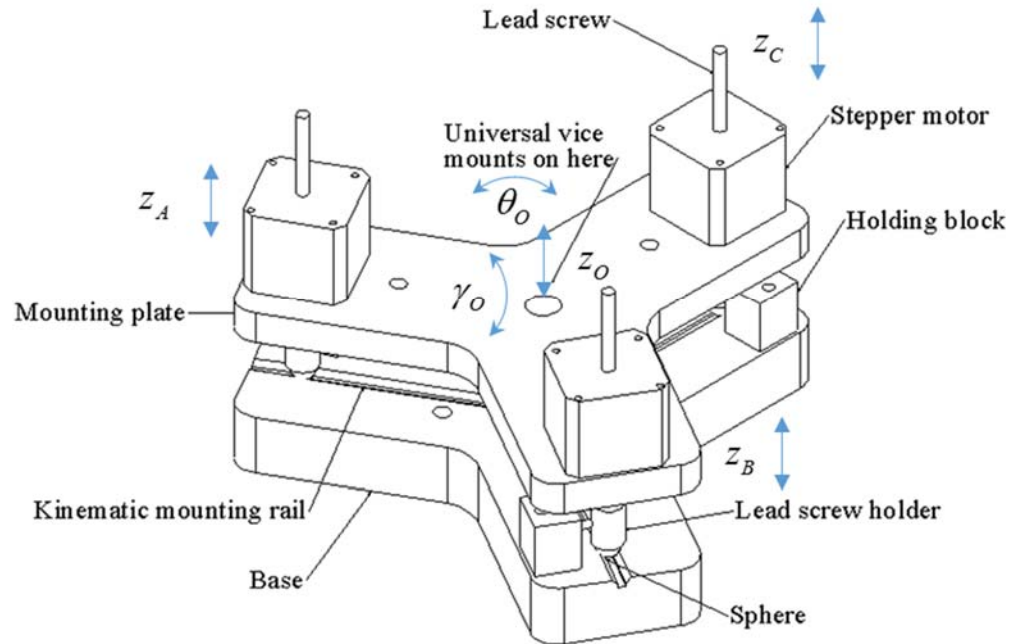


Figure 3-10. Automatic 3-axes motion control apparatus assembled

Figure 3-11 illustrates a top view of the base plate which can be used to geometrically calculate the tilt and axial motions of the center point O. The stepper motors attached to the points A, B and C enables z_A , z_B and z_C respectively.

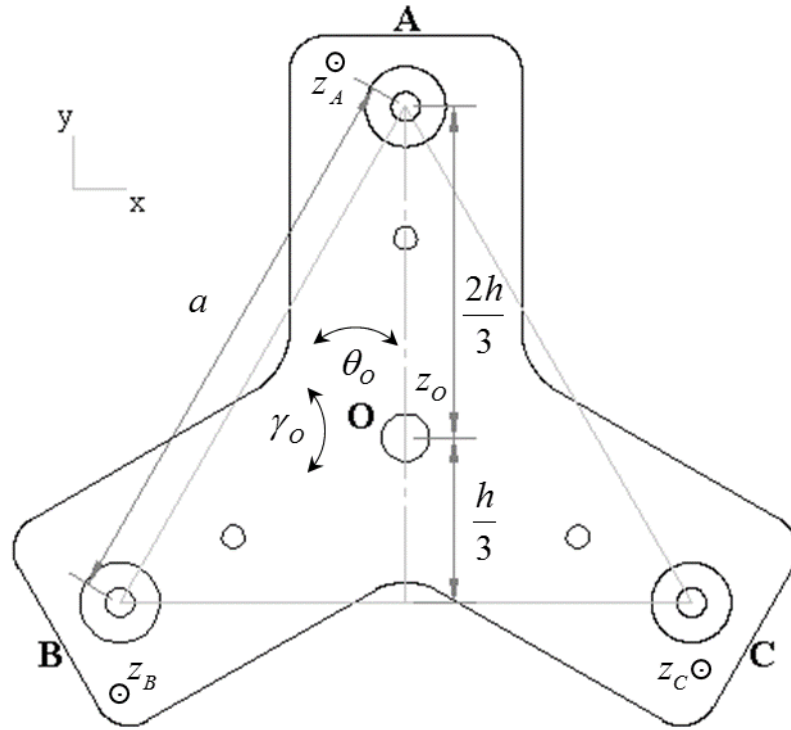


Figure 3-11. Top view of base plate with three circles A, B, and C indicating the location of the three actuators of this system. Linear motion at these points enables geometrical calculation of the tilt and axial motions of the center point O, motion is perpendicular to the page.

A top view of the mounting plate shown in Figure 3-11 can be used to calculate the geometrical relation of the tilt and axial motions of the system. In this figure, point O is the geometrical center of the equilateral triangle. A further device can be mounted onto the mounting plate and will typically be centered at this point. A, B and C are the points where motor driven feed-screws are located and can move up and down along the z axis which is perpendicular to x and y axes. z_A , z_B and z_C are the axial motion of motors at the points A, B and C respectively. θ_O is the tilt motion around the y axis, γ_O is the tilt motion around the x axis, and z_O the axial motion along the z axis of the center point O. For small angles,

the relationship between the translations of each actuator to motion of the mounting plate can be obtained from the equations:

$$\begin{aligned}
 h &= \frac{\sqrt{3}}{2} a \\
 \theta_O &= \frac{z_C - z_B}{a} \\
 \gamma_O &= \frac{\frac{z_A}{2h} - \frac{z_B}{h} - \frac{z_C}{h}}{\frac{3}{2h} - \frac{3}{h} - \frac{3}{h}} = \frac{3z_A}{2h} - \frac{3z_B}{h} - \frac{3z_C}{h} = \frac{3z_A}{\sqrt{3}a} - \frac{6z_B}{\sqrt{3}a} - \frac{6z_C}{\sqrt{3}a} \\
 z_O &= \frac{z_A}{3} + \frac{z_B}{3} + \frac{z_C}{3}
 \end{aligned} \tag{3}$$

Or, in matrix form

$$\begin{pmatrix} \theta_O \\ \gamma_O \\ z_O \end{pmatrix} = \begin{bmatrix} 0 & -\frac{1}{a} & \frac{1}{a} \\ \frac{3}{\sqrt{3}a} & \frac{-6}{\sqrt{3}a} & \frac{-6}{\sqrt{3}a} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{pmatrix} z_A \\ z_B \\ z_C \end{pmatrix} \tag{4}$$

An embodiment of this proposed system is comprised from four major components. Two key structural components are the base plate, Figure 3-12, and the mounting plate, Figure 3-13. These two plate are connected together via three motor driven feed screws.

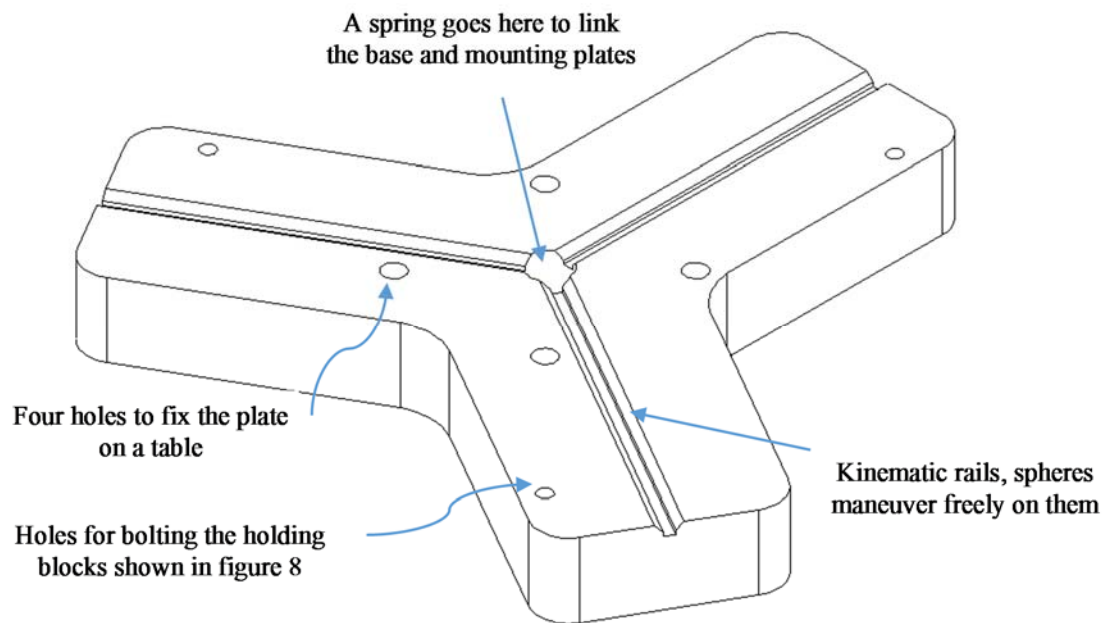


Figure 3-12. Base plate.

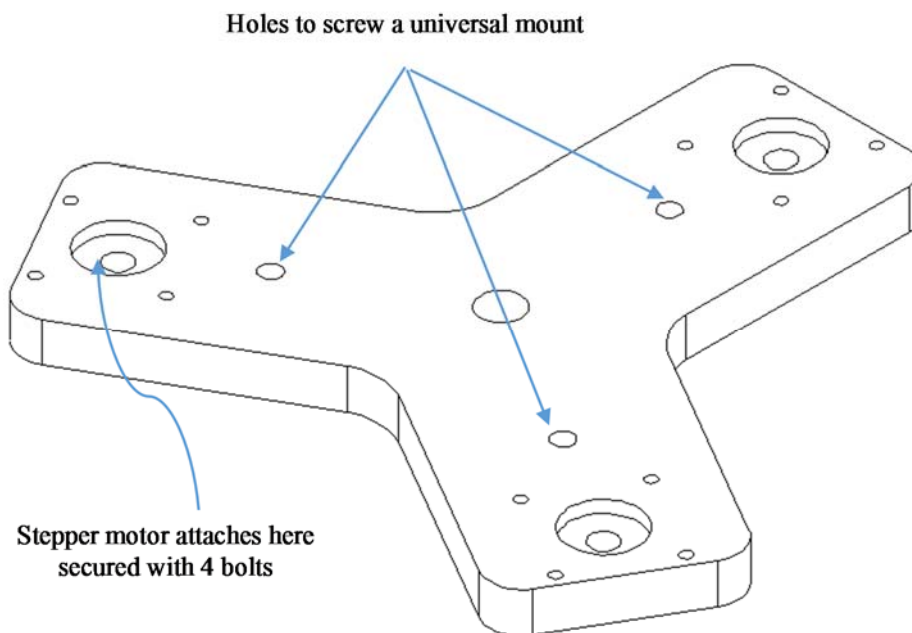


Figure 3-13. Mounting plate.

The design of this mechanism has to provide maximum flexibility and interchangeability for users convenience. For example, the stepper motors can simply be de-attached from the setup. Similarly, one can use any size or customized design of a universal vise to mount a bigger or smaller components on it. Furthermore, a finer feed-rate or a wider control of tilt motions can be obtained using a smaller pitch lead-screw or longer one respectively.

Lead-screw attaches to the lead-screw holder shown in Figure 3-14-a and gets tightened by a set screw. A dowel pin represented in Figure 3-14-b inserts to a radial hole shown on the left. A small sphere goes between this part and the kinematic rail of the base plate illustrated in Figure 3-12. This sphere should not be glued or firmly attached to the lead-screw holder by any means, otherwise, the mechanism will be over constrained. Holding block represented in Figure 3-15 is used to prevent the lead-screw from rotation driven by motors. Two dowel pins insert to the right hand holes. The distance between these two holes is slightly bigger than the diameter of the dowel pin attached to the lead-screw holder to compensate the possible backlash of the system. Holding block can be fixed to the base plate by screw and this can be done by inserting a screwdriver to the hole on the left hand of the Figure 3-15.

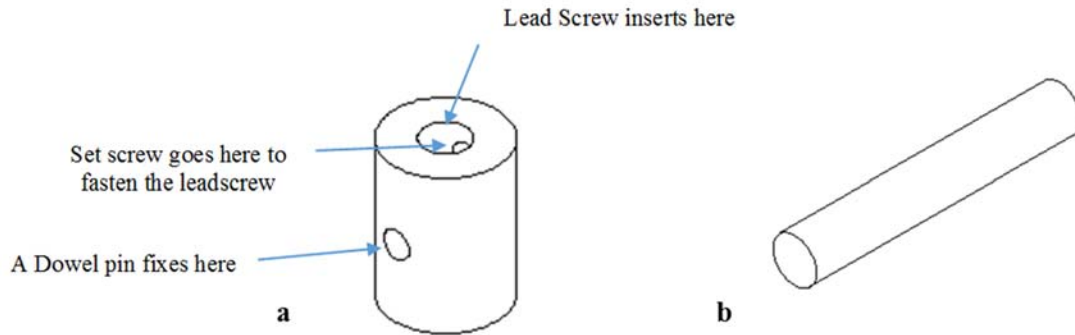


Figure 3-14. a) Lead screw holder. Dowel pin inserts into the radial hole shown on the left. b) Dowel pin.

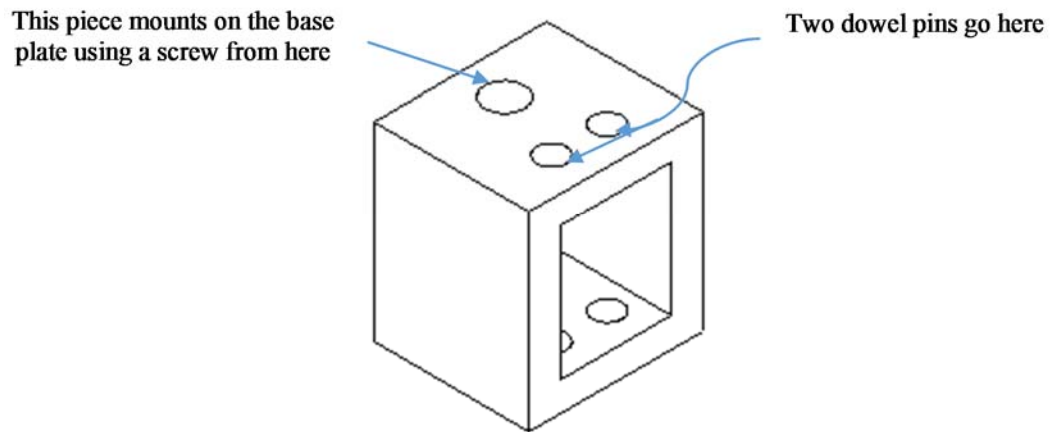


Figure 3-15. Holding block for preventing rotation of the actuator lead-screws.

The base plate can be bolted to a table by four $\frac{1}{4}$ ", 20 holes with 2" spacing indicated in Figure 3-12. A spring inserts to the hole in the middle of this plate to link it with the mounting plate illustrated in Figure 3-13. Three bolts is used to mount a universal vise or any custom-designed mechanism for holding the device on the mounting plate. Stepper motors are fixed to this plate using four screws. One can simply add blocks under the base plate or use/design an adjustable vise/mechanism placed on the mounting plate to get any desired height for the motion-controlled device for their own application.

Shown in the Figure 3-16 is the 3-axis motion control apparatus used for camera positioning in the particle manipulation experiments.

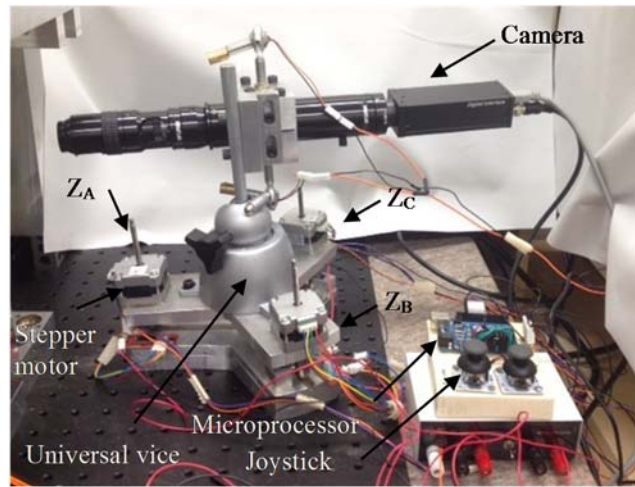


Figure 3-16. Camera positioning stage with microprocessor control.

3.1.5 SAJI waveform generator (STQ2)

The proposed frequency generator is an inexpensive, easy to use device that is capable of generating different types of electrical signals (sine, square and triangular waves). When multiple signals are needed and each of which requires independent computer-control, this can replace commercial signal generators in a more compact space and can be integrated with other circuits. The design shown here can generate up to 2 MHz waveforms with developed software that provides the ability to generate frequency sweeps and multi-channel signals.

The board shown in Figure 3-17 comes with WaveGenerate library that can be used with any type of micro-controller. This library provides four different example programs to use the proposed Waveform Generator for different applications. This includes single wave form generation, sweep, sweep in multiple channels, and Labview controlled signal generation.

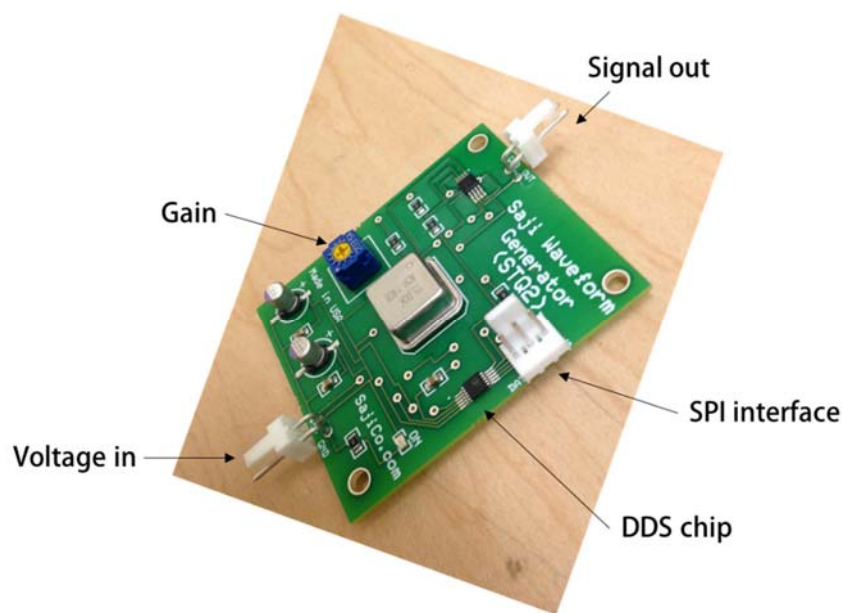


Figure 3-17. WaveforGenerate Board with description of its main features.

Figure 3-17 illustrates the function generator board with labels indicating its key functional features. Table 3-1 shows the pin-out for the board. CLK and DATA pin must be connected to the pin 10 and 11 in your microcontroller respectively. Any microcontroller can be used to drive the board.

Table 3-1. Pin function description and the typical pin numbers for connecting to Arduino or similar micro-processors

Mnemonic	Description
SS	Select slave pin. Connects to any pins (except 10 and 11) in an Arduino microcontroller. When 'low' data can be transferred to the board thereby enabling data transfer to as many boards as there are available pins on the micro-controller. The board will not respond if this is set 'high'.
CLK	Clock pin. Connects to pin 10 in microcontroller.
DATA	Data pin. Connects to pin 11 in microcontroller.
GND	Ground.
+5V	5 volt supply.
OUT	Output voltage. The output signal of the board is available at this pin. No external load is required.
GND	Output ground.
Gain	Controllable gain. By adjusting the potentiometer, output voltage amplitudes ranging from 0.65 up to 4.6 V can be set by the user.

Figure 3-18 illustrates the pin connections to an Arduino microcontroller. The output signal is available at OUT and GND pin on top left of the board. For different resolution and range, this clock can be replaced by one having a different frequency.

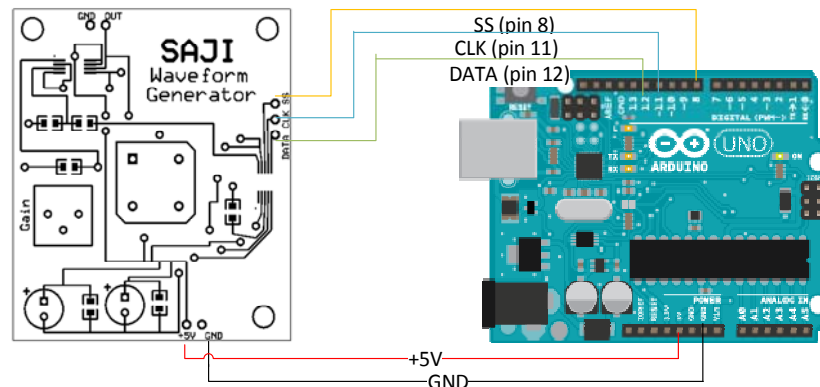


Figure 3-18. WaveGenerate pin connections to Arduino-UNO.

Figure 3-19 shows the maximum voltage (at maximum gain) achievable at different frequencies. The square wave generated is 5 V pk-pk, while the output voltage for sine and triangle wave can be amplified up to 4.6 V using the Gain potentiometer for output frequencies of up to 500 kHz.

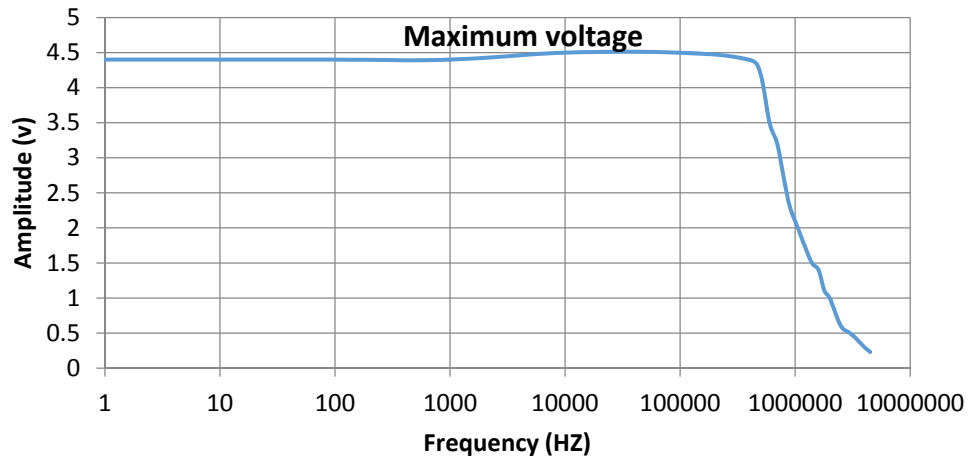


Figure 3-19. Maximum voltage generated at different frequencies.

Figure 3-20 shows an example of a library program illustrating how to generate a signal with a specific frequency, and frequency type by simply uploading this program to microcontroller. This shows a standard program in which the function library ‘wavegenerate.h’ has been included. With this included, a signal can be generated using the ‘waveGenerate’ function command with parameters transferred to the function explained in the comments at the top of the program. Figure 3-21 shows a GUI developed in Labview™ that can be used on any computer platform to also send controls to the waveform generator board. As an example, for frequency Sweep mode, one might change the Frequency Start, Frequency Stop and Step.

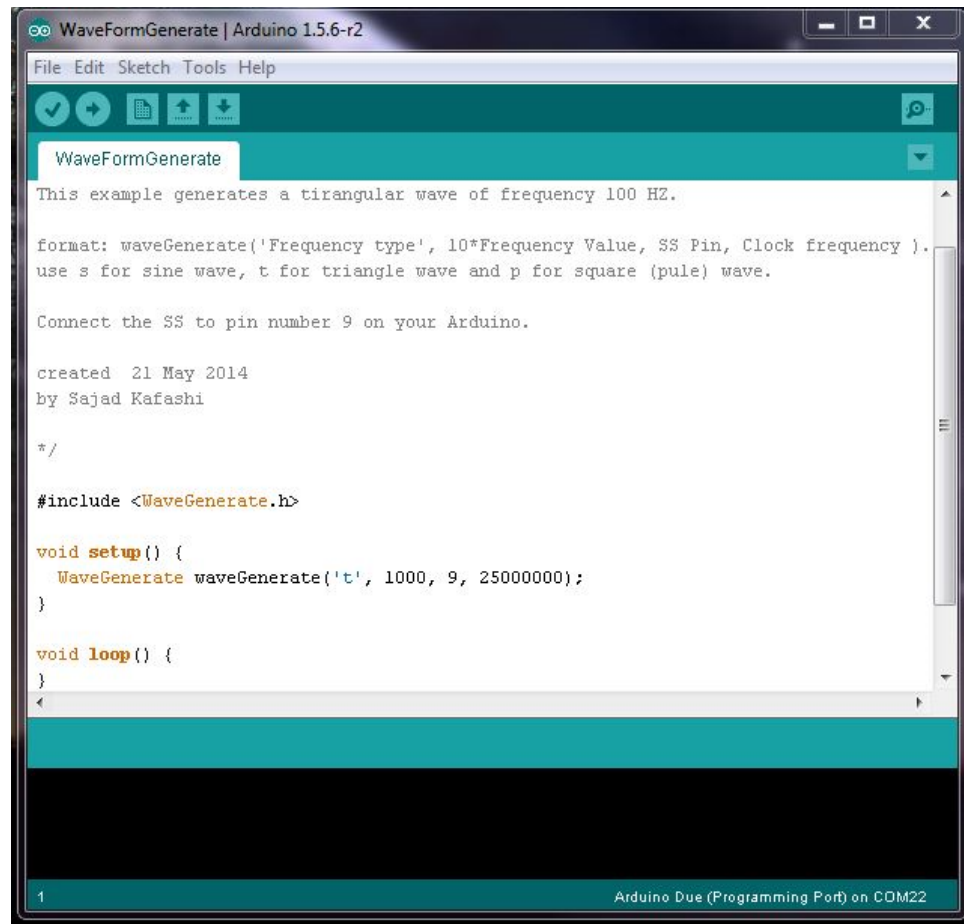


Figure 3-20. WaveformGenerate.ino for generating a simple signal using the board when connected to an Arduino™ micro-computer.

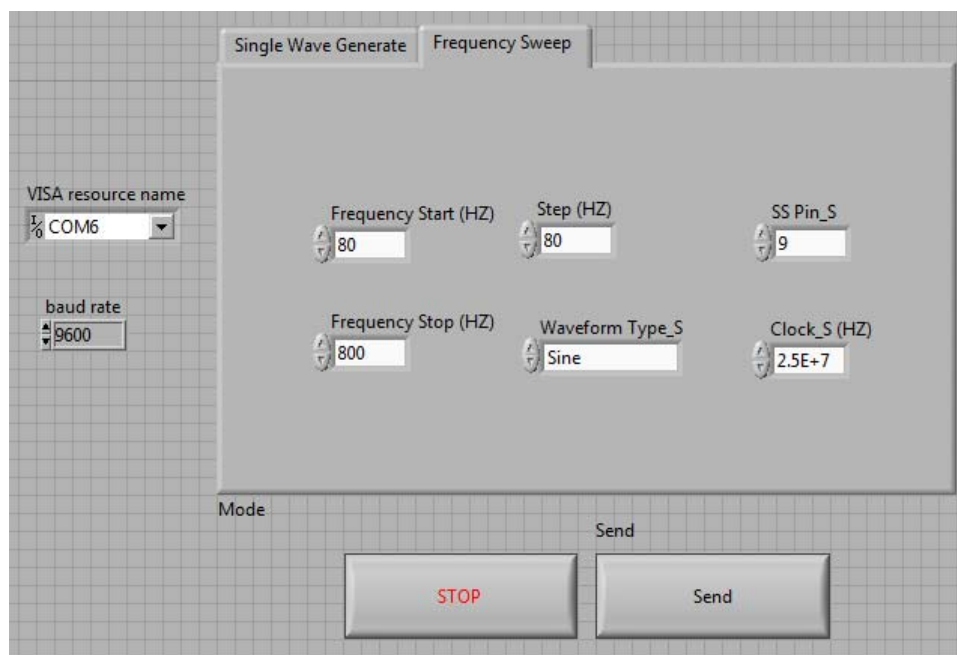


Figure 3-21. WaveGenerateLabviewControl.VI (Frequency Sweep Mode) that can be operated from any windows based operating system.

3.2 Experimental Procedures

3.2.1 Volume of particles

To obtain a specific concentration of particles in a given volume of fluid, a master spoon (Shown in Figure 3-22, a) is used to dispense a known volume of particles. A flat spoon of particles is obtained by filling up the bowl of this spoon and then using a flat knife edge to create a level surface up to the rim. The volume contained in this level spoon is, for the purposes of this paper called a grain (G), in other words 1 G indicates a full level spoon of particles. To measure the volume of the master spoon, modeling clay was used to fill to the spoon, again flat to the rim. Then, the clay was shaped to a rectangular cube similar to that shown in Figure 3-22(b). The dimensions of a , b and c were measured using a digital caliper as 1.80 mm, 1.12 mm and 1.87 mm respectively.

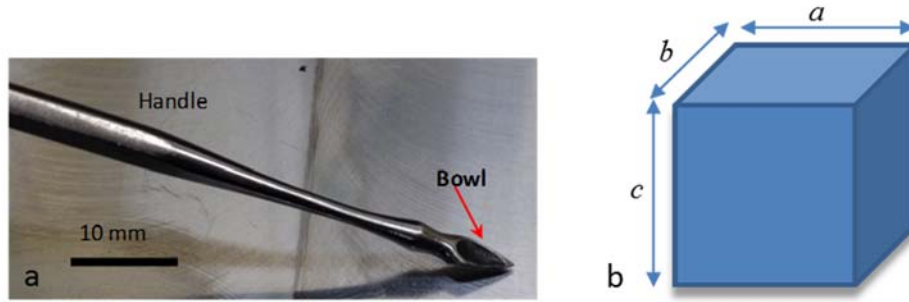


Figure 3-22. a) Image of the master spoon, and b) Geometry of rectangular cube

Therefore, the volume within a full flat spoon is:

$$V = abc = 3.77 \text{ mm}^3 \approx 0.0038 \text{ mL} \quad (5)$$

These spoons are relatively inexpensive and manufacturers do not supply any specifications on geometry. Hence, the spoon measured in this exercise is marked and used as a reference for creating particle concentrations throughout all experiments.

3.2.2 Number of particles

To obtain a more standard criteria to specify particle densities, the number of particles in a given volume of fluid is used. To calculate the number of particles in 1 G, it is assumed that they have a close packed structure (i.e. F.C.C). The atomic packing factor for a F.C.C structure is approximately 74%:

$$APF = \frac{V_a}{V_c} = 74\% \quad (6)$$

Where V_a is volume of atoms and V_c is the volume of unit cell. Therefore, the number of particles N can be calculated from the following equation:

$$N = \frac{V_{spoon} \times APF}{V_{sphere}} = \frac{3.8 \times 0.74}{\frac{4\pi}{3} r^3} \quad (7)$$

In which V_{spoon} is the volume of spoon and V_{sphere} is the volume of a sphere, and r is the radius of spheres measured in millimeters. For particles with average diameter of 137 μm , the number of particles in 1 G is calculated at 2072.

3.2.3 Solution concentration

Density of particles is defined as $\frac{x/20 \text{ G}}{20\text{mL}}$, e.g. to obtain a unit concentration, one flat spoon of particles needed to be mixed into 400 mL of water. Biocompatible surfactant (Cospheric Tween20) is used to prevent agglomeration. A solution containing 0.1% of tween20 mixed with distilled water and heated to 80 °C is used (i.e. 0.4 mL of tween20 in 400mL of distilled water). One flat spoon of spheres are added to the solution soaking them at room temperature. Particles in these solutions are spheres with 125-150 micrometer diameter. In this step we have $\frac{1}{400} \text{ G}\cdot\text{mL}^{-1}$ concentration. To create concentrations of higher density, 100 mL of this solution is emptied in a reservoir and 1 G is added to the remaining 300 mL solution, which provides $\frac{2 \frac{1}{3}}{400} \text{ G}\cdot\text{mL}^{-1}$ of concentration. These steps repeated several times providing different concentrations such as those represented in Table 3-2.

Table 3-2 . Achieving solutions with different concentration in multiple steps

	Empty 100 mL	Add 1 G
Step 1 (concentrations)	$\frac{1}{400} \text{ G}\cdot\text{mL}^{-1}$	$\frac{2 \frac{1}{3}}{400} \text{ G}\cdot\text{mL}^{-1}$
Step 2	$\frac{2 \frac{1}{3}}{400} \text{ G}\cdot\text{mL}^{-1}$	$\frac{26}{6} \text{ G}\cdot\text{mL}^{-1}$
Step 3	$\frac{26}{6} \text{ G}\cdot\text{mL}^{-1}$	$\frac{19}{3} \text{ G}\cdot\text{mL}^{-1}$

Solutions with various slurry concentration are shown in Figure 3-23.

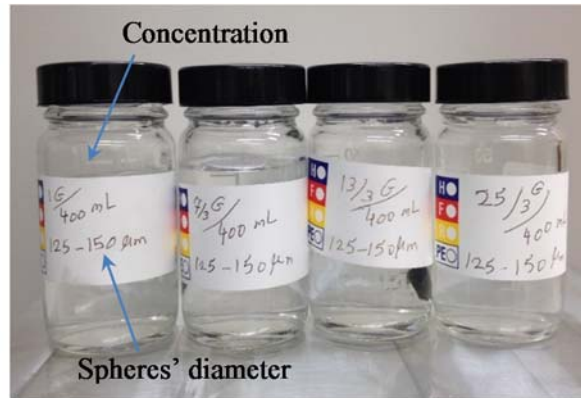


Figure 3-23. Photograph showing four bottles of slurry with concentrations of particles ranging from $1/400 \text{ G}\cdot\text{mL}^{-1}$ to $25 \frac{1}{3}/400 \text{ G}\cdot\text{mL}^{-1}$ (see definition in text) used in experiments.

3.3 Results

Various experiments have been carried out with $\frac{25}{3} \text{ G}\cdot\text{mL}^{-1}$ concentration solution

with Grey Polyethylene Microspheres particles with $1.00 \text{ g}\cdot\text{cc}^{-1}$ and $125\text{-}150 \mu\text{m}$ diameter (Cospheric). Images taken from experiments are usually filled with a lot of noises and redundant objects that makes it difficult to obtain the useful information out of them. In

order to facilitate the analysis of the experiments, several images processing techniques are developed which are explained in the following section.

3.3.1 Digital image filtering

3.3.1.1 Filtering by thresholding

Figure 3-26-a shows an image from the experiment. As it is labeled in the image, it contains noises due to redundant light and stationary particles. The small bright spots are settled particles on the bottom of the slide. Images are filtered to remove the redundant and stationary objects. Being stationary, they are exposed with an intensity proportional to the number of pulses per frame. As such these can be readily identified as having a value above the threshold for moving particles. Once identified, these are replaced by the intensity of neighborhood pixels. The algorithm of filtering implemented in Matlab™ is shown in Figure 3-24.

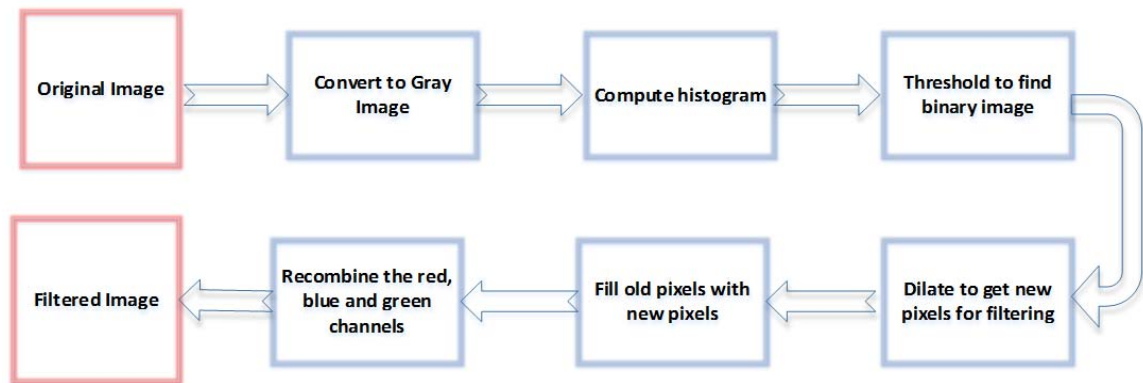


Figure 3-24. Filtering by thresholding algorithm.

Figure 3-25, illustrates the filtering sequence and the result for one sample picture.

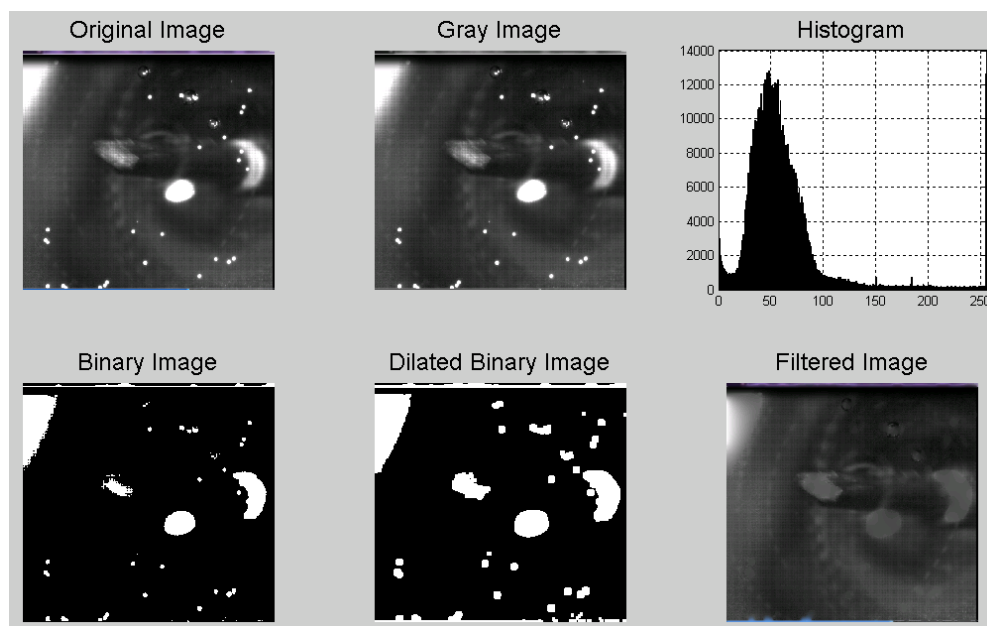


Figure 3-25. Filtering by thresholding sequence.

Figure 3-26 shows an example of an original image versus the filtered images after the filtering algorithm being implemented.

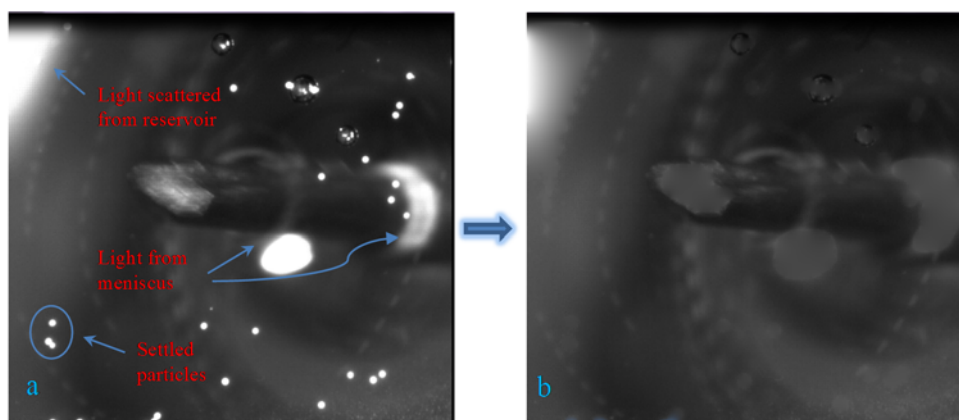


Figure 3-26. a) Original image, b) Filtered image.

3.3.1.2 Filtering by subtracting two images

A more robust method of filtering noise and removing stationary features in the images is obtained by subtracting the images from each other. In this procedure, all the

stationary objects that are common in the two images are removed from the output images, leaving only moving objects. This way, it is clear which object has been moved. This concept is illustrated in Figure 3-27 with various examples. For instance, in Figure 3-27-e, all the stationary pixels between Image1 and Image2, i.e. everything that is common among these two images are shown as black. However, it only includes the pixels from Image2 that does not appear in Image1. This are the particles that have been moved in the experiment in the time interval that Image1 and Image2 has been taken.

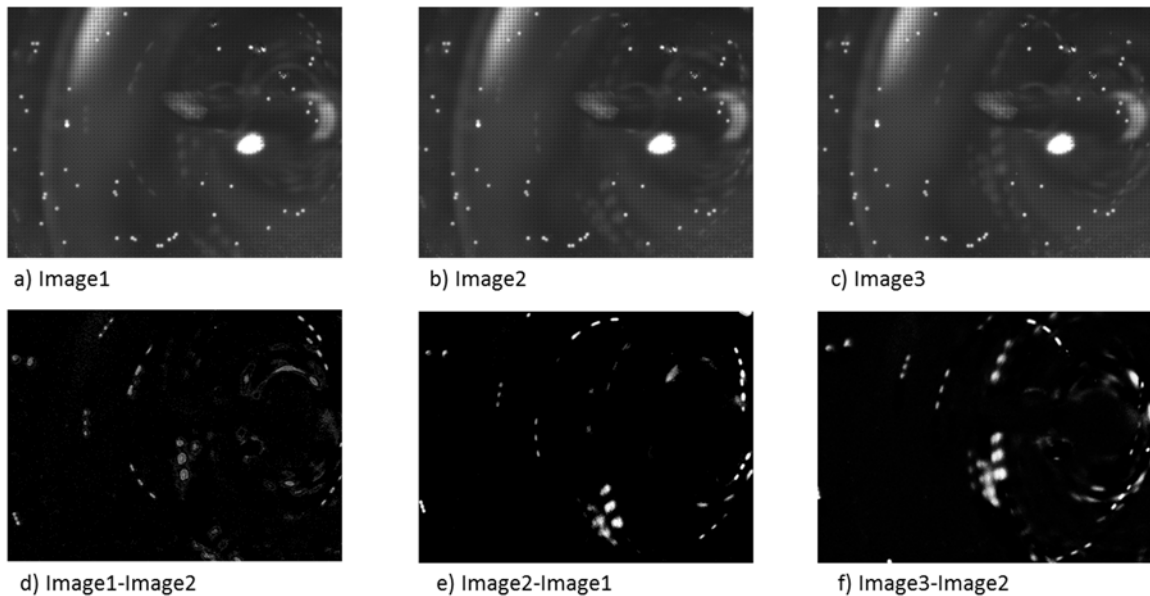


Figure 3-27. Filtering by subtracting two images. a, b and c show different images. d, e and f show filtering the noise and stationary objects by subtracting images from each other.

It is also obvious in the images that there are several complicated flow patterns in the experiments due to patterns crossing each other making it very difficult to pick the trajectories and velocity vectors using the commercial PIV/PTV softwares.

3.3.2 Trajectory tracking

Images of recorded videos with different LED pulsing patterns are shown in Figure 3-28. The first of these, Figure 3-28-a, shows an image from double LED pulsing, meaning that LED goes on twice per camera exposure. As expected this results in particle pairs. Some particles pairs are indicated with ellipses in the image. For turning LED on and off three times during an exposure, particles show up in the image at three different positions. Blinking the LED 10 times for one exposure shows the paths that particles travel during that time, Figure 3-28-c, and 20 LED pulses per exposure is demonstrated in Figure 3-28-d. This almost shows the complete particle trajectories spanning the whole image.

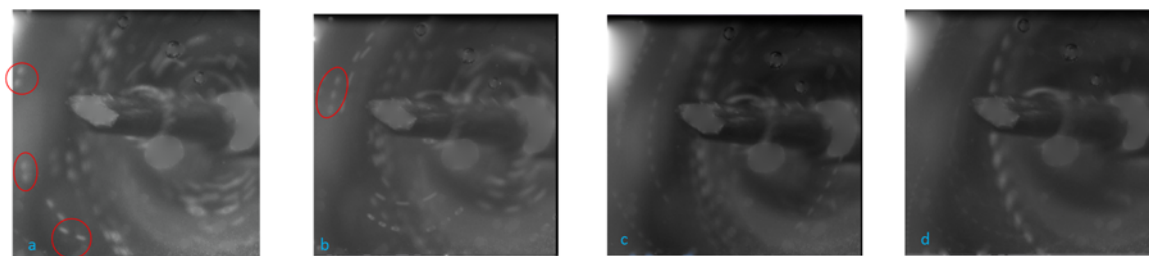


Figure 3-28. Flow fields image with 270 Hz rod frequency, 10.6 mm reservoir inner diameter, 30 ms LEDon, 30 ms LEDoff, 30ms camera off, 120-150 μm diameter spheres, and 25/3/400GmL⁻¹ concentration. a) 2 blinks, b) 3 blinks, c) 10 blinks, d) 20 blinks

One of the main advantages of this technique is its flexibility. Multiple ePTV parameters like the LED on and off time, distance between two subsequent particles in the experiments can be controlled and adjusted easily with the light frequency. Also, from a literature review in PIV/PTV pulsing techniques, it is likely that the coding technique (multi pulsing with different timing) in multiple frames has not been implemented so far. It is expected that this technique can be helpful in tracking the same particle in its trajectory avoiding from being mixed with other particles.

3.3.3 Velocity measurement

Figure 3-29-a illustrates a particle in two different positions in one image frame. The velocity of this particle is the distance traveled divided by the time that takes for the particle to move from position#1 to position#2. The distance that particle travels in subpixels, R_{12} is calculated using equation 8:

$$\text{Centroid1} = (X_{c1}, Y_{c1})$$

$$\text{Centroid2} = (X_{c2}, Y_{c2})$$

$$R_{12} = \sqrt{(X_{c2} - X_{c1})^2 + (Y_{c2} - Y_{c1})^2} \quad (8)$$

Where Centroid1 and Centroid2 are the coordinates centroids of the particle at position#1 and position#2 respectively.

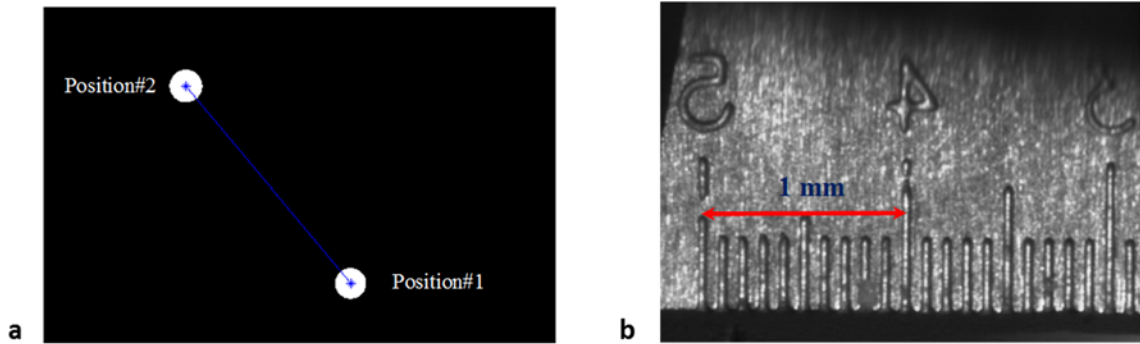


Figure 3-29. a) example of a particle in two different position and b) Image of a ruler for calibration.

To convert this value into actual distance travel by the particle in the experiment, number of pixels per millimeter needed to be determined. For this calibration, a ruler with 0.1 mm resolution is imaged as shown in Figure 3-29b. Number of sub pixels per millimeter, C in the equation 8 is used to obtain the distance in millimeters, d_{12} .

$$d_{12} = R_{12} / C \quad (9)$$

$$V_{12} = d_{12} / t \quad (10)$$

And the velocity can be calculated using the equation 8. Where t is the time. This time is the LED off time for this specific frame which is sent from a microprocessor to the LED driver circuit. To illustrate particle velocity extraction, a specific area in Figure 3-30-a highlighted in the red box has been analyzed which is shown in Figure 3-30-b. This figure consist of a three pulse measurement with equispaced time intervals, 270 Hz rod frequency, 10.6 mm reservoir inner diameter, 30 ms LEDon, 30 ms LEDoff, 30ms camera off, 120-150 μm diameter spheres, and 25/3/400GmL⁻¹ concentration. The highlighted area encompasses 4 particle trajectories which are labeled a,b,c and d. The particles seems to following an approximately circular path around the oscillating probe. Figure 3-30-b shows this region after filtering all the redundant objects and settled particles from the image.

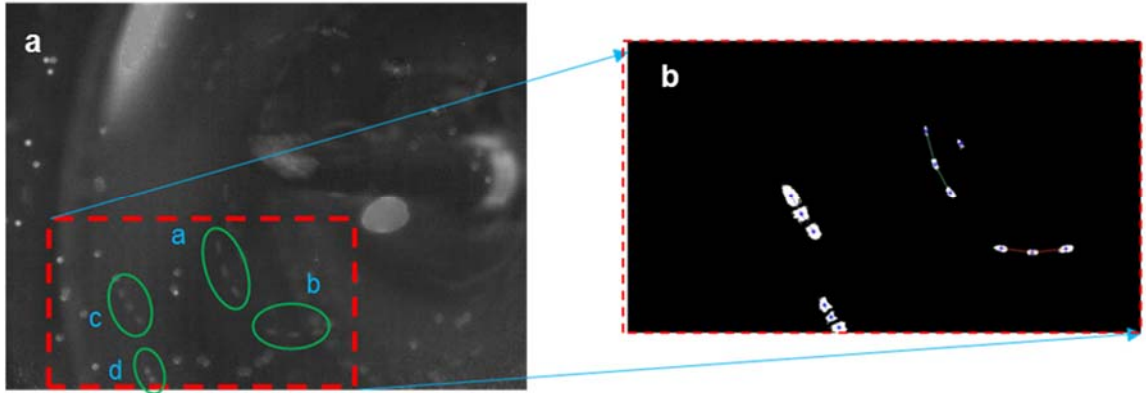


Figure 3-30. a) An image of three-pulsed video experiment, and b) Trajectories of particles shown in red box after image processing.

A velocity measurement program for pulsed-PTV developed in Matlab is used to extract the velocity of the moving particles. This is obtained by finding the distance

between the centroids of particles and dividing it by the LED off time, which is 30 ms in this case. As it demonstrated in Table 3-3, trajectories closer to the rod (a and b) tend to have higher velocity compared to the ones further away from it (c and d).

Table 3-3. Velocities of particle trajectories shown in figure 22.

Trajectory	a	b	c	d
Velocity (mm/s)	4.85	4.69	3.08	1.93

3.3.4 Single probe experiment

To investigate the consistency of the motion of particles with theoretical model, a single probe experiment has been carried out. A detailed study of inertial particle transport and trapping in a viscous streaming flow generated by a single oscillating cylinder is represented in [38]. A mathematical model that has been developed for particle transport in a generic background flow that captures the important aspects of particle trapping previously observed in experiments [3,73]. Figure 3-31 depicts contour of streamline and vorticity from Boundary Element Methods (BEM) obtained from analytical solutions discussed in [74].

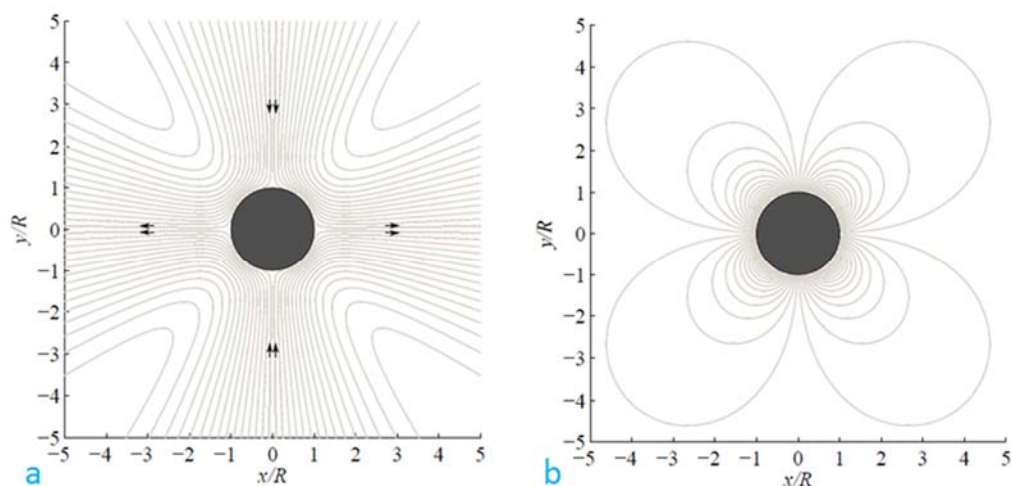


Figure 3-31. Contour of a) streamline and b) vorticity from Boundary Element Methods [74].

Technical parameters for implementing the experiment is listed in Table 3-4.

Table 3-4. Technical parameters for single probe experiment

Frequency	V _{pkpk}	Particle Size	Particle concentration	Grommet diameter	Height of fluid	Height of the fluid above slide
262Hz	135v	10 μm	$\frac{10}{400} \text{ G}\cdot\text{mL}^{-1}$	0.25 in	0.25 in	200 μm

The result of this experiment with the experimental conditions listed in Table 3-4 are shown in Figure 3-32. Figure 3-32-a represents the raw video of the experiment where due to the high density of particles, noise and steady particles, it is difficult to identify the particle's trajectories. However, Figure 3-32-b illustrates the filtered image which contains only the moving particles in the video. And new movie constructed based on the filtered images of the original movie frames represents the motion of the particle's around the oscillating probe. The particle velocity field is reconstructed from filtered video frames

using commercial PIV software (Dynamic Studio from Dantec Dynamics). The result of this analysis is shown in figure 30-c. The main four streaming cells predicted in the simulation results (Figure 3-31-b) is observed around the oscillating probe in the experiment. It is speculated that bi-directional oscillation of the probe creates different particle's dynamics, however a consistency between experimental result and mathematical model for single probe oscillation can be observed.

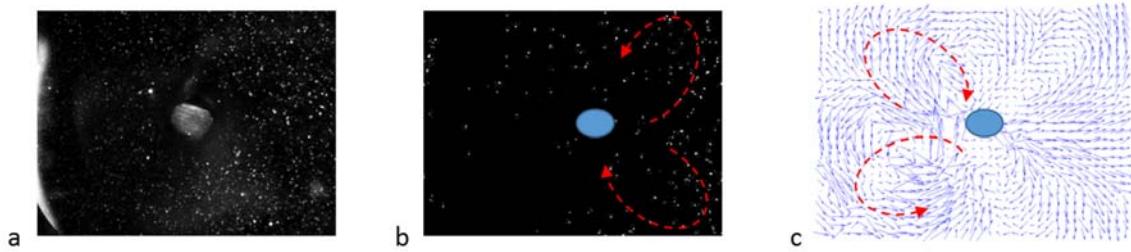


Figure 3-32. a) Raw video frame of particle motion around the probe with circular cross section b) filtered image showing only the moving particles c) Velocity field data reconstructed through comparisons of individual particle positions in successive frames. The blue ellipse shows the position of the probe.

CHAPTER 4: ENCODED-PARTICLE TRACKING VELOCIMETRY (EPTV)

In highly localized flow regions (turbulent, flows with complex or dynamically varying boundaries, or microfluidic flows) or environments with substantial light ‘noise’, multi-particle, particle tracking velocimetry (PTV) measurements can mistake different particles and/or ‘noise’ as a particle track. In an effort to mitigate erroneous measurements, encoded PTV (ePTV) illumination represents a method for the identification of individual particles and the rejection of false readings. To test the effectiveness of the developed technique, images containing 10 to 100 randomly located ‘false’ particles have been generated to represent data containing no flow information. These images are then processed by the ePTV program to extract the coded patterns in the image thereby representing the number of wrong, or false, identified patterns and this is compared to a mathematical model that predicts statistically the number of false patterns. Several simulated studies for different decoding strategies, for instance three pulsed imaging with time intervals in the ratios 1 and 2, and 1 and 3 are evaluated and the results again compared with the statistical model. This model is also modified to incorporate a maximum spacing constraint within the search to eliminate particle velocities that could not provide a complete encoded sequence. Assessment of the theoretical model with simulation using a standard normal variate, Z indicates that the theoretical model and simulated result are consistent within one standard deviation of the simulated results. Similarly, the standard chi-squared, χ^2 analysis validates the model at 95% confidence. A final analytic model is

developed to determine the number of lost particles due to the finite image size based on a calculation of the probability that particles of a specific mean velocity or having a uniform velocity distribution and encoding pattern will exit the field of view. This last model, along with a knowledge of particle densities, can be used to determine the number of particles that will be lost due to a specific encoding strategy and pulse time. The encoded pulse technique has been implemented in experiments for which images containing 100-200 objects including encoded trajectories have been measured. Visually, the encoding patterns can be observed and are identified using the ePTV algorithm and constituted approximately 30 % of the identified objects.

4.1 Mathematical modeling

For modelling purposes it is assumed that there are a large number of particles in a flow field and that each particle has arrived randomly at its location with the overall distribution of particles being uniform when imaged. When an image is produced via an encoded sequence of pulses, each particle will be imaged at multiple locations depending on its velocity, trajectory and the pulse pattern. For a large number of particles within a complex flow it will be necessary to extract each particle trajectory. To do this, every particle in the image is identified by the computer algorithm and assigned an x and y coordinate of its centroidal point. Once all particles in the image are identified, as a first step the algorithm computes all pairs for each particle, the number of these pairs being 1 less than the total number of particles. Following this, the algorithm searches for a further particle in the general direction determined by the vector of the initial pair and with approximately the right distance predicted by the pulse pattern. This repeats until all

encoded particles are found. Two main identifying parameters in the searching algorithm are the maximum searching angle, θ , and the ratiometric spacing tolerance, γ , expressed as a ratio of variation in distance relative to that computed based on an ideal distance for a particle travelling in a straight line with uniform velocity. This ratio is based on the previously measured spacing and encoding pattern, see Figure 4-1. ΔR is the maximum allowable zone for R_2 variation which is obtained from:

$$\Delta R_2 = \gamma R_2, \text{ or, more generally, } \Delta R_i = \gamma R_i \quad (1)$$

This defines a region between two circles with ΔR difference in radius, and of average radius R_2 . Any particle found in this region, makes it acceptable to be part of the potential coded pattern. Figure 4-1 illustrates an image that contains particles in random locations. In theory, none of these should be identified and a single particle with separation of points corresponding to the encoded illumination pulse pattern. However, through chance and with sufficiently large spacing tolerance, there is the possibility that one or more particles might be falsely identified. To decode a specific encoding pattern, defined by the ratiometric values $\{\omega_1, \omega_2, \dots = 1, \frac{\omega_2}{\omega_1}, \dots\}$, the searching algorithm starts with a particle in the image, for instance O_1 . For each of the other particles the distance is calculated and ascribed the first value R_1 for distance with a neighboring particle at coordinates O_2 . Knowing the encoding pattern it is possible to calculate the expected distance R_2 to the next location to search for a particle. If found, this is considered to be the trajectory of a

single encoded particle at coordinate O_3 . For more pulses, the algorithm is continued until all particle locations are found after which the particle trajectory is known.

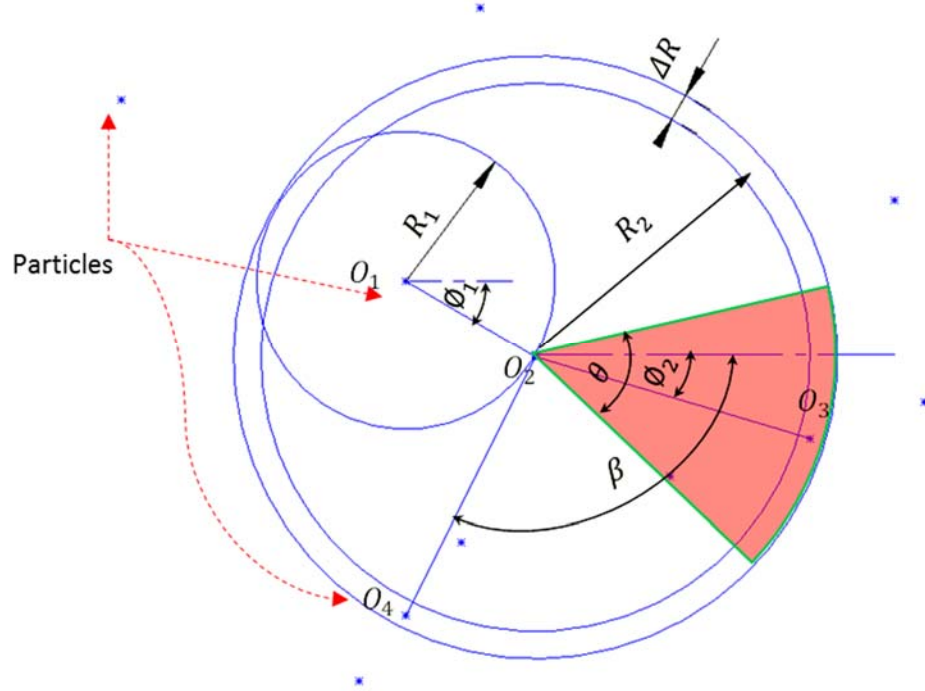


Figure 4-1. Definition of the geometric factors of the mathematical model for encoded-PTV for which the particle located at coordinates O_1 is being evaluated. In this diagram the angular orientation of the θ segment is arbitrary and drawn to visualize the angle limit for the condition in equation (2). All parameters are geometrically determined while θ and γ (thereby enabling calculation of ΔR) are set by the user.

After a pair of particles is selected it is then necessary to seek further instances at the encoded distance. In practice, depending on pulse times and dynamics of the particle stream, the next particle will follow a relatively shallow arc and it may therefore be reasonable to restrict the search to a limited angle, θ , chosen to encompass the largest expected angle deviations. To satisfy the limiting angle requirement in the searching algorithm, the following must be true for a trajectory to be identified as an encoded pattern:

$$|\phi_1 - \phi_2| \leq \theta \quad (2)$$

In which ϕ_1 and ϕ_2 are the angle that O_1O_2 and O_2O_3 make with the horizontal axes respectively. Further illustrated is a particle at O_4 that satisfies the radius requirement but not the angle limit (in this case $|\phi_1 - \phi_2| > \theta$). If the light pulse intervals are small compared with the maximum flow velocities, it can be assumed that the direction of the flow is not changing substantially during an encoded pulse train. This enables a relatively small angle requirement that significantly decreases the possibility of finding wrong trajectories in the flow. However, at too fast pulse intervals, the images of particles may converge and be indistinguishable. For a decoding strategy $\{\omega_1, \omega_2\}$, the spacing tolerance requires:

$$\frac{\omega_2}{\omega_1} R_1 - \gamma R_1 < R_2 < \frac{\omega_2}{\omega_1} R_1 + \gamma R_1 \quad (3)$$

For particles moving at uniform velocity, the pulse encoding will correspond to the special separation of images. However, for non-uniform flows and for measurement systems having finite image noise, there will be a variation in particle separation from the ideal. To accommodate these deviations the spacing tolerance limitation is used. For this purpose, an effective spacing tolerance γ is added to the searching algorithm. Both the limiting angle and spacing tolerance play a significant role in the ability to reject false pattern. This is illustrated in Figure 4-1 where the spacing tolerance rejects all the particles in the image except O_3 and O_4 that satisfies the requirement in equation (3). Similarly, the limiting angle requirement θ rules out O_4 leaving O_1, O_2, O_3 as an identified decoded trajectory with a $\{\omega_1, \omega_2\}$ pattern in the image.

The analysis provided above is based on 2-D PTV and has been applied using our microscope based imaging system that has limited depth of field and therefore approximates to a 2-D measurement. Additionally, the flow patterns are produced by lateral oscillation of a small diameter rod immersed in a cylindrical reservoir of particles. However, this technique can be extended to 3-D PTV. For a small variations of angle, θ , 3-D PTV is equivalent to 2-D, since the projection of 3-D flows on any of the xy , xz and yz plane would maintain the encoded ratio of distances between particles (i.e. $\frac{R_2}{R_1}$ and etc.). The small variation on angle is dependent on particle velocities and pulse time.

For 3-D flow dynamics, the area sections of the mathematical model described for 2-D PTV will be elements of volume. For instance, as shown in Figure 4-2, the angle and maximum spacing form a segment of a spherical shell over a solid angle measured in steradians, ψ , which is equivalent to the highlighted sector in Figure 4-1. Any particle found in this region, will be accepted to be part of a trajectory search.

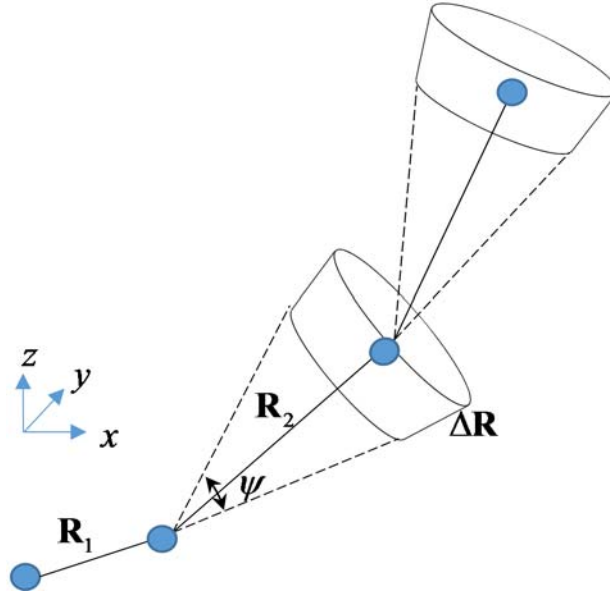


Figure 4-2. Image of particle traveling in 3-D environment. \mathbf{R}_1 and \mathbf{R}_2 represent the vector distances between consecutive particle in a trajectory. ψ and $\Delta\mathbf{R}$ which are angle (steradians) and maximum spacing. Any particle found in this volume would be accepted to be part of the trajectory.

4.1.1 Probability analysis

To calculate the probability of incorrectly finding coded patterns in an image containing randomly located particles, a probability analysis based on the encoder algorithm described in the previous section has been carried out. Particle density ρ is defined as $\rho = \frac{N}{A}$ where N is total number of particles and A is the total area of the image. To simplify the mathematical calculation, the square images are generated in 1 X 1 size, which makes A to be equal to 1. ϕ_i is the angle which $O_i O_{i+1}$ makes with X axis. In a general case, different limiting angles, θ_i can be applied for particles in a pattern, this is illustrated in Figure 4-3 below.

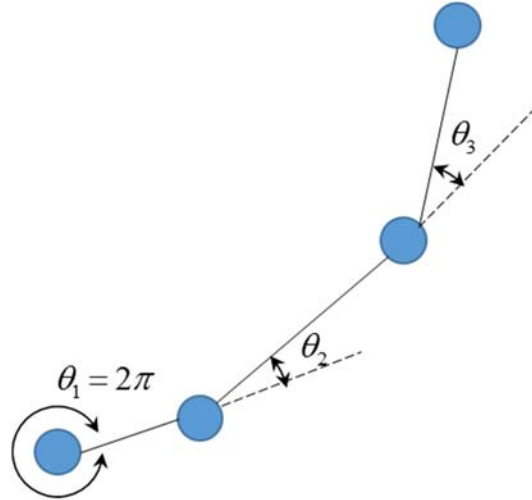


Figure 4-3. Image showing particles having different limiting angle in a pattern.

As an example, for a decoding pattern of $\{1\ 2\}$, $(O_1, O_2, O_3 \text{ trajectory})$ the probability of finding O_2 , the second particle is P_1 , and O_3 the third particle is P_2 which is obtained from:

$$P_1 = \rho \frac{\theta_1 R_1 \gamma}{A} \quad (\theta_1 = 2\pi) \quad (4)$$

$$P_2 = \rho \frac{\theta_2 (2R_1) \gamma}{A} \quad (5)$$

Therefore, the probability of finding a trajectory in $\{1\ 2\}$ pattern, P_T is:

$$P_T = P_1 \cdot P_2 = \rho^2 \frac{(\theta_1 \cdot \theta_2) 2R^2}{A^2} (\gamma)^2 \quad (6)$$

In general if a coding strategy of $\{\omega_1, \omega_2, \dots, \omega_i, \dots, \omega_n\}$ is used the following can be applied:

$$P_i = \rho \frac{\theta_i (\omega_i \cdot R_1) \gamma}{A} \quad (7)$$

$$P_T = P_1 \cdot P_2 \cdot \dots \cdot P_n = \left(\frac{\rho R \gamma}{A} \right)^n \left(\prod_{i=1}^n \theta_i \omega_i \right) \quad (8)$$

This number shows the probability of finding a coded pattern in a random image. Hence, this probability should be as low as possible to reject wrong trajectories. Equation (4) shows that reducing θ_i and γ is desired, however, this increases the likelihood of missing a true trajectory if the coding is used. On the other hand, it increases the probability of type II error. Differences in angular path changes will be related to vorticity. For relatively fast pulse timing, particle paths are likely to show only small curvature. Hence the difference in trajectory angles can provide a further measure to increase confidence in particle identification. For example, when multiple points are determined the ones with least deviation might be selected. Also, when a false trajectory is identified and it shares a particle with another identified trajectory it is likely that the angular deviation of the false trajectory would violate the angle requirement. This implies the importance of choosing an appropriate values for θ_i and γ . A statistical study for this purpose has been carried out which is presented in the result section.

4.1.2 Probability of encoded particle trajectories going outside of the boundary of the image

The previous section developed equations for predicting the probability of falsely detecting a coded particle in a pattern of particles generated by uniformly distributed noise. Another concern when using multiple pulses is that of a real particle exiting from the field of view during measurement. This section develops an approximate theoretical model to calculate the probability that a particle of a given velocity will be lost from the

measurement. Two velocity distributions are considered, the first contain particles that all have the same velocity, the second for a field of particles with uniform velocity distribution up to a maximum value.

4.1.2.1 Particles all traveling at the same velocity

Figure 4-4 shows a model of the image in which the outer rectangular box (solid lines) represents the image border. Again a uniform, random distribution of particles is assumed with a complex flow within which particles can be moving in any direction. In this figure, T is the distance that a particle will travel for the given encoding strategy, i.e. the initial and final position of the particle in the image. In this case, because all particles are assumed to be moving at the same speed, this will be a constant. Therefore, the area enclosed by the dashed rectangle inside the image border represents the region that ensures the particle will not exit from the field of view. In other words, any particle that is identified with the initial pulse inside this region is guaranteed to stay inside the image throughout the pulse train. However, if the particle is located outside the 100% region, the probability of the pattern exiting from the image increases. When the initial pulse is from a particle at the image border, for a random flow field, there will be a 50% probability that its full trajectory will be imaged. The circle of radius T represents a particle in the flow field with the origin being its location at the initial pulse and the radius representing the end point of its trajectory at the final pulse in the encoding train. Because a random direction is assumed, the circle represents a locus of all possible endpoints each having the same probability. As shown in Figure 4-4, h represents the distance from the dashed line to the center of the circle.

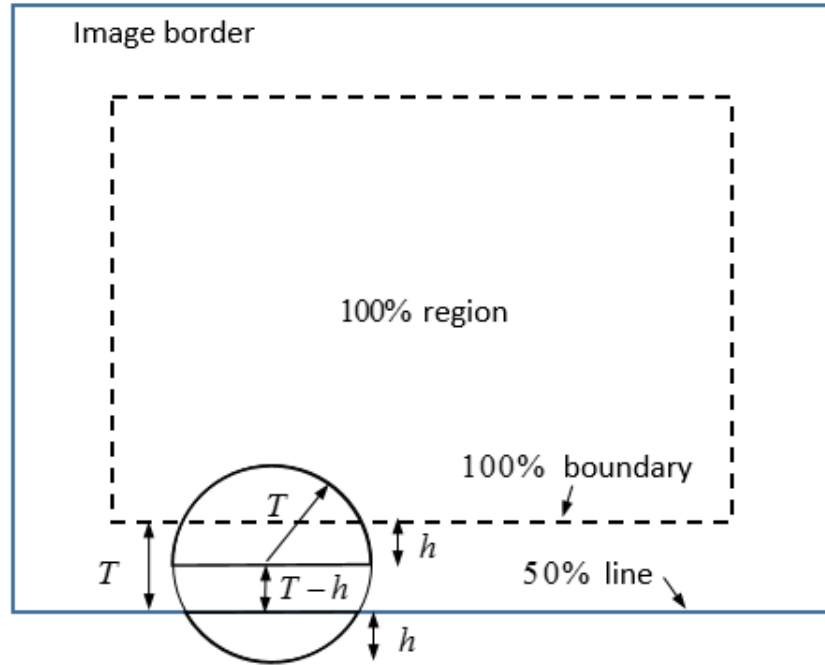


Figure 4-4. Diagram of the Image when particles are traveling at maximum velocity

$$V_{\max} = T.$$

To derive a relationship to determine the probability of losing an encoded particle, the model shown in Figure 4-5 represents an expanded view of the circle shown in Figure 4-4. The length of the end-point circle chord that is outside of the image border, L^* , is proportional to the probability of missing a particle. The remaining length, L , that is inside the image border is proportional to the probability of finding a good pattern inside the image.

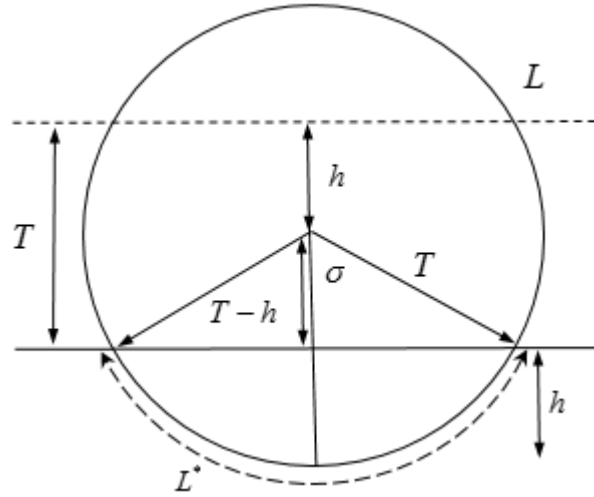


Figure 4-5. Section view of the circle in Figure 4-4 for probability calculation.

The length of chord outside of the image border, L^* can be determine from

$$L^* = 2\sigma T = 2R \cos^{-1} \left(\frac{T-h}{T} \right) \quad (9)$$

Where

$$\cos \sigma = \frac{T-h}{T} \rightarrow \sigma = \cos^{-1} \left(\frac{T-h}{T} \right) \quad (10)$$

Therefore, the length of the chord that is inside the image, L which is proportional to the probability of finding a complete trajectory is

$$L = 2\pi T - L^* = 2\pi T - 2T \cos^{-1} \left(\frac{T-h}{T} \right) \quad (11)$$

The probability of finding a pattern in the image while particles are traveling at maximum velocity, P_T is the ratio of this length to the circle circumference given by

$$P_T = \frac{2\pi T - 2T \cos^{-1} \left(\frac{T-h}{T} \right)}{2\pi T} = 1 - \frac{1}{\pi} \cos^{-1} \left(1 - \frac{h}{T} \right) \quad (12)$$

At the corners of the image, the distance from the 100% line to the image border will increase in a direction towards the corners. A complete analysis would lead to an increase in the effective chord length remaining within the image border. There will be a corresponding increase in the probability of finding particles that for most imaging condition will be small and is not included in this analyses.

4.1.2.2 Particles traveling with uniformly distributed velocity

In a general case particles are moving with a uniform distribution of velocities up to a maximum value. In the case of particles moving with a uniform velocity distribution, any particular particle in the pattern will move a distance τ for a given encoding strategy. This distance is any value less than or equal to the maximum distance T traveled at maximum velocity. Therefore, to determine the probability of finding any particular particle, the circle shown in Figure 4-6 has a radius of τ . For an arbitrary distribution of velocities the mathematical model of Figure 4-5 applies only in this case with the circle having radius τ .

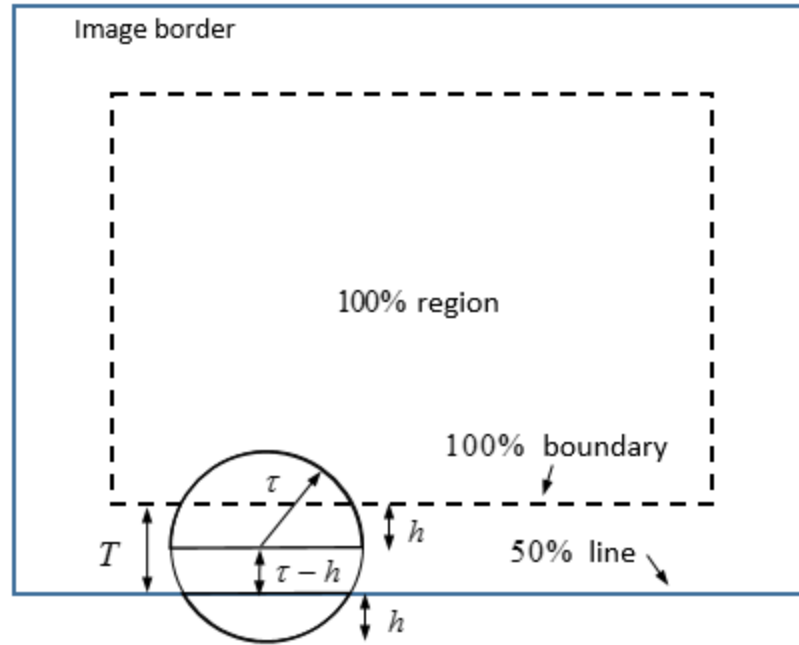


Figure 4-6. Diagram of the Image when particles are traveling with uniformly distributed velocity $V = \tau$.

Again the angle θ is calculated as

$$\cos \sigma = \frac{T-h}{\tau} \rightarrow \sigma = \cos^{-1} \left(\frac{T-h}{\tau} \right) \quad (13)$$

and the length of chords L^* and L are

$$L^* = 2\sigma T = 2T \cos^{-1} \left(\frac{T-h}{\tau} \right) \quad (14)$$

$$L = 2\pi T - L^* = 2\pi T - 2T \cos^{-1} \left(\frac{T-h}{\tau} \right) \quad (15)$$

The probability of finding a pattern in the image while particles are traveling with a specific velocity, τ is again the ratio of the length L by the circle circumference:

$$P(\tau) = \frac{2\pi T - 2T \cos^{-1}\left(\frac{T-h}{T}\right)}{2\pi T} = 1 - \frac{1}{\pi} \cos^{-1}\left(\frac{T-h}{\tau}\right) \quad (16)$$

This needs to be integrated from 0 to T to provide an evaluation of the probability for the whole spectrum of uniformly distributed particle velocities. In this case it is apparent from the figure that while τ is less than $T-h$ the probability of finding a good pattern is still 100% because the whole circle is inside the image border, i.e. the length of the chord outside the image is 0. Thus the probability $P(\tau)$ for this condition is 1. So, $P(\tau)$ needs to be integrated through τ only when τ is greater than $T-h$. Therefore, the total probability of finding a good pattern in the image while particles are moving with a uniformly distributed velocities, P_τ is

$$P_\tau = \frac{\int_0^{T-h} (1) d\tau + \int_{T-h}^T P(\tau) d\tau}{\int_0^T (1) d\tau} \quad (17)$$

Where

$$\begin{aligned} \int_{T-h}^T P(\tau) d\tau &= \int_{T-h}^T \left\{ 1 - \frac{1}{\pi} \cos^{-1}\left(\frac{T-h}{\tau}\right) \right\} d\tau \\ \frac{T-h}{\tau} &= u \rightarrow (T-h) \frac{-1}{\tau^2} d\tau = du \\ \int_{T-h}^T P(\tau) d\tau &= h + \frac{T-h}{\pi} \int_{T-h}^T \frac{\cos^{-1} u}{u^2} du \\ \int_{T-h}^T P(\tau) d\tau &= h - \frac{1}{\pi} \left[T \cos^{-1}\left(\frac{T-h}{T}\right) - (T-h) \log \left| \frac{T \left(1 + \sqrt{1 - \left(\frac{T-h}{T}\right)^2} \right)}{T-h} \right| \right] \end{aligned} \quad (18)$$

Substituting (19) into (18) and simplifying yields

$$P_r = \frac{\int_0^{T-h} (1)d\tau + \int_{T-h}^T P(r)d\tau}{\int_0^T (1)d\tau} = 1 - \frac{1}{\pi} \cos^{-1} \left(1 - \frac{h}{T} \right) + \frac{1}{\pi} \log \left| \frac{\left(1 + \sqrt{\frac{h}{T} \left(2 - \frac{h}{T} \right)} \right)}{\left(1 - \frac{h}{T} \right)} \right|. \quad (19)$$

Equation (19) can be used to determine the probability that a particle from a uniform distribution of randomly oriented velocities will be detected as a function of its location relative to the image border.

4.1.3 Physical interpretation

For particles undergoing accelerations the trajectory of a particle will deviate from the spacing predicted from a uniform velocity. These deviations are accounted for with the spacing tolerance outlined in section 4.1. However, when selecting pulse timing it is necessary to take into account the acceleration effect. The deviation of the particle trajectory from the encoder pattern will be a function of velocity, acceleration and the time between pulses. Figure 4-7 shows the modeling parameters for a particle in an image captured by camera for a three pulse encoder pattern. The particle in the initial position has velocity of v_0 . In time interval of Δt_1 it travels a distance of l_1 reaching a velocity of v_1 . In the second blink of LED, the particle is captured in the image while it traveled distance l_2 in time interval Δt_2 .



Figure 4-7. Image of particles in a trajectory coded in pattern $\{1\ c\}$.

Assuming that $\{1\ C\}$ coding is used for LED pulsing, therefore $\Delta t_2 = c\Delta t_1$.

Considering the initial velocity of particles at each position and the acceleration, the distances R_1 and R_2 can be obtained from the following:

$$\begin{aligned} R_1 &= v_0 \Delta t_1 + \frac{a \Delta t_1^2}{2} \\ R_2 &= v_1 (c \Delta t_1) + \frac{a (c \Delta t_1)^2}{2} \end{aligned} \quad (20)$$

These equations are consistent with the development of [59], see equation 11 in this chapter, for triple pulse measurement that these authors use to determine particle velocity and acceleration. For the purpose of this analysis, it is necessary to determine the effects of the acceleration on the encoded pattern. As a first step, in an ideal situation it is, temporarily, assumed that particles traverse without acceleration ($a = 0$):

$$\begin{aligned} R_1 &= v_0 \Delta t \\ R_2 &= v_1 (c \Delta t) \quad (\text{If } a=0 \Rightarrow v_0 = v_i) \\ \frac{R_2}{R_1} &= c \end{aligned} \quad (21)$$

While in experimental circumstances, there is always acceleration involved. When there is an acceleration ($a \neq 0$):

$$\begin{aligned} R'_1 &= v_0 \Delta t_1 + \frac{a \Delta t_1^2}{2} \\ R'_2 &= v_1 (c \Delta t_1) + \frac{a (c \Delta t_1)^2}{2} \end{aligned} \quad (22)$$

Where R'_1 and R'_2 are the actual distances that particle traverse in the time intervals

Δt_1 and Δt_2 with the presence of acceleration.

$$\frac{R'_2}{R'_1} = \frac{v_1(c\Delta t_1) + \frac{a(c\Delta t_1)^2}{2}}{v_0\Delta t_1 + \frac{a\Delta t_1^2}{2}}$$

$$(v_1 = v_0 + a\Delta t_1) \quad (23)$$

$$\frac{R'_2}{R'_1} = \frac{(v_0 + a\Delta t_1)(c\Delta t_1) + \frac{a(c\Delta t_1)^2}{2}}{v_0\Delta t_1 + \frac{a\Delta t_1^2}{2}} = \frac{v_0c + ac\Delta t_1 + \frac{ac^2\Delta t_1}{2}}{v_0 + \frac{a\Delta t_1}{2}} = c \frac{v_0 + a\Delta t_1 + \frac{ac\Delta t_1}{2}}{v_0 + \frac{a\Delta t_1}{2}} \quad (24)$$

Therefore the error (\mathcal{E}) between ideal situation ($a = 0$) and experimental situation ($a \neq 0$) is:

$$\mathcal{E} = \frac{R_2}{R_1} - \frac{R'_2}{R'_1} = c - c \frac{v_0 + a\Delta t_1(1 + \frac{c}{2})}{v_0 + \frac{a\Delta t_1}{2}} \quad (25)$$

For extreme conditions either a highly accelerated dynamical situation (i.e. $a \rightarrow \infty$) or when the particle has no initial velocity (i.e. $v_0 \rightarrow 0$) equation (25) converges to

$$\lim_{a \rightarrow \infty} \mathcal{E} = \lim_{a \rightarrow \infty} \left(c - c \frac{v_0 + a\Delta t_1(1 + \frac{c}{2})}{v_0 + \frac{a\Delta t_1}{2}} \right) = \lim_{v_0 \rightarrow 0} \left(c - c \frac{v_0 + a\Delta t_1(1 + \frac{c}{2})}{v_0 + \frac{a\Delta t_1}{2}} \right) = -c - c^2 \quad (26)$$

In conditions that particle has infinite initial velocity (i.e. $v_0 \rightarrow \infty$):

$$\lim_{v_0 \rightarrow 0} \varepsilon = \lim_{v_0 \rightarrow 0} \left(c - c \frac{v_0 + a\Delta t_1 (1 + \frac{c}{2})}{v_0 + \frac{a\Delta t_1}{2}} \right) = (c - c) = 0 \quad (27)$$

When the time interval is infinitesimally small (i.e. $\Delta t_1 \rightarrow 0$):

$$\lim_{\Delta t_1 \rightarrow 0} \varepsilon = \lim_{\Delta t_1 \rightarrow 0} \left(c - c \frac{v_0 + a\Delta t_1 (1 + \frac{c}{2})}{v_0 + \frac{a\Delta t_1}{2}} \right) = 0 \quad (28)$$

The error goes to zero. This demonstrate that it is desired to use Δt_1 as small as possible but, in practice, will be limited to the experimental capability and camera resolution in detecting movements in very small time intervals. The above equations enable the selection, or at least provide quantitative value for consideration, when selecting pulse patterns and timing and well as deciding the spacing tolerance for post measurement analysis.

4.2 Experimental Apparatus

The main elements of an optical PTV a depicted in Figure 4-8. The illumination part plays a significant role in particle's flow encoding which is then recorded by camera. Raw images recorded from the experiments are usually extremely noisy which makes it substantially hard to extract information. Therefore, multiple image processing techniques are needed to be implemented on the images in order to analyze the information in the images. Subsequently, the statistical analysis of the extracted information is used to summarize the kinematic/dynamic behavior of the fluid flow. By pulsing the light source twice per camera exposure, each particle shows up twice in an image. By coding the light

by filling up the bowl of this scoop and then using a flat knife edge to create a level surface up to the rim. The volume contained in this level scoop is, S, in other words 1 S indicates a full level scoop of particles. Density of particles is defined as $\frac{x/20}{20\text{mL}} \text{ S}$, e.g. to obtain a unit concentration, one flat scoop of particles needed to be mixed into 400 mL of water. Biocompatible surfactant (Cospheric Tween20) is used to minimize the contact area between particles and prevent agglomeration. A solution containing 0.1% of tween20 mixed with distilled water and heated to 80 °C is used (i.e. 0.4 mL of tween20 in 400mL of distilled water). One flat scoop of spheres are added to the solution soaking them at room temperature. Particles in these solutions are spheres with 125-150 micrometer diameter. In this step we have $\frac{1}{400} \text{ S} \cdot \text{mL}^{-1}$ concentration. To create concentrations of higher density, 100 mL of this solution is emptied in a reservoir and 1 S is added to the remaining 300 mL solution, which provides $\frac{2 \frac{1}{3}}{400} \text{ S} \cdot \text{mL}^{-1}$ of concentration.

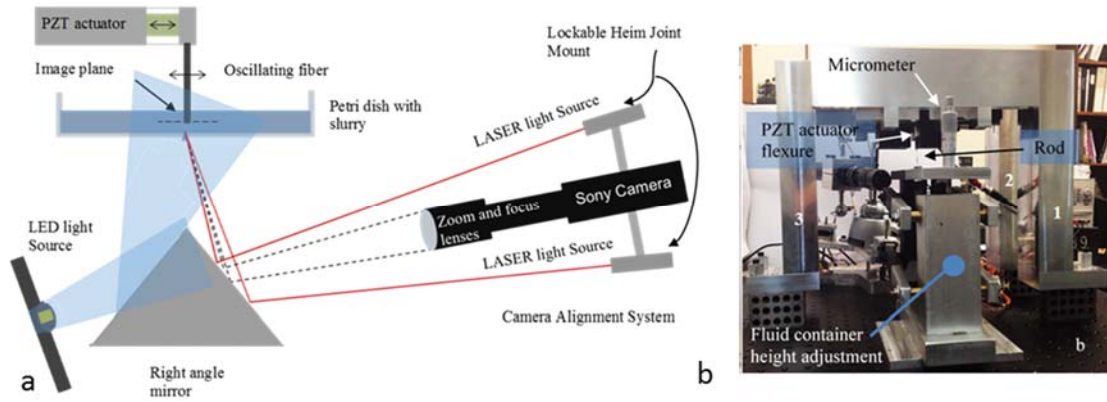


Figure 4-9. Particle tracking velocimetry a) design of the optical setup and imaging system b) experimental apparatus

Image processing techniques are used to enhance the quality of the image and improve the signal-to-noise ratio of the particle images relative to the background and to eliminate redundant light and particles in the image. Figure 4-10 shows the algorithm implemented in the developed technique. The image taken in the experiment converts to a binary image after transmuting to black and white (B&W) and thresholding the histogram. More details about the developed experimental facility for particle manipulation study are represented in [42,43]. To filter the small particles and dust in the image that are redundant, filling of localized regions of high intensity in the image is implemented. Then, objects having an area below a specified value are removed from the image. The centroids of the remaining particles are then identified as objects. Next, all objects are labeled in the image giving each of them a separate identity. The distances between the centroids of each object and all other objects are calculated. Using this data, the search algorithm looks for the pulse sequence pattern among the distances of a specific particle with all others, for instance, ($t, 2t$ or, more specifically, ratios of 1 to 2). Once it is recognized, the pattern becomes highlighted with a distinct color in the output image as an identified coded trajectory.

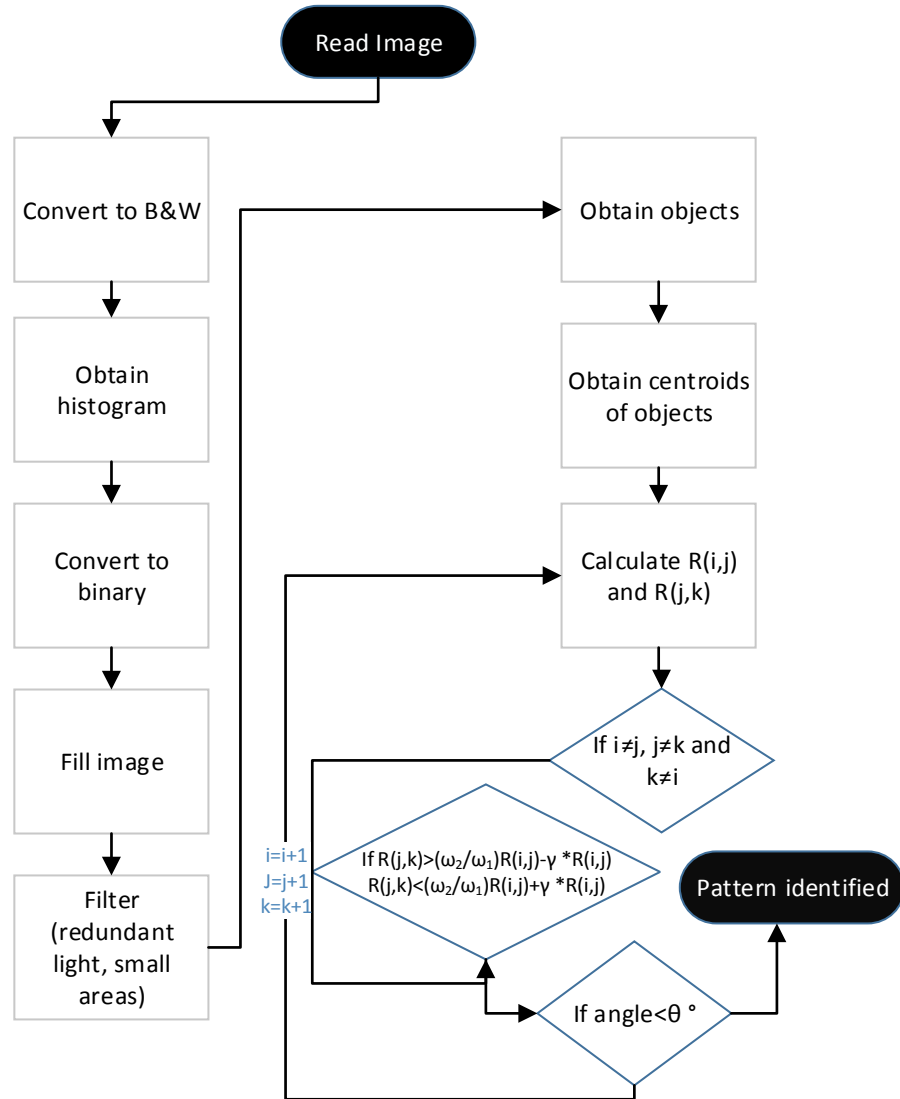


Figure 4-10. Flowchart of the encoding algorithm for encoding $\{\omega_1, \omega_2\}$ and maximum angle θ . Sequence shows how to identify encoded trajectory in an image.

Two examples of this implementation are represented in the next section 4.2.1.

4.2.1 Particle trajectory tracking

To test the functionality of the algorithm, multiple PTV cases has been studied.

Figure 4-11-a illustrates a combination of particles that has been modeled in the $\{1\ 2\}$

coding pattern. As mentioned before, this means the distance of centroids between consecutive particles in a trajectory are $t, 2t$. The algorithm was able to decode the patterns as is shown in blue, red and white colors in Figure 4-11-b. Each trajectories contains three particles with $t, 2t$ spacing.

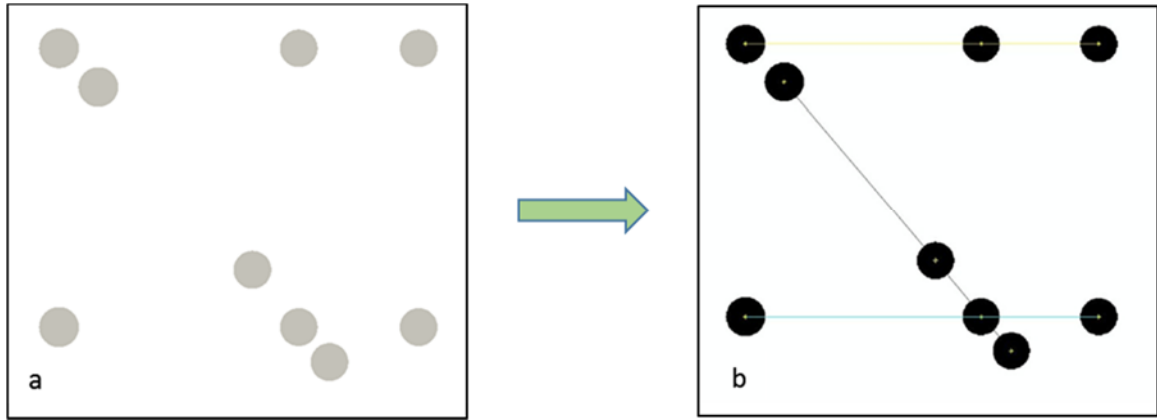


Figure 4-11. Encoding with $\{1\ 2\}$ pattern, a) image containing particles in a coded patterns b) identified coded trajectories after implementing the encoded-PTV algorithm.

A more complex case study is shown in Figure 4-12-a which includes particles in $\{1\ 2\ 3\}$ pattern with linear and circular trajectories and all particles moving with the same velocity. Not only it is hard to identify these pattern by eye but also most current PTV software are also unable in detecting the trajectories of particles precisely. The outcome of the algorithm applied to this image is illustrated in Figure 4-12-b highlighting the linear and circular trajectories with distinct colors.

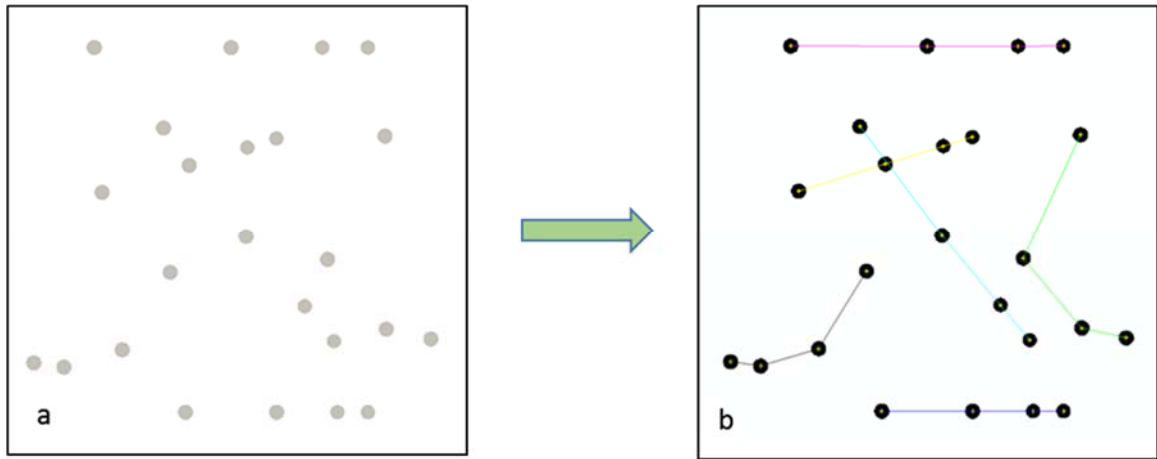


Figure 4-12. Encoding with $\{1\ 2\ 3\}$ pattern, a) image containing particles in a coded patterns b) identified coded trajectories after implementing the encoded-PTV algorithm.

In the above examples no random particles, i.e. not belonging to a specific coded pattern, were added to the images. Also, the distance between the coded particles in each pattern was perfectly precise. Clearly, these situation rarely can happen in an experimental circumstance. There will be random particles in the image due to the stationary particles or noise or even the particles that go off the chart during their dynamic movement. Correspondingly, the space between the particles is not exactly matched with the LED pulsing due to acceleration or imperfect pulse timing. In order to understand the effect of imperfect spacing and present of random particles on the encoded-PTV results, a simulation studies has been carried out to compare the result with the ideal mathematical model developed in section 4.1. Figure 4-13 illustrate a sample image containing 100 randomly positioned particles used for Monte Carlo simulation that will be described in the next section.

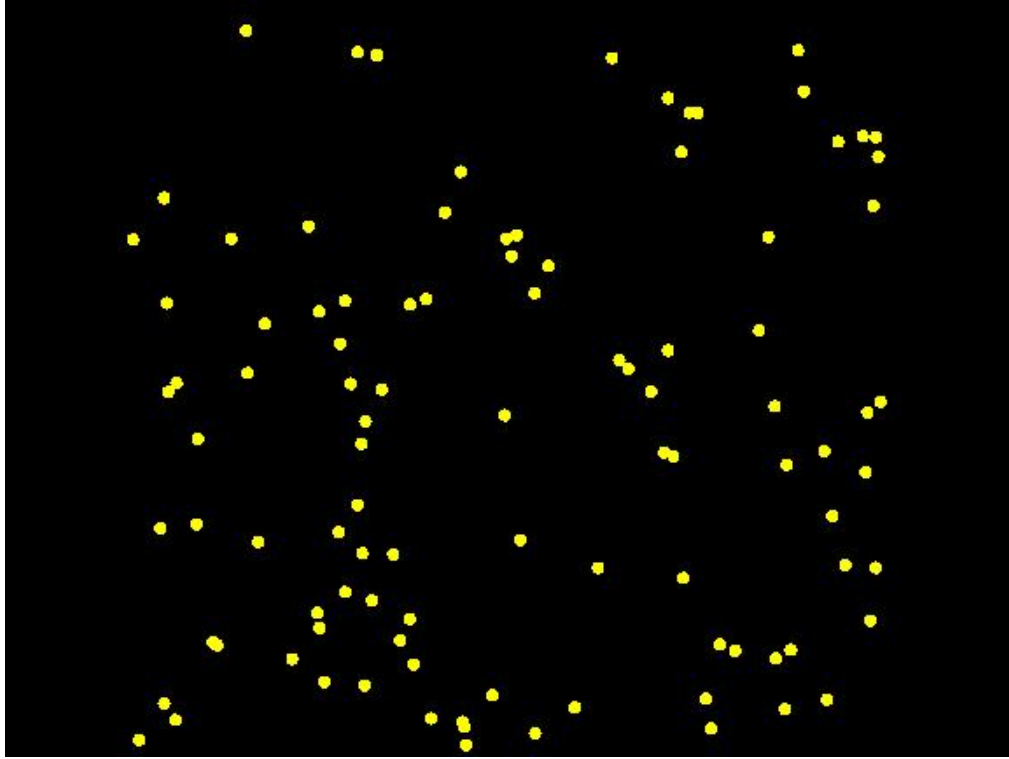


Figure 4-13. Photograph of an image with randomly generated particles density of 100. These images has been used in the developed ePTV software to find the number of (false) found patterns for specific coding strategies.

4.3 Results

A computer simulation has been implemented to explore the effect of various parameters of the mathematical model (i.e. limiting angle, spacing tolerance, density and encoding strategy) in the outcome of the developed encoded-PTV algorithm. In every simulation for a particular decoding pattern, images are created for density from 10 to 100 in step of 10. Density is equal to the number of randomly generated particles in an image in a normal distribution condition. Furthermore, to get enough data for statistical analysis, 10 images have been created at each of the specific densities with random located particles having a uniform distribution in the image. For each image the limiting angle θ is varied

from 10 to 360 degrees in step of 20 degrees. Similarly, the spacing tolerance γ is varied from 0 to 1 (percent) in steps of .01. Next, the Encoded-PTV algorithm is applied for these images to identify trajectories in a demand decoded pattern. The statistical average of the number of founded patterns \bar{N}_s for 10 different images for a specific decoding and density are calculated and plotted. Figure 4-14 shows this plot for density of 30 and decoding $\{1 \ 2\}$.

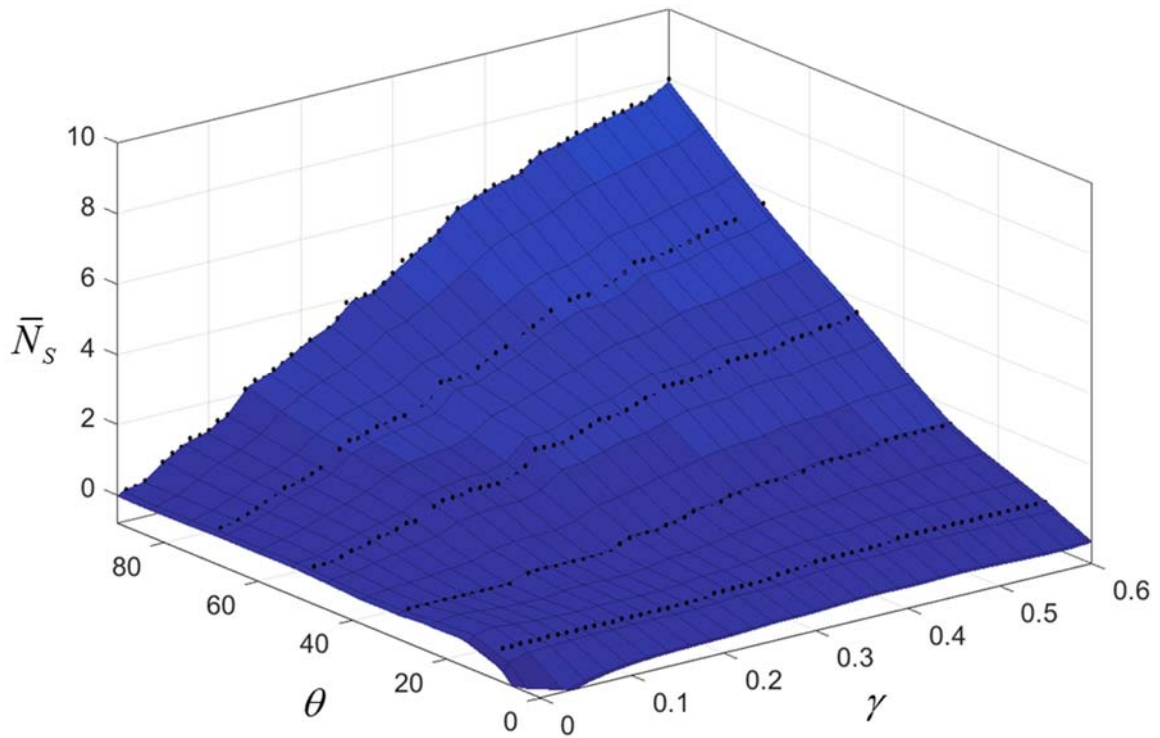


Figure 4-14. Simulation results for $\rho = 30$, $\theta = 10-90$, $\gamma = 0.01-0.6$, and decoding pattern $\{1 \ 2\}$. \bar{N}_s represents the average simulated number of found patterns.

Figure 4-15 represents the plot for density of 100 for the same decoding strategy.

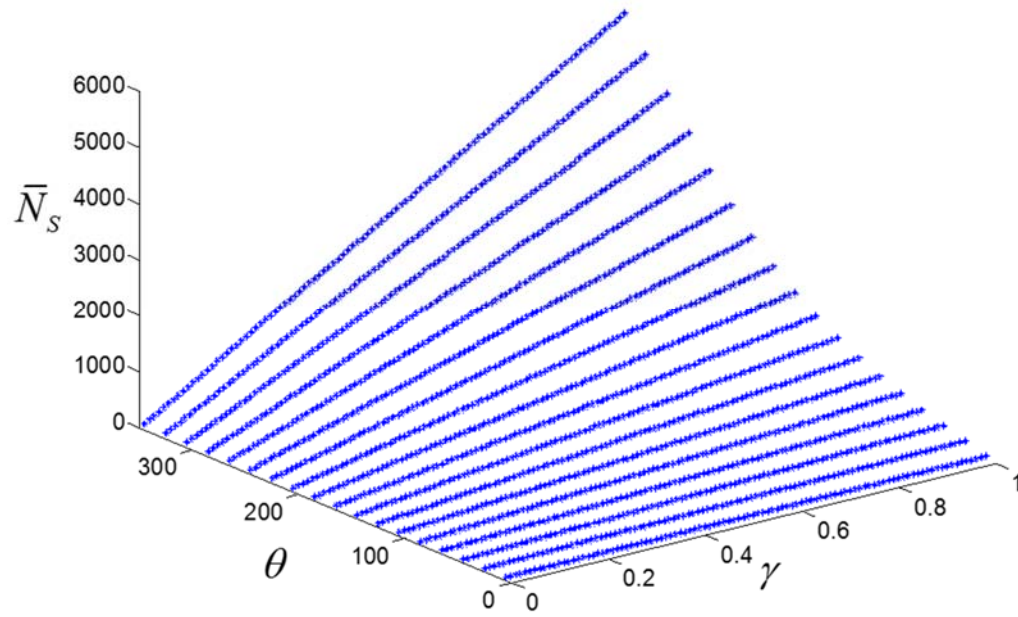


Figure 4-15. Simulation results for $\rho = 100$, $\theta = 10-360$, $\gamma = 0.01-1$, and decoding pattern $\{1\ 2\}$.

Figure 4-16 shows all the calculated data for decoding $\{1\ 2\}$ for all 10 different densities.

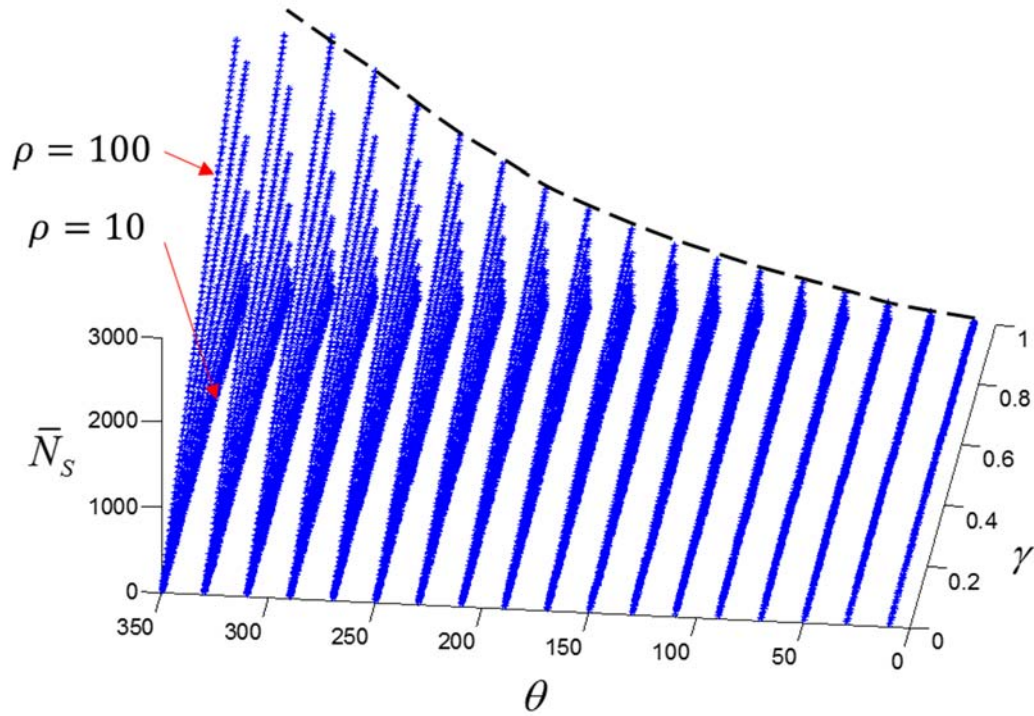


Figure 4-16. Simulation results for $\rho = 10-100$, $\theta = 10-360$, $\gamma = 0.01-1$, and decoding pattern $\{1\ 2\}$.

From figures 12-14 it can be observed that by increasing the density, the \bar{N}_s increases. Equivalently, \bar{N}_s increases when θ and γ increases. It can be observed that the relationship with \bar{N}_s and γ is a linear relationship, while θ has a nonlinear relationship with \bar{N}_s which are explained in more detail in the next section. The decoding pattern $\{2\ 1\}$ has been also explored which revealed an interesting result showing \bar{N}_s for decoding pattern $\{2\ 1\}$, \bar{N}_s is twice as \bar{N}_s . This is depicted in Figure 4-17 where it can be seen that $2\bar{N}_s$ and \bar{N}_s are almost overlaid.

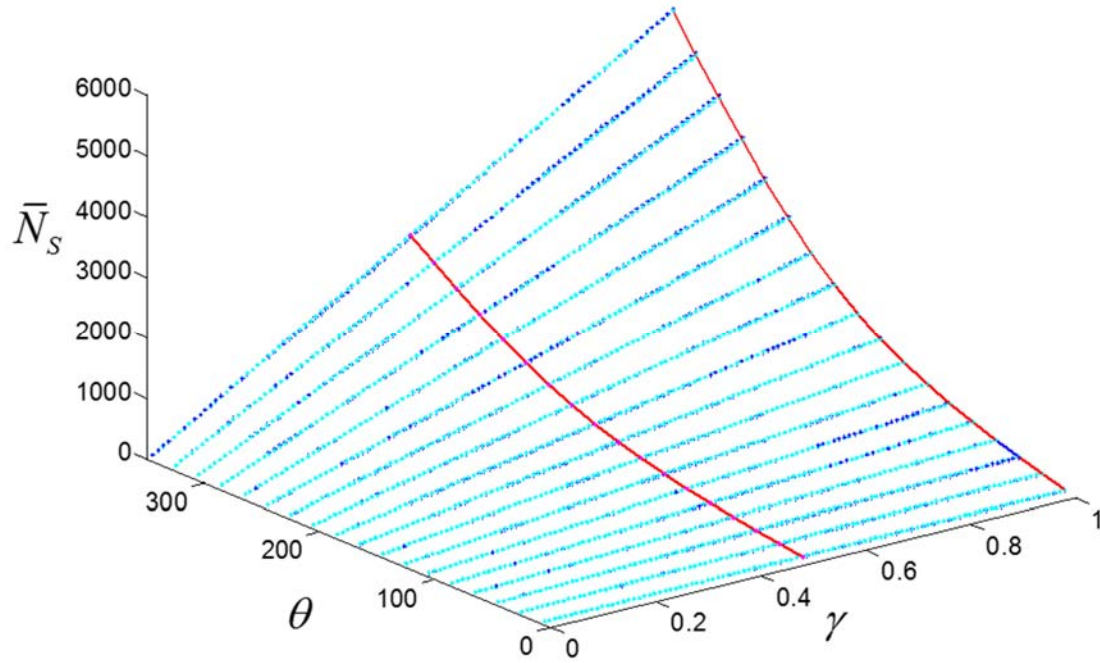


Figure 4-17. Simulation results for $\rho = 100$, $\theta = 10-360$, $\gamma = 0.01-1$, decoding pattern $\{1\ 2\}$ vs decoding pattern $\{2\ 1\}$ showing the same rejection capability. Blue color represents \bar{N}_s and cyan color represents $2\bar{N}_s$.

The same relationship occurred between \bar{N}_s and \bar{N}_s for decoding pattern $\{3\ 1\}$ and $\{1\ 3\}$ simulated results. This can be straightforwardly explained from the linear relationship of \bar{N}_s and γ . From equation (1) the spacing tolerance for decoding pattern $\{2\ 1\}$ is twice as the spacing tolerance for decoding pattern $\{1\ 2\}$. This basically leads us to the conclusion that when γ gets multiplied by factor of two, the number of founded coded pattern also doubles.

4.3.1 Comparison between theoretical model and simulation studies

The theoretical number of found (N_T) is obtained using the following formula:

$$N_t = \rho \alpha M \quad (29)$$

Where α is the edge effect parameter and M is the spacing constraint effect. These two parameters are explained in the following sections.

4.3.1.1 Spacing constraint effect

M is the sum of number of founds for all the possible spacing limit which is calculated in the mathematical model (see equation 4). This is obtained from:

$$M = \frac{\sum_{R=0.1}^{R_E} \rho \theta R^2 \gamma}{n} \quad (30)$$

$$n = \frac{R_E - 0.1}{0.1} + 1 \quad (31)$$

R_E is the maximum spacing constraint which depends on the coding strategy. The mathematical number of found as described in section 4.1 is obtained by $\rho \theta R^2 \gamma$. For a specific case study all the parameters in this formula are known except for R which is varying based on the actual locations of the particles found in the image. Therefore R is assumed to be varying in the equation (30) and the results are added together as the total number of found. However, R cannot exceed more than a specific limit, R_E regarding the specific coding strategy that is used. This is illustrated in Figure 4-18 for coding pattern of $\{1\ 2\}$. In other words, if the first spacing between the particles, R exceeds R_E , the particle goes off the image which makes it impossible be identified as a trajectory with that coding strategy.

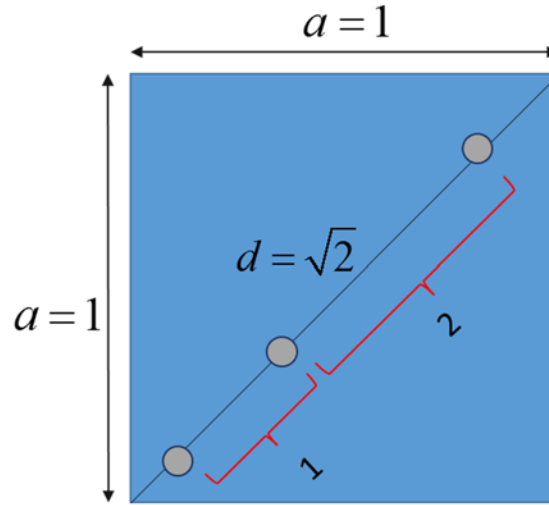


Figure 4-18. Maximum spacing constraint for coding pattern {1 2} in an image size of 1x1.

As described before, each image is set to 1x1 unit dimension. It is obvious that the longest spacing between the particles can occur along the diagonal in the image. The maximum spacing constant is defined as the ratio of the diagonal to the spacing elements of coding. For instance for {1 2} coding pattern:

$$R_E = \frac{d}{1+2} = \frac{\sqrt{2}}{3} = 0.47 \approx 0.5 \quad (32)$$

Similarly, if coding {1 2 1} is used then:

$$R_E = \frac{d}{1+2+1} = \frac{\sqrt{2}}{4} = 0.35 \approx 0.4 \quad (33)$$

4.3.1.2 Edge effect

For particles located near the center of the image it is possible to sweep the angle up to 360 degrees in search for next particles in a trajectory. However, for the particles

closer to the edges of the image the searching angle cannot exceed more than a value because there will be no particles there. This is illustrated in Figure 4-19:

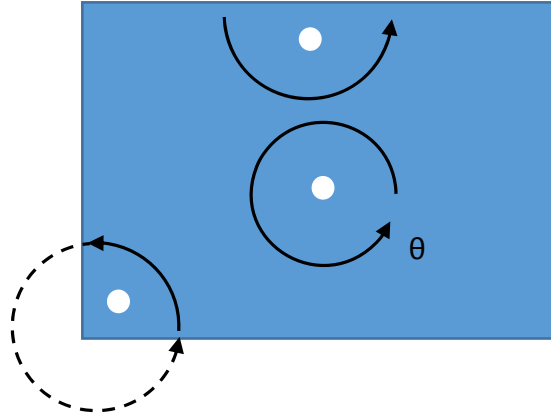


Figure 4-19. Representation of edge effect problem limiting the searching angle for particles at different locations in an image.

After running the algorithm for different case studies, it is observed that the edge effect parameter, α is a quadratic function of angle θ for different coding and densities.

$$\alpha = a_1\theta^2 + a_2\theta + a_3 \quad (34)$$

The α value for different angles and spacing tolerances is plotted in Figure 4-20 for decoding pattern {1 2} in an image of density 100. It is obvious that the spacing tolerance has no effect in α while it has a quadratic relationship with angle θ in all cases.

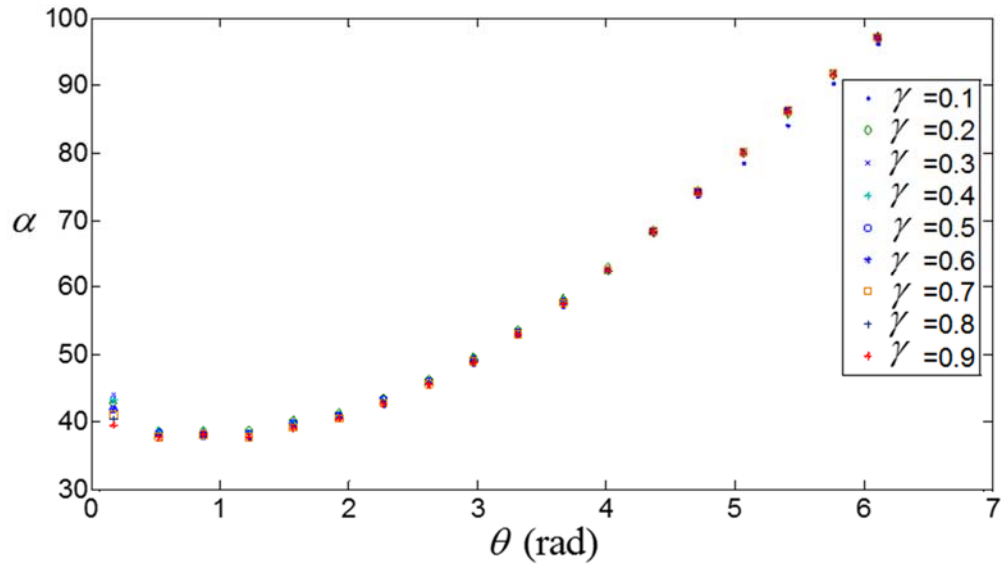


Figure 4-20. Edge effect relationship with angle for different γ in coding pattern $\{1\ 2\}$, $\rho = 100$.

For density of 100 in coding $\{1\ 2\}$, the edge effect is obtained as:

$$\alpha = 2.04\theta^2 - 2.60\theta + 39.41 \quad (35)$$

This equation is substituted in the equation (29) and calculated as theoretical number of found, N_t which is plotted versus the average simulated number of found, \bar{N}_s . This plot is shown in Figure 4-21 for decoding pattern $\{1\ 2\}$ and density 100 as an example, which illustrates the closeness of the two sets of data.

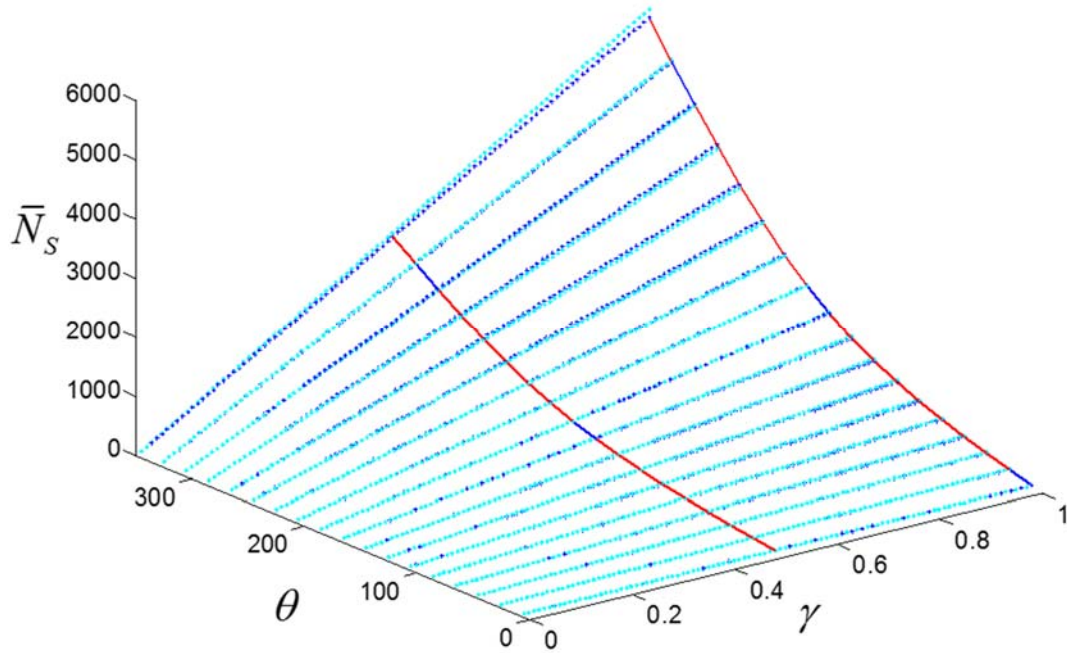


Figure 4-21. Theoretical and simulated results for $\rho = 100$, $\theta = 10-360$, $\gamma = 0.01-1$, decoding pattern $\{1\ 2\}$, Blue color represents the simulated \bar{N}_s and cyan color represents the \bar{N}_s obtained from theoretical model.

This comparison has been carried out for several decoding strategies and the results showed a very close consistency among theoretical and simulated results. Statistical analysis represented in the next section provides a demonstration of this claim.

4.3.2 Statistical analysis

The standard deviation, σ_s from m randomly generated images (for each coding strategy and density) is obtained by:

$$\sigma_s = \sqrt{\frac{\sum_{i=1}^m (N_{s_i} - \bar{N}_s)^2}{m-1}} \quad (36)$$

Where N_{s_i} is the simulated number of particles found for each generated image, and \bar{N}_s

$$\text{is the statistical mean, } \bar{N}_s = \sum_{i=1}^m \frac{N_{s_i}}{m} \quad (37)$$

The statistical normal variate, Z is calculated from

$$Z = \frac{N_t - \bar{N}_s}{\sigma_s} \quad (38)$$

Also, to check the correspondence of measured frequencies of the simulated results with the theoretical model, the chi-squared, χ^2 statistic is calculated. Chi-square is a statistical test usually implemented to compare observed data with expected data to test the null hypothesis that the two results are the same hypothesis. This can be obtained using the following equation

$$\chi^2 = \frac{(N_t - \bar{N}_s)^2}{\bar{N}_s} \quad (39)$$

The values of Z and χ^2 for decoding pattern {1 2} and density 50 is depicted in Figure 4-22.

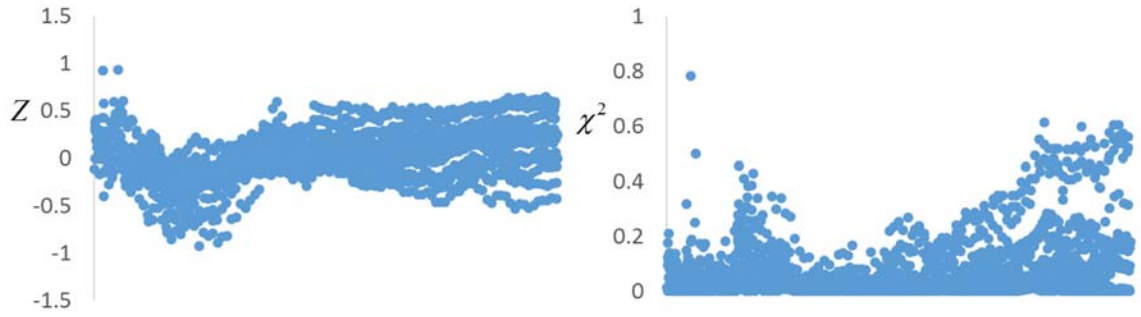


Figure 4-22. Normal variate, Z , and Chi-squared, χ^2 for $\rho = 50$ and decoding pattern $\{1\ 2\}$.

Similarly, the statistical Z and χ^2 for decoding pattern $\{1\ 2\}$, density of 100, and decoding pattern $\{1\ 3\}$, density of 100, are represented in Figure 4-23 and 22 respectively.

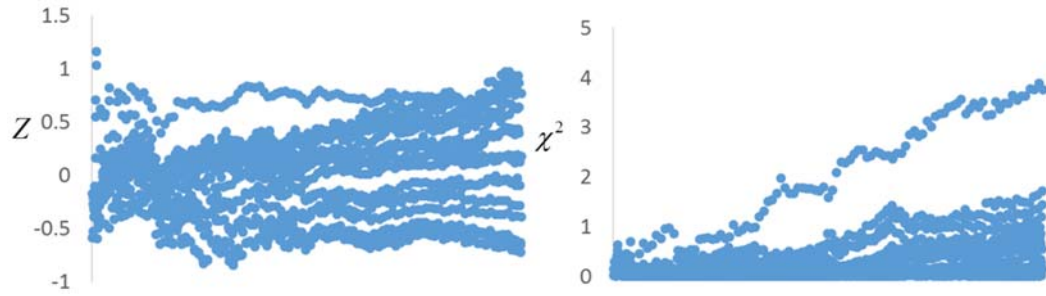


Figure 4-23. Normal variate, Z , and Chi-squared, χ^2 for $\rho = 100$ and decoding pattern $\{1\ 2\}$.

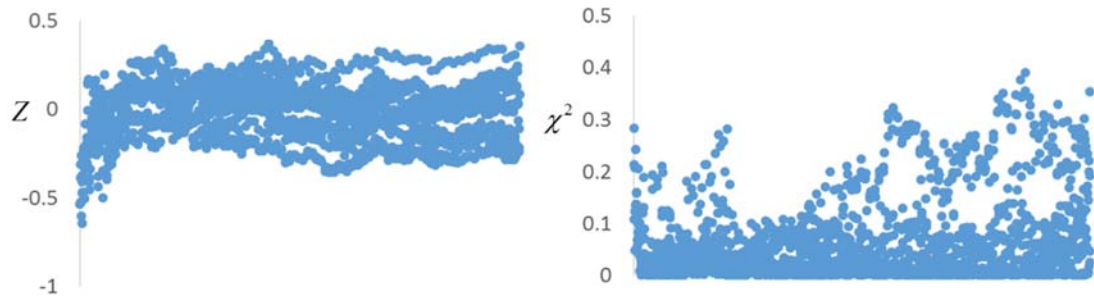


Figure 4-24. Normal variate, Z , and Chi-squared, χ^2 for $\rho = 100$ and decoding pattern $\{3\ 1\}$.

The values of Z are in the range of $[-1\ 1]$ which demonstrates an expected statistical variation. As it is shown in figures 17, 18 and 19, the χ^2 values is in the range of $[0\ 4]$. For the degree of freedom of 10 (i.e. 10 images) $\chi^2 = 3.94$ at a 95 % significance level indicating that, even for the largest variations shown, the null hypothesis would be accepted.

This demonstrates that theoretical and simulated results are consistent at a 95% confidence interval. Therefore, the developed mathematical model for the encoded-PTV technique is capable of predicting the number of randomly wrong patterns versus the key parameters, i.e. limiting angle, spacing tolerance, density and coding strategy.

4.3.3 Probability results versus velocity distribution

To see the probabilities of particle trajectories exiting the image border, both equations 12 and 19 are corresponding to all particles being of the same velocity and particles with uniformly distributed velocity are plotted as a function of the ratio h/T in Figure 4-25 and 24 respectively. For both of these plots when h is zero all the particles are inside the 100% region therefore the probability of finding a good pattern is 1. However, when h is equal to T the first particle point is at the image border. Consequently for a random direction of motion half of the particle are inside the image and the other half is outside. This means that the probability of finding (or missing) a set of particles as an identified encoded pattern is half. In other words when:

$$\begin{aligned} h \rightarrow 0 &\Rightarrow L \rightarrow 2\pi T \Rightarrow P_r \rightarrow 1 \\ h \rightarrow T &\Rightarrow L \rightarrow \pi T \Rightarrow P_r \rightarrow 0.5 \end{aligned}$$

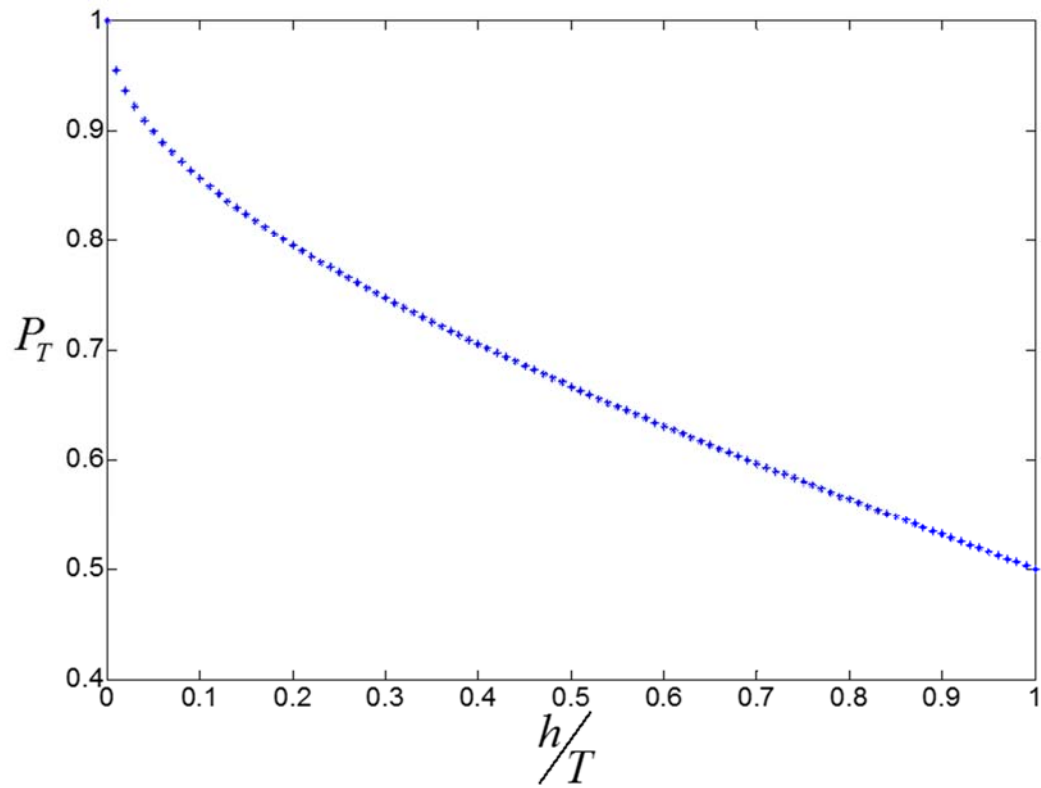


Figure 4-25. Image showing relationship of probability, P_T , of finding good pattern versus h/T when particles are all traveling with the same velocity V_{\max} whereby all particles travel distance T over the duration of a pulse train corresponding to a given coding strategy.

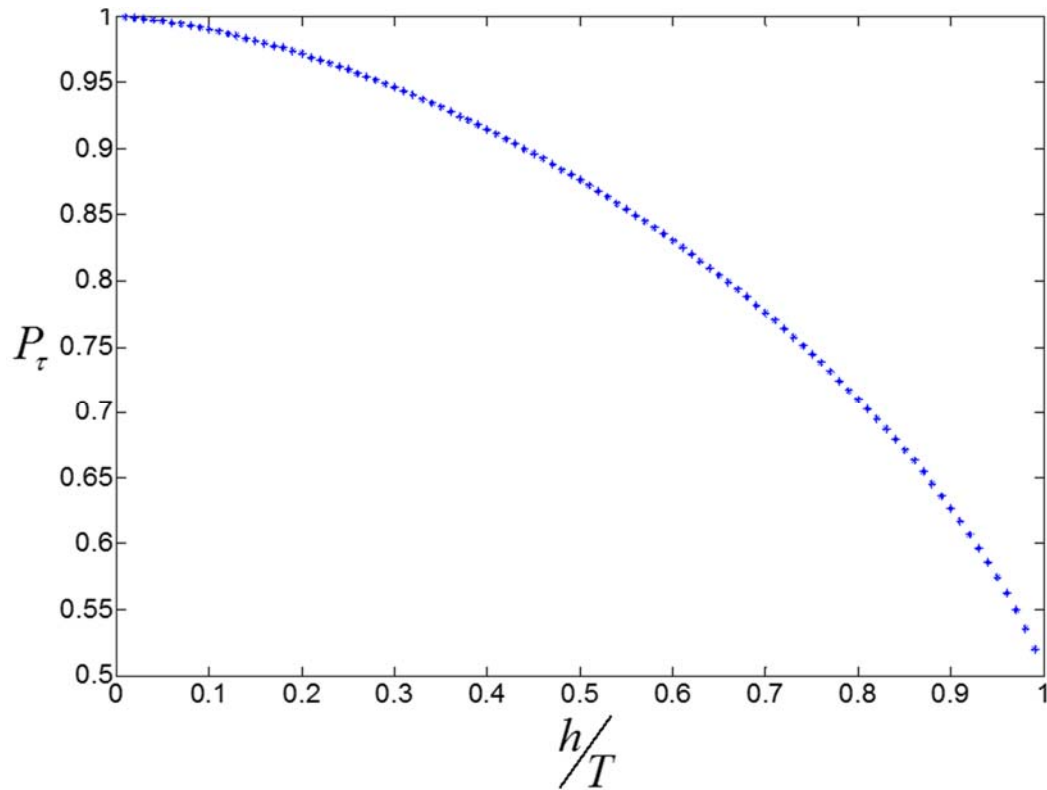


Figure 4-26. Image showing relationship of probability of finding good pattern P_τ versus h/T when particles are traveling with a uniformly distributed velocity between 0 and V during which particles travel distance $0 < \tau \leq T$.

For both distributions for which particles can be moving in a random direction and have a limited velocity, the probability of measuring a complete particle trajectory will be 100% if the initial image point is within or at the 100% region. Similarly this will drop to 5% for an initial particle image point at the image border. The major difference between the plots in Figure 4-25 and 24 is that, as expected, there is a larger overall probability of identifying particles for the distributed particle velocities. Other distributions could be analyzed using a similar approach. For a given size of image, the total percentage of

identified particles can be computed from the integral of probabilities over the complete image field divided by the area of the image.

4.3.4 Implementation of pulse coding in experimental trajectory tracking

Various experiments have been carried out with $\frac{25}{400}$ S·mL⁻¹ concentration solution with Grey Polyethylene Microspheres particles with 1.00g/cc and 125-150 μ m diameter (Cospheric). Images of recorded videos with different LED pulsing are shown below. Figure 4-27-a shows an image from double LED pulsing, meaning that LED goes on twice per camera exposure. As expected this results in particle pairs. Some particles pairs are indicated with ellipse in the image. For turning LED on and off three times for an exposure, particles show up three times in the image at three different positions. Blinking the LED 10 times for one exposure. This shows the paths that particles travel during that time. 20 LED pulsing per exposure is demonstrated in Figure 4-27-d. This almost shows the particle trajectories. However, for better illustration, the light timing needs to be adjusted with particle velocities.

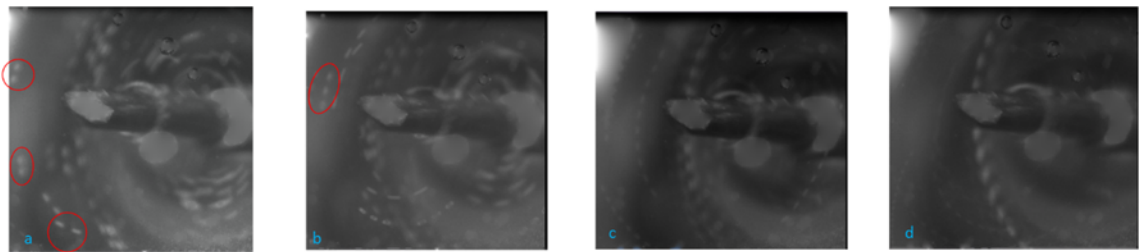


Figure 4-27. Flow fields image with 270 Hz rod frequency, 10.6 mm reservoir inner diameter, 30 ms LEDon, 30 ms LEDoff, 30ms camera off, 120-150 μ m diameter spheres, $\frac{25}{400}$ GmL⁻¹ concentration. a) 2 blinks, b) 3 blinks, c) 10 blinks, d) 20 blinks

Figure 4-28 shows the implementation of encoding technique in the experiments with pulses spacing spaced at {180 284}, {230 330}, and {132 180 232} with the numbers

corresponding to the off times in milliseconds. Ratio values for the encoding search can simply be obtained by dividing these timings by the first value of the pulse train. The particles appear in these sequences in the image frames with some of them identified by yellow circles for clarity. Particles used in these experiments are 10 μm diameter S-HGS, Silver Coated Hollow Glass Spheres from Dantec Dynamics. The Molybdenum rod is oscillating at its second mode at 1750 Hz. The distance between particles in each trajectory depends on the velocity of the particle and the LED off-time.

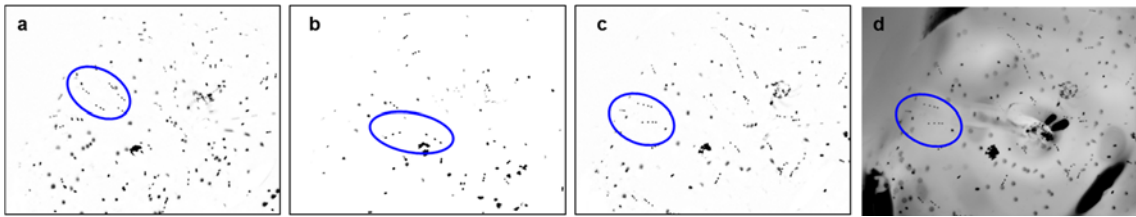


Figure 4-28. Flow fields image with 1750 Hz rod frequency, 15.8 mm reservoir inner diameter, 30 ms LED on, 50ms camera off, 10 μm diameter spheres, $1/400\text{GmL}^{-1}$ concentration. a) {180 284} encoding- filtered image, b) {230 330} encoding- filtered image, c) {132 180, 232} encoding- filtered image, d) {132 180 232} coding-original image. (The colors in the image are inverted for clarity)

Figure 4-29 and Figure 4-30 show the result of ePTV decoding algorithm on the experimental images for encoding patterns {180 284} and {132 180 232} respectively. In these images the blue stars show the centroids of each object (particles plus noise in this case) and the white line between the particles represent the identified trajectories in the specified coding sequence. Additional filtering has been applied to these images to remove the small and large objects that are clearly not particles and velocities higher than 8.6 mm/s have also been removed. In the image shown in Figure 4-29, after filtering, there are 151 objects. Given that 13 trajectories have been identified by the ePTV algorithm 112 of the particle are not considered part of the particle flow. This is, at least qualitatively, validated

by the apparent lack of sequential of the particles. Furthermore, the direction of the particle trajectories can be recognized using the encoding technique. For instance, in Figure 4-29, in each identified trajectory, the particle travel direction can be determined as the direction from the short to longer interval. Correspondingly, in Figure 4-30, there are total number of 127 objects where 10 trajectories have been identified. This indicates that almost 30% of the objects in this images are part of the identified coded patterns.

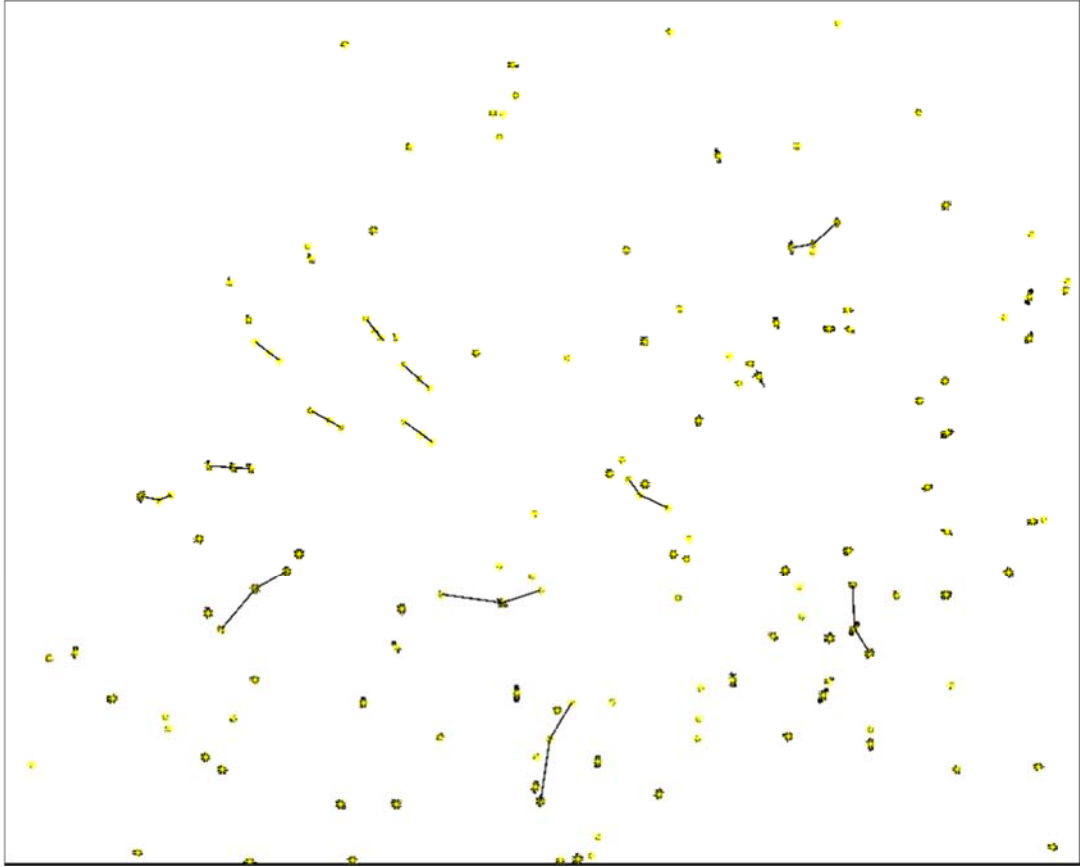


Figure 4-29. ePTV algorithm results for experimental image of encoding $\{180\ 284\}$ for $\theta = 40$ degrees and $\gamma = 0.2$.

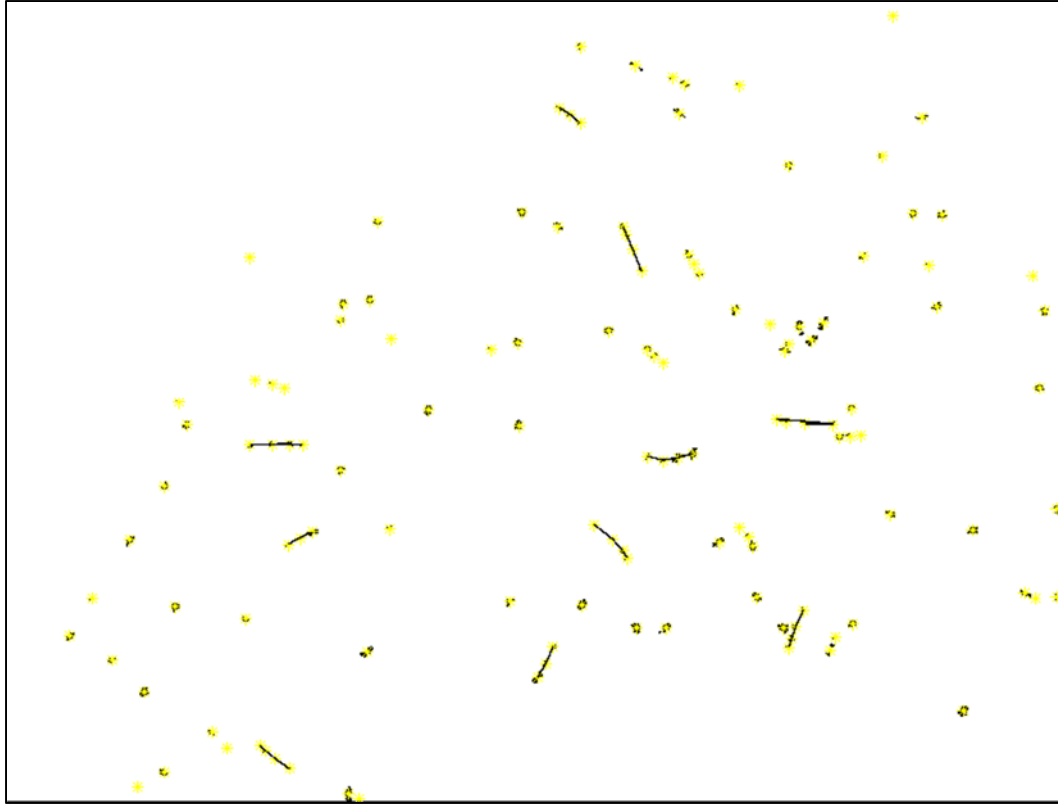


Figure 4-30. ePTV algorithm results for experimental image of encoding $\{132\ 180\ 232\}$ for $\theta = 40$ degrees and $\gamma = 0.3$.

One of the main advantages of this technique is the flexibility of it. Multiple parameters of the LED on and off time, distance between two subsequent particles experiments can be controlled and adjusted easily with the light frequency.

This chapter provides a theoretical framework for the evaluation of pulse encoded, or ePTV. Theoretical models for counting encoded particles are presented and an algorithm developed and used in a simulation study to evaluate the probability of identifying false particles in a noisy image field for any given encoding pattern. In particular, it is shown that even for images containing upwards of 100 spurious (or false) particles these can be efficiently eliminated as candidate particles for relatively uniform particle velocities and

accelerations. The theoretical model has been evaluated using Monte Carlo simulations for which the correspondence is observed at better than 95% confidence.

An additional approximate model has been derived to determine the loss of information due to encoding caused by the particle pattern exiting the field of view. Explicit equations for the probability of losing a particle as a function of the particles location in the image at the time of the initial pulse are presented for two velocity distributions. These two distributions correspond to a field of random particles of nominally the same velocity and a field of particles having a uniform velocity distribution up to a maximum value. These equations can be used to predict the number of lost particles for a given particle density.

As is generally the case for uniform pulse trains, the efficiency of encoded PTV as a noise rejection technique will be influenced by the turbulence and turbulent dimensions of the flow. Ultimately, the goal of this study is to determine the uncertainties of particle measurement. To do this it will be necessary to build upon this work and the work of [75–77] to produce a quantitative models for uncertainty estimates using ePTV.

CHAPTER 5: DYNAMIC ABSORBER BASED, MICRO-SCALE OSCILLATORY PROBES FOR MICRO-VORTEX EXPERIMENTS

This chapter presents two models for predicting the frequency response of micro-scale oscillatory probes. These probes are manufactured by attaching a thin fiber to the free end of one tine of a quartz tuning fork oscillator. In these studies, the attached fibers were either 75 μm diameter tungsten or 7 μm diameter carbon with lengths ranging from around 1 to 15 mm. The oscillators used in these studies were commercial 32.7 kHz quartz tuning forks. The first theoretical model considers lateral vibration of two beams serially connected and provides a characteristic equation from which the roots (eigenvalues) are extracted to determine the natural frequencies of the probe. A second, lumped model approximation is used to derive an approximate frequency response function for prediction of tine displacements as a function of a modal force excitation corresponding to the first mode of the tine in the absence of a fiber. These models are used to evaluate the effect of changes in both length and diameter of the attached fibers. Theoretical values of the natural frequencies of different modes show an asymptotic relationship with the length and a linear relationship with the diameter of the attached fiber. Similar results are observed from experiment, one with a tungsten probe having an initial fiber length of 14.11 mm incrementally etched down to 0.83 mm, and another tungsten probe of length 8.16 mm incrementally etched in diameter, in both cases using chronocoulometry to determine incremental volumetric material removal. The lumped model is used to provide a frequency response again reveals poles and zeros that are consistent with experimental measurements.

Finite element analysis (FEA) shows mode shapes similar to experimental microscope observations of the resonating carbon probes. This model provides a means of interpreting measured responses in terms of the relative motion of the tine and attached fibers. Of particular relevance is that, when a ‘zero’ is observed in the response of the tine, one mode of the fiber is matched to the tine frequency and is acting as an absorber. This represents an optimal condition for contact sensing and for transferring energy to the fiber for fluid mixing, touch sensing and surface modification applications.

5.1 Theory

In most of the micro-probe applications, a tuning fork is used as an oscillation mechanism to drive a fiber attached to it. This section presents two vibration models of the tuning fork-based probes: one as two Euler-beams serially connected, the other a simplified lumped model approximation. A 3D solid model of a 75 μm diameter tungsten fiber attached to a tuning fork tine with the important geometric parameters is shown in Figure 5-1.

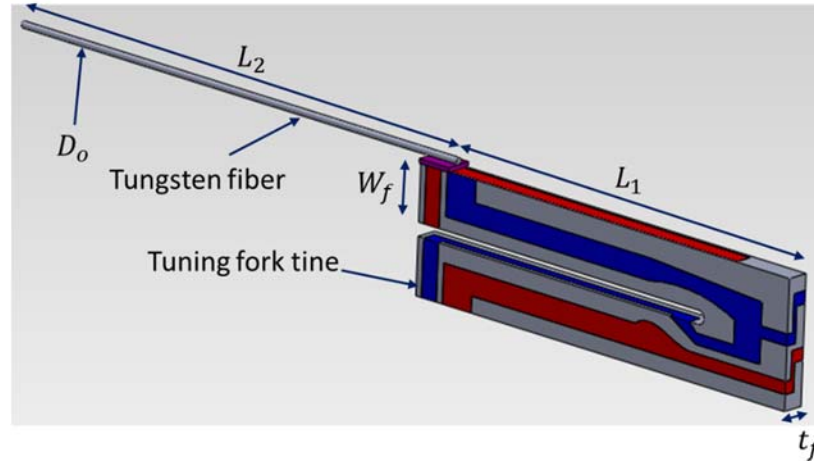


Figure 5-1. Model of tuning fork with a tungsten fiber attached to its upper tine showing the critical dimensional parameters used for vibration analysis of the probe system.

The blue and red regions in the model represents the electrodes to which an oscillating potential difference is applied.

5.1.1 Lateral vibration of two beams serially connected

Using the Euler beam equation[78], mode shapes for these two beams are given by

$$\frac{y_1(x_1, q_1)}{q_1} = A_1 \cos(\alpha_1 x_1) + A_2 \cosh(\alpha_1 x_1) + A_3 \sin(\alpha_1 x_1) + A_4 \sinh(\alpha_1 x_1) \quad (1)$$

$$\frac{y_2(x_2, q_2)}{q_2} = B_1 \cos(\alpha_2 x_2) + B_2 \cosh(\alpha_2 x_2) + B_3 \sin(\alpha_2 x_2) + B_4 \sinh(\alpha_2 x_2) \quad (2)$$

$$\alpha_1^4 = \frac{m_1}{L_1 E_1 I_1} \omega_s^2, \alpha_2^4 = \frac{m_2}{L_2 E_2 I_2} \omega_s^2$$

The parameters y_1 and y_2 are the lateral deflections of the tuning fork tine and fiber as a function of axial distances x_1 and x_2 measured from the fixed end, respectively. Additionally, parameters m , E , L and I represent the mass, elastic modulus, length and

second moment of area about the neutral axis of bending with subscripts 1 and 2 representing the tuning fork tine and the fiber, respectively. To determine mode shapes and natural frequencies, it is necessary that the above equations satisfy the eight boundary conditions (a-f)

$$\begin{aligned}
 a) \quad y_1 &= \frac{dy_1}{dx_1} \Big|_{x_1=0} = 0 \\
 b) \quad y_1 \Big|_{x_1=L_1} &= y_2 \Big|_{x_2=0} \\
 c) \quad \frac{dy_1}{dx_1} \Big|_{x_1=L_1} &= \frac{dy_2}{dx_2} \Big|_{x_2=0} \\
 d) \quad E_1 I_1 \frac{d^2 y_1}{dx_1^2} \Big|_{x_1=L_1} &= E_2 I_2 \frac{d^2 y_2}{dx_2^2} \Big|_{x_2=0} \\
 e) \quad E_1 I_1 \frac{d^3 y_1}{dx_1^3} \Big|_{x_1=L_1} &= E_2 I_2 \frac{d^3 y_2}{dx_2^3} \Big|_{x_2=0} \\
 f) \quad \frac{d^2 y_2}{dx_2^2} &= \frac{d^3 y_2}{dx_2^3} \Big|_{x_2=L_2} = 0
 \end{aligned} \tag{3}$$

From the first of these conditions, equations (1) and (2) can be rearranged in a simplified form

$$\frac{y_1(x_1, q_1)}{q_1} = A_1 (\cos(\alpha_1 x_1) - \cosh(\alpha_1 x_1)) + A_3 (\sin(\alpha_1 x_1) - \sinh(\alpha_1 x_1)) \tag{4}$$

$$\frac{y_2(x_2, q_2)}{q_2} = B_1 \cos(\alpha_2 x_2) + B_2 \cosh(\alpha_2 x_2) + B_3 \sin(\alpha_2 x_2) + B_4 \sinh(\alpha_2 x_2). \tag{5}$$

To solve for the natural frequency of this combined system it is necessary to express the α coefficients in terms of a common eigenvalue form

$$\omega_s^2 = (\alpha_{s1} L_1)^4 \frac{E_1 I_1}{\rho_1 A_1 L_1^4} = (\alpha_{s1} L_1)^4 \frac{E_1 I_1}{m_1 L_1^3} = (\alpha_{s2} L_2)^4 \frac{E_2 I_2}{m_2 L_2^3} \tag{6}$$

$$(\alpha_{s1}L_1)^4 = (\alpha_{s2}L_2)^4 \frac{m_1L_1^3}{m_2L_2^3} \frac{E_2I_2}{E_1I_1} = (\alpha_{s2}L_2)^4 \lambda\kappa = \beta^4 (\alpha_{s2}L_2)^4 \quad (7)$$

$$\gamma = \frac{L_2}{L_1}, \varphi = \alpha_2 L_2 \quad (8)$$

$$\alpha_{s1} = \beta \frac{L_2}{L_1}, \alpha_{s2} = \beta \gamma \alpha_{s2} . \quad (9)$$

The description of these parameters and their values in the model are given in Tables 2 and 3. In equation (6), the subscript S represents the modal frequency. It is important to note that the constant β is a function only of the materials and geometry of the fiber and tine. Appendix A details the derivation of the equations of this section.

Equations (A3) through (A7) from appendix A can be written in matrix form

$$\begin{bmatrix}
 \cos(\beta\alpha_2 L_2) - \cosh(\beta\alpha_2 L_2) & \sin(\beta\alpha_2 L_2) - \sinh(\beta\alpha_2 L_2) & -1 & 0 & 0 \\
 -\beta\gamma(\sin(\beta\alpha_2 L_2) + \sinh(\beta\alpha_2 L_2)) & \beta\gamma(\cos(\beta\alpha_2 L_2) - \cosh(\beta\alpha_2 L_2)) & 0 & -1 & -1 \\
 -E_1 I_1 (\beta\gamma)^2 (\cos(\beta\alpha_2 L_2) + \cosh(\beta\alpha_2 L_2)) & -E_1 I_1 (\beta\gamma)^2 (\sin(\beta\alpha_2 L_2) + \sinh(\beta\alpha_2 L_2)) & E_2 I_2 & 0 & 0 \\
 E_1 I_1 (\beta\gamma)^3 (\sin(\beta\alpha_2 L_2) - \sinh(\beta\alpha_2 L_2)) & -E_1 I_1 (\beta\gamma)^3 (\cos(\beta\alpha_2 L_2) + \cosh(\beta\alpha_2 L_2)) & 0 & E_2 I_2 & -E_2 I_2 \\
 0 & 0 & -\cos(\alpha_2 L_2) & \cosh(\alpha_2 L_2) & \sinh(\alpha_2 L_2) \\
 0 & 0 & \sin(\alpha_2 L_2) & \sinh(\alpha_2 L_2) & \cosh(\alpha_2 L_2)
 \end{bmatrix}
 \begin{Bmatrix}
 A_1 \\
 A_3 \\
 B_1 \\
 B_2 \\
 B_3 \\
 B_4
 \end{Bmatrix}
 = \begin{Bmatrix}
 A_1 \\
 A_3 \\
 B_1 \\
 B_2 \\
 B_3 \\
 B_4
 \end{Bmatrix}
 \quad (10)$$

Substituting $\psi_s = \alpha_{s2} L_2$ $s=1, 2, \dots$ equation (10) can be expressed as

$$\begin{bmatrix}
 \cos(\beta\psi_{s2}) - \cosh(\beta\psi_{s2}) & \sin(\beta\psi_{s2}) - \sinh(\beta\psi_{s2}) & -1 & 0 & 0 \\
 -\beta\gamma(\sin(\beta\psi_{s2}) + \sinh(\beta\psi_{s2})) & \beta\gamma(\cos(\beta\psi_{s2}) - \cosh(\beta\psi_{s2})) & 0 & -1 & -1 \\
 -(\beta\gamma)^2 (\cos(\beta\psi_{s2}) + \cosh(\beta\psi_{s2})) & -(\beta\gamma)^2 (\sin(\beta\psi_{s2}) + \sinh(\beta\psi_{s2})) & \kappa & 0 & 0 \\
 (\beta\gamma)^3 (\sin(\beta\psi_{s2}) - \sinh(\beta\psi_{s2})) & -(\beta\gamma)^3 (\cos(\beta\psi_{s2}) + \cosh(\beta\psi_{s2})) & 0 & \kappa & -\kappa \\
 0 & 0 & -\cos(\psi_{s2}) & \cosh(\psi_{s2}) & \sinh(\psi_{s2}) \\
 0 & 0 & \sin(\psi_{s2}) & \sinh(\psi_{s2}) & \cosh(\psi_{s2})
 \end{bmatrix}
 \begin{Bmatrix}
 A_1 \\
 A_3 \\
 B_1 \\
 B_2 \\
 B_3 \\
 B_4
 \end{Bmatrix}
 = \begin{Bmatrix}
 A_1 \\
 A_3 \\
 B_1 \\
 B_2 \\
 B_3 \\
 B_4
 \end{Bmatrix}
 \quad (11)$$

A Matlab™ program used to find the eigenvalues of this matrix for given κ, β and γ . The length of the fiber is varying by etch increment in the program, and the obtained eigenvalues (ψ_{s2}) of matrix in equation 11 are the natural frequencies of the system, representing different modal frequencies of the probe at each fiber length.

5.1.2 Lumped absorber model

Generally, the tuning fork tine is excited with an applied distortion characteristic of the first mode shape for lateral vibration in the plane of the two tines. Theoretically, an excitation force distributed along the tine corresponding to this mode shape will excite only modes with single node lateral deflection. Because the system to be modeled comprises the tuning fork tine with a relatively small fiber attached (in practice, with a small amount of epoxy adhesive) at the free end, to a first approximation, the tine can be modeled as a single degree of freedom system. Assuming that the attached fiber has little influence on the tine, it is reasonable to expect that the fiber will only have major significance on the tine response if the modes of the fiber correspondingly closely to that of the tine. Indeed, if the fiber attached to the tine has a coincident eigenvalue, it can act as an absorber. To understand the expected behavior of a fiber probe as a function of fiber dimensions (later evaluated using electrochemical etching), a lumped model of the complete probe system is shown in Figure 5-2 below.

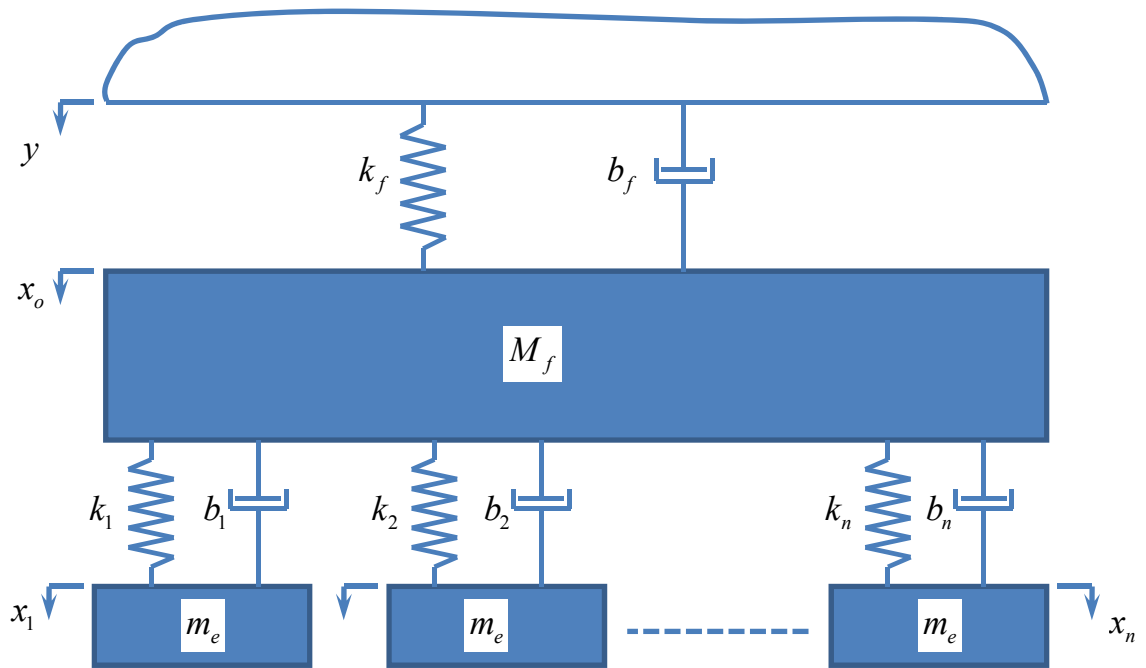


Figure 5-2. Lumped absorber model of the fiber probe.

The various parameters of this model are given in Table 5-1.

Table 5-1. Descriptions of mathematical parameters shown in Figure 5-2 above and their units.

Description	Symbols	Units
Tuning fork tine equivalent mass	M_f	kg
Tuning fork tine equivalent stiffness	k_f	$\text{N}\cdot\text{m}^{-1}$ or $\text{kg}\cdot\text{s}^{-2}$
Tuning fork tine equivalent damping	b_f	$\text{N}\cdot\text{s}\cdot\text{m}^{-1}$ or $\text{kg}\cdot\text{s}^{-1}$
Fiber modal mass	$m_i \quad i=1..n$	kg
Fiber modal stiffness	$k_i \quad i=1..n$	$\text{N}\cdot\text{m}^{-1}$ or $\text{kg}\cdot\text{s}^{-2}$
Fiber modal damping	$b_i \quad i=1..n$	$\text{N}\cdot\text{s}\cdot\text{m}^{-1}$ or $\text{kg}\cdot\text{s}^{-1}$
Tuning fork excitation	y	m
Number of modes used in lumped model	n	1

Generally, values for the equivalent stiffness and mass of the fiber elements are chosen to result in similar modal frequencies. Damping coefficients are more difficult to assess. For tuning fork tines in air, experience shows Q values ranging from a few hundred up to a few thousand. The fibers are near to ideal being essentially bonded to the ends of the tines and extending freely. Hence a low value for the intrinsic modal damping has been assumed. For simplification of computing, each mode is assumed to have a similar damping ratio for which free vibration exponential decay will be proportional to the modal frequency. (i.e. damping proportionate to stiffness in modal analysis models) [79].

For the tuning fork tine undergoing a first mode oscillation, Rayleigh's method produces a reasonable estimate of the natural frequency. For this method the Rayleigh's quotient can be derived by assuming a deflection shape corresponding to the deflection of

the beam due to a load at the free end. In this case, the stiffness at this free end, k_f , can be derived from the equation

$$k_f = \frac{3E_f I_f}{L_f^3} \quad (12)$$

$$I_f = \frac{w_f t_f^3}{12}.$$

All symbols and their values used for theoretical calculation are defined in Table 5-2. Additionally, it is known that the tuning fork in its free state (i.e. with no fiber attached) has a defined first mode frequency typically around 32 kHz (or 40 kHz in some cases). Hence, its representation as a single degree of freedom system can be represented by the lumped equation from which its free-state, undamped natural frequency, ω_f , can be expressed as

$$\omega_f^2 = \frac{k_f}{M_f} \quad (13)$$

For a given free state natural frequency the above equation can be rearranged to give an expression for the equivalent lumped mass. Based on these values, the damping coefficient of the system is readily given by

$$b_f = 2\xi_f \sqrt{k_f M_f} \quad (14)$$

Table 5-2. Description of tuning fork parameters used in equations to determine values for the parameters of the lumped model and serially connected beams analysis, and their values used for images represented in the result section.

<i>Description</i>	<i>Parameter</i>	<i>Value in model</i>	<i>Units</i>
Width of the tuning fork tine	W_f	0.0008	m
Thickness of the tuning fork tine	t_f	0.0004	m
length of the tuning fork tine	L_f	0.004	m
Quartz elastic modulus	E_f	75	GPa
Second moment of area of tuning fork	I_f	$\frac{1}{12}W_f t_f^3$	m ⁴
Damping ratio of tuning fork tine	ξ_f	0.001	(dimensionless)
Damping coefficient for tuning fork tine	b_f	$2\xi_f\sqrt{k_f M_f}$	N·s·m ⁻¹
Equivalent mass of tuning fork tine	M_f	$\frac{k_f}{\omega_f^2}$	kg
Tuning fork tine stiffness	k_f	$\frac{E_f W_f t_f^3}{4L_f^3}$	N·m ⁻¹
Density of tuning fork tine	ρ_f	2620	kg·m ⁻³

In all cases \mathcal{X} represents the modal coordinate for displacement at the free end. For the fiber, each mode is represented by a single degree of freedom system with energy storage and dissipation elements. Equating the elements of the individual modes with a discrete lumped mass it is reasonable to assume that a system in which at the extreme stiffness values will converge to a lumped system of comprising the tine plus total fiber mass. From a modal equation for the fiber as a cantilever independent of the tine (this will occur when the fiber is acting as a perfect absorber), it is found that the modal mass is

constant for each modal coordinate while the stiffness will increase in proportion to the eigenvalue. Based on these assumptions, the lumped parameters can be obtained from the eigenvalues given by

$$\lambda_i^2 = (\alpha_i L)^4 \frac{EI}{\rho AL^4} \quad (15)$$

Therefore,

$$k_i = (\alpha_i L)^4 \frac{EI}{nL^3}, \quad m_e = \rho AL / n, \quad b_i = 2\xi_i \sqrt{k_i m_e} \quad (16)$$

Two different fibers, i.e. carbon and tungsten are used in the experiments for comparison with the theoretical results (represented in results section 4). The description of two fiber parameters and their values used in theoretical calculation are given in Table 5-3 below.

Table 5-3. Description of carbon and tungsten fiber parameters used in equations to determine values for the parameters of the lumped model and serially connected beams analysis, and their values used for calculation represented in the result section.

<i>Description</i>	<i>Parameter</i>	<i>Value in model, Tungsten</i>	<i>Value in model, Carbon</i>	<i>Units</i>
Density of fiber	ρ	19250	2250	kg.m ⁻³
Damping ratio of fiber	ξ_i	0.000001	0.000001	(dimensionless)
Damping coefficient of fiber for i th mode	b_i	$2\xi_i\sqrt{k_i m_e}$	$2\xi_i\sqrt{k_i m_e}$	N·s·m ⁻¹
Fiber diameter	D_o	75	7	μ m
Equivalent mass of fiber	m_e	ρAL	ρAL	kg
Etched length of fiber	L	varying	varying	m
Area of fiber	A	$\pi \frac{D_o^2}{4}$	$\pi \frac{D_o^2}{4}$	m ²
Elastic modulus of fiber	E	411	235	GPa
Second moment of area of fiber	I	$\pi \frac{D_o^4}{64}$	$\pi \frac{D_o^4}{64}$	m ⁴

The root ($\alpha_i L$) are obtained from the boundary conditions of the beam. For a free cantilever, these roots for the first ten modes are given in Table 5-4 below. Due to the slow convergence of modal series and sensitivity to small variations in constants of subsequent simulations, the number of significant digits provided in this table is often necessary for modal analysis.

Table 5-4. The first ten roots for lateral natural frequencies of a free cantilever beam.

Frequency r	$(\alpha\mu_r l)$
1	1.87510406873
2	4.694091133
3	7.8547574382
4	10.995540742
5	14.137168391
6	17.278759532085
7	20.420352251041
8	23.5619449018064
9	26.7035375555183
10	29.84513020910288

Equations (12) through (16) provide the necessary relationships between the probe assembly and lumped model. The dissipation function, D and potential, U , and kinetic energy, T , for the lumped model in Figure 5-2 are given by

$$D = \frac{1}{2}b_f(\dot{x}_o - \dot{y})^2 + \frac{1}{2}\sum_{i=1}^n b_i(\dot{x}_i - \dot{x}_o)^2$$

$$U = \frac{1}{2}k_f(x_o - y)^2 + \frac{1}{2}\sum_{i=1}^n k_i(x_i - x_o)^2$$
(17)

$$T = \frac{1}{2}M_f\dot{x}_o^2 + \frac{m_e}{2}\sum_{i=1}^n \dot{x}_i^2.$$

Lagrange equation provided below is used to obtain the equations governing motion for the model. Therefore equations in (17) are substituted in the Lagrange equation [78–80] below

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} + \frac{\partial U}{\partial q_i} = 0 \quad (18)$$

Correspondingly, the equations governing motion of this system are given by

$$\begin{aligned} M_f \ddot{x}_o + \left(b_f + \sum_{i=1}^n b_i \right) \dot{x}_o + \left(k_f + \sum_{i=1}^n k_i \right) x_o + \sum_{i=1}^n k_i x_i - \sum_{i=1}^n b_i \dot{x}_i &= b_f \dot{y} + k_f y \\ [m_e \ddot{x}_i + b_i (\dot{x}_i - \dot{x}_o) + k_i (x_i - x_o) &= 0] \quad (i = 1, \dots, n). \end{aligned} \quad (19)$$

Figure 5-3 is a linear system diagram relating the output displacement of each coordinate for which y is the displacement generated by the input electrical voltage that causes a static deflection closely approximating the shape of the first mode of the tuning fork tine. x_o represents the displacement of the tuning fork tine, and x_i represents the displacement of the fiber in its i^{th} mode. The steady stated frequency responses of the individual coordinates of this system can be obtained using the assumed solutions

$$\begin{aligned} y &= Ae^{j\omega t} \\ x_o &= H_{oy}(j\omega)y \\ x_i &= H_{iy}(j\omega)y \end{aligned} \quad (20)$$

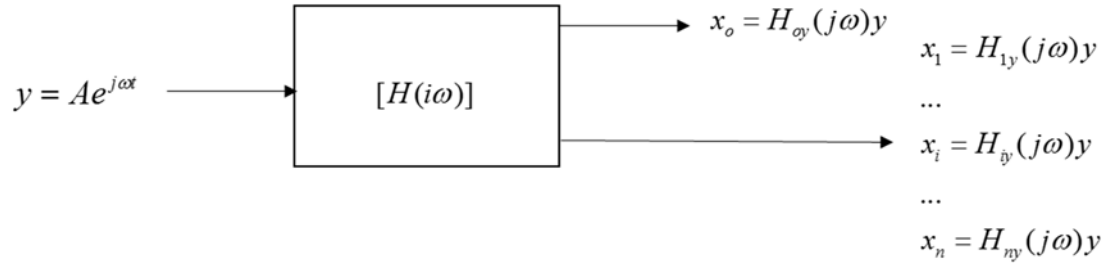


Figure 5-3. Input-output linear system model for the probe demonstration.

Substituting (20) into (19) and rearranging yield an expression for the steady state frequency response at coordinates x given by

$$H_{oy}(j\omega) = \frac{x_o}{y} = \frac{k_f + j\omega b_f}{-M_f \omega^2 + j\omega \left(b_f + \sum_{i=1}^n b_i \right) + \left(k_f + \sum_{i=1}^n k_i \right) - \sum_{i=1}^n \left\{ \frac{k_i + j\omega b_i \left(1 + j2\xi_i \frac{\omega}{\lambda_i} \right)}{\left(1 - \frac{\omega^2}{\lambda_i^2} \right) + j2\xi_i \frac{\omega}{\lambda_i}} \right\}} \quad (21)$$

$$H_{iy}(j\omega) = \frac{x_i}{y} = \left\{ \frac{k_i + j\omega b_i \left(1 + j2\xi_i \frac{\omega}{\lambda_i} \right)}{\left(1 - \frac{\omega^2}{\lambda_i^2} \right) + j2\xi_i \frac{\omega}{\lambda_i}} \right\} H_{oy}(j\omega) \quad (22)$$

A Matlab™ code is developed to compute these frequency response functions as the fiber length is shortened. In practice for a tuning fork based probe, it will be the tine displacement that is sensed and therefore will be measured as the frequency response of the probe, see results section 4.3.

5.2 Overview of experimental approach

Fiber probes are fabricated by manually aligning the fiber and tuning fork tine under an optical microscope using a multi-axis micrometer driven stage. Once aligned the tungsten and carbon fibers are attached by high strength epoxy (Norland optical adhesive #61). Figure 5-4 shows a 75 μm diameter tungsten fiber glued to the upper tine of a quartz tuning fork.

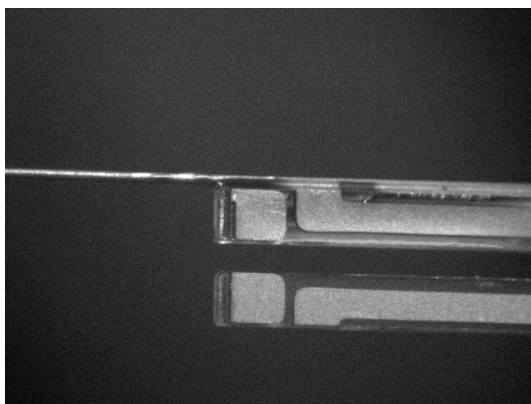


Figure 5-4. Photograph of 75 μm diameter tungsten fiber attached to the upper tine of tuning fork along with the tine axis.

An electrochemical etching apparatus is used for precise material removal for the purpose of controlled shortening of the length or diameter reduction of the fiber. The etching solution consisted of deionized water ($15.3 \text{ M}\Omega\cdot\text{cm}$ resistivity) with a 1 molar solution of potassium hydroxide (KOH). Mass was removed from the probes in small increments (5 to 10 μg) and measured from the electrode charge transfer during etching (using chronocoulometry). Figure 5-5 shows the etching experimental setup. A voltage applies across the carbon rods and the probe fiber. Current through this cell (i) is in series with a resistor (R_{ref}) that provides a voltage (V_R) that is amplified (by factor of G which generates V_O). This output voltage is then integrated to measure the amount of charge (q) from which mass removal is calculated. For the conditions of these experiments, 1 tungsten atom is removed for every 6 electrons through the cell. More details about the development of the etching instrument is presented in [81].

5.3.1 Varying length

Figure 5-6 illustrates the first ten modes of the carbon probe obtained from the theory for lateral vibration of two beams serially connected, section 2.1. In this figure the asymptotic loci of markers represented with blue dots show the modes of the probe which demonstrates the natural frequencies relationship with length of the fiber. The asymptotic lines represented with pink circles correspond closely to the natural frequencies for the carbon fiber modeled as a single cantilever beam. The consistency between these two lines indicates that the carbon probe is behaving similar to a cantilever beam.

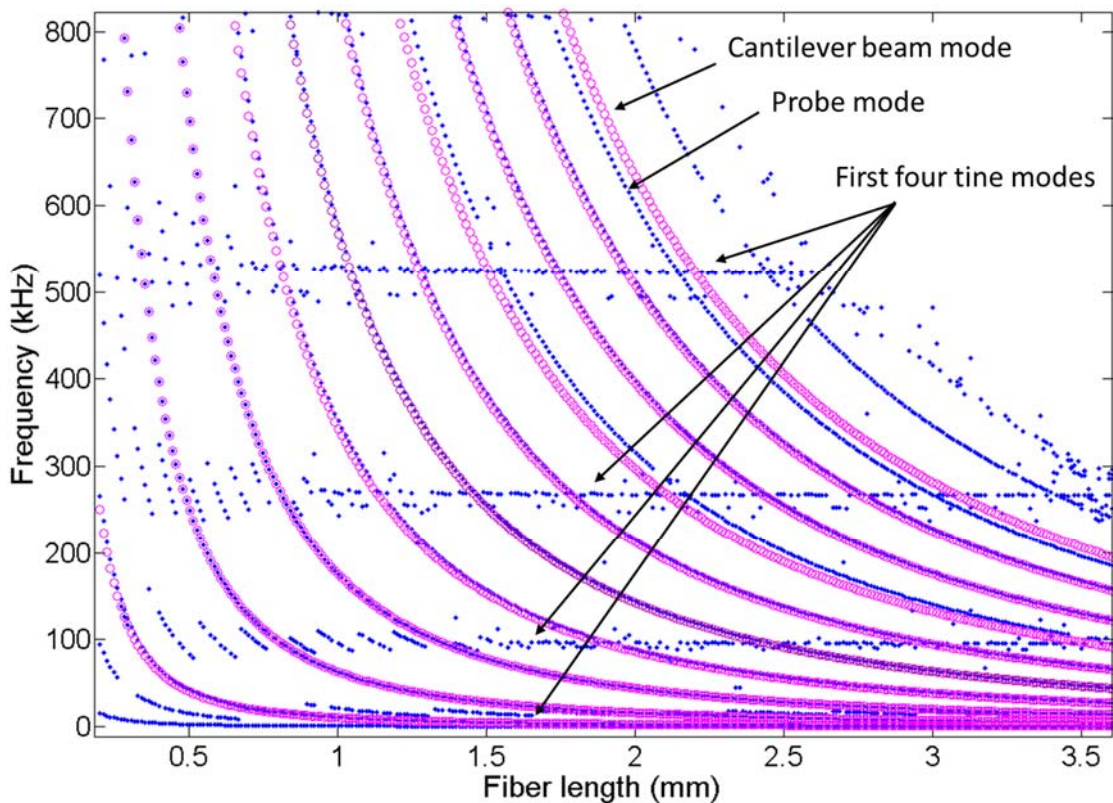


Figure 5-6. Theoretical calculation of frequency as a function of length of fiber for first ten resonant modes obtained from theoretical model for two serially connected beams.
(Carbon probe)

Oscillation of carbon fiber with 7 μm diameter attached to tuning fork shows a clear illustration of modal frequencies. The photograph of these observation (obtained

experimentally) for third, fourth, and fifth modes are shown in Figure 5-7 below as an illustration [82].

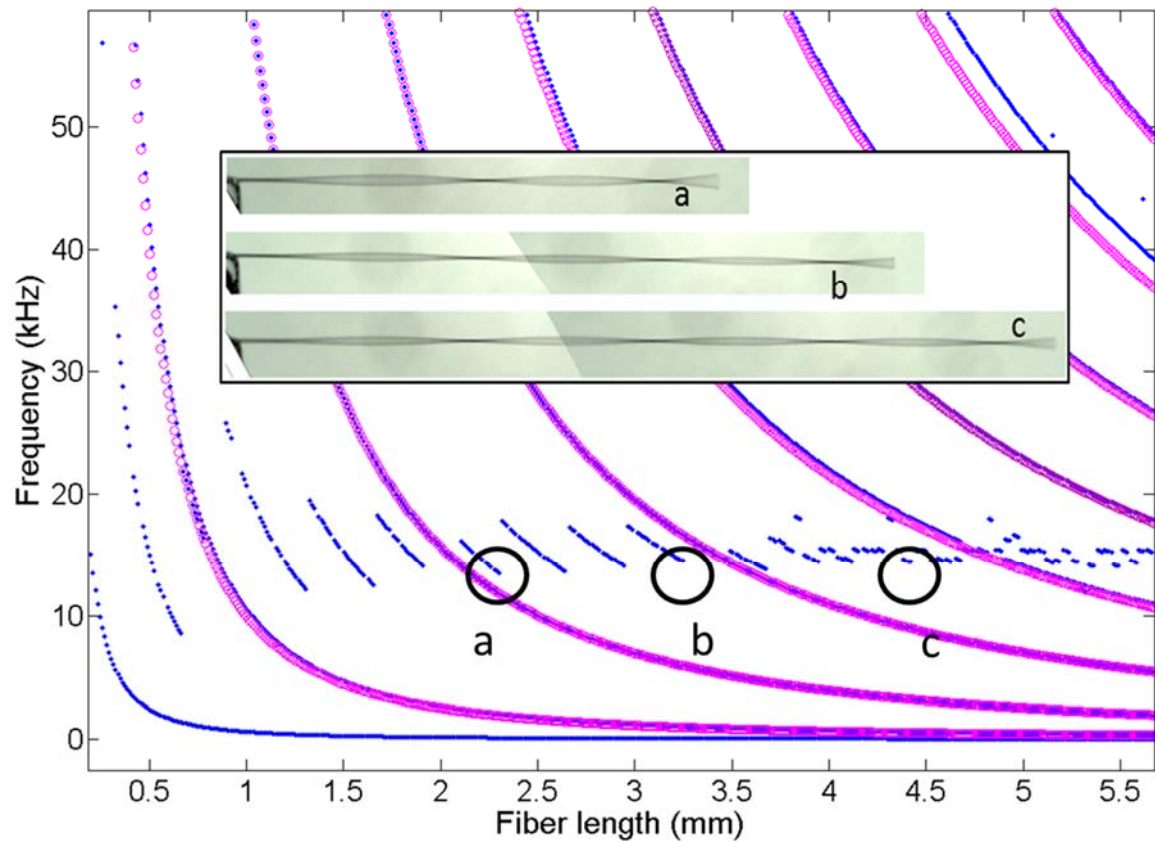


Figure 5-7. Three visible modes of carbon fiber with 7 μm diameter, a) third mode with length 2.4 mm, b) fourth mode with length 3.3 mm and c) fifth mode with length 4.3 mm. The theory model predicts these length when the natural frequencies of fiber collide with the tine frequency

Figure 5-8 shows the frequency-fiber length plot for the tungsten probe. In this figure, the difference between lines with blue dots and pink circles is more significant than the one depicted in Figure 5-6. It is speculated that since the tungsten fiber has larger diameter and is heavier than the carbon fiber, it therefore results in the probe deviating further from simple cantilever type behavior. Generally, for ‘touch’ sensitivity, the probe length is optimal when the fiber oscillates at maximum (or near to maximum) amplitude. This corresponds closely to the zeros in the frequency response plot. Not coincidentally,

these are also located in a region where the probe, and closely related independent fiber, frequencies coincide closely to that of the independent tine first mode resonance. It might be expected that the fiber will have a node at the attachment point at the tip of the tine at which point it will act as an absorber. Consequently, under these combined favorable conditions, the energy supplied to the tine will be maximally transferred to the fiber that is typically excited in a much higher mode. For a simple cantilever, the maximum total potential, V , and kinetic, T , energy associated with oscillation of the fiber is

$$\begin{aligned} V &= \frac{EI}{2l^3} \sum_{s=1}^{\infty} (\alpha \mu_s l)^4 q_s^2 \\ T &= \frac{\rho A l}{2} \sum_{s=1}^{\infty} \dot{q}_s^2 = \left(\frac{EI}{2l^3} \right) \sum_{s=1}^{\infty} (\alpha \mu_s l)^4 q_s^2. \end{aligned} \tag{23}$$

Where q_s represents the modal (or normal) coordinate associated with the S th mode. In fact, under these optimal conditions, the fiber may be considered to be an optimal absorber

for the time.

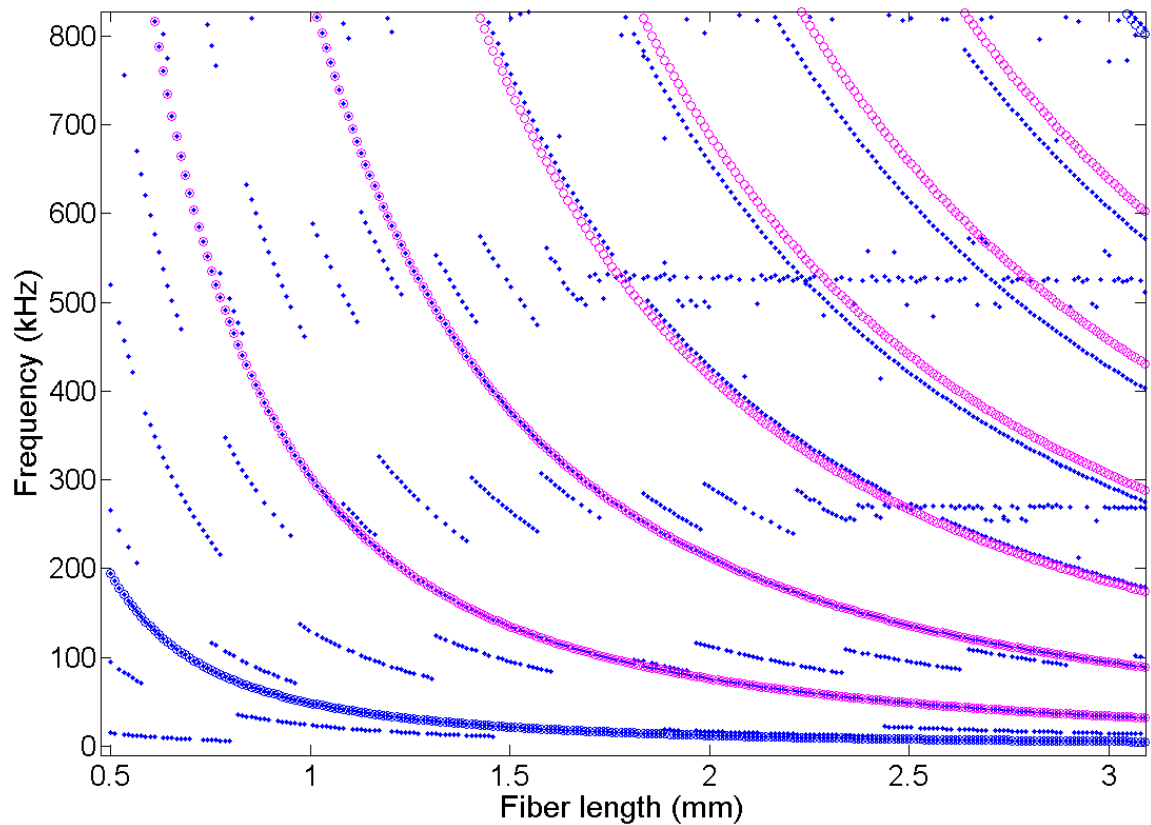


Figure 5-8. Theoretical calculation of frequency as a function of length of fiber for first ten resonant modes obtained from theoretical model for two serially connected beams. (Tungsten probe)

Also shown in Figure 5-6 and Figure 5-8 are eigenvalues of the matrix that, while exhibiting asymptotic behavior over short changes in length, these then jump to form a series of short segments that correspond to a nearly constant frequency. These lines correspond to the natural frequencies of the tuning fork tine in different modes, for which four of them are highlighted in Figure 5-6. For the longer fiber lengths, values for the first five frequencies have been extracted from these theoretical plots, and the ratio between them are compared with different cantilever beam types, see Table 5-5. The ratio of the modal frequencies shows that the tuning fork tine is behaving as beam with frequency

being between clamped-pinned and clamped-free conditions. One side of the tine is assumed to be clamped as it is stated in the boundary condition (equation 3). The other end of the tine will experience varying dynamic forces that will depend on the free natural frequencies of both beams. For high frequencies, forces on the tine due to fiber mass and the second moment of mass will be relatively large, effectively acting to a pin the ‘free’ end of the tine. For lower frequencies however, the effect of the fiber mass is less significant thus, the fiber forces at the end of the tine become less significant and it responds more closely to being free at the end.

Table 5-5. Natural frequencies of tine obtained from theoretical calculation demonstrating its behavior acting as clamped-clamped beam.

Mode (r)	Natural Frequencies, Simulation (kHz)	$\frac{\omega_r}{\omega_1} = \frac{\alpha_r^2}{\alpha_1^2}$ for clamped-pinned beam	Natural Frequencies, clamped-pinned beam (kHz)	$\frac{\omega_r}{\omega_1} = \frac{\alpha_r^2}{\alpha_1^2}$ for clamped-free beam	Natural Frequencies, clamped-free beam (kHz)
f_1	20	1.0	20	1.0	20
f_2	100	3.24	64.8	6.26	125.2
f_3	275	6.76	135.2	17.54	350.8
f_4	530	11.56	231.2	34.38	687.6
f_5	875	17.64	352.8	56.84	1136.8

Figure 5-9 shows the experimentally measured frequency response in terms of magnitude and frequency as a function of the length of the tungsten fiber. The initial length of the fiber used was 14.11 mm after which it was etched down to 0.83 mm in non-equal

steps (due to difficulties associated with controlling the immersed length of the fiber tip in the electrolyte solution). It can be noticed that the natural frequencies of the probe at different modes are changing asymptotically with respect to the length of the fiber similar to the observations obtained from theoretical graphs in Figure 5-6 and Figure 5-8. This will be further discussed in section 4.3.

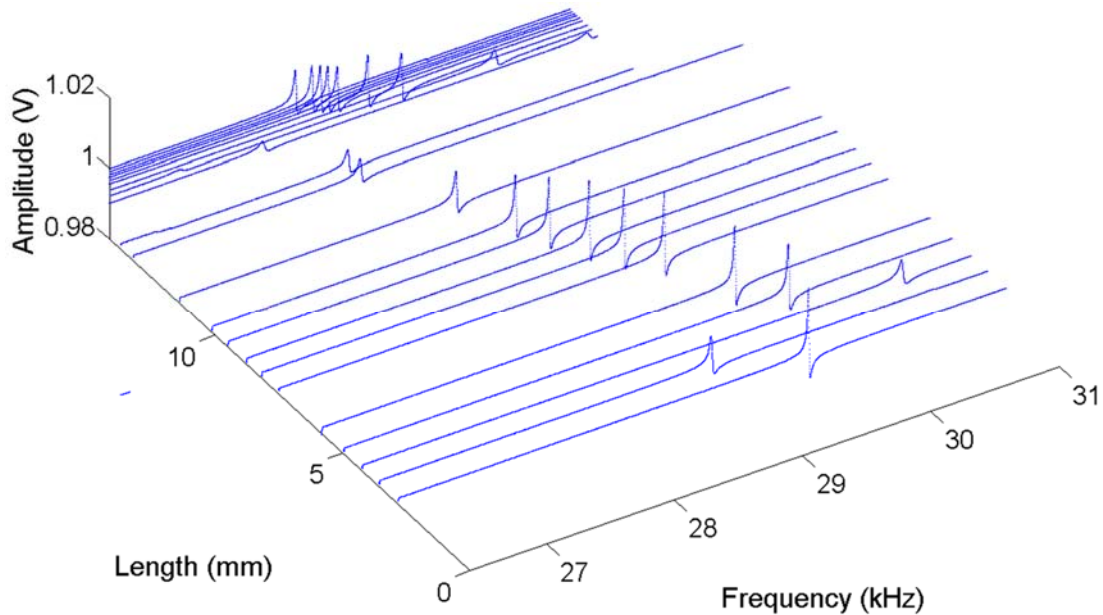


Figure 5-9. Experimentally measured frequency response of a tungsten probe sequentially etched to different lengths

5.3.2 Varying diameter

Another study has been carried out to examine the effect on probe's natural frequency of changing the fiber diameter. Figure 5-10 plots the theoretical eigenvalues for a tungsten fiber of length 8.16 mm attached to a tuning fork tine. Different colors in the graph show different modes of the probe, and demonstrate a linear relationship with varying fiber diameter.

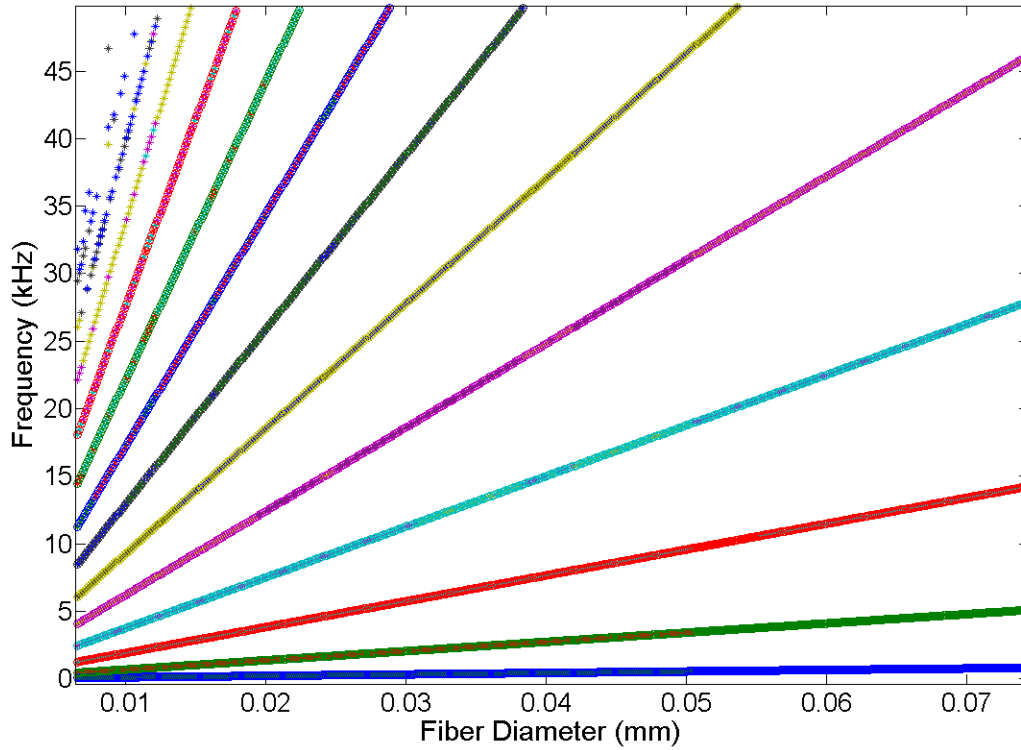


Figure 5-10. Theoretical calculation of eigenvalues for an 8.16 mm long Tungsten fiber as a function of varying diameter.

This can be predicted from the vibrating cantilever beam equations. The natural frequencies of a cantilever beam is obtained from

$$f_n = \frac{(\alpha_n l)^2}{2\pi} \sqrt{\frac{EI}{mL^3}}. \quad (24)$$

Substituting I and m for a cylindrical rod yields

$$f_n = \frac{(\alpha_n l)^2}{2\pi} \sqrt{\frac{E \left(\frac{\pi D^4}{64} \right)}{\left(\rho \frac{\pi D^2}{4} L \right) L^3}} = \frac{(\alpha_n l)^2 D}{8\pi L^2} \sqrt{\frac{E}{\rho}}. \quad (25)$$

Therefore, the natural frequency is expected to vary linearly with the diameter of the rod and hyperbolically with length. The result of the experimental test where a tungsten

probe (8.16 mm long fiber attached to tuning fork) is etched in step of 5 μg (decreasing diameter) is shown in Figure 5-11.

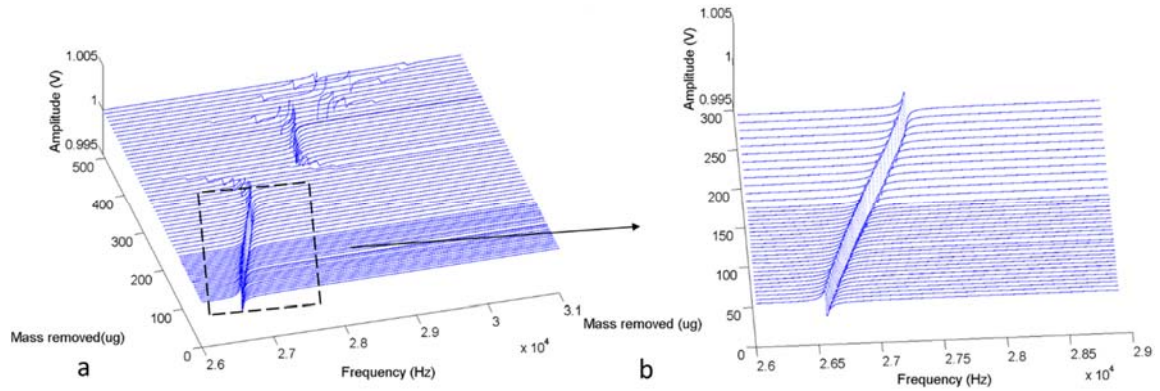


Figure 5-11. a) Frequency response of tungsten fiber obtained from experimental testing
b) subplot of 14-a demonstrating the linear relationship of modal frequencies with fiber diameter.

5.3.3 Lumped system model

Figure 5-12 represents the amplitude frequency response plot showing the magnitude of H_{oy} (equation 21) as a function of length of a tungsten fiber of radius 37.5 μm .

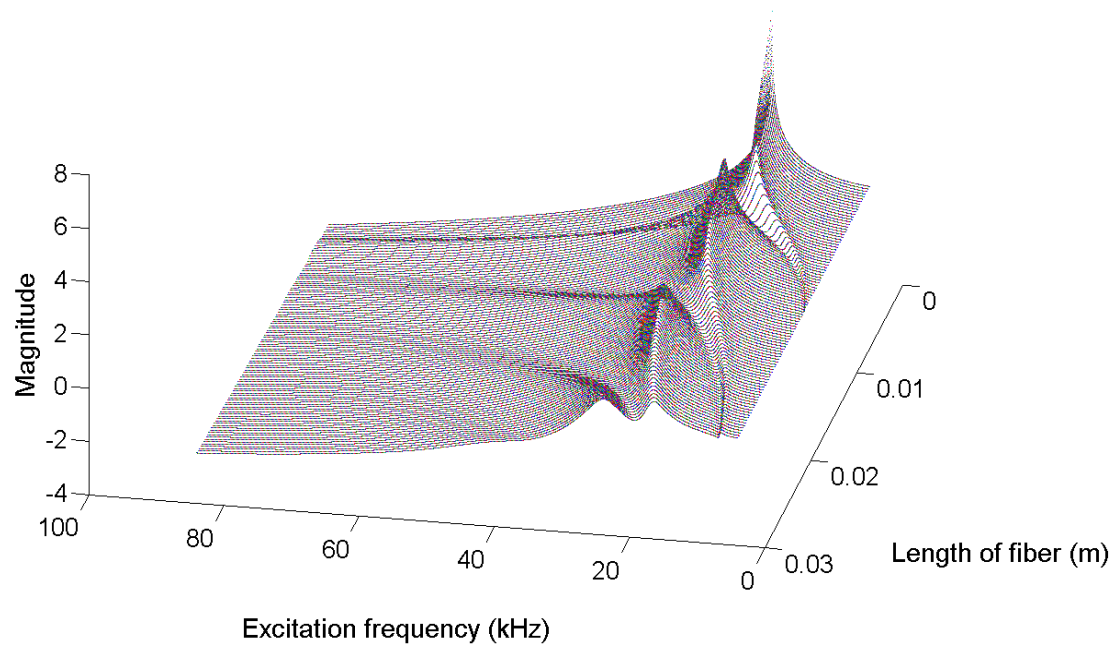


Figure 5-12. Amplitude frequency response of the tuning fork tine, H_{oy} , as a function of tungsten fiber length based on lumped model approximation (tungsten probe).

The phase response of H_{oy} as a function of frequency and diameter is shown in Figure 5-13.

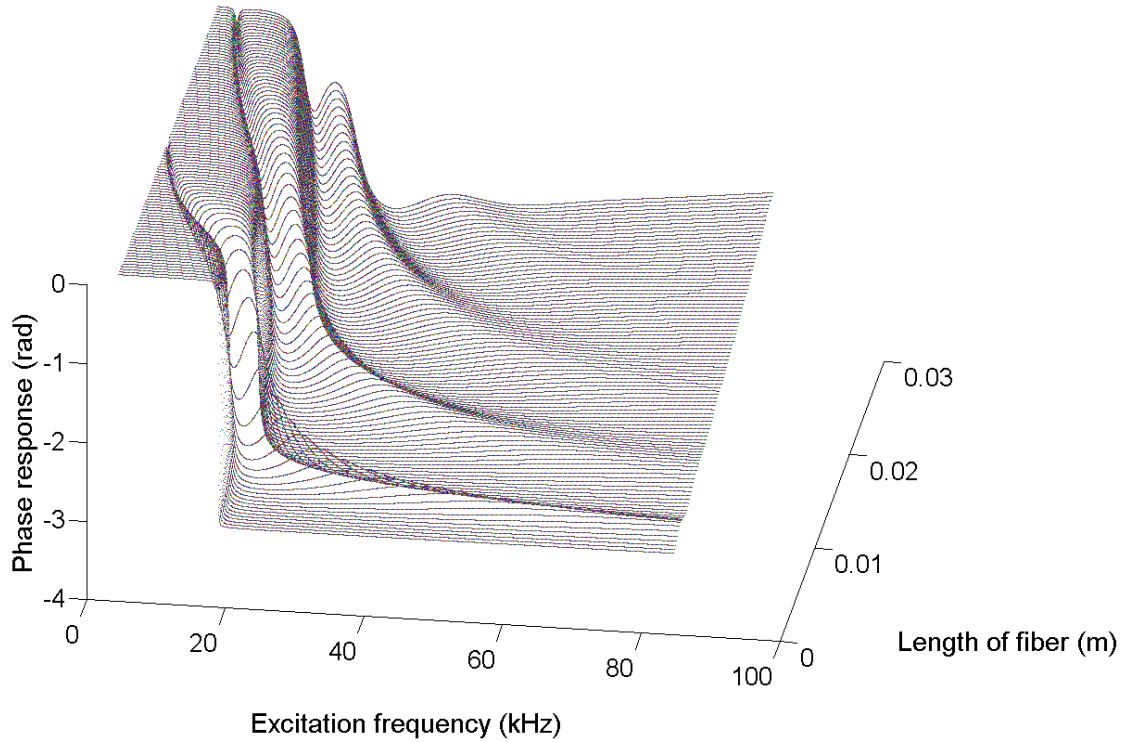


Figure 5-13. Phase frequency response of the tuning fork tine as a function of attached tungsten fiber length based on lumped model approximation.

Similarly, the amplitude frequency response of individual lumped modes H_{iy} (equation 22) are plotted relative to the length of the fiber. These plots for the first four modes of a tungsten probe are shown in Figure 5-14. The same asymptotic lines represented in Figure 5-6 and Figure 5-8 are noticeable in these images. Most significantly, and as expected from this lumped model approximation, the response for each fiber dominates while, because all coordinates of the system response share common roots, the influence of the surrounding system remains. However, it is apparent that the resonant peak for each fiber mode passes through the location of the zero in the tuning fork response.

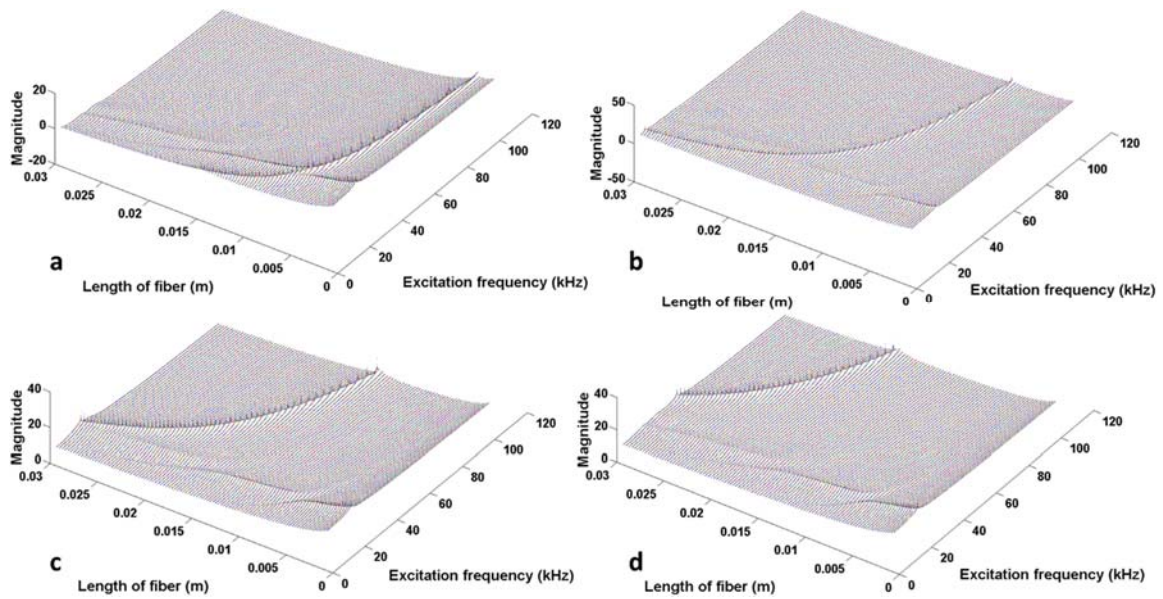


Figure 5-14. Amplitude frequency response of the fiber, H_{iy} for $i = 1, 2, 3$, and 4 respectively as a function of fiber length based on the lumped model approximation (tungsten probe).

5.3.4 Finite element analysis

A frequency study using FEA software available in SolidWorks™ has been carried out on a 3D solid model of the probes. Figure 5-15 shows the fifth mode shape of the oscillating carbon probe acquired from FEA, and its consistency with the experimental image.

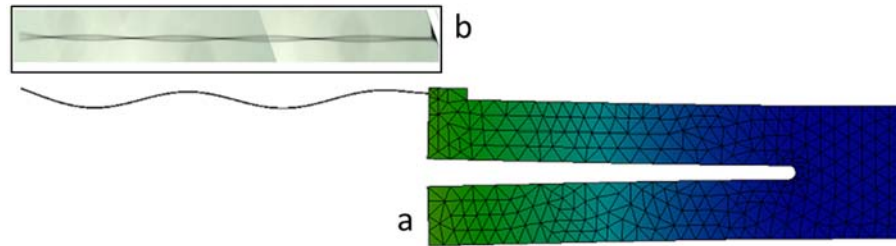


Figure 5-15. Fifth oscillating mode shape of carbon probe of 4.3 mm fiber length obtained from a) FEA, b) experimental testing.

The third and fourth mode shapes of tungsten probe with 5.5 mm fiber length obtained from an FEA study are shown in Figure 5-16.

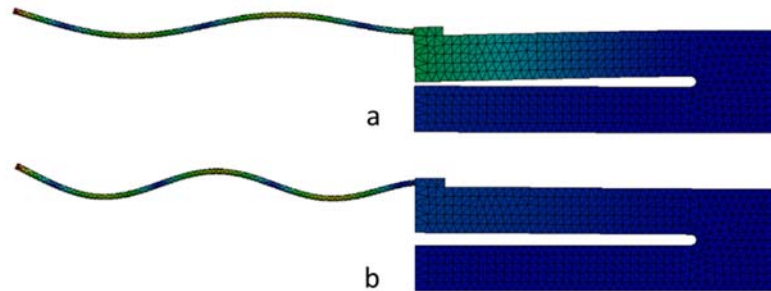


Figure 5-16. FEA result showing modes shapes of tungsten probe of 5.5 mm fiber length oscillating in a) third mode, and b) fourth mode.

The natural frequencies of first five modes for tungsten probe (fiber length=5.5 mm) acquired from theoretical calculation (two beams serially connected method) and FEA is represented in Table 5-6, which represent a good correlation between two approaches.

Table 5-6. First five natural frequencies of the tungsten probe (fiber length= 5.5 mm) obtained from theoretical calculation and FEA.

Mode (r)	Natural Frequencies, from theory (Hz)	Natural Frequencies from FEA (Hz)
f_1	1801.5	1580
f_2	11289.6	9709
f_3	31611.3	31417
f_4	61945.4	55768
f_5	101332.0	90607

It is noted that the theoretical model predicts frequencies that are higher than of those obtained from FEA. Two possible sources of this difference are mesh size used and the fact that the fiber bonds a short distance on to the end of the tine. Consequently, the additional mass due to the short length of fiber on the end of the tine is not included in the theoretical model. Such a reduction in mass would be expected to result in higher natural frequencies.

5.4 Multi-oscillator, micro-vortex experimental apparatus

In this section, the design and fabrication of an apparatus for multi-oscillator vortex studies is presented. The purpose of this is to study particle sorting and shepherding effects using vortices induced by oscillating micro-probes. Visualization and measurement of particle trajectories is achieved using microscope-based particle image velocimetry (PIV) [1]. To enable shepherding and transfer of particle clusters, multiple probes are placed into a fluid containing suspended particles. Some applications of this study includes the sorting

of particles in fluids based on their geometry, size and/or density, and separating cells in biological samples based on their cell type [2].

5.4.1 Design

The design and early investigation of micro-particle manipulation have been presented in [3, 4]. The setup represented in the previous chapters can only hold one oscillating probe for single probe experiments.

In this section a new apparatus is designed to provide flexibility in the number of implemented micro-probes and their arrangement as well as providing translation during the experiments. Figure 5-17 shows a schematic representation of the apparatus for this development.

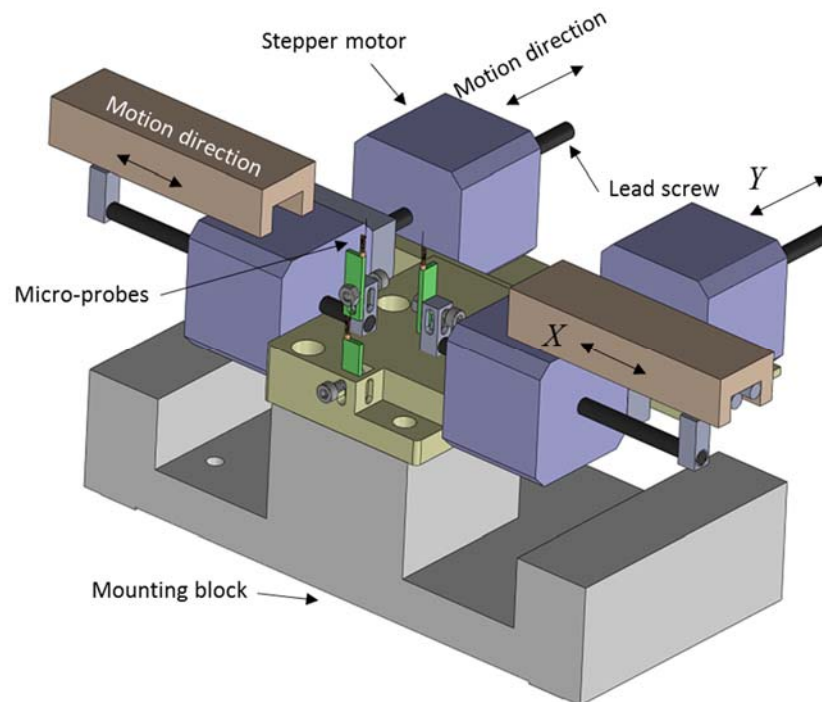


Figure 5-17. Solid model of multi-oscillator apparatus. Four stepper motors provide motion in X and Y for two moving probes.

The mounting block which bolt to the particle manipulation apparatus, holds the stepper motors and micro-probes. The lead screw via a holding mechanism (holding blocks, see Figure 5-18) holds the micro-probes. One probe is stationary while the other two probes can be translated along two perpendicular axes, X and Y . These motions are provided by two stepper motor and lead screw mechanisms, one for each axis. The moving probe's motions provide the position adjustment of the probes relative to each other. Similarly, this can provide the capability of moving the probes during the experiments, for instance two oscillating probes moving with different speeds inside the fluid. The holding blocks for the fibers are designed in such a way that provides maximum flexibility in probe orientation as well as the number of probes that they can support. As it is shown in Figure 5-18, the mounting block have four different orientation that probes can be attached to. The stationary probe is fixed to the apparatus so it has a fixed position and direction of oscillation. However, the stationary probe can also have four different orientations, based on the way it is attached to the apparatus, providing two different directions of oscillation. Moving probes can be attached to the holding blocks in a way that they oscillate either coplanar with the stationary probe or in a plane orthogonal to it with all probe axes being vertical. The moving probes that oscillate coplanar to the stationary probe's oscillation, are called in-plane moving probes. Similarly, probes oscillating perpendicular to the stationary probe's oscillation, are called out-of-plane moving probe. Besides providing 4 probe orientation options, each holding block can hold up to 4 micro-probes at the same time.

Figure 5-18 shows a close up view of the apparatus around the probes. In the photograph inset on the right hand side of this figure, these probes are made of a $7\text{ }\mu\text{m}$ diameter carbon fiber glued to the upper tine of a quartz tuning fork. A more comprehensive

study about the dynamic response of these resonant-based micro-probes are presented in [5]. These probes are driven at their resonant frequency using a Saji waveform generator (model STQ2) to transfer the maximum energy into the fluid to create vortex flows. The left hand side probe, is a moving probe in which its oscillating plane is the same as the stationary probe shown in the middle. The probe on the right hand side oscillates in the direction perpendicular to the other two probes, therefore it's a moving out-of-plane probe.

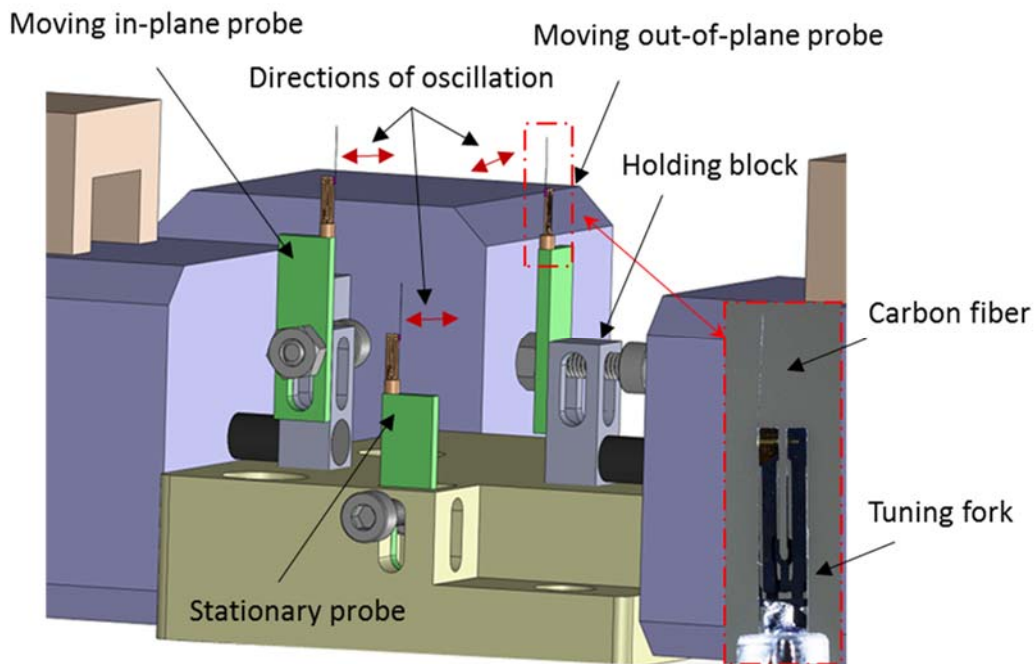


Figure 5-18. Close up view of the solid model for the micro-vortex multi-oscillator probing arrangement. Bottom right hand image illustrate the image of the micro probes containing a $7\text{ }\mu\text{m}$ diameter carbon fiber glued to a quartz tuning fork tine.

Figure 5-19 is a schematic diagram of the probes in a fluid represented by the circular reservoir. In this configuration one in-plane and one out-of-plane moving probe are used. The probes are inside the fluid with container boundary shown by the dashed line.

The moving in-plane and out-of-plane probes can scan within the container having (X_1, Y_1) and (X_2, Y_2) coordinates respectively.

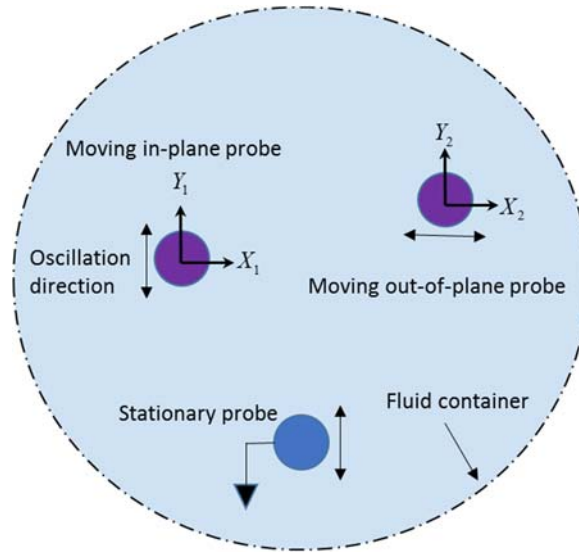


Figure 5-19. Schematic diagram of the probing system in the fluid container. The two upper moving probes are able to move anywhere inside the container along X and Y axes.

5.4.2 Experimental setup

Figure 5-20 shows the experimental apparatus of the designed system. Because the probes have to enter the surface of the fluid vertically from above, the mounting block of the multi-oscillator apparatus is bolted to the underside of the structural frame (Figure 5-20-a).

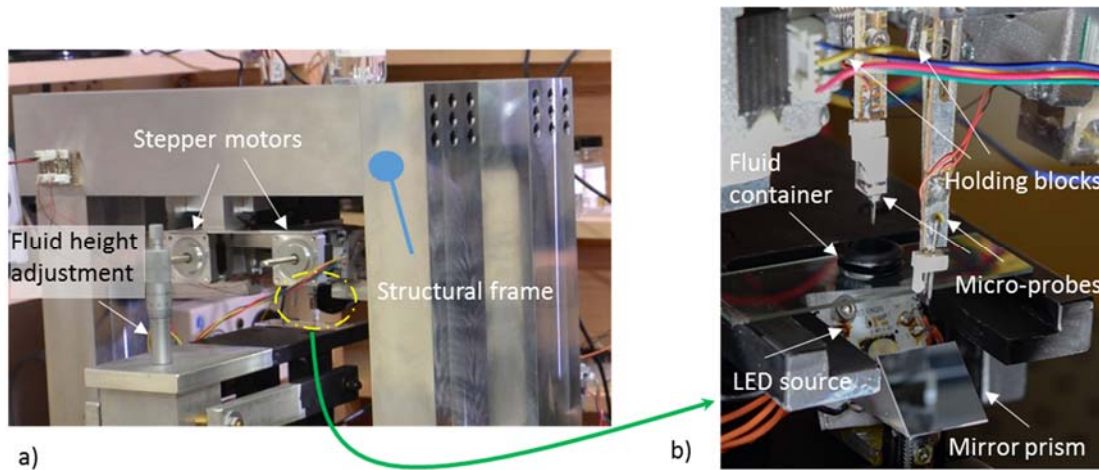


Figure 5-20. Experimental multi-oscillator apparatus. a) multi-oscillator apparatus attached to the structural frame of the particle manipulation framework [4,5]. b) Close shot of the micro-probes system.

The fluid height adjustment stage is used to control the depth of the probes into the fluid container. Typically this height is set by contacting the probes with the bottom surface of the container and then retracting them 10 – 100 μm so that they can freely oscillate. The camera images the probe and particles from the underside of this container. Figure 4 shows a photograph of the micro-probes system attached to the underside of the structural frame. The two stepper motors shown in Figure 5-20-a) are used to position the probes before and during the experiment to create different layouts (i.e. triangular or circular shapes). Figure 5-20-b) shows two probes, the lighting source and prism. One face of the prism deflects the bright light source vertically into the fluid container preventing image saturation and the other face provides a viewing mirror that reflects light scatter from the moving probes and particles back into the camera. In the photograph shown the fluid container is a cylindrical reservoir bonded to a glass slide. Depending on the probe arrangement and/or boundary condition(s) different size/types of reservoir are used.

CHAPTER 6: CONCLUSIONS

6.1 Summary

This chapter will outline the major achievements of this dissertation. The accomplishments are in three main areas, micro-particle manipulation, ePTV and micro probes and vortex studies.

6.1.1 Micro-particle manipulation

The outcome is the development of a laboratory platform providing the means to excite particle-laden flows using millimeter to micrometer-scale vibrating probes, to track the trajectories of individual particles optically, and to analyze the resulting trajectory data computationally. Applications of this research include sorting micro particles in fluids based on their geometry, size and/or density, separating cells in biological sample based on their cell type, or the study of abrasive particle dynamics in polishing and vortex machining. More generally, the developed facility provides a method for micrometer-scale, dynamic, flow visualization around solid surfaces.

6.1.2 ePTV

In this thesis a theoretical framework for the evaluation of pulse encoded particle tracking velocimetry, or ePTV is provided. Theoretical models for counting encoded particles are presented and an algorithm developed and used in a simulation study to evaluate the probability of identifying false particles in a noisy image field for any given encoding pattern. In particular, it is shown that even for images containing upwards of 100

spurious (or false) particles these can be efficiently eliminated as candidate particles for relatively uniform particle velocities and accelerations. The theoretical model has been evaluated using Monte Carlo simulations for which the correspondence is observed at better than 95% confidence.

An experimental platform to study particle manipulation using oscillating probes was fabricated. By using a suspension of micro-particles, subsequent imaging with encoded pulse trains provided snap-shots of the complex flow patterns. Typically, even after filtering, the images contain a density of around 100 to 200 particles from which encoded trajectories have been extracted and typically account for about 30% of the objects identified in the image.

An additional approximate model has been derived to determine the loss of information due to encoding caused by the particle pattern exiting the field of view. Explicit equations for the probability of losing a particle as a function of the particles location in the image at the time of the initial pulse are presented for two velocity distributions. These two distributions correspond to a field of random particles of nominally the same velocity and a field of particles having a uniform velocity distribution up to a maximum value. These equations can be used to predict the number of lost or falsely identified particles for a given particle density.

6.1.3 Micro probes and vortex studies

Micro-scale oscillatory probes are widely used in metrology, manufacturing and assembly for precision applications. Two beams serially connected, and lumped model approximate vibration models are developed in this thesis for detailed understanding of the dynamics of these probes.

Experiments have been carried out with tungsten and carbon probes where the fiber length or diameter were electrolytically etched and their frequency response acquired. The results from theoretical models and the experimental data, displayed natural frequencies of probes retain an asymptotic relationship with the fiber length and a linear relationship with the fiber diameter. Additionally, the tuning forks in these probes can be modelled with eigenvalues varying between clamped-pinned and clamped-free beams that depend on how closely the fiber resonance (a function of the fiber length or diameter) coincides with that of the tine. Moreover, a frequency study in SolidWorks™ predicted the probe mode shapes and its natural frequencies which were consistent with the data obtained from experiments, hence the theoretical model is adequate for predicting dynamic behavior and probe sensitivity thereby eliminating the necessity of complex FEA studies.

A key functional characteristic of these probes is the matching of the fiber modes to that of the first mode of the tine. Generally, the electrode pattern on the tuning fork tines are designed to excite the fundamental mode. Consequently, this excitation shape is not suited for exciting other probe modes. When the frequencies of fiber modes that are not closely matched to that of the tine, even though they may independently have a relatively high Q value, the tine excitation, and therefore the fiber, will have a low amplitude. The approaching and receding fiber modes for tungsten fiber lengths are visible at 3 mm and 13 mm. When the fiber length contains a mode close to the tine natural frequency, the energy of excitation is transferred into the fiber that then acts as an absorber for the tine mode resulting in a zero for the frequency response of the tine (i.e. the frequency response that is measured by these types of probe).

6.2 Future work

Continuation of work in these areas can implement some of the aides described below.

6.2.1 Micro-particle manipulation and PTV/PIV

Theoretical modeling of the micro-particle flow patterns and comparison of these to experimental studies [6] has been initiated over the last four years. A more systematic study to assess the influence of boundary conditions, particle geometry and density with multiple micro-fiber oscillators are planned. Furthermore, because of the large number of particles and noise in the camera images, the concept of novel encoded particle tracking velocimetry (ePTV) for extracting particle trajectories from these complex flow patterns is being explored and some preliminary studies have been published [72]

As is generally the case for uniform pulse trains, the efficiency of encoded PTV as a noise rejection technique will be influenced by the turbulence and turbulent dimensions of the flow. Ultimately, the goal of this study is to determine the uncertainties of particle measurement. To do this it will be necessary to build upon this work and the work of [75–77] to produce a quantitative models for uncertainty estimates using ePTV. Also, in the future, this work can be extended to 3 dimensional PIV/PTV system by adding two more cameras to the setup. This will provide the capability of obtaining 3-dimensional measurement of flow velocities in both in-plane and out-of-plane components of the flow fields.

6.2.2 Micro probes and vortex studies

While the focus of this study is the analysis of probes undergoing planar oscillations, in practice there are additional modes resulting in out-of-plane motion. Planar analysis is justified by the fact that the excitation force is applied only in this plane and that the results from both theory and experiment are consistent in terms of pole and zero locations in the probe response. Non-linear effects have been observed, more commonly occurring with high Q systems or those having measurable assembly errors such as misalignment between the axis of the fiber and that of the tine. Additionally, when these probes are used as contact sensors, the dynamics of the interaction is typically chaotic. Both of these topics, while interesting, are outside of the scope of this paper. However, there are applications for which multi-directional oscillators for contact sensing applications are of interest, particularly in the field of micrometer-scale touch-sensors for coordinate measuring machines [83–86]. It is expected that the technique for optimizing performance for the planar probe could be similarly determined for any resonant probe designs comprising slender elements attached to an oscillator operating at fundamental or higher modes.

While the probes of this study might oscillate in a single plane, the frequency response does change when the probe is brought into proximity with a surface having any orientation relative to the direction of oscillation. Studies to quantify this orientation dependence have yet to be undertaken. Additional studies are also necessary to determine the optimal parameters for probe designs in terms of subsequent signal sensitivity as a function of proximity to a specimen surface. Sensitivity is also complicated by the choice of excitation parameters (amplitude, frequency, and phase) in the region of the resonant

peak. Studies to quantify these parameters are planned, ultimately aimed at achieving optimal performance in terms of bandwidth and signal to noise ratio of micro-resonator probes.

REFERENCES

- [1] Tabeling P 2005 *Introduction to Microfluidics* vol 24(OUP Oxford)
- [2] Lutz B R, Chen J and Schwartz D T 2005 Microscopic steady streaming eddies created around short cylinders in a channel: Flow visualization and Stokes layer scaling *Phys. Fluids* **17** 23601
- [3] Lutz B R, Chen J and Schwartz D T 2006 Hydrodynamic tweezers: 1. Noncontact trapping of single cells using steady streaming microeddies. *Anal. Chem.* **78** 5429–35
- [4] Bongso T A, Sathananthan A H, Wong P C, Ratnam S S, Ng S C, Anandakumar C and Ganatra S 1989 Human fertilization by micro-injection of immotile spermatozoa. *Hum. Reprod.* **4** 175–9
- [5] Papathanasiou A 2005 Micromanipulation in Assisted Conception - A User's Manual and Troubleshooting Guide *Obstet. Gynaecol.* **7** 142–142
- [6] Howard S C, Chesna J W, Smith S T and Mullany B a. 2013 On the Development of an Experimental Testing Platform for the Vortex Machining Process *J. Manuf. Sci. Eng.* **135** 51005
- [7] Howard S C, Chesna J W, Mullany B A and Smith S T 2011 Subcomponent developments for a vortex machining test facility
- [8] Howard S C, Chesna J W, Mullany B and Smith S T 2012 Observations During Vortex Machining Process Development *ASME 2012 International Manufacturing Science and Engineering Conference* (ASME) p 25
- [9] Lutz B R, Chen J and Schwartz D T 2005 Microscopic steady streaming eddies created around short cylinders in a channel: Flow visualization and Stokes layer

scaling *Cit. Phys. Fluids* **17**

- [10] Sokoray-Varga B and Józsa J 2008 Particle tracking velocimetry (PTV) and its application to analyse free surface flows in laboratory scale models *Ř Period. Polytech. Civ. Eng.* **522** 63–71
- [11] Nowakowski B K, Smith S T, Mullany B A and Woody S C 2009 VORTEX MACHINING: LOCALIZED SURFACE MODIFICATION USING AN OSCILLATING FIBER PROBE *Mach. Sci. Technol.* **13** 561–70
- [12] Ashkin A 1970 Acceleration and Trapping of Particles by Radiation Pressure *Phys. Rev. Lett.* **24** 156–9
- [13] Ashkin A 2008 *Optical trapping and manipulation of neutral particles using lasers: a reprint volume with commentaires* (Hackensack NJ: World Scientific)
- [14] Molloy J E and Padgett M J 2002 Lights, action: optical tweezers *Contemp. Phys.* **43** 241–58
- [15] Chiou P Y, Ohta A T and Wu M C 2005 Massively parallel manipulation of single cells and microparticles using optical images. *Nature* **436** 370–2
- [16] Hertz H M 1995 Standing-wave acoustic trap for nonintrusive positioning of microparticles *J. Appl. Phys.* **78** 4845
- [17] Haake A, Neild A, Radziwill G and Dual J 2005 Positioning, displacement, and localization of cells using ultrasonic forces. *Biotechnol. Bioeng.* **92** 8–14
- [18] Hughes M P 2002 Strategies for dielectrophoretic separation in laboratory-on-a-chip systems. *Electrophoresis* **23** 2569–82
- [19] Nieuwenhuis J H, Jachimowicz A, Svasek P and Vellekoop M J 2005 Optimization of microfluidic particle sorters based on dielectrophoresis *IEEE Sens. J.* **5** 810–5

- [20] Di Carlo D, Wu L Y and Lee L P 2006 Dynamic single cell culture array. *Lab Chip* **6** 1445–9
- [21] Di Carlo D, Aghdam N and Lee L P 2006 Single-cell enzyme concentrations, kinetics, and inhibition analysis using high-density hydrodynamic cell isolation arrays. *Anal. Chem.* **78** 4925–30
- [22] Murtsovkin V A and Muller V M 1993 Inertial Hydrodynamic Effects in Electrophoresis of Particles in an Alternating Electric Field *J. Colloid Interface Sci.* **160** 338–46
- [23] Mao W and Alexeev A 2011 Hydrodynamic sorting of microparticles by size in ridged microchannels *Phys. Fluids* **23** 51704
- [24] Di Carlo D, Edd J F, Humphry K J, Stone H A and Toner M 2009 Particle Segregation and Dynamics in Confined Flows *Phys. Rev. Lett.* **102** 94503
- [25] Chun B and Ladd A J C 2006 Inertial migration of neutrally buoyant particles in a square duct: An investigation of multiple equilibrium positions *Phys. Fluids* **18** 31704
- [26] Nagai M, Oishi M, Sakaki N, Ducloux O, Oshima M, Asai H and Fujita H 2008 Application of vorticella's feeding mechanism as a micromixer 2008 *IEEE 21st International Conference on Micro Electro Mechanical Systems* (IEEE) pp 555–8
- [27] Williams D <https://youtu.be/YHb2JaujIPo>
- [28] Vogel S 2013 *Comparative biomechanics: life's physical world* (Princeton University Press)
- [29] Shukla R K and Eldredge J D 2007 An inviscid model for vortex shedding from a deforming body *Theor. Comput. Fluid Dyn.* **21** 343–68

- [30] SLEIGH M A 1984 The Integrated Activity of Cilia: Function and Coordination1
J. Protozool. **31** 16–21
- [31] da C. Andrade E N 1931 On the Circulations Caused by the Vibration of Air in a
Tube *Proc. R. Soc. London A Math. Phys. Eng. Sci.* **134**
- [32] Holtsmark J, Johnsen I, Sikkeland T and Skavlem S 1954 Boundary Layer Flow
Near a Cylindrical Obstacle in an Oscillating, Incompressible Fluid *J. Acoust. Soc.
Am.* **26** 26
- [33] Sritharan K, Strobl C J, Schneider M F, Wixforth A and Guttenberg Z 2006
Acoustic mixing at low Reynold's numbers *Appl. Phys. Lett.* **88** 54102
- [34] Yang Z, Matsumoto S, Goto H, Matsumoto M and Maeda R 2001 Ultrasonic
micromixer for microfluidic systems *Sensors Actuators A Phys.* **93** 266–72
- [35] Barry R. Lutz, Jian Chen and and Schwartz* D T 2006 Hydrodynamic Tweezers:
1. Noncontact Trapping of Single Cells Using Steady Streaming Microeddies
- [36] Lutz B R, Chen J and Schwartz D T 2003 Microfluidics without microfabrication.
Proc. Natl. Acad. Sci. U. S. A. **100** 4395–8
- [37] Zhou Z-G and Liu Z-W 2008 Biomimetic Cilia Based on MEMS Technology *J.
Bionic Eng.* **5** 358–65
- [38] Chong K, Kelly S D, Smith S and Eldredge J D 2013 Inertial particle trapping in
viscous streaming *Phys. Fluids* **25** 33602
- [39] de Bono J S, Scher H I, Montgomery R B, Parker C, Miller M C, Tissing H, Doyle
G V, Terstappen L W W M, Pienta K J and Raghavan D 2008 Circulating tumor
cells predict survival benefit from treatment in metastatic castration-resistant
prostate cancer. *Clin. Cancer Res.* **14** 6302–9

- [40] Chong K, Kelly S D, Smith S T and Eldredge J D 2016 Transport of inertial particles by viscous streaming in arrays of oscillating probes. *Phys. Rev. E* **93** 13109
- [41] Schlichting. H 1932 Berechnung ebener periodischer Grenzschichtströmungen **33** 327–335
- [42] Kafashi S, Eldredge J, Chong K, Thousand J, Kelly S and Smith S T 2014 Development of experimental facilities for investigations of microscopic mapping of fluid velocities *29th Annu. Proc. Am. Soc. Precis. Eng. ASPE2014*
- [43] Kafashi S, Eldredge J, Kelly S and Smith S T 2015 Microscopic-based imaging for precision measurement of micro scale flows around dynamically oscillating objects *30th Annu. Proc. Am. Soc. Precis. Eng. ASPE2015*
- [44] Willert C, Raffel M, Kompenhans J, Stasicki B and Kähler C 1996 Recent applications of particle image velocimetry in aerodynamic research *Flow Meas. Instrum.* **7** 247–56
- [45] Jahanmiri M 2012 *Particle Image Velocimetry: Fundamentals and Its Applications: Fundamentals and Modern Techniques of Particle Image Velocimetry for Fluid Dynamics Research* (LAP Lambert Academic Publishing)
- [46] Lindken R, Rossi M, Große S and Westerweel J 2009 Micro-Particle Image Velocimetry (μ PIV): Recent developments, applications, and guidelines *Lab Chip* **9** 2551
- [47] Charogiannis A, An J S and Markides C N 2015 A simultaneous planar laser-induced fluorescence, particle image velocimetry and particle tracking velocimetry technique for the investigation of thin liquid-film flows *Exp. Therm. Fluid Sci.* **68** 516–36

- [48] Winzen A, Roidl B, Klän S, Klaas M and Schröder W 2014 Particle-Image Velocimetry and Force Measurements of Leading-Edge Serrations on Owl-Based Wing Models *J. Bionic Eng.* **11** 423–38
- [49] Oguma Y, Yamagata T and Fujisawa N 2013 Measurement of sound source distribution around a circular cylinder in a uniform flow by combined particle image velocimetry and microphone technique *J. Wind Eng. Ind. Aerodyn.* **118** 1–11
- [50] Schröder A and Willert C E 2008 *Particle Image Velocimetry: New Developments and Recent Applications* (Springer Science & Business Media)
- [51] Kreizer M, Ratner D and Liberzon A 2010 Real-time image processing for particle tracking velocimetry *Exp. Fluids* **48** 105–10
- [52] Ponchaut N F, Mouton C A, Hornung H G and Dabiri D 2005 3D Particle Tracking Velocimetry Method: Advances and Error Analysis
- [53] Li D-X, Zhong Q, Yu M-Z and Wang X-K 2013 Large-scale particle tracking velocimetry with multi-channel CCD cameras *Int. J. Sediment Res.* **28** 103–10
- [54] Kitzhofer J and Brücker C 2010 Tomographic particle tracking velocimetry using telecentric imaging *Exp. Fluids* **49** 1307–24
- [55] Adrian R J 1991 Particle-Imaging Techniques for Experimental Fluid Mechanics *Annu. Rev. Fluid Mech.* **23** 261–304
- [56] Adrian R J 2005 Twenty years of particle image velocimetry *Exp. Fluids* **39** 159–69
- [57] Adrian R J 1986 Multi-point optical measurements of simultaneous vectors in unsteady flow—a review *Int. J. Heat Fluid Flow* **7** 127–45
- [58] Adrian R J 1986 Image shifting technique to resolve directional ambiguity in

double-pulsed velocimetry *Appl. Opt.* **25** 3855

- [59] Westerweel J, Elsinga G E and Adrian R J 2013 Particle Image Velocimetry for Complex and Turbulent Flows
- [60] Sader J E, Larson I, Mulvaney P and White L R 1995 Method for the calibration of atomic force microscope cantilevers *Rev. Sci. Instrum.* **66** 3789
- [61] Sader J E, Chon J W M and Mulvaney P 1999 Calibration of rectangular atomic force microscope cantilevers *Rev. Sci. Instrum.* **70** 3967
- [62] Karrai K and Grober R D 1995 Piezoelectric tip-sample distance control for near field optical microscopes *Appl. Phys. Lett.* **66** 1842
- [63] Brice J C 1985 Crystals for quartz resonators *Rev. Mod. Phys.* **57** 105–46
- [64] Torralba M, Hastings D J, Thousand J D, Nowakowski B K and Smith S T 2015 A three-fingered, touch-sensitive, metrological micro-robotic assembly tool *Meas. Sci. Technol.* **26** 125902
- [65] Bauza M B, Hocken R J, Smith S T and Woody S C 2005 Development of a virtual probe tip with an application to high aspect ratio microscale features *Rev. Sci. Instrum.* **76** 95112
- [66] Woody S, Nowakowski B, Bauza M and Smith S 2008 Standing wave probes for microassembly. *Rev. Sci. Instrum.* **79** 85107
- [67] Gabrielli C 1991 Calibration of the Electrochemical Quartz Crystal Microbalance *J. Electrochem. Soc.* **138** 2657
- [68] Qi H 2010 *High Speed Motion Generated by an Oscillating Microfiber*
- [69] Kafashi S, Eldredge J, Kelly S and Smith S T Microscopic-based imaging for precision measurement of micro scale flows around dynamically oscillating objects

- [70] Kafashi S, Eldredge J, Chong K, Thousand J, Kelly S and Smith S T 2014 Development of experimental facilities for investigations of microscopic mapping of fluid velocities *Proceedings - ASPE 2014 Annual Meeting* pp 302–6
- [71] Anon www.sajico.com
- [72] Kafashi S, Eldredge J, Kelly S and Smith S T Encode-PTV (ePTV) for indentification of particle trajectories in a noisy background (In prepration)
- [73] Lutz B R, Chen J and Schwartz D T 2005 Microscopic steady streaming eddies created around short cylinders in a channel: Flow visualization and Stokes layer scaling *Phys. Fluids* **17** 23601
- [74] Chong K 2013 Particle Manipulation in Viscous Streaming
- [75] Wieneke B 2015 PIV uncertainty quantification from correlation statistics *Meas. Sci. Technol.* **26** 74002
- [76] Sciacchitano A, Wieneke B and Scarano F 2013 PIV uncertainty quantification by image matching *Meas. Sci. Technol.* **24** 45302
- [77] Sciacchitano A, Neal D R, Smith B L, Warner S O, Vlachos P P, Wieneke B and Scarano F 2015 Collaborative framework for PIV uncertainty quantification: comparative assessment of methods *Meas. Sci. Technol.* **26** 74004
- [78] Bishop R E D and Johnson D C 1979 *The Mechanics of Vibration* (Cambridge University Press)
- [79] Rayleigh J W S B 1894 *The Theory of Sound, Volume I* (Macmillan)
- [80] Smith S T 2000 *Flexures: Elements of Elastic Mechanisms* (CRC Press)
- [81] Nowakowski B K, Smith S T, Pratt J R and Shaw G A 2012 Chronocoulometry for

quantitative control of mass removal in micro-structures and sensors. *Rev. Sci. Instrum.* **83** 105115

- [82] Hunt J B 2014 *Experimental Investigation of Process Parameters in Vortex Machining* (University of North Carolina at Charlotte)
- [83] Claverley J D and Leach R K 2013 Development of a three-dimensional vibrating tactile probe for miniature CMMs *Precis. Eng.* **37** 491–9
- [84] Claverley J D and Leach R K 2009 A vibrating micro-scale CMM probe for measuring high aspect ratio structures *Microsyst. Technol.* **16** 1507–12
- [85] Haitjema H, Pril W O and Schellekens P H J 2001 Development of a Silicon-based Nanoprobe System for 3-D Measurements *CIRP Ann. - Manuf. Technol.* **50** 365–8
- [86] Woody S C and Smith S T 2003 Resonance-based vector touch sensors *Precis. Eng.* **27** 221–33

APPENDIX A: EQUATIONS FOR GENERATING THE MATRIX GOVERNING
MOTION OF MICRO-SCALE OSCILLATORY PROBES

Dropping the S subscript, equations (4) and (5) can be expressed in the form

$$\frac{y_1(x_1, q_1)}{q_1} = A_1 (\cos(\beta\gamma\alpha_2 x_1) - \cosh(\beta\gamma\alpha_2 x_1)) + A_3 (\sin(\beta\gamma\alpha_2 x_1) - \sinh(\beta\gamma\alpha_2 x_1))$$

(A1)

$$\frac{y_2(x_2, q_2)}{q_2} = B_1 \cos(\alpha_2 x_2) + B_2 \cosh(\alpha_2 x_2) + B_3 \sin(\alpha_2 x_2) + B_4 \sinh(\alpha_2 x_2). \quad (\text{A2})$$

Satisfying the rest of the boundary conditions is not as straightforward. Consider each of the conditions in equation (3) in turn.

$$y_1(L_1) = A_1 (\cos(\beta\alpha_2 L_2) - \cosh(\beta\alpha_2 L_2)) + A_3 (\sin(\beta\alpha_2 L_2) - \sinh(\beta\alpha_2 L_2)) = y_2(0) = B_1 + B_2 \quad (\text{A3})$$

$$\begin{aligned} \frac{dy_1(L_1)}{dx_1} &= A_1 \beta\gamma\alpha_2 (-\sin(\beta\alpha_2 L_2) - \sinh(\beta\alpha_2 L_2)) + A_3 \beta\gamma\alpha_2 (\cos(\beta\alpha_2 L_2) - \cosh(\beta\alpha_2 L_2)) \\ &= \frac{dy_2(0)}{dx_2} = B_3 \alpha_2 + B_4 \alpha_2. \end{aligned}$$

(A4)

or

$$\begin{aligned} &A_1 \beta\gamma (-\sin(\beta\alpha_2 L_2) - \sinh(\beta\alpha_2 L_2)) + A_3 \beta\gamma (\cos(\beta\alpha_2 L_2) - \cosh(\beta\alpha_2 L_2)) \\ &= B_3 + B_4 \end{aligned}$$

$$\begin{aligned}
E_1 I_1 \frac{d^2 y_1}{dx_1^2} \Big|_{x_1=L_1} &= E_1 I_1 (\beta \gamma \alpha_2)^2 \left[-A_1 (\cos(\beta \alpha_2 L_2) + \cosh(\beta \alpha_2 L_2)) \dots \right. \\
&\quad \left. \dots - A_3 (\sin(\beta \alpha_2 L_2) + \sinh(\beta \alpha_2 L_2)) \right] \\
&= E_2 I_2 \frac{d^2 y_2}{dx_2^2} \Big|_{x_2=0} = E_2 I_2 \alpha_2^2 [-B_1 + B_2],
\end{aligned}$$

or

$$E_1 I_1 (\beta \gamma)^2 \left[-A_1 (\cos(\beta \alpha_2 L_2) + \cosh(\beta \alpha_2 L_2)) \dots \right. \\
\left. \dots - A_3 (\sin(\beta \alpha_2 L_2) + \sinh(\beta \alpha_2 L_2)) \right] = E_2 I_2 [B_2 - B_1].$$

(A5)

$$\begin{aligned}
E_1 I_1 \frac{d^3 y_1}{dx_1^3} \Big|_{x_1=L_1} &= E_1 I_1 (\beta \gamma \alpha_2)^3 \left[A_1 (\sin(\beta \alpha_2 L_2) - \sinh(\beta \alpha_2 L_2)) \dots \right. \\
&\quad \left. \dots A_3 (-\cos(\beta \alpha_2 L_2) - \cosh(\beta \alpha_2 L_2)) \right] \\
&= E_2 I_2 \frac{d^3 y_2}{dx_2^3} \Big|_{x_2=0} = E_2 I_2 \alpha_2^3 [-B_3 + B_4],
\end{aligned}$$

or

$$E_1 I_1 (\beta \gamma)^3 \left[A_1 (\sin(\beta \alpha_2 L_2) - \sinh(\beta \alpha_2 L_2)) \dots \right. \\
\left. \dots A_3 (-\cos(\beta \alpha_2 L_2) - \cosh(\beta \alpha_2 L_2)) \right] = E_2 I_2 [-B_3 + B_4].$$

(A6)

$$\frac{d^2 y_2(L_2)}{dx_2^2} = -B_1 \alpha_2^2 \cos(\alpha_2 L_2) + B_2 \alpha_2^2 \cosh(\alpha_2 L_2) - B_3 \alpha_2^2 \sin(\alpha_2 L_2) + B_4 \alpha_2^2 \sinh(\alpha_2 L_2) = 0$$

$$\frac{d^3 y_2(L_2)}{dx_2^3} = B_1 \alpha_2^3 \sin(\alpha_2 L_2) + B_2 \alpha_2^3 \sinh(\alpha_2 L_2) - B_3 \alpha_2^3 \cos(\alpha_2 L_2) + B_4 \alpha_2^3 \cosh(\alpha_2 L_2) = 0,$$

or

$$-B_1 \cos(\alpha_2 L_2) + B_2 \cosh(\alpha_2 L_2) - B_3 \sin(\alpha_2 L_2) + B_4 \sinh(\alpha_2 L_2) = 0$$

$$B_1 \sin(\alpha_2 L_2) + B_2 \sinh(\alpha_2 L_2) - B_3 \cos(\alpha_2 L_2) + B_4 \cosh(\alpha_2 L_2) = 0.$$

(A7)

Equations A3 to A7 are combined and written in a matrix form which represents the matrix governing motion of the system. The eigenvalues of this matrix are the natural frequencies of the probe.

APPENDIX B: CODES AND PROGRAMMS

This appendix contains programs that have been generated for the purpose of computing algorithms and hardware interfacing codes used in this these. These 26 programs and their purpose are listed in Table B-1.

Table B-1. Programs' name and their purpose used in this thesis.

Program	Purpose
Wavegenerate.h	Library header program for controlling the waveform generator output signal.
Wavegenerate.cpp	C++ library of functions used by the wavegenerate program.
JoystickThreeMotor.ino	Arduino program for controlling camera 3-axis stages with variable speed adjustment
LEDPulseSerialParametrized.ino	Arduino program for controlling LED on/off time, camera off time, and number of blinks for ePTV application
LEDShutterFrameRateSajad.ino	Arduino program for synchronizing LED timing with camera shutter speed
JoystickFourMotor.ino	Arduino program for controlling automatic micro-fiber apparatus

COModelDeltaLwithDampingBit ByBitHiy.m	Program for obtaining amplitude frequency response of the fiber, H_{iy} for different modes ($i = 1, 2, 3$, and 4) as a function of fiber length based on the lumped model approximation (tungsten probe)
COModelDeltaLwithDampingBit ByBit.m	Program for obtaining amplitude and phase frequency response of the tuning fork tine, H_{oy} as a function of attached (tungsten) fiber length based on lumped model approximation
PsiDeterminant.m	Matlab function program to find the eigenvalues of this matrix governing motion using two beams serially connected method. (Used in running the three below codes)
TungstenFiberVarying Length.m	Program for obtaining the frequency responses of micro-fibers versus fiber varying length using two beams serially connected method. (Tungsten probe)

Tungsten_VaryingDiameter.m	Program for obtaining the frequency responses of micro-fibers versus fiber varying diameter using two beams serially connected method. (Tungsten probe)
CarbonFiber Varying Length.m	Program for obtaining the frequency responses of micro-fibers versus fiber varying length using two beams serially connected method. (Carbon probe)
Makelist.m	Program for making a list of the probe's experimental frequency responses interactively
readtext_prb_freq_MagPrb16_Length_Rev2.m	Program for generating experimental frequency responses, plotting 3D graphs of magnitude versus resonant frequency versus length of the fiber. (Probe 16 in this example)
newVmaxChord-Final.m	Matlab program for calculating the probability, P_T , of finding good pattern versus h/T when particles are all traveling with the same velocity

ProbablityUniformIntegChord- Final.m	Matlab program for calculating the probability, P_τ , of finding good pattern versus h/T when particles are all traveling with a uniformly distributed velocity
ImageGenerationDecodingStatistic sNoGuiCluster, [1 3].m	Program for generating images with random particles. Folder must include matlab_r2014a.sh and DecodingAutomatedNumbers.m (code below) for running. (Need to be run on UNCC Viper cluster computer)
DecodingAutomatedNumbers.m	Program for finding encoded pattern in randomly generated images. (Decoding strategy of [1 3] is shown here as an example) (Run on UNCC Viper cluster computer)
ReadFromExcelPlot.m	Program for reading ePTV excel file outputs produced by the above two programs (Monte Carlo Simulation) and plotting the 3D graphs (\bar{N}_s versus θ and γ)
AlphaCalculation	Program for calculating edge effect value α in ePTV results

TheoreticalNfound.m	Program for calculating ePTV theoretical number of found
AllStatisticalAnalysis.m	Program for calculating the statistical analysis between the result of theoretical and simulated ePTV results
VideotoImages.m	Program for converting videos to image frames
FilteredVideoMaking.m	Program for making videos from images (filtering by subtracting two images)
VelocityMeasurment.m	Program for calculating velocities of particles (filtering by thresholding)
FilteringThresholding.m	Program for filtering the recorded images with thresholding method

WaveGenerate Library program (WaveGenerate.h)

```

#ifndef WaveGenerate_h
#define WaveGenerate_h

class WaveGenerate
{
public:
    WaveGenerate(char type, long atoivar, int ss, long clk);
    void Transfer(int ControlWord, long FrequencyReg);

```

```

private:

    long _atoivar, _clk;

    char _type;

    int i,j,cw,_ss;

    long atoivar, Freg, clk;

};

```

```

#endif

```

WaveGenerate Library program (WaveGenerate.cpp)

```

// This is the main DLL file.

```

```

#include "Arduino.h"

```

```

#include "WaveGenerate.h"

```

```

WaveGenerate::WaveGenerate(char type, long atoivar, int ss, long clk){
    _type=type;
    _atoivar=atoivar;
    _ss=ss;
    _clk=clk;
    Freg=atoivar*(pow(2,28)/(10*_clk)); // Value which goes to the Freq
Register
    pinMode(11,1);
    pinMode(12,1);
    pinMode(_ss,1);
    // Transfer(128,2); // reset

```

```

switch (_type){
case 's': //S for Sin wave
    cw=0;

    break;

case 't': //t for Triangle
    cw=16384;

    break;

case 'p': //S for Sin wave
    cw=5120;

    break;

} //end of switch

Transfer(cw,Freg);

digitalWrite(_ss,1);

Serial.println();

delay(100);

} // end of void loop

/// Defining Transfer Function here:///

void WaveGenerate::Transfer(int ControlWord, long FrequencyReg){

    ////////// writing Control Word //////////

    digitalWrite(12,0);

    digitalWrite(_ss,1);

    digitalWrite(12,1);

    delayMicroseconds(1);

    digitalWrite(_ss,0);

    for (byte i=0; i<16; i++) {

        byte cwState = bitRead(ControlWord, i);

        digitalWrite(11,cwState);

        delayMicroseconds(1);

```

```

    digitalWrite(12,0);

    delayMicroseconds(1);

    digitalWrite(12,1);
}

//////// writing Fine Tune //////////
digitalWrite(12,0);
digitalWrite(_ss,1);
digitalWrite(12,1);
delayMicroseconds(1);
digitalWrite(_ss,0);
//////// writing 01 to F reg //////////
digitalWrite(11,0);
delayMicroseconds(1);
digitalWrite(12,0);
delayMicroseconds(1);
digitalWrite(12,1);
digitalWrite(11,1);
delayMicroseconds(1);
digitalWrite(12,0);
delayMicroseconds(1);
digitalWrite(12,1);
//////// Fine Tune to Freg //////////
for (byte i=0; i<14; i++) {
    byte ftState = bitRead(FrequencyReg, 13-i);
    digitalWrite(11,ftState);
    delayMicroseconds(1);
    digitalWrite(12,0);
    delayMicroseconds(1);

```

```

    digitalWrite(12,1);
}

//////// writing Control Word //////////
digitalWrite(12,0);
digitalWrite(_ss,1);
digitalWrite(12,1);
delayMicroseconds(1);
digitalWrite(_ss,0);
for (byte i=0; i<16; i++) {
    byte cwState2 = bitRead(ControlWord+8, i);
    digitalWrite(11,cwState2);
    delayMicroseconds(1);
    digitalWrite(12,0);
    delayMicroseconds(1);
    digitalWrite(12,1);
}

//////// writing Coarse Tune //////////
digitalWrite(12,0);
digitalWrite(_ss,1);
digitalWrite(12,1);
delayMicroseconds(1);
digitalWrite(_ss,0);

///// writing 01 to F reg //////////
digitalWrite(11,0);
delayMicroseconds(1);
digitalWrite(12,0);
delayMicroseconds(1);
digitalWrite(12,1);

```

```

digitalWrite(11,1);

delayMicroseconds(1);

digitalWrite(12,0);

delayMicroseconds(1);

digitalWrite(12,1);

///// Coarse Tune to Freq /////

for (byte i=0; i<14; i++) {

    byte state = bitRead(FrequencyReg, 27-i);

    Serial.print(state);

    digitalWrite(11,state);

    delayMicroseconds(1);

    digitalWrite(12,0);

    delayMicroseconds(1);

    digitalWrite(12,1);

}

}

```

Joystick control for camera 3-axis automatic motion stage

```

// define chanel and variables

int xPot=A0;

int yPot=A1;

int zPot=A2;

int xSpeed=8;

int xDirection=9;

int ySpeed=10;

int yDirection=11;

int zSpeed=12;

int zDirection=13;

```

```
int x=0;

int y=0;

int z=0;

int dx1=0;

int dx2=0;

int dy1=0;

int dy2=0;

int dz1=0;

int dz2=0;


void setup() {

    // Define input and output:

    pinMode (xSpeed, OUTPUT);

    pinMode (xDirection, OUTPUT);

    pinMode (ySpeed, OUTPUT);

    pinMode (yDirection, OUTPUT);

    pinMode (zSpeed, OUTPUT);

    pinMode (zDirection, OUTPUT);

    pinMode (xPot, INPUT);

    pinMode (yPot, INPUT);

    pinMode (zPot, INPUT);

    // set initial digital values at LOW:

    digitalWrite (xSpeed, LOW);

    digitalWrite (xDirection, LOW);

    digitalWrite (ySpeed, LOW);

    digitalWrite (yDirection, LOW);

    digitalWrite (zSpeed, LOW);

    digitalWrite (zDirection, LOW);

    Serial.begin (9600);
```



```

}

void loop() {
    // x:
    x= analogRead (xPot);
    if (x > 516) {
        digitalWrite (xDirection, HIGH);
        if ( x<440 or x>560) {
            dx1=-1.18*x+1609;
            digitalWrite (xSpeed, HIGH);
            delayMicroseconds (dx1);
            digitalWrite (xSpeed, LOW);
            delayMicroseconds (dx1);
        }
    }
    else {
        digitalWrite (xDirection, LOW);

        if ( x<440 or x>560) {
            dx2=1.18*x+400;
            digitalWrite (xSpeed, HIGH);
            delayMicroseconds (dx2);
            digitalWrite (xSpeed, LOW);
            delayMicroseconds (dx2);
        }
    }

    // y:
    y= analogRead (yPot);

```

```

if (y > 516) {
    digitalWrite (yDirection, HIGH);
    if ( y<440 or y>560) {
        dy1=-1.18*y+1609;
    digitalWrite (ySpeed, HIGH);
    delayMicroseconds (dy1);
    digitalWrite (ySpeed, LOW);
    delayMicroseconds (dy1);
    }
}
else {
    digitalWrite (yDirection, LOW);

    if ( y<440 or y>560) {
        dy2=1.18*y+400;
    digitalWrite (ySpeed, HIGH);
    delayMicroseconds (dy2);
    digitalWrite (ySpeed, LOW);
    delayMicroseconds (dy2);
    }
}

// z:
z= analogRead (zPot);
if (z > 516) {
    digitalWrite (zDirection, HIGH);
    if ( z<440 or z>560) {
        dz1=-1.18*z+1609;
    digitalWrite (zSpeed, HIGH);

```

```

delayMicroseconds (dz1);

digitalWrite (zSpeed, LOW);

delayMicroseconds (dz1);
}

}

else {

    digitalWrite (zDirection, LOW);

    if ( z<440 or z>560) {

        dz2=1.18*z+400;

digitalWrite (zSpeed, HIGH);

delayMicroseconds (dz2);

digitalWrite (zSpeed, LOW);

delayMicroseconds (dz2);

    }

}

}

```

LED pulsing program through serial communication with Arduino microcontroller

```

/* format:
LEDOnTime, LEDOffTime, CameraOffTime, NumberOfBlink
*/

int trigger = 6;
int LED = 7;
int i;
String datastring;
boolean start = false;
long LEDOnTime, LEDOffTime, CameraOffTime, NumberOfBlink;

void setup() {
    pinMode(trigger, 1);
    pinMode(LED, 1);
    digitalWrite(LED, 1);
}

```

```

    digitalWrite(trigger, 0);
    Serial.begin(9600);
}

void loop() {
    while (Serial.available() > 0) {
        char data = Serial.read();
        datastring += data;
        if (data == '\n') {
            start = true;
        }
        if (start == true) {
            int c1 = datastring.indexOf(',');
            int c2 = datastring.indexOf(',', c1 + 1);
            int c3 = datastring.indexOf(',', c2 + 1);
            String LEDOnTimeStr = datastring.substring(0, c1);
            String LEDOffTimeStr = datastring.substring(c1 + 1, c2);
            String CameraOffTimeStr = datastring.substring(c2 + 1, c3);
            String NumberOfBlinkStr = datastring.substring(c3 + 1);
            LEDOnTime = LEDOnTimeStr.toInt();
            LEDOffTime = LEDOffTimeStr.toInt();
            CameraOffTime = CameraOffTimeStr.toInt();
            NumberOfBlink = NumberOfBlinkStr.toInt();
            Serial.println(LEDOnTime);
            Serial.println(LEDOffTime);
            Serial.println(CameraOffTime);
            Serial.println(NumberOfBlink);
            Serial.println(data);
            while (Serial.available() > 0) {
                LEDPulse(LEDOnTime, LEDOffTime, CameraOffTime, NumberOfBlink);
            }

            datastring = "";
            start = false;
        }
    }
}

void LEDPulse(long LEDOnTime, long LEDOffTime, long cameraOffTime, long
numberOfBlink) {
    for (i = 1; i < numberOfBlink; i++) {
        digitalWrite(trigger, 1);
        delay(1);
        digitalWrite(trigger, 0);
        delay(1);
        digitalWrite(LED, 0);
        delay(LEDOnTime);
        digitalWrite(LED, 1);
        delay(LEDOffTime);
    }
    delay(cameraOffTime);
}

```

Camera and LED synchronization code with SAJI WaveGenerate

```

#include <WaveGenerate.h>
void setup(){
    pinMode(7,1);
    pinMode(6,0);
    delay(4000);
    WaveGenerate waveGenerate('p', 1, 9, 25000000);
}

void loop(){
    boolean trigger =digitalRead(6);
    if (trigger=1){
        digitalWrite(7,0);
        delayMicroseconds(200000);
        digitalWrite(7,1);
        delayMicroseconds(5000);
        digitalWrite(7,0);
        delayMicroseconds(200000);
        digitalWrite(7,1);
        delayMicroseconds(800000);
    }
}

```

Multi-oscillator, micro-vortex apparatus drive program

```

    // define chanel and variables
int xPot=A0;
int yPot=A1;
int zPot=A2;
int qPot=A3;
int xSpeed=8;
int xDirection=9;
int ySpeed=10;
int yDirection=11;
int zSpeed=12;
int zDirection=13;
int qSpeed=6;
int qDirection=7;
int x=0;
int y=0;
int z=0;
int q=0;
int dx1=0;
int dx2=0;
int dy1=0;
int dy2=0;
int dz1=0;
int dz2=0;
int dq1=0;
int dq2=0;

void setup() {
    // Define input and output:
    pinMode (xSpeed, OUTPUT);
    pinMode (xDirection, OUTPUT);
    pinMode (ySpeed, OUTPUT);
    pinMode (yDirection, OUTPUT);

```

```

pinMode (zSpeed, OUTPUT);
pinMode (zDirection, OUTPUT);
pinMode (qSpeed, OUTPUT);
pinMode (qDirection, OUTPUT);
pinMode (xPot, INPUT);
pinMode (yPot, INPUT);
pinMode (zPot, INPUT);
pinMode (qPot, INPUT);
// set initial digital values at LOW:
digitalWrite (xSpeed, LOW);
digitalWrite (xDirection, LOW);
digitalWrite (ySpeed, LOW);
digitalWrite (yDirection, LOW);
digitalWrite (zSpeed, LOW);
digitalWrite (zDirection, LOW);
digitalWrite (qSpeed, LOW);
digitalWrite (qDirection, LOW);
Serial.begin (9600);
}

void loop() {
  // x:
  x= analogRead (xPot);
  if (x > 516) {
    digitalWrite (xDirection, LOW);
    if ( x<440 or x>560) {
      dx1=-1.18*x+1609;
      digitalWrite (xSpeed, HIGH);
      delayMicroseconds (dx1);
      digitalWrite (xSpeed, LOW);
      delayMicroseconds (dx1);
    }
  }
  else {
    digitalWrite (xDirection, HIGH);

    if ( x<440 or x>560) {
      dx2=1.18*x+400;
      digitalWrite (xSpeed, HIGH);
      delayMicroseconds (dx2);
      digitalWrite (xSpeed, LOW);
      delayMicroseconds (dx2);
    }
  }

  // y:
  y= analogRead (yPot);
  if (y > 516) {
    digitalWrite (yDirection, HIGH);
    if ( y<440 or y>560) {
      dy1=-1.18*y+1609;
      digitalWrite (ySpeed, HIGH);
      delayMicroseconds (dy1);
      digitalWrite (ySpeed, LOW);
      delayMicroseconds (dy1);
    }
  }
}

```

```

}
else {
    digitalWrite (yDirection, LOW);

    if ( y<440 or y>560) {
        dy2=1.18*y+400;
        digitalWrite (ySpeed, HIGH);
        delayMicroseconds (dy2);
        digitalWrite (ySpeed, LOW);
        delayMicroseconds (dy2);
    }
}

// z:
z= analogRead (zPot);
if (z > 516) {
    digitalWrite (zDirection, LOW);
    if ( z<440 or z>560) {
        dz1=-1.18*z+1609;
        digitalWrite (zSpeed, HIGH);
        delayMicroseconds (dz1);
        digitalWrite (zSpeed, LOW);
        delayMicroseconds (dz1);
    }
}
else {
    digitalWrite (zDirection, HIGH);

    if ( z<440 or z>560) {
        dz2=1.18*z+400;
        digitalWrite (zSpeed, HIGH);
        delayMicroseconds (dz2);
        digitalWrite (zSpeed, LOW);
        delayMicroseconds (dz2);
    }
}

// q:
q= analogRead (qPot);
if (q > 516) {
    digitalWrite (qDirection, HIGH);
    if ( q<440 or q>560) {
        dq1=-1.18*q+1609;
        digitalWrite (qSpeed, HIGH);
        delayMicroseconds (dq1);
        digitalWrite (qSpeed, LOW);
        delayMicroseconds (dq1);
    }
}
else {
    digitalWrite (qDirection, LOW);

    if ( q<440 or q>560) {
        dq2=1.18*q+400;
        digitalWrite (qSpeed, HIGH);

```

```

delayMicroseconds (dq2);
digitalWrite (qSpeed, LOW);
delayMicroseconds (dq2);
}
}
}

```

Matlab program for calculating amplitude frequency response of the fiber, H_{iy} for $i = 1, 2, 3$, and 4 as a function of fiber length based on the lumped model approximation (tungsten probe)

```

clear all

fontsize=16;

rho=19250; % density of tungsten
Ew=410e9; % elastic modulus for tungsten
lo=.0300; % Original fiber length
Ro=37.5e-6; % Original fiber radius
Io=pi*(Ro^4)/4; % second moment of area
AlphaL=[1.87510406873
        4.694091132974
        7.85475743823765
        10.995540734875
        14.1371683910465
        17.2787595320882
        20.4203522510412
        23.5619449018065
        26.7035375555183
        29.8451302091028]; % Alpha values for predicting modal
frequencies lambdai

```



```

Ao=pi*Ro^2; % Area of tungsten wire

n=8; % for computation including up to 11 modal frequencies (probably a
little high)

epsi=0.000001; % assumes the same damping ratio for each modes
(proportionate damping)

mo=rho*Ao*lo; % original mass of fiber


% Frequency generation parameters

FreqStart=4000; % frequency for start of plot
FreqStop=104000; % defines frequency band
FreqBand=FreqStop-FreqStart; % Obvious
DataPoints=5000; %number of data ponits to be used in computation
FrequencyInc=FreqBand/DataPoints; % frequency sample interval


% tuning fork parameters

Eq=71e9; % Quartz elastic modulus

Lf=0.00365; % length of the tuning fork tine
tf=0.00024; % thickness of the tuning fork tine (Depth)
Wf=0.00059; % width of the tuning fork tine (Thickness)
rhof=2634; % density of the tuning fork tine

Volf= Lf*tf*Wf; % volume of the tuning fork tine

% Mf=rhof*Volf; % complete mass of tuning fork tine to be used for
calculations

Omegaf=2*pi*(FreqStop); % places resonance of the tuning fork tine in
the middle of the plotting range

kf=Eq*Wf*tf*tf*tf/(4*Lf*Lf*Lf); % Tine stiffness

Mf=kf/Omegaf^2; % equivalent mass of the tuning fork tine

```

```

epsf=0.001; % damping ratio of tuning fork tine

bf=2*epsf*sqrt(Mf*kf); % damping coefficient for tuning fork tine


% This section will change length in fixed increments and compute array
of
% subsequent responses

%LengthRemoved=lo*0.6; % assumes that 80% of length is removed during
etching if coeff. is 0.8 etc.

LengthRemoved=0.001;    % Initial Length

NumberEtch=100; % Assumes that this length will be removed in this
number of 'bites'

%LengthInc=LengthRemoved/NumberEtch; % length removed with each etch

LengthInc=(lo-LengthRemoved)/NumberEtch;


for z=1:4

    for m=1:NumberEtch

        l(m)=lo-LengthInc*(m-1); % etched length of fiber

        % section for computing modal parameters of the tungsten wire

        me=rho*Ao*l(m)/(50*n); % equivalent mass divided by the number
of modes to be used

        % me=rho*Ao*l(m); % division removed

        ltemp=l(m);

        DelMass(m)=100000000*(mo-50*n*me); % equivalent mass removed in
ug

        for i=1:n

            lambda(i)=AlphaL(i)*AlphaL(i)*sqrt(Ew*Io/(me*ltemp^3)); %
eigenvalues for nth modes

```

```

FnLambda(m,i)=lambda(i)/(2*pi*1000); % modal frequencies in
kHz

k(i)=(AlphaL(i)^4)*Ew*Io/(n*ltemp^4); % modal stiffness
values

% k(i)=(AlphaL(i)^4)*Ew*Io/(ltemp^4); % division removed

bi(i)=2*epsi*sqrt(k(i)*me)/n; % array of damping values

end

% Computation of sum k(i) and b(i)

sumki=0;

sumbi=0;

for i=1:n

    sumki=sumki+k(i);

    sumbi=sumbi+bi(i);

end

% section for computing the frequency response

for i=1:DataPoints

    omega(i)=2*pi*(FreqStart+(i-1)*FrequencyInc); % excitation
frequency

    jomega=sqrt(-1)*omega(i); % complex jw term

    sumresp=0; % initialize sum response to zero to prevent it
from continuing sum from previous

    for j=1:n % to compute the sum term in the denominator for
Hoy

        CompPartResp=2*jomega*epsi*omega(i)/lambda(j); %
complex part of response

        TempModalResp=(k(j)+jomega*bi(j))*(1+CompPartResp)/((1-
(omega(i)^2)/(lambda(j)*lambda(j)))+CompPartResp);

```

```

sumresp=sumresp+TempModalResp;

end

Hoy(m,i)=(kf+jomega*bf)/(kf+sumki+jomega*(bf+sumbi)-
Mf*omega(i)^2-sumresp); % complex frequency response

FreqkHz(i)=omega(i)/(2*pi*1000); % Array of excitation
frequency in kHz

Hiy(m,i)=Hoy(m,i)*(((k(z)+jomega*bi(z))*(1+2*jomega*epsi/lambda(z)))/((
1-(omega(i)^2)/(lambda(z)^2))+2*jomega*epsi/lambda(z)));

end

% progress=m/NumberEtch

end

for m=1:NumberEtch

    l(m)=lo-LengthInc*(m-1); % etched length of fiber

    P3Freq(m,:)=FreqkHz;

end

for i=1:DataPoints

    P3l(:,i)=1;

    P3m(:,i)=DelMass;

end

figure(z)

plot3(P3Freq,P3l,log(abs(Hiy)),'.','MarkerSize',4)

xlabel('Excitation frequency (kHz)','FontSize', fontsize)

ylabel('Length of fiber (m)','FontSize', fontsize)

zlabel('Magnitude frequency response','FontSize', fontsize)

% hold on

```

end

Matlab program for calculating amplitude and phase frequency response of the tuning fork tine as a function of attached tungsten fiber length based on lumped model approximation

```
clear all
```

```
fontsize=16;
```

```
rho=19250; % density of tungsten
```

```
Ew=410e9; % elastic modulus for tungsten
```

```
lo=.0300; % Original fiber length
```

```
Ro=37.5e-6; % Original fiber radius
```

```
Io=pi*(Ro^4)/4; % second moment of area
```

```
AlphaL=[1.87510406873
```

```
4.694091132974
```

```
7.85475743823765
```

```
10.995540734875
```

```
14.1371683910465
```

```
17.2787595320882
```

```
20.4203522510412
```

```
23.5619449018065
```

```
26.7035375555183
```

```
29.8451302091028]; % Alpha values for predicting modal
```

```
frequencies lambda
```

```
Ao=pi*Ro^2; % Area of tungsten wire
```

```

n=8; % for computation including up to 11 modal frequencies (probably a
little high)

epsi=0.000001; % assumes the same damping ratio for each modes
(proportionate damping)

mo=rho*Ao*lo; % original mass of fiber

% Frequency generation parameters

FreqStart=4000; % frequency for start of plot
FreqStop=84000; % defines frequency band
FreqBand=FreqStop-FreqStart; % Obvious
DataPoints=5000; %number of data ponits to be used in computation
FrequencyInc=FreqBand/DataPoints; % frequency sample interval

% tuning fork parameters
Eq=71e9; % Quartz elastic modulus
Lf=0.00365; % length of the tuning fork tine
tf=0.00024; % thickness of the tuning fork tine
Wf=0.00059; % width of the tuning fork tine
rhof=2634; % density of the tuning fork tine
Volf= Lf*tf*Wf; % volume of the tuning fork tine
% Mf=rhof*Volf; % complete mass of tuning fork tine to be used for
calculations

Omegaf=2*pi*(FreqStop); % places resonance of the tuning fork tine in
the middle of the plotting range

kf=Eq*Wf*tf*tf*tf/(4*Lf*Lf*Lf); % Tine stiffness
Mf=kf/Omegaf^2; % equivalent mass of the tuning fork tine
epsf=0.0001; % damping ratio of tuning fork tine
bf=2*epsf*sqrt(Mf*kf); % damping coefficient for tuning fork tine

```

```

% This section will change length in fixed increments and compute array
of
% subsequent responses

%LengthRemoved=lo*0.6; % assumes that 80% of length is removed during
etching if coeff. is 0.8 etc.

LengthRemoved=0.001;    % Initial Length

NumberEtch=100; % Assumes that this length will be removed in this
number of 'bites'

%LengthInc=LengthRemoved/NumberEtch; % length removed with each etch

LengthInc=(lo-LengthRemoved)/NumberEtch;

for m=1:NumberEtch

    l(m)=lo-LengthInc*(m-1); % etched length of fiber

    % section for computing modal parameters of the tungsten wire

    me=rho*Ao*l(m)/(50*n); % equivalent mass divided by the number of modes
    to be used

    % me=rho*Ao*l(m); % division removed

    ltemp=l(m);

    DelMass(m)=100000000*(mo-50*n*me); % equivalent mass removed in ug

    for i=1:n

        lambda(i)=AlphaL(i)*AlphaL(i)*sqrt(Ew*Io/(me*ltemp^3)); %
eigenvalues for nth modes

        FnLambda(m,i)=lambda(i)/(2*pi*1000); % modal frequencies in kHz

        k(i)=(AlphaL(i)^4)*Ew*Io/(n*ltemp^4); % modal stiffness values

        % k(i)=(AlphaL(i)^4)*Ew*Io/(ltemp^4); % division removed

        bi(i)=2*epsi*sqrt(k(i)*me)/n; % array of damping values

```

```

end

% Computation of sum k(i) and b(i)

sumki=0;

sumbi=0;

for i=1:n

    sumki=sumki+k(i);

    sumbi=sumbi+bi(i);

end

% section for computing the frequency response

for i=1:DataPoints

    omega(i)=2*pi*(FreqStart+(i-1)*FrequencyInc); % excitation
frequency

    jomega=sqrt(-1)*omega(i); % complex jw term

    sumresp=0; % initialize sum response to zero to prevent it from
continuing sum from previous

    for j=1:n % to compute the sum term in the denominator for Hoy

        CompPartResp=2*jomega*epsi*omega(i)/lambda(j); % complex part
of response

        TempModalResp=(k(j)+jomega*bi(j))*(1+CompPartResp)/((1-
(omega(i)^2)/(lambda(j)*lambda(j)))+CompPartResp);

        sumresp=sumresp+TempModalResp;

    end

    Hoy(m,i)=(kf+jomega*bf)/(kf+sumki+jomega*(bf+sumbi)-Mf*omega(i)^2-
sumresp); % complex frequency response

    FreqkHz(i)=omega(i)/(2*pi*1000); % Array of excitation frequency in
kHz

end

```



```

progress=m/NumberEtch

end

figure(1)
mesh(FreqkHz,l,log(abs(Hoy)))
xlabel('Excitation frequency (kHz)','FontSize', fontsize)
ylabel('Length of fiber (m)','FontSize', fontsize)
zlabel('Magnitude frequency response','FontSize', fontsize)

figure(2)
mesh(FreqkHz,l,(angle(Hoy)))
xlabel('Excitation frequency (kHz)','FontSize', fontsize)
ylabel('Length of fiber (m)','FontSize', fontsize)
zlabel('Phase response (rad)','FontSize', fontsize)

for m=1:NumberEtch

    l(m)=lo-LengthInc*(m-1); % etched length of fiber
    P3Freq(m,:)=FreqkHz;

end

for i=1:DataPoints

    P3l(:,i)=l;

    P3m(:,i)=DelMass;

end

```

```

figure(3)

plot3(P3Freq,P3l,log(abs(Hoy)),'.','MarkerSize',4)

xlabel('Excitation frequency (kHz)','FontSize', fontsize)
ylabel('Length of fiber (m)','FontSize', fontsize)
zlabel('Magnitude frequency response','FontSize', fontsize)

```

```

figure(4)

plot3(P3Freq,P3l,(angle(Hoy)),'.','MarkerSize',4)

xlabel('Excitation frequency (kHz)','FontSize', fontsize)
ylabel('Length of fiber (m)','FontSize', fontsize)
zlabel('Phase response (rad)','FontSize', fontsize)

```

```

figure(5)

plot3(P3Freq,P3m,log(abs(Hoy)),'.','MarkerSize',4)

xlabel('Excitation frequency (kHz)','FontSize', fontsize)
ylabel('Mass removed (ug)','FontSize', fontsize)
zlabel('Magnitude frequency response','FontSize', fontsize)

```

Matlab function program used to find the eigenvalues of this matrix governing motion using two beams serially connected method

```

function [ DetA ] = PsiDeterminant( beta, Eigpsi,gamma, k)

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

A1=[cos(beta*Eigpsi)-cosh(beta*Eigpsi) sin(beta*Eigpsi)-
sinh(beta*Eigpsi) -1 -1 0 0];

```

```

A2=[-beta*gamma*(sin(beta*Eigpsi)+sinh(beta*Eigpsi))
beta*gamma*(cos(beta*Eigpsi)-cosh(beta*Eigpsi)) 0 0 -1 -1];

A3=[(-(beta*gamma)^2)*(cos(beta*Eigpsi)+cosh(beta*Eigpsi)) (-
(beta*gamma)^2)*(sin(beta*Eigpsi)+sinh(beta*Eigpsi)) k -k 0 0];

A4=[((beta*gamma)^3)*(sin(beta*Eigpsi)-sinh(beta*Eigpsi)) (-
(beta*gamma)^3)*(cos(beta*Eigpsi)+cosh(beta*Eigpsi)) 0 0 k -k];

A5=[0 0 -cos(Eigpsi) cosh(Eigpsi) -sin(Eigpsi) sinh(Eigpsi)];
A6=[0 0 sin(Eigpsi) sinh(Eigpsi) -cos(Eigpsi) cosh(Eigpsi)];

A=[A1;A2;A3;A4;A5;A6]; %Define Entire Matrix as A

DetA=det(A); %Find Determinant of Matrix A

End

```

Matlab program used to find the frequency responses versus fiber varying length using two beams serially connected method, tungsten probe

```

clear all

%Lateral Vibration of two beams serially connected

%Tine Parameters

old=digits(32);

L_tine=0.00365; %Length of Tine up until the radial (m)
b=0.00024; %Depth of Tine across the top (m)
h=0.00059; %Thickness of Tine across a side (m)
rho_tine=2634; %Density of Quartz in Tine (kg/m^3)

```

```

E_tine=71000000000;      %Young's Modulus of Quartz in Tine (kg/m*s^2)

%Tine Secondary Calculations

I_tine=(h*b^3)/12;          %Second Moment of Area for Tine
(x -- parallel to top)

m_tine=rho_tine*L_tine*h*b;      %Mass of Tine (kg)

%k_tine=(3*E_tine*I_tine)/(L_tine^3); %stiffness of tine

% Etching paramters

etchsteps=500;    % number of steps to etch total length from fiber
(MINIMUM = 2)

%Loop Operations

L_wire=0.00816

Datapoints=500;      % number of datapoints in plot

PsiStart=0.3;        % value for start of plot

PsiStop=50;          % maximum value of psi

PsiRange= PsiStop-PsiStart; % range of psi to use in determinant
calculation

PsiInc=PsiRange/Datapoints; % Increments for psi

n=round(PsiStart/PsiInc); %Define "n" to start the loop

s=n+Datapoints;      %Define "s" to stop the loop

%Wire Parameters

```

```

D_start=0.000075;          % Starting length of the wire
D_final=0.000005;          % Final length of fiber after etching
D_Inc=(D_start-D_final)/(etchsteps-1); % Length of fiber to remove at
each etch step
AlphaL=[1.87510406873
        4.694091132974
        7.85475743823765
        10.995540734875
        14.1371683910465
        17.2787595320882
        20.4203522510412
        23.5619449018065
        26.7035375555183
        29.8451302091028]; % Alphas values for predicting modal
frequencies lambda_i

for j=1:etchsteps

SlopeTolerance=0.5;          % Values of psi to be recorded for the slope of
the determinant
SlopeTemp=0;                 % dummy variable for computing slope without
requiring previous value at outset
PsiExtractCount=0;           % index variable for collecting psi values that
are within tolerance for the slope of D this resets each time around
the loop
TransitionPsi=15;            % gage for when slope transitions from positive
to negative

```

```

D_wire(j)=D_start-(j-1)*D_Inc; %Length of wire from tip of tine (m)
Wire_D(j)=D_wire(j); % Array of length of wire at each step.

%r=0.000075/2; %Radius of wire (m)

rho_wire=2634; %Density of Tungsten 19250(kg/m^3)

E_wire=71000000000; %Young's Modulus of Tungsten
411000000000(kg/m*s^2)

%Wire Secondary Calculations

I_wire(j)=(pi*((D_wire(j)/2)^4)/4); %Second Moment of Area for Wire
m_wire(j)=rho_wire*L_wire*pi*((D_wire(j)/2)^2);

%%%% AlphaL Canteliver

for ii=1:10

OmegaNFiberCant(j,ii)=(AlphaL(ii)^2)*(sqrt((E_wire*I_wire(j))/(m_wire(j)
)*L_wire^3)));

end

%Primary Calculations

%omega=((k.tine/m_tine)^(1/2)); %Omega (free state, undamped, natrual
frequency)

%alpha=((rho_wire*(omega^2))/(E_wire*I_wire))^(1/4);

%Simplification into Alpha (don't need yet)

beta(j)=((m_tine*(L_tine^3)*E_wire*I_wire(j))/(m_wire(j)*(L_wire^3)*E_t
ine*I_tine))^(1/4);

%Simplification into Beta

gamma(j)=L_wire/L_tine;

```

```

        %Simplification into Gamma

%psi=alpha*L_wire;

        %Simplification into Psi (dont need because psi is varied)
k(j)=E_wire*I_wire(j);

        %Simplification into K


for i=1:Datapoints      %start of for loop to find solution
(startvalue:stepinterval:endvalue)

    Eigpsi(j,i)=PsiStart+i*PsiInc; % Create array variable psi
%   %A1,A2....A6 are rows of matrix A
    B=[0;0;0;0;0;0;0];      %Define Vector B
    %Inverse=pinv(A)
    D(j,i)=PsiDeterminant(beta(j), Eigpsi(j,i), gamma(j), k(j));
%Find Determinant of Matrix A
    EigSlope(j,i)=(SlopeTemp-log(D(j,i)))/PsiInc; % Slope of log of
determinant
    SlopeTemp=log(D(j,i)); % Update slope temp.
    % if(abs(EigSlope(i))<SlopeTolerance) % search for zero values of log
(D)
    if(i==1) % this avoidstrying to look at changes in slope for the
first data point in psi
        elseif(EigSlope(j,i-1)-EigSlope(j,i)>TransitionPsi) % this searches
for rapid changes in slope above some transition value
            PsiExtractCount=PsiExtractCount+1; % for each etch step
            PsiRoots(j,PsiExtractCount)=Eigpsi(j,i); % drops the roots into
the Jth row

```

```

D_Wire_Extract(j,PsiExtractCount)=D_wire(j); % Not sure whether
this needs to be two dimensional !!!!!!!

```

```

I_Wire_Extract(j,PsiExtractCount)=I_wire(j);
m_Wire_Extract(j,PsiExtractCount)=m_wire(j);
end

```

```

n=n+0.1; %increase "n" by set amount (step value)
and show new "n" (x,y format)

Progress=[100*i/Datapoints 100*j/etchsteps];
end
end

```

```

FuncForFzero=@(x) PsiDeterminant(beta(j), x, gamma(j),k(j));

% Look for zero using fzero Matlab function
TolX=0.00000001;
options=optimset('TolX', 0.000001);
for j=1:etchsteps
    for i=1:PsiExtractCount
        DFzeroRoot(j,i)=fzero(FuncForFzero, PsiRoots(j,i), options);
    end
end
end

```

```

[Steps NumRoots]=size(DFzeroRoot);

```

```

for j=1:Steps
    for i=1:NumRoots

```



```

Omegas(j,i)=DFzeroRoot(j,i)*DFzeroRoot(j,i)*sqrt(E_wire*I_Wire_Extract(
j,i)/(m_Wire_Extract(j,i)*L_wire^3));

FreqkHz(j,i)=Omegas(j,i)/(2*pi*1000);

    end

end

z=zeros(100,100);

    for j=1:100      %(j indexes the etch steps)
        for k=1:10      %(k indexes the modes)
            for i=1:100      %(i indexes the frequency)
                if (i==round(FreqkHz(j,k)))
                    z(j,i)=1;
                end
            end
        end
    end

end

figure(5)

plot(D_Wire_Extract(1:etchsteps,1:15)*1000,FreqkHz(1:etchsteps,1:15), 'b
. ')

title('Roots (Natural Frequencies)at Diffrent Lenghts', 'FontSize', 16)
ylabel('Frequency (kHz)')
xlabel('Fiber Diameter (mm)')

hold on

plot(D_wire*1000,(OmegaNFiberCant)/(2000*pi), 'o')

hold off

```

```

figure(6)

p=mesh(z);

title('3-D plot, Frequency vs Etch Step vs Z value', 'FontSize', 16)

xlabel('Frequency (kHz)')

ylabel('Etch steps')

zlabel('Z')

% axis ([25,36 0,50 0,5]);

aH = ancestor(p,'axes');

x = get(p,'XData');

y = get(p,'YData');

new_y = D_start-(y-1)*D_Inc;

z=get(p,'ZData');

figure (7), mesh(x,new_y,z);

save allvariables

```

Matlab program used to find the frequency responses versus fiber varying diameter using two beams serially connected method, tungsten probe

```

clear all

%Lateral Vibration of two beams serially connected

%Tine Parameters

old=digits(32);

L_tine=0.00365;           %Length of Tine up until the radial (m)

b=0.00024;               %Depth of Tine across the top (m)

h=0.00059;               %Thickness of Tine across a side (m)

```

```

rho_tine=2634;           %Density of Quartz in Tine (kg/m^3)

E_tine=71000000000;      %Young's Modulus of Quartz in Tine (kg/m*s^2)

%Tine Secondary Calculations

I_tine=(h*b^3)/12;        %Second Moment of Area for Tine
(x -- parallel to top)

m_tine=rho_tine*L_tine*h*b; %Mass of Tine (kg)

%k_tine=(3*E_tine*I_tine)/(L_tine^3); %stiffness of tine

% Etching paramters

etchsteps=200; % number of steps to etch total length from fiber
(MINIMUM = 2)

%Loop Operations

Datapoints=500; % number of datapoints in plot

PsiStart=0.3; % value for start of plot

PsiStop=50; % maximum value of psi

PsiRange= PsiStop-PsiStart; % range of psi to use in determinant
calculation

PsiInc=PsiRange/Datapoints; % Increments for psi

n=round(PsiStart/PsiInc); %Define "n" to start the loop

s=n+Datapoints; %Define "s" to stop the loop

```

```

%Wire Parameters

L_startLength=0.016;          % Starting length of the wire
L_finalLength=0.001;          % Final length of fiber after etching
L_Inc=(L_startLength-L_finalLength)/(etchsteps-1); % Length of fiber to
remove at each etch step

AlphaL=[1.87510406873
        4.694091132974
        7.85475743823765
        10.995540734875
        14.1371683910465
        17.2787595320882
        20.4203522510412
        23.5619449018065
        26.7035375555183
        29.8451302091028]; % Alpha values for predicting modal
frequencies lambda_i

for j=1:etchsteps

    SlopeTolerance=0.5;        % Values of psi to be recorded for the slope of
the determinant

    SlopeTemp=0;                % dummy variable for computing slope without
requiring previous value at outset

    PsiExtractCount=0;          % index variable for collecting psi values that
are within tolerance for the slope of D this resets each time around
the loop

```

```

TransitionPsi=15;          % gage for when slope transitions from positive
                             to negative

L_wire(j)=L_startLength-(j-1)*L_Inc; %Length of wire from tip of tine
(m)

Wire_Length(j)=L_wire(j);    % Array of length of wire at each step.

r=0.000075/2;              %Radius of wire (m)

rho_wire=19250;             %Density of Tungsten 19250(kg/m^3)

E_wire=411000000000;        %Young's Modulus of Tungsten
411000000000(kg/m*s^2)

%Wire Secondary Calculations

I_wire(j)=(pi*(r^4)/4); %Second Moment of Area for Wire

m_wire(j)=rho_wire*L_wire(j)*pi*(r^2);

%%%% AlphaL Canteliver

for ii=1:10

    OmegaNFiberCant(j,ii)=(AlphaL(ii)^2)*(sqrt((E_wire*I_wire(j))/(m_wire(j)
    )*L_wire(j)^3)));

end

%Primary Calculations

%omega=((k.tine/m_tine)^(1/2)); %Omega (free state, undamped, natrual
frequency)

%alpha=((rho_wire*(omega^2))/(E_wire*I_wire))^(1/4);

    %Simplification into Alpha (don't need yet)

```

```

beta(j)=(m_tine*(L_tine^3)*E_wire*I_wire(j))/(m_wire(j)*(L_wire(j)^3)*
E_tine*I_tine))^(1/4);

    %Simplification into Beta
gamma(j)=L_wire(j)/L_tine;

    %Simplification into Gamma
%psi=alpha*L_wire;

    %Simplification into Psi (dont need because psi is varied)
k(j)=E_wire*I_wire(j);

    %Simplification into K

for i=1:Datapoints      %start of for loop to find solution
(startvalue:stepinterval:endvalue)

    Eigpsi(j,i)=PsiStart+i*PsiInc; % Create array variable psi
%   %A1,A2....A6 are rows of matrix A
    B=[0;0;0;0;0;0;0];      %Define Vector B

    %Inverse=pinv(A)

    D(j,i)=PsiDeterminant(beta(j), Eigpsi(j,i), gamma(j), k(j));

%Find Determinant of Matrix A

    EigSlope(j,i)=(SlopeTemp-log(D(j,i)))/PsiInc; % Slope of log of
determinant

    SlopeTemp=log(D(j,i)); % Update slope temp.

    % if(abs(EigSlope(i))<SlopeTolerance) % search for zero values of log
(D)

    if(i==1) % this avoids trying to look at changes in slope for the
first data point in psi

        elseif(EigSlope(j,i-1)-EigSlope(j,i)>TransitionPsi) % this searches
for rapid changes in slope above some transition value

```

```

PsiExtractCount=PsiExtractCount+1; % for each etch step

PsiRoots(j,PsiExtractCount)=Eigpsi(j,i); % drops the roots into
the Jth row

L_Wire_Extract(j,PsiExtractCount)=L_wire(j); % Not sure whether
this needs to be two dimensional !!!!!!!

I_Wire_Extract(j,PsiExtractCount)=I_wire(j);

m_Wire_Extract(j,PsiExtractCount)=m_wire(j);

end

n=n+0.1; %increase "n" by set amount (step value)
and show new "n" (x,y format)

Progress=[100*i/Datapoints 100*j/etchsteps];

end

end

FuncForFzero=@(x) PsiDeterminant(beta(j), x, gamma(j),k(j));

% Look for zero using fzero Matlab function

TolX=0.00000001;

options=optimset('TolX', 0.000001);

for j=1:etchsteps

    for i=1:PsiExtractCount

        DFzeroRoot(j,i)=fzero(FuncForFzero, PsiRoots(j,i), options);

    end

end

syms A11 A33 B11 B22 B33 B44

[Steps NumRoots]=size(DFzeroRoot);

```

```

for j=1:Steps
    for i=1:NumRoots

        Omegas(j,i)=DFzeroRoot(j,i)*DFzeroRoot(j,i)*sqrt(E_wire*I_Wire_Extract(
j,i)/(m_Wire_Extract(j,i)*L_Wire_Extract(j,i)^3));

        FreqkHz(j,i)=Omegas(j,i)/(2*pi*1000);

    end
end

z=zeros(200,150);

    for j=1:200        %(j indexes the etch steps)
        for k=1:11      %(k indexes the modes)
            for i=1:150    %(i indexes the frequency)

                if (i==round(FreqkHz(j,k)))

                    z(j,i)=1;

                end

            end

        end

    end

end

figure(5)

plot(L_Wire_Extract(1:etchsteps,1:15)*1000,FreqkHz(1:etchsteps,1:15),'b
*')

title('Roots (Natural Frequencies)at Diffrent Lenghts', 'FontSize', 16)
ylabel('Frequency (kHz)')
xlabel('Fiber length (mm)')

hold on

```



```

plot(Wire_Length*1000,(OmegaNFiberCant)/(2000*pi), 'mo')

hold off

%

% %freq=1:110;

% %figure(4)

% %plot3(freq,z,'x')

%

figure(6)

p=mesh(z);

title('3-D plot, Frequency vs Etch Step vs Z value', 'FontSize', 16)

xlabel('Frequency (kHz)')

ylabel('Etch steps')

zlabel('Z')

% axis ([25,36 0,50 0,5]);

aH = ancestor(p, 'axes');

x = get(p, 'XData');

y = get(p, 'YData');

new_y = L_startLength-(y-1)*L_Inc;

z=get(p, 'ZData');

figure (7), mesh(x,new_y,z);

xlabel('Frequency (kHz)')

ylabel('Length (mm)')

zlabel('Z')

save allvariables

```

Matlab program used to find the frequency responses versus fiber varying length using two beams serially connected method, carbon probe

```
clear all

%Lateral Vibration of two beams serially connected

%Tine Parameters

old=digits(32);

L_tine=0.00365;           %Length of Tine up until the radial (m)
b=0.00024;               %Depth of Tine across the top (m)
h=0.00059;               %Thickness of Tine across a side (m)
rho_tine=2634;            %Density of Quartz in Tine (kg/m^3)
E_tine=71000000000;       %Young's Modulus of Quartz in Tine (kg/m*s^2)

%Tine Secondary Calculations

I_tine=(h*b^3)/12;        %Second Moment of Area for Tine
(x -- parallel to top)

m_tine=rho_tine*L_tine*h*b; %Mass of Tine (kg)
%k_tine=(3*E_tine*I_tine)/(L_tine^3); %stiffness of tine

% Etching paramters

etchsteps=500; % number of steps to etch total length from fiber
(MINIMUM = 2)

%Loop Operations

Datapoints=500;          % number of datapoints in plot
```

```

PsiStart=0.3;                % value for start of plot
PsiStop=50;                  % maximum value of psi
PsiRange= PsiStop-PsiStart; % range of psi to use in determinant
                             calculation
PsiInc=PsiRange/Datapoints; % Increments for psi
n=round(PsiStart/PsiInc);    %Define "n" to start the loop

s=n+Datapoints;              %Define "s" to stop the loop


%Wire Parameters

L_startLength=0.006;         % Starting length of the wire
L_finalLength=0.0002;        % Final length of fiber after etching
L_Inc=(L_startLength-L_finalLength)/(etchsteps-1); % Length of fiber to
                             remove at each etch step
AlphaL=[1.87510406873
        4.694091132974
        7.85475743823765
        10.995540734875
        14.1371683910465
        17.2787595320882
        20.4203522510412
        23.5619449018065
        26.7035375555183
        29.8451302091028]; % Alphas values for predicting modal
                             frequencies lambda_i

```

```

for j=1:etchsteps

SlopeTolerance=0.5;      % Values of psi to be recorded for the slope of
the determinant

SlopeTemp=0;             % dummy variable for computing slope without
requiring previous value at outset

PsiExtractCount=0;       % index variable for collecting psi values that
are within tolerance for the slope of D this resets each time around
the loop

TransitionPsi=15;        % gage for when slope transitions from positive
to negative

L_wire(j)=L_startLength-(j-1)*L_Inc; %Length of wire from tip of tine
(m)

Wire_Length(j)=L_wire(j); % Array of length of wire at each step.

r=0.000007/2;           %Radius of wire (m)

rho_wire=2634;           %Density of Carbon 2250(kg/m^3)

E_wire=71000000000;      %Young's Modulus of Carbon
235000000000(kg/m*s^2)

%Wire Secondary Calculations

I_wire(j)=(pi*(r^4)/4); %Second Moment of Area for Wire

m_wire(j)=rho_wire*L_wire(j)*pi*(r^2);

%%%% AlphaL Canteliver

```

```

for ii=1:10

    OmegaNFiberCant(j,ii)=(AlphaL(ii)^2)*(sqrt((E_wire*I_wire(j))/(m_wire(j)
    )*L_wire(j)^3)));

end

%Primary Calculations

%omega=((k.tine/m_tine)^(1/2)); %Omega (free state, undamped, natrual
frequency)

%alpha=((rho_wire*(omega^2))/(E_wire*I_wire))^(1/4);

    %Simplification into Alpha (don't need yet)

beta(j)=((m_tine*(L_tine^3)*E_wire*I_wire(j))/(m_wire(j)*(L_wire(j)^3)*
E_tine*I_tine))^(1/4);

    %Simplification into Beta

gamma(j)=L_wire(j)/L_tine;

    %Simplification into Gamma

%psi=alpha*L_wire;

    %Simplification into Psi (dont need because psi is varied)

k(j)=E_wire*I_wire(j);

    %Simplification into K


for i=1:Datapoints %start of for loop to find solution
    (startvalue:stepinterval:endvalue)

    Eigpsi(j,i)=PsiStart+i*PsiInc; % Create array variable psi

% %A1,A2....A6 are rows of matrix A

    B=[0;0;0;0;0;0]; %Define Vector B

    %Inverse=pinv(A)

```

```

D(j,i)=PsiDeterminant(beta(j), Eigpsi(j,i), gamma(j), k(j));

%Find Determinant of Matrix A

EigSlope(j,i)=(SlopeTemp-log(D(j,i)))/PsiInc; % Slope of log of
determinant

SlopeTemp=log(D(j,i)); % Update slope temp.

% if(abs(EigSlope(i))<SlopeTolerance) % search for zero values of log
(D)

if(i==1) % this avoids trying to look at changes in slope for the
first data point in psi

elseif(EigSlope(j,i-1)-EigSlope(j,i)>TransitionPsi) % this searches
for rapid changes in slope above some transition value

    PsiExtractCount=PsiExtractCount+1; % for each etch step

    PsiRoots(j,PsiExtractCount)=Eigpsi(j,i); % drops the roots into
the Jth row

    L_Wire_Extract(j,PsiExtractCount)=L_wire(j); % Not sure whether
this needs to be two dimensional !!!!!!!

    I_Wire_Extract(j,PsiExtractCount)=I_wire(j);

    m_Wire_Extract(j,PsiExtractCount)=m_wire(j);

end

n=n+0.1; %increase "n" by set amount (step value)
and show new "n" (x,y format)

Progress=[100*i/Datapoints 100*j/etchsteps];

end

end

FuncForFzero=@(x) PsiDeterminant(beta(j), x, gamma(j),k(j));

```

```

% Look for zero using fzero Matlab function

TolX=0.00000001;

options=optimset('TolX', 0.000001);

for j=1:etchsteps

    for i=1:PsiExtractCount

        DFzeroRoot(j,i)=fzero(FuncForFzero, PsiRoots(j,i), options);

    end

end

syms  A11 A33 B11 B22 B33 B44

%A11=1;

%EigenVector=[A11;A33;B11;B22;B33;B44];

%for i=20:22

    %EigenVector(i)=[A1(i);A3(i);B1(i);B2(i);B3(i);B4(i)]

    step=49;

    mode=7;

    e=17;

    %A3=1;

    %PsiRoot(e)=DFzeroRoot(e);

    % PsiRoot(e)=PsiRoots(step, mode)

    MA=MatrixA(beta(step),DFzeroRoot(step,mode),gamma(step),k(step));

    % AEigenVector=vpa(MA)*EigenVector;

    %[A1, B1, B2, B3, B4]=solve(AEigenVector(1)==0, AEigenVector(2)==0,
AEigenVector(3)==0,AEigenVector(4)==0,AEigenVector(5)==0,AEigenVector(6
)==0);

    % sol=solve(AEigenVector(1)==0,AEigenVector(2)==0,
AEigenVector(3)==0,AEigenVector(4)==0,AEigenVector(5)==0,AEigenVector(6
)==0);

```

```

    %SS=[SS.A11 SS.B11 SS.B22 SS.B33 SS.B44]

%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% All COEFFICENTS %%%%%%%%%

%A3=1

% SSS(A33)=[SS(1),SS(2),SS(3),SS(4),SS(5)]

[U S V] = svd(vpa(MA));

x = V(:,end)

%SSSS=SSS(A3)

A1=x(1);

A3=x(2);

B1=x(3);

B2=x(4);

B3=x(5);

B4=x(6);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOTTING %%%%%%%%%

L_wiree(1)=.008;

x1=0:.00001:L_tine;

x2=L_tine:.00001:(L_tine+L_Wire_Extract(step,mode));

y1=A1*(cos(beta(step)*gamma(step)*DFzeroRoot(step,mode)*x1/L_Wire_Extra
ct(step,mode))-
cosh(beta(step)*gamma(step)*DFzeroRoot(step,mode)*x1/L_Wire_Extract(ste
p,mode)))+A3*(sin(beta(step)*gamma(step)*DFzeroRoot(step,mode)*x1/L_Wir
e_Extract(step,mode))-

```



```

sinh(beta(step)*gamma(step)*DFzeroRoot(step,mode)*x1/L_Wire_Extract(step,mode)));
y2=B1*cos(DFzeroRoot(step,mode)*x2/L_Wire_Extract(step,mode))+B2*cosh(DFzeroRoot(step,mode)*x2/L_Wire_Extract(step,mode))+B3*sin(DFzeroRoot(step,mode)*x2/L_Wire_Extract(step,mode))+B4*sinh(DFzeroRoot(step,mode)*x2/L_Wire_Extract(step,mode));
[Steps NumRoots]=size(PsiRoots);

for j=1:Steps
    for i=1:36

        Omegas(j,i)=DFzeroRoot(j,i)*DFzeroRoot(j,i)*sqrt(E_wire*I_Wire_Extract(j,i)/(m_Wire_Extract(j,i)*L_Wire_Extract(j,i)^3));
        FreqkHz(j,i)=Omegas(j,i)/(2*pi*1000);

    end
end

z=zeros(100,100);

    for j=1:100        %(j indexes the etch steps)
        for k=1:10      %(k indexes the modes)
            for i=1:100    %(i indexes the frequency)
                if (i==round(FreqkHz(j,k)))
                    z(j,i)=1;
                end
            end
        end
    end

end

```

```

figure(5)

plot(L_Wire_Extract(1:etchsteps,1:15)*1000,FreqkHz(1:etchsteps,1:15),'b
*')

title('Roots (Natural Frequencies)at Diffrent Lenghts', 'FontSize', 16)
ylabel('Frequency (kHz)')
xlabel('Fiber length (mm)')

hold on

plot(Wire_Length*1000,(OmegaNFiberCant)/(2000*pi), 'mo')

hold off


figure(6)

p=mesh(z);

title('3-D plot, Frequency vs Etch Step vs Z value', 'FontSize', 16)

xlabel('Frequency (kHz)')

ylabel('Etch steps')

zlabel('Z')

% axis ([25,36 0,50 0,5]);


aH = ancestor(p,'axes');

x = get(p,'XData');

y = get(p,'YData');

new_y = L_startLength-(y-1)*L_Inc;

z=get(p,'ZData');

figure (7), mesh(x,new_y,z);


save allvariables

```

Matlab program for making list of the probe's experimental frequency responses interactively

```
function list=makelist

% returns a list of files (in a cell array named list), made
interactively

exit=0; % The program quits when exit=1

list={};

while (exit == 0)

%%% SET UP DISPLAY

wd=pwd; % current directory

disp(['Current directory ' wd]);

sz=size(list,1);

if (sz);

disp(' '); disp(['CURRENT LIST: ' num2str(sz) ' entries: ' list{1} '
... ' list{end}]);

else

disp('List: 0 entries')

end

choices={'MENU: '; 's=show list'; 'i=Info about image';...
'd(or ls)=dir'; 'sk=skip'; 'c=cut';...
'cc=include'; 'x=erase list & start over';...
'cd=change directory'; 'e=edit'};

disp(char(choices))

beep

%%% GET INPUT FROM USER

inp=input ('Type base name (wild cards OK) (ENTER=use current
list)\n\n','s');
```

```

switch inp

case '' % ENTER pressed, time to quit program (unless list is empty)

clc;

if ~(isempty(list)); exit=1; end

case 'i' % get information about file(s) in list

prompt=['Which number? (1-' num2str(sz) '; ENTER=all)\n\n'];

inp=input(prompt,'s');

if isempty(inp);

for j=1:sz

try

info=imfinfo(list{j})

catch; disp(['Error reading ' list{j}])

end;

end

else

try

info=imfinfo(list{str2num(inp)})

catch; disp(['Error reading ' list{inp}])

info

end;

end

case 'cd' % change directory

[picpath]=uigetfile('*.','Pick any file in Directory');

cd (picpath) % change to new directory

case 's' % show list

clc

disp(char(list))

```

```

input ('Press ENTER');

case 'ls' % display all files in this directory
ls
disp('Press ENTER'); pause

case 'd' % display all files in this directory
dir; disp('Press ENTER'); pause

case 'c' % Cut files from list that contain string
c=input('Omit if it contains string - type string
(ENTER=abort)\n\n','s');

if ~(isempty(c));
nn=1; list2={};
for j=1:length(list);
if isempty(findstr(c,list{j}));list2(nn)=list(j);nn=nn+1;end
end
end

list=list2';

case 'cc' % CCut files from list that do NOT contain string
c=input('Include if it contains string - type string
(ENTER=abort)\n\n','s');

if ~(isempty(c));
nn=1;list2={};
for j=1:length(list);
if findstr(c,list{j}); list2(nn)=list(j); nn=nn+1; end
end
end

list=list2';

case 'sk' % skip files
inp=input('Take how many, skip how many? (ENTER= 1 1)\n\n','s');
[n1 n2]=strtok(inp,' '); % strtok splits inp at spaces

```

```

% e.g., '1 3' becomes '1' and '3'. Still strings, not numbers

if (isempty (n1)); n1='1'; n2='1';end

nn=0; n1=str2num(n1); n2=str2num(n2);list2={};

for j=1:n1+n2:size(list,1)

for k=1:n1

nn=nn+1;

try;list2(nn)=list(j+k-1);

catch; nn=nn-1; end

end

end

list=list2';

case 'x' % erase list

list=[];

case 'e'

dlmwrite('junk.txt',char(list),'')

edit 'junk.txt' %

input ('Press ENTER when done','s')

list=textread('junk.txt','%s');

otherwise % something was entered

structlist=dir(inp); % DIR returns a structure (4 fields: name, date,
byes, isdir)

celllist={structlist.name}; % make cell array

celllist=celllist'; % 1 row becomes 1 column

list=[list; celllist]; % append to old list

list=sortrows(list); % sort alphabetically

end % switch inp

end % while exit==0

```

Matlab program for generating experimental frequency responses (of probe 16 in this example), 3D plot representing magnitude versus resonant frequency versus length of the fiber.

```
% Program Plot_data_for_probe;

clear all

global fileName len

n=0;

len=0;

MassRemoved=[0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
100 105 110 115 120 130 140 150 160 170 180 190 200 210 220 230 240 250
260 270 280 290 300 310 320 330 340 350 360 370 380 390 400 400 410 420
430 440 450 460 470 480 490 500 510];

MassRemovedMax=510; % Max range for the plotting of mass removal

MagRange=0.01; %plotting range for the frequency response magnitude on
y axis

YmaxAmp=1+MagRange/2;

YminAmp=1-MagRange/2;

XminFreq=26400; % minimum and maximum frequency values for plotting

XmaxFreq= 31100; % max

XfreqDifference=XmaxFreq-XminFreq;

list=makelist;

disp(list); %displays lists on command window

len=length(list); %sets length of list equal to 'len'

ArrayTemp=0; % sets this array indexing parameter to zero to intialize

for n=1:len

    clear Cnew Dnew Bnew Anew A B C D;
```

```

fileName = list{n};

[A,B,C,D] = textread(fileName,'%f %f %f %f');

[Arraylength(n), columns]=size(B);

clear Cnew Dnew Bnew Anew;

for k=1:Arraylength(n)

    Anew(k)=A(k)-A(1000)+1; % Response to vary about a value of
1

    Bnew(k)=B(k)-B(1000)+1;

    Cnew(k)=C(k); % Frequency corresponding to data point

    % attempt to get a 3-D plot

    y(ArrayTemp+k)=MassRemoved(n); % use this to plot dummy y
axis in 3-D plot

    XvaluesFreq(ArrayTemp+k)=Cnew(k);

    ZvaluesAmp(ArrayTemp+k)=Anew(k);

end

progress=n/len

figure(1)

plot(Cnew.'-XminFreq,Anew.') %reads data [A B C D] from
[Amp Phase Freq Temp] in x,y format

axis([0 XfreqDifference YminAmp YmaxAmp])

xlabel('Frequency (Hz)')

set(xaxis,'fontsize',16)

ylabel('Amplitude (V)')

set(yaxis,'fontsize',16)

```



```

        title(fileName)

        M=getframe(gcf);

        hold on

        ArrayTemp=ArrayTemp+Arraylength(n); % this provides an index to
        continue building the 3-D array

    end

    figure(2)

    plot3(XvaluesFreq,y,ZvaluesAmp, '.', 'MarkerSize',4)

    axis([XminFreq XmaxFreq 0 MassRemovedMax YminAmp YmaxAmp])

    xaxisl=xlabel('Frequency (Hz)')

        set(xaxisl, 'fontsize',16)

        yaxisl=ylabel('Mass removed (ug)')

        set(yaxisl, 'fontsize',16)

    zaxisl=zlabel('Amplitude (V)')

    set(zaxisl, 'fontsize',16)

    save CrunchedNumbers

    movie(M,2);

    % movie2avi(M, 'FrequencyPlotHoffprblA')

    % movie(MM,5)

```

Matlab program for calculating the probability, P_T , of finding good pattern versus h/T

when particles are all traveling with the same velocity V_{\max}

```
clear
clc
fontSize=20;
syms R h

P(h,R)=1-(1/pi)*(acos((R-h)/R));

R=1,

for h=0:.01:1
    figure (1)
    plot(h,P(h,R),'*')
    title('Probablity for Vmax', 'FontSize', fontSize);
    xlabel('h/R', 'FontSize', fontSize)
    ylabel('P', 'FontSize', fontSize)
    hold on
end
hold off
```

Matlab program for calculating the probability, P_τ , of finding good pattern versus h/T

when particles are all traveling with a uniformly distributed velocity

```
clear;
```

```

clc;

fontSize=20;

R=1;

hstart=0;

hstop=1;

hdatapoints=200;

hink=(hstop-hstart)/(hdatapoints-1);

for i=1:hdatapoints
    h=hstart+(i-1)*hink;
    y=@(r) 1-(1/pi)*acos((R-h)./r);

    I=quad(y,R-h,R);

    Pt=(R-h+I)/R;

    figure (1)
    plot(h,Pt,'*')
    title('Probablity for Uniform V', 'FontSize', fontSize);
    xlabel('h/R', 'FontSize', fontSize)
    ylabel('PP', 'FontSize', fontSize)
    hold on
end
hold off

```

Matlab program for generating images with random particles (run on UNCC Viper cluster computer).

To access files on cluster drive: Right click on My Computer, Map network drive, enter [\\dviper.unc.edu\users](https://dviper.unc.edu/users) in the folder, then enter unccharlotte-nt\skafashi as username and enter your password. To run programs on cluster, go to Putty, enter viper.unc.edu, then enter your username and password.

```
tic
```

```
clear all;
```

```
close all;
```

```
clc;
```

```
FontSize=16;
```

```
% %%%GUI Parameters
```

```
% prompt = {'Enter marker size:', 'Enter Number of Random  
Particles:', 'Enter Number of Coded Patterns:', 'Enter Maximum  
Angle:', 'Enter Code:', 'Enter Minimum Area:', 'Enter DeCode:', 'Enter  
RmarginStart:', 'Enter RmarginInc:', 'Enter RmarginEnd:', 'Enter  
MaxAngleStart:', 'Enter MaxAngleInc:', 'Enter MaxAngleEnd:', 'Enter  
Iteration Number:'};
```

```
% dlg_title = 'Image Generation';
```

```
% num_lines = 1;
```

```
% def = {'10', '1', '5', '60', '[1 2]', '10', '[1  
2]', '.01', '.05', '1', '10', '10', '60', '5'};
```

```
% options.Resize='on';
```

```
% answer = inputdlg(prompt,dlg_title,num_lines,def,options);
```

```
%
```

```

%%%Convert Strings to Numbers for Parameters

%

% MarkerSize=str2num(answer{1})

% NofRandomParticles=str2num(answer{2})

% NofCodedPatterns=str2num(answer{3})

% MaxAngle=str2num(answer{4})

% Code=str2num(answer{5})

%

% MinArea=str2num(answer{6});

% DeCode=str2num(answer{7});

% RmarginS=str2num(answer{8});

% RmarginI=str2num(answer{9});

% RmarginE=str2num(answer{10});

% MaxAngleS=str2num(answer{11});

% MaxAngleI=str2num(answer{12});

% MaxAngleE=str2num(answer{13});

% IterationNumber=str2num(answer{14});

%%% No Gui parameters

MarkerSize=10;

NofRandomParticles=20;

NofCodedPatterns=0;

MaxAngle=60;

Code=[1 2];

MinArea=10;

DeCode=[1 3];

RmarginS=.01;

```

```

RmarginI=.01;

RmarginE=1;

MaxAngleS=10;

MaxAngleI=20;

MaxAngleE=360;

IterationNumber=10;

for ImageNumber=1:IterationNumber

    %%(Random Particles Generation)

    if (NofRandomParticles==0)

        k=figure(1) % Image with trajectories

        plot(NaN, NaN);

        title('Image with trajectories', 'FontSize', FontSize);

        daspect([1,1,1]);

        axis ([0,1 0,1]);

        hold on
    else

        for i=1:NofRandomParticles

            x(i)=rand()

            y(i)=rand()

            k=figure(1) % Image with trajectories

            plot(x(i),y(i), '.', 'MarkerSize', MarkerSize);

```

```

        title('Image with trajectories', 'FontSize', FontSize);

        daspect([1,1,1]);

        axis ([0,1 0,1]);

        hold on

        Xvalues(i)=x(i);

        Yvalues(i)=y(i);

    end

end

```

```

%%%(Coded Patern Generation)

[m,n]=size(Code)

betha=zeros(NofCodedPatterns,n);

%% i representing the number of coded patterns
%% j representing the particles inside the pattern

```

```

if (NofCodedPatterns==0)

    h=figure(2) % Image with trajectories

    plot(NaN, NaN);

    daspect([1,1,1]);

    axis ([0,1 0,1]);

    hold on

else

    for i=1:NofCodedPatterns

```

```

        t(i)=rand()/5;

        X(i,1)=rand();

        Y(i,1)=rand();

```

```

Xvalues(i+NofRandomParticles+(i-1)*n)=X(i,1)
Yvalues(i+NofRandomParticles+(i-1)*n)=Y(i,1)

for j=2:n+1
    betha(i,j)=(MaxAngle*2*pi/360).*rand()+betha(i,j-1);
    X(i,j)=Code(j-1)*t(i)*cos(betha(i,j))+X(i,j-1)
    Y(i,j)=Code(j-1)*t(i)*sin(betha(i,j))+Y(i,j-1);

    Xvalues(i+j+NofRandomParticles-1+(i-1)*n)=X(i,j);
    Yvalues(i+j+NofRandomParticles-1+(i-1)*n)=Y(i,j);

end

whitebg([0 0 0]);
set(gca,'Color',[0 0 0]);
set(gcf,'Color',[0 0 0]);
axis off;

k=figure(1) % Image with trajectories
plot([X(i,:)],[Y(i:)],'g--o')
hold on

h=figure(2) %Image without trajectories
plot(X(i,1),Y(i,1),'.','MarkerSize',
MarkerSize,'Color',[0.313725501298904 0.313725501298904
0.313725501298904])
axis ([0,1 0,1]);
hold on

```



```

        for j=2:n+1

            plot(X(i,j),Y(i,j),'.','MarkerSize',
MarkerSize,'Color',[0.313725501298904 0.313725501298904
0.313725501298904]))

        end

    end

end

if (NofRandomParticles==0)
    plot(NaN, NaN);
else
    for i=1:NofRandomParticles

        plot(x,y,','MarkerSize', MarkerSize);

    end

end

hold off

[a NumberOfObjects]=size(Xvalues);

ImageFileName=[ 'Density', num2str(NumberOfObjects), ',Angle',
num2str(MaxAngleS), '-',num2str(MaxAngleE), ',dR', num2str(RmarginS), '-
',num2str(RmarginE), ',DeCode',num2str(DeCode), ',#',num2str(ImageNumber)
, '.jpg'];

whitebg([0 0 0]);

```

```

set(gca,'Color',[0 0 0]);
set(gcf,'Color',[0 0 0]);
axis off;

daspect([1,1,1]);
set(gca,'position',[0 0 1 1],'units','normalized')
set(gcf, 'InvertHardCopy', 'off');
axis off;
saveas(gcf,ImageFileName);

%saveas(k, 'Image With Trajectories.jpg', 'jpg');

DecodingAutomatedNumbers

end

xlwrite(filename,Data);
toc

```

Matlab program for finding encoded pattern in randomly generated images. Decoding strategy of [1 3] is shown here as an example (run on computer cluster)

```

MarkerSize=20;
FontSize=16;
TimeInterval=0.002; % seconds

[m,n]=size(DeCode);

%%%%% Encoding Pattern%%%%%%%%%

```

```

[a NumberOfObjects]=size(Xvalues);

for i=1:NumberOfObjects
    for j=1:NumberOfObjects

        R(i,j)=norm([Xvalues(i) Yvalues(i)]-[Xvalues(j) Yvalues(j)]);

    end
end

figure(4)
plot(Xvalues(1),Yvalues(1),'.','MarkerSize', MarkerSize)
hold on
axis off;
whitebg([0 0 0]);
set(gca,'Color',[0 0 0]);
set(gcf,'Color',[0 0 0]);
daspect([1,1,1]);

hold on

for i=2:NumberOfObjects
    plot(Xvalues(i),Yvalues(i),'.','MarkerSize', MarkerSize)
end

%c=0;

```

[illegible]

```

v1=[Xvalues(j) Yvalues(j)]-
[Xvalues(i) Yvalues(i)];

v2=[Xvalues(k) Yvalues(k)]-
[Xvalues(j) Yvalues(j)];

Angle1=acosd(dot(v1,v2)/(norm(v1)*norm(v2)));

if (Angle1<MaxAngle)
    c=c+1;
    R(i,j);
    R(j,k);
    X=[Xvalues(i), Xvalues(j),
Xvalues(k)];
    Y=[Yvalues(i), Yvalues(j),
Yvalues(k)];

    plot(X,Y,'color', [rand()
rand() rand()]])

end
end
end
end
case 3
for l=1:NumberOfObjects
    if (DeCode(2)*R(i,j)>DeCode(1)*R(j,k)-
(Rmargin/100)*R(i,j) &&
DeCode(2)*R(i,j)<DeCode(1)*R(j,k)+(Rmargin/100)*R(i,j))

```

```

                                if
(DeCode(3)*R(j,k)>DeCode(2)*R(k,l)-(Rmargin/100)*R(j,k)&&
DeCode(3)*R(j,k)<DeCode(2)*R(k,l)+(Rmargin/100)*R(j,k))
                                if (R(i,j)~=0 && R(j,k)~=0)
                                    if (R(k,l)~=0)
                                        if (i~=j && j~=k)
                                            if (k~=1)
                                                v1=[Xvalues(j)
Yvalues(j)]-[Xvalues(i) Yvalues(i)];
                                                v2=[Xvalues(k)
Yvalues(k)]-[Xvalues(j) Yvalues(j)];
                                                v3=[Xvalues(l)
Yvalues(l)]-[Xvalues(k) Yvalues(k)];
Angle1=acosd(dot(v1,v2)/(norm(v1)*norm(v2)));
Angle2=acosd(dot(v2,v3)/(norm(v2)*norm(v3)));
                                if
(Angle1<MaxAngle && Angle2<MaxAngle )
                                    c=c+1;
                                    R(i,j);
                                    R(j,k);
                                    R(k,l);
X=[Xvalues(i), Xvalues(j), Xvalues(k), Xvalues(l)];
Y=[Yvalues(i), Yvalues(j), Yvalues(k), Yvalues(l)];

```



```

                                if (i~=j &&
j~=k)

                                if (k~=l &&
l~=m)

v1=[Xvalues(j) Yvalues(j)]-[Xvalues(i) Yvalues(i)];

v2=[Xvalues(k) Yvalues(k)]-[Xvalues(j) Yvalues(j)];

v3=[Xvalues(l) Yvalues(l)]-[Xvalues(k) Yvalues(k)];

v4=[Xvalues(m) Yvalues(m)]-[Xvalues(l) Yvalues(l)];

Angle1=acosd(dot(v1,v2)/(norm(v1)*norm(v2)));

Angle2=acosd(dot(v2,v3)/(norm(v2)*norm(v3)));

Angle3=acosd(dot(v3,v4)/(norm(v3)*norm(v4)));

                                if

(Angle1<MaxAngle && Angle2<MaxAngle )

                                if

(Angle3<MaxAngle)

c=c+1;

R(i,j);

R(j,k);

```


$R(k, l);$
$$R(l, m);$$

```
X=[Xvalues(i), Xvalues(j), Xvalues(k), Xvalues(l), Xvalues(m)];
```

```
Y=[Yvalues(i), Yvalues(j), Yvalues(k), Yvalues(l), Yvalues(m)];
```

```
plot(X,Y, 'color', [rand() rand() rand()])
```

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

```

hold off

%%%% Saving Statistical Data

Data(aa+1,:,Sheet) = [NumberOfObjects MaxAngle Rmargin DeCode
c];

%           Row=['A', num2str(aa+1),':F', num2str(aa+1)];
%           xlwrite(filename,Data,Sheet,Row);

end

end

% xlwrite(filename,Data);

```

Matlab program for reading ePTV excel files produced by the above two programs and

plotting the 3D graphs (\bar{N}_s versus θ and γ)

```

clear

clc

fontSize = 13;

name='Density20,Angle10-360,dR0.01-1,DeCode1 3'

filename=[name, '.xlsx'];

Iteration=10;

```

```

Angle=xlsread(filename,'dimension=1','B2:B1801');

dR=xlsread(filename,'dimension=1','C2:C1801');


for i=1:Iteration

    Sheet=['dimension=',num2str(i)];

    N(:, :, i)=xlsread(filename,Sheet,'F2:F1801');
end

NMean=mean(N,3);


h=figure;

plot3(dR,Angle,NMean,'*');

title(name, 'FontSize', fontSize)

xlabel('dR', 'FontSize', fontSize);

ylabel('Angle', 'FontSize', fontSize);

zlabel('N of Found', 'FontSize', fontSize);


savefig(h,[name, '.fig']);

```

Matlab program for calculating edge effect value α in ePTV results

```

clear

clc

fontSize = 13;

Density=20;

name='Density20,Angle10-360,dR0.01-1,DeCode1 3';

filename=[name, '.xlsx'];

```

```

Iteration=10;

Angle=xlsread(filename,'dimension=1','B2:B1801');
AngleRadian=Angle*pi/180;
dR=xlsread(filename,'dimension=1','C2:C1801');

for i=1:Iteration
    Sheet=['dimension=',num2str(i)];
    N(:, :, i)=xlsread(filename,Sheet,'F2:F1801');
end

NMean=mean(N,3);

j=0;
syms alpha;
for k=1:10:1800
    i=0;
    j=j+1;
    for R=.1:.1:.4
        i=i+1;
        NTheory(j,i)=AngleRadian(k)*Density*(dR(k)/100)*R^2;
    end

    M(j)=mean(NTheory(j, :));

    Objective(j)=M(j)*Density*alpha;
    alphaFunction(j)=Objective(j)-NMean(k);

```

```

Alpha(j)=solve(alphaFunction(j)==0, 'alpha');

end

double(Alpha);
min=min(double(Alpha))
max=max(double(Alpha))
MeanAlpha=double(mean(Alpha))

Matlab program for calculating theoretical number of found in ePTV

clear;
clc;

name='Density90,Angle10-360,dR0.01-1,DeCode1 2';
filename=[name, '.xlsx'];

Density=90;
Rend=0.5;
dR=1;
j=0;

for dR=0.01:.01:1

    for Angle=10:20:360
        i=0;
        j=j+1
    end
end

```

```

for R=.1:.1:Rend
    i=i+1;
    NTheory(j,i)=(Angle*pi/180)*Density*(dR/100)*R^2;

end

M(j)=mean(NTheory(j,:));

Alpha=(0.0001*Density^2-0.005*Density)*(Angle*pi/180)^2-(-8e-
5*Density^2+.0048*Density)*(Angle*pi/180)+0.0025*Density^2-
0.0833*Density;

NFound(j)=M(j)*Density*Alpha;

end

end

Nt=transpose(NFound);

%xlswrite(filename,'New Nt','dimension=10','J1')

xlswrite(filename,Nt,'dimension=10','J2:J1801')

```

Matlab program for calculating the statistical analysis between the simulated and theoretical result of ePTV

```

tic
clear
clc

fontSize = 13;

name='Density50,Angle10-360,dR0.01-1,DeCode1 2'%%%File Name

filename=[name, '.xlsx'];

Iteration=10; %%% Number of Sheets in Excel

Density=90; %% Denisty

Rend=0.5;

```

```

Angle=xlsread(filename,'dimension=1','B2:B1801'); %%% Read Angle Data
AngleRadian=Angle*pi/180;

dR=xlsread(filename,'dimension=1','C2:C1801'); %%% Read dR Data

NTheory=xlsread(filename,'dimension=10','K2:K1801');

%%%%Ns Mean Calculation

for i=1:Iteration

    Sheet=[ 'dimension=',num2str(i)];

    N(:, :, i)=xlsread(filename,Sheet,'F2:F1801');

end

NMean=mean(N,3); %%% Get Simulated N of Found Mean

xlswrite(filename,{ 'Ns Found Mean'}, 'dimension=10', 'H1') %%% Write
Simulated N of Found Mean

xlswrite(filename,NMean, 'dimension=10', 'H2:H1801')

%%%% Alpha Calculation

j=0;

syms alpha;

for k=1:1:1800

    i=0;

    j=j+1;

    for R=.1:.1:.5

        i=i+1;

        NTheory(j,i)=AngleRadian(k)*Density*(dR(k)/100)*R^2;

    end

    M(j)=mean(NTheory(j, :));

```

```

Objective(j)=M(j)*Density*alpha;

alphaFunction(j)=Objective(j)-NMean(k);

Alpha(j)=solve(alphaFunction(j)==0, 'alpha');

end

AlphaT=transpose(double(Alpha));

xlswrite(filename,{'Alpha'}, 'dimension=10', 'I1')%% Write Alpha
xlswrite(filename,AlphaT, 'dimension=10', 'I2:I1801')

%%Coefficients a1,a2,a3 Calculation
for j=1:99
    AngleStart=j*18+1
    AngleEnd=AngleStart+17
    c=0;
    for i=AngleStart:AngleEnd
        c=c+1
        X(j,c)=Angle(i)*pi/180;
        Y(j,c)=AlphaT(i);
    end
    p(:, :, j)=polyfit(X(j, :), Y(j, :), 2);
    Y1(j)=p(1, 1, j);
    Y2(j)=p(1, 2, j);
    Y3(j)=p(1, 3, j);
    dR1(j)=dR(AngleStart);
end

```



```

a1T=Y1(1,10:99);

a1=mean(a1T);

a2T=Y2(1,10:99);

a2=mean(a2T);

a3T=Y3(1,10:99);

a3=mean(a3T);

plot(X(1,:), Y(1,:));

hold on

for j=10:10:99

    plot(X(j,:), Y(j,:));

end

%%%% Theoretical N FFound Calculation

j=0;

for dR=0.01:.01:1

    for Angle=10:20:360

        i=0;

        j=j+1;

        for R=.1:.1:Rend

            i=i+1;

            NTheory(j,i)=(Angle*pi/180)*Density*(dR/100)*R^2;

        end

        M(j)=mean(NTheory(j,:));

        Alpha=a1*(Angle*pi/180)^2+a2*(Angle*pi/180)+a3;

        NFound(j)=M(j)*Density*Alpha;

    end

end

Nt=transpose(NFound);

```

```

xlswrite(filename,{'NTheory'},'dimension=10','K1')%%% Write N Theory

xlswrite(filename,Nt,'dimension=10','K2:K1801')

%%%% Deviation Calculation

Deviation=Nt-NMean;

xlswrite(filename,{'Deviation'},'dimension=10','L1')%%% Write Deviation

xlswrite(filename,Deviation,'dimension=10','L2:L1801')

%%%% Chi-Squared Calculation

Chi=(Deviation.*Deviation)./NMean;

xlswrite(filename,{'Chi-Squared'},'dimension=10','M1')%%% Write Chi-
Squared

xlswrite(filename,Chi,'dimension=10','M2:M1801')

%%%% Standard Deviation and Normal Variate Calculation

for i=1:1800

    for j=1:Iteration

        S(i,j)=N(i,1,j)-NMean(i);

%         Sigma(i)=Sigma(i)+S(i,j)^2;

    end

    Sigma(i)=sum(S(i,:).^2);

    StandardDivation(i)=sqrt(Sigma(i)/9);

    Z(i)=(NFound(i)-NMean(i))/StandardDivation(i);

end

StandardDivationT=transpose(StandardDivation);

ZT=transpose(Z);

xlswrite(filename,{'Standard Deviation'},'dimension=10','N1')%%% Write
Standard Deviation

xlswrite(filename,StandardDivationT,'dimension=10','N2:N1801')

xlswrite(filename,{'Normal Variate Z'},'dimension=10','O1')%%% Write Z

xlswrite(filename,ZT,'dimension=10','O2:O1801')

```

toc

Matlab program for converting videos to image frames

```
vid=mmreader('Test5.avi');

numFrames = vid.NumberOfFrames;

n=numFrames;

for i = 1:2:n

frames = read(vid,i);

imwrite(frames,['Image' int2str(i), '.png']);

im(i)=image(frames);

end
```

Matlab program for making videos from image using filtering by subtracting two images

```
clear,

clc,

j=0;

for i=1:2:100

    j=j+1;

    filename=['Image',num2str(i), '.png'];

    ie(:, :, j)=imread(filename);

end

cmap = gray(256);

for i=1:49

    ieFiltered(:, :, i)=ie(:, :, i+1)-ie(:, :, i);

    f(i)=im2frame(ieFiltered(:, :, i),cmap);

end
```

```

% for i=1:10

%     filename=[num2str(i*10),'.fig'];

%     openfig(filename);

%     set(gcf, 'units','normalized','outerposition',[0 0 1 1]);

%     zlim([0 300000]);

%     ylim([0 360])

%     xlim([0 .39])

%     f(i)=getframe(gcf);

% % close all

% end

```

```

movie2avi(f, 'Test5Filtered.avi', 'fps', '1')

```

Matlab program for calculating velocities of particles

```

clear all;

close all;

clc;

FontSize=16;

MinArea=80;

TimeInterval=0.002; % seconds

PixPerMM=288;

%%% 288 pixels is 1 mm in 1.4x zoom in %%%%%%

%%%%% Show Original Image %%%%%%%%%%

Image1=imread('t,2t pattern Rev2.png');

subplot(2, 2, 1);

imshow(Image1);

title('Original Image', 'FontSize', FontSize);

```

```

##### Black and White Image#####

BlackWhiteImage=Image1(:, :, 1)+Image1(:, :, 2)+Image1(:, :, 3);

imshow(BlackWhiteImage)

hist=imhist(BlackWhiteImage);

#####Binary Image #####

BinaryImage=im2bw(BlackWhiteImage,.3);

subplot(2, 2, 2);

imshow(BinaryImage);

title('Binary Image', 'FontSize', FontSize);


##### Fill Holes#####

FilledImage=imfill(BinaryImage, 'holes');

subplot(2, 2, 3);

imshow(FilledImage);

title('Filled Image', 'FontSize', FontSize);


#####Remove Smaller Areas#####

cc1=bwconncomp(FilledImage);

states=regionprops(cc1, 'Area');

idx=find([states.Area]>MinArea);

AreaFiltteredImage=ismember(labelmatrix(cc1), idx);

subplot(2, 2, 4);

imshow(AreaFiltteredImage);

title('Area Filttered Image', 'FontSize', FontSize);


##### Show only two Particles#####

cc2=bwconncomp(AreaFiltteredImage);

```

```

Label=labelmatrix(cc2);

figure,

Im1=Label==5;

imshow(Im1);

title('Particle Labeled #1', 'FontSize', FontSize);

figure,

Im2=Label==9;

imshow(Im2);

title('Particle Labeled #2', 'FontSize', FontSize);

figure,

ImCombined=Im1+Im2;

imshow(ImCombined);

hold on

title('Particles Combined', 'FontSize', FontSize);

%%%% Calculate Centroids %%%

Cent1=regionprops(Im1, 'centroid');

centroid1=cat(1, Cent1.Centroid);

plot(centroid1(:,1),centroid1(:,2), 'b*')

Cent2=regionprops(Im2, 'centroid');

centroid2=cat(1, Cent2.Centroid);

plot(centroid2(:,1),centroid2(:,2), 'b*')

%%%% Calculate Velocity %%%

R=norm((centroid2-centroid1));

X=(R/PixPerMM); %%%Distance in mm

Velocity=X/TimeInterval, disp(sprintf('mm/s')) %%% mm/sec

plot([centroid1(:,1),centroid2(:,1)], [centroid1(:,2),centroid2(:,2)]);

```

```
hold off
```

Matlab program for filtering with thresholding

```
clc;      % Clear the command window.

close all; % Close all figures (except those of imtool.)

imtool close all; % Close all imtool figures if you have the Image
Processing Toolbox.

clear; % Erase all existing variables. Or clearvars if you want.

format long g;

format compact;

fontSize = 18;

folder = 'U:\Sajad\Project\Particle Manipulation\PIV Matlab Program';
baseFileName = 'Image258.png';

% Get the full filename, with path prepended.
fullFileName = fullfile(folder, baseFileName);

if ~exist(fullFileName, 'file')

    % Didn't find it there. Check the search path for it.
    fullFileName = baseFileName; % No path this time.

    if ~exist(fullFileName, 'file')

        % Still didn't find it. Alert user.
        errorMessage = sprintf('Error: %s does not exist.',
fullFileName);

        uiwait(warndlg(errorMessage));

        return;
    end
end
```

```

rgbImage = imread(fullFileName);

% Get the dimensions of the image.  numberOfColorBands should be = 3.
[rows, columns, numberOfColorBands] = size(rgbImage);

% Display the original color image.
subplot(2, 3, 1);
imshow(rgbImage);

%axis on;

title('Original Image', 'FontSize', fontSize);

% Enlarge figure to full screen.
set(gcf, 'units','normalized','outerposition',[0 0 1 1]);

% Convert to gray scale.
[rows columns numberOfColorChannels] = size(rgbImage);

if numberOfColorChannels > 1
    grayImage = rgb2gray(rgbImage);
else
    grayImage = rgbImage; % It's already gray.
end

% grayImage = rgb2gray(rgbImage);
subplot(2, 3, 2);
imshow(grayImage);

%axis on;

title('Gray Image', 'FontSize', fontSize);

% Let's compute and display the histogram.
[pixelCount, grayLevels] = imhist(grayImage);

subplot(2, 3, 3);
bar(grayLevels, pixelCount);

```



```

grid on;

title('Histogram', 'FontSize', fontSize);

xlim([0 grayLevels(end)]); % Scale x axis manually.

% Threshold to get binary image
binaryImage = grayImage > 74;

subplot(2, 3, 4);

imshow(binaryImage);

%axis on;

title('Binary Image', 'FontSize', fontSize);

% Dilate the binary image to tell it what pixels should be used to fill
in holes.

DilateImage = imdilate(binaryImage, true(11));

subplot(2, 3, 5);

imshow(DilateImage);

%axis on;

title('Dilated Binary Image', 'FontSize', fontSize);

drawnow;

% promptMessage = sprintf(' filling the white areas.\nNote: may take
some time.');
```

```

% titleBarCaption = 'Continue?';

% button = questdlg(promptMessage, titleBarCaption, 'Continue',
'Cancel', 'Continue');
```

```

% if strcmpi(button, 'Cancel')

%     return;

% end
```

```

% tic

% Extract the individual red, green, and blue color channels.
if numberOfColorChannels > 1
    redChannel = rgbImage(:, :, 1);
    greenChannel = rgbImage(:, :, 2);
    blueChannel = rgbImage(:, :, 3);

    % Fill holes
    redChannel = roifill(redChannel, DilateImage);
    greenChannel = roifill(greenChannel, DilateImage);
    blueChannel = roifill(blueChannel, DilateImage);

    % Recombine separate color channels into a single, true color RGB
    image.

    rgbImage = cat(3, redChannel, greenChannel, blueChannel);
else
    redChannel = rgbImage(:, :, 1);
    redChannel = roifill(redChannel, binaryImage);

    b=redChannel<100;

    p=find(b==0);

    redChannel(p)=30;

    biaryImage(p)=0;
figure; imshow(biaryImage);

    rgbImage = cat(1, redChannel);
end

FilteredName=fullfile(folder, 'Filtered.png')

```

```
imwrite(binaryImage,FilteredName);

%subplot(2, 3, 6);

%imshow(rgbImage);

%axis on;

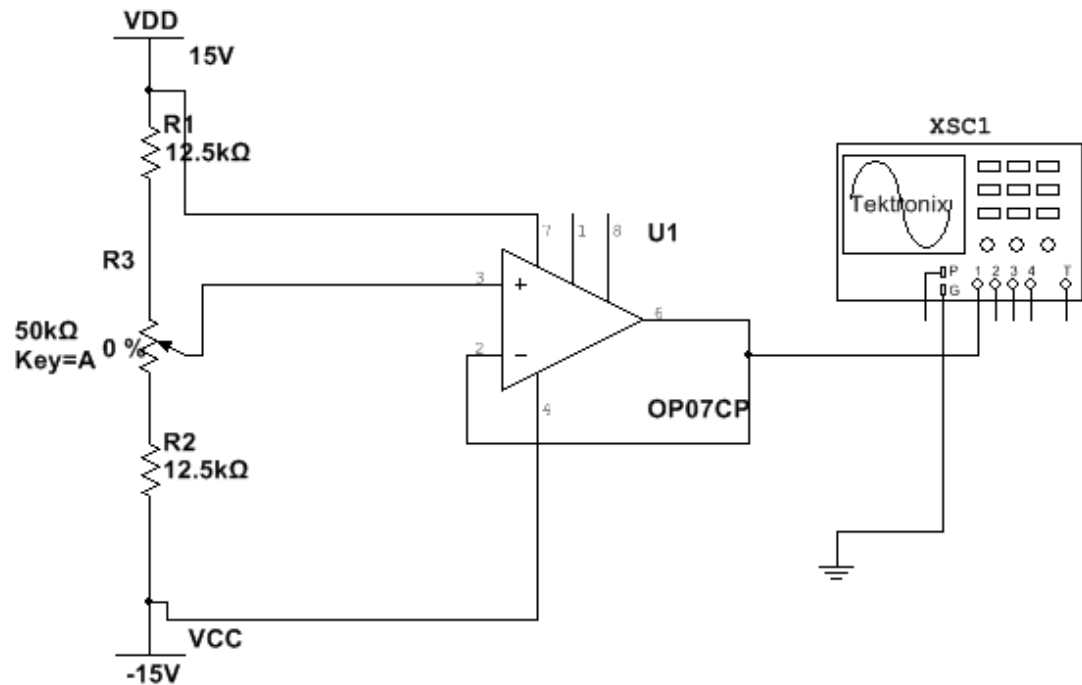
title('Filtered Image', 'FontSize', fontSize);

toc;

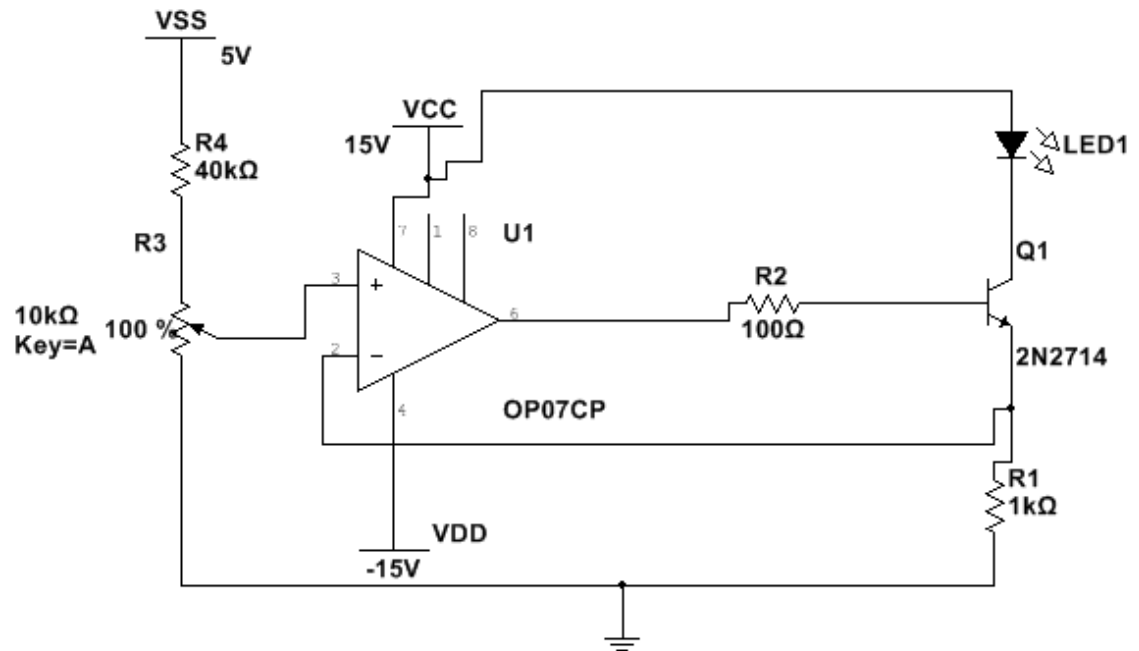
msgbox('Done');
```

APPENDIX C: CIRCUITS AND NOTEBOOK

Reservoir height adjustment measurement Multisim circuit. Displacement is measured using a linear potentiometer.



LED illumination current control Multisim model. Potentiometer is used to control current through the LED.



Instruction on how to setup a Real-Time system using Labview

TITLE		Project No.	3
From Page No.		Book No.	Confidential
<p>Purpose: The purpose of this Experiment is to setup a real time desktop for particle manipulation project to communicate with the labview program in Real time.</p>			
<p>Equipment used: - Flash Drive 1 GIG</p> <ul style="list-style-type: none">- PC Core2 duo- FCEI 1394 FireWire- Labview and MAX (2010)- DAQ card- Ethernet cable (crossover)			
<p>Procedure: Step 1: We put the DAQ card and FireWire inside the PC Case.</p> <p>Step 2: Setting up the utility USB drive (USB flash drive should be less than 2 GB and formatted as FAT or FAT32)</p> <p>MAX → Tools → RT disk Utilities → Create desktop PC utility USB drive</p> <p>Step 3: preparing the desktop system to boot from the USB drive</p> <p>change the BIOS setting to boot from USB drive</p>			

4	Project No. _____	TITLE <u>RT computer Setup</u>
Confidential	Book No. _____	
From Page No. _____		

Then format hard disk
(choose Erase all partitions and create a single new partition)
Type selection: select Reliance

Step 4: on MAX, change the IP of your
RT system under "Remote system" option
to the IP as shown on RT system.
Right click and Install the software

Step 5: using crossover cable, connect the
RT computer to Labview computer.

Conclusion; ~~re~~ The camera (SONY XCD-SX910CR)
has been tested on MAX successfully.

Operational guideline for running an experiment in Labview project

Purpose: Is to build a real time LabView project to be able to see Camera Image on host computer

Procedure: - Go to File → New → Real-Time project
- Name the project and select project folder
- Select Sajad's Real Time System for target under Real-Time Desktop
- There are two computers: Host (which is my computer and Target which is real Time computer)

File Saved on: C: \ Sajad \ FirstProjectEffort \ CameraProject

- Right click on Sajad Real Time System (target computer) and select New → Variable
under Data Type select Image and name it (Sony Camera Image)
- You can Drag it to block Diagram.
- In target program write the simple program for capturing Image

6 Project No. _____ TITLE LabView Programming
 Confidential Book No. _____
 From Page No. _____

Steps in Vision Emotion:

open Camera → Configure Acquisition → start Acquisition
 → Get Image! → stop Acquisition → unconfigure Acq
 loop → close Camera → General error handler

In Front Panel Add a Display Image under Vision
 and connect it to Image out

Note: Always first run target program then host.
 Note: Select Camo for open camera. which should
 be shown on max → Sajads RealTimeSystem →
 Devices & Interfaces → NI-Imagdx Devices

Conclusion: Camera Image was Successfully shown
 on the program while running.

IP address: 169.254.22.241

Instruction on building a piezo-actuator amplifier

TITLE <u>Piezo Amplifier</u>		Project No. _____	23
From Page No. _____		Book No. _____	Confidential
<p>Abstract: The goal of this experiment is to build a high voltage amplifier that can drive piezo electrics attached to the fibers.</p>			
<p>Equipments:</p> <ul style="list-style-type: none"> - Instrumentation Amplifier (AD 624) (unit: 1) - OP-Amp (741 op) (unit: 1) - Transistors (TIP 120) (unit: 2) - power supply (15 Volts and 130 V) 			
<p>File saved on my computer: C:\Sajad\PiezoAmplifier</p>			
<p>Procedure: The circuit is shown in Figure(1), we amplify the signal (coming from dds function Generator) in two stages. First with a Instrumentation Amplifier. The gain in this stage is 5. ($R_G = 10K$)</p> $\text{Gain}_1 = \frac{40K}{R_G} + 1 = \frac{40K}{10K} + 1 = 5$			

24

Project No. _____

Book No. _____

TITLE

Piezo Amplifier

Confidential

Form Page No. _____

Second, the signal is amplified with a 741 OP-Amp. The gain of this amplification is 38.

$$\text{Gain}_2 = 1 + \frac{R_f}{R} = 1 + \frac{3700\text{ K}}{100\text{ K}} = 38.$$

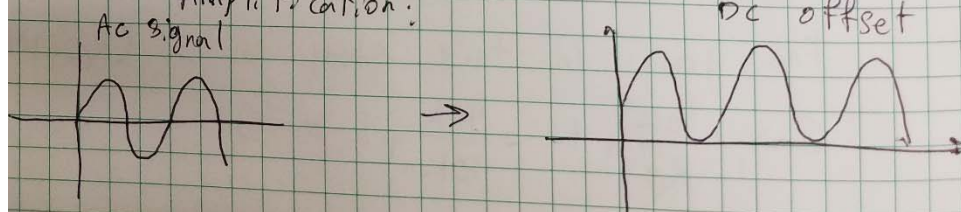
$$\text{Total Gain} = 38 \times 5 = 190$$

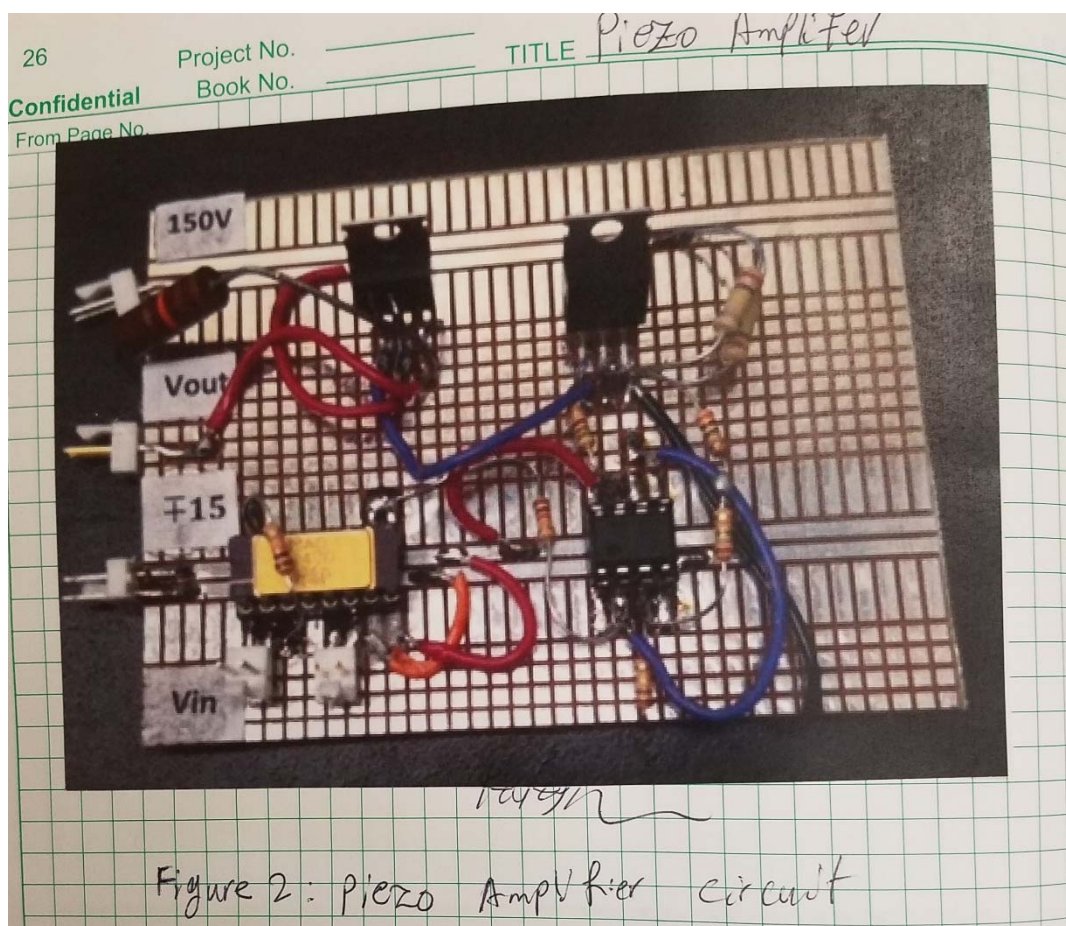
The power supply (Institu Tec) goes up to 128 Volts.

MultiSim file name Saved:

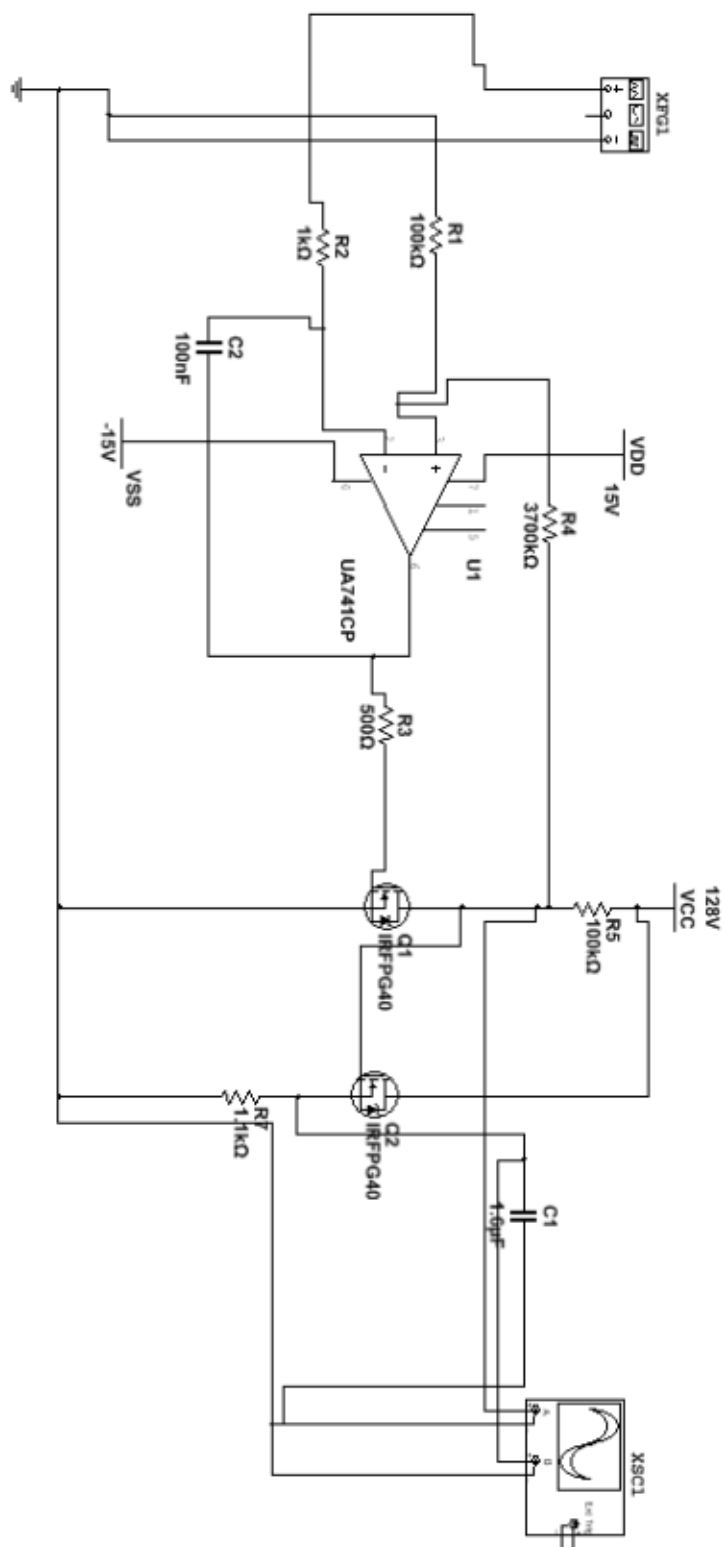
C: / Sajad / Piezo Amplifier / PiezoAmplifierEmitterFollowerGain190

Note: The input signal should start from 0. So, appropriate offset must apply to the input signal in order to get the correct Amplification.





Piezo Amplifier Multisim circuit. (Emitter-follower, Gain 190 with input from Saji Board)



Setup and procedure of attaching fiber to tuning fork

TITLE microscope setup, attaching Fiber on TF Project No. _____ 11
From Page No. _____ Book No. _____ Confidential

purpose: In this experiment, which has two parts, first we attach a caliper on the microscope frame, and then we attach a Tungsten fiber on the top surface of Tuning Fork

Part 1:

In order to be able to measure the length of fiber under microscope we needed a caliper. we glue two needle of caliper on moving and stationary parts of microscope fram. (using Epoxy)

Figure below shows the setup:

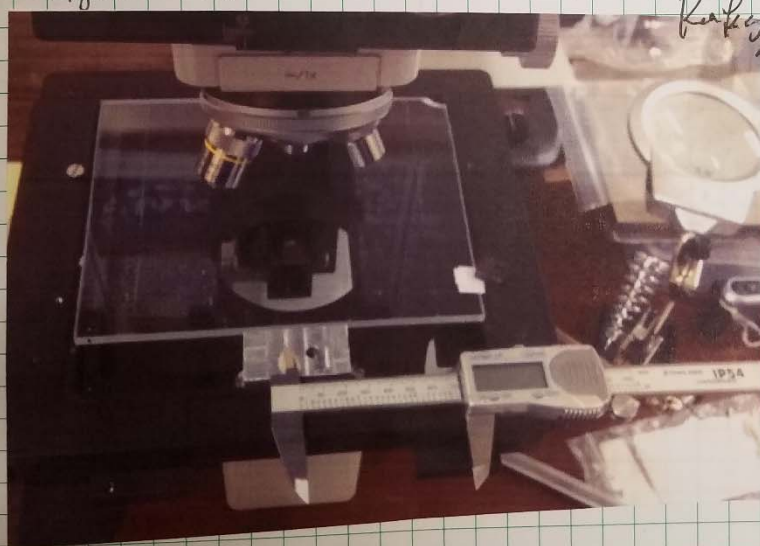
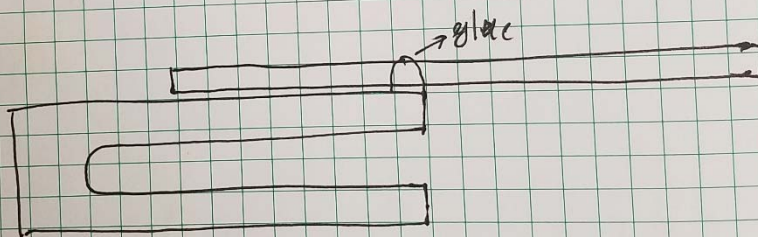


Fig 1: attaching
micrometer on
microscope setup

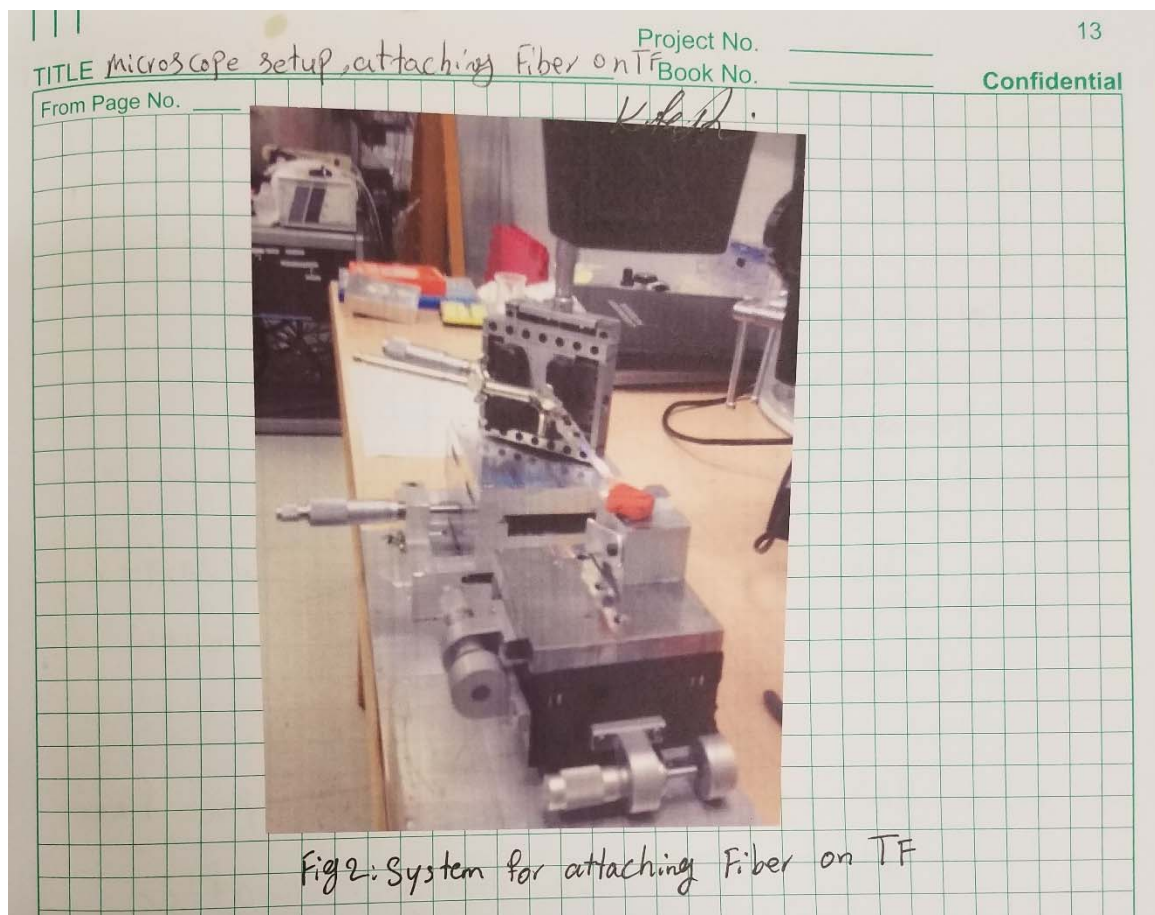
To Page No. _____

Part 2: In this part using a system developed by Jeffrey H. we carefully attached a tungsten fiber on the tiny surface of Tuning Fork.

- procedure :
1. Cut the fiber to about 11 mm
 2. Grab it by Q-tip and fix it on the gripper
 3. using different adjustments in different angles and axes, put the fiber close to the top surface of tuning fork exactly in the middle.
 4. put very tiny amount of glue on the position close to the edge (shown in fig below)



5. put the glue on the TF under the UV (ultraviolet) light



Purpose: The purpose of this experiment is to attach Tungsten fiber on TF using UV glue.

Equipments: - Norland Optical Adhesive 61 Lot 293 (ultraviolet curing)
 - LAMP ASSY, MINI model UVL4 95-0125-05
 - Tungsten CARBIDE SCISSOR (DR SLICK, SAP4TCBL)
 - Tungsten Wire W99.9, 0.075 mm (Good Fellow)

procedures: 1- Cut the fiber at about 10 mm

2- Using the system shown on Fig 2 (page 13)

bring fiber close enough on TF top surface.

Note that fiber should be exactly in the middle of TF surface.

- Make the glue ready to use. put very tiny amount of glue close to the tuning fork edge.

- Hold UV light close to glued part for 20 min. Then you can the gripper back. The fiber should stick on TF.

- Then put the tuning Fork with attached fiber under UV light for 24 hours to be well cured.

TITLE Attaching fiber on TF using epoxy
 From Page No. _____

Project No. _____

19

Book No. _____

Confidential

purpose, same as lab 7 (page 17) but this time we use epoxy instead of UV glue

equipments used: - LOCTITE M-20CL (Hysol) epoxy
 - LOCTITE M-31CL (Hysol) epoxy
 - Fische Scientific oven
 - Tungsten wire W 99.9, 0.075 mm Diameter (Goodfellow)
 - Tungsten carbide scissor, Dr Slick (SAP 4TCBL)

Procedure: 1. follow the steps 1, and 2 in procedure (page 17)
 2. make a small amount of epoxy, mix them well for at least 2 minutes.
 3. put very tiny amount of glue on the TF
 4. leave it for 24 hours in temperature of room
 5. put the TF in oven for 6 hours at 70°C

I made one (TF + fiber) using M20CL, and one with M-31CL, named #1 and #2 respectively

TITLE Trimmed Fiber w/ data

Project No. _____

21

From Page No. _____

Book No. _____

Confidential

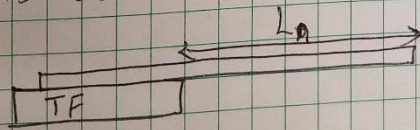
Purpose: The purpose of this experiment is to get frequency response of Tungsten Fiber attached on TF in different length. After getting data from signal Analyzer, we can plot all frequency responses in one figure and get the rainfall of them.

Equipments used:

- W Fiber attached on Tuning Fork
- TS View Camera (OPTICS CAM)
- W Carbide Scissor (SAP 4TCBL)
- Leitz Microscope
- HP Dynamic signal Analyzer
- MATLAB software
- LOCTITE M-31 CL (Hysol)

Process Diagram: Attach Fiber on TF → Connect to signal Analyzer → get frequency Response → Cut the length in step of 0.2 mm → Repeat 3 & 4 till close to 4.8 mm → Save Trace on discet → import data to matlab program → get rain flow

Proceduere: After I attached ~~gave~~ W-Fiber on TF using M-31 Hysol epoxy, the initial length of fiber (free end) is $L_0 = 13.77$. First I Trimm it to $L_1 = 6.8$ mm



and connect it to signal Analyzer (HP).

To Page No. _____

Date _____

Buffer circuit for tuning fork drive circuit, in order to eliminate the noise coming from cables

TITLE Building Buffer circute **Project No.** _____ **25**
From Page No. _____ **Book No.** _____ **Confidential**

purpose: In order to eliminate the noise and error caused by connecting wires, specially long wires, I build a Buffer in the circuite.

Equipments used: 2 Resistors ($10\text{K}\Omega$)
 2 OP-Amps (OPA137P - 7CZ238W)
 DC power source (AAK-model: LT5.3-1.5)

Procedure: Build the circute shown in the Figure below:

use 15 DCV for OP-AMP

TITLE Programming DDS (AD9833)		Project No. _____	27
From Page No. _____		Book No. _____	Confidential
<p>Purpose; The goal of this lab is to program a DDS (Direct Digital Synthesizer) chip using a SPI (Serial Peripheral Interface) programming. As a result the chip will be able to generate Sine, Triangle, and Square (pulse) waves.</p>			
<p>Equipments: - chip kit, digilent microcontroller (MAX32) - DDS chip (AD9833) - 20 MHz Clock, ECLIPTEK, (EC1100) - Break out (DigiKey part #: 309-1100-ND) - Precision Voltage regulator (REF192)</p>			
<p>File saved (all info): G:/Sajad/DDS Function Generator</p>			
<p>Procedure: a) First we build the circuit shown in AD9833 data sheet (Evaluation Board, Page 22). The circuit is also shown in the next page. Figure (1)</p>			

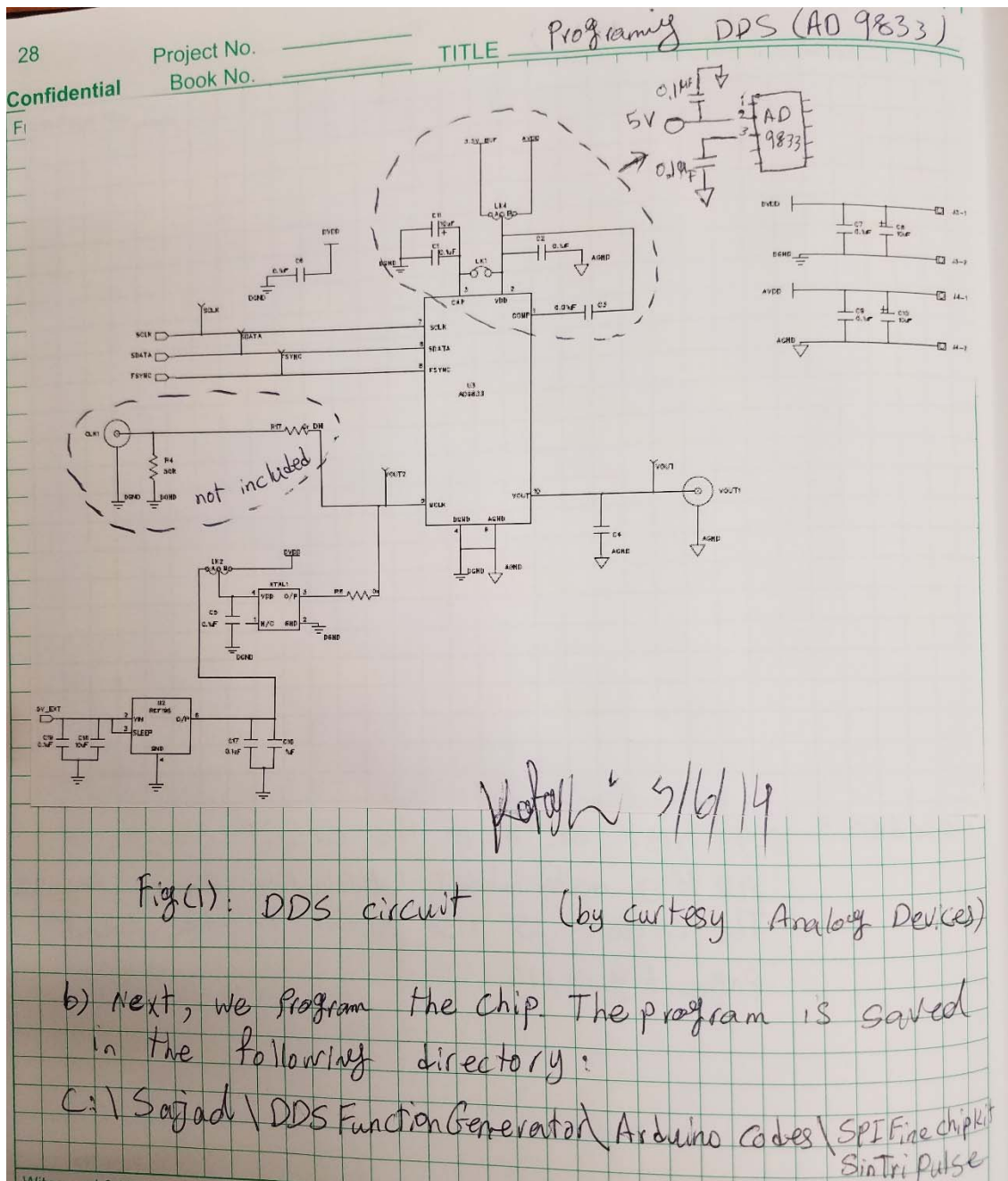
TITLE	Programming DDS (AD9833)	Project No.	_____	27
From Page No.	_____	Book No.	_____	Confidential

Purpose; The goal of this lab is to program a DDS (Direct Digital Synthesizer) chip using a SPI (Serial Peripheral Interface) programming. As a result the chip will be able to generate Sine, Triangle, and Square (pulse) waves.

Equipments: - chip kit, digilent microcontroller (MAX32)
- DDS chip (AD9833)
- 20 MHz Clock, ECLIPTEK, (EC1100)
- Break out (Dig:Key Part #: 309-1100-ND)
- Precision Voltage regulator (REF192)

File saved (all info): C:/Sajad/DDS Function Generator

Procedure: a) First we build the circuit shown in AD9833 data sheet (Evaluation Board, Page 22). The circuit is also shown in the next page. Figure (1)



TITLE Programming DDS (AD9833) Project No. _____ 29
 From Page No. _____ Book No. _____ Confidential

The code is also shown in the next pages.

Note 1: We use MAX32, digital microcontroller instead of Arduino, because it can go up to 32 bits while Arduino is limited to 16 bits.

Important notes of the coding are as following, it can also be observed in the code too:

1. Control word (CW) for Sin wave less than 1221 Hz $= 0$, bigger than 1221 Hz \rightarrow CW $= 4$
2. CW for Tri wave less than 1221 Hz $= 16384$
 CW $\sim \sim \sim$ bigger $\sim \sim \sim$ $= 16388$
3. $\sim \sim$ pulse \sim less $\sim \sim$ $= 5120$
4. $\sim \sim \sim \sim$ bigger $\sim \sim$ $= 5124$
5. while we put the frequency value in the Serial monitor we add
 S for Sin wave
 T for Triangle wave
 P for Pulse (Square) Wave

To Page No. _____

Confidential

Book No.

From Page No.

For example P350 or S9000 or T60

6 The Sine and Triangle wave generated by DDS chip is 650 mV P_{PK} , But the square wave is 5 V P_{K-PK} .

7. a voltage divider is added to the output to control the gain.

8 The Labview Connection with Arduino code is saved at: C:\Sajeed\DDFunction Generator\ArduinoLabviewSerial Interface\DDSLabviewArduino.VI

TITLE How to use PIV software		Project No. _____	35
From Page No. _____		Book No. _____	Confidential
Open Dynamic Studio V3.31			
New database → choose a file			
Open data base			
File → import → import images			
add Images			
Import as single frame.			
Trigger rate			
Right click on the Images			
and analyze ...			
Image conversion → make double frames → ok			
1-2 , 2-3 , 3-4 -			

36	Project No. _____	TITLE How to use PIV Software
Confidential	Book No. _____	
From Page No. _____		

Right click on make Double Frame
Analyze → PIV Signal
→ Adaptive Correlation

Right click on Adaptive Correlation
Analyze → Statistics → Vector Statistics

TITLE Particle density
From Page No. _____

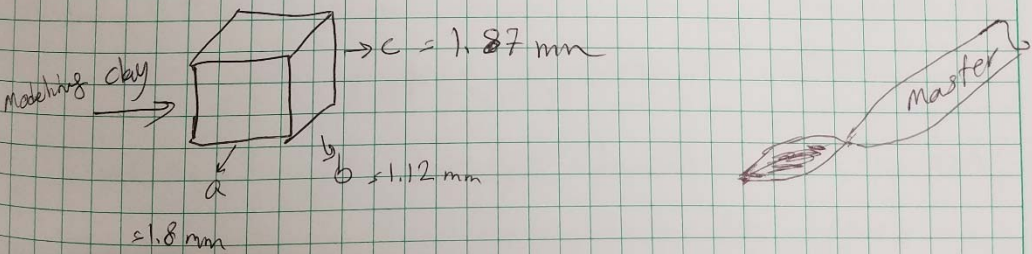
Project No. _____
Book No. _____

37
Confidential

Purpose: To measure the volume of the spoon

Equipments: Spoon (master)

- A flat spoon fork of particles was called a grain in previous measurements,
- Use a clay and make it fit to the spoon edge, and flat to the surface.
- Take that clay off of spoon and make it as a cube.
- Measure the length of the cube ^{side} using a caliper.



$$V = a \times b \times c = 3.7699 \text{ mm}^3 \approx 3.77 \text{ mm}^3$$

To Page No. _____

density of particles is defined as $\frac{1/20 \text{ g}}{20 \text{ ml}}$

e.g. to obtain a unit concentration place one flat spoon into 400 ml solution.

~~Note: 1 mm~~

Note: 1 milliliter = 1000 mm³

Therefore one flat spoon full = $\frac{3.77}{1000} \text{ mL}$

= 0.00377 mL

Number of particles / spoon full

Assuming closed packed structure (i.e. F.C.C.)

Project No. _____
Book No. _____
Confidential

TITLE _____
From Page No. _____

Atomic packing factor is approximately 74%, (APF)
therefore:
$$\frac{\text{Volume of Spoon} \times \text{APF}}{\text{Volume of a sphere}} = \text{Number of Particles}$$

i.e.
$$= \frac{3.77 \text{ mm}^3 \times 0.74}{\frac{4\pi}{3} r^3}$$

for $r = 0.005 \text{ mm}$

$\rightarrow N = 5.3 \text{ millions}$

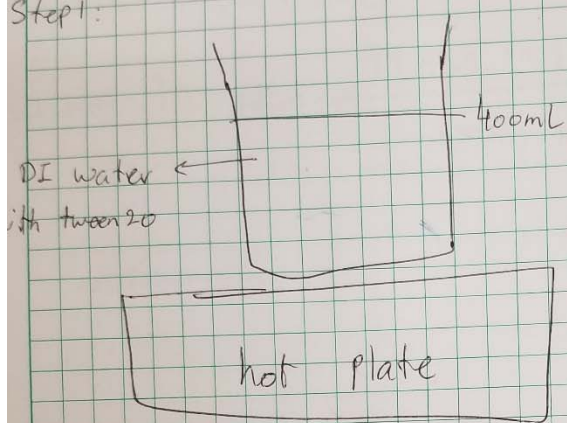
recommendation: From Cospheric.com : use less than
0.1% of tween 20 ~~in~~ with DI water

therefore 0.4 ml in 400 ml of water
heated to 80°C ; Soaking the spheres in
solution at room temperature.

40 Project No. _____ TITLE _____
 Confidential Book No. _____
 From Page No. _____

Mixing Sequence:

Step 1: heat to 80°C and check with IR gun.



DI water with tween 20

400mL

hot plate

Step 2: Empty 1 flat spoon of cospheric particles
 125-150 μ m Diameter (GRYPMS-1.00)
 therefore we have $1\text{ G} / 400\text{ mL}$

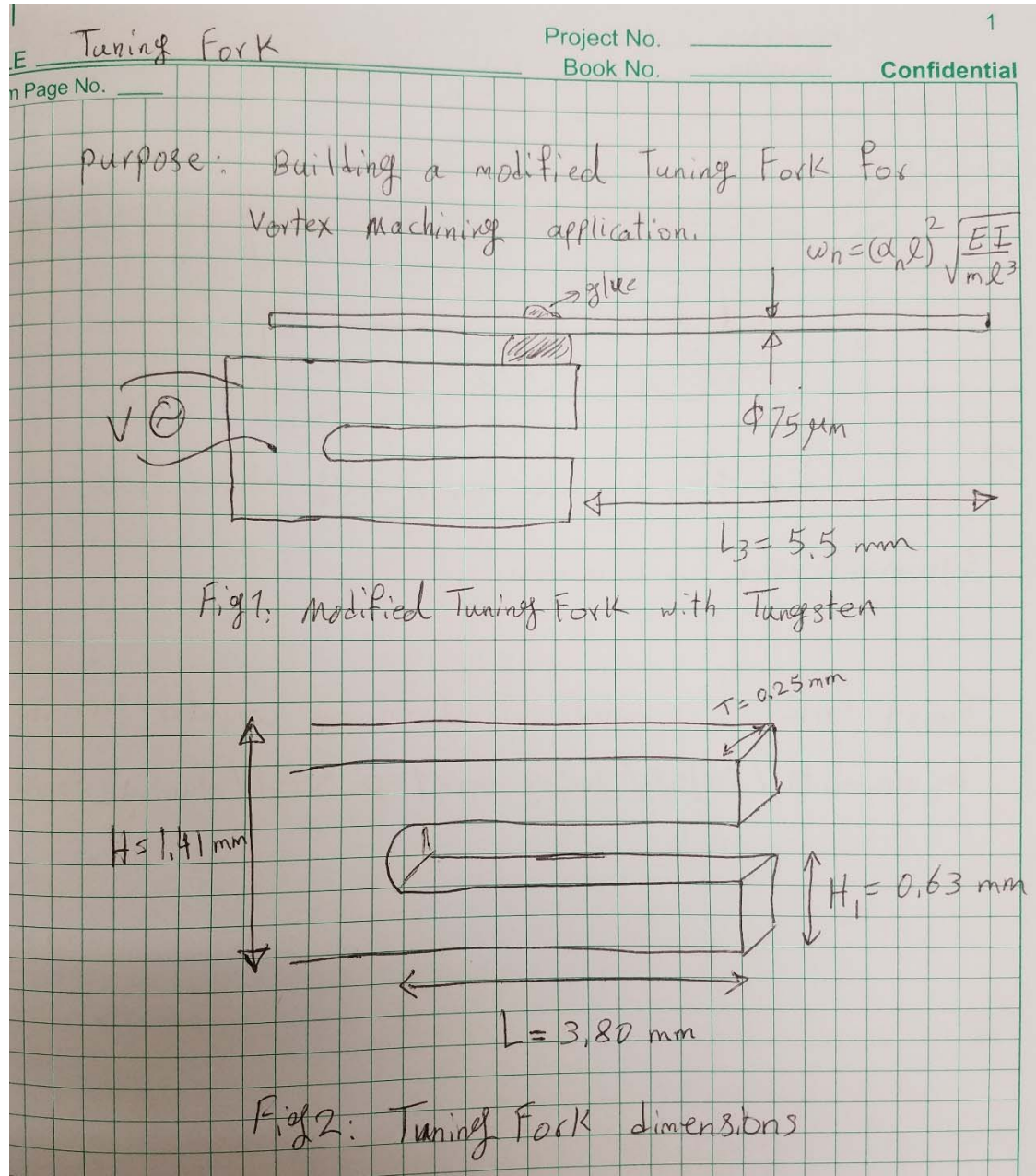
Step 3: Empty 100 mL

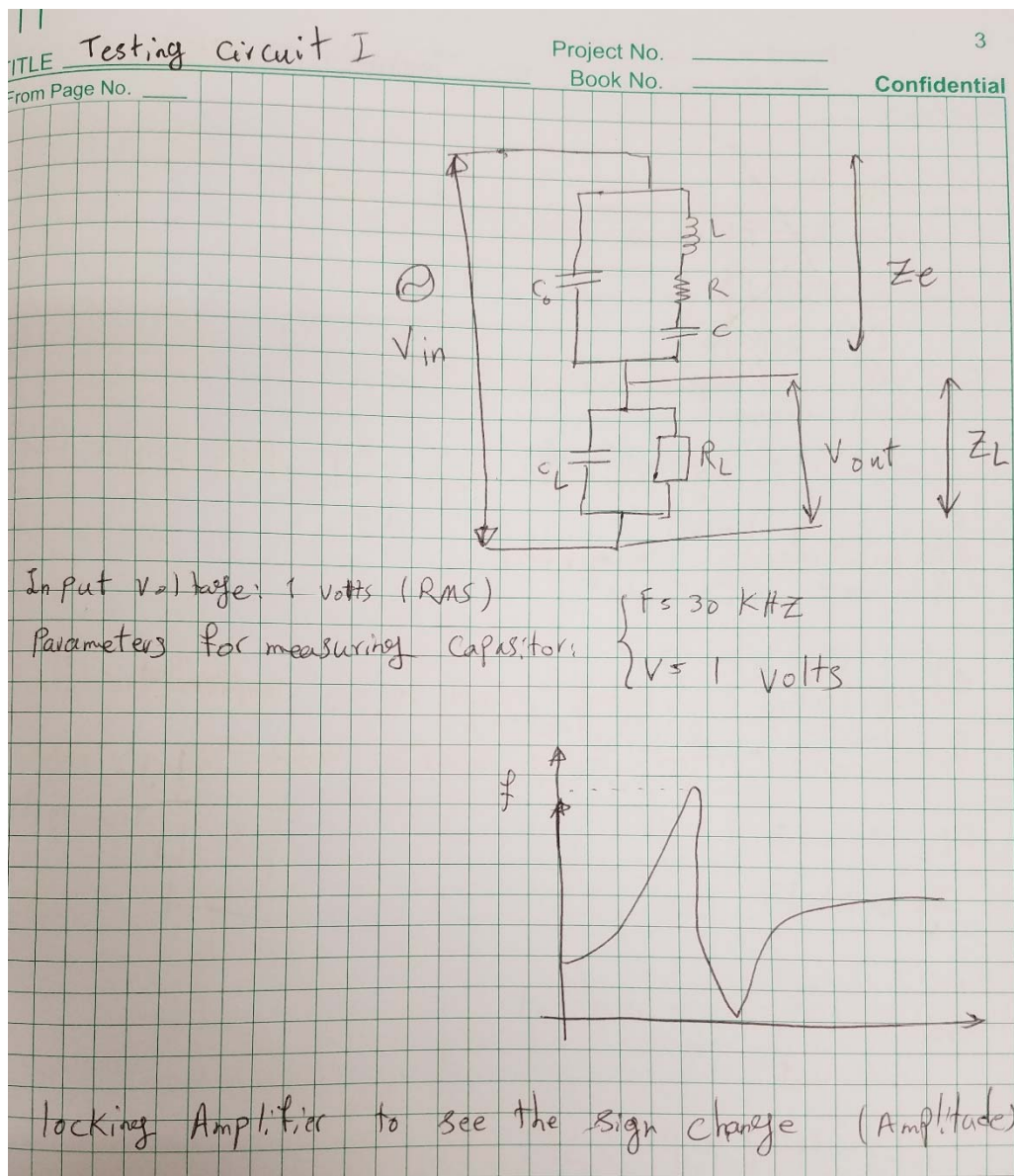
Step 4: add 1 Grain \rightarrow empty 100 mL which give
 us $2\frac{1}{3} \text{ G} / 400$ (~~$2\frac{1}{3} \text{ G} / 300$~~)

Step 5: add 1 G \rightarrow empty 100

To Page No. _____

TITLE _____		Project No. _____	41
From Page No. _____		Book No. _____	Confidential
which gives us $26\frac{1}{6} \text{ G}$			
		$\frac{400 \text{ mL}}$	Empty 100
Step 6: add 1 G, and you will have			
$19\frac{1}{3} \text{ G}$			
		$\frac{400 \text{ mL}}$	
After pouring, the last batch adds less than 50 mL			
therefore, the concentration is expected to be higher.			





4 Project No. _____ TITLE Testing circuit I
 Confidential Book No. _____
 From Page No. _____

~~$C_1 = 12 \text{ pF}$~~
 ~~$C_2 = 50 \text{ pF}$~~
 ~~$C_3 = 100 \text{ pF}$~~
 ~~$C_4 = 500 \text{ pF}$~~

$C_0 = 12.5 \text{ pF}$

measuring circuit:
 $32.7 \rightarrow 32.8$ with 400 data points

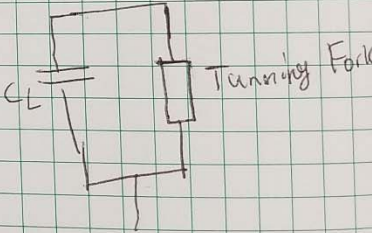
$R_L = 61.4 \text{ k}\Omega$ with C_{L0}
 file name (CLO.dat)

$C_{L1} = 97.85 \text{ pF}$
 $D = 0.0002$ (loss factor) (in parallel with T.F.)

$C_{L2} = 218.1 \text{ pF}$
 $D = 0.00013$

File names: $C_{L1}.dat$ $C_{L3}.dat$
 $C_{L2}.dat$

$C_{L3} = 499.6 \text{ pF}$
 $D = 0.00013$



TITLE Testing TF circuit I

Project No. _____

Book No. _____

Confidential

From Page No. _____

Measurement data

Frequency X Y

→ Amplitude transfer function $H = \sqrt{X^2 + Y^2}$

phase transfer function $\phi = \text{Arctan} \frac{Y}{X}$

Theoretical model:

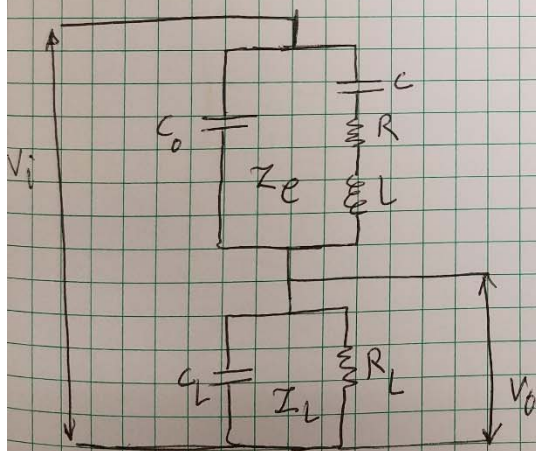
$$H(j\omega) = \frac{A(\omega) + jB(\omega)}{C(\omega) + jD(\omega)}$$

→ Amplitude transfer function $|H| = \sqrt{\frac{A^2 + B^2}{C^2 + D^2}} \times \text{Gain}$

phase transfer function $\phi = \text{Atan} \frac{B}{A} - \text{Atan} \frac{D}{C}$

Put $C_0 = 0$ in Z_e formula → $\omega_n^2 \frac{1}{LC} \rightarrow \omega_n = \sqrt{\frac{1}{LC}}$

Resonant frequency



$$\frac{V_o}{V_i} = \frac{Z_e}{Z_e + Z_L}$$

$$\frac{1}{Z_e} = \frac{1}{\frac{1}{j\omega C_0}} + \frac{1}{\frac{1}{j\omega C} + j\omega L + R}$$

$$\frac{1}{Z_e} = \frac{C_0 - \omega^2 L C_0 + j\omega C_0 R + 1}{\frac{1}{j\omega C} + j\omega L + R}$$

$$\frac{1}{Z_L} = \frac{1}{\frac{1}{j\omega C_L}} + \frac{1}{R_L} = \frac{1 + j\omega C_L R_L}{R_L}$$

To Page No. _____

Witnessed & Verified by: _____

Date: _____

Invented by: _____

Date: _____

6 Project No. _____ TITLE Testing TF circuit I
 Confidential Book No. _____
 From Page No. _____

$$Z_{es} = \frac{\frac{1}{j\omega C} + j\omega L + R}{\frac{C_0}{C} - \omega^2 L C_0 + j\omega C_0 R + 1}$$

$$Z_{Ls} = \frac{R_L}{1 + j\omega C_L R_L}$$

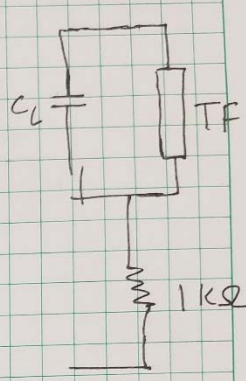
$$|H(j\omega)| = \frac{R_L \left(\frac{C_0}{C} - \omega^2 L C_0 + 1 \right) + j(R_L \omega C_0 R)}{\left(\frac{C_0}{C} R_L - \omega^2 L C_0 R_L + R_L + R + \frac{C_L R_L}{C} - \omega^2 L R_L L \right) + j(\omega C_0 R R_L + \omega L - \frac{1}{\omega C} + \omega C_L R R)}$$

7 Project No. _____ TITLE Testing TF circuit II
 Confidential Book No. _____
 From Page No. _____

$C_{L0} = 0.0$
 $C_{L0} \rightarrow f_0 = 32.732 \text{ KHz}$
 $C_{L1} = 100 \text{ PF} \rightarrow f_1 = 32.728 \text{ KHz}$
 $C_{L2} = 500 \text{ PF} \rightarrow f_2 = 32.726 \text{ KHz}$
 $C_{L3} = 200 \text{ PF} \rightarrow f_3 = 32.726 \text{ KHz}$

$32.720 \rightarrow L = ? \quad 5.1 \text{ mm}$

electronic parts: mouser.com & digikey.com



APPENDIX D: DRAWINGS AND MODELS

This appendix contains the 3D part design and assembly files used in the accomplishment of this dissertation. Please refer to the electronic file source, “Appendix D, Drawings&Models” folder for the data files.

APPENDIX E: LIST OF COMPONENTS

Tween 20 Biocompatible Surfactant - 2ml

http://www.cospheric.com/Tween80_Surfactant_density_gradients.htm

Fluorescent Green Polyethylene Microspheres, Item #: UVPMS-BG-1.00 10-20um - 0.1g

http://www.cospheric.com/UVPMSBG_fluorescent_green_spheres_density100.htm

Grey Polyethylene Microspheres, GRYPMS-1.00 45-53um - 5g

http://www.cospheric.com/Grey_polyethylene_microspheres_all_densities_all_sizes.htm

Grey Polyethylene Microspheres, Item #: GRYPMS-1.00 125-150um - 10g

http://www.cospheric.com/Grey_polyethylene_microspheres_all_densities_all_sizes.htm

Piezoelectric Actuator, Max Displacement 17.4 μ m, 6.5 x 6.5 x 20 mm

<https://www.thorlabs.com/thorproduct.cfm?partnumber=AE0505D16F>

Stepper Motor - 29 oz.in (200 steps/rev, Threaded Shaft)

<https://www.sparkfun.com/products/10848>

EasyDriver Stepper Motor Driver

<https://www.sparkfun.com/products/12779>

High-Speed-Steel Multiple Flute End Mill, Four Flute, 7/8" Mill Diameter, 1-7/8" Length
of Cut

<https://www.mcmaster.com/#3052a86/=pg6g7o>

Stainless Steel Truncated Ball, Part # 250-TB

<http://www.precisionballs.com/KinePriceBook.php>

Molybdenum - Wire, .5 mm diameter, 2 pcs

http://www.goodfellowusa.com/catalog/GFUS4I.php?ewd_token=3lVLEe6tX2jJjvpXAOqamle6ExcXLh&n=kFL9ITjED92De47tfGqqj1n8vGAXtn&ewd_urlNo=GFUS411&Cate=MO005340&CatSearNum=2

CONV DC/DC 12V 62MA SIP

<http://www.digikey.com/product-detail/en/murata-power-solutions-inc/HPR107C/811-1766-5-ND/1980224>

Laser Diodes (Quarton Inc. VLM-650- 03-LPT)

<http://www.digikey.com/product-detail/en/quarton-inc/VLM-650-03-LPT/VLM-650-03-LPT-ND/1010454>

HC-06 Serial Bluetooth Slave Wireless RF Transceiver Module

https://www.amazon.com/Arduino-Wireless-Bluetooth-Transceiver-Module/dp/B0093XAV4U/ref=sr_1_1?ie=UTF8&qid=1389302048&sr=8-1&keywords=bluetooth+arduino

High magnification camera (SONY XCD-SX910)

<http://www.graftek.com/pdf/Brochures/Sony/XCD-SX91andX71brochure.pdf>

40W RGBW LED high brightness light sources (LED Engin, model: LZC-B3MD07)

<http://www.ledengin.com/files/products/LZC/LZC-03MD07.pdf>

20mm, N-BK7 90° Specialty Mirror Enhanced Aluminum Coated

<https://www.edmundoptics.com/optics/prisms/retroreflection-prisms/90-degree-specialty-mirrors/49414/>

Ultra-Bright Flashlight, 1315 Lumens

<https://www.mcmaster.com/#6551t4/=16lgbdl>

High bandwidth class A amplifier (Thorlabs model MDT690)

<https://www.thorlabs.com/catalogpages/V21/640.PDF>

FET operational amplifier (LT1252)

<http://cds.linear.com/docs/en/datasheet/lt1252.pdf>

APPENDIX F: FOLDERS STRUCTURE

This appendix contains the structure of the file folders for MATLAB/LabView/Arduino/Multisim programs, codes and libraries, and 3D design, and assembly files used in the accomplishment of this dissertation.

- Appendix A, Equations for Matrix
- Appendix B, Codes& Programms
 - Arduino
 - CameraAutomaticPositioning
 - Arduino Code
 - JoystickThreeMotor
 - LEDPulseSerialParametrized
 - LEDShutterFrameRateSajad
 - MicroFiber
 - Ardoino code
 - JoystickFourMotor
 - WaveGenerate
 - examples
 - FrequencySweep
 - FrequencySweep3Channels
 - WaveFormGenerate
 - WaveGenerateLabviewControl
 - Labview
 - Project
 - Program Files (x86)
 - Sajad
 - Matlab
 - ePIV Codes
 - Probability of Encoded Particle Trajectories, Vmax & Vuniform
 - Micro Fiber Analysis Codes
 - Amplitude and Phase Frequency Response Hoy, Lupmed model, Lagrange
 - Amplitude Frequency Response Hiy, Lupmed model, Lagrange
 - Experimental Frequency Responce Plots, Probe 1A
 - Experimental Frequency Responce Plots, Probe 16
 - Two Beams Serially Connected, Tungsten, Varying Length
- Appendix C, Circuits&Notebook
 - MultiSim
 - LED Current Control Circuit
 - Piezo Amplifier
 - Potentiometer
 - Notebook Scanns
- Appendix D, Drawings&Models
 - Automatic 3-axis stage
 - LED&Mirror Stage 45 degree
 - Micro-Fiber Assembly
 - Multi-MicroFiber Apperatus
 - Images
 - Rev2
 - Rev3
 - Drawings
 - STL files
 - Related Works
 - Cutting Tips for Fibers, Scissors Assembly
 - Fiber Height Adjustment Flexure
 - LED Automatic Manipulator
 - Shelf
 - StwartPlatform
 - Solidworks Files
- Appendix E, List of Components
- Appendix F, Folder Structure