ABSTRACT

This thesis describes a method for measuring the flow velocity dynamics of microcirculatory blood using the laser Doppler technique. This technique is noninvasive and permits continuous measurement of microcirculatory blood flow. The principle of this method is to measure the Doppler shift - the frequency change that light undergoes when reflected by moving scatterers, such as red blood cells (RBC). Using this Doppler shift, the velocity of moving scatterers can be calculated from the Doppler equation. A prototype instrument has been assembled using equipment in the lab. The system has been calibrated and shows a good degree of accuracy ($\pm 1\%$) and repeatability ($\pm 2.2\%$). The system characterization parameters such as SNR and drift in velocity are also calculated and were found to be $\sim 24.62$ dB and 0.16%, respectively, over a period of 1 hour.

We also fabricated lensed structure at the tip of the optical fiber probes in order to facilitate blood flow measurements at a safe working distance from the fiber tip so as not to injure the subject under study. We present a detailed analysis of the design considerations for fabricating a lens at the tip of an optical fiber. Characterization procedures for the beams emanating from the two probes, namely the bare and the lensed fiber-optic probe, are presented. The experimental values of the beam parameters, i.e., working distance and spot size, are measured and compared with the calculated theoretical values. Typical values of the working distance are between 300$\mu$m-700$\mu$m and typical spot sizes are between 2-10 microns for the lensed probes fabricated. A novel
technique for measuring the angle between light propagation and velocity vector is developed, and the accuracy in the angle measurements is calculated to be ±6%. Results from velocity measurements using the bare fiber-optic probe on aortic arches of stage 24 chick embryos are also presented. The tools for display, acquisition and analysis of the data in real time are developed using MATLAB.
# TABLE OF CONTENTS

ABSTRACT ..............................................................................................................iv  
ACKNOWLEDGEMENTS ......................................................................................viii  

CHAPTERS  
1. INTRODUCTION .................................................................................................1  
   1.1 Motivation ......................................................................................................1  
   1.2 Introduction ...................................................................................................3  
   1.3 Other Velocity Measurement Techniques ......................................................4  
   1.4 The Laser Doppler Technique .....................................................................9  

2. LASER DOPPLER VELOCIMETRY .................................................................14  
   2.1 The Doppler Effect .......................................................................................14  
   2.2 Laser Doppler Techniques for Measuring Velocities .................................15  
   2.3 Comparison of Single Beam and Dual Beam Techniques .........................30  
   2.4 Laser Doppler Geometries .........................................................................34  

3. DEVICE CHARACTERIZATION AND VALIDATION .....................................39  
   3.1 System Calibration .......................................................................................39  
   3.2 System Characterization .............................................................................45  
   3.3 Angle Measurement .....................................................................................49  

4. PROBE CHARACTERIZATION AND LENS DESIGN ....................................61  
   4.1 Bare Fiber-optic Probe ...............................................................................61  
   4.2 Lensed Fiber-optic Probe ..........................................................................65  
   4.3 Lens Fabrication and Design Considerations .............................................66  

5. DISPLAY, ACQUISITION AND ANALYSIS SOFTWARE ..............................96  
   5.1 Simultaneous Display, Acquisition and Analysis .........................................97  
   5.2 Display and Acquisition GUI Tool ...............................................................98  
   5.3 Data Analysis ...............................................................................................101
6. VELOCITY MEASUREMENT RESULTS ..............................................104
   6.1 Chick Embryo Results .......................................................104
7. CONCLUSIONS AND FUTURE WORK ........................................110
   7.1 Conclusions .................................................................110
   7.2 Future Work ...............................................................114

APPENDIX ..............................................................................115

REFERENCES ..........................................................................143
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to Dr. Douglas Christensen for his invaluable guidance and support throughout the last three years. I am grateful to him for being a teacher and a mentor, for the pure, unselfish and honest passion with which he taught me to think like a researcher and for helping me to become a better person. He has never ceased to make himself available to provide the direction and expertise necessary for me to complete this thesis. I would like to express my gratitude to Dr. Norman Hu for his encouragement and guidance throughout this project and for allowing me to work in his laboratory. I would also like to extend my appreciation to Dr. Steve Blair for all his help and guidance and for proving me the opportunity to assist in his classes. Without his support, I wouldn’t have had the opportunity to write this thesis. I am also very thankful to Dr. Ajay Nahata for his encouragement and constant motivation, for believing in my abilities and providing me the opportunity to pursue my goals under his guidance. I would also like to thank all my friends for their support throughout this thesis work. Without them, this thesis though still possible, would have been lot less exciting. Lastly, and most importantly, I wish to express my indebtedness to my family, for without their love and motivation over the years, I would not be at this point today. I am grateful to them for everything they have given me.
CHAPTER 1

INTRODUCTION

1.1 Motivation

The understanding of functional characteristics of the embryonic cardiovascular system is essential to understanding cardiac organogenesis and vascular development. Embryonic cardiovascular functions are generally studied by measuring the vascular and aortic flow velocities [1]. Blood flow rate and heart physiology are studied so that the relationship between form and function can be better understood. Changing blood flow rate is directly correlated with contractility, preload, and afterload of the heart and thus indicates the development of contractile cells [2]. A functional alteration is a mechanism for morphological abnormality. When the aortic and vascular velocities are measured, the information on cardiac development can be extrapolated. Vascular parameters provide information on hemodynamic changes during organogenesis. Changes in vascular flow velocities correlate with changes in embryonic heart contractility [2]. Aortic flow measurements are used to determine the cardiac output of the heart. It is also representative of the hemodynamic changes during organogenesis and hence the relationship between structure and function [3].

Moreover, four developmental mechanisms (namely mesenchymal tissue migration, cardiac hemodynamics, cellular death and extracellular matrix abnormality), either singularly or in combination, play a major role in causing cardiac malformations
Altered cardiac hemodynamics, i.e., disordered intracardiac blood flow is a common mechanism in the spectrum of congenital heart defects. Some congenital heart defects with an intact ventricular septum and variable chamber volume are due to changes in intracardiac blood flow after the period of cardiac organogenesis [4]. This group includes obstructive defects in the left side of the heart like coarctation of the aorta and aortic valve atresia, and abnormalities on the right side of the heart that range from pulmonary valve stenosis to pulmonary atresia with an intact ventricular septum [4]. Some of these malformations related to abnormal blood flow are listed in Table 1.1.

Much of the cardiac research is currently being conducted on chick embryos. Chick embryos provide an appropriate medium for studying hemodynamic changes during cardiovascular development because of several reasons. First, this is a widely accepted model of vertebrate heart development. Second, the access to the chick embryo, although complicated, is not as difficult as a mammalian heart. Third, the vascular walls

<table>
<thead>
<tr>
<th>Table 1.1 Malformations related to altered cardiac hemodynamics.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarctation of the aorta with intact ventricular septum</td>
</tr>
<tr>
<td>Hypoplastic left heart syndrome</td>
</tr>
<tr>
<td>Aortic valvular stenosis</td>
</tr>
<tr>
<td>Interruption of aortic arch</td>
</tr>
<tr>
<td>Secundum atrial septal defect</td>
</tr>
<tr>
<td>Pulmonary atresia without ventricular septum defect</td>
</tr>
<tr>
<td>Perimembranous ventricular septal defect</td>
</tr>
</tbody>
</table>
of the developing tissues in them are translucent, making forward and retrograde flow visible in the blood vessels [5]. The similarities between mammalian and avian heart during early embryonic stages [1] also makes them an appropriate medium for studying cardiovascular development. Our primary motive in this work is to measure blood flow velocities in chick embryos using the laser Doppler technique and hence extrapolate information about the defects related to altered cardiac hemodynamics.

1.2 Introduction

Optical sensors are very popular in biomedical diagnostics. The primary advantage of these sensors over conventional ones is that the measurements can be done noninvasively, i.e., without injuring the tissue. Moreover, these methods are safe to use, as there is no direct electrical contact between the measurement device and the sample. All the information needed is transferred via light with powers typically less than a couple of milliwatts. The properties of laser light, such as monochromaticity, coherence, low divergence and high intensity are also ideally suited for measurements in the biomedical field.

One of the most widely used optical techniques in biomedical diagnostics is the laser Doppler technique. The first demonstration of measuring velocities with the Doppler shift of laser light was done by Cummins et al. [6] in 1964, soon after the invention of first laser in 1960 [7]. They observed the shift of light scattered from particles carried in water flow. Since then, numerous clinical and experimental applications of the laser Doppler technique have been demonstrated. Some of the important current applications of this technique are briefly discussed later in the chapter.
1.3 Other Velocity Measurement Techniques

Before discussing the laser Doppler velocimeter in further detail, we briefly discuss some other popular velocity measurement techniques (the laser Doppler velocimeter will be discussed in great detail starting next chapter).

1.3.1 Particle Image Velocimetry (PIV)

PIV is a nonintrusive planar measurement technique for simultaneously measuring the velocities at many points in a fluid flow. It uses a laser light source to illuminate the flow of moving particles. An image of moving particles in the illuminated region is captured by a camera and then, a short time period later, another image is taken. From the displacement of the moving particles, provided that the time interval between image captures is known, an instantaneous velocity vector map of the flow area can be generated.

The theory of PIV was introduced by Adrian [8] in the late 1980s. At that time, due to hardware limitations, a single photographic frame was multiply exposed and analyzed using an auto-correlation technique. However, improved speeds of photographic recording have allowed images to be captured on separate frames for analysis using cross-correlation techniques. Moreover, with the introduction of digital cameras, direct recording of images at the expense of reduced resolution is now possible and has led to the development of digital PIV (DPIV) [9].

The technique of PIV consists mainly of two stages: Image Acquisition and Image Evaluation. A general PIV process is illustrated in Figure 1.1.
1.3.1.1 Image Acquisition

Figure 1.2 shows a typical experimental setup for carrying out the image acquisition process. The scattering particles in the flow are illuminated by a laser source and images of the illuminated flow are captured and stored for later analysis. Images can either be captured photographically or digitally using Charge Coupled Device (CCD) arrays.

1.3.1.2 Image Evaluation

Since the introduction of first PIV, a number of different image evaluations algorithms have been developed to improve the speed and accuracy of evaluation procedure. A classical PIV analysis method [10] that forms the basis of many other algorithms is shown in Figure 1.3. The analysis is done by calculating the correlation of regions of input images with each other and generating the velocity vector map. This technique can be used both for single frame multiply-exposed (auto-correlation) and multiple frames singly-exposed (cross-correlation). Finally, to speed up the convolution process, correlation between pair of input images is carried out in the Fourier space [10].
Figure 1.2 A typical experimental setup for PIV.

Figure 1.3 Basic PIV analysis process.
1.3.2 Ultrasound Bio-microscopy (UBM) Doppler

Ultrasound Bio-microscopy (UBM) is an important *in vivo* imaging method for analyzing cardiovascular development and congenital heart diseases during early embryonic stages. UBM-Doppler uses high frequency ultrasound both for *in utero* imaging and for Doppler detection of blood velocity waveforms in developing embryos [2].

This Doppler technique is based on measuring small changes in ultrasound frequency (Doppler shift) from transmission of the pulse to reception of echo at the transducer. The echo comes from the scattering of input pulse by moving particles in the flow. Recently, UBM systems employing a 40-MHz ultrasound transducer with spatial resolutions of 30 µm axial and 90 µm lateral have been demonstrated [2]. The UBM analysis of cardiovascular functions is further extended by including 40-50 MHz pulsed Doppler, enabling precise image guided Doppler measurements of blood flow velocities [2]. UBM image and Doppler data are acquired with separate transducers enabling precise and accurate UBM-guided Doppler measurements in embryonic blood vessels over a wide range of gestational stages [2]. This approach has the advantage of allowing the transducers to be optimized independently, either for imaging or Doppler detection, and enables the simultaneous acquisition of UBM images and Doppler blood velocity measurements. An experimental setup of UBM-Doppler system to simultaneously image and measure blood velocities is shown in Figure 1.4.

UBM-Doppler differs from conventional echocardiography systems in the use of high frequency ultrasound, which permits higher spatial resolution, although with poorer depth of penetration.
The higher frequency Doppler transducers also facilitate the acquisition of low velocity blood flow signals. A two-dimensional UBM image is generated by mechanical motion of the imaging transducer, whereas the pulsed-wave Doppler transducer, guided by the UBM image itself, gives the blood flow waveform.

1.3.3 Velocity Measurement Using Time-Varying Laser Speckle

Another approach of measuring the velocities of particles in fluids is to use laser speckle [11]. Laser speckle is a random interference pattern produced by light reflected or scattered from different parts of the illuminated surface. When an object illuminated with laser light moves, the speckle pattern it produces fluctuates, leading to so-called time-varying laser speckle. These intensity fluctuations can thus be used to measure the velocity of the scattering particles.
1.4 The Laser Doppler Technique

Besides the velocity measurement techniques discussed in the previous section, one of the most widely used methods for measuring microcirculatory blood velocities is based on the laser Doppler technique in the form of laser Doppler velocimeters. The primary advantage of a laser Doppler velocimeter is its linear response over a very broad range of velocities. Other advantages include fast response, temperature independence and high spatial resolution. However, a major limitation of this technique is that the signal to noise ratio (SNR) depends primarily on the presence of scattering particles in the flow. Moreover, this technique also requires that the medium be transparent and there be an optical access to the flow area. Some of popular applications of the laser Doppler technique are discussed next along with a brief introduction of its use in velocity measurement applications.

1.4.1 Laser Doppler Perfusion Monitoring and Imaging

A number of common diseases – including inflammatory conditions, allergic reactions and tumors – influence perfusion of blood in the skin (defined as the product of average velocity and concentration of blood cells in the illuminated tissue volume). In addition, impaired circulation in association with diabetes and peripheral vascular disease may lead to the ulcers and ultimately to tissue necrosis [12]. Consequently, the perfusion of blood in the skin as well as in other tissue models was needed to be investigated by the use of noninvasive methods with a minimal influence on the parameters under study. This was made possible by the commercialization of the Laser Doppler Perfusion Monitoring (LDPM) instrument during early 1980s [12]. In LDPM, optical fibers are used to guide light from the laser to the probe attached to the tissue to be monitored. The backscattered
and Doppler-broadened light is subsequently picked up by the same fiber and brought back to the photodetector unit in the instrument. An output signal that scales linearly with the perfusion in the tissue is extracted by processing the Doppler signal and is fed to the display for continuous tracking of the perfusion.

One of the features of blood perfusion that must be accounted for is its large spatial and temporal variability over the region to be monitored. Consequently, the small sampling volume of fiber-optic-based LDPM instruments constitutes one of the major limitations of this technology. With the spatial variability in the tissue blood perfusion, differences in readings may appear at recordings from adjacent sites. This disadvantage of LDPM triggered the development of Laser Doppler Perfusion Imaging (LDPI) technology towards the end of 1980s [13]. In LDPI, a laser beam successively probes a large number of tissue volumes and a data set of perfusion values is generated from which a color coded perfusion map is compiled. LDPM is well suited for continuous real time monitoring of tissue blood perfusion at a single site, whereas LDPI is primarily intended for studies of the spatial variability of the tissue blood perfusion. Thus, although complimentary, LDPM and LDPI constitute a versatile pair of related noninvasive Doppler techniques that facilitate the study of temporal and spatial variability of tissue blood perfusion.

1.4.2 Laser Doppler Microscopy

A laser Doppler microscope employs a differential Doppler flow-meter system along with a microscope system and a photodetector to assess and measure flow velocities in a tissue sample. The two optical beams in the differential Doppler flow-meter are aligned such that they overlap exactly at the flow area. The light scattered by
the particles in the flow is made to pass through a movable slit and is detected by the photodetector. The microscope system consists of a monitoring as well as an illuminating system to observe and correctly align the sample object in which the velocity fluctuations of the flow are to be measured. Thus, statistical signal processing on the Doppler signal collected by the photodetector gives information about the velocity of moving particles in the flow. Moreover, the microscope system can also be used to photographically record the interference fringes in the flow area at different times during the flow for further analysis [14].

1.4.3 Self-mixing in Laser Diode Cavity

Self-mixing effect in a laser diode (LD) occurs when an external feedback (backscattered light) into the laser cavity interferes with the light already present in the cavity. Depending on the delay and hence the phase of the backscattered light, the LD threshold condition is changed. This induces an intensity modulation in the output of the laser as the pump current is held constant. These power fluctuations can be detected using a photodetector placed on the opposite side of the laser cavity from the primary light output [15]. If the light coming out of the signal arm (the arm carrying light to the moving scatterers) is frequency-shifted and is mixed coherently with the original laser light in the diode cavity, then the frequency of intensity modulation can be related to the Doppler shift. In literature, this technique is also sometimes referred to as the ‘autodyne effect’ or ‘self modulation’. This intensity modulation was first noticed in the output of gas lasers by King et al. [16] in 1963. A typical experimental setup for explaining the self-mixing effect is shown in Figure 1.5.
The primary advantage of self-mixing interferometry is that no optical interferometer external to the source is needed. Besides, the photodetector is often accommodated in the LD package resulting in a very simple, cheap and compact setup. The optical alignment of the instrument is very simple as there is only one optical axes to worry about.

Besides having applications for velocimetry, vibration, distance and displacement measurements, this technique has also been used in biomedical applications such as blood pressure pulse registration, Laser Doppler Flowmetry and measurement of skin vibrations [17]. Recently, it is reported to have been used in optical touch sensitive interfaces [18], where two self-mixing interferometers measure the movement of the fingertip.

1.4.4 Single-beam Laser Doppler Velocimetry

For the work described in this thesis, we have used the single-beam laser Doppler technique to measure blood velocities in microcirculatory vessels. This technique along with the dual-beam (differential) laser Doppler technique is discussed in greater detail in
the next chapter. A comparison between the velocity measurement techniques discussed above and the laser Doppler technique is also presented towards the end of Chapter 2.

For reference purposes numerous other methods and measurement applications of the laser Doppler technique are summarized in books by Belcaro et al. [19] and Shepherd et al. [20].
CHAPTER 2

LASER DOPPLER VELOCIMETRY

2.1 The Doppler Effect

The Doppler effect was discovered in the middle of the nineteenth century by an Austrian physicist Christian Doppler, and is hence named after him [21]. It is the change in frequency that occurs when there is relative movement between the source of a wave and an observer. It occurs because the waves emitted by the source are either compressed (if the source and observer are moving towards each other) or spread out (if they are moving away from each other). The most familiar everyday example is the drop in pitch (which is determined by the frequency) as a sound source moves past an observer. As this frequency change is dependent on the relative velocities of the source and the observer, this effect can be used to measure velocities.

The Doppler effect also occurs with light. However, the frequencies of light waves are very high and difficult to measure directly. This problem is solved by using the phenomenon of ‘beats’ -- the effect that is produced when two waves of slightly different frequency are superimposed. As the two waves come into and out of phase with each other, the result is the detection of a frequency that is equal to the difference in frequency between the two waves.

By mixing the Doppler-shifted wave with a reference wave of the original frequency, a beat frequency is produced that is much lower than either of the two
constituent waves and is therefore much easier to measure. As this beat frequency is equal to the difference between the two frequencies, it is hence exactly equal to the frequency shift produced by the Doppler effect. It can be shown [22] that the relationship between the frequency change and the relative velocity of source and detector is given by

\[ f' - f = \frac{v}{c-v} f \]  

(2.1)

where \( f \) is the transmitted frequency, \( f' \) is the received shifted frequency, \( v \) is the relative velocity between the source and the detector and \( c \) is the velocity of the wave. In the case of light waves, \( c \) is the velocity of light, which is usually much larger than the velocities being measured and thus (2.1) can be approximated as

\[ f' - f = \frac{v}{c} f \]  

(2.2)

Hence the frequency shift, and therefore the beat frequency when the Doppler-shifted light is mixed with a reference beam of the original frequency, is directly proportional to the velocity being measured.

### 2.2 Laser Doppler Techniques for Measuring Velocities

A number of different optical geometries for laser Doppler configurations have been demonstrated over the period of last 40 years. Each geometrical arrangement is based on a specific application of the laser Doppler velocimeter (LDV). The two very
basic categories under which all the geometries fall are the Doppler-shift velocimeter and the fringe-shift velocimeter, described in the next sections [23]:

2.2.1 Doppler-shift Velocimeter

In Doppler-shift velocimeters, the frequency shift of one light beam due to the Doppler effect is measured by superimposing it with another reference light beam [22]. This method of measuring velocities is commonly referred to as the reference beam (or the single beam) method and is discussed in further detail below:

2.2.1.1 Reference Beam Method

As mentioned in the previous section, the frequencies of light waves are much too high to be measured directly. Because of this, the basic implementation of the laser Doppler technique to measure fluid velocities is to illuminate part of the flow field with laser light, collect some of the light reflected from it, mix it with the original light and measure the resulting beat frequency [24]. This beat frequency is directly proportional to the velocity of the particles in the fluid. However, there is a small change from the Doppler theory given in the previous section. There it was assumed that the object was a source of waves. If it is instead a reflector of waves sent out by a source near the observer, then one should consider the fact that the object is moving with respect to the incoming wave also, i.e., with respect to the source as well as detector. This has the effect of doubling the frequency shift compared with that given by (2.2). Thus the new equation for Doppler frequency shift is now given as

\[
f' - f = \frac{2v}{c} f \tag{2.3}
\]
Let us rewrite this equation to give the velocity in terms of the beat frequency. For our convenience, let the beat frequency (i.e., the Doppler frequency shift) be written as

\[ \Delta f = f - f' \]

We can thus rearrange (2.3) to give the velocity of the moving particles as

\[ v = \frac{c \Delta f}{f'^2} \]

Using the relationship \( c = f\lambda \), where \( \lambda \) is the wavelength of the laser light, we get

\[ v = \frac{\lambda}{2} \Delta f \]  \hspace{1cm} (2.4)

Remember here that \( v \) is the velocity of the object (or more correctly the relative velocity of the object with respect to the observer) measured along the line of sight, i.e., along the direction of the incident light beam. However, if the object is moving at an angle \( \phi \) with respect to the direction of incident light beam, then (2.4) gives the component of velocity \( (v \cos \phi) \) along the line of sight. Thus rewriting (2.4) by taking into account the angle, we have

\[ v \cos \phi = \frac{\lambda}{2} \Delta f \]
Rewriting,

\[ \nu = \frac{\lambda}{2\cos\varphi} \Delta f \]  

(2.5)

This is the basic equation to measure the velocity of moving particles using the laser Doppler technique. Equation (2.5) above is derived using the frequency shift due to the Doppler effect explanation given in section 2.1. This same equation can be derived from a more rigorous treatment, as is done next.

2.2.1.2 Alternate Description of the Reference Beam Method

Consider the geometry shown in Figure 2.1 showing waves emitted by a stationary point source S. Assuming that the source S is far enough away (compared to a wavelength) from the observation plane that the wavefronts reaching that plane are approximately planar.

Figure 2.1 Plane wavefronts of radiation emitted from a stationary point source.
Here $r_n$ represents the distance of wavefront from source S. In the present geometry the source S is stationary, hence the distance between two consecutive wavefronts $r_n$ and $r_{n+1}$ can be written as

$$r_{n+1} - r_n = \lambda \quad (2.6)$$

At any instant $t$, the disturbance occurring on wavefront $r_n$ has left the source S at time $t_n$, which can be deduced from the geometry of Figure 2.1 to be

$$t_n = t - \frac{r_n}{c} \quad (2.7)$$

At the same instant $t$, the disturbance occurring on wavefront $r_{n+1}$ has left the source S at time $t_{n+1}$ given as

$$t_{n+1} = t - \frac{r_{n+1}}{c} \quad (2.8)$$

This time interval, $t_{n+1} - t_n$, equals the period of the wave $T$, i.e.,

$$t_{n+1} - t_n = T = \frac{(r_{n+1} - r_n)}{c} = \frac{\lambda}{c} \quad (2.9)$$
In the discussion above, we assumed that the source S is stationary. Let us now assume that the source S is moving with a constant velocity \( v \) in the forward direction as shown in Figure 2.2.

During the time interval \( T \), the distance moved by the source can be written as

\[
vT = \frac{v\lambda}{c} \tag{2.10}
\]

Since the source is in motion, hence as shown in Figure 2.2, the distance of the source from wavefront \( r_n \) is displaced in the direction of motion by an amount \( (v\lambda/c) \), with respect to the distance between the source and wavefront \( r_{n+1} \). Therefore the waves emitted will have an apparent wavelength, \( \lambda' \), expressed as

\[
\lambda' = \lambda - \frac{v\lambda}{c} = \lambda \left(1-\frac{v}{c}\right) \tag{2.11}
\]

Figure 2.2 Apparent wavefronts of the radiation emitted from a source moving with a constant velocity.
Hence the apparent frequency, \(v'\), can be written as

\[
\nu' = \frac{\nu}{1 - \frac{v}{c}}
\]  

(2.12)

Using similar arguments, it can be shown that for a source moving in the opposite direction, the apparent wavelength will be

\[
\lambda' = \lambda \left(1 + \frac{v}{c}\right)
\]  

(2.13)

Equations (2.11) and (2.13) represent the apparent wavelength observed at a point located along the direction of motion the source. However, the general expression for the frequency observed at a point \(P\) located at an angle \(\theta\) with respect to the direction of motion of the source can be derived as follows. Consider the geometry shown in Figure 2.3.

Here \(S_1\) is the position of source at time \(t=0\), \(S_2\) is the position of source at time \(t\) and \(r_1\) and \(r_2\) are the distance of observation point \(P\) from the source positions \(S_1\) and \(S_2\) respectively. Also assuming that the time interval \(t\) is sufficiently short such that

\[
S_1S_2 = vT << r_1
\]  

(2.14)
Figure 2.3 General geometry for the observation point at an arbitrary angle from the direction of motion of the source.

Now the time it takes for the wave emitted by source at position S₁ to reach point P is given as

\[ t_1 = \frac{r_1}{c} \quad (2.15) \]

Similarly the wave emitted by source at position S₂ reaches the observation point P at time

\[ t_2 = t + \frac{r_2}{c} \quad (2.16) \]

Also the number of wave trains received by the observer at point P in the time interval, \( t_2 - t_1 \), is equal to the wave trains emitted by the source in time \( t \). Therefore we can write
\[ t \nu = (t_2 - t_1) \nu' \]  

(2.17)

Rewriting using (2.15) and (2.16),

\[ \nu' = \frac{t \nu}{(t_2 - t_1)} = \frac{t \nu}{t - \left( \frac{r_2 - r_1}{c} \right)} \]  

(2.18)

From the geometry of Figure 2.3 and the assumption in (2.14) we can write the distance, \( r_2 - r_1 \), as

\[ r_2 - r_1 = S_1 S_2 \cos \theta = vt \cos \theta \]  

(2.19)

Substituting (2.19) in (2.18), we get the apparent frequency at the observation point P as

\[ \nu' = \frac{\nu}{\left( 1 - \frac{\nu \cos \theta}{c} \right)} \]  

(2.20)

Note that (2.20) reduces to (2.12) for \( \theta = 0 \) and \( \theta = \pi \). Using this equation the velocity of the moving scatterers can be derived from the Doppler equation.

2.2.2 Fringe Velocimeter

Fringe velocimeters are dual beam systems in which optical arrangements are made to interfere the two beams exactly at the flow area. The interference of two beams
creates a fringe pattern at the flow area and the rate at which these fringes are crossed by moving particles gives the desired Doppler frequency. These velocimeters are based on the dual beam or Doppler difference method discussed in further detail below.

2.2.2.1 Dual Beam (or Doppler Difference) Method

The bringing together of the light scattered from the moving object and the reference beam is not trivial and for many applications a slightly different technique is better. This technique uses two laser beams that cross at an angle in the volume where the velocity is to be measured [25]. A typical set-up is illustrated in Figure 2.4.

Consider a particle moving with velocity \( v \) in the direction perpendicular to the bisector of the two incident laser beams. The components of this velocity parallel to the two laser beams are \( v \sin \theta \) and \(-v \sin \theta\). (It is clear from Figure 2.4 above that for the beam coming in from the upper left, the component of particle velocity is in the direction opposite to the direction the light is traveling, whereas for the other beam it is in the same direction: this accounts for the difference in sign.) Hence there will be a frequency increase when the upper beam is reflected by the particle and a corresponding decrease (of the same magnitude) when the lower beam is reflected.

It is clear from the arguments expressed above that the frequency change in each case will be given by (2.2), provided the velocity is replaced by its component \( v \sin \theta \), i.e., the Doppler frequency shift is given as

\[
f' - f = \frac{v \sin \theta}{c} f
\]
However, the difference in frequency between the two beams reflected by the particle will be double this (as the two frequency shifts are equal but of opposite sign). Thus, we have

$$\Delta f = \frac{2v \sin \theta}{c} f$$  \hspace{1cm} (2.21)

As in the reference-beam case, there will also be a frequency shift caused by the velocity of the particle with respect to the reflected light traveling towards the observer. However, this will be the same shift for each beam and hence will not contribute to the frequency difference observed. Equation (2.21) therefore represents the total frequency difference (and hence the beat frequency) produced by the motion of the particle.

Rearranging (2.21) to get an expression for the velocity of the particle in terms of the beat frequency observed, we have
In other words, to determine the velocity of the moving particle, the angle between the two laser beams, the (original) wavelength of the laser light and the beat frequency observed should be known.

Notice that here we have assumed that the particle is traveling at right angle to the bisector of the two incident laser beams. However, if the particle is moving at an angle \( \phi \) with respect to the direction shown, what is measured is the component of the velocity (\( v \cos \phi \)) in this direction. Accounting for this angle, we can rewrite (2.22) as

\[
v \cos \phi = \frac{\lambda}{2 \sin \theta} \Delta f
\]

Solving for \( v \),

\[
v = \frac{\lambda}{2 \sin \theta \cos \phi} \Delta f
\]

This is the required Doppler equation to measure the velocity of the moving particle in the dual beam geometry. Similar to the reference beam case of section 2.2.1, (2.23) above is derived using the frequency shift due to the Doppler effect explanation given in section 2.1. Let us now look at an alternate way of explaining the dual beam phenomenon discussed above.
2.2.2.2 Interference Explanation of the Dual Beam Doppler Technique

The beat frequency mentioned above (and in section 2.2) is observed as an amplitude modulation in the light scattered by the moving particles. There is another way to look at this intensity fluctuation. Two laser beams crossing at an angle will interfere as shown in Figure 2.5.

Consider two light beams crossing at an angle $2\theta$ as shown in Figure 2.5. The light parallel lines represent the crests of the waves at a particular time. The bold horizontal lines in the area of overlap indicate the locations where the crests of one wave fall on the crests of the other wave. (Inspection of the diagram shows that the troughs of the two waves are also superimposed in the horizontal direction -- the troughs are midway between the crests in the two waves.) Hence at these locations, which are lines in the diagram but planes in three dimensions, the two waves are in phase and there will be a maximum light intensity.

![Figure 2.5 Interference explanation of the dual beam Doppler technique.](image)
Midway between these lines the two waves will be in anti-phase, with the crest of one falling on the trough of the other, and the two waves will cancel out. Hence, there will be minimum intensity at these points. From this interference explanation, it can also be shown that these lines of equal phase are fixed in space. Hence they represent a pattern of interference fringes in space. It can also be shown that these fringes are equally spaced and are parallel to the line bisecting the angle between the two waves. Moreover, simple trigonometry applied to the area of overlap shown in Figure 2.5 leads to the following expression for the fringe spacing $s$ in terms of the wavelength $\lambda$ of the light and the angle $\theta$ between each light beam and the angular bisector of the beams (which is parallel to the fringes).

$$s = \frac{\lambda}{2\sin \theta}$$  \hspace{1cm} (2.24)

Now consider a particle traveling through this area of overlap. When it is in a bright fringe, it will reflect light to the detector. However, when it is in a dark fringe, where the two light beams have cancelled out due to them being in anti-phase, there will be no light reflected from the particle. Hence the light received from the particle will fluctuate with a regular period as it passes through the interference fringes. If the velocity of the particle (in the direction perpendicular to the fringes) is $v$, it travels a distance $v$ in one second. If the fringe spacing is $s$, this means that it will pass through $v/s$ bright fringes per second and the light will therefore oscillate with this frequency. Using the value for $s$ from (2.24) gives the beat frequency or the frequency of oscillation as
\[ \Delta f = \frac{v}{\lambda} 2 \sin \theta \]

Solving for \( v \),

\[ v = \frac{\lambda}{2 \sin \theta} \Delta f \]  \hspace{1cm} (2.25)

For a particle moving at an angle \( \phi \) with respect to the normal to the bisector of the two incident laser beams, we can rearrange (2.25) (from the discussion in section 2.3.1) to write the velocity of the moving particles as

\[ v = \frac{\lambda}{2 \sin \theta \cos \phi} \Delta f \]  \hspace{1cm} (2.26)

This is exactly identical to (2.23). In other words, considering the experiment where two light beams interfere to form a fringe pattern and the intensity fluctuations in the light reflected by the particle are caused by passing it through these fringes, gives exactly the same answer as considering the Doppler shifts caused by light reflecting from a moving particle. In one approach, we treated the situation where frequency of the light was changed by the Doppler effect and then we combined two beams of different frequency to give a beat frequency. In the second, we considered the interference of two beams of same frequency and counted the number of interference fringes traversed by the particle through them. Hence it is clear from the fact that both approaches give the same answer that they are two different ways of looking at the same phenomenon.
2.3 Comparison of Single Beam and Dual Beam Techniques

2.3.1 Effect of Particle Density

The interference-fringe interpretation of the dual beam technique discussed in section 2.3.2 assumed that there is only one particle in the overlap area. But the situation will be different with multiple scatterers (which is almost always the case with all real systems). In that case, if one particle is in a bright fringe when a second is in a dark fringe, then the variation in the total intensity detected will be lower than discussed above. This is because of partial cancellation of the phases of radiated fields from multiple scatterers. In other words, the depth of modulation will be reduced. This leads to a reduction in the strength of the beat signal detected. We assume that the scattering of laser radiation from individual particles is random. Hence if there are large numbers of particles in the flow at the same time then from the concept of random phasor sums [26] it is possible for the resultant of the scattered signal to be reduced. Hence the dual-beam technique is better when there is a low density of particles in the flow (preferably only one in the overlap area at any time). This restriction does not apply to the reference-beam technique, which is therefore preferred when the particle density is high.

2.3.2 Detecting the Direction of Flow

In the fringe interpretation of the dual-beam technique, it is obvious that a particle crossing the fringe pattern in one direction produces the same signal as a particle moving in the opposite direction. In the Doppler interpretation of the reference-beam technique, the beat frequency measured is the difference between two frequencies and there is
nothing to indicate which of the two frequencies is higher or lower. Hence in both basic Doppler techniques there is an ambiguity about the direction of flow.

However, in the dual-beam technique this ambiguity in direction can be removed by employing a couple of techniques discussed by Abbiss et al. [27]. One method is to change the frequency of one of the probe beams by using a Bragg cell. The beat frequency is then offset by the amount of this change in frequency. Whether the beat frequency increases or decreases in response to the moving scatterer will determine the direction of the motion. Another method is to use light beams polarized at right angles to each other (to produce two sets of fringes in the scattering area) and use polarization-sensitive detectors.

2.3.3 Frequency Resolution

For the sake of comparison, let us rewrite (2.5) and (2.23) for the beat frequencies in the single and dual beam case

\[ \Delta f = \frac{2v \cos \phi}{\lambda} \quad \text{(Doppler shift for the reference beam case)} \quad (2.27) \]

\[ \Delta f = \frac{2v \sin \theta \cos \phi}{\lambda} \quad \text{(Doppler shift for the dual beam case)} \quad (2.28) \]

Comparing (2.27) and (2.28), it is clear that the beat frequency for the two systems differ by a factor of \( \sin \theta \), where \( 2 \theta \) is the angle between the two laser beams in the dual-beam system. For almost all practical applications, this angle is usually very small. Hence the Doppler frequency measured in a dual-beam system is smaller than the
corresponding single beam system. Thus a single beam system with higher Doppler frequency gives a better frequency resolution.

From the discussions in the previous two sections, it is also clear that a dual beam system is sensitive to the motion of particles in the direction perpendicular to the bisector of the two beams; whereas a single beam system is sensitive to the motion of particles in the direction of the light beam. Moreover, with only one optical axes, the single beam system is easier to align and hence its setup is simpler than a dual-beam system.

2.3.4 Effect of Finite Velocity Distribution

Notice that in the discussion above, we have focused on scattering only from a single particle moving with a certain velocity. However, in real systems, there are large numbers of scattering particles moving with a distribution of velocity around some mean value. Hence, instead of having a single Doppler frequency, we will instead have a spread of frequencies in the Doppler spectrum around some mean value ($f_d$). In literature this effect is also sometimes referred to as the heterodyning effect. There is another phenomenon responsible for this finite linewidth of frequencies in the Doppler spectrum; the wavefront of light exiting the fiber-optic probes is not planer, i.e., there is a spread of angles in the light incident on the blood vessels. This spread in angles would directly correspond to a spread in the Doppler frequencies measured using the instrument.

In addition to the mixing of scattered light from particles moving with a finite velocity distribution with the reference beam to give the Doppler spectrum, another phenomenon also occurs. The light scattered from two particles moving with different velocities will mix with each other (as well as with the reference beam) to give a self-beating effect. The self-beating effect will hence give a distribution in the frequency
about the zero value, in addition to the spectrum we already have. This effect is also sometimes referred to as the homodyning effect. A typical spectrum thus obtained will look similar to the one shown in Figure 2.6.

Notice that the strength of Doppler signal with distribution about zero frequency is weak compared to the spectrum about some finite mean Doppler frequency. This is because the distribution about zero is obtained by optical beating of two relatively weak scattered light signals, whereas the higher frequency Doppler spectrum of interest is obtained by the optical mixing of a strong reference beam with a weak scattered signal. A high-pass filter is used in the actual setup to remove the low-frequency undesirable signals obtained due the self-beating effect.

Figure 2.6 Typical Doppler spectrum obtained by scattering from particles moving with a finite velocity distribution.
2.4 Laser Doppler Geometries

The laser Doppler velocimeter used for measuring blood velocities in our application was based on the reference beam technique discussed in the previous section. The setup was implemented using the principle of Michelson interferometer. However, we investigated three different configurations of our laser Doppler velocimeter, each of which had shortcomings that lead to our final design.

The setup of the first version of our laser Doppler velocimeter (called the ‘free-space version 1’) is shown in Figure 2.7.

![Figure 2.7 The ‘free-space version 1’ of LDV.](image-url)
Notice here that the light in the signal arm (the arm carrying light to the flow area) is focused on the moving scatterers by a fixed microscope objective. This fixed setup makes it extremely difficult for the light to be focused at exactly the flow area. Hence as an alternative, a second laser Doppler velocimeter (called the ‘free-space version 2’) was proposed and is shown in Figure 2.8.

In this version, the light coming out from the microscope objective in the signal arm is focused on a single mode optical fiber (Corning F-SV optical fiber, single mode at 633 nm, NA=0.12, Core Diameter=6-9 µm, Non-polarization preserving), which then carries it to the moving scatterers.

Figure 2.8 The ‘free-space version 2’ of LDV.
The optical fiber removes the problem due to fixed setup by having an easier optical access to the flow area. Notice however that the light inside the interferometer is still guided in free space. Hence any phase disturbance in free space such as turbulence in the air seriously affects the performance of the velocimeter. Moreover, this setup is not based on the backscatter geometry, hence any phase noise uncommon to the signal and reference arm also affects the performance of the velocimeter.

Thus to remove these shortcomings a final version of the laser Doppler instrument (called the ‘fiber-optic version’) was proposed as shown in Figure 2.9.

![Figure 2.9 The ‘fiber-optic version’ of LDV.](image-url)
The ‘fiber-optic version’ of the laser Doppler velocimeter uses single mode optical fibers (Corning F-SV optical fiber, single mode at 633 nm, NA=0.12, Core Diameter=\(~6-9~\mu\text{m}\), Non polarization preserving) and a fiber power splitter to guide light throughout the interferometer and hence remove any noise due to air turbulence etc. The laser light (HeNe laser, 632.8 nm, ~1-2 mW) coupled into a single mode fiber passes through the 3 dB fiber-optic beam splitter and is transmitted to the probe. To improve the noise performance of the system, the backscatter geometry is used where the reference beam (shown in Figure 2.9) is absorbed by a black absorber. The light from the probe is aimed at the moving object and the frequency-shifted scattered light is collected simultaneously by the same fiber-optic probe. The frequency-shifted light along with the light scattered back from the stationary walls of the vessel and reflected from the distant end of the fiber travel to the detector through the beam splitter. The beam reflected from the stationary walls of the blood vessel and the fiber end now acts as the reference beam [28]. At the detector the Doppler-shifted beam and the combined reference beam interfere to give the Doppler signal.

The advantage of having just a single arm in the interferometer is that any noise inherent in the system is now common to both the signal and the reference arm and thus cancels during interference. It is also easier to control the interferometer as there is only one optic axis in the system. Moreover, removing the extra arm (reference arm) simplifies the instrument.

The system and beam characterization parameters for the ‘fiber-optic version’ of the LDV are reported and discussed in the next chapter. The system calibration procedure and the validation results are also presented in the next chapter.
We will now briefly compare the single beam laser Doppler technique used for velocity measurements in our application with other velocity measurement techniques discussed in the previous chapter. The various velocity measurement techniques discussed in the previous chapter are not practical for our application due to the following reasons:

1. The probe (transducer) size in UBM-Doppler is too big compared to the size of the blood vessels in our applications.

2. The small vessel size makes it difficult to access them. So it is difficult to put a camera and image them at regular intervals during Particle Image Velocimetry (PIV).

3. Moreover, the scattering particles in blood flow are the red blood cells (RBC) and it is difficult to track individual RBC’s under camera images of the vessel using PIV.

4. Due to non common beam paths in self-mixing interferometry, any noise in the signal arm shows up as a signal at the detector and thus seriously reduces the SNR of the system.
CHAPTER 3

DEVICE CHARACTERIZATION AND VALIDATION

The validation and characterization procedure of the fiber-optic laser Doppler velocimeter is discussed in this chapter. The procedure for the calculation of device characterization parameters such as signal-to-noise ratio (SNR), stability, drift, etc. is explained and the results are discussed in a section later in the chapter. A section is also dedicated to explain a novel technique developed to simplify accurate measurements of the angle between the light and the velocity vector.

3.1 System Calibration

The laser Doppler velocimeter has been calibrated by a rotating disk with white paper attached to it. The rotating white reflective paper acts as random Gaussian scatterer, which is statistically similar to the scattering from moving particles such as red blood cells. The disk was rotated at a constant angular velocity \( \omega \) and the bare fiber-optic probe was placed next to the rotating disk. The angle \( \varphi \) that the light vector made with the linear velocity vector was predetermined using a procedure similar to one discussed later in the chapter. The distance \( d \) of the tip of the fiber-optic probe from the centre of rotation was varied. By varying this distance, the linear velocity \( v_{\text{calc}} \) of the rotating disk was varied and calculated from (3.1) below. The calculated velocity is given as

\[
 \text{(3.1)}
\]
\( v_{\text{calc}} = \omega \ast d \) \hspace{1cm} (3.1)

Placing the fiber-optic tip next to the rotating disk gives a frequency-shifted signal due to scattering from the white reflective paper. This backscattered frequency-shifted light is collected by the same fiber-optic probe and mixed coherently with a reference light beam to obtain the Doppler signal. Note here that the reference signal does not come from scattering due to some stationary target near the rotating disk. Instead, it comes from Fresnel reflections at the glass-air interface on the exit tip of the fiber-optic probe.

The Doppler waveform thus obtained was collected through a National Instruments data acquisition card (NI DAQCard-6062E) at the rate of 40000 samples per second. The frequency components of the signal were calculated with a FFT routine in Matlab using a window size of 512 samples. Breaking the data into 512 sample size windows gives 781 windows over a 10 second acquisition time. The FFT over all the 781 windows were averaged to obtain the mean Doppler spectrum of the signal. The measured mean Doppler frequency \( (f_d) \) of the signal was calculated from this mean Doppler spectrum using

\[ f_d = \frac{\sum (A_i f_i)}{\sum A_i} \] \hspace{1cm} (3.2)
here $A_i$ is the amplitude of the frequency component $f_i$. From this Doppler frequency, the measured linear velocity ($v_{\text{meas}}$) of the rotating disk was calculated using the Doppler equation (2.5).

In our calibration experiment, the angular frequency ($\omega$) of rotation was kept constant at $(2\pi / 60)$ s$^{-1}$. The angle ($\phi$) between the light propagation vector and linear velocity vector was measured directly using a protractor to be approximately 45°. This angle was measured to within an accuracy of $\pm 0.86\%$ (or $\pm 0.5^\circ$). This accuracy in the measurement of angle corresponds to a variation of $\pm 1\%$ in the measurement of velocity (from 2.5). By varying the distance ($d$) of the fiber end from the centre of rotation of the disk between 10 mm to 70 mm, the linear velocity of the disk was varied. Five trials at each of seven velocities were run (not in sequence) and the mean measured velocity ($v_{\text{meas}}$) and standard deviation of these trials were plotted versus the calculated velocity ($v_{\text{calc}}$). Figure 3.1 shows the correlation between measured and calculated values. The error bars showing the standard deviation at various velocities are not clearly visible in the figure owing to their small values (typical values are in the order of 0.02 mm/sec) compared to the velocities measured (1.0 mm/sec to 7.0 mm/sec).

Along the 45° line in Figure 3.1 the measured and calculated velocities are equal. The statistics of the calibration data is shown in Table 3.1. By a comparison of the measured calibration data with the 45° line, it can be calculated that the device has an accuracy of $\pm 1\%$. Further we define repeatability as one standard deviation in the calibration data. Using Table 3.1, the repeatability in the calibration experiment was found to be $\pm 2.2\%$. 
Figure 3.1 Calibration curve displaying the measured versus the calculated velocity.

Table 3.1 Linear Regression Line Fit Data.
The average spectrum of the Doppler signal collected from the rotating wheel during the calibration experiment for the case of $\omega = (2\pi / 60) \text{ s}^{-1}$ and $d=40 \text{ mm}$ is shown in Figure 3.2. If we look closely at the average Doppler spectrum in Figure 3.2, it can be noticed that the tail on both sides of the spectrum does not die off to the zero value due to broadband noise in the temporal signal. Instead, it remains at some threshold value throughout the entire spectrum. It can also be noticed that due to the non ideality of high-pass filter used in the experimental setup, a signal component shows up near the dc origin. These unwanted noise signals throughout the frequency spectrum cause the measured mean Doppler frequency (and hence the velocity) to be corrupted.

![Figure 3.2 Example of average spectrum of the Doppler signal from the rotating wheel.](image-url)
In order to remove the effects due to these unwanted noise components, signal conditioning is done in the acquired Doppler signal as follows: In calculating the measured velocity ($v_{meas}$), amplitude of all the frequency components in the Doppler spectrum that were two standard deviations away from the mean frequency on either sides was set to zero. Moreover, during the analysis of the acquired data in Matlab, the data are passed through a ‘virtual’ high pass filter to remove the undesirable effects due to non-ideality of the filters. The average Doppler spectrum after such thresholding and filtering is shown in Figure 3.3.

Figure 3.3 Average spectrum of the Doppler signal after signal conditioning.
It was found that after this signal processing of the Doppler spectrum the accuracy in the measurements of the velocity increased from 10% to 1%. The measured velocity values plotted in Figure 3.1 depict velocities after signal processing on the acquired data was done.

3.2 System Characterization

The laser Doppler device was characterized by the procedure discussed below and the calculation of system parameters, such as SNR and stability are discussed next.

3.2.1 Stability and Drift

The stability and the drift of the laser Doppler velocimeter were calculated from the data collected during the calibration experiment described in the previous section. As mentioned in the previous section, the system calibration experiment was carried out in five trials, each at seven different velocities. This experiment roughly took three and a half hours to complete. In Figure 3.4, the time course of the calibration data acquired using the laser Doppler device over this time period is shown.

A linear regression line was fitted to the calibration data at each individual velocities and the corresponding slope and intercepts were calculated as shown in Table 3.1. From the linear regression line fit data, a mean change of $11 \times 10^{-5}$ mm/sec in velocity over a period of one minute was noticed. Hence the drift in velocity over a time period of one hour was calculated for each individual velocity using (3.3).

$$Drift = \frac{slope \ in \ mm/\ sec/\ min}{mean \ in \ mm/\ sec} \times 60 \ min/\ hour \times 100\%$$

(3.3)
Figure 3.4 The time course of the calibration data.

From the regression data shown in the Table 3.1, a mean drift of 0.16% in velocity over a period of 1 hour was determined.

3.2.2 Signal-to-Noise Ratio (SNR)

Typical average Doppler spectra obtained due to scattering from both the rotating white wheel and the blood vessel of chick embryo are shown in Figures 3.5 and 3.6. From these spectra, the typical SNR of the laser Doppler device can be calculated for both cases. The mean Doppler frequency for the spectrum from a rotating white wheel shown in Figure 3.5 is 9.5 kHz and the Full Width at Half Maximum (FWHM) is 2.0 kHz.
Figure 3.5 A typical average spectrum of the Doppler signal from rotating white wheel.

Figure 3.6 A typical average spectrum of the Doppler signal from blood vessel of chick embryo.
Similarly the mean Doppler frequency for the spectrum obtained from the blood vessel of a chick embryo shown in Figure 3.6 is 2.42 kHz and the corresponding FWHM is 1.43 kHz. Based on these plots we define the amplitude of the signal level as the amplitude of the mean Doppler frequency component; whereas, the amplitude of the noise level is defined as the average signal amplitude of all the frequency components that are beyond twice the FWHM from the mean frequency on either sides. From the discussion in the previous chapter, the signal level depends on the concentration of scattering particles in the flow; hence the signal levels are weak in chick embryos, where the concentration of red blood cells is relatively low, compared to scattering from a rotating white wheel. The signal level is also expected to be low because of the broader distribution of velocities around the probe sitting close to a beating heart. Thus it can be expected that the SNR reduces as we go from rotating wheel to a blood vessel. Typical SNR’s for both these cases were calculated using

\[
SNR = 20 \log_{10} \left( \frac{\text{Signal Amplitude}}{\text{Noise Amplitude}} \right)
\]

(3.4)

where the signal and noise amplitudes are obtained using the procedure described earlier from the Doppler spectra shown in Figures 3.5 and 3.6. Using these values of signal and noise amplitudes, the SNR values for the two cases were calculated as:

\[
\text{SNR}_{\text{wheel}} = 37.57 \text{ dB}
\]

and

\[
\text{SNR}_{\text{vessel}} = 24.62 \text{ dB}
\]
As expected the SNR reduces in blood vessels compared to the rotating wheel.

### 3.3 Angle Measurement

One of the challenges in calculating the velocity of a moving target from the Doppler equation given by (2.5) in the previous chapter is to determine the angle $\phi$ between the light propagation vector and the velocity vector. The angle determination is difficult as we are working inside a chick embryo with a 125 µm diameter fiber-optic probe shining light on a ~20 µm blood vessel. Moreover, to add to the complexity, the view of the embryo is also limited to a single two-dimensional plane (i.e. the focal plane of the microscope). A method for accurately determining this angle is proposed and explained in further detail below.

Let us depict the geometry of the angle determination setup with respect to a Cartesian coordinate system, with its side and top view shown in Figure 3.7. The three coordinate axes are defined as shown in the figure, with the microscope objective focusing on the X-Y plane. The microscope objective in Figure 3.7 can be moved up and down (with the help of a focusing knob on the side of the microscope) along the Z axis. This movement of microscope objective in the direction perpendicular to the plane of focus (i.e. along the Z axes) can be calibrated in absolute distance, for example by determining the number of turns needed to move the focus of objective from the top to the bottom surface of a 1mm thick glass slide. The top and side view of a typical geometry inside the chick embryo, i.e., a fiber-optic probe shining light on the blood vessel are depicted with reference to the Cartesian geometry in Figure 3.7.
Figure 3.7 Angle measurement setup depicting the geometrical measurements.
The variables used in Figure 3.7 are described below:

A1, A2: Two markers placed on the fiber-optic probe.

B1, B2: Two distinguishable point features on the blood vessel as seen with the microscope.

$\Delta Z_1$: Vertical height between the points A1 and A2.

$\Delta Z_2$: Vertical height between the points B1 and B2.

$\Delta L_1$: Horizontal length between points A1 and A2 as measured in the X-Y plane.

$\Delta L_2$: Horizontal length between points B1 and B2 as measured in the X-Y plane.

$\alpha$: Angle as measured in the X-Y plane between the projections of the fiber-optic probe and blood vessel onto that plane.

$\beta_1$: Angle between fiber-optic probe and the X-Y plane.

$\beta_2$: Angle between blood vessel and the X-Y plane.

A camera image of this configuration, taken by placing a camera on the microscope will look similar to the top view shown in Figure 3.7 (i.e., it will image the X-Y, or focal plane of the microscope). From this image, the values of $\Delta L_1$, $\Delta L_2$ and $\alpha$ can be directly measured. It should be noted here that a proper scaling factor must be applied to all the distances measured in the picture to compensate for the magnification provided by the microscope. The scaling factor is determined by measuring the diameter of the bare fiber portion of the probe in the image and calibrating it against its known diameter (125 µm).

As discussed, the movement of microscope objective in the Z (focus) direction is calibrated. Hence, by moving the microscope objective along the Z axes in such a way to
shift the focus from marker A1 to A2 along the fiber-optic probe (and similarly from marker B1 to B2 along the blood vessel), distances $\Delta Z_1$ and $\Delta Z_2$ can be determined.

Using these values and applying simple trigonometry to the geometry of Figure 3.7, we can determine the value of angles $\beta_1$ and $\beta_2$ from the following:

\[
\tan \beta_1 = \frac{\Delta Z_1}{\Delta L_1} \quad (3.5)
\]

and

\[
\tan \beta_2 = \frac{\Delta Z_2}{\Delta L_2} \quad (3.6)
\]

From the sign convention for angle we will be following, $\beta_1$ is taken positive if A2 is below A1 and negative if A2 is above A1. Similarly $\beta_2$ is taken positive if B2 is above B1 and negative if B2 is below B1. These angles are to the angle of interest $\phi$, which is the angle between the light propagation vector and the blood velocity vector, in the manner shown in Figure 3.8.

From Figure 3.8, it is clear that the light propagation and velocity vectors are making angles $\beta_1$ and $\beta_2$, with the focal plane of the microscope respectively. There will be two cases when these vectors are on opposite sides of plane of focus (when both $\beta_1$ and $\beta_2$ are positive or both are negative). One of these cases is depicted schematically in Figure 3.8. For these two cases both $\beta_1$ and $\beta_2$ will have same signs from the conventions we are following. However, there are two other cases when both these vectors can be on the same side of plane of focus (both are either above or below the plane of focus). For these two cases, angles $\beta_1$ and $\beta_2$ will have opposite signs. One of these cases is depicted schematically in Figure 3.9.
Figure 3.8 Relationship between various angles in Cartesian geometry when light propagation vector and velocity vector are on opposite sides of plane of focus. In this case both $\beta_1$ and $\beta_2$ are positive numbers from the sign convention we are following.

Figure 3.9 Relationship between various angles in Cartesian geometry when light propagation vector and velocity vector are on same sides of plane of focus. In this case $\beta_1$ is positive and $\beta_2$ is a negative number from the sign convention we are following.
Now we determine the relationship between the angle of interest $\phi$ in terms of the known angles $\beta_1$, $\beta_2$ and $\alpha$ using the Cartesian geometry of Figure 3.8. The other three cases mentioned above are just special cases of the geometry depicted in Figure 3.8.

Assuming in Figure 3.8,

$$\text{OA} = 1 \text{ units} \quad (3.7)$$

Applying simple trigonometry to right triangle OAB, we can write

$$\text{AB} = \sec \beta_1 \quad (3.8)$$

and

$$\text{OB} = \tan \beta_1 \quad (3.9)$$

Similarly from right triangle OAD, we have

$$\text{AD} = \sec \alpha \quad (3.10)$$

and

$$\text{OD} = \tan \alpha = \text{EC} \quad (3.11)$$

Applying same identities to right triangle ADC, we get

$$\text{DC} = \sec \alpha \tan \beta_2 = \text{OE} \quad (3.12)$$

and

$$\text{AC} = \sec \alpha \sec \beta_2 \quad (3.13)$$
Now from right triangle BEC, we can write the hypotenuse BC in terms of the other sides of the triangle, by using Pythagoras theorem, as

\[ BC^2 = BE^2 + EC^2 \]

Expanding BE in terms of BO and OE, we get

\[ BC^2 = (BO + OE)^2 + EC^2 \]

Substituting the values of BO, OE and EC from (3.9), (3.12) and (3.11) respectively, we get

\[ BC^2 = \tan^2 \beta_1 + \tan^2 \beta_2 \sec^2 \alpha + 2 \tan \beta_1 \tan \beta_2 \sec \alpha + \tan^2 \alpha \quad (3.14) \]

From (3.8), (3.13) and (3.14), the lengths of all the three sides of the triangle ABC are known. Remember here that the sides AB and AC of this triangle represent the light propagation and the velocity vector respectively. Hence the angle \( \phi \) between these two vectors can be determined using the following geometrical identity

\[ \cos \phi = \frac{AB^2 + AC^2 - BC^2}{2 \ AB \ AC} \quad (3.15) \]

Substituting the values from (3.8), (3.13) and (3.14) on the RHS of (3.15), we get
\[
\cos \varphi = \frac{\sec^2 \beta_1 + \sec^2 \alpha \sec^2 \beta_2 - \sec^2 \alpha \tan^2 \beta_2 - \tan^2 \beta_1 - 2 \tan \beta_1 \tan \beta_2 \sec \alpha - \tan^2 \alpha}{2 \sec \alpha \sec \beta_1 \sec \beta_2}
\]

From trigonometry, we already know that for any angle \( \theta \) of a right triangle

\[
\sec^2 \theta - \tan^2 \theta = 1
\]

Hence, using this identity in the equation for \( \cos \varphi \) above, we have

\[
\cos \varphi = \frac{1 - \tan \beta_1 \tan \beta_2 \sec \alpha}{\sec \alpha \sec \beta_1 \sec \beta_2}
\]

Solving,

\[
\cos \varphi = \cos \alpha \cos \beta_1 \cos \beta_1 - \sin \beta_1 \sin \beta_2
\]

Solving for \( \varphi \), we get

\[
\varphi = \cos^{-1}\{\cos \alpha \cos \beta_1 \cos \beta_1 - \sin \beta_1 \sin \beta_2\} \quad (3.16)
\]

From the value of angles \( \alpha, \beta_1 \) and \( \beta_2 \) determined from the experimental procedure described above, the angle \( \varphi \) can be determined from (3.16) above.
A pseudo-phantom representing the actual optical geometry inside the embryo was built using two optical fibers glued together at an angle representing the fiber-optic probe and the blood vessel, respectively. Two markers on each of the two optical fibers were placed to represent the points A1, A2, B1 and B2 discussed above. The eye piece of the microscope can be replaced with a camera having lens of similar dimensions. The camera can be fixed on the eye piece socket of the microscope to take pictures of the plane of focus (X-Y plane), at different positions of the objective along the Z axes. The phantom was placed at 10 different randomly chosen orientations under the objective, with one of the geometries depicted in Figure 3.10. Figure 3.10 is exactly similar to Figure 3.7, except that the geometrical distances and angles in the geometry are depicted in a greater detail. The angle between the two optical fibers at these 10 positions was then determined from these measurements, following the procedure described above.

The angle between the two optical fibers of the phantom was determined for each of these 10 different randomly chosen orientations of the phantom using the procedure described above. These orientations of the rigid phantom (with fixed angle \( \phi \)) were chosen such that most of the different possibilities of combination for angle \( \alpha, \beta_1 \) and \( \beta_2 \) (which ultimately determine \( \phi \)) were covered as shown in Table 3.2. The fixed angle \( \phi \) between these optical fibers was determined by direct measurement with a protractor. The angles measured above were then compared with the actual angle between the two optical fibers and an accuracy of \( \pm 2.41\% \) was determined in the measurement technique.
Figure 3.10 The general setup for angle measurement.
The actual angle between the two optical fibers in the pseudo-phantom measured using a protractor was 43.50°. From the angle measurement data of Table 3.2 the mean of measured angle $\varphi$ is determined to be 42.45° and the standard deviation of angle $\varphi$ is found to be 2.63°.

It should be mentioned here that there will be an uncertainty in the measured angle due to the depth of focus of microscope objective used in the measurement setup. In simple terms, depth of focus is defined as the vertical distance along the optic axes (z-axis in Figure 3.7) through which the features within the image are simultaneously in focus. Hence there will be a spread in the measurement of distances $\Delta Z_1$ and $\Delta Z_2$, which

<table>
<thead>
<tr>
<th>Orientation</th>
<th>Angle $\alpha$ (Degrees)</th>
<th>Angle $\beta_1$ (Degrees)</th>
<th>Angle $\beta_2$ (Degrees)</th>
<th>Angle $\varphi$ (Degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle 1</td>
<td>27.5</td>
<td>10.66</td>
<td>24.05</td>
<td>43.91</td>
</tr>
<tr>
<td>Angle 2</td>
<td>33</td>
<td>10.86</td>
<td>22.25</td>
<td>46.29</td>
</tr>
<tr>
<td>Angle 3</td>
<td>36.5</td>
<td>15.52</td>
<td>12.21</td>
<td>45.54</td>
</tr>
<tr>
<td>Angle 4</td>
<td>12.5</td>
<td>13.07</td>
<td>26.39</td>
<td>41.31</td>
</tr>
<tr>
<td>Angle 5</td>
<td>29</td>
<td>18.19</td>
<td>15.43</td>
<td>44.12</td>
</tr>
<tr>
<td>Angle 6</td>
<td>7.5</td>
<td>9.44</td>
<td>-47.78</td>
<td>38.86</td>
</tr>
<tr>
<td>Angle 7</td>
<td>36</td>
<td>10.96</td>
<td>-35.02</td>
<td>40.57</td>
</tr>
<tr>
<td>Angle 8</td>
<td>42.5</td>
<td>16.74</td>
<td>-25.01</td>
<td>40.39</td>
</tr>
<tr>
<td>Angle 9</td>
<td>44</td>
<td>5.01</td>
<td>-12.81</td>
<td>44.10</td>
</tr>
<tr>
<td>Angle 10</td>
<td>29</td>
<td>7.22</td>
<td>-35.16</td>
<td>38.57</td>
</tr>
<tr>
<td>Angle 11</td>
<td>42</td>
<td>11.76</td>
<td>0</td>
<td>43.31</td>
</tr>
</tbody>
</table>
would result in a corresponding uncertainty in the measurement of angle \( \varphi \). The depth of focus \( (D) \) for a microscope objective is expressed as [29]

\[
D = \frac{2\lambda \ n}{NA^2}
\]  

(3.17)

where \( \lambda \) is the wavelength of incident light, \( n \) is the refractive index of the medium and \( NA \) is the numerical aperture of the microscope objective. The microscope objective used in the experiments had a numerical aperture of 0.40 and a magnification of 20x. The wavelength of incident radiation is in the visible regime (~530 nm) and the refractive index of the medium was 1 (air) for our verification study here. Using these values, the depth of focus is calculated to be 6.25 \( \mu \)m. From the angle measurement data, a variation of 6.25 \( \mu \)m in measuring distances \( \Delta Z_1 \) and \( \Delta Z_2 \) results in a corresponding uncertainty of 0.27\% in measuring the angle \( \varphi \). This small spread in the measured angle due to the depth of focus of microscope objective is insignificant because of the cosine dependence of \( \varphi \) in the Doppler equation.

The next chapter is entirely dedicated to discuss the characterization of the beam emanating from the bare and lensed fiber-optic probe. A detailed analysis of the design considerations and procedure for fabricating lenses at the tip of an optical fiber is also presented.
CHAPTER 4

PROBE CHARACTERIZATION AND LENS DESIGN

Besides calculating the system parameters such as SNR and stability of the laser Doppler (LD) device, one of the important parameters in the characterization of the LD device is to analyze the beam emanating from the fiber-optic probe. Two different probes, namely the bare and the lensed fiber-optic probe, were used in our device. The characterization of the beam emanating from both these probes is discussed in this chapter. The beam characterization procedure and the results for both the optical probes are presented below. A detailed analysis of the design considerations to be taken into account while fabricating lenses at the end of an optical fiber is presented. The beam characterization parameters for the lensed probe, such as beam waist and working distance, are measured by observing the focused beam under a microscope. A comparison of the measured values with the expected theoretical values is presented.

4.1 Bare Fiber-optic Probe

The bare fiber-optic probe is a single-mode optical fiber, stripped and cleaved at one end and connectorized with a FC connector at the other. The beam emanating from the probe was analyzed using the procedures discussed below. Two different methods were used to analyze and calculate the size of the beam exiting the bare probe. In the first approach, the beam exiting the bare probe diverges in space due to the diffraction effect, and it is assumed that the diameter of beam at the exit tip of bare probe is equal to the
beam waist of a Gaussian beam. Note here that all the further calculations in this section are carried out under the assumption that the beam emanating from the fiber-optic probe is Gaussian in spatial profile.

From the theory of the propagation of Gaussian beams in free space [30], the equation for the radius of a Gaussian beam at a distance \( z \) from the beam waist is given by

\[
R(z) = R_o + z w_o \left(1 - \frac{z}{z_R}\right)
\]

(4.1)

Here \( R(z) \) is the radius of the beam at a distance \( z \) from the beam waist, \( 2R_o \) is the beam waist diameter and \( z_R \) is the distance which the beam travels from the waist before the beam diameter increases by \( \sqrt{2} \), called the Rayleigh range. It is expressed by the following equation:

\[
z_R = \frac{\pi w_o^2}{\lambda}
\]

(4.2)

Here \( \lambda \) is the wavelength of the laser light in the medium of propagation. However, in the far-field region, \( z \gg z_R \), thus (4.1) can be rewritten as

\[
w(z) = \frac{w_o z}{z_R}
\]

(4.3)
From (4.2) and (4.3), solving for the beam waist diameter $2w_o$, we get

$$2w_o = \frac{2z\lambda}{\pi w(z)} \quad (4.4)$$

Hence, the beam waist of a Gaussian beam of wavelength $\lambda$ can be calculated by measuring the radius of the beam at a distance $z$ from the tip of the optical fiber.

The far-field pattern of the beam from the bare-tip probe in air was observed with a CCD camera and a Spiricon® laser beam analyzer assembly. The radius of the beam emanating from a bare probe was measured to be 1650 micrometers, at a distance of 2.5 cms from the fiber tip. Substituting these values in (4.4), the waist diameter of the beam was calculated to be 6.19 micrometers. Note that this value of the beam waist is approximately equal to the core size (~6-9 microns) of the single-mode optical fiber.

In the second method, the beam exiting the bare probe was analyzed by observing the fluorescence in a solution of Rhodamine 6G (R6G) fluorescence dye into which the probe end was immersed. The excitation wavelength for R6G dye is at green wavelengths; hence a tunable Argon-Ion laser working at a wavelength of ~514 nm was used to couple light to the bare probe. The fluorescence was observed under a high magnification microscope and a camera image of this fluorescence excited by the light exiting the probe was taken, shown in Figure 4.1.
Figure 4.1 Fluorescence excited by light exiting a bare probe observed in R6G fluorescence dye.

From Figure 4.1, the diameter of beam immediately exiting the probe can be calculated by measuring the beam diameter in the image and multiplying it with a scaling factor (to account for the magnification from the microscope). The scaling factor is determined by dividing the known clad diameter (125 micrometers) of the optical fiber by the clad diameter measured in the camera image. The beam waist diameter calculated by this method was found to be 5.32 micrometers, which is in close agreement to the value determined by the first approach.

The divergence angle of the expanding beam can also be calculated from the known numerical aperture of the single-mode fiber. The single-mode fiber used for our experiments was an F-SV optical fiber from Corning Inc, New York, with a numerical
aperture of 0.12. The half divergence angle in water, with refractive index of \( n = 1.33 \), can be calculated from

\[
n \sin \theta = NA
\]  

(4.5)

Solving (4.5) for \( \theta \), the calculated value of half divergence angle in water is \( \theta = 5.17^\circ \). The measured value of this half angle from Figure 4.1 is 5.04\(^\circ\), which is in close agreement with the value calculated above.

There are, however, certain disadvantages with the bare fiber-optic probe, and hence the need for a lensed probe, as described next.

4.2. Lensed Fiber-optic Probe

From the discussions in the previous section, it is known that the beam emanating from a bare fiber-optic probe expands due to the diffraction effect. Due to this diffraction effect, the focusing of light from a bare fiber-optic probe to the minute blood vessels is not very strong. This results in a decrease in the strength of the light scattered by the moving red blood cells. The diverging beam also results in scattering from surrounding tissues, as not all the light is concentrated on the desired blood vessels. Since the tissues surrounding the vessels may also be moving, this results in the probe picking up undesired noise signals superimposed on the desired Doppler signal. (Notice, however, that most of these noise signals are of low frequencies and may be partially filtered out by the high-pass filter in the experimental setup.)

To reduce the noise due to beam divergence as well as to increase the strength of light scattered by the moving red blood cells (by increasing the focusing), a lensed fiber-
optic probe is proposed. It is a single-mode optical fiber with a spherical lens element at its end to focus light exiting the bare probe on the minute blood vessels. These micro-lenses were fabricated in the lab and characterized using the theory of geometrical and Gaussian beam optics. Design considerations for fabricating these lenses are discussed in the next section, and the lens parameters such as working distance and spot size are calculated and compared with the expected theoretical values in a section later in the chapter.

4.3 Lens Fabrication and Design Considerations

Two methods were used to fabricate lenses at the end of an optical fiber. In the first method, the melting method, the tip of a bare optical fiber was melted in an arc fusion splicer. The melted glass at the fiber tip takes the shape of a sphere due to surface tension and hence acts as a focusing element for the light exiting the optical fiber.

In the second method, micro-spherical balls made of materials with refractive indices higher than glass were affixed at a distance from the tip of an optical fiber using epoxy. These micro-spherical balls act as the focusing element for light exiting the bare fiber. Both of these focusing arrangements can be analyzed under geometrical optics approximations. Two specific geometries for the refraction of light at a spherical interface are shown in Figure 4.2.

In Figure 4.2, a point source is placed at a distance $S_o$ from a spherical interface between media with refractive indices (R.I.) $n_1$ and $n_2$. The radius of curvature $R$ of the interface is chosen positive for one case and negative for the other. All the rays emitted by the point source diverge in medium with refractive index $n_1$ until they encounter the medium with different refractive index $n_2$. 
Figure 4.2 Refraction of light at a spherical interface between media with different refractive indices. (a) Case A for $n_1 < n_2$ and $R$ positive value, (b) Case B for $n_2 < n_1$ and $R$ negative value.

From the Snell’s law, these rays on entering medium 2 will either bend towards or away from the normal to the interface, depending on the values of $n_1$, $n_2$ and $R$. If $n_2 > n_1$ and $R$ is positive or if $n_1 > n_2$ and $R$ is negative, then all the rays entering medium 2 will converge towards a focal point at a distance $S_i$ from the interface as shown in Figure 4.2. The source and image distances are related to each other and to the refractive indices of the two media and the radius of curvature by the “lens maker’s formula” [31], given by
\[
\frac{n_1}{S_o} + \frac{n_2}{S_i} = \frac{n_2 - n_1}{R}
\]

(4.6)

It should be noted here that the distances \( R, S_o \) and \( S_i \) are taken positive in the directions shown in Figure 4.2(a); whereas in the Figure 4.2(b), \( S_o \) and \( S_i \) remains positive with \( R \) changing sign. We will follow the same sign convention throughout the entire discussion ahead. Also note that (4.6) is only valid for rays traveling under the paraxial approximation; i.e. for rays arriving at the interface at shallow angles with respect to the optic axis.

4.3.1 Melted Tip Design and Fabrication

In this section the design and fabrication of fiber-optic probes fabricated using “the melting method” is discussed. The two design methodologies used to fabricate lensed probes using this method and the resulting beam images are shown next.

4.3.1.1 Melted Tip Design

Two extremes for an optical beam traveling through an interface between media with refractive indices \( n_1 \) and \( n_2 \) are shown in Figures 4.3 and 4.4. Note in these figures that the radius of curvature is negative from the sign convention we are following.

In the first extreme the image is formed at infinity, whereas in the second the source is placed at infinity. The corresponding source and image distances are referred to as the first and the second focal lengths (\( f_1 \) and \( f_2 \)) respectively.
Figure 4.3 Geometry for image formation at infinity, with the source placed at the first focal point.

Figure 4.4 Geometry for the source placed at infinity, the image is formed at the second focal point.
The expressions for the first and second focal lengths are derived using (4.6) and are given by

\[ f_1 = \frac{n_1 R}{n_2 - n_1} \]  \hspace{1cm} (4.7)

and

\[ f_2 = \frac{n_2 R}{n_2 - n_1} \]  \hspace{1cm} (4.8)

It can easily be interpreted from the Figures 4.3 and 4.4 that for focusing the source can’t be closer to interface than \( f_1 \) and the focal point can’t be closer than \( f_2 \) to interface. Hence the optical beam can only focus if the distance of source from the interface is between the two extremes. At all other source distances, the image will be virtual because the image distance reduces from infinity to \( f_2 \) as one move the source farther away from \( f_1 \).

Using the equation for the first focal length given by (4.7), we can calculate \( f_1 \) to be equal to \( 3R \) for a ray going from glass (R.I. =1.5) to air (R.I. =1.0) through an interface with radius of curvature \( R \). Note, however, that for our applications, the rays are going from glass into tissues (R.I. approx. equal to that of water, \( n_2=1.33 \)). Hence, the corresponding \( f_1 \) changes to \( 9R \). From a design point of view, \( f_1 \) is the closest the source (i.e. the core) can be to the interface, to get any focusing from the interface.

Hence, this distance of \( f_1=9R \) sets one limit on the closest distance of the source from the interface. We, however, also want all the rays leaving the core to pass through the spherical interface before meeting any transverse boundaries. That is, it is not desirable for the diverging rays leaving the core, to meet the air-glass interface in the
transverse direction before passing through the spherical interface in the direction of beam travel. This is known as the case of clear aperture. Hence, this sets the limit on the farthest the source can be from the interface. This limit on the distance of the source for the case of clear aperture can be calculated from the half divergence angle of the light leaving the single mode fiber in glass. From the formula for the numerical aperture used in previous section, for \( n=1.5 \), the half divergence angle is calculated to equal to \( \theta=4.59^\circ \). Hence, using simple trigonometry, the equation for the maximum source distance (\( S_{o_{\text{max}}} \)) is determined as,

\[
S_{o_{\text{max}}} = \frac{R}{\tan \theta}
\]

Thus for \( \theta=4.59^\circ \), the maximum source distance is calculated to be \( 12.46R \) for a spherical interface with radius of curvature \( R \). This distance of \( 12.46R \) therefore sets another limit on the distance of core from the interface. It can thus be concluded that for the proper focusing of light through a spherical interface, the distance of the source should be between these two extremes; i.e. between distance of \( 9R \) to \( 12.46R \) from the spherical interface between glass and water.

For \( n_1=1.5 \) and \( n_2=1.33 \), a relationship between the source and image distances (\( S_o \) and \( S_i \)) in terms of \( R \), is determined using the lens maker’s formula. A plot showing the relationship between source and image distances for different values of \( R \) is shown in Figure 4.5. It should be noted here that this plot holds for lenses manufactured using the melting method because of reasons that will become clear later in the section.
Figure 4.5 Relationship between the source and image distances, in lenses fabricated using the melting method, for different values of radius of curvature.

Figure 4.5 shows the relationship between source and image distances for glass-water interfaces with five different radius of curvature. The black markers in each plot depict the desired operating region to obtain a decent focusing from one of these interfaces. The circle mark on these plots represents the maximum source distance from the interface for clear aperture. The bold red line at the bottom of Figure 4.5 depicts the separation between the near and the far field. This transformation between the near and the far field occurs at a distance $Z_R$ (Rayleigh range) from the curved interface. The Rayleigh range over a range of beam diameters at the interface is also plotted in the
Figure 4.5. The beam diameter \( D \) at the interface is calculated from the half beam divergence angle \( \theta \) and the source distance \( S_o \) using (4.10).

\[
D = 2 S_o \tan \theta \tag{4.10}
\]

Using these design principles, two melted tip fiber-optic probes with different geometries were fabricated and characterized as discussed next.

4.3.1.1.1 Melted tip design 1. A schematic diagram of the fiber-optic probe fabricated by directly melting the tip of a single-mode optical fiber is shown in Figure 4.6. These tips were fabricated based on the design considerations discussed in the previous section.

As already discussed, there should be an optimum distance between the core and spherical interface for these lenses to focus. In the melting method, the fused tip of the optical fiber gets spherical in shape due to surface tension. The source, i.e., the core after fusion is too close to the spherical interface and is hence nowhere near the operating region as shown in Figure 4.5. Moreover in this method, there is no control over varying the distance of source from the interface, so as to move the operating point towards the desired operating region. Hence, it is not possible to focus light exiting the bare fiber, by directly melting the fiber tip. The fluorescence images of beam exiting a bare and a melted tip single-mode optical fiber observed in R6G fluorescence dye are shown in Figures 4.7 and 4.8. It can be noticed that the beam exiting the melted probe is not focused, however just by directly melting the tip of the optical fiber the divergence of the beam exiting the probe is reduced.
Figure 4.6 Schematic geometry of beam propagation through a melted tip fiber-optic probe.

Figure 4.7 Fluorescence excited by light exiting a 6 micron bare probe observed in R6G fluorescence dye.
4.3.1.1.2 Melted tip design 2. As already discussed, the core of the single mode fiber is too close to the melted interface for the focusing to work. An alternate of fusing a small section of larger core diameter multimode fiber (MMF) after the single mode fiber (SMF) is proposed. The advantage of having a small section of MMF at the end of a SMF is to give the beam enough space to expand before meeting the spherical interface. Note that, now the exit tip of the MMF fiber is melted to form the spherical interface at the end. Moreover, just by varying the length of MMF fiber, the distance of source from the interface can be controlled and hence, it becomes easier to satisfy the optimum distance condition. A schematic diagram of the fiber-optic probe fabricated by fusing a SMF with a MMF and melting the tip of the MMF is shown in Figure 4.9.
The fluorescence images of light exiting a bare MMF and a melted tip probe fabricated by melting a SMF and a MMF together and then fusing the tip of the MMF are shown in Figures 4.10 and 11. In the figure, it can be noticed that the beam exiting the melted probe fabricated using this second design is also not focused even though the beam divergence is reduced. In principle, the light exiting these probes should focus based on the design methodology discussed above. However, due to difficulty in fabricating these probes with the available resources, we did not pursue this further.

4.3.1.1.3 Melted tip advantages and disadvantages. As already discussed, the core of a melted single mode fiber is too close to the spherical interface for light exiting the probe to focus. Moreover, there is no means of altering the distance of core from the interface; so as to move the operating point close to the operating region discussed in the previous section. However, these probes hold a couple of advantages over the bare optical fibers.
Figure 4.10 Fluorescence excited by light exiting a bare 100 micron core probe observed in R6G fluorescence dye.

Figure 4.11 Fluorescence excited by light exiting a fused SMF and MMF fiber with the 100 micron core of MMF melted to form a lensed structure observed in R6G fluorescence dye.
Let us discuss these advantages by observing the two images shown in Figures 4.7 and 4.8 for light exiting a bare and a melted 6 micron core diameter probes. From these images, it can be noticed that the beam exiting the fused probe remains collimated for a longer distance. This decrease in divergence is due to refraction at the curved interface on the fiber tip. Due to this, the coupling of light from these probes to the blood vessels is better than with the bare probes. Moreover, using the concept of Total Internal Reflection and applying Snell’s law at the spherical interface between the glass and water, Kato [32] derived an expression for the half acceptance angle of light entering a spherical-ended fiber from air. The expression was modified by us to obtain the half acceptance angle $\theta$, for light entering a spherical-ended optical fiber from water given by (4.11).

$$\theta = \sin^{-1} \left[ \frac{1}{n_3} \sin^{-1} \left[ n_1 \sin^{-1} \left( \frac{d}{2r} \right) + \cos^{-1} \left( \frac{n_2}{n_1} \right) \right] - \sin^{-1} \left( \frac{d}{2r} \right) \right]$$ (4.11)

Here $d$ is the core diameter, $r$ is the radius of curvature of the fiber end, $n_3$ is the refractive index of water, and $n_1$ and $n_2$ are the refractive indices of the core and the cladding respectively. Note that, during the fusion process, the radius of curvature $r$ can be controlled by adjusting the volume of fiber tip under the electrodes of arc-fusion splicer. Optical fibers with four different radius of curvature of the fiber-tips are shown in Figure 4.12.

Also from (4.11), it can be inferred that the acceptance angle into the fiber can be altered by varying the value of $r$. A plot showing variation of the half acceptance angle ($\theta$) with $d/2r$ (and hence $r$, as $d$ is constant) is shown in Figure 4.13. This plot is similar to the one plotted by Kato [31], except that the input medium here is water.
Figure 4.12 Camera Images of melted fiber tips with different radius of curvatures.

Figure 4.13 Half-acceptance angle as a function of core diameter (d) and radius of curvature (r) of the spherical ended fiber.
It can be noticed from the figure that by decreasing the radius of curvature (i.e. by increasing $d/2r$), the acceptance angle into the fiber can be increased. Hence, the amount of light collected by the melted probes, due to scattering from the moving red blood cells, will be larger than the corresponding bare probes. That is, these probes have higher collection efficiency of the scattered light than the bare probes. Further advantages with the melted probes are the ease of manufacturing and high reproducibility during fabrication.

Hence it can be concluded that, although the beam emanating from the melted probes do not focus as well as the micro-ball probes; they, however, have better coupling of light into the blood vessels as well as higher collection efficiency of the scattered light, than the bare probes. For the rest of this chapter, the characterization procedure of the beam emanating from the micro-spherical probes will be discussed.

4.3.2 Ball Tip Design and Fabrication

In this section the design and fabrication of fiber-optic probes fabricated by using micro-spherical balls is discussed. The design methodology used to fabricate lensed probes using this method and the resulting beam images are shown next.

4.3.2.1 Ball Tip Design

The fundamental design idea behind fabricating lensed probes using micro-spherical balls is the similar to the one used in making lensed probes by fusing a SMF and a MMF together. It is to optimize the distance of source from the spherical interface so as to focus the beam exiting the probe. In this method, micro-spherical balls with diameters of 300 and 400 microns were glued at a distance from the end of the single
mode fiber using epoxy. A small section of quartz micro-capillary tubing with internal diameter of 250 microns was slipped over the buffer of a single mode fiber at one end. The diameter of unstripped fiber (i.e., fiber with buffer) is approximately equal to the internal diameter of the capillary tubing. These capillary tubing’s were then filled with epoxy from the other end, and micro-spherical balls were glued at the tip of these tubes. Special care was taken to remove any air bubbles in the epoxy while filling the capillary tubing. The epoxy filled capillary tubing now provides a clear region for the beam, leaving the core, to expand before meeting the spherical interfaces of the micro-spherical ball. By varying the length of micro-capillary tubing, the distance of source, i.e., the core, from the interface can be varied to satisfy the optimum distance condition. A schematic diagram showing the geometry of a fiber-optic lensed probe manufactured using this method is shown in Figure 4.14.

In the melting method, only refraction at the final glass-water spherical interface was taken into account. This is because; the refractive indices of the core, the clear region and the region just before the spherical interface are exactly the same (R.I. =1.5), as they are made of the same material. Notice, however, that in this method, we have to take into account refraction at both the front and back spherical interfaces of the micro-ball. This is because of the difference in refractive indices of the core (R.I. = 1.5), the clear region (epoxy with R.I. = 1.56) and micro-spherical balls (with R.I. ranging from 1.77 to 2.02). Hence as a consequence, the condition for optimum distance of the core from final interface, as well as the relationship between the source and image distance changes. Writing the lens maker’s formula for these two interfaces and combing them (as the
image for the first interface, will act as the source for the second interface), we can plot the relationship between the source and image distances from the final interface as shown in Figure 4.15 and 4.16. The two plots are for spherical balls made of Sapphire (R.I. =1.77) and LASF35 (R.I. =2.02).

From previous discussions, it can be inferred that the working distance (i.e. the image distance $S_i$) can also be varied by changing the length of capillary tubing; as the source and image distances are interrelated by the lens maker’s formula. It can be noticed, however, from Figures 4.15 and 4.16 that the tolerances for changing the source distance over the desired operating region are really tight. That is, the source can only be within a specific range for the focusing to work. An alternate however is to use micro-spherical ball with different refractive index. This modifies the relationship between source and image distances, as shown in Figures 4.15 and 4.16, for balls with two different refractive indices. Therefore, by optimizing the two parameters, i.e., the source
distance and the refractive index of the micro-spheres, one can vary the working distance to the desired value.

We now calculate the theoretical values of the spot size for lenses manufactured using this method. All the calculations for the spot size will be carried out under the approximations of the Gaussian beam optics. The theory of Gaussian beam propagation developed by Siegman [29] will be followed throughout the discussion ahead.

Figure 4.15 Relationship between the source and image distances, in lenses fabricated using micro-spheres of R.I. =1.77, for different values of radius of curvature.
Figure 4.16 Relationship between the source and image distances, in lenses fabricated using micro-spheres of R.I. =2.02, for different values of radius of curvature.

From the theory of Gaussian beam optics, it is known that the properties of a propagating Gaussian beam at any point are expressed by the complex radius parameter $q(z)$, given by (4.12).

\[
\frac{1}{q(z)} = \frac{1}{R(z)} - j\frac{\lambda}{\pi w^2(z)} \tag{4.12}
\]

Here $R(z)$ and $w(z)$ are the radius of curvature and the beam radius of the optical beam at a distance $z$ from the beam waist, and $\lambda$ is the wavelength of light in the medium of
propagation. Beam waist refers to the diameter of beam at the plane, where radius of curvature of the wave front is infinity. For simplicity in the calculations ahead, we assume $z$ to be zero at the waist plane. Hence, the complex q-parameter ($q_0$) at this plane is given as

$$\frac{1}{q_0} = -j\frac{\lambda}{\pi w_0^2}$$

Solving,

$$\tilde{q}_0 = j\frac{\pi w_0^2}{\lambda} = jZ_R$$

(4.13)

Here $Z_R$, as mentioned in the previous section, is the Rayleigh range and is expressed by (4.2). The complex q-parameter defines the properties, i.e., the radius of curvature of the wave front and the spot size, of a propagating Gaussian beam at any point. However, the propagation of these Gaussian beams through any optical element or through free space is described by the Ray matrices or the ABCD matrices. These ray matrices transform the complex q-parameter, for propagation through any medium, between an input and an output plane. Note that here it is assumed that the paraxial nature of the Gaussian beam is maintained on propagation through the optical system described by its ray matrices. The relationship between the complex q-parameters at the input and output planes is given as

$$\frac{\tilde{q}_2}{n_2} = \frac{A(\tilde{q}_1 / n_1) + B}{C(\tilde{q}_1 / n_1) + D}$$

(4.14)
Here $q_1$ and $q_2$ are the complex radius parameters in the input and output plane of the paraxial optical system, described by its ABCD matrices; and $n_1$ and $n_2$ are the refractive indices of the input and output medium. In the lenses fabricated using micro-spherical balls, the Gaussian beam after leaving the core of single mode fiber propagates through three different mediums and two spherical interfaces, before getting focused at a distance $S_i$ from the final interface. The exact geometry for a beam propagating through one of such lenses was shown in Figure 4.14.

It can be noticed that the beam, after leaving the core at plane 1, propagates through a number of optical elements before getting focused on plane 2. The ray matrices explaining the propagation of beam from this input plane 1 to the output plane 2 is given as

$$
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}
= 
\begin{bmatrix}
1 & (S_i/n_2) \\
0 & 1
\end{bmatrix}
\times 
\begin{bmatrix}
1 & 0 \\
(n_2 - n_3) & R
\end{bmatrix}
\times 
\begin{bmatrix}
1 & (2R/n_3) \\
0 & 1
datac
\end{bmatrix}
\times 
\begin{bmatrix}
1 & 0 \\
(n_1 - n_3) & R
\end{bmatrix}
\times 
\begin{bmatrix}
1 & (S_{o1}/n_1) \\
0 & 1
datac
\end{bmatrix}
$$

(4.15)

The five matrices in the R.H.S. of (4.15) represent the propagation of beam through the five optical elements between the input and the output plane, as shown in Figure 4.12. The distances $R$, $S_{o1}$ and $S_i$ and the refractive indices $n_1$, $n_2$ and $n_3$ in (4.15) are as shown in Figure 4.14. Hence, for spherical interfaces with different radius of curvatures, and for different values of the source and the image distances from the interface, the transformation matrix between the input and output plane can be calculated using (4.15). Thus it can be directly inferred that if the complex q-parameter at the input plane is known, the q-parameter at the output can be calculated using (4.14) and (4.15).
From Figure 4.14 it is clear that the input plane (plane 1) is the exit tip of the single-mode optical fiber. We also know that the radius of curvature of the beam at the exit tip of the fiber is infinite, i.e., the wave-fronts of the Gaussian beam are similar to those of plane waves. Hence the diameter of the beam at this plane is equal to the beam waist, and is approximately equal to the diameter of the core of single mode optical fiber. From these values of the radius of curvature and beam radius at plane 1, the complex $q$-parameter at this plane is given as

$$
\tilde{q}_1 = j \frac{\pi w_1^2}{\lambda_1}
$$

(4.16)

Here $w_1$ is the radius of the core of SMF and $\lambda_1$ is the wavelength of light in the input medium. Using this value of complex $q$-parameter ($q_1$) at the input plane, the $q$-parameter ($q_2$) at the output plane (plane 2) can be calculated using (4.14) and (4.15). The value of $q_2$ calculated will be of the form

$$
\frac{1}{\tilde{q}_2} = -j \frac{\lambda_2}{\pi w_2^2}
$$

(4.17)

where $\lambda_2$ is the wavelength of light in the output medium and $w_2$ is the radius of beam at the output plane. From the imaginary part of the complex $q_2$-parameter at the output plane, the radius (and hence the spot size) of the Gaussian beam at plane 2 can be obtained. Plots showing the variation of spot size at the output plane with the distance of source from the interface for balls of different radii of curvatures are shown in Figures 4.17 and
4.18. These two figures hold for lenses fabricated using micro-balls of refractive indices 2.02 and 1.77, respectively. The four plots in these figures are for spherical interfaces with different radii of curvatures.

4.3.2.2 Ball Tip Characterization

The characterization of beam emanating from a lensed fiber-optic probe is discussed in this section. Lens parameters such as working distance and spot size are measured and compared with the calculated theoretical values. A camera image of the ball tip fiber-optic lensed probe fabricated using the procedure discussed in the previous section is shown in Figure 4.19.

Figure 4.17 Variation of spot size with the distance of source from the interface with micro-balls of R.I. =2.02.
Figure 4.18 Variation of spot size with the distance of source from the interface with micro-balls of R.I. =1.77.

The beam exiting the lensed probes fabricated using the ball tip method was analyzed by observing the fluorescence in a solution of Rhodamine 6G (R6G) fluorescence dye into which the probe end was immersed. The image acquisition procedure is the same as used for characterizing bare and melted tip probes. Three fluorescence images of the beam emanating from the lensed probes with different source distances from the spherical interface are shown in Figure 4.20.
Figure 4.19 A camera image of the lensed probe manufactured using micro-spherical balls.
Figure 4.20 Fluorescence excited by light exiting three micro-spherical lensed probes with different source distances.
Under the geometrical optical approximations, the working distances of these lenses were calculated using the lens maker’s formula. These distances were plotted with respect to the source distances, for different radius of curvatures of the spherical interface, in Figures 4.15 and 4.16. It can also be noticed that due to different distances of source from the final interface in the three probes shown in Figure 4.20, their corresponding working distances are also different. These source and working distances are measured directly from the camera pictures of the probes and multiplied with the corresponding scaling factor. These measured values are then compared with the calculated theoretical values. The experimental values thus obtained are plotted along with the theoretical values in Figures 4.15 and 4.16. The experimental values are shown in the figures by black asterisk marks. Upon inspecting the experimental values obtained for the two cases, it can be concluded that the working distance of the lenses manufactured using micro-balls closely agree with the expected theoretical expected values.

Using a similar procedure the value of the spot sizes were also measured directly from the camera pictures of the probes and compared with the expected theoretical values plotted in Figures 4.17 and 4.18. The black asterisk marks in these figures depict the experimental values of spot size measured from the camera pictures of the probes. It can be noticed that the experimental values plotted in these figures agree closely with the expected theoretical values.

The two-way scattering profile of the ball tip lensed fiber-optic probe was also measured as a function of lateral and axial distance from the lens tip. Specifically, the focused beam exiting the lensed probe was directed onto two different vibrating scatterers
(a nylon thread and a cotton thread) and the frequency shifted backscattered radiation was collected by the same probe. The scatterers were mounted on a piezoelectric vibration stage and were vibrated at a specific frequency. The amplitude of Doppler signal at this vibration frequency resulting due to the scattering of light beam from moving scatterer was measured for different axial and lateral positions of scatterer from the probe tip. The light scattering from the threads was believed to be random; hence every three simultaneous data points along each lateral plane were averaged to account for speckle. During scattering measurements it was also assumed that the profile of the beam for every lateral plane along the axis of fiber-optic probe is Gaussian. Hence a Gaussian was fitted to the measured Doppler signal for every lateral plane along the lens axis.

A laser diode (Mitsubishi LD Model Number ML120G, wavelength ~ 658 nm, power ~ 45 mW) was used to couple radiation to the lensed probe. Two different scatterers (Nylon and Cotton thread) were separately mounted on a piezoelectric vibration stage controlled by 3-Axis Piezo-Controller (Thorlabs Model Number MDT 690). The vibration frequency of the Doppler targets was set at 1000 Hz using a function generator (HP model Number 3310A). Typical power exiting the ball lensed probe and focused on the vibrating scatterers was measured to be ~6 mW. The scattered radiation from the threads was frequency-shifted due to the Doppler effect. Part of this scattered radiation was collected simultaneously by the same probe and mixed with the incident radiation reflected from the probe tip due to Fresnel reflection. A photodetector (New Focus Model Number 2031) was used to detect the Doppler signal at this beat frequency. To remove some of the low-frequency undesired noise signals, the Doppler signal was passed through a high pass filter (Stanford Research Systems Model Number SR650)
with the cutoff frequency set at 100 Hz. The Doppler signal was acquired using a National Instruments data acquisition card (NI DAQCard-6062E) at a sampling rate of 10000 Hz. The Doppler spectrum was simultaneously displayed using a LabView program (Benchtop Spectrum Analyzer) with a frame size of 512. The amplitude of Doppler signal at the frequency of 1000 Hz was measured using the LabView program as a function of lateral and axial distance of scatterer from the probe tip. The measured Doppler signal for every lateral plane along the lens axis was averaged for speckle and a Gaussian function was fitted to the averaged data in Matlab. The two-way scattering profile thus obtained by scattering of focused radiation from the nylon and the cotton thread are shown in Figure 4.21 as a function of lateral and axial distance from the probe tip.

In the next chapter, the software tools developed to display, acquire and analyze the data collected using the laser Doppler setup (presented in Chapter 2) are discussed.
Figure 4.21 Two-way scattering profile of focused beam using a lensed probe from (a) Nylon Thread (b) Cotton Thread.
CHAPTER 5

DISPLAY, ACQUISITION AND ANALYSIS SOFTWARE

The Doppler signal, described previously, is fed to the computer through a National Instruments data acquisition card (NI DAQCard-6062E). There should, however, be a software interface between the user and the hardware to display, acquire and analyze the Doppler signals collected during the velocity measurement experiments. It should be kept in mind here that display means real-time display of the Doppler signal and its Fourier transform; acquisition means logging the Doppler data into the disk of the computer; and analysis means analysis of the Doppler data to calculate the mean Doppler frequency (and hence the velocity).

Four separate software programs were written in Matlab at different times during this work. The purpose of the first Matlab program was to simultaneously display and acquire the Doppler signal in real-time for a predetermined time-period. At the completion of acquisition, the program analyzes the acquired Doppler data by itself to generate a plot showing the variation of frequency as a function of time and also calculates the mean velocity of the moving scatterers from the acquired data. This program is no longer used because it has certain shortcomings that are discussed later.

The second program is a Graphical User Interface (GUI) tool developed in Matlab to display the signal input to the data acquisition card and its spectrum in real-time. The third program runs simultaneously with the second program and is used to acquire the
data displayed by the second program for an arbitrary length of time specified by the user. The fourth program is an offline program that analyzes the Doppler data acquired either by the GUI tool of the previous program, or by some other software interface. We now discuss all these programs in further detail.

5.1 Simultaneous Display, Acquisition and Analysis

The first Matlab program (Real-Time.m, see Appendix) was used to simultaneously display, acquire and analyze the Doppler signal for a pre-specified period of time. A snapshot of the display is shown in Figure 5.1.

It can be noticed from Figure 5.1 that this program simultaneously displays the signal fed to the computer and its spectrum in real-time. The display remains in the ‘on’ state for a pre-specified period of time and simultaneously logs the data to the computer disk. Upon completion of this time interval, the display switches to the ‘off’ state, the data acquisition stops and the program analyzes the data logged to the disk. It finally generates a plot showing the variation of Doppler frequency over this period of time and also calculates the mean velocity of the moving scatterers.

This program, however, is no longer used because of its many shortcomings. The first disadvantage with this program is that the display and acquisition are simultaneous. Hence it is not possible with this program to preview the data before acquiring them. Previewing the data before logging them in computer is necessary in order to be certain that the data being acquired are the Doppler signal from the moving scatterers. Another disadvantage is that the acquisition and display parameters such as sampling rate, frame size and acquisition time have to be specified beforehand and that the program can only run for a pre-specified period of time.
Moreover, this program lacks an interactive user interface to assist users with real-time control over the acquisition and display parameters.

5.2 Display and Acquisition GUI Tool

To remove the shortcomings of the program described in the previous section, two GUI tools were developed in Matlab. The purpose of the first GUI tool (Amit.m, see
Appendix) is to display the signal input to the computer and its Fourier transform in real-time. UI controls like push buttons and drop-down menus are also included in this program to provide users with a real-time control over the acquisition and display parameters while visualizing the data at the same time. A snapshot of this GUI tool is shown in Figure 5.2.

It can be noticed from Figure 5.2 that in addition to displaying the real-time plots for the signal and its Fourier transform, a real-time three-dimensional plot showing the variation of frequency with time is also included. The display of this three-dimensional plot can be switched between ‘classic’, ‘classic top’ or ‘cycle all’ visualizations, using the drop-down menu provided at the bottom of Figure 5.2. Another drop-down menu is provided to change the sampling rate for data acquisition and display between three different values of 40000, 20000 and 10000 samples per second. Moreover, the display can be switched between the ‘on’ and ‘off’ states (i.e., paused) as and when required, by pressing the ‘start/stop’ toggle button in the GUI. Finally by pressing the ‘Exit’ button the user can exit the program.

The only shortcoming with this GUI is that it does not have the data saving capability on its own. The data logging property in Matlab depends on the trigger parameter and buffer size specified for individual programs. These parameters can be only at certain specific values for the data logging to work; whereas for the display of Figure 5.2 to work, these parameters should be at some other contradictory values. To incorporate the data saving capability another GUI tool was developed that runs in conjunction with the GUI of Figure 5.2 (simplestartstoplogging.m, daqstopbutton.m and spectrumscope.m, see Appendix).
Figure 5.2 A snapshot of the GUI in Matlab displaying the signal and its Fourier transform in real time.
The first of these three programs is the main function; whereas, the other two are the callback routines for the main function. A snapshot of this GUI is shown in Figure 5.3.

It can be noticed that the interface displays the Fourier transform of the acquired Doppler signal in real-time. Similar to the interface of Figure 5.2, the display can be switched between the ‘on’ and ‘off’ states by pressing the ‘start/stop’ toggle button. However, the advantage with this GUI is that, by pressing the ‘start/stop file logging’ toggle button, data can be logged on the computer disk for an arbitrary period of time specified by the user. On pressing the file logging button a file path and a file name has to be specified for the acquisition to start. The data acquisition will stop on pressing the same toggle button again. This GUI tool by itself has the data saving capability, but the disadvantage is that the display is not good. Notice that the shortcomings for both the GUI tools of Figure 5.2 and Figure 5.3 are removed by running both the interfaces simultaneously. Running them simultaneously gives the advantage of better UI of the first and data saving capabilities of the second.

5.3 Data Analysis

While performing the velocity measurement experiments, real-time Doppler data can be visualized and saved to the disk by simultaneously using the UI’s of Figures 5.2 and 5.3. To calculate the mean Doppler frequency (and hence velocity) from the Doppler signal, the acquired data are analyzed using this Matlab program (Data-Analysis.m, see Appendix).

The purpose of this Matlab program is to calculate the mean Doppler frequency (and hence the mean velocity of the moving scatterers) from the acquired Doppler data. The desired data file to be analyzed is first loaded in the Matlab program.
Figure 5.3 A snapshot of the GUI in Matlab with data saving capabilities.

The experimental and acquisition parameters such as sampling rate, wavelength of light and the angle between the light and velocity vector are specified in the same program. Analysis program then converts this data file into a matrix. The size of this matrix is directly related to the sampling rate and the acquisition time of the experiment. The data matrix is then broken up into windows of equal size. An FFT routine in Matlab is run to calculate the frequency components in each window. This gives the frequency spectrum of the Doppler signal over each window. The spectra over each window are
averaged to obtain the mean Doppler spectrum of the signal. Applying the weighted mean algorithm on this mean spectrum, the mean Doppler frequency is obtained. Using this mean Doppler frequency, the velocity of the moving scatterers can be calculated from the Doppler equation (2.5). It is worth mentioning here that this same program was used to analyze the Doppler data collected for the laser Doppler device calibration experiment discussed in Chapter 3. The analysis procedure has already been discussed in greater detail in that chapter.

In the next chapter, the experimental results obtained by using the laser Doppler device in velocity measurement experiments performed on chick embryos are presented.
CHAPTER 6

VELOCITY MEASUREMENT RESULTS

The laser Doppler velocimeter described in previous chapters was used for blood velocity measurements in chick embryos. Specifically blood velocity measurements were performed on the aortic arches of normal and ligated stage 24 chick embryos. The motive was to determine the dependence of aortic arch morphogenesis on intracardiac blood flow in the left atrial ligated chick embryo. In this chapter the results from the velocity measurement experiments in chick embryos are presented.

6.1 Chick Embryo Results

The ‘fiber-optic version’ of the laser Doppler velocimeter (LDV) was used for blood velocity measurements in chick embryos. The results presented in this chapter were obtained using a bare fiber-optic probe (F-SV optical fiber from Corning Inc, New York, N.A. = 0.12, core diameter = ~6-9 micrometers). The bare probe is a single-mode optical fiber, stripped and cleaved at one end and connectorized with a FC connector at the other. This was the same probe used during device calibration experiments discussed in Chapter 3 and beam characterization experiments discussed in Chapter 4.

Velocity measurements were performed on the 3\textsuperscript{rd} and 4\textsuperscript{th} aortic arches of a number of normal and ligated chick embryos at stage 24. The bare fiber-optic probe was placed close to the blood vessel (at a distance of approx. 100 microns) and the Doppler waveform was acquired by the same procedure used for calibration experiments.
discussed in Chapter 3. The angle between the light vector and the velocity vector was approximately determined in each case from the view in the microscope. Doppler waveform from the 3rd aortic arch of a normal and ligated chick embryo at stage 24 is shown in Figure 6.1. The sampling rate in these experiments was 20000 samples per second and the acquisition time varied between 8 to 12 seconds.

The time-domain Doppler data were Fourier analyzed using the procedure discussed in section 3.1 to obtain the Doppler spectra across each window. The window size in the Fourier analysis was set at 512 samples with 50% overlap. These Doppler spectra were then averaged to obtain the mean Doppler spectrum for the two cases as shown in Figure 6.2.

Finally, the weighted mean algorithm (discussed in Section 3.1) was used to obtain the mean Doppler frequency from the Doppler spectra across each window. From the mean Doppler frequency, the blood velocity can be directly calculated using the Doppler equation (2.5). The variation in the mean Doppler frequency with time for the two cases is shown in Figure 6.3.

Mean blood velocity (mm/sec), blood flow (mm³/sec) and stroke volume (µm³/beat) were calculated for each embryo using the Doppler data and the subsequent Fourier analysis. From the data analysis it was determined that the 3rd and 4th aortic arch blood flow was decreased in ligated embryos when compared to normal embryos. It was hence concluded that partial left atrial ligation prior to cardiac septation redistributes intracardiac blood flow and produces left ventricular hypoplasia in the chick [33]. It was also hypothesized that redistributed intracardiac blood flow adversely alters aortic arch development [33].
Figure 6.1 Doppler signal from the 3\textsuperscript{rd} aortic arch of (a) Normal (b) Ligated chick embryo at stage 24.
Figure 6.2 Mean Doppler spectrum obtained by Fourier analysis of Doppler signal from
(a) Normal (b) Ligated chick embryo at stage 24.
Figure 6.3 Variation of mean Doppler frequency with time obtained from the Doppler data for (a) Normal (b) Ligated chick embryo at stage 24.
These alterations in intracardiac blood flow disrupt both early cardiac morphogenesis and aortic arch selection. These finding may explain the association of aortic arch abnormalities in congenital cardiac defects with reduced left heart blood flow [33].

Velocity measurement experiments were also performed using the ball-lensed fibers. Due to coupling losses from the bare optical fiber to the lensed probe, the output powers of ~1-2 mW with the HeNe laser (wavelength ~ 633 nm) was not sufficient to get any back-scattered Doppler signal. A laser diode (Mitsubishi LD Model Number ML120G, wavelength ~ 658 nm) with output power of ~45 mW was then used to couple radiation to the lensed probe. Though the output power was sufficient to get efficient back-scattering, it was found that the coherence length of this laser was < 1.0 mm. Due to the small coherence length of this laser diode, there was no interference between the reference and back-scattered light to obtain any Doppler signal.
CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

Theory and applications of the laser Doppler technique for measuring blood velocities in minute blood vessels is presented in this work. A prototype laser Doppler instrument was built, characterized, validated and used for velocity measurements in microcirculatory blood vessels. The theory of working of these devices presented helped in understanding the principle of working of these devices. The fabrication and characterization of the bare and lensed fiber-optic probes gave an insight into the importance of beam profiles emanating from these probes. The results from the discussions in previous chapters are summarized here briefly. Furthermore this chapter presents the conclusions drawn from these results and some thoughts on future work.

The numerous advantages and applications of the laser Doppler technique presented in Chapter 1, underlines the importance of this technique in biomedical diagnostics. The primary use of this technique for velocity measurement applications was compared with several other existing techniques. The comparison provided an overview of the pros and cons of this technique over these other techniques. The theory behind the working of two types of laser Doppler devices, namely the single-beam and the dual-beam laser Doppler velocimeters, is also presented and compared. The comparison of
these two, physically different but fundamentally similar, techniques provided further insight into the understanding of these devices.

The exact geometry and working of three prototype laser Doppler velocimeters built in our lab was discussed in Chapter 2. It was concluded that the fiber-optic version enhanced the signal-to-noise ratio of the Doppler device due to reduced effect of air turbulence on the interferometer arms. Moreover, it was found that blocking the separate reference arm also improved the SNR by removing the effect of signals that were not common to the two interferometer arms.

The laser Doppler velocimeter (LDV) validation procedure and results were presented in Chapter 3. The validation experiment was performed with a white wheel rotating at a predetermined linear velocity. By comparing this known value of velocity with the value measured using the LDV, it was found that this technique shows a good degree of accuracy (±1%) and repeatability (±2.2%). The laser Doppler device was then characterized using the procedures discussed in Chapter 3. It was found that the SNR deteriorates while measuring velocities in chick embryos due to reduced strength of Doppler signal from the moving scatterers (RBC’s). From the calibration data collected during the validation experiment, a drift of 0.16% in velocity measurements over a period of one hour was determined. It was concluded that the device shows good stability while working for longer periods.

It is known from the Doppler equation that the measured velocity of moving targets is inversely proportional to the cosine of angle between the light and velocity vectors. A novel technique to measure these angles is devised in this work as discussed in
Chapter 3. It was found that the angles can be measured accurately to within $\pm 2.41\%$ of its actual value using this novel technique.

In Chapter 4, the fiber-optic probe design and fabrication procedure, and the characterization of the beam exiting these probes was discussed. Three different probes, namely the bare, the melted tip and the ball lensed fiber-optic probes, were fabricated for use in this work. The beam exiting these probes was characterized using the theory of geometrical and Gaussian beam optics. It was found that the focusing of radiation onto blood vessels using the bare fiber-optic probe was weak because the beam emanating from these probes diverged rapidly due to the diffraction effect. Also, the collection of the scattered frequency-shifted Doppler signals with these probes was limited by the small core size of the single mode fiber. In order to increase the strength of light coupled into the blood vessels, and simultaneously increase the collection of the scattered signals, two different spherical ended fiber-optic probes were fabricated. The first one, namely the melted tip probe, was fabricated by directly melting the tip of a bare single-mode optical fiber. It was found that the light exiting a melted tip fiber-optic probe remains collimated for a longer distance, as well as the collection efficiency of scattered signals with these probes is higher than corresponding bare probes.

The second one, namely the ball lensed fiber-optic probe, was fabricated by gluing micro-spherical balls with the help of epoxy at a distance from the tip of a bare optical fiber using a capillary tubing. The light exiting these probes were focused by these micro-spherical balls down to a spot size of $\sim 2-10$ microns at a working distance of $\sim 0.20-0.80$ mm. Also for the specific application intended, spot size and working distance
of these probes can be varied by changing the length of capillary tubing or by using micro-spheres of different refractive indices.

To improve the visualization as well as ease the acquisition of Doppler data from the data acquisition card, a graphical user interface (GUI) tool was developed in Matlab. The tool facilitated real-time display of the Doppler signal as well as its frequency spectrum. A three-dimensional plot displaying the change in frequency as a function of time was also included in the GUI. It is also possible with this GUI to simultaneously display and log data into the computer disk at a desired sampling rate. Another Matlab program was written to analyze the Doppler data collected by using the Matlab GUI and calculate the mean velocity of the moving scatterers.

Finally, the laser Doppler device with the bare fiber-optic probes was used to measure blood velocities in vessels of chick embryos. The Doppler signals collected from this probe were analyzed to calculate the mean velocity of the moving red blood cells. The measurements were performed on the 3rd and 4th aortic arches of normal and ligated stage 24 chick embryos. Using the time-domain Doppler data, the mean blood velocity, blood flow and the stroke volume was determined for the two cases. It was found that the blood flow was decreased in ligated embryos when compared to normal embryos. From these findings it was concluded that partial left atrial ligation prior to cardiac septation redistributes intracardiac blood flow and produces left ventricular hypoplasia in the chick. It was also hypothesized that redistributed intracardiac blood flow adversely alters aortic arch development [32].
7.2 Future Work

With regard to experimental applications discussed in this work, the laser Doppler velocimeter was primarily used for velocity measurements in blood vessels of chick embryos. Chick embryos have been a favorite vertebrate model for developmental physiologists because of reasons discussed in Chapter 1. However, recently zebra fish has become a popular vertebrate model with which to study cardiac development [34]. Much of zebra fish cardiac development is relevant to mammalian cardiac development, even though the fish heart is two chambered. The significance of the zebra fish model in understanding cardiac organogenesis and vascular development is discussed in further detail in references [35] and [36]. Some of the major limitations with measuring blood velocities in embryonic zebra fish are small size of blood vessels, smaller number of scatterers and lower blood flow velocities. Due to small size of blood vessels, it is difficult to couple enough light on them to get sufficient scattering from the limited number of RBC’s. However, with the lensed fiber-optic probes it is possible to get small enough spot-sizes at the desired working distances which make them suitable for performing measurements in zebra fish. It will be interesting to study the applications of laser Doppler velocimetry with lensed probes on the minute blood vessels of zebra fish. This could yield further insight into the understanding of functional characteristics of the embryonic cardiovascular system, complementing the information already derived from chick embryos.
clc;
clear all;
AI = anologinput('winsound',0);
chan = addchannel(AI,1);
% AI = analoginput('nidaq',1);
% chan = addchannel(AI,1);
duration = 10;
set(AI,'SampleRate',22050);
ActualRate = get(AI,'SampleRate');
set(AI,'SamplesPerTrigger',duration*ActualRate);
set(AI,'TriggerType','Manual');
blocksize = get(AI,'SamplesPerTrigger');
preview = fix(duration*ActualRate/430);
set(AI,'LoggingMode','Disk&Memory')
set(AI,'LogFileName','data.daq')
Fs = ActualRate;
C = 512;
SampleRate = ActualRate;
H = ((C/2)+1);
B = fix((duration*ActualRate)/C);
delT = 2/Fs;
TotalT1 = (((length(A)/SampleRate)-(delT/2));
TotalT = (((C*B*delT)-(C*delT))/2);
T = ((C*delT)/2);
T1 = (T-(delT/2));
delF = 1/T;
Fo = 1/(delT);
f = [0:delF:Fo];
t2 = [0:delT/2:T1];
t = [0:T:TotalT];
t1 = [0:delT/2:TotalT1]
delT = 2/Fs;
TotalT = (((C*B*delT)-(C*delT)));
T = ((C*delT));
delF = 2/T;
Fo = 1/(delT);
\[ f = [0:delF:Fo]; \]
\[ t = [0:T:TotalT]; \]

```
f = [0:delF:Fo];
t = [0:T:TotalT];
start(AI)
trigger(AI)
figure(1)
subplot(211)
set(gcf,'doublebuffer','on')
P = plot(zeros(preview,1));
set(P,'xdata',t2');
grid on
xlabel('Time (Secs)')
ylabel('Signal (Volts)')
subplot(212)
set(gcf,'doublebuffer','on')
Q = plot(zeros(preview/2,1));
set(Q,'xdata',f');
grid on
xlabel('Frequency (Hz)')
ylabel('Amplitude of the frequency components')
while AI.SamplesAcquired < preview
end
while AI.SamplesAcquired < duration*ActualRate
    data = peekdata(AI,preview);
    set(P,'ydata',data);
drawnow;
data1 = fft(data);
data2 = abs(data1(1:H,1));
set(Q,'ydata',data2);
drawnow;
end
while AI.SamplesAcquired < preview
end
while AI.SamplesAcquired < duration*ActualRate
    data = peekdata(AI,preview);
data1 = abs(fft(data))
set(Q,'ydata',data1)
drawnow
end
A = getdata(AI);
C = 512;
Samplerate = ActualRate;
H = ((C/2)+1);
B = fix((length(A))/C);
k = 1;
for i=1:B
    for j=1:C
```
M(i,j)=A(k);
k=k+1;
end
end
for n=1:B
    D(n,:) = fft(M(n,:));
    E(n,:) = abs(D(n,1:H));
    F(n,:) = fftshift(E(n,:));
end
m = 1;
for i=1:B
    for j=1:H
        J(m)=E(i,j);
        m = m+1;
    end
end
delT = 2/Samplerate;
TotalT1 = ((length(A)/Samplerate)-(delT/2));
TotalT = (((C*B*delT)-(C*delT))/2);
T = ((C*delT)/2);
delF = 1/T;
Fo = 1/(delT);
f = [0:delF:Fo];
t = [0:T:TotalT];
t1 = [0:delT/2:TotalT1];
delT = 2/Samplerate;
TotalT = (C*B*delT)-(C*delT);
T = C*delT;
delF = 2/T;
Fo = 1/(delT);
f = [0:delF:Fo];
t = [0:T:TotalT];
figure(2);
plot(t1',A);
xlabel('Time (Sec)');
ylabel('Amplitude (Volts)');
figure(3);
plot(f,E(3,:));
xlabel('Frequency(Hz)');
ylabel('Amplitude of the frequency components');
figure(5);
waterfall(f,t,E);
xlabel('Frequency(Hz)');
ylabel('Time(Sec)');
zlabel('Amplitude of the frequency components');
figure(3);
pcolor(t,f,E');
shading flat;
for i=1:B
    X(i) = (sum(E(i,:).*[1:H]))/sum(E(i,:));
    Favg(i) = (X(i)*delF);
    i = i+1;
end
figure(4);
plot(t,Favg);
xlabel('Time(Sec)');
ylabel('Frequency(Hz)');
f = mean(Favg);
wl = 633e-9;
theta = 75;
ang = ((pi*theta)/(180));
vel = ((wl*f)/(2*cos(ang)));
disp(f);
disp(vel);
figure(2)
xfft = abs(fft(data));
mag = xfft;
f = (0:length(mag)-1)*Fs/(blocksize);
mag = 20*log10(xfft);
mag = mag(1:blocksize/2);
f = f(:,);
i = length(mag);
k = 1;
while i>=k
    mag1[k] = mag[i];
    i = i-1;
    k = k+1;
end
mag2 = [mag1 mag]
figure(3);
plot(f);
grid on
ylabel('Magnitude')
xlabel('Frequency (Hz)')
title('Frequency Components')
figure(3)
plot(data)
figure(4)
[data5,time] = daqread('data.daq');
plot(time,data5)
grid on
title('All Acquired Data')
xlabel('Samples')
ylabel('Signal level (volts)')
[ymax,maxindex] = max(mag);
disp(['Maximum occurred at ', num2str(maxindex), ' Hz'])
delete(AI)
clear AI

Amit.m

function varargout = Amit(varargin)
    % Amit M-file for Amit.fig
    % Amit, by itself, creates a new Amit or raises the existing
    % singleton*. 
    % H = Amit returns the handle to a new Amit or the handle to
    % the existing singleton*. 
    % Amit('Property','Value','...) creates a new Amit using the
    % given property value pairs. Unrecognized properties are passed via
    % varargin to Amit_OpeningFcn. This calling syntax produces a
    % warning when there is an existing singleton*. 
    % Amit('CALLBACK') and Amit('CALLBACK',hObject,...) call the
    % local function named CALLBACK in Amit.M with the given input
    % arguments. 
    % Edit the above text to modify the response to help Amit

    % Begin initialization code - DO NOT EDIT
    gui_Singleton = 1;
    gui_State = struct('gui_Name', mfilename, ...
                      'gui_Singleton', gui_Singleton, ...
                      'gui_OpeningFcn', @Amit_OpeningFcn, ...
                      'gui_OutputFcn', @Amit_OutputFcn, ...
                      'gui_LayoutFcn', [], ...
                      'gui_Callback', []);
    if nargin & isstr(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargin
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
    % End initialization code - DO NOT EDIT
% --- Executes just before Amit is made visible.
function Amit_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
% command line (see VARARGIN)
% Choose default command line output for Amit

handles.output = hObject;

++++ I N P U T S ++++
% Change these settings to select a different source.
handles.adaptor = 'winsound';
handles.id = 0;
% handles.adaptor = 'nidaq';
% handles.id = 1;
handles.chan = 1;
handles.samplesPerTrigger = 1024;
handles.sampleRate = 44100;
handles.numTraces = 10; % number of traces to show in the waterfall.
handles.cycleTime = .9; % Proportional to the amount of time spent per
% visualization on CycleAll setting.
---- I N P U T S ----

++++ S E T U P   T H E   F I G U R E ++++
set(handles.figure1,'Color',get(handles.frame8,'BackgroundColor'));
axes(handles.axes1);
handles.hLine1 = plot(zeros(1,handles.samplesPerTrigger)');
set(handles.hLine1,'Color', [0 .1 0.5]);
set(handles.axes1,'Color', [235/255 255/255 235/255])
set(handles.axes1,'XGrid','on','YGrid','on')
t=title('Time Domain Signal','Color', [0.05 0.05 0.25], 'FontWeight','Bold','FontSize',9);
xlabel('Time (s)','FontSize',8);
ylabel('Voltage (V)','FontSize',8);
axes(handles.axes2);
handles.hLine2 = plot(zeros(1,handles.samplesPerTrigger/2));
set(handles.hLine2,'Color', [.1 0.5 .1]);
set(handles.axes2,'Color',[235/255 255/255 255 255/255])
set(handles.axes2,'XGrid','on','YGrid','on')
t=title('Frequency Domain Signal','Color',[.05 0.25 .05],'
FontWeight','Bold','FontSize',9);
xlabel('Frequency (Hz)','FontSize',8);
ylabel('Magnitude','FontSize',8);

axes(handles.axes3);
set(handles.axes3,'View',[103 10]);
set(handles.axes3,'Color',[234/255 234/255 255/255]);
grid(handles.axes3,'on');
h = get(handles.axes3,'title');
set(h,'string','Waterfall Plot','FontWeight','Bold','Color',[.25 .05 .05],'FontSize',9);
h = get(handles.axes3,'ylabel');
set(h,'string','Frequency (Hz)','FontSize',8);
h = get(handles.axes3,'zlabel');
set(h,'string','Magnitude','FontSize',8);

set(hObject,'RendererMode','Manual')  % If you don't do this, the surface plot
set(hObject,'Renderer','OpenGL')      % will draw VERY slowly.

% set(handles.tSource,'String',sprintf('%s:%d',handles.adaptor,handles.id));
% set(handles.tChannel,'String',num2str(handles.chan));
set(handles.poSampleRate,'String','[{'44100'},
{'22000'},
{'10000'}]);
set(handles.poPlotType,'String','[{'CycleAll'},
{'Classic'},
{'Classic(Top)'}]);

%---- S E T U P   T H E   F I G U R E ----

ai=localSetupAI(handles);
handles.ai = ai;

% Update handles structure
guidata(hObject, handles);

localStartAI(ai);

% daqstopbutton(handles.figure1,ai);
% UIWAIT makes Amit wait for user response (see UIRESUME)
% uiwait(handles.figure1);
function localStartAI(ai)

++++ S T A R T  A I ++++
start(ai);
trigger(ai);

---- S T A R T  A I ----

function localStopAI(ai)

++++ S T O P  A I ++++
stop(ai);
delete(ai);

---- S T O P  A I ----

function ai=localSetupAI(handles)

++++ S E T U P   T H E   A N A L O G   I N P U T ++++

% Object Configuration.
ai = analoginput(handles.adaptor, handles.id);
addchannel(ai, handles.chan);

% Configure the callback to update the display.
set(ai, 'TimerFcn', @localfftShowData);

% Configure the analog input object.
set(ai, 'SampleRate', handles.sampleRate);

% Configure the analog input object to trigger manually twice.
% We do this because we are using peekdata to acquire the data in
% a timer callback function.
% The first trigger will fill the buffer with handles.samplesPerTrigger
% number of samples. We'll know we have enough samples to start
% processing data when the analog input object's SamplesAvailable property
% is equal to handles.samplesPerTrigger.
The analog input object will then wait for another manual trigger, and while it is waiting the object will still be in its running state, which means the timer event will run. To keep the object in the running state, we need only never manually trigger this second trigger. Had we set the TriggerRepeat to 0, the analog input object would stop after the first trigger and the timer functions would stop running.

```matlab
set(ai, 'SamplesPerTrigger', handles.samplesPerTrigger);
set(ai, 'TriggerRepeat', 1);
set(ai, 'TriggerType', 'manual');
```

% Initialize callback parameters. The TimerAction is initialized after figure has been created.
set(ai, 'TimerPeriod', 0.01);
set(ai, 'BufferingConfig',[handles.samplesPerTrigger*2,20]);

% Initialize time and frequency plots with lines of y=0
```matlab
d=zeros(1,handles.samplesPerTrigger);
time = 1:handles.samplesPerTrigger;
f=1:handles.samplesPerTrigger/2;
mag=zeros(1,handles.samplesPerTrigger/2);
```%

% Store state information in the analog input objects UserData area.
```matlab
data.storedFFTsIndex = 1;
data.plotSurf = 0;
data.ai = ai;
data.getdata = [d time];
data.daqfft = [f mag];
data.handle = [];
data.figureHandles = handles;
data.view = [103 10];
data.rotateStep = 4;
data.counter = 0;
```

% Set the object's UserData to data.
```matlab
set(data.ai, 'UserData', data);
```%

---- SETUP THE ANALOG INPUT ----
```matlab
```
% --- Outputs from this function are returned to the command line.
function varargout = Amit_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pbExit.
function pbExit_Callback(hObject, eventdata, handles)
% hObject    handle to pbExit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
localStopAI(handles.ai);
closereq;

% --- Executes during object creation, after setting all properties.
function poSampleRate_CreateFcn(hObject, eventdata, handles)
% hObject    handle to poSampleRate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
  set(hObject,'BackgroundColor','white');
else
  set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on selection change in poSampleRate.
function poSampleRate_Callback(hObject, eventdata, handles)
% hObject    handle to poSampleRate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns poSampleRate contents as cell array
% contents{get(hObject,'Value')} returns selected item from poSampleRate

% Hints: contents = get(hObject,'String') returns poSampleRate contents as cell array
% contents{get(hObject,'Value')} returns selected item from poSampleRate
% First, stop and delete the current analog input object
localStopAI(handles.ai);

% Extract the new samlerate.
v=get(handles.poSampleRate,'Value');
s=get(handles.poSampleRate,'String');
handles.sampleRate = str2num(s{v});

% Create a new analog input with the new sample rate.
handles.ai = localSetupAI(handles);

% Update handles structure
guidata(hObject, handles);

% Restart the analog input
localStartAI(handles.ai);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%
---- C H A N G E   T H E   S A M P L E   R A T E  ----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%
%
***********************************************************************
% Calculate the fft of the data.  (Copied from demoai_fft.m)
function [f, mag] = localDaqfft(data,Fs,blockSize)
%
***********************************************************************
% Calculate the fft of the data.
xFFT = fft(data);
xfft = abs(xFFT);

% Avoid taking the log of 0.
% index = find(xfft == 0);
% xfft(index) = 1e-17;
% mag = 20*log10(xfft);
mag = xfft;
mag = mag(1:blockSize/2);

f = (0:length(mag)-1)*Fs/blockSize;
f = f(:,);

% % 
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (get(obj,'SamplesAvailable') >= obj.SamplesPerTrigger)

    % Get the handles.
    data = obj.UserData;

    handles = data.figureHandles;

    % Execute a peekdata.
    x = peekdata(obj, obj.SamplesPerTrigger);

    % FFT calculation.
    Fs = obj.SampleRate;
    blockSize = obj.SamplesPerTrigger;
    [f,mag] = localDaqfft(x,Fs,blockSize);

    % Dynamically modify Analog axis as we go.
    maxX=max(x);
    minX=min(x);
    yax1=get(handles.axes1,'YLim');
    yax1(1)=minX - .0001; % need to subtract a value to make sure yax(1) never equals % yax(2)
    yax1(2)=maxX + .0001;
    set(handles.axes1,'YLim',yax1)

    % Dynamically modify Frequency axis as we go.
    maxF=max(f);
    minF=min(f);
    xax=get(handles.axes2,'XLim');
    xax(1)=minF;
    xax(2)=maxF;
    set(handles.axes2,'XLim',xax)

    % Dynamically modify Magnitude axis as we go.
    maxM=max(mag);
    minM=min(mag);
    yax2=get(handles.axes2,'YLim');
    yax2(1)=minM - .0001;
    yax2(2)=maxM + .0001;
    set(handles.axes2,'YLim',yax2)

    % Update the line plots.
set(handles.hLine1, 'XData', [0:(obj.SamplesPerTrigger-1)]/obj.SampleRate, 'YData', x(:,1));
set(handles.hLine2, 'XData', f(:,1), 'YData', mag(:,1));

% % Find the frequency at which the max signal strength is at.
% [ymax,maxindex] = max(mag);
% set(handles.tFreq,'String',sprintf('%4.1d Hz',f(maxindex)));

% Store the current FFT into the array of FFTs used for the waterfall.
  data.storedFFTs(data.storedFFTsIndex,:) = mag';

% This circular shift is used so that when we display the 3D plot, the
% newest FFT will appear in 'front' and the oldest in 'back'.
% To understand this, note how the plotting routines are using this fftOrder
% array to reorder the FFTs stored in data.storedFFTs and also note
% how data.storedFFTsIndex is used to store FFTs in data.storedFFTs.

fftOrder = 1:handles.numTraces;
fftOrder = circshift(fftOrder,[-data.storedFFTsIndex 0]);

  data.storedFFTsIndex = data.storedFFTsIndex + 1;
  if (data.storedFFTsIndex > handles.numTraces)
    data.storedFFTsIndex = 1;
  end

  data.plotSurf = 1; % Indicates a full history is stored.
  end

% Update the surface plot if we have a full history.
  if (data.plotSurf)
    cla(handles.axes3);
    v=get(handles.poPlotType,'Value');
    s=get(handles.poPlotType,'String');
    switch s{v}
      case 'Classic'
        data=view = [103 30];
        data=localClassic(handles,data,f,fftOrder);
      case 'Classic(Top)'
        data.view = [90 -90];
        data=localClassic(handles,data,f,fftOrder);
      case 'Mosaic'
        data.view = [90 -90];
        data=localMosaic(handles,data,f,fftOrder);
      case 'Waterfall'
        data.view = [103 30];
        data=localWaterfall(handles,data,f,fftOrder,yax2);
      case 'Rotate'
% data=localRotate(handles,data,f,fftOrder);

% case 'CycleAll'
% data=localCycleAll(handles,data,f,fftOrder);
% end
%

end

set(data.ai, 'UserData', data);
drawnow;
end


+++ V I S U A L I Z A T I O N ++++

function data=localClassic(handles,data,f,fftOrder)
[X,Y] = meshgrid(1:handles.numTraces,f(1:end));
surf(X,Y,data.storedFFTs(fftOrder,:),'parent',handles.axes3);
set(handles.axes3,'XLim',[1 handles.numTraces],'YLim',[0 f(end)])
shading(handles.axes3,'interp');
set(handles.axes3,'View',data.view)
end

% function data=localMosaic(handles,data,f,fftOrder)
% [X,Y] = meshgrid(1:handles.numTraces,f(1:10:end));
% surf(X,Y,data.storedFFTs(fftOrder,(1:10:end))','parent',handles.axes3);
% set(handles.axes3,'XLim',[1 handles.numTraces],'YLim',[0 f(end)])
% set(handles.axes3,'View',data.view)
% end

% function data=localWaterfall(handles,data,f,fftOrder,yax2)
% [X,Y] = meshgrid(1:handles.numTraces,f(1:end));
% p=plot3(X,Y,data.storedFFTs(fftOrder,:),'parent',handles.axes3);
% % rotate the color map of the lines in the plot3
% map = linspace(0,1,handles.numTraces);
% map2 = linspace(1,0,handles.numTraces);
% rotatemap = map(fftOrder);
% rotatemap2 = map2(fftOrder);
% for k=1:handles.numTraces;
% set(p(k),'Color',[rotatemap(k) .1 rotatemap2(k)]);
% end
% set(handles.axes3,'XLim',[1 handles.numTraces],'YLim',[0 f(end)])
% shading(handles.axes3,'interp');
% set(handles.axes3,'View',data.view)
% end

% function data=localRotate(handles,data,f,fftOrder)
% [X,Y] = meshgrid(1:handles.numTraces,f(1:8:end));
% surf(X,Y,data.storedFFTs(fftOrder,(1:8:end)),'parent',handles.axes3);
% set(handles.axes3,'XLim',[1 handles.numTraces],'YLim',[0 f(end)])
% set(handles.axes3,'View',data.view)
%
% % Rotate the view point.
% data.view(1) = 90;
% if data.view(2) >= 90-data.rotateStep
%     data.rotateStep = -4;
% elseif data.view(2) <= -90-data.rotateStep
%     data.rotateStep = 4;
% end
% data.view(2) = data.view(2)+data.rotateStep;

function data=localCycleAll(handles,data,f,fftOrder)
    data.counter = data.counter + get(data.ai,'TimerPeriod');
    if data.counter > 2*handles.cycleTime
        data.counter = 0;
    elseif data.counter > 4*handles.cycleTime
        data=localRotate(handles,data,f,fftOrder);
    elseif data.counter > 3*handles.cycleTime
        data.view = [103 30];
        data=localWaterfall(handles,data,f,fftOrder);
    elseif data.counter > 2*handles.cycleTime
        data.view = [90 -90];
        data=localMosaic(handles,data,f,fftOrder);
    elseif data.counter > 1*handles.cycleTime
        data.view = [90 -90];
        data=localClassic(handles,data,f,fftOrder);
    else
        data.view = [103 30];
        data=localClassic(handles,data,f,fftOrder);
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
--- V I S U A L I Z A T I O N ---
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes during object creation, after setting all properties.
function poPlotType_CreateFcn(hObject, eventdata, handles)
% hObject    handle to poPlotType (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on selection change in poPlotType.
function poPlotType_Callback(hObject, eventdata, handles)
    % hObject    handle to poPlotType (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % Hints: contents = get(hObject,'String') returns poPlotType contents as cell array
    %          contents{get(hObject,'Value')} returns selected item from poPlotType

% --- Executes on button press in StartStop.
function StartStop_Callback(hObject, eventdata, handles)
    % hObject    handle to StartStop (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % Hint: get(hObject,'Value') returns toggle state of StartStop

val = get(hObject,'Value');
if val==0
    set(hObject,'String','Stop');
    if all(isvalid(handles.ai))
        start(handles.ai);
        trigger(handles.ai);
    end;
else
    set(hObject,'String','Start');
    if all(isvalid(handles.ai))
        stop(handles.ai);
    end;
end;
function simplestartstoplogging

% Create analoginput object
ai = analoginput('winsound');
% ai = analoginput('nidaq',1);
chan = addchannel(ai,1);

% Configure analoginput object
Fs = 44100;  % Sample Rate
set(ai,'SampleRate',Fs);
set(ai,'SamplesPerTrigger',4096);

% Initialize scope
hFig = figure;
set(hFig,'Name','AmitSave');
Nfft = ai.SamplesPerTrigger/2;  % FFT Length.
hSpectrumScope = spectrumscope(Fs,Nfft);  % Initialize spectrum scope

% Configure acquisition to update scope
set(ai,'TriggerRepeat',inf)
set(ai,'TriggerFcn',{@updateplot,hSpectrumScope,Nfft});

% Put a stop button on the scope
daqstopbutton(hFig,ai);

% Put a file logging start/stop button on the scope
localLoggingStopButton(hFig,ai);

function updateplot(ai,event,hSpectrumScope,Nfft)
d = peekdata(ai,Nfft);
spectrumscope(hSpectrumScope,d);

function localLoggingStopButton(hFig,ai);
% Add a button for starting and stopping file logging
startstring = 'Start File Logging';
stopstring = 'Stop File Logging';

hButton = uicontrol(hFig,'style','togglebutton', ...
    'String',startstring, ...
    'Position',[90 20 100 20], ...
    'Callback',{@LoggingStopButtonCallback,ai});


% Store the strings
setappdata(hButton,'StartString',startstring);
setappdata(hButton,'StopString',stopstring);

function LoggingStopButtonCallback(hButton,ev,ai);
% Callback for Logging Stop Button

val = get(hButton,'Value');

if val % Start logging
    [filename,pathname] = uiputfile({'*.daq','Data Acquisition Log Files (*.daq)'}, ...  
        'Log data as:');

    if filename == 0  % If user cancelled
        set(hButton,val,0);
        return
    end;

    % Stop the acquisition
    stop(ai);

    % Configure logging
    set(ai,'LoggingMode','Disk&Memory');
    set(ai,'LogFileName', [pathname filename]);

    % Restart the acquisition
    start(ai);

    % Update string
    stopstring = getappdata(hButton,'StopString');
    set(hButton,'String',stopstring);

else    % Stop logging
    % Stop the acquisition
    stop(ai);

    % Configure logging
    set(ai,'LoggingMode','Memory');

    % Restart the acquisition
    start(ai);

    % Update string
    startstring = getappdata(hButton,'StartString');
    set(hButton,'String',startstring);
end;
function varargout = daqstopbutton(fig,obj,varargin);

%DAQSTOPBUTTON        Add a stop/start button to a data acquisition application
%
% DAQSTOPBUTTON(FIG,OBJ) adds a start/stop button to figure FIG. This
% button can be used to start and stop data acquisition object OBJ.
% DAQSTOPBUTTON will also delete OBJ when FIG is closed (i.e., it sets
% FIG's CloseRequestFcn to delete the object)
%
% DAQSTOPBUTTON(FIG,OBJ,'P1','V1','P2','V2', ...) specifies Property-Value
% pairs for configuring properties of the start/stop button. Any valid
% property of a togglebutton can be specified.
%
% HBUTTON = DAQSTOPBUTTON(...) returns a handle to the start/stop button
%
% Example:
%   fh = figure;                         % Create a figure
%   ai = analoginput('winsound');        % Create an input object
%   addchannel(ai,1);                    % Add a channel
%   set(ai,'TriggerRepeat',inf);         % Configure to run infinitely
%   set(ai,'TimerFcn','plot(peekdata(ai,500))'); % Each timer event will plot recent data
%   hButton = daqstopbutton(fh,ai);      % Add the stopbutton

%Error checking
msg = nargchk(2,inf,nargin);
error(msg)

if ~all(isvalid(obj))
    error('Second input argument must be valid daq object')
eend;

% Create the button
hButton = uicontrol(fig,'style','togglebutton',varargin{:});

%Check current state of the object
val = strcmp(obj.Running,'On');

if val==1     %Already running
    set(hButton,'String','Stop');
    set(hButton,'Value',1)
else
    set(hButton,'String','Start');
    set(hButton,'Value',0)
end;

set(hButton,'Callback',{@localStartStopObject,obj})

% Configure the CloseRequestFcn of the figure
setappdata(fig,'DaqStopButtonObject',obj)
cr = get(fig,'CloseRequestFcn');
cr = ['obj=getappdata(gcf,''DaqStopButtonObject'');stop(obj);delete(obj);' cr];
set(fig,'CloseRequestFcn',cr);

% Return a handle to the button
if nargout
    varargout{1} = hButton;
end;

function localStartStopObject(hButton,action,obj)
% Callback for the start/stop button
val = get(hButton,'Value');
if val==1 %Pushed in
    set(hButton,'String','Stop');
    if all(isvalid(obj))
        start(obj);
        % trigger(obj);
    end;
else
    set(hButton,'String','Start');
    if all(isvalid(obj))
        stop(obj)
    end;
end;

spectrumscope.m

function varargout = spectrumscope(varargin)
% SPECTRUMSCOPE Compute and display a real-time FFT
%
% STEP 1: Initialize the scope
% SPECTRUMSCOPE(FS,NFFT) initializes a spectrum scope in the current axes.
% This spectrum scope will compute and displays the NFFT-point FFT of a vector
% signal with sample rate FS Hz.
%
% STEP 2: Update the scope
% SPECTRUMSCOPE(S) updates the spectrum scope in the current axes with the
% FFT of vector S. The scope should first be initialized as above with
% sample rate and FFT length.
%
% SPECTRUMSCOPE(FS,NFFT,NTRACES) initializes a spectrum scope in the
% current axes with NTRACES traces. A trace is a single line on the scope;
% typically one will display one trace per channel of data.
% SPECTRUMSCOPE(S) must specify a matrix S with shorter dimension length =
% NTRACES. SPECTRUMSCOPE computes the FFT along the longer dimension,
% assuming the shorter dimension corresponds to traces.
% SPECTRUMSCOPE(HAX, ...) defines the scope in specified axes HAX instead of
% GCA. % i.e., SPECTRUMSCOPE(HAX,Fs,NFFT) initializes axes HAX as a spectrum
% scope, and
% SPECTRUMSCOPE(HAX,S) updates axes HAX with vector S.
% HAX = SPECTRUMSCOPE(...) returns a handle to the axes initialized by the
% spectrum scope. This is useful if you allow SPECTRUMSCOPE to create an
% axes for you, and want to be able to easily reference the axes for
% updates. The lines created by SPECTRUMSCOPE all have the tag
% 'SpectrumScope'
%
% Example
%    HAX = SPECTRUMSCOPE(...);
%    HLINE = findobj(HAX,'Tag','SpectrumScope');
%
% % Example
%    Fs = 1024;
%    Nfft = 2048;
%    t = (0:1:Nfft-1)'/Fs;
%    fo = 100:5:300;       % Range of fundamental frequencies
%    s1 = sin(2*pi*t*fo);
%
%    spectrumscope(Fs,Nfft);
%
%    for ii = 1:length(fo)
%        spectrumscope(s1(:,ii));
%        drawnow;pause(.01);
%    end;

%% Parse input arguments
% Decision tree:
% + Initialize or update?
%   o If update -> OK
%   o If initialize -> Axes specified, or use GCA?
error(nargchk(1,4,nargin))

%% Initialize or update?
% If first or second input argument is not a scalar, it must be data - i.e. we are
% updating

if prod(size(varargin{1})) > 1 | prod(size(varargin{2})) > 1 % Update
  action = 'update';

  if nargin==1                % Use current axes
    hAxes = gca;
    data = varargin{1};
  else
    hAxes = varargin{1};    % Axes was specified
    data = varargin{2};
  end;

  % If the user has not initialized this scope, do it for them
  parms = getappdata(hAxes,'SpectrumScopeParameters');

  % Ensure that scope has been initialized
  if isempty(parms)
    % Use default values
    Fs = 1;
    data = rowmajor(data);
    [Nfft,NTraces] = size(data);
    feval(mfilename,hAxes,Fs,Nfft,NTraces);        % Get the new parameter structure
    parms = getappdata(hAxes,'SpectrumScopeParameters');
  end;

else                                    % Initialize
  action = 'init';

  if ~isaxes(varargin{1})             % Easy mode, no handle passed in
    % Use current axes
    hAxes = gca;
    Fs = varargin{1};
    Nfft = varargin{2};
    if nargin==3
      NTraces = varargin{3};
    else
      NTraces = 1;
    end;

  else
    % Use current axes
    hAxes = gca;
    data = varargin{1};
    % Ensure that scope has been initialized
    if isempty(parms)
      % Use default values
      Fs = 1;
      data = rowmajor(data);
      [Nfft,NTraces] = size(data);
      feval(mfilename,hAxes,Fs,Nfft,NTraces);        % Get the new parameter structure
      parms = getappdata(hAxes,'SpectrumScopeParameters');
    end;
  end;
else
    hAxes = varargin{1};
    Fs = varargin{2};
    Nfft = varargin{3};
    if nargin==4
        NTraces = varargin{4};
    else
        NTraces = 1;
    end;
end;
end;

switch action
    case 'init'    % Initialize
        parms.Fs = Fs;                % Sample Rate
        parms.NTraces = NTraces;      % Number of lines in plot
        parms.hAxes = hAxes;          % Handle to axes
        parms.Nfft = Nfft;            % FFT Block size

        % Store parameter structure
        setappdata(hAxes,'SpectrumScopeParameters',parms);

        localInitScope(parms)        % Initialize scope
    case 'update'  % Update
        parms = getappdata(hAxes,'SpectrumScopeParameters');

        % Error checking
        % Ensure that scope has been initialized. This shouldn't slip
        % through to here.
        if isempty(parms)
            error(['The spectrum scope must first be initialized' ...
                   'with the sample rate: spectrumscope(hAxes,Fs)']);
        end;

        % Force data to be in columns. Allow for multiple columns. This will
        % error if data actually has more channels than samples.
        data = rowmajor(data);

        % Check that the number of columns corresponds to the number of lines
        nc = size(data,2);            % Number of columns
        if nc ~= parms.NTraces

error(['Size mismatch. You initialized spectrumscope with ' num2str(parms.NTraces) ... ' lines, but just passed in ' num2str(nc) ' channels of data. These' ... ' numbers must be the same.']);
end;

localUpdateScope(data,parms) % Update the scope
end;

% Return appropriate output argument
if nargout
    varargout{1} = parms.hAxes;
end;

%
******************************************************************************
% Initialize the Scope
function localInitScope(parms)

% Set axes
f = (0:parms.Nfft/2-1)*parms.Fs/parms.Nfft;
f = f(:);

% Add line(s)
parms.hLine = plot(f,NaN*ones(length(f),parms.NTraces), ... 'Tag','SpectrumScope', ... 'Parent',parms.hAxes);
set(parms.hAxes,'XLim',[0 parms.Fs/2]);
setappdata(parms.hAxes,'SpectrumScopeParameters',parms);

%%% Get handle to the figure
% Turn doublebuffer on to eliminate flickering
hFig = get(parms.hAxes,'Parent');

if ~strcmp(get(hFig,'Type'),'figure')
    hFig = get(hFig,'Parent');
end;

%%% % Label the plot.
ca = gca;
set(hFig,'CurrentAxes',parms.hAxes);
xlabel('Frequency (Hz)');
ylabel('Magnitude of the FFT components');
set(hFig,'CurrentAxes',ca);
%%
% Turn doublebuffer on to eliminate flickering
set(hFig,'DoubleBuffer','on');
%
***********************************************************************
% Update the plot.
function localUpdateScope(data,parms)

[f,mag] = localfft(data,parms);
%
% Dynamically modify Magnitude axis as we go.
maxM=max(mag(:));
minM=min(mag(:));
yax2=get(parms.hAxes,'YLim');
if minM<yax2(1),
    yax2(1)=minM;
end
if maxM>yax2(2),
    yax2(2)=maxM;
end
set(parms.hAxes,'YLim',yax2)

hLine = parms.hLine;
%
% Update the plot
for ii=1:parms.NTraces
   % Mag{ii} = mag(:,ii);
   set(hLine(ii), 'YData', mag(:,ii));
end;
% set(parms.hLine, 'XData', f(:,1), 'YData', mag(:,1));
% set(parms.hLine, {'YData'}, Mag');
%
***********************************************************************
% Calculate the fft of the data.
function [f, mag] = localfft(data,parms)
%
% Calculate the fft of the data.
% xfft = 2/parms.Nfft*fft(data,parms.Nfft);
% xfft = fft(data);
% Avoid taking the log of 0.
% xfft(xfft == 0) = 1e-17;
% Compute magnitude, dB
% mag = 20*log10(abs(xfft(1:parms.Nfft/2,:)));
% mag = abs(xfft(1:parms.Nfft/2,:));
mag = abs(xfft);
mag = mag(1:parms.Nfft/2);
f = (0:length(mag)-1)*parms.Fs/parms.Nfft;
f = f(:);

% 
% ***********************************************************************
% Utility - isaxes
% function truefalse = isaxes(h);
% ISAXES(H) True if H is a handle to a valid axes

truefalse = 0; % Start false
if ishandle(h)
    if strcmp('axes',get(h,'Type'))
        truefalse = 1;
    end;
end;

% 
% ***********************************************************************
% Utility - rowmajor
% function data = rowmajor(data);
% Force data to be row major. i.e. more rows than columns

[nr,nc] = size(data);
if nc>nr
    data = data';
    [nr,nc] = size(data);
end;

% Data-Analysis.m

cle;
clear all;
A=load('lal74.TXT');
% A = daqread('DATA.daq');
C = 512;
SampleRate = 20000;
H = ((C/2)+1);
L = fix((length(A))/C);
B = ((2*L)-1);
k = 1;
for i=1:B
    for j=1:C
        M(i,j)=A(k);
k=k+1;
    end
    k=k-256;
end
for n=1:B
    D(n,:) = fft(M(n,:));
    E(n,:) = abs(D(n,1:H));
end
for i=1:H
    AVGE(i)=mean(E(1:B,i:i));
end
delT = 2/SampleRate;
TotalT1 = (((length(A)/SampleRate)-(delT/2));
TotalT = (((C*L*delT)-(C*delT))/2);
T = ((C*delT)/4);
delF = 1/(2*T);
Fo = 1/(delT);
f = [0:delF:Fo];
t = [0:T:TotalT];
t1 = [0:delT/2:TotalT1];
pol=polyfit(f,AVGE,6);
pol1=polyval(pol,f);
table=[f AVGE pol1 AVGE-pol1];
figure(1);
plot(t1,A);
xlabel('Time(Sec)');
ylabel('Amplitude(Volts)');
figure(2);
plot(f,E(85,:));
xlabel('Frequency(Hz)');
ylabel('Amplitude of the frequency components');
hold on;
plot(f,AVGE,'-.r',f,pol1,'-k');
figure(5);
waterfall(f,t,E);
xlabel('Frequency(Hz)');
ylabel('Time(Sec)');
zlabel('Amplitude of the frequency components');
figure(3);
pcolor(t,f,E');
shading flat;
for i=1:B
    \( X(i) = \frac{\text{sum}(E(i,:)[1:H])}{\text{sum}(E(i,:))} \);
    \( \text{Favg}(i) = (X(i) \times \text{delF}) \);
    \( \text{favg}(1,1:H) = \text{Favg}(i) \);
    \( \text{std1}(i) = \frac{\sqrt{\text{sum}(E(i,:)[1:H] \times (\text{delF}) - \text{favg})^2}}{\text{sum}(E(i,:)}) \);
    \( i = i + 1 \);
end
figure(4);
plot(t,Favg);
xlabel('Time(Sec)');
ylabel('Frequency(Hz)');
fd = mean(Favg);
STD1 = std(Favg);
STD2 = mean(std1);
STD3 = std(std1);
wl = 633e-9;
theta = 45;
ang = \((\pi \times \text{theta}) / (180)\);
vel = \((\text{wl} \times \text{fd}) / (2 \times \cos(\text{ang}))\)
REFERENCES


