4-25-2016

Investigation of a Tunable 3-D Patterned Illumination Design Implementation for Structured Illumination Microscopy

Christopher Adam Taylor

INVESTIGATION OF A TUNABLE 3-D PATTERNED ILLUMINATION DESIGN IMPLEMENTATION FOR STRUCTURED ILLUMINATION MICROSCOPY

by

Christopher Adam Taylor

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Electrical and Computer Engineering

The University of Memphis

May, 2016

ACKNOWLEDGEMENTS

I would like to thank Dr. Chrysanthe Preza for the opportunity to work in the Computational Imaging Research Laboratory (CIRL) and her guidance as a mentor. I appreciate Dr. Sharon King's help in setting up the experimental optical design as well as her efforts in peer review of my thesis. Dr. Genaro Saavedra of the University of Valencia provided insightful suggestions when reviewing the data I collected on the novel tunable illumination setup. Dr. Ana Doblas was especially helpful when I came to her with questions about her previous work on a biprism optical setup. I would also like to thank Drs. Aaron Robinson, S. King, and A. Doblas for taking the time to serve on my committee. Finally, I would like to thank CIRL members Nurmohammed Patwary and Hasti Shabani for their contributions to the code development and data analysis.

ABSTRACT

Taylor, Christopher Adam. M.S. The University of Memphis. May, 2016. Investigation of a Tunable 3D Patterned Illumination Design Implementation for Structured Illumination Microscopy. Major Professor: Dr. Chrysanthe Preza.

This thesis proposes methods to investigate a novel tunable incoherent 3D patterned illumination suitable for Structured Illumination Microscopy (SIM). A Matlab simulation was designed for the novel tunable illumination in a single and double slit configuration. An experimental setup of the single and double slit configurations was designed and used to acquire experimental data, which was compared with simulation predictions. The comparison aims to scrutinize the lateral and axial frequencies of the sinusoidal illumination pattern and to determine the accuracy of the simulation with real world optics parameters. The simulation result provides a model of the 3D patterned illumination, which is necessary for future use in a SIM setup. An accurate model of the illumination pattern will facilitate designing the forward and inverse SIM imaging models in a different study. The novel incoherent tunable illumination design will theoretically produce better super resolution and optical sectioning capability than current SIM setups that rely on coherent illumination.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.A Conventional Widefield Microscopy and Structured Illumination Microscopy

Conventional widefield microscopy is a widely used tool for biological researchers to investigate living cells or objects at the micrometer scale, but has some limitations. The resolution of the widefield microscope is fundamentally limited by the wavelength of light and the numerical aperture of the objective lens. In 1873, Abbe defined this limitation as the diffraction limit $d = \frac{\lambda}{2NA}$ where d is the radius of the spot produced by light with a wavelength of λ, and NA is the numerical aperture of the optics [1]. An optical imaging system that can produce images at its maximum theoretical resolution limit is said to be "diffraction limited" [2]. For example, an optical system using light with a wavelength of 635 nm and an objective lens with a NA of 1.4 would have a lateral resolution of 226.8 nm.

Another fundamental limitation of the widefield microscope is lack of optical sectioning capability. Optical sectioning refers to the process of capturing images at different depths of a sample without intensity contributions from objects in adjacent planes. A compilation of two dimensional images can be stacked in the direction of the optic axis to provide a three dimensional image. However, in practice in-focus as well as out-of-focus light from other depths of the sample contributes to each 2D image which can be disadvantageous; especially when trying to focus deep into a sample. The lack of optical sectioning capability can be better understood by looking at the optical transfer function of the microscope.

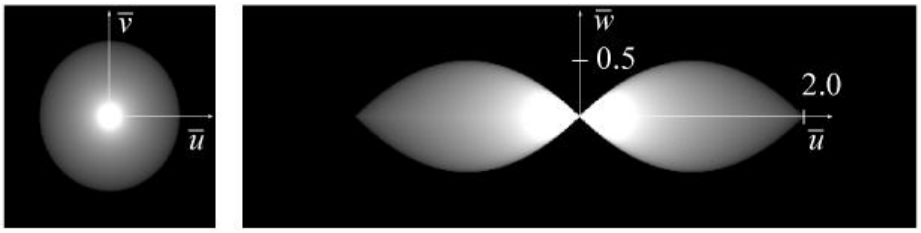

Figure 1-1. Transverse and axial cut of the 3D wide field OTF, normalized coordinates. Reprinted from [3].

The optical transfer function of the conventional widefield microscope has what is colloquially known as the "missing cone problem", because the missing axial spatial frequencies define a cone shape along the Z-Axis [4]. The missing axial frequencies have a profound effect on the axial resolution. The widefield microscope has an axial resolution that is approximately half the transverse resolution. If the missing information could be recovered, the optical system could achieve true optical sectioning capability.

Structured Illumination Microscopy (SIM) attempts to solve these two problems of the widefield microscope by modulating the light distribution which excites the sample. SIM can theoretically extend the lateral resolution to a maximum of two times the diffraction limit. Any resolution above the diffraction limit is known as "super resolution" (SR) [5]. A spatially structured illumination pattern is used to modulate high frequency information into the pass band of a microscope [5]. More recent versions of SIM have also been demonstrated to increase axial resolution and produce optical sectioning capabilities [5, 6]. Filling the "missing-cone" in the optical transfer function is synonymous with capturing axial spatial frequencies that enable improved axial resolution and optical sectioning capability.

The field of structured illumination microscopy encompasses a large variety of different techniques used to produce structured illumination; some of these techniques are discussed in Chapter 2. This research investigates a new method for generating a "tunable structured illumination pattern" with an incoherent light source, a multiple slit aperture, and a biprism [7].

1.B Thesis Research Overview

The proposed research presents a new structured illumination technique that aims to provide lateral super resolution, optical sectioning capability in microscopy, and tunability of the lateral illumination frequency over a certain range of spatial frequencies. The three-dimensional illumination pattern is laterally and axially modulated which allows the optical system to pass "missing-cone" information and lateral super resolution information.

The thesis builds on the work of Ana Doblas *et al.* [7] to investigate an illumination pattern that can be used in an incoherent structured illumination microscopy setup. Doblas *et.al.* have previously established that a sinusoidal illumination pattern can be created through interference by using a slit aperture followed by a converging lens, and a Fresnel biprism. When a wavefront from a point source illuminates the biprism, the exiting wavefront is split into two waves that proceed from two virtual point sources that are mutually coherent. This configuration will be referred to as the fixed lateral frequency setup. The fixed lateral frequency setup is capable of producing a pattern with a spatial frequency close to the cutoff frequency of a microscope objective while maintaining a high contrast intensity pattern despite using an incoherent light source. The novel tunable setup (refer to Figure 2-1) being investigated in this thesis is a variation of the fixed

lateral frequency setup that additionally allows for a tunable illumination pattern in a set range of lateral frequencies. The range of spatial frequencies is predominantly determined by characteristics of the biprism such as its apex angle and refractive index. The frequency of the pattern is tuned by the distance between the biprism and the slit aperture. This type of 3D illumination pattern can theoretically produce an OTF in the optical imaging system capable of lateral super resolution and optical sectioning similar to 3-wave interference, but with the added benefit of eliminating coherent noise (discussed further in Section 2.B). Lateral super resolution and optical sectioning capabilities are not the focus of this thesis; however, they are part of the motivating factors for this research. Based on Doblas *et al.* previous work, the illumination pattern will theoretically have a high signal to noise ratio, which will be useful for 3D super resolution in optically thick samples. The focus of this thesis will be investigating the illumination pattern generated by the new variation of the biprism setup and how it compares to theoretical predictions from simulation.

1.C Objectives

The two main objectives of this research are to: 1) implement both a simulation and experimental setup of the novel tunable illumination pattern, and 2) develop methods for measuring and comparing simulation data, experimental data, and theoretical calculations. The two configurations to be tested contain a single or a double slit aperture. The single slit aperture study focuses on testing the lateral frequency component of the system, while the double slit aperture study focuses on testing the axial frequency component. Both of these studies evaluate varying biprism positions to observe the tunability of the system.

1.D Contributions

This research contributes to Dr. Preza's NSF project to build a novel SIM system. There are 3 parts to this new system: the structured illumination pattern, PSF engineering to reduce spherical aberration, and computational methods for the inverse model to reconstruct the image from raw SIM data. This thesis will contribute to the first part of this project to provide a simulation model for the illumination pattern that can be used in the forward imaging model for the SIM system. Along with the simulation model, robust tools to measure and compare simulation, experimental, and theoretical data will be needed to verify the accuracy of the model.

This study focuses on developing a robust simulation for the novel tunable illumination pattern that can be used in the forward imaging model. It is important that the simulation model accurately describes the data from the experimental setup. The simulation model for the novel illumination setup will eventually be used in the forward imaging model of the SIM system, which is vitally important for developing an accurate inverse model that will be capable of handling raw experimental SIM data. Verification of simulation with the experimental illumination setup is important because it will eventually be paired with an open SIM setup, where the entire SIM system can be studied before attempting to adapt the illumination pattern to a commercial widefield microscope.

There are two methods evaluated in the thesis that were designed to measure the lateral and axial frequency components of the 3D illumination pattern, the Average Frequency Algorithm and the Power Spectral Density method. Both methods will be discussed in Section 3.F and 3.G.

1.E Benefits to Biological Researchers

For biological and medical research, there is a need for improved live cell imaging. Current SIM systems offer super-resolution (SR) imaging for thin samples only (essentially 2D imaging), and confocal techniques allow for 3D imaging with enhanced optical sectioning but minimal improvements in lateral resolution. SIM combines the best advantages of confocal scanning microscopy and conventional widefield microscopy. Imaging applications require the highest resolution, sufficient imaging depths for thick samples, minimal sample disruption, and fast data-acquisition. By transforming SIM into a 3D SR tool, biological researchers can investigate issues around cell motility, behavior, and regulation in 3D environments that mock the spatial organization of organs.

CHAPTER 2 BACKGROUND

2.A Structured Illumination Microscopy (SIM)

As previously mentioned, the motivation behind this research is to investigate a tunable illumination pattern to develop an improved SIM system that can obtain 3D cellular information through optically thick specimens. SIM is an important modality because it pushes the limits of wide field microscopes without the disadvantages of confocal microscopy. A confocal microscope is capable of capturing 3D images, by scanning a 3D object point by point. The images produced have the property of optical sectioning because the setup rejects out-of-focus light, thus eliminating the "missing-cone problem" that was described in Section 1.A. However, because confocal must scan an image pixel by pixel in three dimensions, it is unsuitable for live cell imaging as the process will produced blurred images from a dynamic sample. Generally confocal excels at capturing 3D images of static specimens that can be subjected to large amounts of photons. The small pinhole used in the point-wise scanning method has to be large enough to allow in-focus light to pass, while also being small enough to improve spatial resolution and block out-of-focus light [8]. SIM has an advantage over confocal in that it requires only one dimensional scanning through the Z-axis while no light signal is lost. Certain variations of SIM can also solve the "missing-cone problem" that inhibits conventional wide field microscopy.

To understand how SIM development has evolved, we will review below the three main SIM approaches based on: 1) incoherent illumination; 2) 2-wave coherent illumination; and 3) 3-wave coherent illumination.

2.B Recent Advances in 3D SIM

Incoherent SIM as discussed in Karadaglic and Wilson's paper [9] is demonstrated as a two-dimensional structured illumination pattern; the pattern only has modulation in the lateral direction. The system can achieve optical sectioning capability, but cannot achieve lateral super resolution. The method illuminates the sample with spatially structured excitation light, so that the illumination pattern is strongest in the focus of the sample and decreases in contrast with defocus. The structured illumination is a sinusoidal fringe pattern of bright and dark bands with a certain frequency. At least 3 images must be acquired with a phase shift in the excitation pattern in order to capture adequate SIM data to reconstruct the underlying high resolution image. One disadvantage of this type of SIM system is that the contrast of the illumination pattern is rapidly attenuated due to the nature of the optical transfer function for an incoherent imaging system. In effect, the contrast of an illumination pattern is diminished as the modulation frequency of the pattern approaches the cut-off frequency of the imaging system. For this reason, the incoherent illumination approach cannot achieve lateral super resolution at the theoretical limit of twice the conventional widefield microscope's lateral resolution. The method must use a low frequency pattern (approximately one-third of the cut-off frequency of the objective lens) in order to obtain high contrast fringes [8, 10]. The low frequency pattern enables optical sectioning capability and is capable of achieving axial resolution similar to a confocal microscope. A commercial example of this method is the Zeiss ApoTome™.

In 2-wave coherent illumination SIM, the 2D fringe pattern is created by the interference of the two waves, and thus the system can achieve either optical sectioning

or lateral super resolution, but not both at the same time [8] . The system can achieve maximum lateral super resolution by using a fringe pattern with frequency close to the cut-off frequency of the objective lens. Lowering the frequency of the pattern allows the system to provide imaging with optical sectioning capability [11]. The optical transfer function for coherent imaging does not attenuate the contrast of the fringe pattern. The sample is illuminated by a spatially structured illumination pattern that makes normally inaccessible high-resolution information visible in the observed image in the form of Moire fringes [8]. At least three raw SIM images per illumination direction, (typically 9 total) must be captured by shifting the illumination pattern through three phases. The raw SIM data is then processed to extract the underlying high resolution image.

A 3-wave coherent illumination SIM system as discussed by Gustafsson can simultaneously achieve optical sectioning and lateral super resolution (also known as 3D super resolution) [5]. Unlike the previous SIM modalities presented, 3-wave coherent illumination SIM produces a 3D illumination pattern that is both laterally and axially modulated. The effective OTF of the 3-wave system can fill the missing cone while also achieving lateral super resolution simultaneously. The system requires at least 5 shifted phases of the illumination pattern per illumination direction (typically 15 total) in order to reconstruct a high resolution image from the raw data [5, 8]. The Zeiss Elyra™ is a commercial example of 3-wave Structured Illumination Microscopy. The Zeiss Elyra™ uses 3-wave coherent structured illumination and has been shown to achieve double the resolution at 100 nm laterally and 200 nm axially [12].

2.C Novel Tunable Illumination Pattern

Recently, a new type of structured illumination based on a Fresnel biprism was shown to have advantages over Gustafsson's 3-wave coherent illumination. A 2-wave incoherent pattern that has similar imaging properties to Gustafsson's 3-wave SIM can be created by using at least a two or more slit aperture, a biprism, and a converging lens. In this section, we will highlight the importance of each of these three elements in the illumination system and the overall advantages of the system compared to previous SIM modalities.

The Fresnel biprism is the optical element responsible for generating the sinusoidal illumination pattern. The biprism is essentially two thin equal prisms joined at their base. By illuminating the biprism with a monochromatic point source, two mutually coherent virtual sources are created at the source plane [7]. The interference between these two virtual sources generates a sinusoidal pattern after the biprism, which is similar to Young's double slit experiment where two waves interfere.

A slit can replace the point source and be used to illuminate the biprism (Figure 2-1). The slit acts as a stack of point sources along y and does not influence the visibility of the fringe pattern, but does affect its maximum irradiance [7]. When illuminating the biprism with a single slit, two virtual slits will appear on either side of the source slit at the source plane. The illumination pattern resulting from a single slit and a biprism is comparable to a 2-wave illumination system, where the two virtual sources interfere to form a sinusoidal pattern of fringes. When two slits are spaced symmetrically on either side of the apex of the biprism, a 2-wave interference pattern is generated after the biprism with a sinusoidal pattern laterally and axially. This novel illumination pattern has

SIM imaging properties comparable to 3-wave coherent SIM, but without the disadvantages of coherent light.

In order to obtain a lateral period that does not change with axial distance, a converging lens is placed after the biprism so that the slit aperture is placed in the front focal plane of the converging lens. The spherical wavefront emitted from the slit aperture (and thus the virtual sources) is transformed to a parallel wavefront by the converging lens. In this configuration, the interference pattern after the converging lens maintains a constant lateral period with changing axially distance (z).

There are a few key advantages of this new type of 3D illumination pattern that are investigated and implemented as part of this thesis research. The setup is capable of tuning the illumination pattern's lateral spatial frequency by changing the distance between the biprism and the grating (refer to Figure 2-1). The range of available frequencies is determined by the angle of the biprism and its axial displacement with respect to the slit aperture. Both of these parameters can be chosen in order to achieve the maximum spatial frequency for the microscope objective lens. This system property is in contrast to commercial SIM systems such as the Zeiss ApoTome™ that rely on a discrete number of spatial grid frequencies that are specifically designed for a particular set of objective lenses. The new biprism-based setup is much more flexible by allowing for a continuous range of spatial frequencies. The biprism setup also has an additional advantage over current incoherent SIM systems. The Zeiss ApoTome™, for example, is capable of using only low frequency illumination patterns when compared to the cut off frequency of the objective lens in order to maintain high contrast as discussed in Section 2.B. However, the biprism setup is able to maintain high contrast fringes even while

approaching the cutoff frequency of the objective [7]. Using this new illumination pattern for SIM will allow the system to capture more high frequency information during SIM data acquisition. Another advantage over current incoherent SIM systems is that the illumination pattern is axially modulated. This axial modulation will allow the system to pass frequencies normally lost in the "missing-cone" region while also achieving lateral SR. The axial modulation for two slits is similar to a 3-wave coherent system, but the biprism system eliminates coherent noise by using incoherent light.

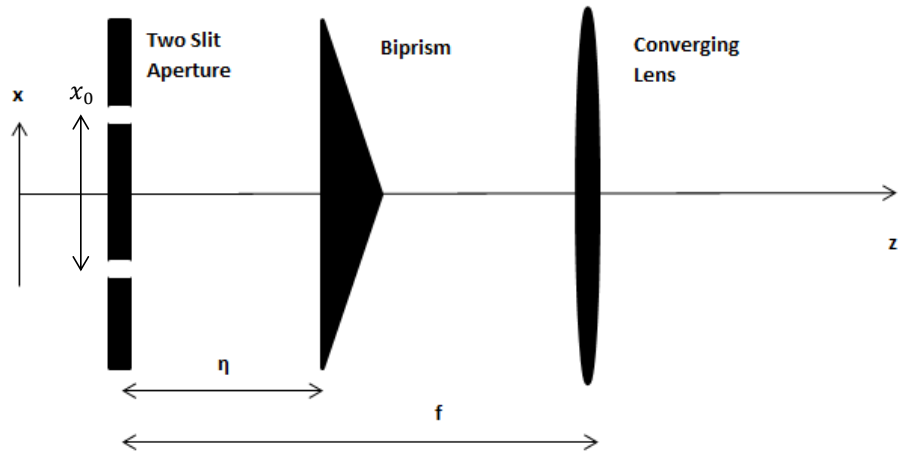

Figure 2-1. Two Slit diagram. A two slit aperture followed by a Fresnel biprism and a converging lens. The aperture is illuminated by a spatially incoherent light source. f is focal length of the converging lens and η is the distance between the biprism and the slit aperture. Increasing η increases the lateral frequency of the illumination pattern behind the converging lens. η is confined to the length of f.

2.D Basic Theory and Simulation of the Irradiance Signal in the Fixed Lateral Frequency Setup

This section discusses the pattern produced by a Fresnel biprism for the fixed lateral frequency setup (also known as the slit - converging lens - biprism setup investigated by Doblas *et al.*). It should be noted that in the paper by Doblas *et al.* [7], the slit-biprism (spherical) and the slit-converging lens-biprism (parallel) setups were discussed and analyzed. Doblas *et al.* named the slit-converging lens-biprism setup as the "parallel configuration" because the parallel wave front after the converging lens can maintain a uniform lateral period through *z*. The novel tunable slit-biprism-converging lens setup investigated in Chapters 3 and 4 of this thesis is based on the model of the fixed lateral frequency setup, so it is important to understand the working principles of the setup. In what follows, the pertinent equations and variables (refer to Table 2-1) are relevant to the fixed lateral frequency setup (non-tunable illumination pattern) based on previous work done by Doblas *et al.* [7].

Table 2-1. List of Important variables related to Simulating the Irradiance Signal. Found in Equation 1-6

λ	Wavelength of Source [mm]
n	Refractive Index of Biprism
δ	Biprism Angle [radians]
x_0	Slit Separation [mm]
Δ	Slit Width [mm]
η	Biprism Position (distance from source plane) [mm]
f	Focal Length of Converging lens (distance from source plane) [mm]

We define the source plane where the point source emits a spherical wave front at distance η before the biprism, and an observation plane where the pattern is observed at distance z after the converging lens (Figure 2-1). When a spherical wave front illuminates a biprism, the wave front is split into two spherical waves that can be traced to two virtual sources that lie at the source plane. The two virtual sources are separated by the distance [13]

$$\boldsymbol{a = 2\eta(n-1)\delta} \tag{2-1}$$

The virtual sources are mutually coherent and a parageometrical description of the irradiance distribution can be written as

$$\boldsymbol{I_0(x,y,z;\eta) = 1 + V(z)cos\left(\frac{2\pi x}{p_x}\right)}, \tag{2-2}$$

where the visibility of the pattern is

$$\boldsymbol{V(z) = \frac{sin\left(\pi N \frac{M_s x_0}{p_x}\right)}{N sin\left(\pi \frac{M_s x_0}{p_x}\right)}\, sinc\left(\Delta \frac{M_s}{p_x}\right)} \tag{2-3}$$

and

$$\boldsymbol{M_s = \frac{-z}{f}}. \tag{2-4}$$

The lateral period of the interference pattern is

$$\boldsymbol{p_x = \frac{\lambda}{2(n-1)\tan(\delta)}}. \tag{2-5}$$

The period for the visibility of the resonant planes for a double slit setup (single slit does not produce an axial frequency term) is

$$\boldsymbol{p_z = \frac{f}{x_0}\, p_x = \frac{\lambda f}{2x_0(n-1)\tan(\delta)}}, \tag{2-6}$$

and the axial period of the interference pattern is

$$\mathbf{p_m = 2p_z}\,. \tag{2-7}$$

The pattern obtained from the parageometrical model is modified by diffraction effects due to the apex of the biprism [7]. The amplitude distribution after the biprism can be calculated by using the Fresnel-Kirchoff integral, assuming the biprism is a thin object. Equations (2-8) and (2-9) more accurately describe the Irradiance signal than Equation (2-2) by taking into account the diffraction effects of the biprism's apex. Equation (2-8) fully describes the illumination pattern and has the same form as equation (2-2) as shown in Ref. [7].

$$\boldsymbol{I_0(x,y,z;\eta)} = \left| \begin{array}{l} \exp\left(-j\frac{\pi}{\lambda z}(x+\lambda z u_0)^2\right)\left\{\frac{1+j}{2} + Fres\left[\sqrt{\frac{2}{\lambda z}}(x+\lambda z u_0)\right]\right\} \\ + \exp\left(-j\frac{\pi}{\lambda z}(x-\lambda z u_0)^2\right)\left\{\frac{1+j}{2} - Fres\left[\sqrt{\frac{2}{\lambda z}}(x-\lambda z u_0)\right]\right\} \end{array} \right|^2 \tag{2-8}$$

where $$\boldsymbol{Fres[\alpha] = \int_0^{\alpha} exp\left(j\frac{\pi}{2}x^2\right)dx = C(\alpha) + jS(\alpha)},\tag{2-9}$$

and $$\boldsymbol{u_0 = \frac{(n-1)\tan\delta}{\lambda}}\,. \tag{2-10}$$

The interference terms of the parageometrical description and the Fresnel-Kirchoff description have the same lateral period as well as axial period for the two slit setup [7]. The fixed lateral frequency simulation model also accounts for the finite width of the slit aperture by convolving the $I_0(x,y,z;\eta)$ function with a rectangle of width Δ to produce the final irradiance signal $I(x,y,z;\eta)$. Furthermore, this convolution of a rect(.) function is modeled as the multiplication of a sinc(.) function in the frequency domain. For the purpose of this thesis research, it is important to note that Doblas concluded the pattern will maintain strong visibility as long as the slit width is sufficiently smaller than

the slit separation for multiple slit configurations. A sufficiently small slit width is approximately an order of magnitude smaller than the slit separation. Large slit widths will significantly reduce visibility of the pattern with increasing *z*. For a more in depth analysis of the visibility of the pattern, refer to [7].

CHAPTER 3 METHODOLOGY

3.A Simulation of Tunable Illumination Pattern

The two main objectives are to investigate a new optical setup for a novel tunable illumination and develop methods to measure the lateral and axial frequencies of its structured illumination pattern. The approach is to extend the previous study of the fixed lateral frequency illumination pattern [7]. First, in order to investigate the new optical setup we simulate and test a configuration consisting of a slit aperture followed by a biprism, then a converging lens (the novel tunable setup, Figure 2-1). The illumination configuration allows for a tunable frequency pattern based on changing the position of the biprism with respect to the slit aperture. In this study, the fixed lateral frequency simulation has been extended to scale the illumination pattern in proportion to the distance of the biprism from the slits. Second, simulation is used to develop methods to measure the frequency of the illumination pattern laterally and axially. The fixed lateral frequency simulation uses a normalized X coordinate; therefore, the lateral frequency cannot be quantified. Doblas *et al.* [7] have developed the theoretical equations for calculating the lateral and axial frequencies, which are used to verify the methods developed as part of this thesis for measuring the lateral and axial frequencies. It is important to note that the extension of the fixed lateral frequency simulation also includes the ability to specify the simulation's lateral and axial resolution in terms of mm/pixel, which is useful for quantitatively comparing simulated and experimental data.

Simulation Objective 1. The following describes how the fixed lateral frequency simulation has been extended to scale the illumination pattern relative to the position of the biprism. Generation of the pattern is based on the principle of the Fourier transform

relationship between the cosine function in the frequency domain and two delta functions in the space domain. The virtual sources created by the biprism act as two delta functions in the space domain. According to the Fourier transform relationship, the distance between the two delta functions is directly proportional to the frequency of the cosine function. From this, we can establish that the distance between the virtual slits is directly proportional to the lateral frequency of the illumination pattern, so if the distance between the two virtual sources is halved then the frequency is also halved. Doblas *et al.* [7] states that the distance between the virtual slits (a) is directly proportional to the distance between the biprism and the slit aperture (η) in Equation (2-1). The frequency of the illumination pattern can be scaled based on the ratio $(\frac{\eta}{f_c})$ of the biprism distance from the slit aperture (η) and the focal distance of the converging lens (f_c).

To calculate the new lateral and axial periods of the pattern produced by the novel tunable illumination setup, Equations (2-5) and (2-6) were multiplied by the inverse of the frequency scaling ratio $(\frac{\eta}{f_c})$ to produce the new lateral and axial period for the novel tunable illumination setup:

Comment [CP(1]: Is this the period of the pattern or of the visibility?

$$\boldsymbol{p_x = \frac{\lambda f_c}{2\eta(n-1)\tan(\delta)}} \quad \textbf{(3-1)}$$

and

$$\boldsymbol{p_m = 2p_z = 2\frac{f_c}{x_0}p_x = \frac{2\lambda f_c^2}{2x_0\eta(n-1)\tan(\delta)}}\ , \quad \textbf{(3-2)}$$

where the variables have already been defined in Table 2-1.

Simulation Objective 2. The following describes a method to remove the x normalization factor as used in Doblas *et al.* [7] such that the revised simulation starts at a base scale of 1 mm/pixel enabling the user to simply specify any mm/pixel value to

simulate any camera with a specific pixel pitch. The fixed lateral frequency simulation's parameters are defined in mm, so for consistency the novel tunable simulation's pixel pitch was defined in mm/pixel.

The *x* normalization factor was not explicitly stated in Doblas *et al.* [7], so it was calculated using the following procedure. Using a set of arbitrary parameters, the theoretical lateral period was calculated using Equation (3-1), which gives the period in millimeters. A simulation using a set of arbitrary parameters then produces an irradiance signal with a lateral period that can be measured in terms of pixels. The normalized mm/pixel value is determined by these two quantities and was found to be

$$\boldsymbol{normalized\ \frac{mm}{pixel} \stackrel{\text{def}}{=} \frac{2}{number\ of\ pixels\ in\ the\ x\ direction}} \tag{3-3}$$

Multiplying the x-axis by the inverse of Equation (3-3) brings the system to a 1 mm/pixel scaling. Now that the simulation starts at a base scale of 1 mm/pixel, the user can simply specify any mm/pixel value to simulate any camera with a specific pixel pitch in order to compare simulation and experimental measured frequencies both laterally and axially.

Simulation Normalization. In all cases where the simulated data is compared to experimental data, the simulation data is normalized so that the total power of the simulation matches experimental data. An average along the y-dimension is calculated for each x value in a given z-plane, assuming that the illumination pattern is vertically aligned. The averages are summed along x to find the total power of the image in a particular z-plane. The simulated data is simply summed along the *x*- dimension because only one *y* value is simulated (the center value). The simulated data for the particular z-

plane is multiplied by the ratio of the total power of the experimental image to the total power of the simulated data (Equation (3-4)).

$$\boldsymbol{intensity\ normalization} = \frac{\frac{\mathbf{1}}{\boldsymbol{N_y}}\sum_x \sum_y (\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z_0}))}{\sum_x \boldsymbol{i}(\boldsymbol{x}, \mathbf{0}, \boldsymbol{z_0})}, \quad \textbf{(3-4)}$$

where $\boldsymbol{N_y} = \boldsymbol{number\ of\ pixels\ in\ the\ y\ direction}$

and $\boldsymbol{z_0} = \boldsymbol{a\ specific\ Z\ plane.}$

Comment [CP(2]: Define N_y below and z_0, is y=0 the center? confusing

3.B Single Slit Variable Aperture Configuration

The purpose of the single slit experiment is to verify the tunability of the frequency pattern by changing the biprism position, and to show that the frequency of the pattern remains constant through the Z-Axis. As discussed in Section 0, the frequency of the pattern should be uniform through Z and there should not be any axial modulation with the single slit setup.

An imperial micrometer variable slit aperture (ThorLabs Model# VA100) is used for the single slit configuration. The variable aperture was not fully closed at tick marker zero. In order to find the actual closed position of the variable slit, the slit was calibrated at 0 when no light reached the camera. It was found the actual closed position of the variable slit to be at tick marker -0.003 inches. Measurements of the variable single slit reported in this thesis are pre-adjusted with the calibration and reported as true values.

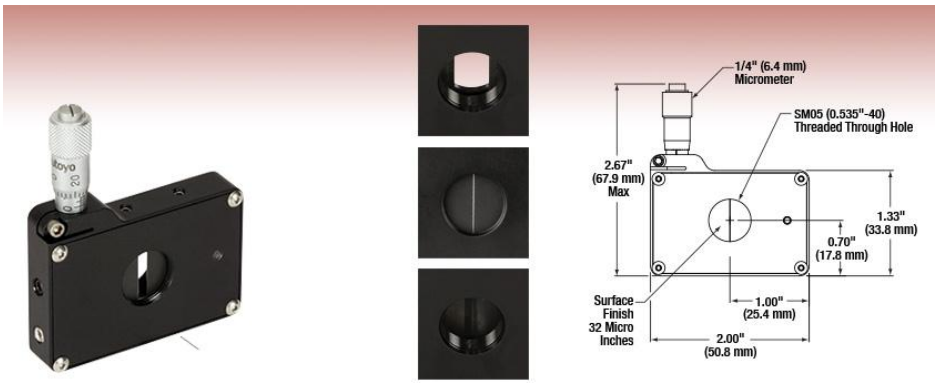

Figure 3-1. Single Slit Aperture from ThorLabs for size reference. Reprinted from [14].

3.C Double Slit Aperture Configuration

The purpose of the double slit experiment is to verify the axial modulation along the Z-Axis as described in Section 2.C. The frequency of the pattern should maintain the same lateral frequency as a single slit experiment with similar parameters (biprism position, biprism angle, converging lens, wavelength of light, etc.). The double slits aperture shown in the figure below has a slit width of 70um and slit separation of 200, 300 and 500 um from left to right. A double slit aperture on a glass plate from Mansion Schools Model# U22014 is used for the double slit experiment.

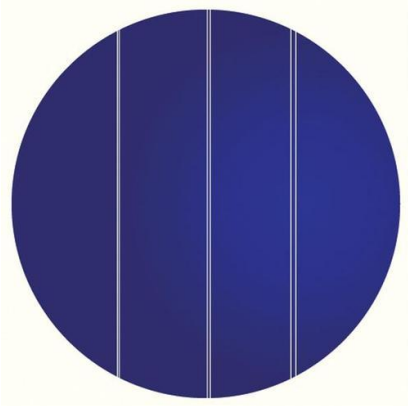

Figure 3-2. Double Slit Aperture from Mansion Schools. Changing slit separations requires readjusting lateral alignment with the biprism. Reprinted from [15].

3.D Coherent and Incoherent Illumination

For coherent illumination, a laser with a wavelength of 635 nm is used to align the optical elements and collect coherent data. The beam of the coherent light was used to verify the alignment of the optical elements. A neutral density filter is placed in the

beam's path, so the Axiocamera can capture pictures of the illumination directly. Without the neutral density filter, the coherent light saturates the camera's exposure.

For incoherent illumination, an LED with a wavelength of 470 nm and variable intensity is used to collect incoherent data. The two sources are toggled in and out of the optical system via a flipper mirror. The variable intensity and exposure settings on the Axio camera allow for direct capture of the illumination without a neutral density filter.

As discussed in Section 2.C, the incoherent illumination is the primary source for investigation. The novel tunable incoherent illumination setup has theoretically better super resolution and optical sectioning capability than previous incoherent illumination setups used in SIM. The novel setup will also theoretically produce a better signal to noise ratio because coherent noise is eliminated from the system.

3.E Biprisms

Two biprisms were investigated (FBP2020G-175 and FBP2020G-179). The angle of the biprism determines the frequency range of the structured illumination. The two biprisms were used to verify that the change in the biprism angle would match the frequency change in simulation and experiment. It should be noted that the higher frequency range of the FBP2020G-175 biprism could not be captured directly by the Axio camera's 6.4 μm per pixel resolution. At that pixel size the camera has 156.25 pixels per mm or approximately a 78 lp/mm sampling rate. According to the Nyquist Theorem, the sampling rate must be twice the frequency of the pattern; therefore, the camera can adequately sample 39 lp/mm without aliasing. The FBP2020G-175 biprism can theoretically produce a pattern with a frequency range of 19.14 lp/mm to 95.68 lp/mm for an arbitrarily chosen (η) range of 30 to 150 mm for f = 150 mm and a

wavelength of 470 nm. Refer to Section 3.A on theoretical calculations for the lateral period/frequency. One solution to this problem is to magnify the illumination pattern with a secondary lens relay in order to avoid aliasing at the camera's lateral resolution.

3.F Calculation of Lateral Frequency in Experimental Data with the Average Frequency Algorithm (AFA)

A new Matlab function was created in order to calculate the frequency of the fringe pattern in an XY plane (lateral frequency). The technique finds the peaks of the sinusoidal pattern, calculates the distances between the peaks using the known mm/pixel ratio, and averages the distances together to find a collective period. This algorithm, referred to the Average Frequency Algorithm (AFA) was created to find the frequency of an intensity profile (Figure 3-3) through the illumination pattern along the *x* direction while ignoring background noise. A profile with a significant number of noisy peaks would create an overestimation of the frequency.

The algorithm follows these steps to find the true peaks. First, it sets a window to include peaks that are within 40% of the max peak, in order to avoid peaks due to the background noise. Second, it finds all the peaks within the window and removes noisy peaks that are beyond a set frequency limit. For example, a frequency limit is set to double (100% error) the measured frequency obtained from simulated data, so the algorithm will accept any peaks that are within 100% error of the simulation frequency and reject any peaks that would overestimate the frequency by 100%. This is the same as rejecting any noisy peaks that are closer than 50% of the simulation period to a real peak.

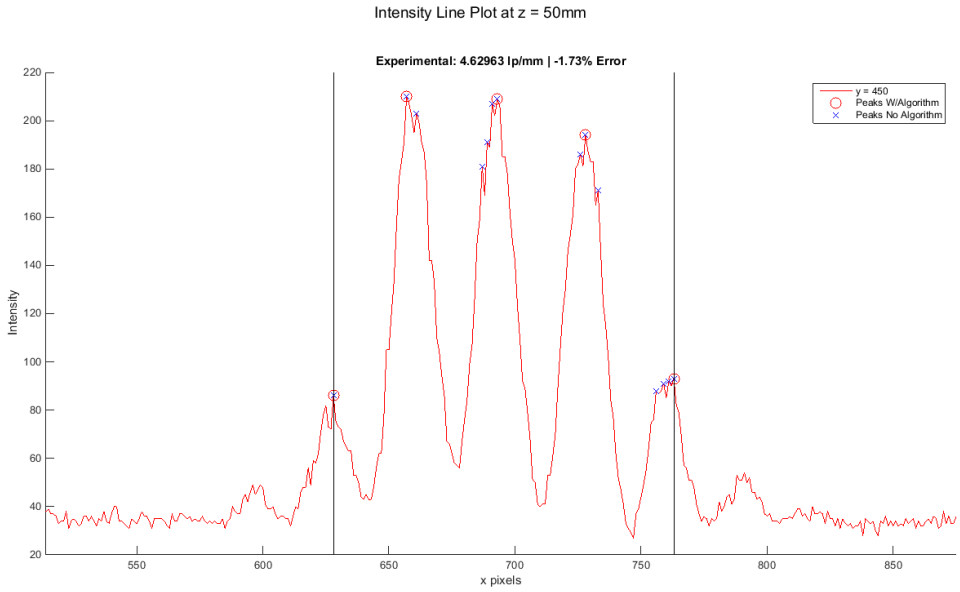

Figure 3-3. Example of Experimental Coherent Data With Noisy Peaks. The blue x's represent peaks found by MATLAB's findpeaks() function. The red o's represent peaks found with the adjusted algorithm. Using the red o's as peak data points, the experimental frequency is calculated as 4.62963 lp/mm. The experimental data has a -1.73% error compared to Simulation frequency: 4.7185 lp/mm. Light wavelength 635nm, Biprism FBP2020G-179, Biprism position η 50mm, converging lens f 150mm, camera resolution 0.0064 mm/pixel.

We expect the algorithm to reach the maximum 100% error if there is no fringe pattern or if the data is pure noise. Figure 3-5 and Figure 3-6 are intensity line profiles at y =400 and y=450 using identical average frequency algorithm parameters, where y is the pixel row in the image being examined (Experimental data in Figure 3-4). At y=400, there is a fringe pattern of 4.88 lp/mm that has a 3.32% error in comparison to the simulation frequency of 4.7185 lp/mm. At y =450, there is no fringe pattern and the algorithm is looking at background noise, therefore the algorithm estimates a frequency at the maximum 100% error.

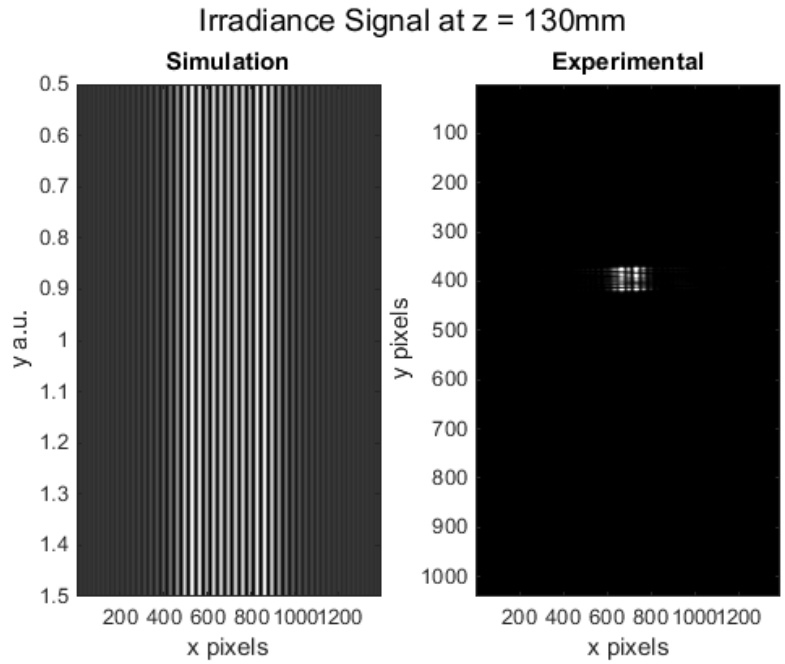

Figure 3-4. Left Incoherent Simulation Data. Right Experimental Coherent Data. Although the left is incoherent data and the right is coherent data, the lateral frequency of both patterns should be the same. Figure 3-5 and Figure 3-6 are intensity line plots through the experimental coherent data.

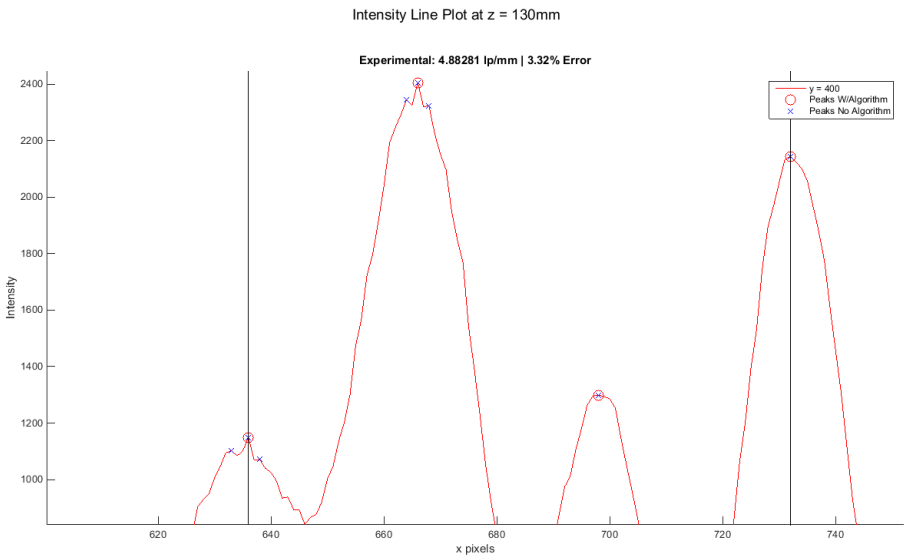

Figure 3-5. Magnified intensity line profile for y = 400. Fringe pattern of 4.88 lp/mm.

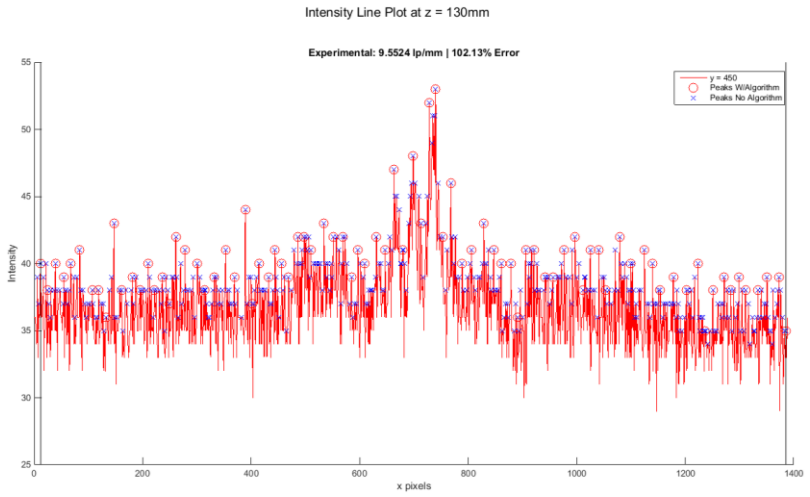

Figure 3-6. Intensity line plot through y = 450. Background noise. AFA approximates a frequency that maxes out at 100% error as defined by the user.

3.G Power Spectral Density (PSD) Analysis of the Illumination Pattern's Lateral Frequency

Another way to calculate the frequency of the pattern is to analyze the PSD of a line profile through the structured illumination. The PSD of the signal describes the power present in the signal as a function of frequency [16]. The PSD can be calculated via the following algorithm in Matlab using the variables described in

Table 3-1:

$$\boldsymbol{Y = fft(signal, nSamples)} \tag{3-5}$$

$$\boldsymbol{Pyy = Y .* conj(Y)/nSamples} \tag{3-6}$$

$$\boldsymbol{f = \frac{1}{\frac{mmPerPixel}{nSamples}} * \left(0 : \left(\frac{nSamples}{2} - 1\right)\right)} \tag{3-7}$$

Table 3-1. Variables used in calculating Power Spectral Density.

$signal$	Line profile through the illumination pattern
$nSamples$	Number of samples in the line profile
Pyy	Power Spectral Density of the signal
$mmPerPixel$	mm/pixel of the signal
f	The frequency axis for the first half of the PSD. (The remainder of the points are symmetric)

In the following figures, an example is shown on the PSD method and how the lateral frequency is calculated in experimental data acquired with a single slit configuration. Figure 3-7 shows the experimental irradiance signal and the line profile taken through the center of the data at y=500. Figure 3-8 shows the line profile as a function of intensity vs. the x axis in pixels. Figure 3-9 shows the PSD of the line profile calculated using Equations (3-5), (3-6), and (3-7). The local maximum in the PSD represents the strongest frequency component, which indicates the frequency of the line profile. Note that there is some low frequency noise in the PSD for experimental data. In the case of simulation data, the low frequency noise is virtually absent.

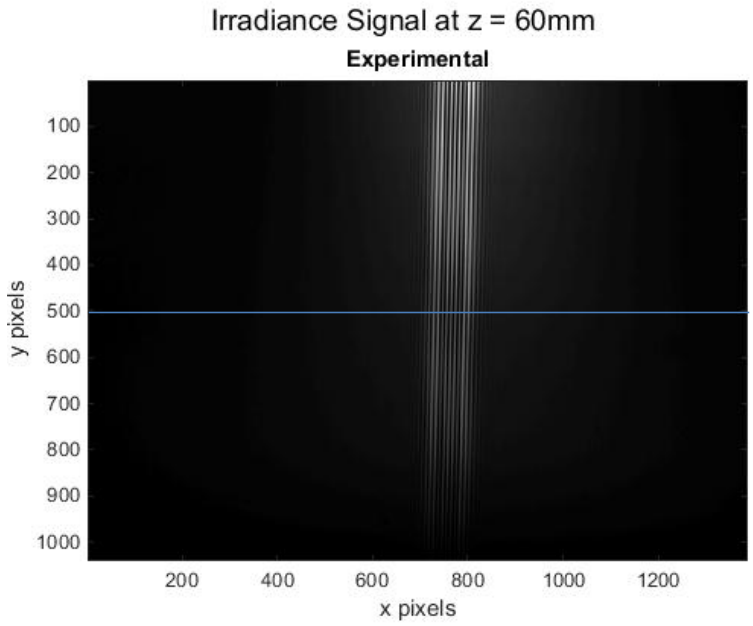

Figure 3-7. Experimental Irradiance Signal. Note: even though the line profile is not taken through the brightest section of the pattern, the PSD can still accurately describe the frequency of the pattern.

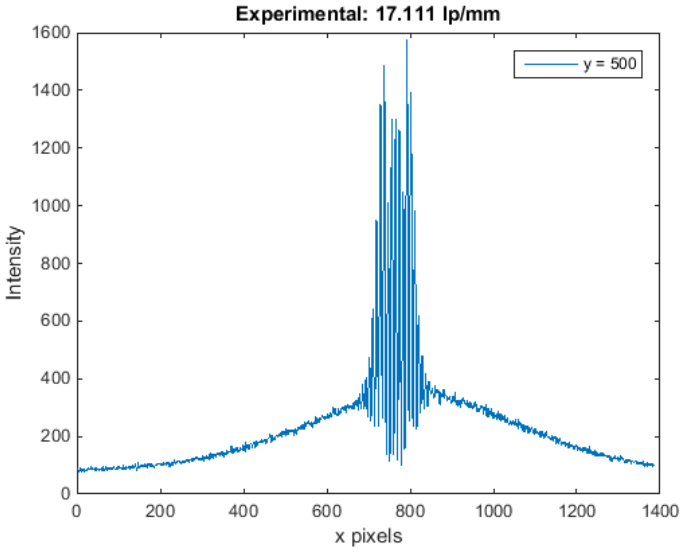

Figure 3-8. Intensity Line Profile taken at y=500 from the Irradiance Signal. By Analyzing the PSD, the frequency of the pattern is determined to be 17.11 lp/mm.

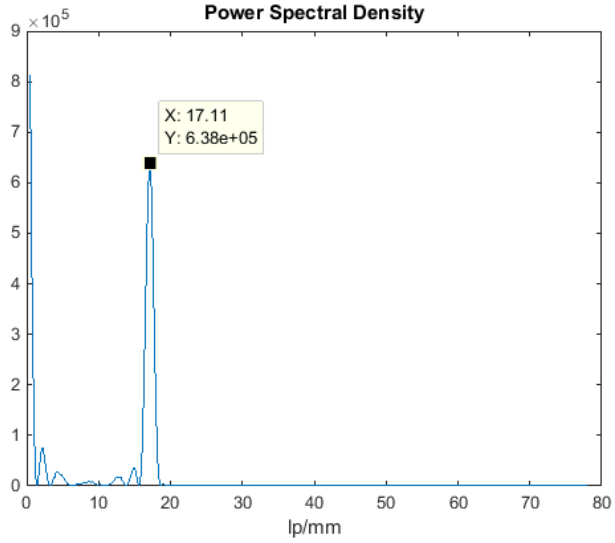

Figure 3-9. Power Spectral Density of the line profile in Figure 3-8. The local maximum at 17.11 indicates the lateral frequency of the pattern.

Figure 3-10 shows the calculated lateral frequencies using the PSD method for Z planes between 50 and 150mm at 5 mm intervals. As in the simulation, the pattern is expected to maintain a consistent lateral frequency through Z. An average of the calculated lateral frequencies through Z is indicated by the solid lines. The average frequencies of the AFA and the PSD methods for simulation and experimental data are compared in Table 3-2. The table indicates that the two methods have minimal differences in calculating the lateral frequency of the illumination pattern.

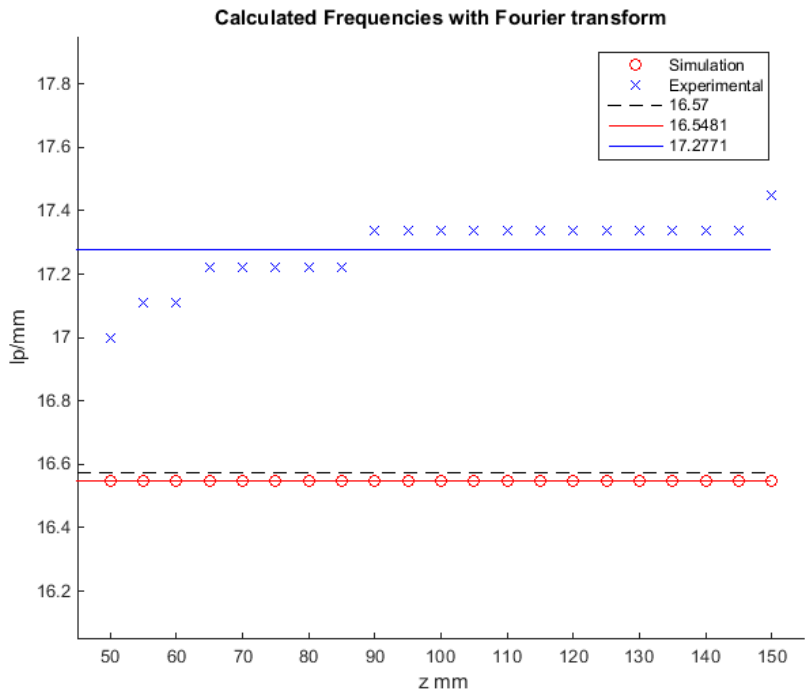

Figure 3-10. Calculated lateral frequencies using the PSD method through z at 5mm intervals of data acquired with a single slit configuration. We expect the pattern to maintain the same lateral frequency through z as indicated by the simulation.
Table 3-2. Comparison of AFA method and PSD method for calculating lateral frequency in simulation and experimental data for a single slit configuration.

	AFA	PSD	% Difference
Simulation	16.59 lp/mm	16.54 lp/mm	0.3%
Experiment	17.24 lp/mm	17.27 lp/mm	0.17%

The procedure of the PSD method is summarized as: For both AFA and PSD analysis, a line profile through X is acquired at each Z plane, and the PSD is calculated for each line profile. The PSD is analyzed for the local maximum, which indicates the lateral frequency of the pattern. The experimental pattern is expected to have one dominant frequency component in planes of resonance, with some low frequency noise. For the single slit case, the lateral pattern through Z is constant. For the double slit case and for configurations with more than two slits, the planes defined as planes of non-resonance may not show the lateral frequency as the most dominant local maximum in the PSD (low frequency noise may be more dominant). Therefore, the average lateral frequency should only be composed of calculated frequencies from the planes defined as planes of resonance.

3.H Power Spectral Density (PSD) Analysis of the Illumination Pattern's Axial Frequency

The AFA method is not suitable for determining the axial frequency in experimental data because a line profile representing the axial sinusoid is required. In the case of experimental data, the pattern can easily skew off the axial axis that the camera is panning through. In simulation data, the AFA method could work to find the axial frequency because of perfect symmetry, but a more robust method was investigated for

experimental data. The new method called the power spectral density (PSD) method was applied to double slit simulation data and could accurately measure the axial frequency.

The PSD method allows for calculating the axial frequency indirectly by analyzing each z plane, meaning the calculation is independent of the axial skew of the pattern with respect to the axial axis of the camera. The planes defined as planes of resonance are correlated to the strength of the PSD at the lateral frequency of the pattern. In other words, by plotting the PSD intensity for the expected lateral frequency as a function of z, the local maximums of that plot represent planes of resonance. Figure 3-11 shows the PSD Intensity plotted through Z for the calculated lateral frequency of 16.71 lp/mm in a simulated double slit configuration. Each local peak represents a plane of resonance where the calculated lateral frequency has the most power. The power increases as the irradiance signal widens with increasing Z. An axial frequency can be calculated from these peaks. Each peak represents an alternating maximum and minimum of the axial sinusoid; it takes two planes of resonance to complete an axial period. The calculated axial frequency from Figure 3-11 is 0.0217 lp/mm, while the theoretical axial frequency is 0.0221 lp/mm (which reflects a 1.81% error).

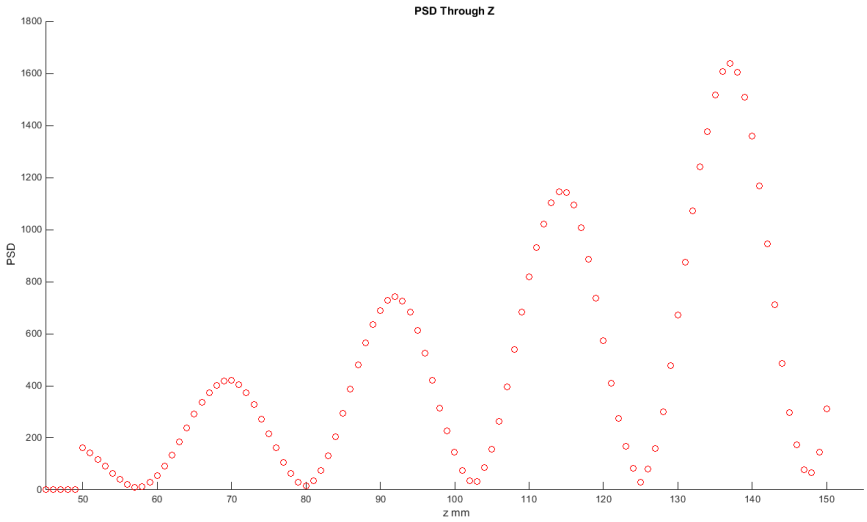

Figure 3-11. Power Spectral Density Intensity (watts/lp/mm) of the lateral frequency plotted through Z between 50 and 150 mm at 1mm intervals. Each local maximum represents a plane of resonance where the predicted lateral frequency is the strongest.

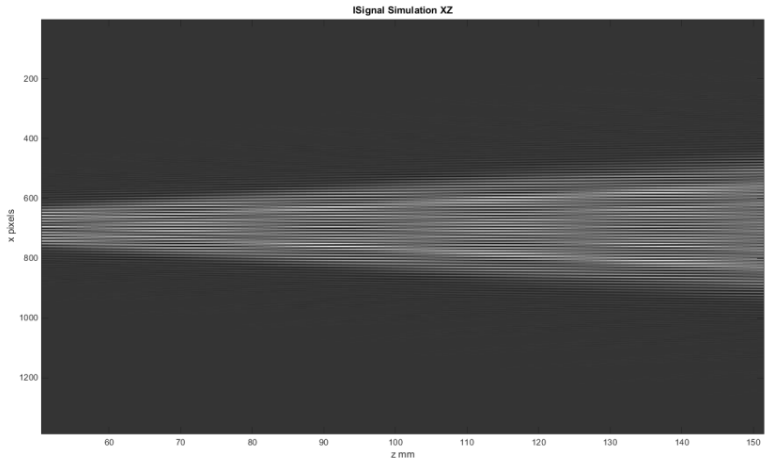

Figure 3-12. Simulated Irradiance Signal for reference. Notice the four resonant planes between 50-150 coincide with the peaks in Figure 3-11.

3.I Setup of ThorLabs BSC203 Stepper Controller with Micro-Manager Software

This section covers the initial setup of the two ThorLabs motorized linear translations stages LNR50S that are used to control the lateral position of the biprism and the axial position of the sample stage. Note that neither of these stages are required to take data on the illumination pattern, but both are vitally important to automating the SIM system for taking raw SIM data. The two motorized stages are connected to the BSC203 stepper controller via RS232 comms pins, which is then connected to a computer via a USB cable.

Before attempting anything with Micro-Manager, the user must first install the APT drivers for BSC203 which are included in the 32-bit ThorLabs APT or Kinesis Software packages. It is recommended to install the newer Kinesis Software found here [17] on ThorLabs website because APT is only made available for legacy purposes and will not be updated in the future. The user must install the 32-bit version for use with Micro-Manager because the 64-bit version of Micro-Manager does not support the 64-bit APT drivers. Once the Kinesis Software has been installed, verify the user can manipulate the stage with the software package to ensure that the APT drivers are working properly.

In order to use Micro-Manager to control the motorized stages, a virtual COM port must be created. Micro Manager is expecting to communicate with the stages through the RS232 connection directly, instead of through USB. [1] Open the windows device manager. [2] Click the USB serial bus controllers and right click the APT USB

Device for the Properties menu. [3] In the Advanced tab of the Properties menu, check ‘Load VCP’ box. [4] Restart the BSC203, and the device should appear under ‘Ports (COM & LPT)’. Take note of the COM port number (e.g. COM3). Please reference the ThorLab’s user manual for additional information [18].

After the above setup is completed, Micro-Manager is ready to auto detect the BSC203 stepper controller. Note that only the 32-bit version of Micro-Manager will recognize the APT drivers. Download an additional APT.dll file available here [19] on the Mirco-Manager website. Unzip the file and place it directly in your Micro-Manager folder. From here use the Micro-Manager’s Hardware Configuration Wizard to add the stage to a configuration setup. When adding the two stages, they are uniquely identified by the serial numbers located on the back of the BSC203 stepper controller. At the time of this writing, the biprism stage has been setup with the first serial number () and the sample stage has been setup with the second serial number (). For more information on the setup of the ThorLabs APT stage with Micro-Manager refer to the documentation on the website [20] and also the APT Communications Protocol on page 11 for information about the RS232 Interface [21].

Comment [CP(3]: Do you mean: “After the above set up is completed, the “?

CHAPTER 4 RESULTS

4.A Preliminary Simulation Results

The basic principle of the novel tunable illumination pattern is established by observing the qualitative effects of changing the position of the biprism. This preliminary simulation shows that the biprism position with respect to the focal length of the converging lens is linearly proportional to the frequency of the pattern as discussed in Chapter 3. Figure 4-1 shows the parameters used and how the biprism was positioned relative to the slit aperture and the focal lens. Refer to Figure 4-2 and Figure 4-3 to see a visual examination of how decreasing the biprism position from 190mm to 100mm causes a decrease in the lateral frequency of the pattern in simulation. This effect is predicted by Equation (3-1) in Section 3.A.

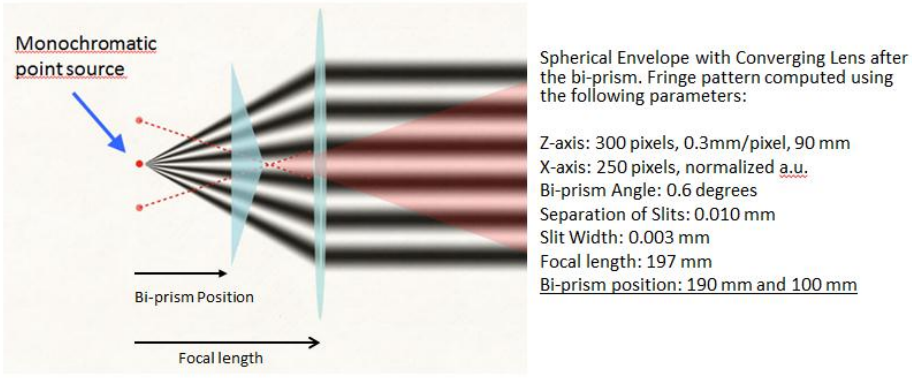

Figure 4-1. Novel Tunable Illumination setup. The slit(s) source plane and the converging lens remain in a fixed position at a distance equal to the focal length of the converging lens. The biprism is allowed to move between the two components, which allows the user to tune the lateral frequency of the pattern or alter the phase.

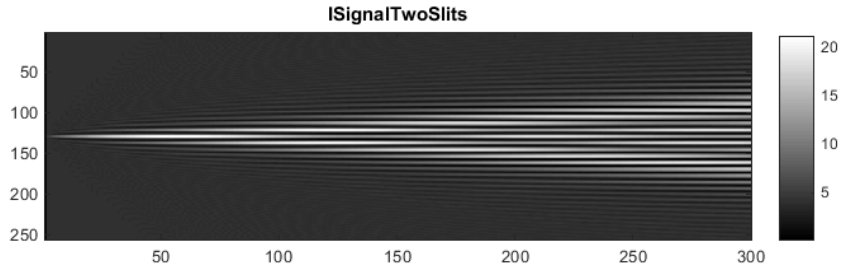

Figure 4-2. Irradiance signal for a two slit aperture where only one plane of resonance is shown. Notice the fringe pattern for a biprism position of 190mm.

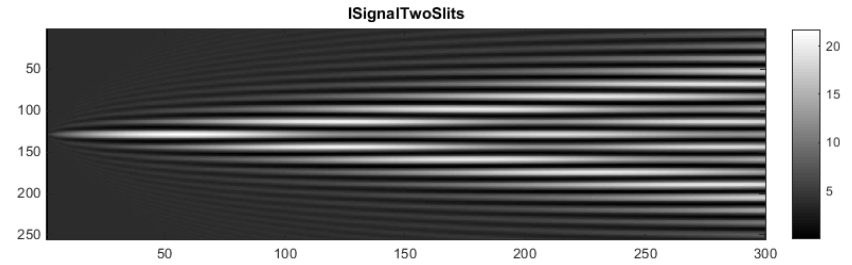

Figure 4-3. Irradiance signal for a two slit aperture where only one plane of resonance is shown. Notice the fringe pattern for a biprism position of 100mm.

Notice in Figure 4-2 and Figure 4-3 how the lateral frequency of the fringe pattern decreases by approximately one-half (lateral period doubles) as the biprism position is reduced by approximately one-half.

4.B Verify the Tunable Lateral Frequency of the Simulation

To test the accuracy of the simulation, the lateral frequency was measured and compared to the theoretical calculation. In this study, by keeping all the parameters constant and varying only the position of the biprism, it is shown that the simulation of the single slit setup produces a pattern that matches the theoretical values predicted by Equation (3-1). Refer to Table 4-1 for specific simulation parameters used in this study.

For instance, Figure 4-4 and Figure 4-5 below show the irradiance signal simulated with the biprism position equal to 50 mm and 130 mm, respectively, for a converging lens position of 150 mm. These data verify that the simulation matches expected theoretical calculations with 0.38% error and 0.07% error respectively. The percent error was calculated using the following: $\left|\frac{measured-theoretical}{theoretical}\right| * 100$. The vertical black bars indicate the window used to calculate the frequency as an average over multiple peaks using AFA.

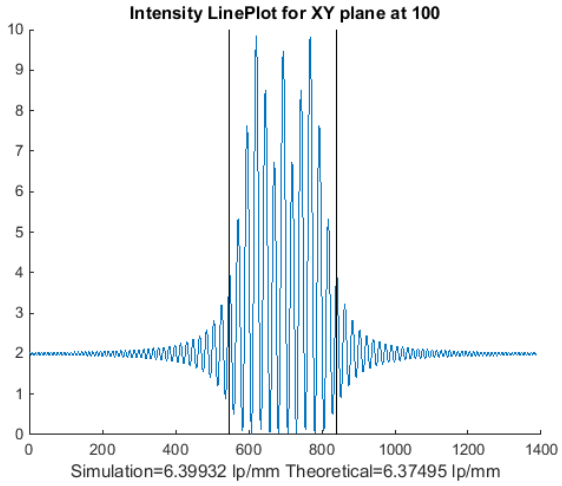

Figure 4-4. An intensity line profile through x at the 100th z-slice for a biprism position of 50mm. Single slit data. The measured lateral frequency for simulated data has a 0.38% error relative to the theoretical calculations.

Comment [CP(4]: expand to explain what this means. DO the same in all cases where you use this phrase below

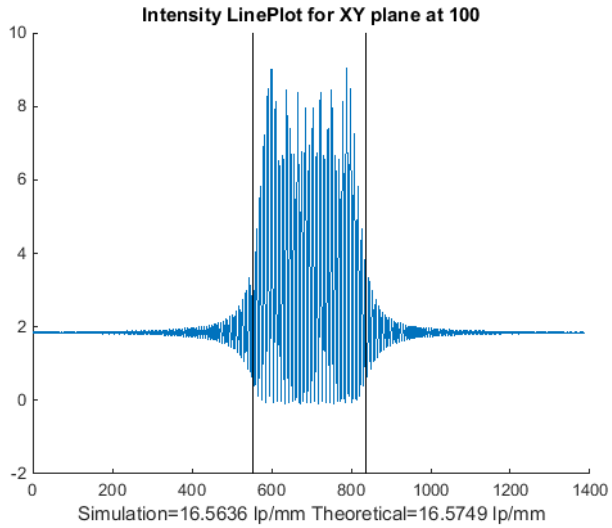

Figure 4-5. An intensity line profile through x and 100th z-slice for a biprism position of 130mm. Single slit data. The measured lateral frequency for simulated data has a 0.07% error relative to the theoretical calculations.

Table 4-1. Simulation parameters describing the source, biprism, slit, and converging lens. Note that the slit separation is irrelevant for the single slit setup.

λ	470 * 10^-6	mm wavelength of source
n	1.515	refractive index of the biprism
δ	$0.5 * \frac{\pi}{180}$	radians biprism angle
x_0	N/A	mm slit separation
Δ	0.0508	mm slit width
η	50 AND 130, respectively	mm position of biprism
f	150	mm focal length of converging lens

4.C Single Slit Data Comparison of Simulated and Experimental Data

Based on the same parameters from Section 4.B, the figures in this section compare the percent error of the calculated lateral frequency between simulated and experimental data for two different biprism positions at η = 50 mm and η = 130 mm. Figure 4-6 and Figure 4-7 show data for $\eta = 50$ mm while Figure 4-8 and Figure 4-9 show data for $\eta = 130$ mm. Refer to Table 4-1 for the specific parameters of the configuration.

The illumination configuration with a single slit is expected, based on theory, to maintain a structured intensity pattern with consistent lateral frequency through all Z planes, without any axial modulation. Figure 4-7 and Figure 4-9 show the theoretically calculated lateral frequencies are consistent with the averages calculated using simulated and experimental data.

Figure 4-6 and Figure 4-8 compare intensity profile from two specific axial planes (at 50 mm and 100 mm along the Z axis) taken from the simulated and experimental data, while Figure 4-7 and Figure 4-9 show the overall average of measured lateral frequencies from all Z planes captured through the illumination pattern. Figure 4-7 and Figure 4-9 show the simulation coincides with theoretical predictions, while the experimental data for η = 50 mm and η = 130 mm have an approximate 7.2% and 3.7% error, respectively compared to the simulated results in average measured lateral frequency.

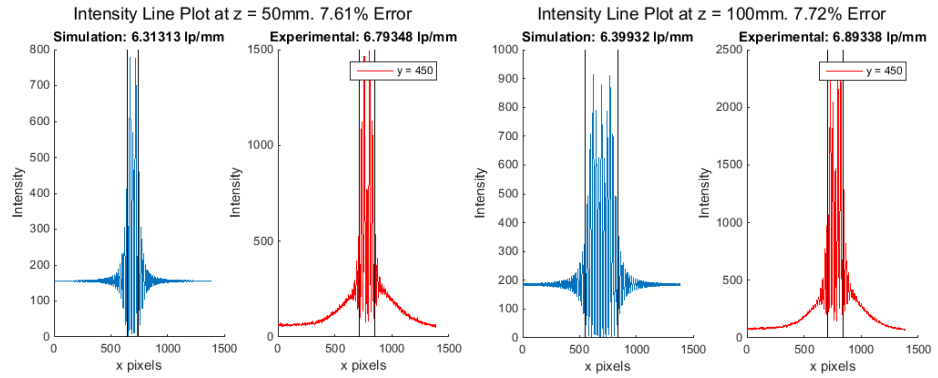

Figure 4-6. Intensity line profile comparison of two z planes at 50mm and 100mm for biprism positon of 50mm. Simulation in blue on the left and experimental data on the right in red. Approximately a 7.7% error in measured lateral frequency between simulated and experimental data at z = 50 mm and z = 100 mm.

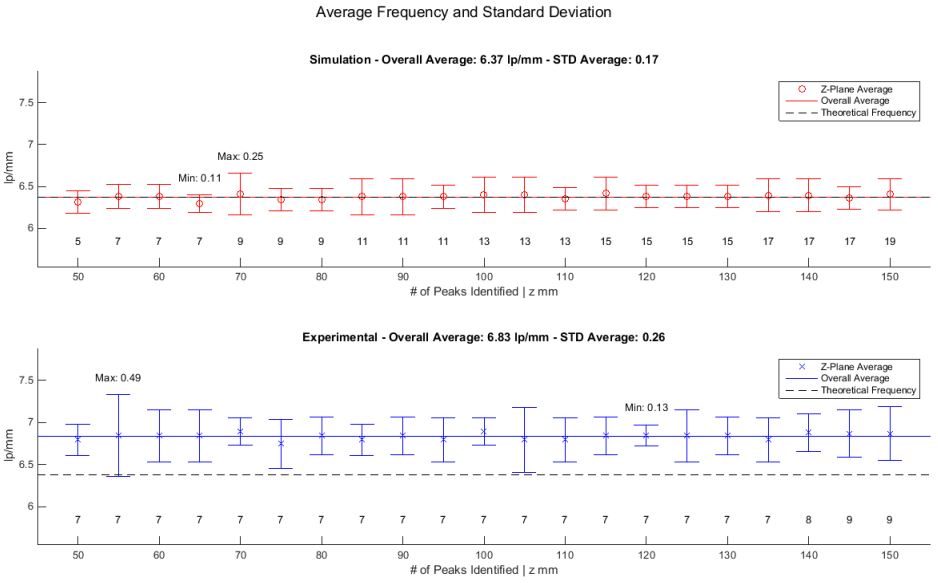

Figure 4-7. Average Frequency calculated by AFA in each z plane where data was collected (5mm increments from 50-150 mm). The data verifies the lateral frequency of the pattern is consistent through z. The experimental data has a relative ~7.2% error compared to the simulated data for the overall average lateral frequency for z planes 50-150 mm.

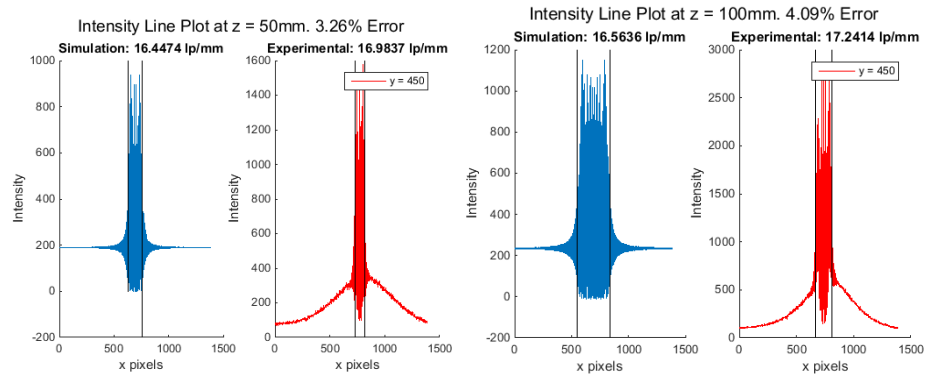

Figure 4-8. Intensity line profile comparison of two z planes at 50mm and 100mm for biprism positon of 130mm. Simulation in blue on the left and experimental data on the right in red. Approximately a 3.7% error in measured lateral frequency between simulated and experimental data at z = 50 mm and z = 100 mm.

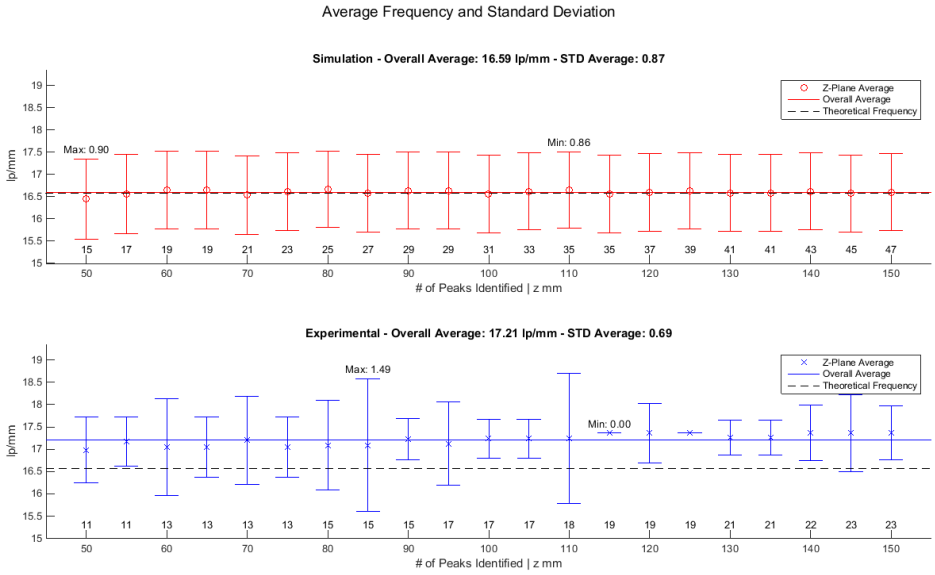

Figure 4-9. Average Frequency calculated by AFA in each z plane where data was collected (5mm increments from 50-150 mm). Verifies the lateral frequency of the pattern is consistent through z. The experimental data has a relative ~3.7% error compared to the simulated data for the overall average lateral frequency for z planes 50-150 mm.

4.D AFA Results for Double Slit Configuration in Simulation

The AFA method works accurately well in planes of resonance, but becomes erratic in planes of non-resonance. In non-resonant planes, the visibility of the pattern is reduced due to deconstructive interference. Therefore, a non-resonant line profile is not periodic. In Figure 4-10, results from the AFA method show a large standard deviation for the average frequency in the planes of non-resonance. When AFA was first developed, the idea of identifying non-resonant planes via these large variances was considered, but, as described in Section 4.E below, the PSD method emerged as a more accurate way of identifying planes of resonance. Therefore, the results of the AFA method are considered only for the lateral frequency of the illumination pattern in two slit configurations.

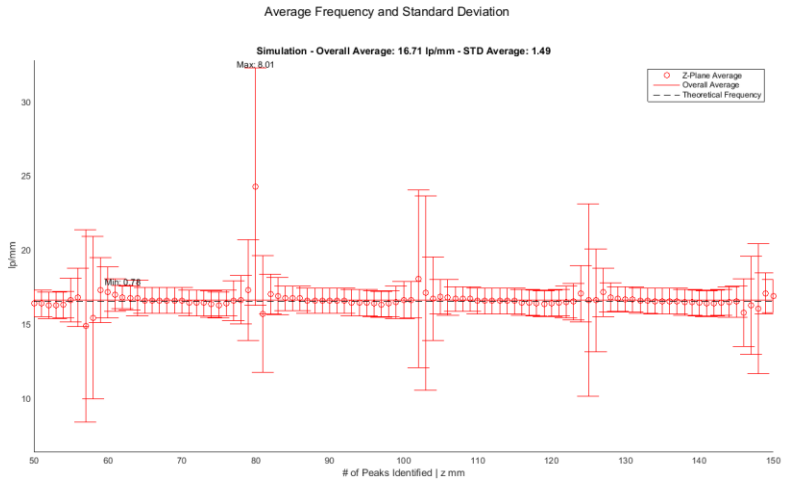

Figure 4-10. Average Frequency and Standard Deviation through z at 1mm intervals calculated using AFA. The non-resonant planes show a significantly higher standard deviation in the frequency than in the frequency in resonant planes.

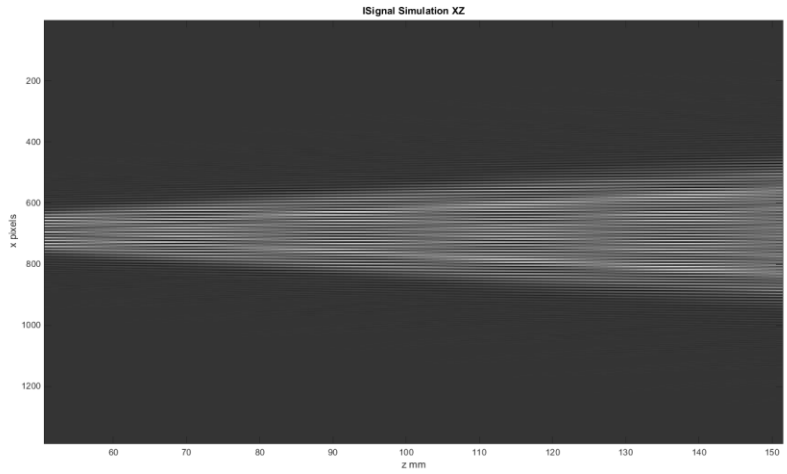

Figure 4-11. Simulated Irradiance Signal for reference to Figure 4-10. Notice the five non resonant planes at Z = 58, 80, 103, 125 and 149 coincide with AFA calculations.

4.E Double Slit Data Comparison of Simulation and Experimental

The PSD method is used to analyze the lateral and axial frequencies of double slit configuration data. Six total configurations are analyzed; three configurations with varying slit separations (0.200 mm, 0.300 mm, and 0.500 mm) and two biprism positions (50 mm and 120 mm) for each slit separation are presented in this section. The 0.200 mm separation and 50 mm biprism position is the best example out of the three 50 mm biprism configurations. The other two (0.300 mm and 0.500 mm) are highly distorted when the XZ image is stitched together (Figure 4-24 and Figure 4-34). When calculating the lateral frequency for each Z plane, the PSD method only searches for the most dominant frequency within 50% of the theoretical frequency. This filter is appropriate because the single slit data comparison reflects that the experimental data is well below 50% error relative to the theoretical frequency. Refer to Table 4-2 for the list of simulation and experimental parameters.

Table 4-2 Simulation and experimental parameters describing the wavelength of the source, biprism, slit, and converging lens.

λ	$470 * 10^{-6}$	mm wavelength of source
n	1.515	refractive index of the biprism
δ	$0.5 * \frac{\pi}{180}$	radians biprism angle
x_0	0.200, 0.300, and 0.500	mm slit separation
Δ	0.070	mm slit width
η	50 and 120	mm position of biprism
f	150	mm focal length of converging lens

Results acquired using the first configuration (0.200 mm separation and 50 mm biprism position) are presented in Figure 4-12 through Figure 4-16. In Figure 4-12, the low frequency outliers in the first 120mm are sample from a non-resonant plane and their value lowers the overall calculated average for the lateral frequency. By removing these outliers, the adjusted average for Z planes 150 through 250 is 5.88 lp/mm.

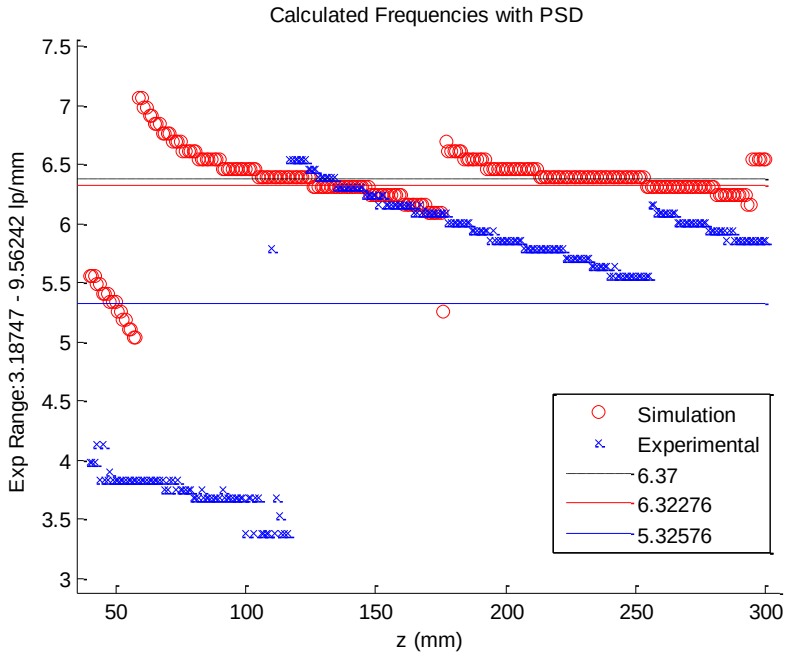

Figure 4-12. Calculated lateral frequencies through Z using PSD method. Experimental Frequency range filter 3.18-9.56 lp/mm.

Figure 4-13 and Figure 4-14 show the correlation between planes of resonance and the PSD intensity in the experimental data. Refer to Section 3.H.

Comment [CP(5]: please explain how the two figures show this correlation. Repeat in all cases where you use this phrase This is not self explanatory!

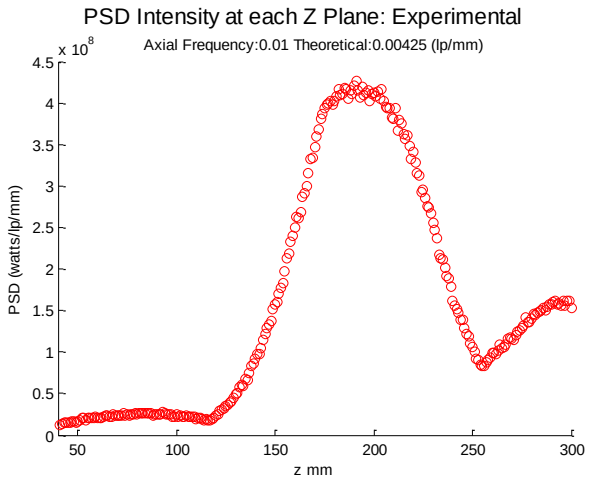

Figure 4-13. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 4-12.

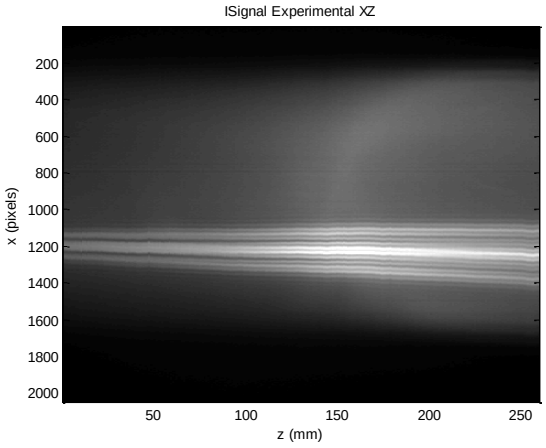

Figure 4-14. Experimental XZ image for 0.200 mm slit separation and 50 mm biprism position.

Figure 4-15 and Figure 4-16 show the correlation between planes of resonance and the PSD intensity in the simulation data.

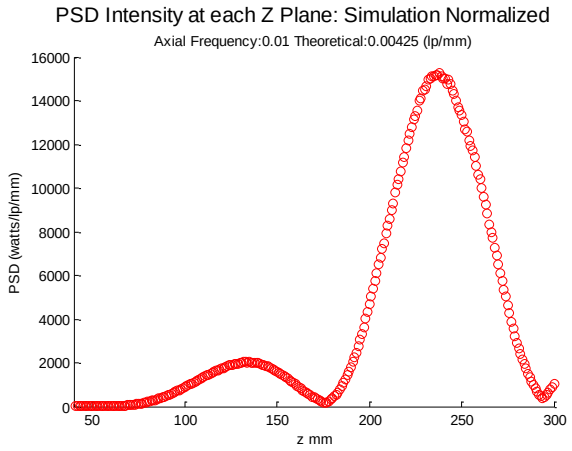

Figure 4-15. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 4-12.

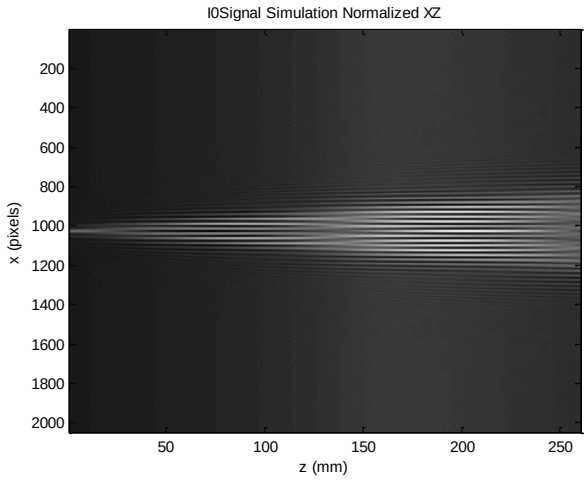

Figure 4-16. Simulation XZ image for 0.200 mm slit separation and 50 mm biprism position.

Results acquired using the second configuration (0.200 mm slit separation, 120 mm biprism position) are presented in Figure 4-17 through Figure 4-21. The outliers in Figure 4-17 correspond to planes of non-resonance and low PSD intensity indicated by Figure 4-18. The adjusted average for Z planes 150 through 250 is 14.0873 lp/mm.

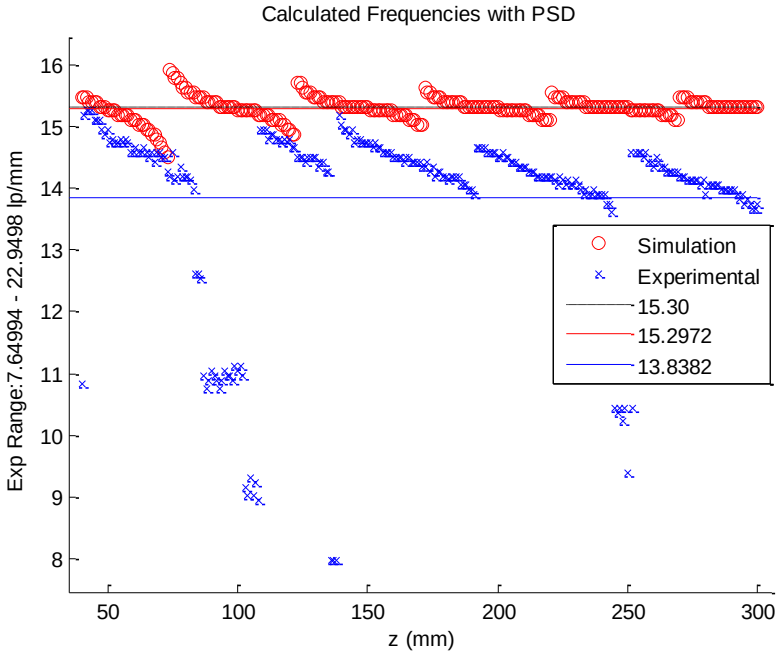

Figure 4-17. Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 7.65-22.95 lp/mm.

Figure 4-18 and Figure 4-19 show the correlation between planes of resonance and the PSD intensity in the experimental data.

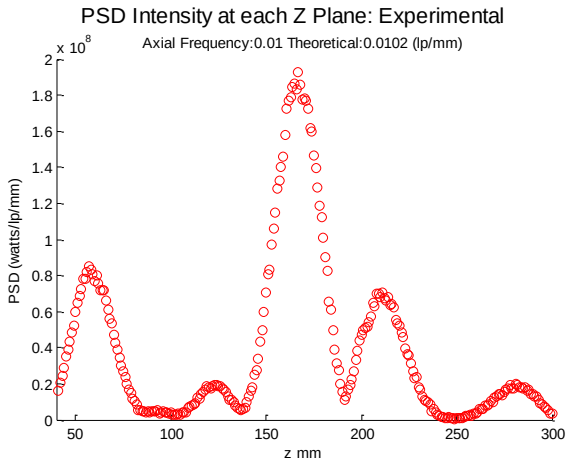

Figure 4-18. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 4-17.

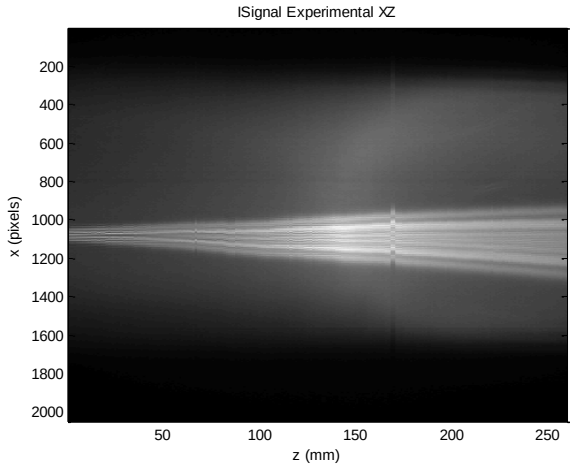

Figure 4-19. Experimental XZ image for 0.200 mm slit separation and 120 mm biprism position.

Figure 4-20 and Figure 4-21 show the correlation between planes of resonance and the PSD intensity in the simulation data.

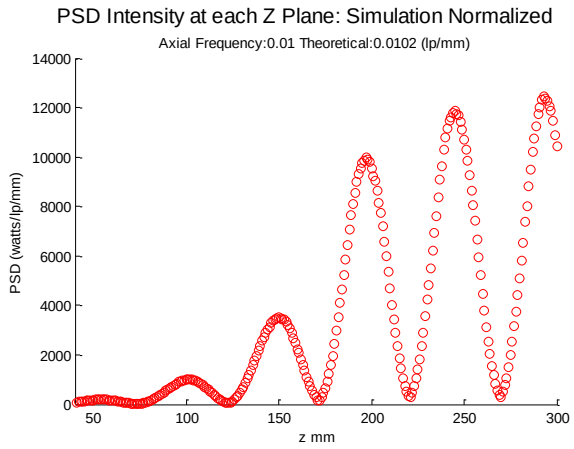

Figure 4-20. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 4-17.

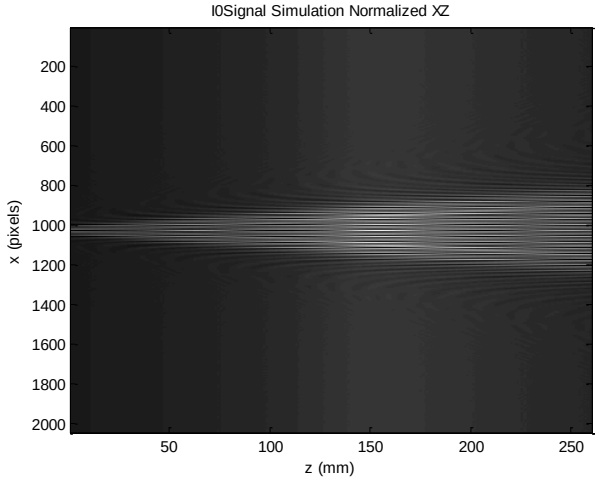

Figure 4-21. Simulation XZ image for 0.200 mm slit separation and 120 mm biprism position.

Results acquired using the third configuration (.300mm slit separation, 50 mm biprism position) are presented in Figure 4-22 through Figure 4-26. The adjusted average for Z planes 150 through 250 is 5.9159 lp/mm.

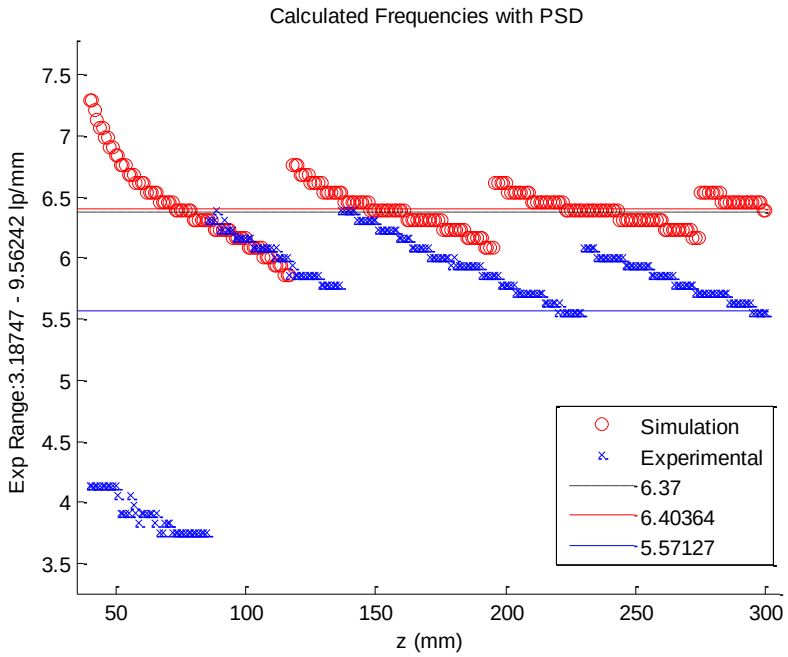

Figure 4-22. Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 3.19-9.56 lp/mm.

Figure 4-23 and Figure 4-24 show the correlation between planes of resonance and the PSD intensity in the experimental data.

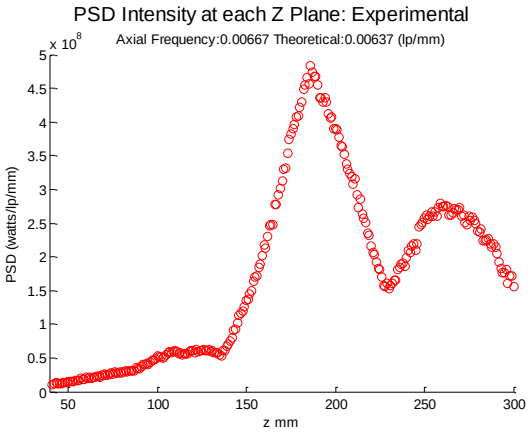

Figure 4-23. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 4-22.

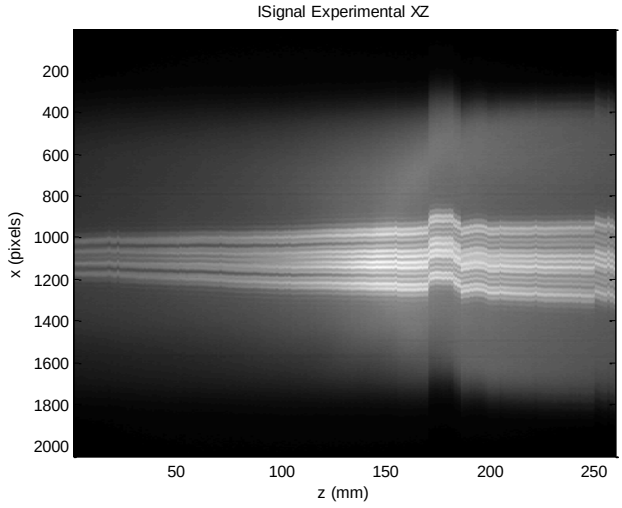

Figure 4-24. Experimental XZ image for 0.300 mm slit separation and 50 mm biprism position.

Figure 4-25 and Figure 4-26 show the correlation between planes of resonance and the PSD intensity in the simulation data.

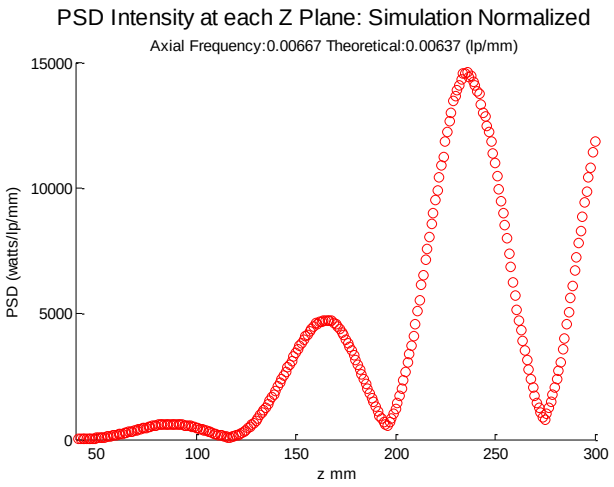

Figure 4-25. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 4-22.

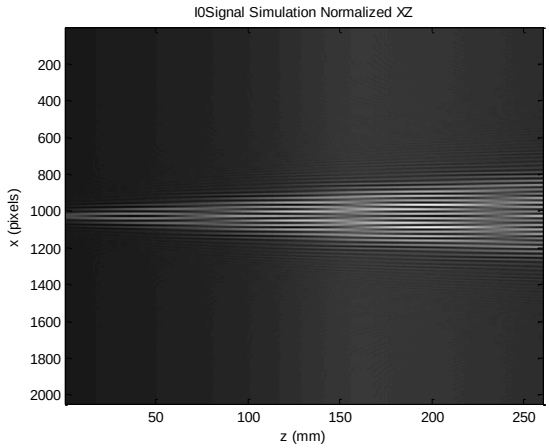

Figure 4-26. Simulation XZ image for 0.300 mm slit separation and 50 mm biprism position.

Results acquired using the fourth configuration (.300 mm slit separation, 120 mm biprism positon) are presented in Figure 4-27 through Figure 4-31. The adjusted average for Z planes 150 through 250 is 14.5912 lp/mm.

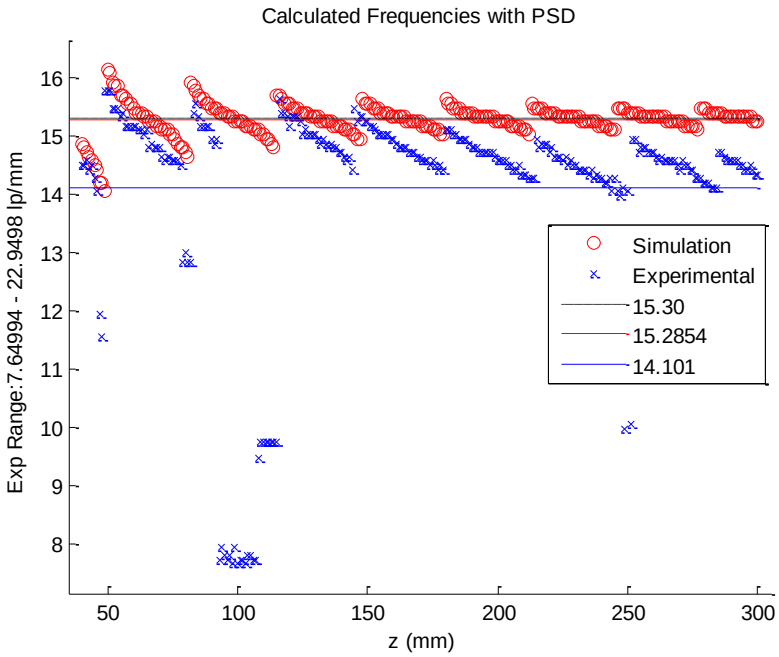

Figure 4-27. Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 7.65-22.95 lp/mm.

Figure 4-28 and Figure 4-29 show the correlation between planes of resonance and the PSD intensity in the experimental data.

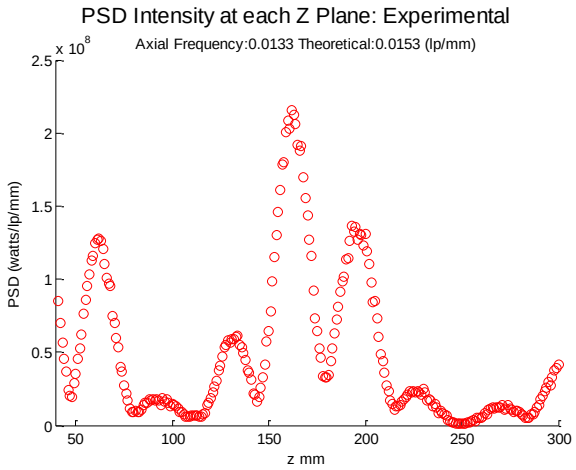

Figure 4-28. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 4-27.

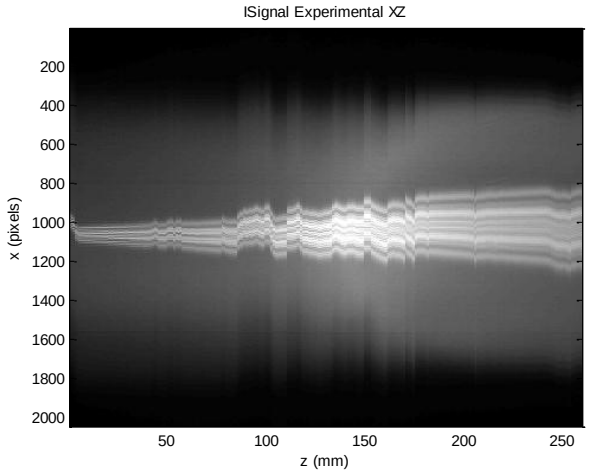

Figure 4-29. Experimental XZ image for 0.300 mm slit separation and 120 mm biprism position.

Figure 4-30 and Figure 4-31 show the correlation between planes of resonance and the PSD intensity in the simulation data.

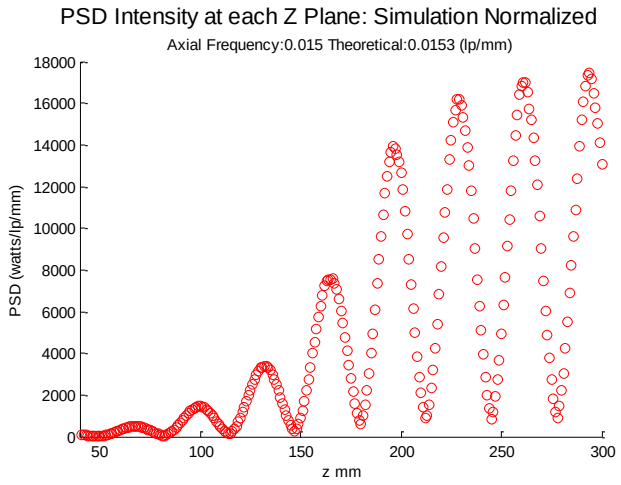

Figure 4-30. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 4-27.

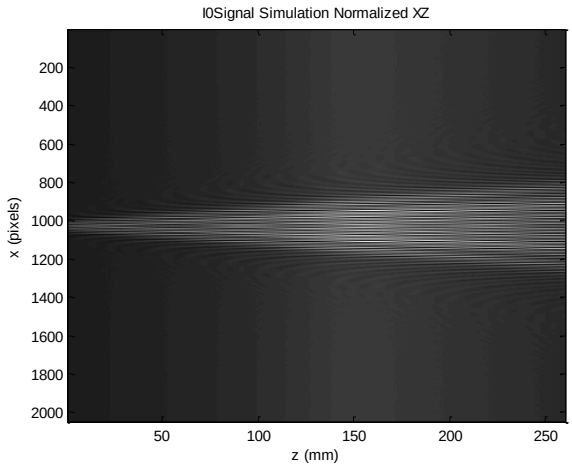

Figure 4-31. Simulation XZ image for 0.300 mm slit separation and 120 mm biprism position.

Results acquired using the the fifth configuration (0.500 mm slit separation, 50 mm biprism position) are presented in Figure 4-32 through Figure 4-36. The adjusted average for Z planes 150 through 250 is 5.9449 lp/mm.

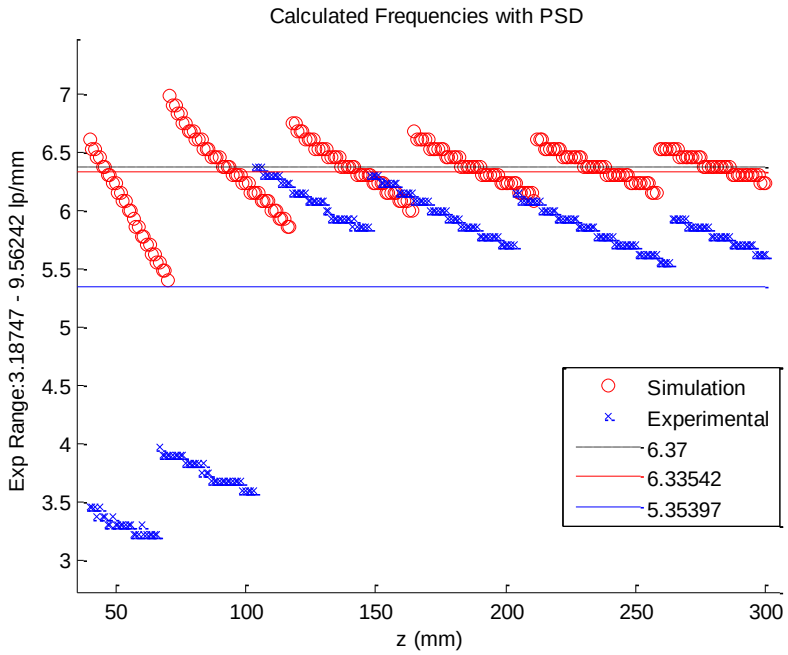

Figure 4-32 Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 3.19-9.56 lp/mm.

Figure 4-33 and Figure 4-34 show the correlation between planes of resonance and the PSD intensity in the experimental data.

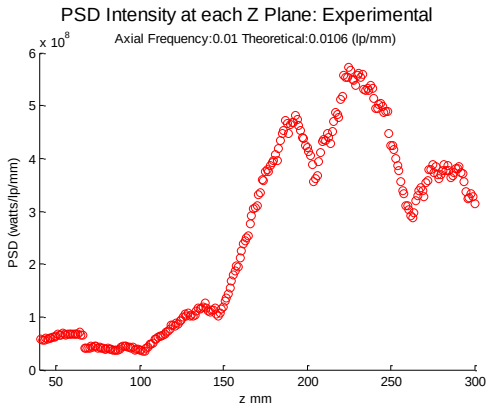

Figure 4-33. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 4-32.

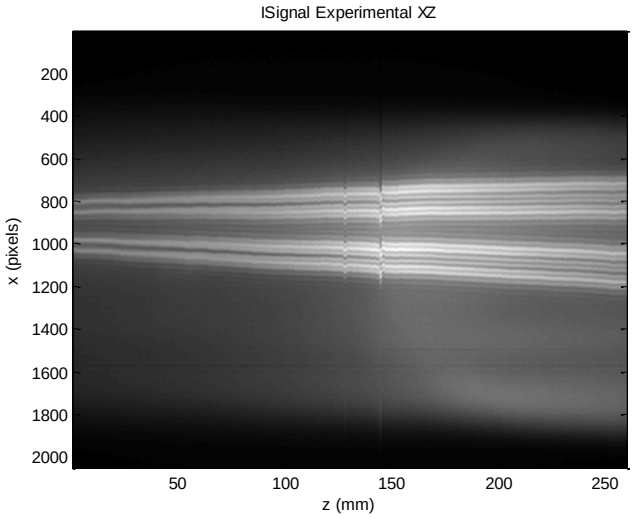

Figure 4-34. Experimental XZ image for 0.500 mm slit separation and 50 mm biprism position

Figure 4-35 and Figure 4-36 show the correlation between planes of resonance and the PSD intensity in the simulation data.

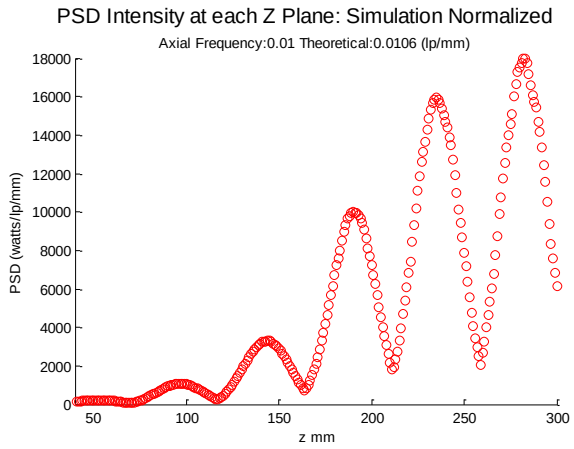

Figure 4-35. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 4-32.

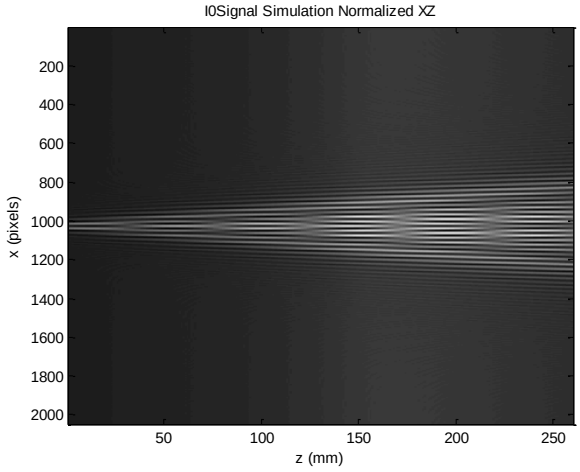

Figure 4-36. Simulation XZ image for 0.500 mm slit separation and 50 mm biprism position.

Results acquired using the sixth configuration (0.500 mm slit separation, 120 mm biprism position) are presented in Figure 4-37 through Figure 4-41. The adjusted average for Z planes 150 through 250 is 14.7169 lp/mm.

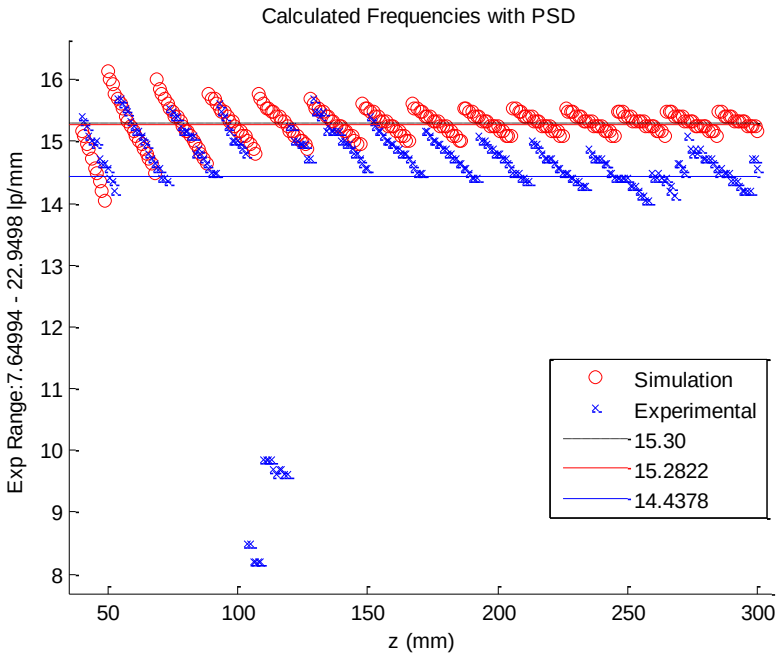

Figure 4-37 Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 7.65-22.95 lp/mm.

Figure 4-38 and Figure 4-39 show the correlation between planes of resonance and the PSD intensity in the experimental data.

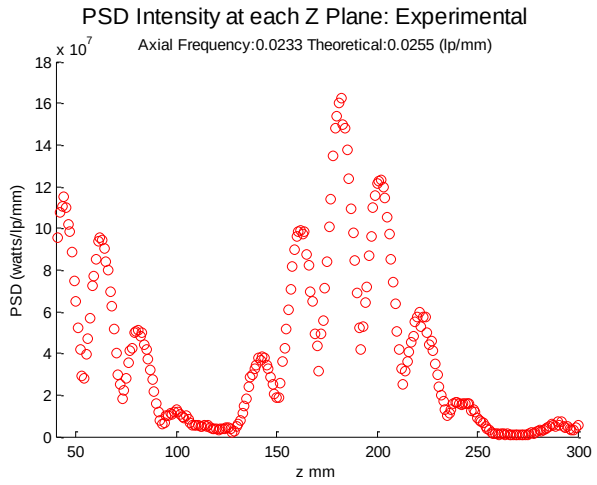

Figure 4-38. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 4-37.

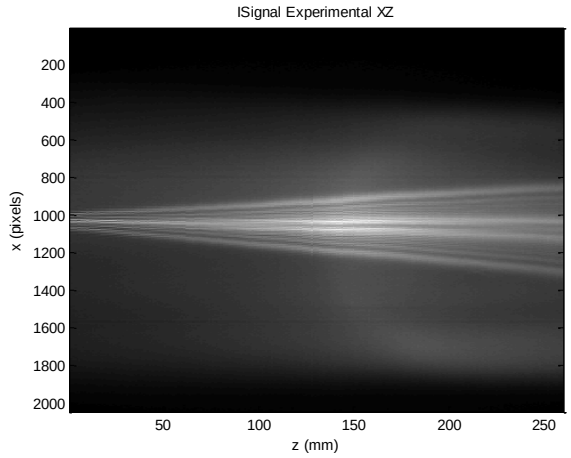

Figure 4-39. Experimental XZ image for 0.500 mm slit separation and 120 mm biprism position.

Figure 4-40 and Figure 4-41 show the correlation between planes of resonance and the PSD intensity in the simulation data.

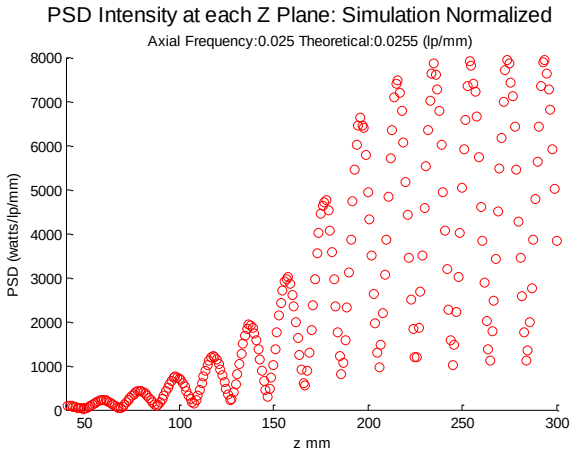

Figure 4-40. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 4-37.

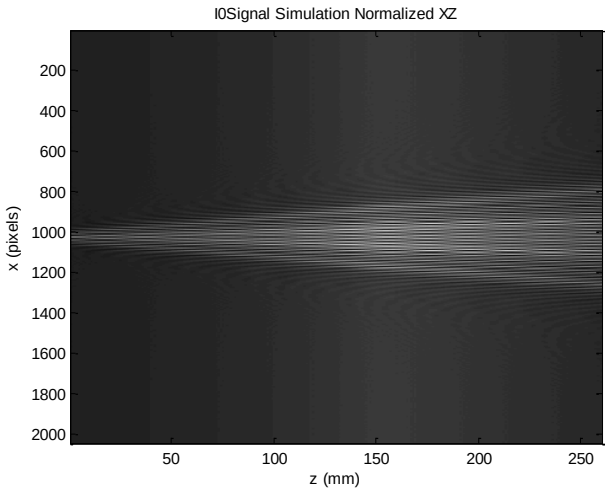

Figure 4-41. Simulation XZ image for 0.500 mm slit separation and 120 mm biprism position.

Table 4-3 summarizes the percent error between the experimental data and the simulation data for the measured average lateral frequency for each of the six double slit configurations.

Table 4-3 Comparison of measured experimental and simulation lateral frequencies for various two slit configurations using the PSD method.

Slit Separation, Biprism Position (mm)	Simulation Lateral Frequency Average (lp/mm)	Experimental Lateral Frequency Adjusted Average (lp/mm)	Percent Error
0.200, 50	6.32	5.88	6.99 %
0.200, 120	15.30	14.04	8.21 %
0.300, 50	6.40	5.92	7.61 %
0.300, 120	15.28	14.59	4.54 %
0.500, 50	6.34	5.94	6.16%
0.500, 120	15.28	14.72	3.70 %

Table 4-4 summarizes the percent error between the experimental data and the theoretical calculations for the measured axial frequency for each of the six double slit configurations. First, The PSD method was applied to the Z range of the whole data set (40-300 mm) to calculate the measured axial frequency. Second, the PSD method was applied to an adjusted Z range of the data set to calculate the measured axial frequency with adjusted Z. Note that the first configuration's axial period (0.200 mm slit separation, 50 mm biprism position).

Table 4-4 Measured axial frequencies without adjustment and with adjusted Z range.

$x_0, \eta\ (mm)$	Measured (lp/mm)	Theoretical (lp/mm)	% Error	Measured with Adjusted Z (lp/mm)	Adjusted Z Range (mm)	Adjusted % Error
0.200, 50	0.0100	0.0043	194 %	0.0100	150:299	194 %
0.200, 120	0.0100	0.0102	1.96 %	0.0100	100:249	1.96 %
0.300, 50	0.0067	0.0064	4.69 %	0.0067	150:299	4.69 %
0.300, 120	0.0133	0.0153	13.1 %	0.0150	125:224	1.96 %
0.500, 50	0.0100	0.0106	5.66 %	0.0100	150:299	5.66 %
0.500, 120	0.0233	0.0255	8.63 %	0.0250	125:224	1.96 %

CHAPTER 5 CONCLUSIONS

5.A Single Slit Configuration

The single slit configuration was designed and tested to compare the lateral frequency of experimental data with simulation and theoretical values. The AFA and PSD methods were used to measure the lateral frequencies at varying positions of the biprism between 50 and 150 mm of the illumination pattern at intervals of 5 mm. The data collected shows that AFA and PSD methods agree with each other within 0.3% difference in simulation and 0.17% difference in experimental data. When using either of these methods to calculate the lateral frequency, the simulation matches theoretical predictions. In the case of experimental data, the average calculated lateral frequency may slightly vary from simulation. This can be explained by variances in the experimental optical setup, such as the positions of the slit aperture, the biprism, or the converging lens. The experimental setup could be calibrated by measuring the offset of the biprism that tunes the lateral frequency of the experimental illumination pattern to the simulated data to achieve better precision in lateral frequency. Furthermore, it is shown that the experimental irradiance signal maintains a consistent lateral frequency through Z as predicted by Equation (3-1). This leads to the conclusion that the experimental data is accurate, but with some degree of missing precision due to the exact positions of the optics along the Z axis.

5.B Double Slit Configuration

The double slit configuration was investigated via the PSD method to compare measured lateral and axial frequencies of experimental data to simulated data and theoretical values. The AFA method is unsuitable for measuring the axial frequency in

experimental data because of issues in image registration and the axial skew of the illumination pattern. These are not problems in simulated data where the pattern is perfectly symmetrical along the propagation axis. The PSD method shows similar results for the measured lateral frequencies that were reported for the single slit data, as expected. In the experimental data, the PSD method is able to accurately calculate the axial frequency of the pattern with a high degree of precision (<5% error) when compared to the theoretical calculations of the axial frequency. The PSD method for measuring the axial frequency in simulated data does match theoretical predictions. Although some of the experimental data for the double slit configuration is not correctly aligned in Z, it has been shown in simulated and experimental data that the PSD method is capable of determining the axial frequency despite the axial skew of the data in a procedure that is independent of the axial axis of the pattern. In simulated data, the PSD method results show an accurate measurement of the theoretical axial frequency of the illumination pattern.

5.C AFA vs PSD Analysis

The AFA method is capable of accurately measuring the lateral frequency of illumination pattern given that the line profile is an accurate sample of the lateral sinusoid. This is relatively easy when the camera setup is perpendicular to the lateral sinusoid in the pattern. However, using AFA to calculate an axial frequency is difficult because the camera is scanning in a direction parallel to the axial sinusoid. In experiment, the axial sinusoid may skew in a different direction than the camera's direction of motion, which makes it difficult to select an accurate line profile that represents the axial sinusoid without a method for registering the images. Registering the images for perfect

alignment of the axial sinusoid is further complicated by the non-resonant planes in the illumination pattern because these planes have low contrast. The low contrast makes it difficult to reference points in the lateral sinusoid for registration with corresponding points in planes of resonance.

The PSD method is shown to perform similarly to the AFA method in single slit configurations when calculating the lateral frequency. This confirms that both methods are accurate in measuring the lateral frequency of any structured illumination, given that the line profile accurately samples the lateral sinusoid. In contrast to the AFA method, the PSD method is capable of identifying planes of resonance by the power spectrum strength of the calculated lateral frequency. In this way, the axial sinusoid can be identified independent of the axial skew of experimental data. By measuring the power spectral intensity of the calculated lateral frequency of the pattern through Z, a sinusoid pattern emerges that identifies planes of resonances at its peaks.

5.D Future Work

I would like more precise control of the optics for better alignment. I had some difficulty in correcting the alignment height of the optical elements. Some of the posts were too short or too large to be put in the correct position; I believe this can be fixed by reconfiguring the height of the system. A height should be chosen in the middle of the smallest post, so that all posts have some degree of freedom. In the z direction, the rail is split in three sections because the biprism must be mounted on a separate rail for the stepper motor in the *x* direction. In its current state, the three rails are aligned as best as possible, but there may be some error here. As for the x direction, many of the posts are

mounted on an extra foot that allows small corrections in *x*, which is adequate. The main concern of the setup should be focused on better positioning in the *z* and *y* directions.

The illumination code that generates the pattern currently works for the I0 signal, but does not take into account the width of the slit(s). The width of the slit(s) causes a slight fading of the pattern in *z*. Considering that the data analyzed in this thesis is before 300mm in *z*, this function does not significantly affect the visibility of the pattern, nor does it affect the lateral or axial frequency of the pattern. The research in this thesis was focused on analyzing the lateral and axial frequencies of the pattern, so the function related to the slit width has little impact on the conclusions stated above. However, for a more comprehensive simulation this function should be reviewed.

The data collected in Chapter 4 show an envelope of visibility where the brightness of the pattern increases at the back focal plane of the converging lens and fades after the back focal plane. Preliminary observations suggest that this phenomenon may be due to the simulation not accounting for the height of the biprism.

CHAPTER 6 REFERENCES

[1] (2015). *The Diffraction Barrier in Optical Microscopy*. Available: http://www.microscopyu.com/articles/superresolution/diffractionbarrier.html.

[2] M. Born and E. Wolf, *Principles of Optics.* Cambridge University Press, 1997.

[3] *Structured Illumination Microscopy*. Available: http://www.uv.es/imaging3/lineas/SIM.html.

[4] F. Macias-Garza, A. C. Bovik, K. R. Diller, S. J. Aggarwal and J. K. Aggarwal, "The missing cone problem and low-pass distortion in optical serial sectioning microscopy," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference On*, 1988, pp. 890-893 vol.2.

[5] M. G. L. Gustafsson, L. Shao, P. M. Carlton, C. J. R. Wang, I. N. Golubovskaya, W. Z. Cande, D. A. Agard and J. W. Sedat, "Three-Dimensional Resolution Doubling in Wide-Field Fluorescence Microscopy by Structured Illumination," *Biophys. J.*, vol. 94, pp. 4957-4970, 2008.

[6] J. Frohn, H. Knapp and A. Stemmer, "Three-dimensional resolution enhancement in fluorescence microscopy by harmonic excitation," *Opt. Lett.*, vol. 26, pp. 828-830, 2001.

[7] A. Doblas, G. Saavedra, M. Martinez-Corral, J. C. Barreiro, E. Sanchez-Ortiga and A. Llavador, "Axial resonance of periodic patterns by using a Fresnel biprism," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 30, pp. 140-148, 2013.

[8] M. Gustafsson, "Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy," *Journal of Microscopy-Oxford*, vol. 198, pp. 82-87, 2000.

[9] D. Karadaglić and T. Wilson, "Image formation in structured illumination wide-field fluorescence microscopy," *Micron*, vol. 39, pp. 808-818, 2008.

[10] *Education in Microscopy and Digital Imaging*. Available: http://zeiss-campus.magnet.fsu.edu/articles/superresolution/supersim.html.

[11] H. Shabani, E. Sanchez-Ortiga and C. Preza, "Investigating the performance of reconstruction methods used in structured illumination microscopy as a function of the illumination pattern's modulation frequency," *Proc. SPIE 9713, Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XXIII*, vol. 9713, pp. 9713-9714, 2016.

[12] ***Zeiss ELYRA S1 (SR-SIM) Super Resolution Microscope***. Available: http://bmc.uga.edu/equipment/zeiss-elyra-s1-sr-sim-super-resolution-microscope/.

[13] R. W. Wood, *Physical Optics.* New York: Dover Publications, 1988.

[14] (2016). *Adjustable Mechanical Slits*. Available: http://www.mansionschools.com/apertures-diffraction-elements-and-filters-laboratory-equipment-double-slits-on-glass-plate-u22014-l.html?channelid=GoogleAdwords&gclid=CKTlvr35tsoCFQEdaQodMzwPZA.

[15] (2016). *Double Slits On Glass Plate*. Available: http://www.mansionschools.com/apertures-diffraction-elements-and-filters-laboratory-equipment-double-slits-on-glass-plate-u22014-l.html?channelid=GoogleAdwords&gclid=CKTlvr35tsoCFQEdaQodMzwPZA.

[16] (2016). *FFT For Spectral Analysis*. Available: http://www.mathworks.com/help/matlab/examples/fft-for-spectral-analysis.html.

[17] (2015). ***Motion Control Software***. Available: https://www.thorlabs.com/software_pages/ViewSoftwarePage.cfm?Code=Motion_Control&viewtab=1.

[18] (2015). *BSC202 and BSC203 Benchtop Stepper Motor Controller User Guide*. Available: https://www.thorlabs.com/thorcat/24700/BSC203-Manual.pdf.

[19] (2014). *File:APT.zip*. Available: https://micro-manager.org/wiki/File:APT.zip.

[20] (2014). ***ThorlabsAPTStage***. Available: https://micro-manager.org/wiki/ThorlabsAPTStage.

[21] (2015). *Thorlabs APT Controllers Host-Controller Communications Protocol*. Available: http://www.thorlabs.de/software/apt/APT_Communications_Protocol_Rev_15.pdf.

CHAPTER 7 APPENDICIES

7.A Illumination Code (TunableIlluminationPattern_SingleAndDoubleSlits.m)

```
%% GRATING - FRESNEL BIPRISM - CONVERGING LENS
% Optics Model
% 1. A One and Two Slit grating with slit seperation x0
% 2. An axially tunable fresnel biprism with position eta (1-f)
% 3. A converging lens with focal distance f.
% 4. The irradiance pattern along Z

% script calculates the pattern in an XZ plane
% Z-slices are independently calculated
% script autosaves.mat file to current folder

%Dependencies:
%findfrequency.m
%fresnelc.m and fresnels.m (included in matlab 2014a)
%signal_toolbox and symbolic_toolbox

%% SETUP
clear
lambda  = 470 * 10^-6;   %mm wavelength of source
n       = 1.515;         %Refractive Index of biprism
delta   = 0.5*pi/180;    %radians biprism angle
x0      = .300;          %mm slit seperation
DELTA   = 0.005;         %mm slit width
eta     = 150;           %mm position of biprism
f       = 150;           %mm focal length of converging lens

u0 = (n-1)*tan(delta)/lambda;
FresnelComp = @(t) fresnelc(t) + 1i*fresnels(t);
ML = @(z) z./f;
xTheoreticalFrequency = (eta/f)*(2*u0) %from equation (2) Doblas.
x is normalized by (2*pi)/p. theroretical calculation
%p = (lambda*f) / (2*eta*(n-1)*tan(delta)) % lateral period
zTheoreticalPeriod = (f/x0)*1/(2*u0)*(f/eta); %theroretical
calculation
zTheoreticalFrequency = 1/zTheoreticalPeriod; %theroretical
calculation

%Irradiance Signal
I0 = @(z,x) abs(exp(-
1i.*pi./(lambda.*z).*(x+lambda.*z.*u0).^2)...
        .* ((1+1i)/2 +
FresnelComp(sqrt(2./(lambda.*z)).*(x+lambda.*z.*u0)))...
        + exp(-1i.*pi./(lambda.*z).*(-x+lambda.*z.*u0).^2) ...
        .* ((1+1i)./2 - FresnelComp(sqrt(2./(lambda.*z)).*(x-
lambda.*z.*u0)))).^2;

%X and Z Scaling
mmPerPixel = .0064; %lateral mm/pixel AxioVisionCamera=0.0064
xdim = 1388; %discrete lateral size
```

```
      xscale = (eta/f) * (xdim/2) * mmPerPixel; % (scaling ratio for
biprism position) * (inverse scaling factor for Doblas normalization) *
(mm/pixel)

      zmmperpixel = 1;
      zdim = 300; %discrete axial size
      zscale = (eta/f) * 2 * zmmperpixel; % mm/pixel

      % Create XZ Domain Space and U Frequency space
      z = linspace(1,zscale*zdim,zdim);
      x = linspace(-1*xscale,1*xscale,xdim); %x is normalized from -1
to 1
      % Frequency Space u
      u = linspace(-0.5/mmPerPixel,0.5/mmPerPixel,xdim);
      % [u,~] = freqspace(length(x)); Matlab's freqspace function
unused

      %setup grid
      z = repmat(z,size(x,2),1);
      x = repmat(x(:),1,size(z,2));
      u = repmat(u(:),1,size(z,2));

      %The slit width transfer function is a rect in space domain (sinc
in
      %frequency domain)
      T = (DELTA*z/f) ; %  * f/eta *.01
      FTSlitWidthTransferFunction = 1/mmPerPixel*T.* sinc(T.*u );
%1/mmPerPixel*T.* %broken 2/8/16,
      % setting a small DELTA (<.005) will make this function = 1, thus
not
      % affecting the simulation
      % FTSlitWidthTransferFunction =
(DELTA*ML(z)).*(xdim/2).*sinc(DELTA*ML(z).*u); alternate

      %% ONE SLIT
      %CALCULATION
      I0SignalOneSlit = I0(z,x);

      ISignalOneSlit =
real(ifft(fft(I0SignalOneSlit).*FTSlitWidthTransferFunction));

      %% TWO SLITS
      %SETUP
      I0TwoSlits = @(z,x) I0(z,x+x0.*z/(2*f)) + I0(z,x-x0.*z/(2*f));

      %Calculation
      I0SignalTwoSlits = I0TwoSlits(z,x);

      ISignalTwoSlits =
real(ifft(fft(I0SignalTwoSlits).*FTSlitWidthTransferFunction));

      %% Find Frequency (calculate average peak to peak distance)
      % function findfrequency(data,mm/pixel,frequencyCeiling)
      % Axial calculation for axial frequency
```

```
      [axialFrequency, axialWindow] =
findfrequency(ISignalTwoSlits(xdim/2,:),zmmperpixel,50);
      % Lateral calculation for line pairs per mm
      zslice = axialWindow(2); %look at a z slice in resonant plane
      [lateralFrequency, lateralWindow] =
findfrequency(ISignalTwoSlits(:,zslice),mmPerPixel,50);

      axialToLateralRatio = axialFrequency/lateralFrequency

      %% One Slit Figures
      % Lateral calculation for line pairs per mm
      zslice = 130; %look at a z slice in a resonant plane
      [lateralFrequency, lateralWindow] =
findfrequency(ISignalOneSlit(:,zslice),mmPerPixel,50);

      %Intensity Line Plot figures
      figure,
      hold on
      plot(ISignalOneSlit(:,zslice))
      title(sprintf('Intensity LinePlot for XY plane at %g',zslice));
      line([lateralWindow(1)
lateralWindow(1)],get(gca,'YLim'),'Color','k','LineWidth',1); %draw
window boundaries
      line([lateralWindow(2)
lateralWindow(2)],get(gca,'YLim'),'Color','k','LineWidth',1);
      hold off
      xlabel(sprintf('Simulation=%g lp/mm Theoretical=%g
lp/mm',lateralFrequency,xTheoreticalFrequency));

      %Irradiance Signal
      figure,
      imagesc(ISignalOneSlit);
      title('ISignalOneSlit');
      colorbar; colormap(gray);
      figure,
      imagesc(ISignalOneSlit(:,zslice)');
      title(sprintf('ISignalOneSlit %g',zslice));
      colormap(gray);

      figure,
      imagesc(real(FTSlitWidthTransferFunction));
      colorbar; colormap(gray);
      title('FTSlitWidthTransferFunction');

      %% Two Slit Figures
      %Intensity Line Plot figures
      figure,
      hold on
      plot(ISignalTwoSlits(:,zslice))
      title(sprintf('Intensity LinePlot for XY plane at %g',zslice));
      line([lateralWindow(1)
lateralWindow(1)],get(gca,'YLim'),'Color','k','LineWidth',1); %draw
window boundaries
      line([lateralWindow(2)
lateralWindow(2)],get(gca,'YLim'),'Color','k','LineWidth',1);
      hold off
```

```
      xlabel(sprintf('Simulation=%g lp/mm Theoretical=%g
lp/mm',lateralFrequency,xTheoreticalFrequency));

      figure,
      hold on
      plot(ISignalTwoSlits(xdim/2,:))
      title('Intensity LinePlot for XZ plane at middle horizontal')
      line([axialWindow(1)
axialWindow(1)],get(gca,'YLim'),'Color','k','LineWidth',1); %draw
window boundaries
      line([axialWindow(2)
axialWindow(2)],get(gca,'YLim'),'Color','k','LineWidth',1);
      hold off
      xlabel(sprintf('Simulation=%g lp/mm Theoretical=%g
lp/mm',axialFrequency,zTheoreticalFrequency));

      %Figures
      figure,
      imagesc(I0SignalTwoSlits);
      title('ISignalTwoSlits');
      colorbar; colormap(gray);
      figure,
      imagesc(ISignalTwoSlits(:,zslice)');
      title(sprintf('ISignalTwoSlits %g',zslice));
      colormap(gray);

      %% One and Two Slits Virtual Sources Analysis
      % a = 2*eta*(n-1)*tan(delta); %distance of virtual slits
      % scaledDistanceOfVirtualSlitsOnBackApertureOfObjectiveLens = a *
160/150; %mm
      %
      % figure
      % hold on
      % plot(0,0,'bo')
      % plot(0,0.5*a,'bx',0,-0.5*a,'bx')
      % hold off
      % ylim([-1,1])
      % title('One Slit Virtual Sources')
      % xlabel(sprintf('a = %g mm',a))
      % ylabel('x (mm)')
      % legend('Real Slit', 'Virtual Slits')
      %
      % figure
      % hold on
      % plot(0,0.5*x0,'bo',0,-0.5*x0,'ro')
      % plot(0,0.5*x0 + 0.5*a,'bx',0,-0.5*x0 + -0.5*a,'rx')
      % plot(0,0.5*x0 - 0.5*a,'bx',0,-0.5*x0 - -0.5*a,'rx')
      % hold off
      % ylim([-2,2])
      % title('Two Slit Virtual Sources')
      % xlabel(sprintf('a = %g mm | Slit Seperation = %g mm',a,x0));
      % ylabel('x (mm)')
      % legend('Real Slits','', 'Virtual Slits','')

      %% Auto save .mat
```

```
      %
"PRISM_WAVELENGTH_SLITSEPARATION_BIPRISMPOSITION_FOCALLENGTHOFCONVERGIN
GLENS_SLITWIDTH_XZDIMENSIONS_MMPERPIXEL_ZMMPERPIXEL.MAT"
      biprismAngle = 180 - delta*360/pi;
      wavelength =  lambda*10^6;
      slitWidth = DELTA*10^3;
      filename = sprintf('FBP2020G-
%g_%gnm_%.3fmmslits_%gEta_%gFocal_%gumSlitWidth_%gx%g_%gx_%gzmmperpixel
.mat',biprismAngle,wavelength,x0,eta,f,slitWidth,xdim,zdim,mmPerPixel,z
mmperpixel)
      save(filename);
```

7.B AFA Code (findfrequency.m)

```
      function [lateralFrequency,window,standardDeviation,peaks] =
findfrequency(data,mmPerPixel,frequencyCeiling)
          %Expecting data to be 2D XY Intensity of fringe pattern

          %Tune these two parameters if the frequency is being
overestimated for
          %experimenatal
          %Use peaks that are 40% of the max intensity (create a
window)
          minIntensity = max(data)*.4;
          %minimum peak to peak distance laterally in pixels
          minPixels = floor(1/(frequencyCeiling*mmPerPixel));

          [~,lateralPeakLocations] =
findpeaks(data,'MinPeakHeight',minIntensity,'MinPeakDist',minPixels);
          %Calculate the pixel difference between each peak in the
window multiply by
          %the mm/pixel to obtain the distance form peak to peak. Take
the
          %average of the periods.
          lateralfringePeriod =
mean(diff(lateralPeakLocations).*mmPerPixel);
          lateralFrequency = 1/lateralfringePeriod;

          if isnan(lateralFrequency)
              error( 'Can not compute lateral frequency in the window,
try PSD method')
          end

          %Report standard deviation
          standardDeviation =
std(1./(diff(lateralPeakLocations).*mmPerPixel));
          standardError =
standardDeviation/sqrt(length(lateralPeakLocations));

          peaks = length(lateralPeakLocations);
          %Store the boundaries of the window
          window(1) = lateralPeakLocations(1);
          window(2) = lateralPeakLocations(end);
      end
```

7.C Compare Experimental and Simulation with AFA

(CompareExperimentalandSimulation.m)

```
      %% Compare Experimental and Simulation Data
      %Dependencies:
      % loadDataDirectories.m
      % findlateralfrequency.m
      % findYAxisLimits.m
      % findMaxMinStandardDeviation.m

      %script generates figures and lineplots of the average
intensities at
      %specific zslices. the experimentalDataDirectory should contain
XY images
      %that are labeled as "'zslice'.tif". example: "100.tif" for the
zslice at
      %100mm. the simulationDataDirectory should contain the .mat file
saved from
      %"SphericalEnvelopeWithConvergingLens.m". Specify which
Iraddiance Signal
      %you are working with when loading the simulation data

      %Refer to loadDataDirectories.m for list of configurations

      clear;close all;
      %% Initialize Parameters
      %Choose Configuration index
      index = 17;
      %Choose yslice for experimental data to compare to simulation.
(there is only 1 yslice for simulation)
      yslice = 250;
      %Parameters to find lateral frequency
      mmPerPixel = 0.0064; %lateral mm/pixel. expecting same value for
simulation and experiment

      %Choose findFrequency.m parameters
      %We calculate an estimated frequency ceiling to throw out the
      %outliers/noise. Set to a very large percent error to collect all
data
      %points (i.e. 100.0 or 10000%). The 1.0 setting sets the
      %estimated Frequency Ceiling to double of the corresponding
zslice simulation
      %frequency. Therefore, any peaks within 50% of the estimated
period are thrown out.
      estimatedpercentError = 1.0; %1=100% %best practice to
overestimate. applies to percent error of lp/mm
      simulationFrequencyCeiling = 50; %lp/mm %because our simulation
data is noise free, this can remain relatively high for low frequency
patterns
```

```
      lateralFrequency.simulation.window = zeros(2,1); %initialize
window
      lateralFrequency.experimental.window = zeros(2,1); %initialize
window
      %% Set Directories and Specfic Directory Params
      %set directory for experimental data and simulation data
      directory = loadDataDirectories();

      %% Load, Normalize, and Calculate Frequency fo the Data
      %Load Simulation Data, specify the iraddiance signal variable
      load(directory(index).simulation,
directory(index).simulationData, 'xTheoreticalFrequency');
      ISignal.simulation.data = eval(directory(index).simulationData);
% simulated irradiance signal
      theoreticalLateralFrequency = xTheoreticalFrequency;
      %some of the early simulation data does not contain
xTheoreticalFrequency
      %and is added in loadDataDirectories()
      if ~exist('xTheoreticalFrequency')
          theoreticalLateralFrequency =
directory(index).xTheoreticalFrequency;
      end

      %Load Experimental Data. expects X-Y plane with vertical fringes.
      %naming Convention-- zslice should correspond to the zslices
taken in
      %experimental data. example: 100.tif for the 100mm zslice.
      for zslice =
directory(index).zsliceStart:directory(index).zsliceInterval:directory(
index).zsliceEnd;
          %convert to double precision
          ISignal.experimental.data(:,:,zslice) =
double(imread(strcat(directory(index).experimental,
sprintf('%g.tif',zslice))));

          experimentalNormalizationfactor =
sum(mean(ISignal.experimental.data(:,:,zslice)));% take average along y
dim and sum

          %normalize the simulation data to experimental.
          simulationNormalizationFactor =
sum(ISignal.simulation.data(:,zslice)'); %take the transpose to align
fringes vertically and sum.
          ISignal.simulation.normalized(:,zslice) =
ISignal.simulation.data(:,zslice) *
(experimentalNormalizationfactor/simulationNormalizationFactor);

          %Calculate frequency

[lateralFrequency.simulation.data(zslice),lateralFrequency.simulation.w
indow(:,zslice),standardDeviation.simulation.data(zslice),lateralFreque
ncy.simulation.peaks(zslice)] =
findfrequency(ISignal.simulation.normalized(:,zslice),mmPerPixel,simula
tionFrequencyCeiling);
```

```
          experimentalFrequencyCeiling =
lateralFrequency.simulation.data(zslice) +
2*estimatedpercentError*lateralFrequency.simulation.data(zslice);

[lateralFrequency.experimental.data(zslice),lateralFrequency.experiment
al.window(:,zslice),standardDeviation.experimental.data(zslice),lateral
Frequency.experimental.peaks(zslice)] =
findfrequency(ISignal.experimental.data(yslice,:,zslice),mmPerPixel,exp
erimentalFrequencyCeiling);

      end

      percentError = (lateralFrequency.experimental.data-
lateralFrequency.simulation.data)./lateralFrequency.simulation.data.*10
0;

      %% Plot Figures
      for zslice =
directory(index).zsliceStart:50:directory(index).zsliceEnd;
          %Irradiance Signal
          figure,
          subplot(1,2,1)
          imagesc(ISignal.simulation.normalized(:,zslice)');
          ylabel('y a.u.');
          xlabel('x pixels');
          title('Simulation');
          colormap(gray);
          subplot(1,2,2)
          imagesc(ISignal.experimental.data(:,:,zslice));
          ylabel('y pixels');
          xlabel('x pixels');
          title('Experimental');
          suptitle(sprintf('Irradiance Signal at z = %gmm',zslice))

          %Intensity LinePlots
          figure,
          subplot(1,2,1)
          hold on
          plot(ISignal.simulation.normalized(:,zslice)); %draw
intensity
          line([lateralFrequency.simulation.window(1,zslice)
lateralFrequency.simulation.window(1,zslice)],get(gca,'YLim'),'Color','
k','LineWidth',1); %draw window boundaries
          line([lateralFrequency.simulation.window(2,zslice)
lateralFrequency.simulation.window(2,zslice)],get(gca,'YLim'),'Color','
k','LineWidth',1);
          hold off
          title(sprintf('Simulation: %g
lp/mm',lateralFrequency.simulation.data(zslice)));
          ylabel('Intensity');
          xlabel('x pixels');
          subplot(1,2,2)
          hold on
          plot(ISignal.experimental.data(yslice,:,zslice),'r');%draw
intensity
```

```
          line([lateralFrequency.experimental.window(1,zslice)
lateralFrequency.experimental.window(1,zslice)],get(gca,'YLim'),'Color'
,'k','LineWidth',1);%draw window boundaries
          line([lateralFrequency.experimental.window(2,zslice)
lateralFrequency.experimental.window(2,zslice)],get(gca,'YLim'),'Color'
,'k','LineWidth',1);
          hold off
          title(sprintf('Experimental: %g
lp/mm',lateralFrequency.experimental.data(zslice)));
          ylabel('Intensity');
          xlabel('x pixels');
          legend(sprintf('y = %g',yslice));
          suptitle(sprintf('Intensity Line Plot at z = %gmm. %.2f%%
Error',zslice,percentError(zslice)))

      end

      %Stitch experimental data into on XZ image
      for zslice =
directory(index).zsliceStart:directory(index).zsliceInterval:directory(
index).zsliceEnd;
          XZImage.experimental(:,zslice) =
ISignal.experimental.data(yslice,:,zslice);
      end
      %Experimental XZ image
      figure,
      imagesc(XZImage.experimental)
      ylabel('x pixels')
      xlabel('z mm')
      title('ISignal Experimental XZ');
      colormap(gray);

      %Simulation XZ image
      XZImage.simulation = ISignal.simulation.data(:,1:300);
      figure,
      imagesc(XZImage.simulation)
      ylabel('x pixels')
      xlabel('z mm')
      title('ISignal Simulation XZ');
      colormap(gray);

      %Plot Average Frequencies at each Z plane
      %Calculate limits for y axis
      ylimit =
findYAxisLimits(lateralFrequency.simulation.data,standardDeviation.simu
lation.data,lateralFrequency.experimental.data,standardDeviation.experi
mental.data);
      %Calculate Overall Averages
      overallAverage.simulation =
sum(lateralFrequency.simulation.data)./sum(lateralFrequency.simulation.
data~=0);
      overallAverage.experimental =
sum(lateralFrequency.experimental.data)./sum(lateralFrequency.experimen
tal.data~=0);
      %Calculate Standard Deviation Averages
```

```
      standardDeviation.simulation.average =
sum(standardDeviation.simulation.data)./sum(standardDeviation.simulatio
n.data~=0);
      standardDeviation.experimental.average =
sum(standardDeviation.experimental.data)./sum(standardDeviation.experim
ental.data~=0);
      %Calculate Max and Min Standard Deviation variables for use with
text()
      [standardDeviation.simulation.text] =
findMaxMinStandardDeviation(standardDeviation.simulation.data,lateralFr
equency.simulation.data);
      [standardDeviation.experimental.text] =
findMaxMinStandardDeviation(standardDeviation.experimental.data,lateral
Frequency.experimental.data);

      %Plot Average Frequencies Figure in two subplots
      figure,
      subplot(2,1,1) %simulation subplot
      hold on
      errorbar(lateralFrequency.simulation.data,standardDeviation.simul
ation.data,'ro')
      line(get(gca,'XLim'),[overallAverage.simulation
overallAverage.simulation],'Color','r','LineWidth',1); %plot overall
average
      line(get(gca,'XLim'),[theoreticalLateralFrequency
theoreticalLateralFrequency],'Color','k','LineWidth',1,'LineStyle','--
'); %plot Theoretical Lateral Frequency
      text(standardDeviation.simulation.text.max.x,standardDeviation.si
mulation.text.max.y,standardDeviation.simulation.text.max.string,'Horiz
ontalAlignment','center');
      text(standardDeviation.simulation.text.min.x,standardDeviation.si
mulation.text.min.y,standardDeviation.simulation.text.min.string,'Horiz
ontalAlignment','center');
      for zslice =
directory(index).zsliceStart:directory(index).zsliceInterval:directory(
index).zsliceEnd

text(zslice,ylimit(1)+.3,sprintf('%g',lateralFrequency.simulation.peaks
(zslice)),'HorizontalAlignment','center');
      end
      hold off
      set(gca, 'Ticklength', [0 0])
      ylim(ylimit)
      xlim([directory(index).zsliceStart-5
directory(index).zsliceEnd+5])
      ylabel('lp/mm')
      xlabel('# of Peaks Identified | z mm')
      legend('Z-Plane Average','Overall Average', 'Theoretical
Frequency')
      title(sprintf('Simulation - Overall Average: %.2f lp/mm - STD
Average:
%.2f',overallAverage.simulation,standardDeviation.simulation.average))
      subplot(2,1,2) %experimental subplot
      hold on
      errorbar(lateralFrequency.experimental.data,standardDeviation.exp
erimental.data,'bx')
```

```
      line(get(gca,'XLim'),[overallAverage.experimental
overallAverage.experimental],'Color','b','LineWidth',1); %plot overall
average
      line(get(gca,'XLim'),[theoreticalLateralFrequency
theoreticalLateralFrequency],'Color','k','LineWidth',1,'LineStyle','--
'); %plot Theoretical Lateral Frequency
      text(standardDeviation.experimental.text.max.x,standardDeviation.
experimental.text.max.y,standardDeviation.experimental.text.max.string,
'HorizontalAlignment','center');
      text(standardDeviation.experimental.text.min.x,standardDeviation.
experimental.text.min.y,standardDeviation.experimental.text.min.string,
'HorizontalAlignment','center');
      for zslice =
directory(index).zsliceStart:directory(index).zsliceInterval:directory(
index).zsliceEnd

text(zslice,ylimit(1)+.3,sprintf('%g',lateralFrequency.experimental.pea
ks(zslice)),'HorizontalAlignment','center');
      end
      hold off
      set(gca, 'Ticklength', [0 0])
      ylim(ylimit)
      xlim([directory(index).zsliceStart-5
directory(index).zsliceEnd+5])
      ylabel('lp/mm')
      xlabel('# of Peaks Identified | z mm')
      legend('Z-Plane Average','Overall Average', 'Theoretical
Frequency')
      title(sprintf('Experimental - Overall Average: %.2f lp/mm - STD
Average:
%.2f',overallAverage.experimental,standardDeviation.experimental.averag
e))
      suptitle(sprintf('Average Frequency and Standard Deviation'))


      %% Old Figures
      % figure,
      % plot(lateralFrequency.experimental.peaks,'d')
      %
ylim([min(lateralFrequency.experimental.peaks(lateralFrequency.experime
ntal.peaks~=0))-2 max(lateralFrequency.experimental.peaks)+2])
      % xlim([45 155])
      % ylabel('# of Peaks')
      % xlabel('z mm')

      % figure
      % subplot(1,2,1)
      % plot(XZExpimage(1388/2,:))
      % title('Intensity LinePlot for XZ plane at middle horizontal')
      % subplot(1,2,2)
      % plot(XZSimImage(1388/2,:))
      % title('Intensity LinePlot for XZ plane at middle horizontal')

      % xaxis = 1:length(ISignal.simulation.data);
      % figure,
```

```
      %
plot(xaxis,storeAverage(:,50),'r',xaxis,storeAverage(:,100),'k',xaxis,s
toreAverage(:,150),'b')
      % title(sprintf('Experimental Average Intensity LinePlot'));
      % legend('z=50','z=100','z=150');
      %
      % figure,
      %
plot(xaxis,storeLine(:,50),'r',xaxis,storeLine(:,100),'k',xaxis,storeLi
ne(:,150),'b')
      % title(sprintf('Experimental Intensity LinePlot for
y=%g',yslice));
      % legend('z=50','z=100','z=150');
```

7.D Plot Automation for Comparing Experimental and Simulation data

(findYAxisLimits.m)

```
      function ylimit =
findYAxisLimits(lateralSimulationFrequency,standardDeviationSimulation,
lateralExperimentalFrequency,standardDeviationExperimental )
      % For Use with Average Frequency and Standard Deviation Plot
      % calculates the y axis limits based on the height of the largest
standard
      % deviation line from both simulation and experimental, so that
the axes
      % are the same for comparison.
      ylimit(1) =
min([min(lateralSimulationFrequency(lateralSimulationFrequency~=0))-
max(standardDeviationSimulation)-.5
min(lateralExperimentalFrequency(lateralExperimentalFrequency~=0))-
max(standardDeviationExperimental)-.5 ]);
      ylimit(2) =
max([max(standardDeviationSimulation)+max(lateralSimulationFrequency)+.
5
max(standardDeviationExperimental)+max(lateralExperimentalFrequency)+.5
]);
      end
```

(findMaxMinStandardDeviation.m)

```
      function [ data ] = findMaxMinStandardDeviation(
standardDeviationExperimental, lateralExperimentalFrequency )
      %   FINDMAXMINSTANDARDDEVIATION
      %   Created by Chris Taylor
      %   Specified for matlab text() function. Used to highlight the
max and min
      %   of standard deviation in the Average Frequnecy and Stadard
Deviation
      %   Plot. Expects the input data to be a vector of standard
deviations by
      %   zslices
          standardDeviationMax = max(standardDeviationExperimental);
```

```
          data.max.x = find(standardDeviationMax ==
standardDeviationExperimental,1);
          data.max.y =
lateralExperimentalFrequency(data.max.x)+standardDeviationMax+.2;
          data.max.string = sprintf('Max: %.2f',standardDeviationMax);

          standardDeviationMin =
min(standardDeviationExperimental(standardDeviationExperimental~=0));
          data.min.x = find(standardDeviationMin ==
standardDeviationExperimental,1);
          data.min.y =
lateralExperimentalFrequency(data.min.x)+standardDeviationMin+.2;
          data.min.string = sprintf('Min: %.2f',standardDeviationMin);

      end
```

7.E PSD Code (findFrequencyWithFourier.m)

```
      function [ frequency, peakPSD ] = findFrequencyWithFourier( data,
mmPerPixel, frequencyRange )
      %Created By Chris Taylor Jan 26 2016
      %Expects data to be an even line profile

          nSamples = length(data);

          dataFourier = fft(data,nSamples);
          Pyy = dataFourier.*conj(dataFourier)/nSamples; %PSD
          if ~mod(nSamples,2) %check even
              f = (1/mmPerPixel)/nSamples*(0:(nSamples/2-1));
          else %odd
              error('Expecting data to be even');
          end

          [peak locations] = findpeaks(Pyy(1:nSamples/2),
'SortStr','descend');

      %     figure
      %     plot(f(5:694),Pyy(5:694))
      %     title('Power Spectral Density')
      %     xlabel('lp/mm')

          %remove peaks that are outside of the frequency range
          removeIndicies = [];
          for index = 1:length(locations)
              if f(locations(index)) < frequencyRange(1) ||
f(locations(index)) > frequencyRange(2)
                  locations(index) = NaN;
              end
          end
          ind = find(isnan(locations));
          locations(ind) = [];
          peak(ind) = [];
```

```
            %Return the frequency at the max peak within the frequency
range
            frequency = f(locations(1));
            %Return the PSD Intensity of that peak
            peakPSD = peak(1);
        end
```

7.F Compare Experimental and Simulation Data with PSD Method

(CompareExperimentalandSimulationWithFourier.m)

```
        %% CompareExperimentalandSimulationWithFourier
        % Dependencies:
        % loadDataDirectories.m
        % findFrequencyWithFourier.m
        % findYAxisLimits.m

        %Expecting Two slit data for Axial Frequency Analysis
        %Singal slit data does not have axial frequency

        clear;close all;
        %% Initialize Parameters
        %Choose Experiment index; refer to loadDataDirectories()
        index = 14;
        %Choose yslice for experimental data to compare to simulation.
(there is only 1 yslice for simulation)
        yslice = 250;
        %Parameters to find lateral frequency
        mmPerPixel = 0.0064; %lateral mm/pixel. expecting same value for
simulation and experiment

        estimatedpercentError = .5; %1=100% %best practice to
overestimate first. applies to percent error of lp/mm. 50% default
        simulationFrequencyCeiling = 50; %lp/mm %because our simulation
data is noise free, this can remain relatively high for low frequency
patterns


        %% Set Directories and Specfic Directory Params
        %set directory for experimental data and simulation data
        directory = loadDataDirectories();

        %% Load, Normalize, and Calculate Frequency fo the Data
        %Load Simulation Data, specify the iraddiance signal variable
        load(directory(index).simulation,
directory(index).simulationData,
'xTheoreticalFrequency','zTheoreticalFrequency');
        ISignal.simulation.data = eval(directory(index).simulationData);
% simulated irradiance signal

        %some of the early simulation data does not contain
xTheoreticalFrequency
        %and is added in loadDataDirectories()
```

```
      if ~exist('xTheoreticalFrequency')
          theoreticalLateralFrequency =
directory(index).xTheoreticalFrequency;
      else
          theoreticalLateralFrequency = xTheoreticalFrequency;
      end

      %Load Experimental Data. expects X-Y plane with vertical fringes.
      %naming Convention-- zslice should correspond to the zslices
taken in
      %experimental data. example: 100.tif for the 100mm zslice.
      for zslice =
directory(index).zsliceStart:directory(index).zsliceInterval:directory(
index).zsliceEnd;
          %convert to double precision
          ISignal.experimental.data(:,:,zslice) =
double(imread(strcat(directory(index).experimental,
sprintf('%g.tif',zslice))));

          experimentalNormalizationfactor =
sum(mean(ISignal.experimental.data(:,:,zslice)));% take average along y
dim and sum

          %normalize the simulation data to experimental.
          simulationNormalizationFactor =
sum(ISignal.simulation.data(:,zslice)'); %take the transpose to align
fringes vertically and sum.
          ISignal.simulation.normalized(:,zslice) =
ISignal.simulation.data(:,zslice) *
(experimentalNormalizationfactor/simulationNormalizationFactor);


          %Calculate frequency
          [lateralFrequency.simulation.dataWithFourier(zslice),
lateralFrequency.simulation.peakPSD(zslice)] =
findFrequencyWithFourier(ISignal.simulation.normalized(:,zslice),mmPerP
ixel,[1, simulationFrequencyCeiling]);

          %experimentalFrequencyCeiling =
lateralFrequency.simulation.dataWithFourier(zslice) +
2*estimatedpercentError*lateralFrequency.simulation.dataWithFourier(zsl
ice);
          experimentalFrequencyCeiling = xTheoreticalFrequency +
estimatedpercentError*xTheoreticalFrequency;
          experimentalFrequencyFloor = xTheoreticalFrequency -
xTheoreticalFrequency*estimatedpercentError;
          experimentalFrequencyFloor =
max(experimentalFrequencyFloor,0);
          experimentalFrequencyRange = [experimentalFrequencyFloor,
experimentalFrequencyCeiling];

          [lateralFrequency.experimental.dataWithFourier(zslice),
lateralFrequency.experimental.peakPSD(zslice)]  =
findFrequencyWithFourier(ISignal.experimental.data(yslice,:,zslice),mmP
erPixel,experimentalFrequencyRange);
```

```
    end


    %% Plot frequencies found using Fourier transform technique

    %Calculate limits for y axis
    ylimit =
findYAxisLimits(lateralFrequency.simulation.dataWithFourier,0,lateralFr
equency.experimental.dataWithFourier,0);
    %Calculate Overall Averages
    overallAverageSimulation =
sum(lateralFrequency.simulation.dataWithFourier)./sum(lateralFrequency.
simulation.dataWithFourier~=0);
    overallAverageExperimental =
sum(lateralFrequency.experimental.dataWithFourier)./sum(lateralFrequenc
y.experimental.dataWithFourier~=0);

    figure,
    hold on
    plot(lateralFrequency.simulation.dataWithFourier,'ro')
    plot(lateralFrequency.experimental.dataWithFourier,'bx')
    line(get(gca,'XLim'),[theoreticalLateralFrequency
theoreticalLateralFrequency],'Color','k','LineWidth',1,'LineStyle','--
'); %plot Theoretical Lateral Frequency
    line(get(gca,'XLim'),[overallAverageSimulation
overallAverageSimulation],'Color','r','LineWidth',1); %plot overall
average
    line(get(gca,'XLim'),[overallAverageExperimental
overallAverageExperimental],'Color','b','LineWidth',1);
    hold off
    ylim(ylimit)
    xlim([directory(index).zsliceStart-5
directory(index).zsliceEnd+5])
    ylabel(sprintf('Exp Range:%g - %g lp/mm
',experimentalFrequencyRange(1),experimentalFrequencyRange(2)))
    xlabel(' z mm')
    legend('Simulation','Experimental',sprintf('%.2f',theoreticalLate
ralFrequency),sprintf('%g',overallAverageSimulation),sprintf('%g',overa
llAverageExperimental))
    title('Calculated Frequencies with PSD')

    %The PSD Intensity at each Z plane Experimental
    [axialFrequency.experimental, ~] =
findFrequencyWithFourier(lateralFrequency.experimental.peakPSD(125:250)
, 1, [0,50]);
    axialFrequency.experimental = axialFrequency.experimental/2; %Two
planes of resonance represent an axial period

    figure,
    hold on
    plot(lateralFrequency.experimental.peakPSD,'ro')
    xlim([directory(index).zsliceStart directory(index).zsliceEnd])
    xlabel(' z mm')
    ylabel('PSD')
    title(sprintf('Axial Frequency:%.3g Theoretical:%.3g
(lp/mm)',axialFrequency.experimental,zTheoreticalFrequency))
```

```
      suptitle('PSD Intensity at each Z Plane: Experimental')

      %The PSD Intensity at each Z plane Simulation Normalized
      [axialFrequency.simulation, ~] =
findFrequencyWithFourier(lateralFrequency.simulation.peakPSD, 1,
[0,50]);
      axialFrequency.simulation = axialFrequency.simulation/2; %Two
planes of resonance represent an axial period

      figure,
      hold on
      plot(lateralFrequency.simulation.peakPSD,'ro')
      xlim([directory(index).zsliceStart directory(index).zsliceEnd])
      xlabel(' z mm')
      ylabel('PSD')
      title(sprintf('Axial Frequency:%.3g Theoretical:%.3g
(lp/mm)',axialFrequency.simulation,zTheoreticalFrequency))
      suptitle('PSD Intensity at each Z Plane: Simulation Normalized')

      %% Stitch experimental data into on XZ image
      for zslice =
directory(index).zsliceStart:directory(index).zsliceInterval:directory(
index).zsliceEnd;
          XZImage.experimental(:,zslice) =
ISignal.experimental.data(yslice,:,zslice);
      end
      XZImage.experimental(:,1:directory(index).zsliceStart) =
[];%remove blank zslices
      %Experimental XZ image
      figure,
      imagesc(XZImage.experimental)
      ylabel('x pixels')
      xlabel('z mm')
      title('ISignal Experimental XZ');
      colormap(gray);

      %Simulation XZ image
      XZImage.simulation =
ISignal.simulation.normalized(:,1:directory(index).zsliceEnd);
      XZImage.simulation(:,1:directory(index).zsliceStart) = [];
%remove blank zslices
      figure,
      imagesc(XZImage.simulation)
      ylabel('x pixels')
      xlabel('z mm')
      title('IOSignal Simulation Normalized XZ');
      colormap(gray);

      %Adjusted Percent Error
      % choose the Z planes to compute a new average of the alteral
frequencies
      % in experimental data
      resonantplanes =
lateralFrequency.experimental.dataWithFourier(150:270);
      adjustedaverage = sum(resonantplanes)./sum(resonantplanes~=0)
```

```
      adjustedPercentError = abs((overallAverageSimulation -
adjustedaverage) / overallAverageSimulation *100)

      %Calculate lateral frequency at each Z plane Using PSD
      % for zslice = 10:1:300;
      % [lateralFrequency.simulation.dataWithFourier(zslice),
lateralFrequency.simulation.peakPSD(zslice)] =
findFrequencyWithFourier(ISignal.simulation.data(:,zslice),mmPerPixel,[
1, 50]);
      % end
      % figure,
      % hold on
      % plot(lateralFrequency.simulation.dataWithFourier,'ro')
      % line(get(gca,'XLim'),[theoreticalLateralFrequency
theoreticalLateralFrequency],'Color','k','LineWidth',1,'LineStyle','--
'); %plot Theoretical Lateral Frequency
      % line(get(gca,'XLim'),[overallAverageSimulation
overallAverageSimulation],'Color','r','LineWidth',1); %plot overall
average
      % hold off
      % ylim(ylimit)
      % xlim([10 300])
      % ylabel('lp/mm')
      % xlabel(' z mm')
      %
legend('Simulation','Experimental',sprintf('%.2f',theoreticalLateralFre
quency),sprintf('%g',overallAverageSimulation),sprintf('%g',overallAver
ageExperimental))
      % title('Calculated Lateral Frequencies at each Z plane with
PSD')
```

7.G Fresnel Integrals required for Matlab versions before 2014a

(fresnels.m)

```
      function FSint = fresnels(X,fresnelType)
      % fresnelS - Fresnel sine integrals, S(X), S1(X), or S2(X)
      % usage: FSint = fresnelS(X,fresnelType)
      %
      % Fresnel sine integrals fall into three classes, simple
      % transformations of each other. All three types described
      % by Abramowitz & Stegun are supported.
      %
      % The maximum error of this code has been shown to be less
      % than (approximately) 1.5e-14 for any value of X.
      %
      % arguments: (input)
      %  X - Any real, numeric value, vector, or array thereof.
      %      X is the upper limit of the Fresnel sine integral.
      %
      %  fresnelType - scalar numeric flag, from the set {0,1,2}.
      %
```

```
%      The type 0 Fresnel sine integral (A&S 7.3.1)
%        S(x) = \int_0^x sin(pi*t^2/2) dt,
%
%      Type 1 (A&S  7.3.3a)
%        S_1(x) = \sqrt(2/pi) \int_0^x sin(t^2) dt
%
%      Type 2 (A&S  7.3.3b)
%        S_2(x) = \sqrt(1/2/pi) \int_0^x sin(t) / \sqrt(t) dt
%
% arguments: (output)
%  FSint - array of the same size and shape as X, containing
%      the indicated Fresnel sine integral values.
%
%
% Example:
% % Evaluate the Fresnel sine integral S(x) at x = pi
% fresnelS(pi,0)
%
% % ans =
% %       0.598249078090266
%
% % Verify the correctness of this value using quadgk
% fresnelSObj = @(t) sin(pi*t.^2/2);
% quadgk(fresnelSObj,0,pi,'abstol',1e-15')
%
% % ans =
% %       0.598249078090268
%
% % Now, how fast is fresnelS? Using Steve Eddins timeit code
% % to yield an accurate estimate of the time required, we see
% % that it is reasonably fast for scalar input.
% timeit(@() fresnelS(pi))
% % ans =
% %       0.0002935014515
%
% % More importantly, fresnelS is vectorized.  So 1 million
% % evaluations are easy to do, and are much faster than
% % 1 million times the time taken for one evaluation.
% T = rand(1000000,1);
% tic
% FSpred = fresnelS(T);
% toc
% % Elapsed time is 0.220848 seconds.
%
%
% REFERENCES
% [1] Abramowitz, M. and Stegun, I. A. (Eds.). "Error Function
and Fresnel
%     Integrals." Ch. 7 in Handbook of Mathematical Functions
with
%     Formulas, Graphs, and Mathematical Tables, 9th printing.
New York:
%     Dover, pp. 295-329, 1970.
%
% [2] Mielenz, K. D.; "Computation of Fresnel Integrals", Journal
of
```

```
      %     Research of the National Institute of Standards and
Technology,
      %     Vol 102, Number 3, May-June 1997
      %        http://nvl.nist.gov/pub/nistpubs/jres/102/3/j23mie.pdf

      persistent FSspl

      if (nargin < 1) || (nargin > 2)
        error('FRESNELS:improperarguments','1 or 2 arguemtns are
required.')
      end

      % default for fresnelType
      if (nargin < 2) || isempty(fresnelType)
        fresnelType = 0;
      else
        if ~isnumeric(fresnelType) || ~ismember(fresnelType,[0 1 2]) ||
(numel(fresnelType) ~= 1)
          error('FRESNELS:fresnelType', ...
            'fresnelType must be scalar, one of {0,1,2} if supplied.')
        end
      end

      % X must be real, but of any shape.
      % if any(imag(X) ~= 0)
      %   warning('FRESNELS:complexarguments','X should be real.
Imaginary part will ignored.')
      %   X = real(X);
      % end

      % preallocate FSint to the proper size
      FSint = zeros(size(X));

      % flag any negative X, make it positive.
      S = X < 0;
      X(S) = -X(S);

      % transform the type 1 and 2 problems into type 0
      switch fresnelType
        case 1
          X = sqrt(2/pi)*X;
        case 2
          X = sqrt(2*X/pi);
      end

      % The upper limit of the tables is 7.5.
      Xlim = 7.5;
      % klim is a boolean variable that indicates values that exceed
Xlim.
      klim = (X > Xlim);
      if any(klim(:))
        % we found some values that exceed the limit. Use
        % the rational approximations provided in Mielenz [2]
        % for the associated functions f(z) (see (4a)) and
        % g(z) (see (4b)). The approximations are carried to
```

```
        % additional terms beyond that displayed in Mielenz.
        %
        % For abs(X) >= 7.5, these yield results with
        % roughly 15 significant digits.
        xk = X(klim);

        FSint(klim) = 0.5 - (1 - 3/pi^2 ./xk.^4 + 105/pi^4 ./xk.^8 -
...
          10395/pi^6 ./xk.^12 + 2027025/pi^8
./xk.^16).*cos(pi/2*xk.^2)./(pi*xk) - ...
          (1 - 15/pi^2 ./xk.^4 + 945/pi^4 ./xk.^8 - 135135/pi^6
./xk.^12 + ...
          34459425/pi^8 ./xk.^16).*sin(pi/2*xk.^2)./(pi^2*xk.^3);

      end
      klim = ~klim;

      % for abs(Xlim) <= Xlim, we will use a spline interpolant of the
      % sine integral itself.
      if any(klim(:))
        % have we loaded the appropriate spline?
        if isempty(FSspl)
          load _Fresnel_data_ FSspl
        end

        % do the interpolation itself using ppval. This will be
        % better than calling interp1 with the 'spline' option,
        % since it avoids overhead of calling an already created
        % and stored spline. It will be better than pchip or the
        % 'cubic' option for interp1 since the spline will be
        % considerably more accurate.
        FSint(klim) = ppval(FSspl,X(klim));
      end

      % The Fresnel sine and cosine integrals are odd functions of X,
      % so swap signs for any negative X.
      FSint(S) = - FSint(S);

      end % mainline end
```

(fresnelc.m)

```
      function FCint = fresnelc(X,fresnelType)
      % fresnelC - Fresnel cosine integrals, C(X), C1(X), or C2(X)
      % usage: FCint = fresnelC(X,fresnelType)
      %
      % Fresnel cosine integrals fall into three classes, simple
      % transformations of each other. All three types described
      % by Abramowitz & Stegun are supported.
      %
      % The maximum error of this code has been shown to be less
      % than 1.5e-14 for any value of X.
      %
      % arguments: (input)
      %  X - Any real, numeric value, vector, or array thereof.
```

```
%      X is the upper limit of the Fresnel cosine integral.
%
%  fresnelType - scalar numeric flag, from the set {0,1,2}.
%
%      The type 0 Fresnel cosine integral (A&S 7.3.1)
%        C(x) = \int_0^x cos(pi*t^2/2) dt,
%
%      Type 1 (A&S  7.3.3a)
%        C_1(x) = \sqrt(2/pi) \int_0^x cos(t^2) dt
%
%      Type 2 (A&S  7.3.3b)
%        C_2(x) = \sqrt(1/2/pi) \int_0^x cos(t) / \sqrt(t) dt
%
% arguments: (output)
%  FCint - array of the same size and shape as X, containing
%      the indicated Fresnel cosine integral values.
%
%
% Example:
% % Evaluate the Fresnel cosine integral C(x) at x = 1.38
%
% fresnelC(1.38,0)
%
% % ans =
% %       0.562975925772444
%
% % Verify the correctness of this value using quadgk
% FresnelCObj = @(t) cos(pi*t.^2/2);
% quadgk(FresnelCObj,0,1.38,'abstol',1e-15')
%
% % ans =
% %       0.562975925772444
%
% % Now, how fast is fresnelC? Using Steve Eddins timeit code
% % to yield an accurate estimate of the time required, we see
% % that it is reasonably fast for scalar input.
% timeit(@() fresnelC(1.38))
% % ans =
% %       0.000193604455833333
%
% % More importantly, fresnelC is vectorized. So 1 million
% % evaluations are easy to do, and are much faster than
% % 1 million times the time taken for one evaluation.
% T = rand(1000000,1);
% tic
% FCpred = fresnelC(T);
% toc
% % Elapsed time is 0.226884 seconds.
%
%
% REFERENCES
% [1] Abramowitz, M. and Stegun, I. A. (Eds.). "Error Function
and Fresnel
%     Integrals." Ch. 7 in Handbook of Mathematical Functions
with
%     Formulas, Graphs, and Mathematical Tables, 9th printing.
New York:
```

```
%     Dover, pp. 295-329, 1970.
%
% [2] Mielenz, K. D.; "Computation of Fresnel Integrals", Journal
of
%     Research of the National Institute of Standards and
Technology,
%     Vol 102, Number 3, May-June 1997
%        http://nvl.nist.gov/pub/nistpubs/jres/102/3/j23mie.pdf

persistent FCspl

if (nargin < 1) || (nargin > 2)
  error('FRESNELC:improperarguments','1 or 2 arguemtns are
required.')
end

% default for fresnelType
if (nargin < 2) || isempty(fresnelType)
  fresnelType = 0;
else
  if ~isnumeric(fresnelType) || ~ismember(fresnelType,[0 1 2]) ||
(numel(fresnelType) ~= 1)
    error('FRESNELC:fresnelType', ...
      'fresnelType must be scalar, one of {0,1,2} if supplied.')
  end
end

% % X must be real, but of any shape.
% if any(imag(X) ~= 0)
%   warning('FRESNELC:complexarguments','X should be real.
Imaginary part will ignored.')
%   X = real(X);
% end

% preallocate FCint to the proper size
FCint = zeros(size(X));

% flag any negative X, make it positive.
S = X < 0;
X(S) = -X(S);

% transform the type 1 and 2 problems into type 0
switch fresnelType
  case 1
    X = sqrt(2/pi)*X;
  case 2
    X = sqrt(2*X/pi);
end

% The upper limit of the tables is 7.5.
Xlim = 7.5;
% klim is a boolean variable that indicates values that exceed
Xlim.
klim = (X >= Xlim);
if any(klim(:))
```

```
        % we found some values that exceed the limit. Use
        % the rational approximations provided in Mielenz [2]
        % for the associated functions f(z) (see (4a)) and
        % g(z) (see (4b)). The approximations are carried to
        % additional terms beyond that displayed in Mielenz.
        %
        % For abs(X) >= 7.5, these yield results with
        % roughly 15 significant digits.
        xk = X(klim);

        FCint(klim) = 0.5 + (1 - 3/pi^2 ./xk.^4 + 105/pi^4 ./xk.^8 -
...
          10395/pi^6 ./xk.^12 + 2027025/pi^8
./xk.^16).*sin(pi/2*xk.^2)./(pi*xk) - ...
          (1 - 15/pi^2 ./xk.^4 + 945/pi^4 ./xk.^8 - 135135/pi^6
./xk.^12 + ...
          34459425/pi^8 ./xk.^16).*cos(pi/2*xk.^2)./(pi^2*xk.^3);

      end
      klim = ~klim;

      % for abs(Xlim) <= Xlim, we will use a spline interpolant of the
      % cosine integral itself.
      if any(klim(:))
        % have we loaded the appropriate spline?
        if isempty(FCspl)
          load _Fresnel_data_ FCspl
        end

        % do the interpolation itself using ppval. This will be
        % better than calling interp1 with the 'spline' option,
        % since it avoids overhead of calling an already created
        % and stored spline. It will be better than pchip or the
        % 'cubic' option for interp1 since the spline will be
        % considerably more accurate.
        FCint(klim) = ppval(FCspl,X(klim));
      end

      % The Fresnel sine and cosine integrals are odd functions of X,
      % so swap signs for any negative X.
      FCint(S) = - FCint(S);

      end % mainline end

      % ==============================================================
      %     Code used only to generate and save the integral tables
      % ==============================================================
      function generateTables

      % Generate the integral tables, more accurate than Abramowitz &
      % Stegun provide, since they give only 7 digits.
      FresnelCObj = @(t) cos(pi*t.^2/2);
      FresnelSObj = @(t) sin(pi*t.^2/2);

      p = 1.75;
```

```
      T0 = linspace(1,7.5.^p,501).' .^(1/p);
      dt = T0(2) - T0(1);
      T0 = [linspace(0,1 - dt,ceil(1./dt))';T0];
      plot(diff(T0))

      n = length(T0);
      FC75 = zeros(n,1);
      FS75 = zeros(n,1);

      h = waitbar(0,'Computing Fresnel integrals');
      for i = 2:n
        waitbar(i/n,h)
        FC75(i) = quadgk(FresnelCObj,0,T0(i),'abstol',1.e-
16,'reltol',100*eps('double'));
        FS75(i) = quadgk(FresnelSObj,0,T0(i),'abstol',1.e-
16,'reltol',100*eps('double'));
      end
      delete(h)

      % Turn them into splines, then save the splines. These splines
are
      % first built in a Hermite form, since I can supply the 1st and
second
      % derivatives of the function. Then I turn them into a pp form,
for use
      % in fresnelC and fresnelS.
      FCspl = hermite2slm([T0,FC75,FresnelCObj(T0), -
pi*T0.*sin(pi*T0.^2/2), ...
        -pi*(sin(pi*T0.^2/2) + pi*T0.^2 .*cos(pi*T0.^2/2))]);
      FCspl = slm2pp(FCspl);

      FSspl =
hermite2slm([T0,FS75,FresnelSObj(T0),pi*T0.*cos(pi*T0.^2/2), ...
        pi*(cos(pi*T0.^2/2) - pi*T0.^2 .*sin(pi*T0.^2/2))]);
      FSspl = slm2pp(FSspl);

      save _Fresnel_data_ FCspl FSspl


      % test the result
      clear functions

      n = 1000;
      T = sort(rand(n,1)*10);
      FCquad = zeros(n,1);
      FSquad = zeros(n,1);
      for i = 1:n
        FCquad(i) = quadgk(FresnelCObj,0,T(i),'abstol',1.e-16);
        FSquad(i) = quadgk(FresnelSObj,0,T(i),'abstol',1.e-16);
      end
      FCpred = fresnelc(T,0);
      FSpred = fresnels(T,0);

      subplot(1,2,1)
      plot(T,FCquad - FCpred,'.')
```

```
grid on
subplot(1,2,2)
plot(T,FSquad - FSpred,'.')
grid on

end
```

7.H Verification of AFA Code (VerificaitonOfFindFrequency.m)

```
%Test for Find Frequency Algorithm
%Parameters
%Coherent Source 635nm
%BiprismFBP2020G-179
%One 2um Slit
%birpirsm position eta 50mm
%convergining lens focal length 150mm

clear;
experimentalDataDir =
'C:\Users\ctylor10\Documents\ExperimentalDATA\10_7_15\Coherent Source
Red\BiprismFBP2020G-179\One 5um Slit 100mm FtoBseperation 400us
exposure\';
simulationDataDir = 'C:\Users\ctylor10\Documents\MATLAB\Fresnel
Biprism\SimulationData\10_7_15\FBP2020G-
179_635nm_0.200mmslits_50Eta_150Focal_1388x600_0.0064x_1zmmperpixel.mat
';

yslice = 400;
%Parameters to find lateral frequency
mmPerPixel = 0.0064; %lateral mm/pixel. expecting same value for
simulation and experiment

%Load Simulation Data
load(simulationDataDir,'ISignalOneSlit')
ISignalSimData = ISignalOneSlit;
for zslice = 130;
    %convert to double precision
    ISignalExpData(:,:,zslice) =
double(imread(strcat(experimentalDataDir, sprintf('%g.tif',zslice))));

    average = mean(ISignalExpData(:,:,zslice)); % take average
along y dim
    experimentalNormalizationfactor = sum(average);

    %normalize the simulation data to experimental. take the
transpose to
    %align fringes vertically.
    simulationNormalizationFactor =
sum(ISignalSimData(:,zslice)');
    ISignalSimNormalized(:,zslice) = ISignalSimData(:,zslice) *
(experimentalNormalizationfactor/simulationNormalizationFactor);
end

%Calculate frequency
estimatedpercentError = 1.0;
```

```
      simulationFrequencyCeiling = 50;

      [lateralSimulationFrequency,~,~,~] =
findfrequency(ISignalSimNormalized(:,zslice),mmPerPixel,simulationFrequ
encyCeiling);
      experimentalFrequencyCeiling = lateralSimulationFrequency +
2*estimatedpercentError*lateralSimulationFrequency;


      data = ISignalExpData(yslice,:,zslice);
      frequencyCeiling = experimentalFrequencyCeiling;

      minIntensity = max(data)*.4;
      minPixels = floor(1/(frequencyCeiling*mmPerPixel));

      [~,lateralPeakLocationsWithAlgorithm] =
findpeaks(data,'MinPeakHeight',minIntensity,'MinPeakDist',minPixels);
      lateralFrequencyWithAlgorithm =
1/(mean(diff(lateralPeakLocationsWithAlgorithm).*mmPerPixel));
      %Store the boundaries of the window
      window(1) = lateralPeakLocationsWithAlgorithm(1);
      window(2) = lateralPeakLocationsWithAlgorithm(end);

      percentErrorWithAlgorithm = (lateralFrequencyWithAlgorithm-
lateralSimulationFrequency)./lateralSimulationFrequency.*100;

      %NO ALGORITHM
      minIntensity = max(data)*.4;

      [~,lateralPeakLocationsNoAlgorithm] =
findpeaks(data,'MinPeakHeight',minIntensity);
      lateralFrequencyNoAlgorithm =
1/(mean(diff(lateralPeakLocationsNoAlgorithm).*mmPerPixel));

      figure,
      subplot(1,2,1)
      imagesc(ISignalSimNormalized(:,zslice)');
      ylabel('y a.u.');
      xlabel('x pixels');
      title('Simulation');
      colormap(gray);
      subplot(1,2,2)
      imagesc(ISignalExpData(:,:,zslice));
      ylabel('y pixels');
      xlabel('x pixels');
      title('Experimental');
      suptitle(sprintf('Irradiance Signal at z = %gmm',zslice))

      %Intensity LinePlots
      figure,
      hold on
      plot(ISignalExpData(yslice,:,zslice),'r');%draw intensity
      plot(lateralPeakLocationsWithAlgorithm,ISignalExpData(yslice,late
ralPeakLocationsWithAlgorithm,zslice),'ro','MarkerSize',10);%draw Peaks
Identified with Algorithm
```

```
      plot(lateralPeakLocationsNoAlgorithm,ISignalExpData(yslice,latera
lPeakLocationsNoAlgorithm,zslice),'bx');%draw Peaks Identified No
Algorithm
      line([window(1)
window(1)],get(gca,'YLim'),'Color','k','LineWidth',1);%draw window
boundaries
      line([window(2)
window(2)],get(gca,'YLim'),'Color','k','LineWidth',1);
      hold off
      title(sprintf('Experimental: %g lp/mm | %.2f%%
Error',lateralFrequencyWithAlgorithm,percentErrorWithAlgorithm));
      ylabel('Intensity');
      xlabel('x pixels');
      legend(sprintf('y = %g',yslice),'Peaks W/Algorithm', 'Peaks No
Algorithm');
      suptitle(sprintf('Intensity Line Plot at z = %gmm',zslice))
```

7.I Verify PSD method for lateral frequency analysis (powerspectraldensity.m)

```
      %plot power spectral density at a specified y- and z- slice.
specify a
      %frequency range to search for a maximum in the power spectral
density

      % Dependencies:
      % loadDataDirectories.m

      clear; close all
      index = 9;
      directory = loadDataDirectories();
      xTheoreticalFrequency = directory(index).xTheoreticalFrequency;
      load(directory(index).simulation,
directory(index).simulationData, 'xTheoreticalFrequency');
      ISignal.simulation.data = eval(directory(index).simulationData);

      yslice = 500
      zslice = 50
      frequencyRange = [0,40]
      ISignal.experimental.data(:,:,zslice) =
double(imread(strcat(directory(index).experimental,
sprintf('%g.tif',zslice))));
      %set y.sin to experimental or simulation data
      %y.sin = ISignal.simulation.data(:,zslice)
      y.sin = ISignal.experimental.data(yslice,:,zslice)


      y.fft = fft(y.sin,1388);
      Pyy = y.fft.*conj(y.fft)/1388;
      f = 156.25/1388*(0:693);

      [peak locations] = findpeaks(Pyy(1:694), 'SortStr','descend');
      removeIndicies = [];
      for index = 1:length(locations)
```

```
          if f(locations(index)) < frequencyRange(1) ||
f(locations(index)) > frequencyRange(2)
              locations(index) = NaN;
          end
      end
      ind = find(isnan(locations));
      locations(ind) = [];

      frequency = f(locations(1));

      %plot line intensity profile
      figure
      plot(y.sin)
      title(sprintf('Experimental: %g lp/mm',frequency));
      ylabel('Intensity');
      xlabel('x pixels');
      legend(sprintf('y = %g',yslice));

      %Plot power spectral density
      figure
      plot(f(5:694),Pyy(5:694))
      title('Power Spectral Density')
      xlabel('lp/mm')

      %image experimental irradiance signal
      figure
      imagesc(ISignal.experimental.data(:,:,zslice))
      ylabel('y pixels');
      xlabel('x pixels');
      title('Experimental');
      suptitle(sprintf('Irradiance Signal at z = %gmm',zslice))
      colormap(gray);
```

7.J Verify PSD method for axial frequency analysis

(powerspectraldensitythroughZ.m)

```
      %power spectral density through XZ for simulation
      %calculate axial frequency
      %testing debugging script* refer to
      %CompareExperimentalandSimulationWithFourier
      %
      % Dependencies:
      % loadDataDirectories.m

      clear; close all
      index = 9;
      directory = loadDataDirectories();
      xTheoreticalFrequency = directory(index).xTheoreticalFrequency;
      load(directory(index).simulation,
directory(index).simulationData, 'zTheoreticalFrequency');
      ISignal.simulation.data = eval(directory(index).simulationData);
```

```
      ISignal.simulation.data(:,1) = []; %Remove first two columns to
maintain even number of z, First row is NaN
      ISignal.simulation.data(:,2) = [];

      xslice = length(ISignal.simulation.data)/2; %Choose middle x to
evaluate
      frequencyRange = [0,50];

      %set data to y.sin
      y.sin = ISignal.simulation.data(xslice,:)
      nsamples = size(y.sin,2)

      y.fft = fft(y.sin,nsamples);
      Pyy = y.fft.*conj(y.fft)/nsamples;
      f = 1/nsamples*(0:nsamples/2-1);

      [peak locations] = findpeaks(Pyy(1:nsamples/2),
'SortStr','descend');
      removeIndicies = [];
      for i = 1:length(locations)
          if f(locations(i)) < frequencyRange(1) || f(locations(i)) >
frequencyRange(2)
              locations(i) = NaN;
          end
      end
      ind = find(isnan(locations));
      locations(ind) = [];

      frequency = f(locations(1));

      %% plot line intensity profile
      ind = 1;
      xtickind = 0;
      xticklabels = {}
      xticks = [];
      for i = 50:10:150
          xticklabels(ind) = cellstr(sprintf('%g',i));
          xticks(ind) = xtickind;
          ind = ind +1;
          xtickind = xtickind + 10;
      end

      figure
      plot(y.sin(50:150))
      title(sprintf('Simulation: %g lp/mm Theoretical: %g
lp/mm',frequency,zTheoreticalFrequency));
      ylabel('Intensity');
      xlabel('z mm');
      ax = gca;
      ax.XTickLabel = xticklabels;
      ax.XTick = xticks;
      legend(sprintf('x = %g',xslice));
      suptitle('Axial Frequency Analysis');
```

```
%% Plot power spectral density
figure
plot(f(5:nsamples/2),Pyy(5:nsamples/2))
title('Power Spectral Density')
xlabel('lp/mm')
```

7.K List of Experimental and Simulation Data Configurations

(loadDataDirectories.m)

```
function [directory] = loadDataDirectories()
%experimental and simulation directories must be in the matlab current
folder
%the current matlab folder is concatenated with the data folder at the
end
%of this script

%All Experimental data taken at 1388x600 resolution and 0.0064 mm/pixel
in
%XY (Axio Vision camera parameters) except for the alst set (Redux)
%All Simulations use 1 mm/pixel in Z

%List of Configurations
% 1 - Coherent 635nm Biprism FBP2020G-179 One 0.127um Slit
50Eta_150Focal 1388x600 resolution 0.0064xmmperpixel_1zmmperpixel
% 2 - Coherent 635nm Biprism FBP2020G-179 One 0.127um Slit
130Eta_150Focal 1388x600 resolution 0.0064xmmperpixel_1zmmperpixel
% 3 - Incoherent 470nm Biprism FBP2020G-179 One 0.1524um Slit
50Eta_150Focal 1388x600 resolution 0.0064xmmperpixel_1zmmperpixel
% 4 - Incoherent 470nm Biprism FBP2020G-179 One 0.1524um Slit
130Eta_150Focal 1388x600 resolution 0.0064xmmperpixel_1zmmperpixel
% 5 - Incoherent 470nm Biprism FBP2020G-175 One 0.0254um Slit
40Eta_150Focal 1388x600 resolution 0.0064xmmperpixel_1zmmperpixel
% 6 - Incoherent 470nm Biprism FBP2020G-179 One 0.0508um Slit
50Eta_150Focal 1388x600 resolution 0.0064xmmperpixel_1zmmperpixel
% 7 - Incoherent 470nm Biprism FBP2020G-179 One 0.0508um Slit
130Eta_150Focal 1388x600 resolution 0.0064xmmperpixel_1zmmperpixel

%Two Slits - 70um Slit Width - 5mm ZSlices
% 8  - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 200um
Separation 50Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 9  - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 200um
Separation 130Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 10 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 300um
Separation 50Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 11 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 300um
Separation 130Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 12 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 500um
Separation 50Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
```

```
% 13 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 500um
Separation 130Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel

%Two Slits - 70um Slit Width - 1mm ZSlices
% 14 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 200um
Separation 50Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 15 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 200um
Separation 120Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 16 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 300um
Separation 50Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 17 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 300um
Separation 120Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 18 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 500um
Separation 50Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel
% 19 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 500um
Separation 120Eta_150Focal 1388x600 resolution
0.0064xmmperpixel_1zmmperpixel

%Two Slits - 70um Slit Width - 1mm ZSlices - Redux - HC Flash 4.0 V2
% 20 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 200um
Separation 50Eta_150Focal 2048x2048 resolution
0.0065xmmperpixel_1zmmperpixel
% 21 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 200um
Separation 120Eta_150Focal 2048x2048 resolution
0.0065xmmperpixel_1zmmperpixel
% 22 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 300um
Separation 50Eta_150Focal 2048x2048 resolution
0.0065xmmperpixel_1zmmperpixel
% 23 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 300um
Separation 120Eta_150Focal 2048x2048 resolution
0.0065xmmperpixel_1zmmperpixel
% 24 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 500um
Separation 50Eta_150Focal 2048x2048 resolution
0.0065xmmperpixel_1zmmperpixel
% 25 - Incoherent 470nm Biprism FBP2020G-179 Two 70um Slits 500um
Separation 120Eta_150Focal 2048x2048 resolution
0.0065xmmperpixel_1zmmperpixel


%Directories from 10/7/15
    directory(1).experimental = '\Data\Experimental\10_7_15\Coherent
Source Red\BiprismFBP2020G-179\One 0.127um Slit 100mm FtoBseperation
400us exposure\';
    directory(1).simulation = '\Data\Simulation\10_7_15\FBP2020G-
179_635nm_0.200mmslits_50Eta_150Focal_1388x600_0.0064x_1zmmperpixel.mat
';
    directory(1).zsliceInterval = 10;
    directory(1).zsliceStart = 50;
    directory(1).zsliceEnd = 150;
    directory(1).xTheoreticalFrequency = 4.7185;
```

```
    directory(1).simulationData = 'I0SignalOneSlit';

    directory(2).experimental = '\Data\Experimental\10_7_15\Coherent
Source Red\BiprismFBP2020G-179\One 0.127um Slit 20mm FtoBseperation
400us exposure\';
    directory(2).simulation = '\Data\Simulation\10_7_15\FBP2020G-
179_635nm_0.200mmslits_130Eta_150Focal_1388x600_0.0064x_1zmmperpixel.ma
t';
    directory(2).zsliceInterval = 10;
    directory(2).zsliceStart = 50;
    directory(2).zsliceEnd = 150;
    directory(2).xTheoreticalFrequency = 12.2680;
    directory(2).simulationData = 'I0SignalOneSlit';

    directory(3).experimental = '\Data\Experimental\10_7_15\Incoherent
Source 470nm Blue\BiprismFBP2020G-179\0.1524um OneSlit
100mmFtoBseperation 1.0msExposure 1388x1040\';
    directory(3).simulation = '\Data\Simulation\10_7_15\FBP2020G-
179_470nm_0.200mmslits_50Eta_150Focal_1388x600_0.0064x_1zmmperpixel.mat
';
    directory(3).zsliceInterval = 10;
    directory(3).zsliceStart = 50;
    directory(3).zsliceEnd = 150;
    directory(3).xTheoreticalFrequency = 6.3749;
    directory(3).simulationData = 'I0SignalOneSlit';

    directory(4).experimental = '\Data\Experimental\10_7_15\Incoherent
Source 470nm Blue\BiprismFBP2020G-179\0.1524um OneSlit
20mmFtoBseperation 1msExposure 1388x1040\';
    directory(4).simulation = '\Data\Simulation\10_7_15\FBP2020G-
179_470nm_0.200mmslits_130Eta_150Focal_1388x600_0.0064x_1zmmperpixel.ma
t';
    directory(4).zsliceInterval = 10;
    directory(4).zsliceStart = 50;
    directory(4).zsliceEnd = 150;
    directory(4).xTheoreticalFrequency = 16.5749;
    directory(4).simulationData = 'I0SignalOneSlit';

    directory(5).experimental = '\Data\Experimental\10_7_15\Incoherent
Source 470nm Blue\BiprismFBP2020G-
175\0.0254umOneSlit_1080_110mmFtoBseperation_90msExposure_IncoherentSou
rce\';
    directory(5).simulation = '\Data\Simulation\10_7_15\FBP2020G-
175_470nm_0.200mmslits_40Eta_150Focal_1388x600_0.0064x_1zmmperpixel.mat
';
    directory(5).zsliceInterval = 10;
    directory(5).zsliceStart = 50;
    directory(5).zsliceEnd = 150;
    directory(5).xTheoreticalFrequency = 25.5153;
    directory(5).simulationData = 'I0SignalOneSlit';

%Directories from 10/20/15
    directory(6).experimental = '\Data\Experimental\10_20_15\Incoherent
Source 470nm Blue\0.0508umOneSlit 100mmFtoBseperation 4.8ms exposure
1388x1040\';
```

```
    directory(6).simulation = '\Data\Simulation\10_20_15\FBP2020G-
179_470nm_0.200mmslits_50Eta_150Focal_2umSlitWidth_1388x300_0.0064x_1zm
mperpixel.mat';
    directory(6).zsliceInterval = 5;
    directory(6).zsliceStart = 50;
    directory(6).zsliceEnd = 150;
    directory(6).xTheoreticalFrequency = 6.37495;
    directory(6).simulationData = 'I0SignalOneSlit';

    directory(7).experimental = '\Data\Experimental\10_20_15\Incoherent
Source 470nm Blue\0.0508umOneSlit 20mmFtoBSeperation 8.4msExposure
1388x1040\';
    directory(7).simulation = '\Data\Simulation\10_20_15\FBP2020G-
179_470nm_0.200mmslits_130Eta_150Focal_2umSlitWidth_1388x300_0.0064x_1z
mmperpixel.mat';
    directory(7).zsliceInterval = 5;
    directory(7).zsliceStart = 50;
    directory(7).zsliceEnd = 150;
    directory(7).simulationData = 'I0SignalOneSlit';

%Directories from 12/15/15
    directory(8).experimental =
'\Data\Experimental\12_15_15\70umTwoSlit 200umSeparation
50mmSlitToBiprism 1.0msExposure 1388x1040\';
    directory(8).simulation = '\Data\Simulation\12_15_15\FBP2020G-
179_470nm_0.200mmslits_50Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1z
mmperpixel.mat';
    directory(8).zsliceInterval = 5;
    directory(8).zsliceStart = 50;
    directory(8).zsliceEnd = 150;
    directory(8).simulationData = 'I0SignalTwoSlits';

    directory(9).experimental =
'\Data\Experimental\12_15_15\70umTwoSlit 200umSeparation
130mmSlitToBiprism 1.0msExposure 1388x1040\';
    directory(9).simulation = '\Data\Simulation\12_15_15\FBP2020G-
179_470nm_0.200mmslits_130Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1
zmmperpixel.mat';
    directory(9).zsliceInterval = 5;
    directory(9).zsliceStart = 50;
    directory(9).zsliceEnd = 150;
    directory(9).simulationData = 'I0SignalTwoSlits';

    directory(10).experimental =
'\Data\Experimental\12_15_15\70umTwoSlit 300umSeparation
50mmSlitToBiprism 1.6msExposure 1388x1040\';
    directory(10).simulation = '\Data\Simulation\12_15_15\FBP2020G-
179_470nm_0.300mmslits_50Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1z
mmperpixel.mat';
    directory(10).zsliceInterval = 5;
    directory(10).zsliceStart = 50;
    directory(10).zsliceEnd = 150;
    directory(10).simulationData = 'I0SignalTwoSlits';
```

```
    directory(11).experimental =
'\Data\Experimental\12_15_15\70umTwoSlit 300umSeparation
130mmSlitToBiprism 1.6msExposure 1388x1040\';
    directory(11).simulation = '\Data\Simulation\12_15_15\FBP2020G-
179_470nm_0.300mmslits_130Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1
zmmperpixel.mat';
    directory(11).zsliceInterval = 5;
    directory(11).zsliceStart = 50;
    directory(11).zsliceEnd = 150;
    directory(11).simulationData = 'I0SignalTwoSlits';

    directory(12).experimental =
'\Data\Experimental\12_15_15\70umTwoSlit 500umSeparation
50mmSlitToBiprism 1.0msExposure 1388x1040\';
    directory(12).simulation = '\Data\Simulation\12_15_15\FBP2020G-
179_470nm_0.500mmslits_50Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1z
mmperpixel.mat';
    directory(12).zsliceInterval = 5;
    directory(12).zsliceStart = 50;
    directory(12).zsliceEnd = 150;
    directory(12).simulationData = 'I0SignalTwoSlits';

    directory(13).experimental =
'\Data\Experimental\12_15_15\70umTwoSlit 500umSeparation
130mmSlitToBiprism 1.0msExposure 1388x1040\';
    directory(13).simulation = '\Data\Simulation\12_15_15\FBP2020G-
179_470nm_0.500mmslits_130Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1
zmmperpixel.mat';
    directory(13).zsliceInterval = 5;
    directory(13).zsliceStart = 50;
    directory(13).zsliceEnd = 150;
    directory(13).simulationData = 'I0SignalTwoSlits';

%Directories from 2/23/16
    directory(14).experimental =
'\Data\Experimental\2_23_16\70umTwoSlit 200umSeparation
50mmSlitToBiprism 1mmZSlices20-300\';
    directory(14).simulation = '\Data\Simulation\2_23_16\FBP2020G-
179_470nm_0.200mmslits_50Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1z
mmperpixel.mat';
    directory(14).zsliceInterval = 1;
    directory(14).zsliceStart = 20;
    directory(14).zsliceEnd = 300;
    directory(14).simulationData = 'I0SignalTwoSlits';

    directory(15).experimental =
'\Data\Experimental\2_23_16\70umTwoSlit 200umSeparation
120mmSlitToBiprism 1mmZSlices20-300\';
    directory(15).simulation = '\Data\Simulation\2_23_16\FBP2020G-
179_470nm_0.200mmslits_120Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1
zmmperpixel.mat';
    directory(15).zsliceInterval = 1;
    directory(15).zsliceStart = 20;
    directory(15).zsliceEnd = 300;
    directory(15).simulationData = 'I0SignalTwoSlits';
```

```
    directory(16).experimental =
'\Data\Experimental\2_23_16\70umTwoSlit 300umSeparation
50mmSlitToBiprism 1mmZSlices20-300\';
    directory(16).simulation = '\Data\Simulation\2_23_16\FBP2020G-
179_470nm_0.300mmslits_50Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1z
mmperpixel.mat';
    directory(16).zsliceInterval = 1;
    directory(16).zsliceStart = 20;
    directory(16).zsliceEnd = 300;
    directory(16).simulationData = 'I0SignalTwoSlits';

    directory(17).experimental =
'\Data\Experimental\2_23_16\70umTwoSlit 300umSeparation
120mmSlitToBiprism 1mmZSlices20-300\';
    directory(17).simulation = '\Data\Simulation\2_23_16\FBP2020G-
179_470nm_0.300mmslits_120Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1
zmmperpixel.mat';
    directory(17).zsliceInterval = 1;
    directory(17).zsliceStart = 20;
    directory(17).zsliceEnd = 300;
    directory(17).simulationData = 'I0SignalTwoSlits';

    directory(18).experimental =
'\Data\Experimental\2_23_16\70umTwoSlit 500umSeparation
50mmSlitToBiprism 1mmZSlices20-270\';
    directory(18).simulation = '\Data\Simulation\2_23_16\FBP2020G-
179_470nm_0.500mmslits_50Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1z
mmperpixel.mat';
    directory(18).zsliceInterval = 1;
    directory(18).zsliceStart = 20;
    directory(18).zsliceEnd = 270;
    directory(18).simulationData = 'I0SignalTwoSlits';

    directory(19).experimental =
'\Data\Experimental\2_23_16\70umTwoSlit 500umSeparation
120mmSlitToBiprism 1mmZSlices20-300\';
    directory(19).simulation = '\Data\Simulation\2_23_16\FBP2020G-
179_470nm_0.500mmslits_120Eta_150Focal_70umSlitWidth_1388x300_0.0064x_1
zmmperpixel.mat';
    directory(19).zsliceInterval = 1;
    directory(19).zsliceStart = 20;
    directory(19).zsliceEnd = 300;
    directory(19).simulationData = 'I0SignalTwoSlits';


%Directories from 4/5/16
    directory(20).experimental = 'Z:\Chris\Experimental
DATA\4_5_16\D2mmSlitSeparation50mmBiPrism\Image';
    directory(20).simulation = 'E:\MATLAB\FBP2020G-
179_470nm_0.200mmslits_50Eta_150Focal_70umSlitWidth_2048x300_0.0065x_1z
mmperpixel.mat';
    directory(20).zsliceInterval = 1;
    directory(20).zsliceStart = 40;
    directory(20).zsliceEnd = 300;
    directory(20).simulationData = 'I0SignalTwoSlits';
```

```
    directory(21).experimental = 'Z:\Chris\Experimental
DATA\4_5_16\D2mmSlitSeparation120mmBiPrism\Image';
    directory(21).simulation = 'E:\MATLAB\FBP2020G-
179_470nm_0.200mmslits_120Eta_150Focal_70umSlitWidth_2048x300_0.0065x_1
zmmperpixel.mat';
    directory(21).zsliceInterval = 1;
    directory(21).zsliceStart = 40;
    directory(21).zsliceEnd = 300;
    directory(21).simulationData = 'I0SignalTwoSlits';

    directory(22).experimental = 'Z:\Chris\Experimental
DATA\4_5_16\D3mmSlitSeparation50mmBiPrism\Image';
    directory(22).simulation = 'E:\MATLAB\FBP2020G-
179_470nm_0.300mmslits_50Eta_150Focal_70umSlitWidth_2048x300_0.0065x_1z
mmperpixel.mat';
    directory(22).zsliceInterval = 1;
    directory(22).zsliceStart = 40;
    directory(22).zsliceEnd = 300;
    directory(22).simulationData = 'I0SignalTwoSlits';

    directory(23).experimental = 'Z:\Chris\Experimental
DATA\4_5_16\D3mmSlitSeparation120mmBiPrism\Image';
    directory(23).simulation = 'E:\MATLAB\FBP2020G-
179_470nm_0.300mmslits_120Eta_150Focal_70umSlitWidth_2048x300_0.0065x_1
zmmperpixel.mat';
    directory(23).zsliceInterval = 1;
    directory(23).zsliceStart = 40;
    directory(23).zsliceEnd = 300;
    directory(23).simulationData = 'I0SignalTwoSlits';

    directory(24).experimental = 'Z:\Chris\Experimental
DATA\4_5_16\D5mmSlitSeparation50mmBiPrism\Image';
    directory(24).simulation = 'E:\MATLAB\FBP2020G-
179_470nm_0.500mmslits_50Eta_150Focal_70umSlitWidth_2048x300_0.0065x_1z
mmperpixel.mat';
    directory(24).zsliceInterval = 1;
    directory(24).zsliceStart = 40;
    directory(24).zsliceEnd = 300;
    directory(24).simulationData = 'I0SignalTwoSlits';

    directory(25).experimental = 'Z:\Chris\Experimental
DATA\4_5_16\D5mmSlitSeparation120mmBiPrism\Image';
    directory(25).simulation = 'E:\MATLAB\FBP2020G-
179_470nm_0.500mmslits_120Eta_150Focal_70umSlitWidth_2048x300_0.0065x_1
zmmperpixel.mat';
    directory(25).zsliceInterval = 1;
    directory(25).zsliceStart = 40;
    directory(25).zsliceEnd = 300;
    directory(25).simulationData = 'I0SignalTwoSlits';



%concatenate directory strings with pwd (present working
directory/current folder)
%directories 20 - 25 are stored on CIRL NAS
```

```
    for index = 1:19
        directory(index).experimental = [pwd,
directory(index).experimental];
        directory(index).simulation = [pwd,
directory(index).simulation];
    end
%     for index = 20:25
%         directory(index).simulation = [pwd,
directory(index).simulation];
%     end

end
```

7.L Double Slit Data Comparison of Simulation and Experimental (Initial Set)

The following sections (7.L and 7.M) cover the analysis of the initial double slit data in which there was significant distortion in the images due to misalignment of the camera (camera shake) and optics (axial skew of the pattern) in the experimental setup. Also note that the simulation's axial scale was improperly configured. At the time of simulating this data, the axial period was incorrectly defined as the period of the visibility of the fringes. This resulted in a mismatch of the axial frequencies in the experimental data vs simulated data by a factor of two. The following two sections are a copy of the original data and analysis for documentation purposes. The analysis shows that the PSD method is capable of determining the axial frequency in a process that is independent of the axial skew of the illumination pattern.

The PSD method is used to analyze the lateral and axial frequencies of double slit configuration data. Six total configurations are analyzed; three configurations with varying slit separations (0.200 mm, 0.300 mm, and 0.500 mm) and two biprism positions (50 mm and 120 mm) for each slit separation are presented in this section. The 0.200 mm separation and 50 mm biprism position is the best example out of the three 50 mm biprism configurations. The other two (0.300 mm and 0.500 mm) are highly distorted when the XZ image is stitched together (Figure 4-24 and Figure 4-34). When calculating the lateral frequency for each Z plane, the PSD method only searches for the most dominant frequency within 50% of the theoretical frequency. This filter is appropriate because the single slit data comparison reflects that the experimental data is well below 50% error relative to the theoretical frequency. Refer to Table 7-1 for the list of simulation and experimental parameters.

Table 7-1 Simulation and experimental parameters describing the wavelength of the source, biprism, slit, and converging lens.

λ	$470 * 10^{-6}$	mm wavelength of source
n	1.515	refractive index of the biprism
δ	$0.5 * \frac{\pi}{180}$	radians biprism angle
x_0	0.200, 0.300, and 0.500	mm slit separation
Δ	0.070	mm slit width
η	50 and 120	mm position of biprism
f	150	mm focal length of converging lens

Results acquired using the first configuration (0.200 mm separation and 50 mm biprism position) are presented in Figure 7-1 through Figure 7-5. In Figure 7-1, the low frequency outliers in the first 120mm are sample from a non-resonant plane and their value lowers the overall calculated average for the lateral frequency. By removing these outliers, the adjusted average for Z planes 150 through 300 is 6.2793 lp/mm.

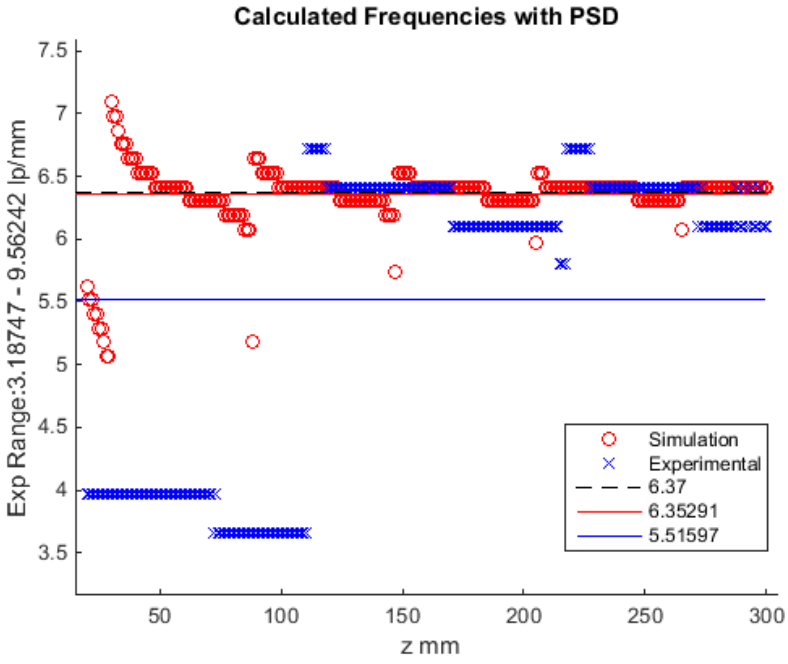

Figure 7-1. Calculated lateral frequencies through Z using PSD method. Experimental Frequency range filter 3.18-9.56 lp/mm.

Figure 7-2 and Figure 7-3 show the correlation between planes of resonance and the PSD intensity in the experimental data.

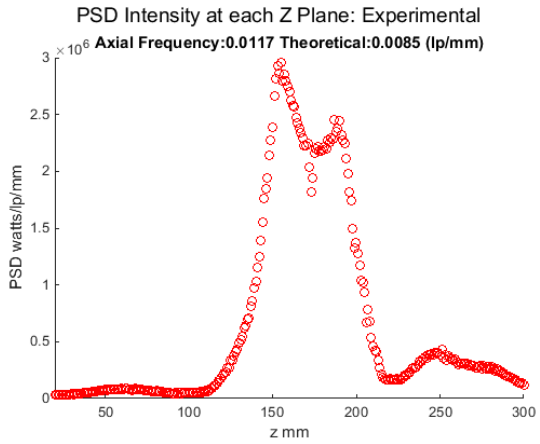

Figure 7-2. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 7-1.

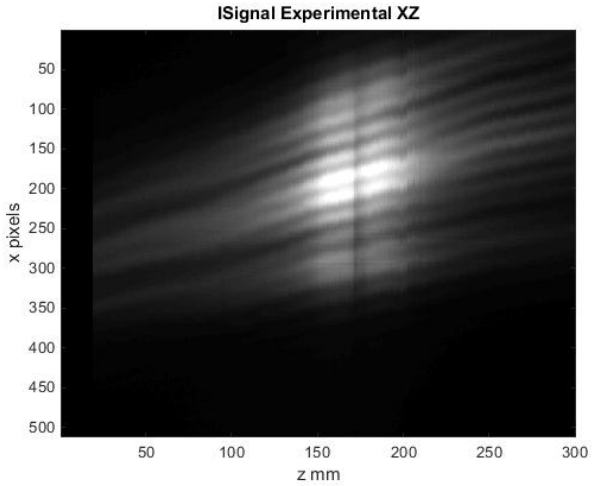

Figure 7-3. Experimental XZ image for 0.200 mm slit separation and 50 mm biprism position.

Figure 7-4 and Figure 7-5 show the correlation between planes of resonance and the PSD intensity in the simulation data.

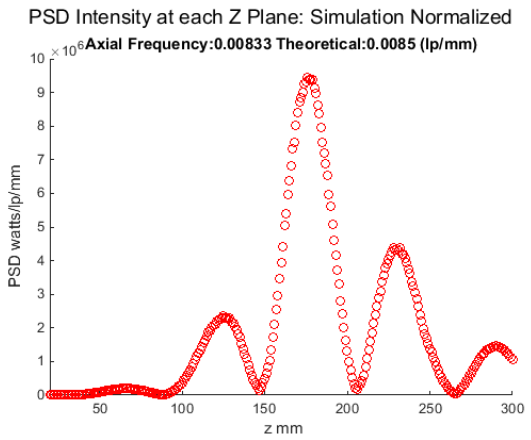

Figure 7-4. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 7-1.

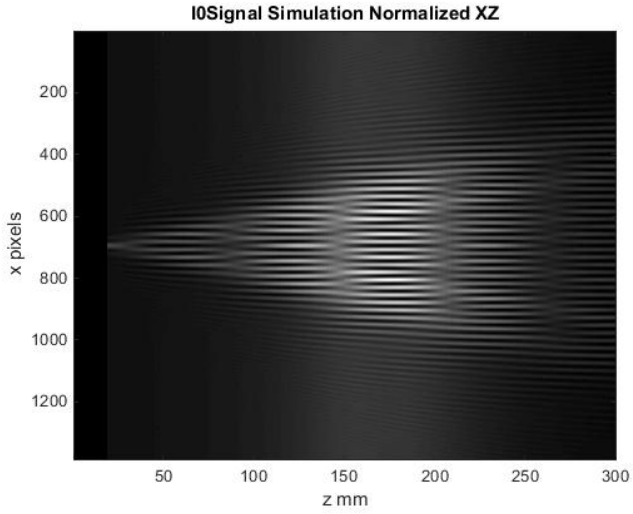

Figure 7-5. Simulation XZ image for 0.200 mm slit separation and 50 mm biprism position.

Results acquired using the second configuration (0.200 mm slit separation, 120 mm biprism position) are presented in Figure 7-6 through Figure 7-10. The outliers in Figure 7-6 correspond to low visibility of the pattern and low PSD intensity indicated by Figure 7-7. The adjusted average for Z planes 150 through 250 is 15.3434 lp/mm.

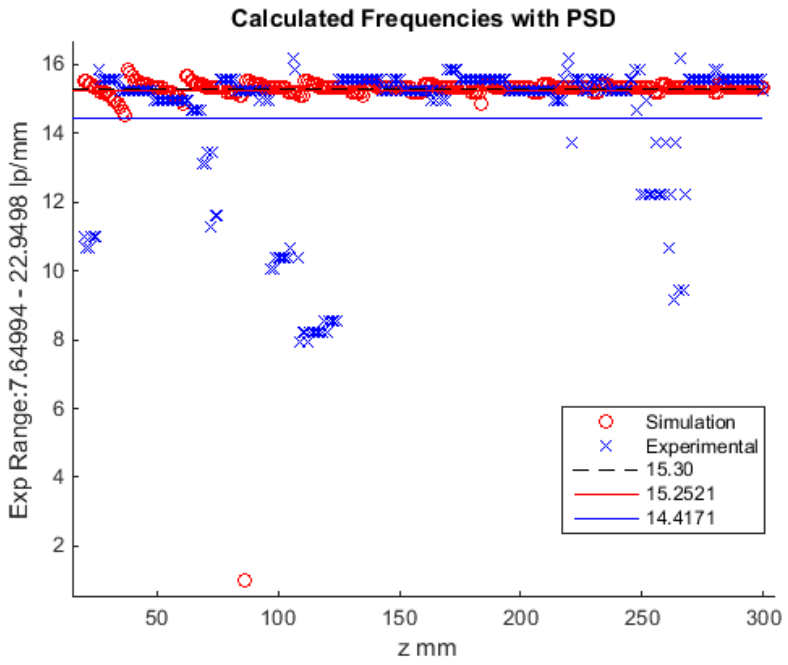

Figure 7-6. Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 7.65-22.95 lp/mm.

Figure 7-7 and Figure 7-8 show the correlation between planes of resonance and the PSD intensity in the experimental data.

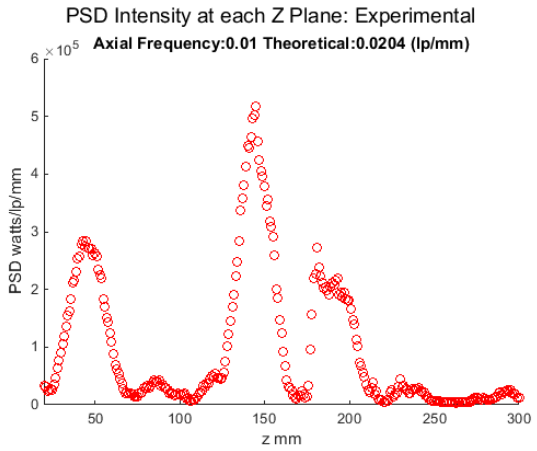

Figure 7-7. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 7-6.

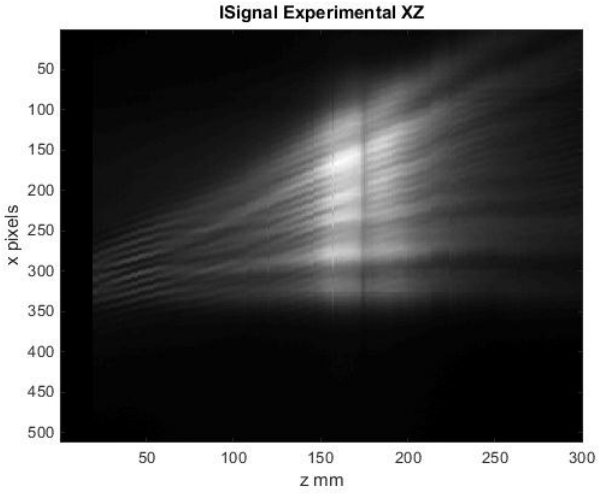

Figure 7-8. Experimental XZ image for 0.200 mm slit separation and 120 mm biprism position.

Figure 7-9 and Figure 7-10 show the correlation between planes of resonance and the PSD intensity in the simulation data.

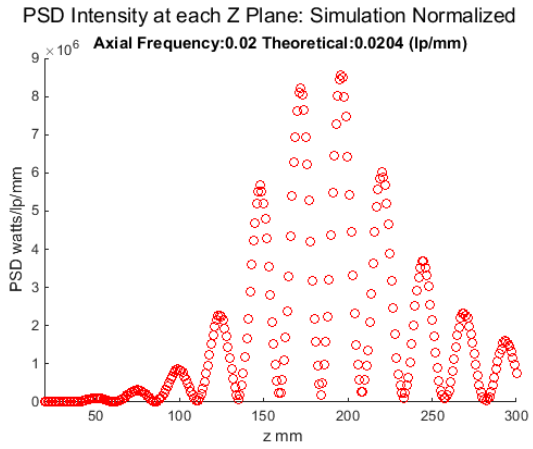

Figure 7-9. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 7-6.

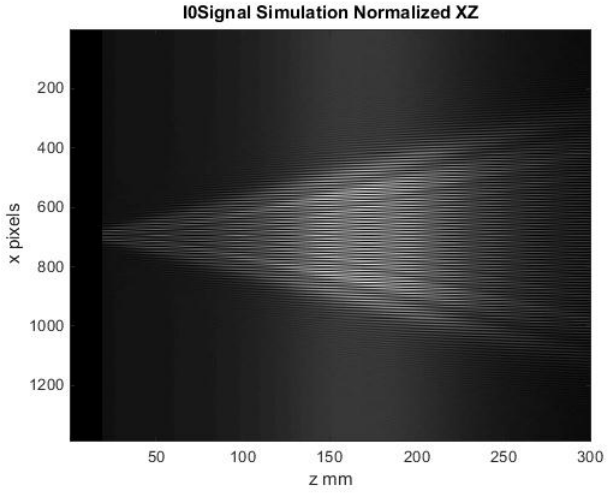

Figure 7-10. Simulation XZ image for 0.200 mm slit separation and 120 mm biprism position.

Results acquired using the third configuration (.300mm slit separation, 50 mm biprism position) are presented in Figure 7-11 through Figure 7-15. The adjusted average for Z planes 150 through 300 is 5.8913 lp/mm.

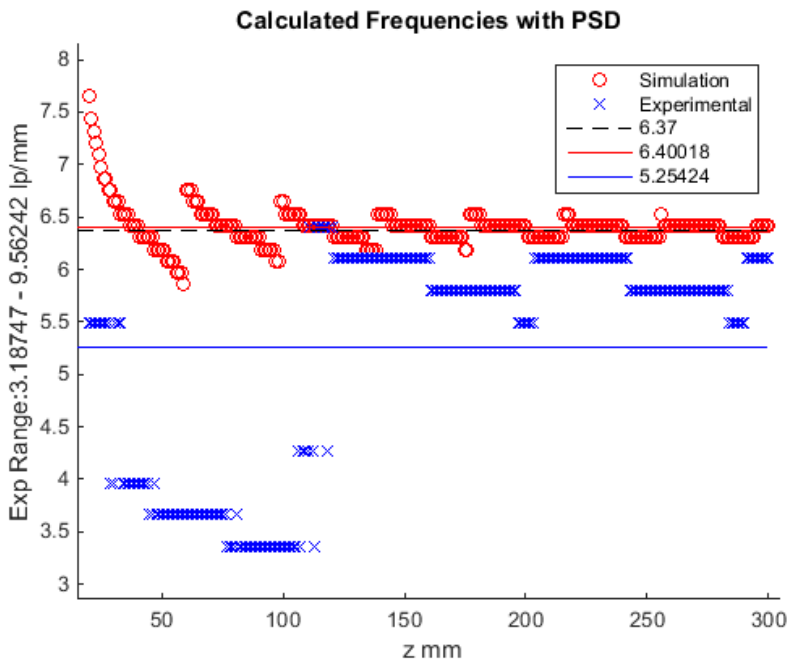

Figure 7-11. Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 3.19-9.56 lp/mm.

Figure 7-12 and Figure 7-13 show the correlation between planes of resonance and the PSD intensity in the experimental data.

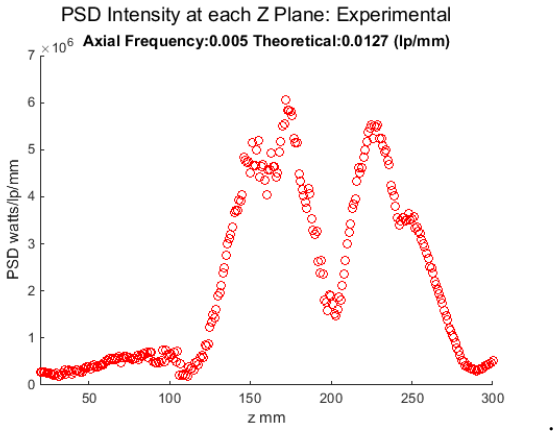

Figure 7-12. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 7-11.

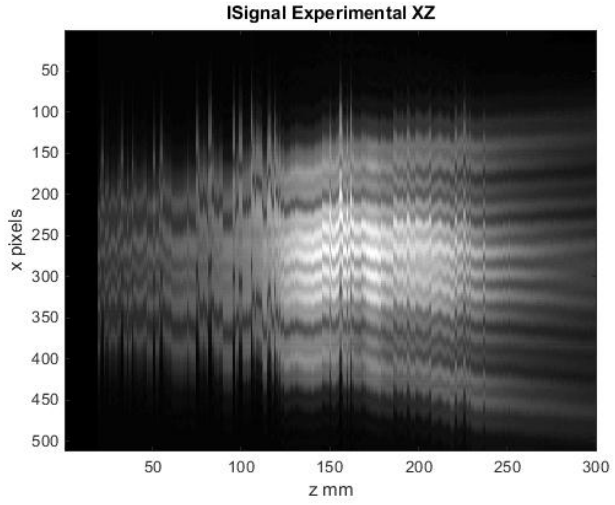

Figure 7-13. Experimental XZ image for 0.300 mm slit separation and 50 mm biprism position.

Figure 7-14 and Figure 7-15 show the correlation between planes of resonance and the PSD intensity in the simulation data.

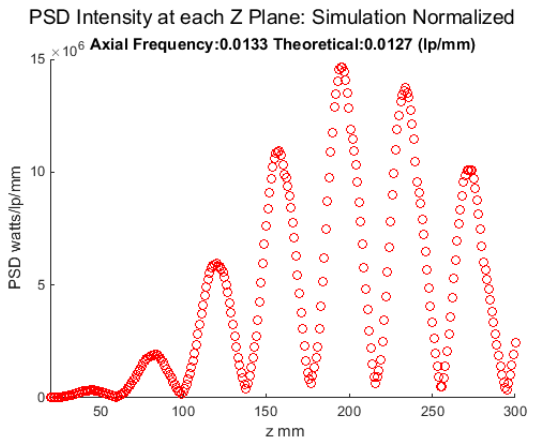

Figure 7-14. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 7-11.

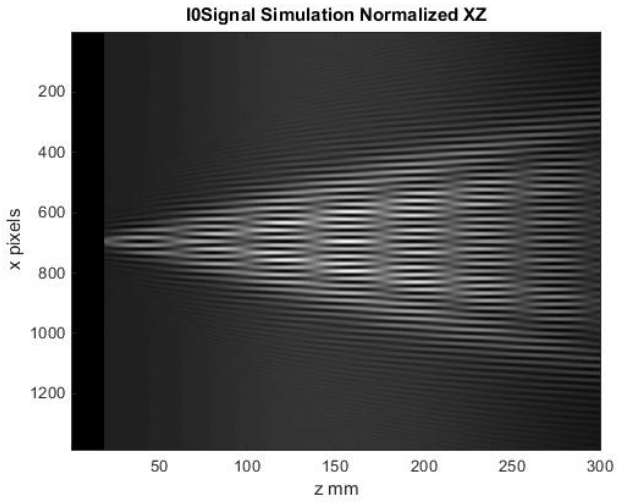

Figure 7-15. Simulation XZ image for 0.300 mm slit separation and 50 mm biprism position.

Results acquired using the fourth configuration (.300 mm slit separation, 120 mm biprism positon) are presented in Figure 7-16 through Figure 7-20. The adjusted average for Z planes 150 through 300 is 15.5983 lp/mm.

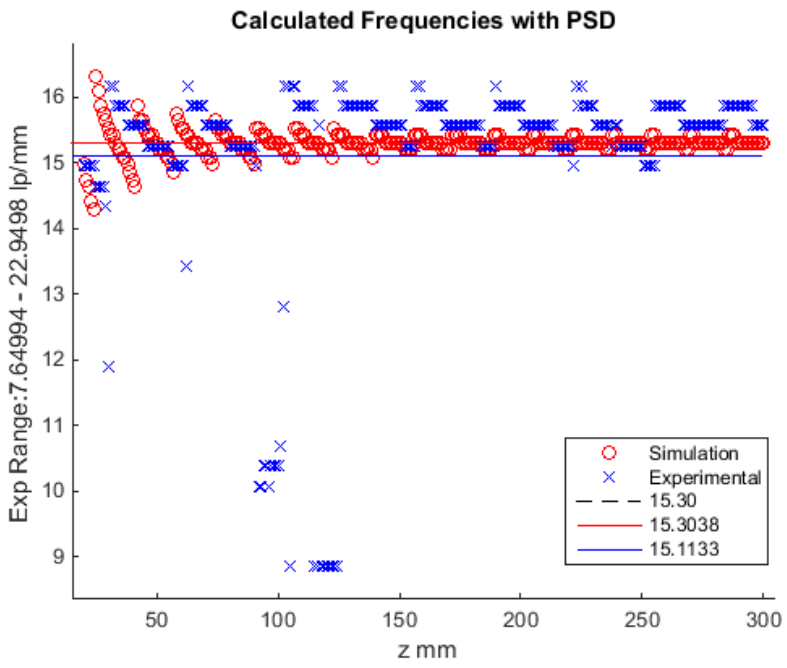

Figure 7-16. Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 7.65-22.95 lp/mm.

Figure 7-17 and Figure 7-18 show the correlation between planes of resonance and the PSD intensity in the experimental data.

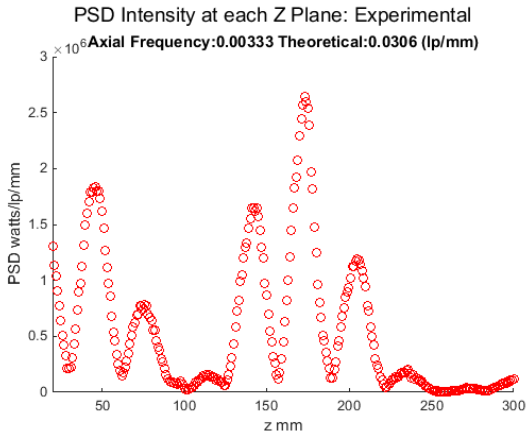

Figure 7-17. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 7-16.

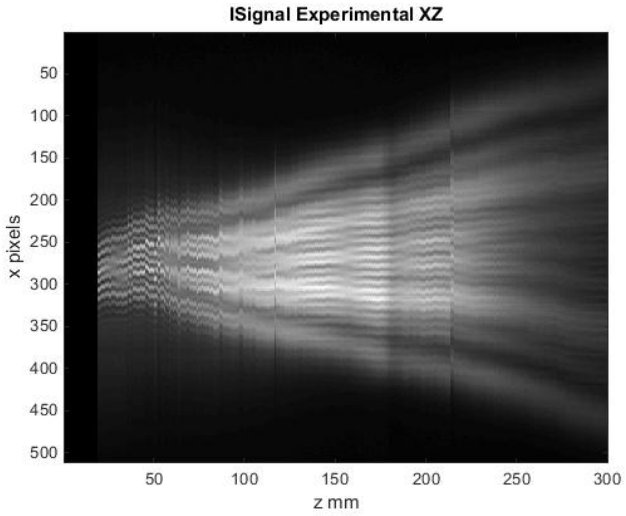

Figure 7-18. Experimental XZ image for 0.300 mm slit separation and 120 mm biprism position.

Figure 7-19 and Figure 7-20 show the correlation between planes of resonance and the PSD intensity in the simulation data.

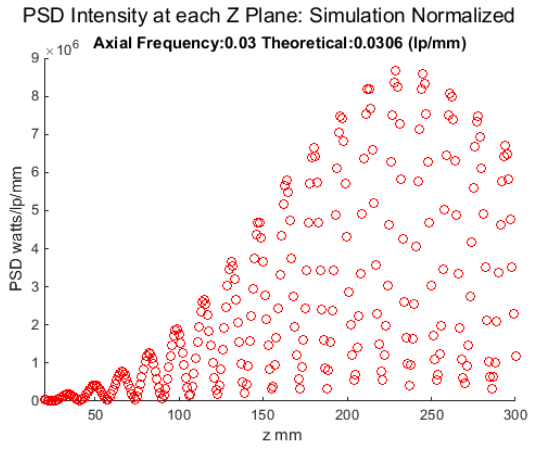

Figure 7-19. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 7-16.

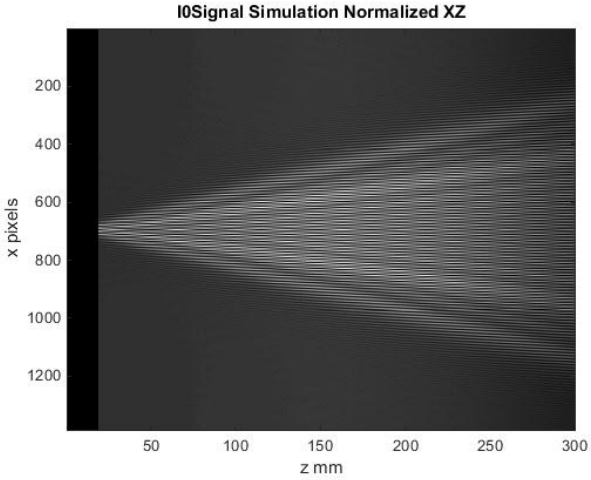

Figure 7-20. Simulation XZ image for 0.300 mm slit separation and 120 mm biprism position.

Results acquired using the the fifth configuration (0.500 mm slit separation, 50 mm biprism position) are presented in Figure 7-21 through Figure 7-25. The adjusted average for Z planes 150 through 270 is 6.1565 lp/mm.

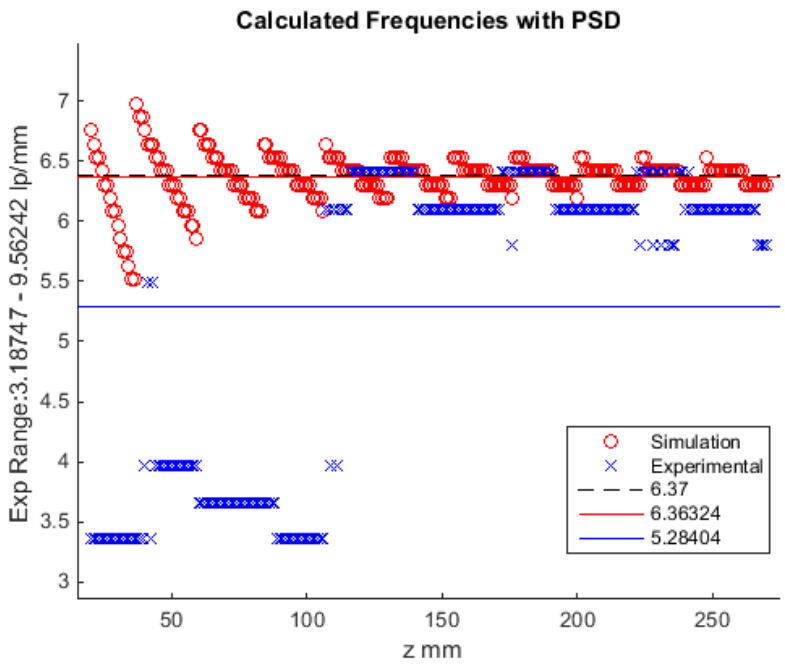

Figure 7-21 Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 3.19-9.56 lp/mm.

Figure 7-22 and Figure 7-23 show the correlation between planes of resonance and the PSD intensity in the experimental data.

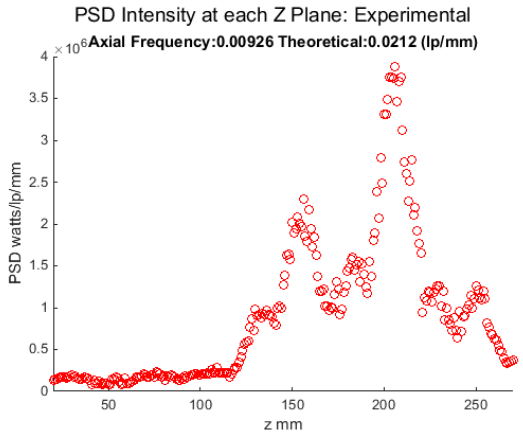

Figure 7-22. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 7-21.

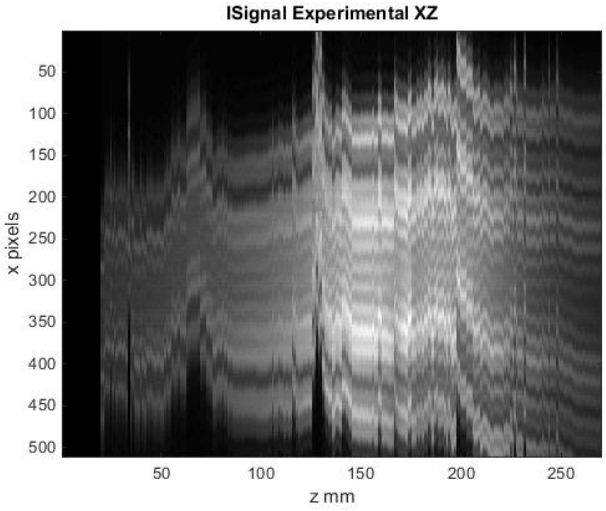

Figure 7-23. Experimental XZ image for 0.500 mm slit separation and 50 mm biprism position

Figure 7-24 and Figure 7-25 show the correlation between planes of resonance and the PSD intensity in the simulation data.

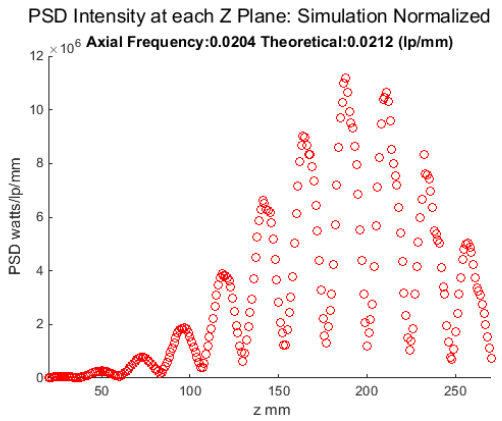

Figure 7-24. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 7-21.

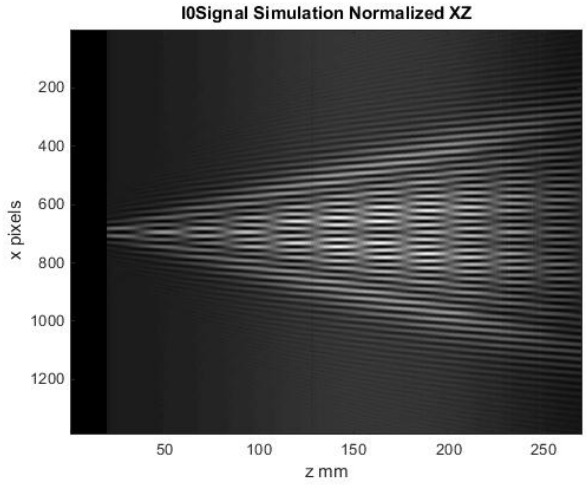

Figure 7-25. Simulation XZ image for 0.500 mm slit separation and 50 mm biprism position.

Results acquired using the sixth configuration (0.500 mm slit separation, 120 mm biprism position) are presented in Figure 7-26 through Figure 7-30. The adjusted average for Z planes 150 through 250 is 15.6063 lp/mm.

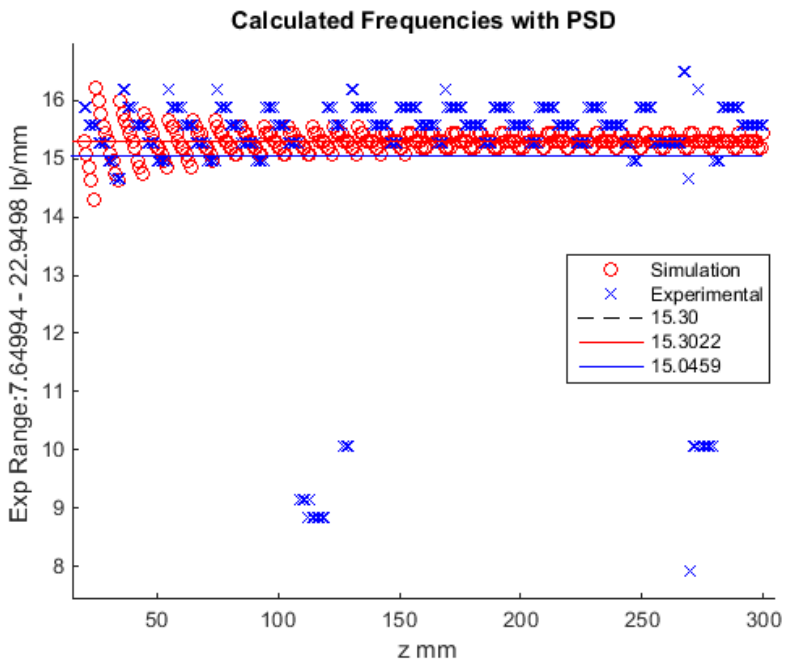

Figure 7-26 Calculated lateral frequencies through Z using PSD method. Experimental frequency range filter 7.65-22.95 lp/mm.

Figure 7-27 and Figure 7-28 show the correlation between planes of resonance and the PSD intensity in the experimental data.

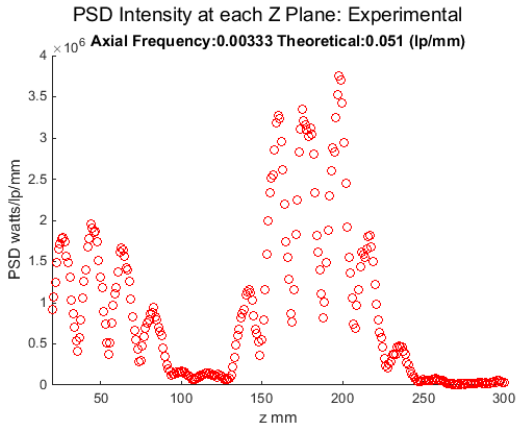

Figure 7-27. The PSD intensities for each Z-plane resulting from analysis of the experimental lateral frequencies at corresponding Z-planes reported in Figure 7-26.

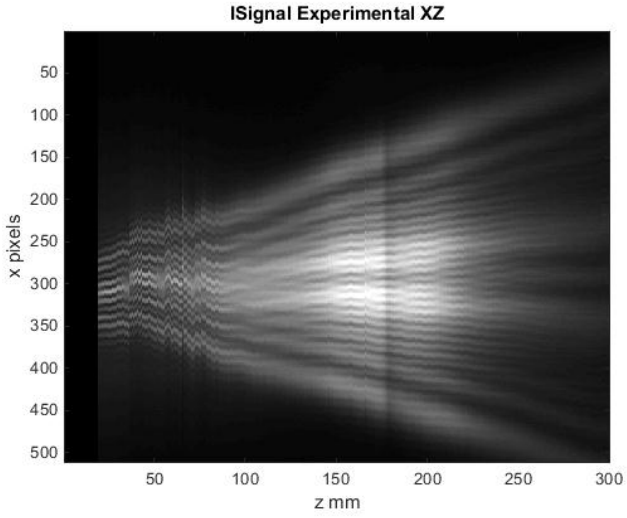

Figure 7-28. Experimental XZ image for 0.500 mm slit separation and 120 mm biprism position.

Figure 7-29 and Figure 7-30 show the correlation between planes of resonance and the PSD intensity in the simulation data.

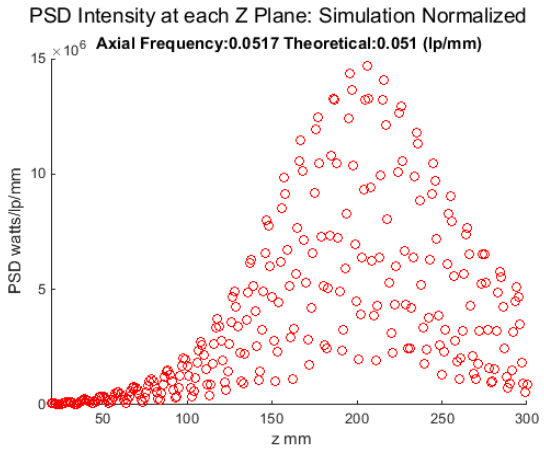

Figure 7-29. The PSD intensities for each Z-plane resulting from analysis of the simulation lateral frequencies at corresponding Z-planes reported in Figure 7-26.

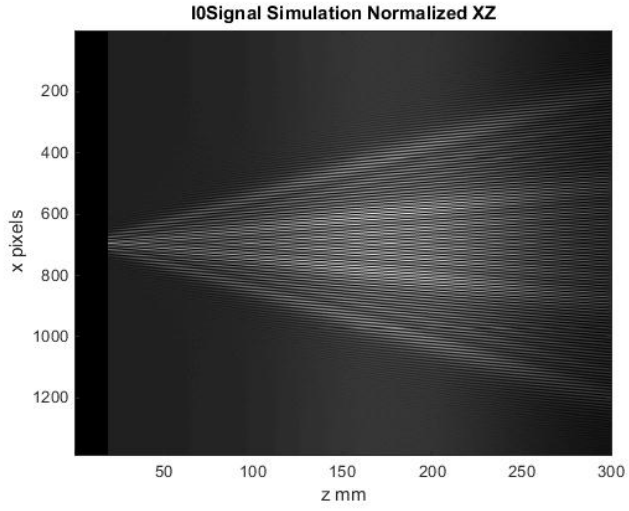

Figure 7-30. Simulation XZ image for 0.500 mm slit separation and 120 mm biprism position.

Table 7-2 summarizes the percent error between the experimental data and the simulation data for the measured average lateral frequency for each of the six double slit configurations.

Table 7-2 Comparison of measured experimental and simulation lateral frequencies for various two slit configurations using the PSD method.

Slit Separation, Biprism Position (mm)	Simulation Lateral Frequency Average (lp/mm)	Experimental Lateral Frequency Adjusted Average (lp/mm)	Percent Error
0.200, 50	6.35291	6.2793	1.1580 %
0.200, 120	15.2521	15.3434	0.5985 %
0.300, 50	6.40018	5.8913	7.9509 %
0.300, 120	15.3038	15.5983	1.9246 %
0.500, 50	6.36324	6.1565	3.2492 %
0.500, 120	15.3022	15.6063	1.9872 %

7.M Additional Axial PSD Analysis of Double Slit Configuration

In simulated data, the PSD method always reports the PSD intensity of the expected lateral frequency of the illumination pattern due to the absence of low frequency noise. In experimental data, the PSD method may report low frequency noise in non-resonant planes because of low visibility of the pattern. The reported low frequency noise reduces the overall average calculated by the PSD method. Two specific conditions were applied to the PSD method to remove the low frequency noise from the calculation: 1) The PSD method was forced to search lateral frequencies in the Power spectrum within +/- 10% of the theoretical lateral frequency, 2) the PSD method was applied to a specific Z range that did not report low frequency noise in the initial analysis from Section 4.E.

Condition 1. The PSD method was forced to +/- 10% of the theoretical lateral frequency so the PSD intensity graph through Z would represent the PSD intensity values for the theoretical lateral frequency. Also, low frequency noise less than 0.01 lp/mm was filtered out while calculating the axial frequency. It is interesting that results of the PSD method for analysis of axial frequency converge to approximately one half of the theoretical axial frequency. After forcing the PSD method to search for the lateral frequency within +/-10% of the theoretical lateral frequency, the PSD intensity graph through Z are similar to the graphs reported in the previous section where +/- 50% was used. Figure 7-31 through Figure 7-33 show the PSD intensity through Z for three configurations with slit separations (0.200 mm, 0.300 mm, 0.500 mm) and a 120 mm biprism position.

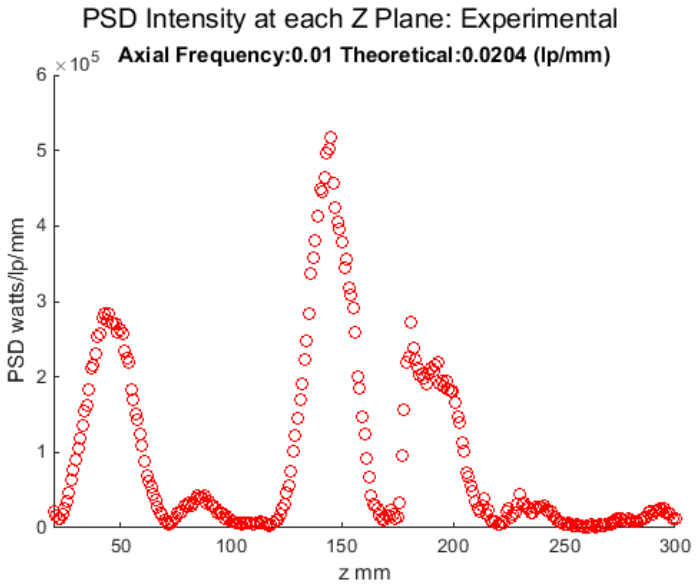

Figure 7-31. PSD Intensity through Z. 0.200mm slit separation. 120 mm biprism position. PSD method with Condition 1 was used to calculate the axial frequency.

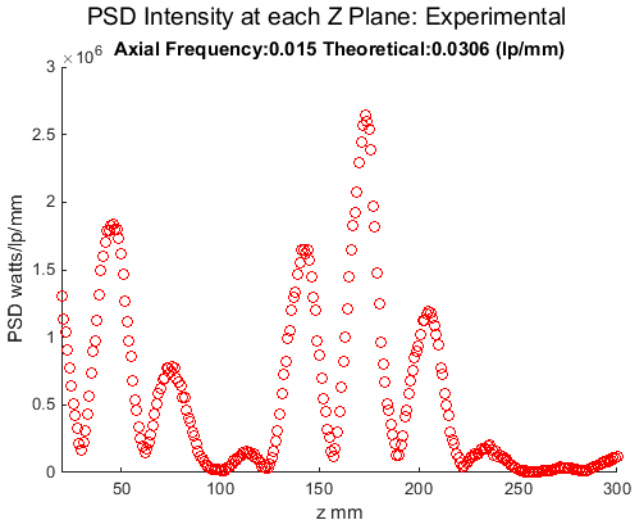

Figure 7-32. PSD Intensity through Z. 0.300mm slit separation. 120 mm biprism position. PSD method with Condition 1 was used to calculate the axial frequency.

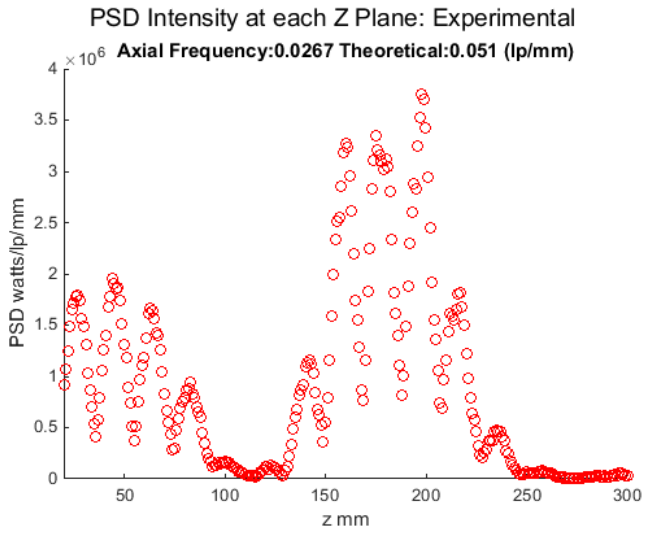

Figure 7-33. PSD Intensity through Z. 0.500mm slit separation. 120 mm biprism position. PSD method with Condition 1 was used to calculate the axial frequency.

Condition 2. Furthermore, the PSD method was applied to a specified *z* range of 125 to 224 mm to calculate the axial frequency (except in the $x_0 = 0.300$ mm, $\eta = 50$ mm case the *z* range was 125 to 250 mm). After viewing the PSD Intensity through Z (i.e. Figure 7-31, Figure 7-32, and Figure 7-33), the sampling of the axial sinusoid is qualitatively better in the center of the *Z* data. In this case, no noise was filtered out. In each of the $\eta = 120$ mm cases, the percent error is 51%.

Table 7-3. Measured axial frequency for a specified z range of 125:224.

$x_0, \eta\ (mm)$	Measured (lp/mm)	Theoretical (lp/mm)	% Error
0.200, 50	0.025	0.0085	194 %
0.200, 120	0.01	0.0204	51.0 %
0.300, 50	0.00794	0.0127	37.5 %
0.300, 120	0.015	0.0306	51.0 %
0.500, 50	0.01	0.0212	52.8 %
0.500, 120	0.025	0.051	51.0 %