3-9-2005

Using Liquid Crystal Spatial Light Modulators for Closed Loop Tracking and Beam Steering with Phase Holography

Michael J. Perry

USING LIQUID CRYSTAL SPATIAL LIGHT MODULATORS FOR
CLOSED LOOP TRACKING AND BEAM STEERING WITH
PHASE HOLOGRAPHY

THESIS

Michael J Perry, Captain, USAF

AFIT/GEO/ENG/05-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Using liquid crystal spatial light modulators for closed loop tracking and beam steering with phase holography

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Michael J Perry, B.S.E.E.

Captain, USAF

March 2005

AFIT/GEO/ENG/05-02

USING LIQUID CRYSTAL SPATIAL LIGHT MODULATORS FOR
CLOSED LOOP TRACKING AND BEAM STEERING WITH
PHASE HOLOGRAPHY

Michael J Perry, B.S.E.E.

Captain, USAF

Approved:

/signed/	9 Mar 2005
Maj Matthew Goda, PhD (Chairman)	date
/signed/	9 Mar 2005
Lt Col Thomas Alley (Member)	date
/signed/	9 Mar 2005
Dr. Robert Hengehold (Member)	date

AFIT/GEO/ENG/05-02

Abstract

The use of Optical Phased Array (OPA) technology in laser communications and target tracking systems offers advantages of reduction in size, weight, and power consumption of the optical steering mechanisms. This thesis examines the use and flexibility of a nematic liquid crystal spatial light modulator (SLM) in a number of beam steering applications including closed loop tracking. The demonstrations use a 512 x 512 SLM to control 632.8nm laser light from a Helium-Neon source.

Control of the SLM is achieved by loading 256 level bitmaps dictating the index of refraction desired. This relation between applied voltage and phase modulation is commonly known as the phase response. Nematic liquid crystals will typically have a non-linear variation in phase in response to a change in applied voltage. As a result, before the system can be used, the SLM's phase response is quantified to ensure proper manipulation of index of refraction.

Once complete, a model of each experiment is created for verification purposes followed by execution of the demonstration. Current beam steering/splitting techniques typically utilize the OPA as a thin phase grating using a modulo 2π representation of a linear phase shift to steer or split the beam. Demonstration and analysis show that there are fringing effects and diffraction limitation issues when using the grating technique for simultaneous beam splitting and steering. To overcome these limitations, this thesis introduces a phase hologram scheme that is developed through an iterative Fourier transform algorithm.

Experimentation is compared to the results predicted by the models. Performance of the phase grating during beam splitting, steering, and variation of intensity closely followed model results. Beam tracking performed well, however there was significant error induced by the SLM due to age of the device. Finally, the phase hologram experienced limited success in beam steering and beam splitting. Several

holograms were created that split and steered the beam in various directions. Results show that although a beam can be successfully split and steered, there is significant loss in intensity. This suggests that further refinement of the iterative algorithm is required.

Acknowledgements

The effort put into a successful thesis extends far beyond the author. Although the conducting of research and documenting these efforts rested upon my shoulders, it couldn't have been done without the support of others. First I would like to thank my thesis advisor, Major Matthew Goda, Ph.D., for his guidance, wisdom, and continually encouraging me to push the research envelope. I also appreciate the enthusiasm and visionary efforts of my sponsoring organization (AFRL/SNJM); I'm indebted to them for loaning me the equipment needed for the experiments. I am grateful also to Scott Harris, Ph.D. for his endless patience in helping me understand the LabVIEW software and providing me LabVIEW code to bypass BLINK. Without his inputs, chances of success would have been minimal. I owe much thanks to my parents for giving me a foundation that has enabled me to get where I am today. Although it's been a long road to get here, the journey began with them sparking my interest in science at a young age. And finally, I want to acknowledge the one who supported me the most, my wife. Thank you for picking up all the family responsibilities in addition to pursuing your own career, ensuring that family life remained as normal as possible. Your unfaltering support will never be forgotten.

Michael J Perry

Table of Contents

List of Figures

x

xi

Using liquid crystal spatial light modulators for closed loop tracking and beam steering with phase holography

I. Introduction

1.1 Research Motivation

In systems currently fielded by the Air Force, optical beam steering is commonly accomplished via mechanical systems such as mirrors on gimbals. These systems tend to suffer from several drawbacks such as movable parts, power requirement, weight, size, and high costs, so a non-mechanical beam steering solution would be preferred. Interest in non-mechanical beam steering has greatly increased in recent years with new advances in liquid crystal technology.

There exist several alternative techniques to accomplish non-mechanical beam steering and one area where there has been a lot of progress is in the use of optical phased arrays (OPA). One such device is a liquid crystal on silicon (LCOS) spatial light modulator (SLM). They are created using very large scale integration (VLSI) technology combining the electro-optical performance characteristics of the liquid crystal materials with silicon-based circuitry creating a small flexible device with low power requirements [5]. These components have several properties that are required for laser beam steering including small pixel pitch and high optical efficiency.

From an application point of view non-mechanical beam steering and beam shaping are desired features in free-space optical communication, laser pointing, laser-based imaging, and illumination and tracking systems. Implementation on airborne platforms, particularly UAVs and satellites, is of particular interest due to the requirements of low weight and volume and low power consumption. The OPA technology may offer enhanced flexibility in future laser pointing and tracking systems. By combining beam shaping and tracking new capabilities can be obtained by altering the

1

divergence of the beam. Another feature may be obtained by splitting the beam into sub-beams in order to track, or communicate to, several targets simultaneously.

1.2   Problem Statement

This thesis focuses on developing techniques in the laboratory environment that use spatial light modulators for laser beam control in optical communication or target tracking scenarios.

1.3   Research Objectives

1. Characterize SLM behavior with concern for diffraction efficiency and non-linear behaviors.

2. Demonstrate beam steering and splitting techniques by applying appropriate phase gratings to the SLM.

3. Develop a methodology to split a beam and control the intensity of each sub-beam.

4. Demonstrate that an SLM can track an incoming laser in a closed loop fashion.

5. Explore the viability of using holograms to create phase maps on SLMs to split and steer beams, comparing this method to traditional blazed grating phase maps currently in use.

1.4   Methodology

Gimballed mirror systems are currently the device of choice for beam steering applications in laser communications and target tracking. In addition to the undesirable characteristics listed above, gimballed mirrors do not have the ability to split beams to allow a source to communicate with multiple receivers simultaneously. In this work a SLM was used as a replacement for the gimballed system in an optical path system to show that they have the ability to overcome all the aforementioned limitations of the gimballed mirror.
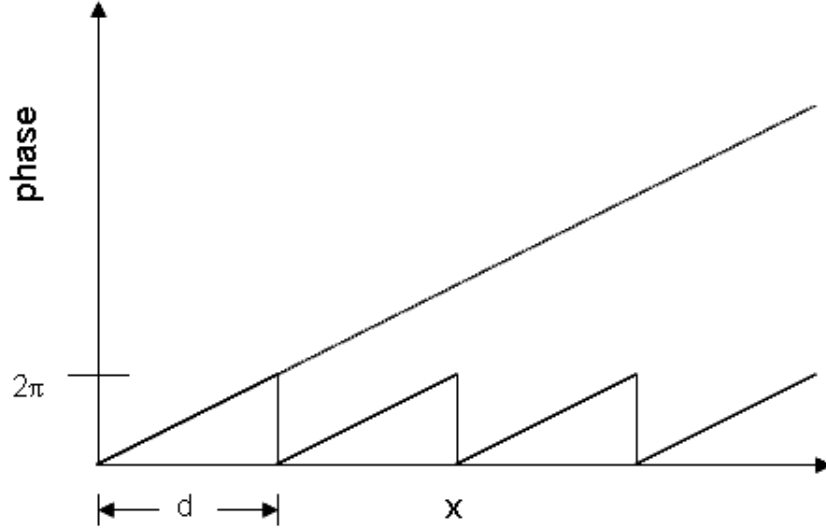
Figure 1.1:   Modulo 2π representation of a tip/tilt phase distribution applied to the SLM

The SLM is composed of an array of pixels where each pixel has a controllable index of refraction. Changing these individual pixel indices allows the wavefront passing through the device to be altered as specified by the user. By using a spatial light modulator (SLM) as an OPA a complex phase modulation is applied onto an outgoing wavefront. Referring to Figure 1.1, a saw-tooth shaped phase pattern (modulo 2π representation of a tip/tilt phase distribution) can be readily applied to the SLM causing a deflection of the beam. By applying more complex patterns, the beam can be shaped or split it into multiple beams.

The patterns were applied to the SLM via a phase screen. To produce phase screens, the angle of the target in reference to the source was determined and a corresponding tip/tilt value was determined. This value was then translated into a modulo 2π gray scale image utilizing 256 gray values. Resultant phase screens can then be loaded onto the SLM so that the phase of the wavefront is manipulated in such a way that the beam is redirected to the target. Similar methodology is used for beam splitting and target tracking. Further phase screen generation topics will be discussed in Chapter 2.
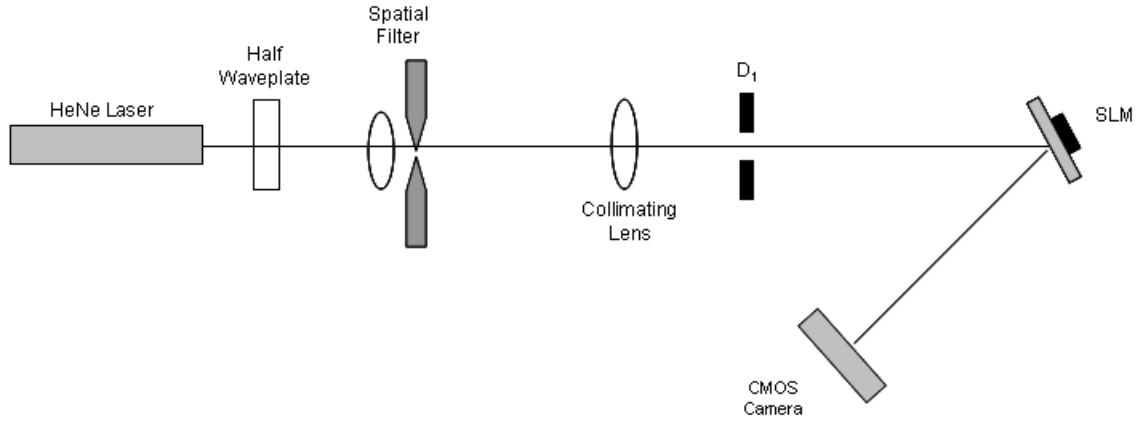
Figure 1.2:   Optical setup for beam splitting/steering experiments

1.5   Experimentation

An optic test was performed in the lab to show that SLMs are a viable substitute for gimballed mirror systems. When implemented, the steering and tracking systems consist of a laser source emitting light onto an SLM that reflects the light to a sensor. For the source, a helium-neon laser with 15mW output power was used. Passing the HeNe's beam through a spatial filter approximates a point source which was then collimated using a lens before reflecting off the SLM and into the camera. Figure 1.2 shows an optical diagram of the experimental setup for beam splitting/steering and Figure 1.3 shows the setup for beam tracking.

To control the beam steering system, a computer interface was used that requires 0-255 gray scale level images to communicate the phase modulation to the SLM. Phase screens representing a modulo 2π implementation of a desired tip/tilt angle can be loaded to the SLM in the gray level format. To maximize efficiency of the phase screens, the pixel values of the screen must be modified based on the characteristics of the SLM. These characteristics include diffraction due to the geometry of the system and a nonlinear phase delay behavior that influence the steering and splitting behavior of the beam in negative ways.

4
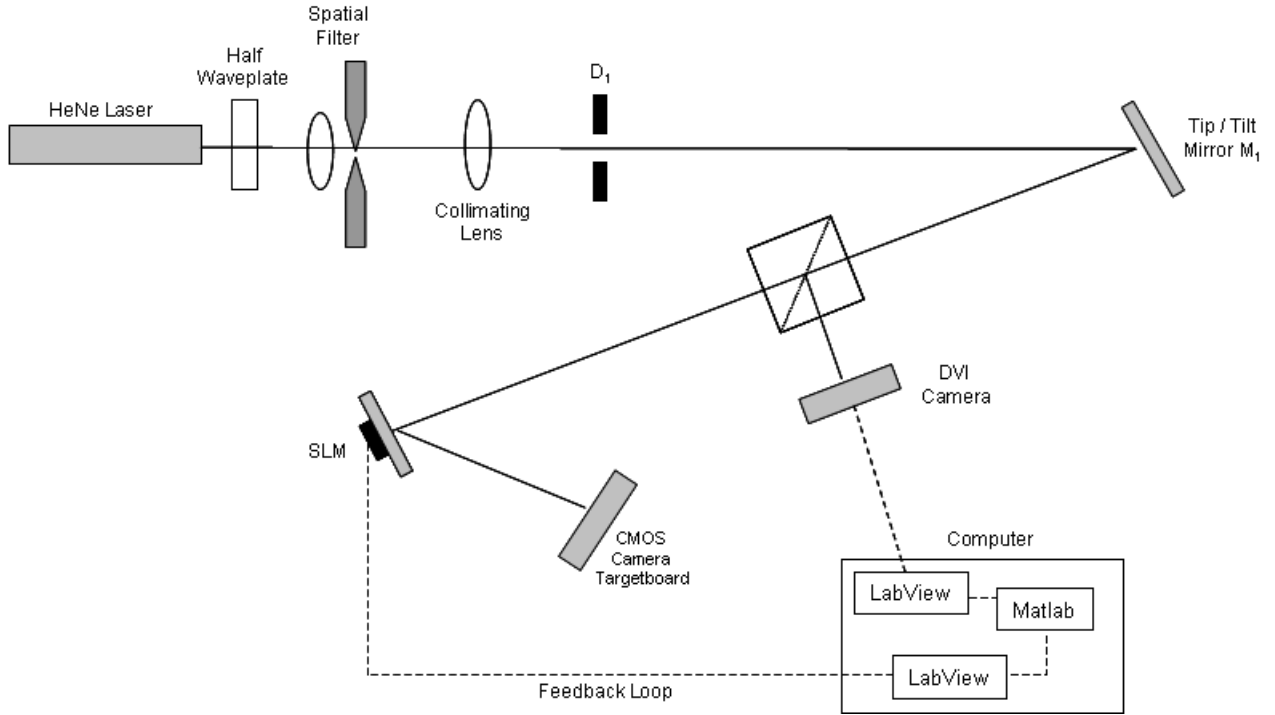
Figure 1.3:   Optical setup for beam tracking and stabilization experiments

1.6   Simulation

Experimentation with the SLM was aided and verified with the use of computer simulations. Prior to conducting an experiment in the lab, it was simulated in Matlab using the mathematical theory described in Chapter 2. Creating phase screens and loading them onto the SLM can be a time intensive process; simulation greatly increases time efficiency when examining the effects of system attributes. A second benefit of simulations is that they predict and verify overall system behavior and performance. Each experiment was first demonstrated in Matlab before execution in the lab. Experimental results were then verified with the Matlab simulations.

1.7   Implications

This research demonstrates the ability of an SLM to split, steer, vary the relative intensity, and track an incoming laser beam. The focus of this work ties in with laser communications and target tracking systems that are looking for a viable

5

substitute for gimballed mirror steering devices. In particular, through an initiative with AFRL/VSSV, the MILSATCOM community is exploring SLMs as an option for communication satellites requiring multiple high-bandwidth links. Use of SLMs on satellites will eliminate risk of mechanical failure, and reduce size, weight, and power requirements of the of the satellite bus. The same benefits can be extended to communication links on air breathing platforms as well again, increasing efficiency of the platform.

1.8   Summary

Non-mechanical beam steering is of great interest in many optical systems. Today beam steering is most often accomplished by the use of gimballed mirrors, which suffer from many drawbacks. The use of SLMs as an optical phased array is one alternative that provides a lot of promise. Laser beams are controlled by placing a phase screen on the SLM representing a modulo 2π variation of tilt required to steer the beam to any given angle. SLMs also offer the added ability of beam splitting, steering multiple beams, and varying the intensity of each beam by simply altering the phase screen as will be discussed in Chapters 3. Prior to use, the SLMs are characterized so that abnormalities, such as the nonlinear phase relationship to gray levels, can be compensated.

II. Theory

2.1 Introduction

The idea of steering light has been around for well over 100 years. In an article published in 1888, Lord Rayleigh suggested that it was at least theoretically possible to shift energy out of the zeroth order into one of the higher-order spectra. Building upon this idea, in 1910 Robert Williams succeeded in ruling grooves in a controlled shape commonly known as a blazed grating (Figure 2.1b) [9]. Given that blazed gratings are rigid in form and can only move light to a position defined by the grating geometry, spatial light modulators used as an optical phased array (OPA), seek to mimic the blazed grating capability but with the added capability of being fluid in geometry. Thus being able to steer light to a myriad of positions limited only by the aperture size and the properties of the liquid crystal.

2.2 Diffraction Theory

The ability to modulate light with liquid crystal for non-mechanical beam steering can be categorized according to the physical mechanisms used to redirect light: refraction and diffraction. Optical phased arrays can be used in refractive mode if no resets are used and the phase ramps continuously across the aperture. With no phase resets, grating dispersion is not present, thus advantageous for steering broadband radiation. However, this technique is limited to steering angles on the order of tens of milliradians [5].

Diffractive beamsteerers can be implemented with an optical phased array analogous to some radar systems. The diffractive optical phased array can be thought of as a quantized multiple level phase grating. As the number of phase levels in the array increases, a higher diffraction efficiency is achieved resulting in greater steering angles. The major advantage of diffractive devices is that the addressable angles can be randomly accessed [1].

The differences between refraction and diffraction can be made clear with a more in depth look at theory. Refraction can be defined as the bending of light rays
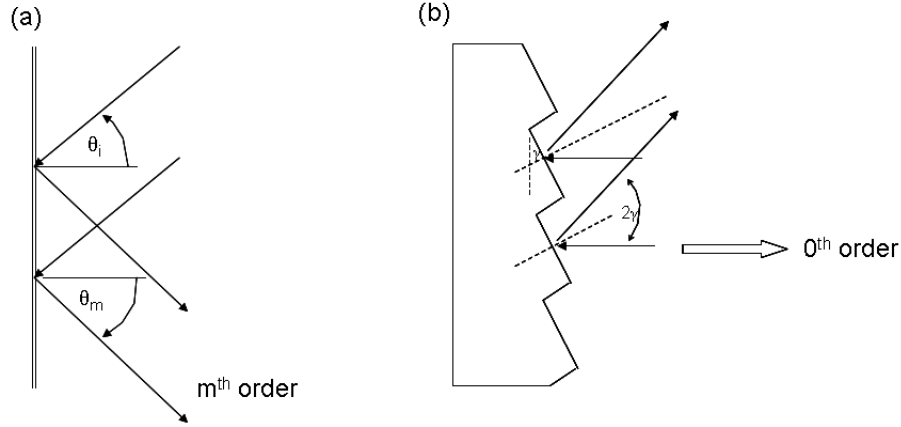
Figure 2.1:   (a) A reflection grating. (b) Blazed grating. [9]

that takes place when they pass through a region in which there is a gradient of the local velocity of propagation of the wave. An example of this occurs when a light wave encounters a sharp boundary between two regions having different refractive indices. As a result, the incident light rays are bent at the interface. The angles of incidence and refraction are related by Snell's law

$$n_1 \sin \theta_1 = n_2 \cos \theta_2 \tag{2.1}$$

Sommerfield defines the term diffraction as any deviation of light rays from rectilinear paths which cannot be interpreted as reflection or refraction [6]. Diffraction is caused by the confinement of the lateral extent of a wave, and is most appreciable when that confinement is to sizes comparable with the wavelength of the radiation being used.

The idea of diffraction is based upon a long history of research that first began in 1665 by a physicist named Francesco Grimaldi. Over the years, work by Huygens, Young, and Fresnel led to the Huygens-Fresnel equation

$$U(x,y) = \frac{z}{j\lambda} \int \int_{\Sigma} U(\xi, \eta) \frac{e^{jkr}}{r^2} d\xi d\eta \tag{2.2}$$

8

where

$$r = \sqrt{(x - \xi)^2 + (y - \eta)^2 + z^2} \qquad (2.3)$$

$U(\xi, \eta)$ is the initial complex wave, and (x, y) are the transverse coordinates of the wave propagated a distance z. By taking the binomial expansion of the root in equation 2.3, r can be rewritten as

$$r \approx z[1 + \frac{1}{2}(\frac{x - \xi}{z})^2 + \frac{1}{2}(\frac{y - \eta}{z})^2]. \qquad (2.4)$$

Plugging this into equation 2.2 and simplifying leads to

$$U(x, y) = \frac{e^{jkz}}{j\lambda z} \int \int_{\Sigma} U(\xi, \eta) e^{\frac{jk}{2z}[(x - \xi)^2 + (y - \eta)^2]} d\xi d\eta \qquad (2.5)$$

Then by the definition of the Fourier transform, the equation becomes

$$U(x, y) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{\lambda z}(x^2 + y^2)} \mathcal{F}\{U(\xi, \eta)\}. \qquad (2.6)$$

Thus, the behavior of light can be predicted at a certain propagation distance z, which is the basis of the simulation tool used to predict the behavior of the optical system being researched in this thesis [4, 6, 7, 9].

The final step in deriving the diffraction theory for SLM optical phased arrays is to discuss diffraction gratings and blazed gratings. The equation

$$d \sin \theta = m\lambda \qquad (2.7)$$

is known as the grating equation for normal incidence [8]. The value d is the period of the grating and the values of m specify the *order* of the various principal maxima. For a source having a broad band spectrum, the grating equation is dependent on λ , and so for any value of $m \neq 0$ the various colored images of the source corresponding to slightly different angles (θ_m) spread out into a continuous spectrum [9]. Given that this work only incorporates a narrow band source, the same theory holds true however

9

there will only be a single value for θ_m. Taking both transmission and reflection into consideration, the grating equation becomes

$$d(\sin\theta_m - \sin\theta_i) = m\lambda \tag{2.8}$$

Most modern gratings are of the blazed variety. The angular positions of the nonzero orders, θ_m-values, are determined by d, λ, and the angle of incidence θ_i. However, θ_m and θ_i are measured from the normal to the grating plane and not with respect to the individual groove surfaces. On the other hand, the location of the peak in the single-facet diffraction pattern corresponds to specular reflection off that face, for each groove. It is governed by the blaze angle γ and can be varied independently of θ_m [9].

Consider the situation represented in Figure 2.1 when the incident wave is normal to the plane of a blazed reflection grating (when $\theta_i = 0$). For specular reflection $\theta_i - \theta_r = 2\gamma$, most of the diffracted radiation is concentrated about $\theta_r = -2\gamma$. θ_r is negative because the incident and reflected rays are on the same side of the grating normal. This will correspond to a particular nonzero order, on the side of the central image, when $\theta_m = -2\gamma$; in other words, $a\sin(-2\gamma) = m\lambda$ for the desired λ and m.

2.3 Liquid Crystal Theory

2.3.1 Background.   Liquid crystals are composed of large, anisotropic molecules existing in a mesotropic phase possessing physical properties that are between conventional fluids and solids. They are fluid-like, yet the arrangements of molecules within them exhibit structural orders, at times being highly correlated. Three types of liquid crystals have been discovered so far: (i) thermotropic, (ii) lyotropic, and (iii) polymeric. Among these three, the thermotropic liquid crystals have been studied extensively and their applications have reached a mature stage.

Lyotropic liquid crystals are receiving increasing scientific attention because of the way they reflect the unique properties of their constituent molecules. A lyotropic

liquid crystal is a group of liquid crystal assemblies that consists of two or more components and exhibits liquid crystalline properties in certain concentration ranges. In the lyotropic phases, solvent molecules fill the space around the compounds to provide fluidity to the system. In contrast to thermotropic liquid crystals, these lyotropics have another degree of freedom of concentration that enables them to induce a variety of different phases.

Polymeric liquid crystals are a class of materials that combine properties of polymers with those of liquid crystals. These hybrids show the same mesophases characteristic of ordinary liquid crystals, yet retain many useful and versatile properties of polymers. These liquid crystals are candidates for electronic devices, such as hand-held games, and ultra-high-strength materials [10].

SLMs fall into the third category of thermotropic liquid crystals which will be the focus of the remainder of this thesis. Thermotropic liquid crystal phases are observed in pure compounds and mixtures. As the temperature increases, these compounds go through a series of phase transitions: solid crystal $\rightarrow$ smectic liquid crystal $\rightarrow$ nematic liquid crystal $\rightarrow$ isotropic liquid $\rightarrow$ vapor phase. Exceptions exist for some compounds: their thermotropic liquid crystal phase appears only during the cooling process from the isotropic phase, but not on the heating process. This kind of phase transition is called monotropic [10]. Although the smectic phase possess the highest degree of order, it is the nematic and chiral nematic (also known as the cholesteric phase) phases which have the greatest number of electro-optical applications.

Due to the orientational ordering of the anisometric molecules, the smectic and nematic liquid crystals are uniaxially symmetric, with the optic axis parallel to the axes of the molecules. The anisotropy of the refractive index is characterized by $\Delta n = n_e - n_o$. In nematic and smectic liquid crystals Δn is typically greater than 0, making them positive uniaxial [3, 15].

2.3.2 Molecular Orientation Effects. Following the discussion of liquid crystals found in [3], it is found that liquids composed of anisotropic molecules typically

11
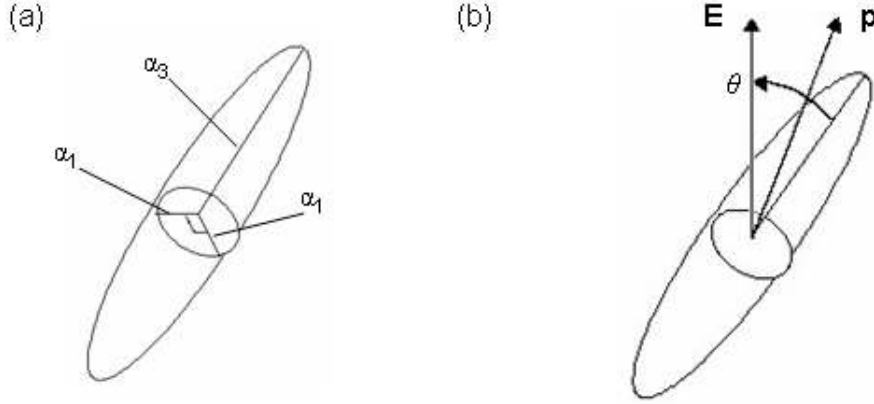
Figure 2.2: (a) A prolate spheroidal molecule. (b) The dipole moment **p** induced by an electric field **E**. [3]

possess a large value for n_e. The origin of this nonlinearity is the tendency of molecules to become aligned in the electric field of an applied optical wave. The optical wave then experiences a modified value of the refractive index because the average polarizability per molecule has been changed by the molecular alignment.

Nematic liquid crystals are typically cigar-shaped molecules (a prolate spheroid) as shown in Figure 2.2. Consequently the polarizability α_3 experienced by an optical field that is parallel to the symmetry axis is larger than the polarizability α_1 experienced by a field that is perpendicular to its symmetry axis, that is,

$$\alpha_3 > \alpha_1 \tag{2.9}$$

Part b of Figure 2.2 shows the molecule being subjected to a dc electric field. Since α_3 is larger than α_1, the component of the induced dipole moment along the molecular axis will be disproportionately long. The induced dipole moment **p** thus will not be parallel to **E**, but will be offset from it in the direction of the symmetry axis. A torque

$$\tau = p \times E \tag{2.10}$$

will thus be exerted on the molecule. This torque is directed in such a manner as to twist the molecule into alignment with the applied electric field.

12

The tendency of the molecule to become aligned in the applied electric field is counteracted by thermal agitation, which tends to randomize the molecular orientation. The mean degree of molecular orientation can be calculated through the use of the Boltzmann factor [3, 15]. In order to do so, we must first calculate the potential energy of the molecule in the applied electric field. If the applied field is changed by an amount dE, the orientational potential energy is changed by the amount

$$dU = -p \cdot dE = -p_3 dE_3 - p_1 dE_1 \tag{2.11}$$

where we have decomposed E into its components along the molecular axis (E_3) and perpendicular to the molecular axis (E_1). Since

$$p_3 = \alpha_3 E_3 \tag{2.12}$$

and

$$p_1 = \alpha_1 E_1 \tag{2.13}$$

it is found that

$$dU = -\alpha_3 E_3 dE_3 - \alpha_1 E_1 dE_1 \tag{2.14}$$

which can be integrated to give

$$U = -\frac{1}{2}(\alpha_3 E_3^2 + \alpha_1 E_1^2) \tag{2.15}$$

Now, introducing the angle θ between E and the molecular axis (see Figure 2.2b), it is found that the orientational potential energy is given by

$$U = -\frac{1}{2}[\alpha_3 E^2 \cos^2 \theta + \alpha_1 E^2 \sin^2 \theta] = -\frac{1}{2}\alpha_1 E^2 - \frac{1}{2}(\alpha_3 - \alpha_1)E^2 \cos^2 \theta. \tag{2.16}$$

In liquid crystals this energy is also known as the relaxation energy [13].

13

2.3.3 Orientational effects and refractive indices. As explained above, the optic axis may be reoriented by the application of an electric field. Uniaxial crystals are characterized by a particular direction known as the optic axis (or c axis). Light polarized perpendicular to the plane containing the propagation vector $\mathbf{k}$ and the optic axis is the ordinary polarization. Such light experiences a refractive index n_o called the ordinary refractive index. Light polarized in the plane containing $\mathbf{k}$ and the optic axis is the extraordinary polarization and experiences a refractive index n_e that depends on the angle ϕ between the optic axis and $\mathbf{k}$ according to the relation [3,9]

$$\frac{1}{n_e^2(\phi)} = \frac{\cos^2\phi}{n_o^2} + \frac{\sin^2\phi}{n_e^2}. \tag{2.17}$$

Here n_e is the principal value of the extraordinary refractive index. Note that $n_e(\phi)$ is equal to the principal value n_e for $\phi = 90$ degrees and is equal to n_o for $\phi = 0$. Phase matching is achieved by adjusting the angle ϕ to achieve the value of $n_e(\phi)$ for which the condition $\Delta k = 0$ is satisfied. This will be important during the SLM alignment procedures described in Chapter 3.

Because of these electrically induced reorientation of the optic axis, the phase retardation of a thin layer of properly oriented liquid crystal can be switched from zero to $2\pi(n_e - n_o)l/\lambda$ and back with the application of an electric field. This is the premise behind spatial light modulators.

2.4   SLMs as Optical Phased Arrays (OPAs)

SLMs typically utilize either the smectic phase or the nematic phase liquid crystals. The device used in this thesis is a commercially available liquid crystal OPA produced by Boulder Nonlinear Systems, Inc. (BNS). The device is a 2-dimensional 512×512 consisting of a little over 26k individually addressable electrodes that induce a birefringent phase shift in a nematic liquid crystal layer [2]. As light passes into the device, it is reflected by the array of electrode mirrors, and then exits the device. In

Incident Light

Transparent Glass
Transparent Electrode
Liquid Crystal
Reflective Electrodes
Control Circuitry
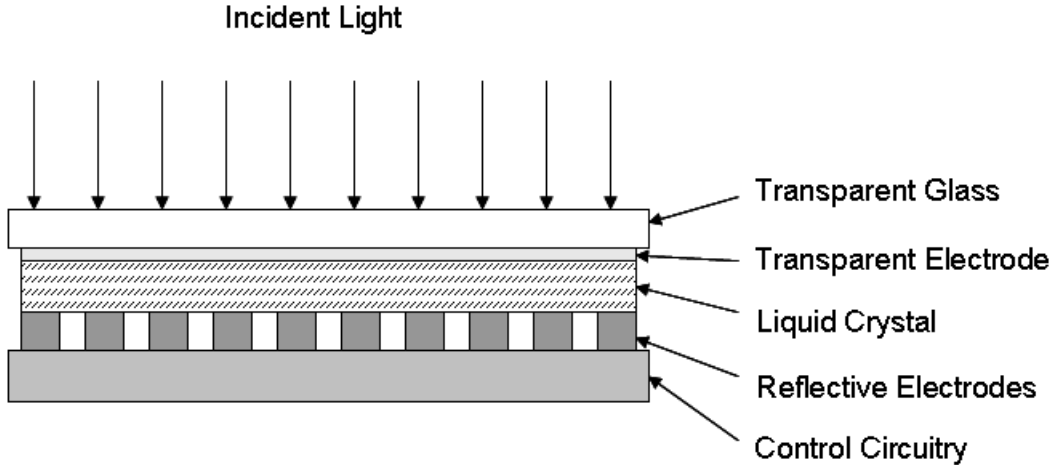
Figure 2.3:   SLM cross-section with incident light entering and exiting from the top layer. [4]

entering and exiting, the phase of light is delayed by the nematic liquid crystal layer. Figure 2.3 shows a cross-section of the SLM and the path that light would follow [4].

Deflection of a laser beam using an SLM is accomplished by applying a linear phase delay to the device. In order to reduce the response time, the thickness of the liquid crystal layer is limited to a thin layer; the device is commonly designed for a phase range of $[0, 2\pi]$. By using blazed gratings with modulo 2π phase reset the steering range is not limited by the small phase range mentioned earlier in the refractive case.

When an SLM is used as an OPA for beam steering, its operation can be analyzed by treating it as a thin phase grating. For normal incidence, the grating equation 2.7 shows that the position of the diffracted orders are determined only by the period of the grating and the wavelength of the light, while the relative energy in each diffraction order is a function of the detailed shape of the phase profile. The diffraction efficiency in the the $m = 1$ order is maximized when the slope of the phase profile is given by

$$\frac{d\phi}{dx} = \tan\theta \tag{2.18}$$

15

where $\phi(x)$ is the phase shift as a function of position [12]. In other words, when the local tilt of the wavefront caused by each grating period is equal to the angle of a diffraction order, the efficiency into that order is maximized.

Given that BNS device is 2-dimensional, the variable grating can be applied in any direction on the device, resulting in full control of beam placement (within the SLMs field of view) on the target.

2.5 Using SLMs for beam tracking

Tracking using a liquid crystal SLM can be realized by using a direction sensitive detector such as a CCD or quad cell that can generate a feedback signal. The methodology focussed upon in this thesis involves a calculation of the wavefront centroid as viewed by the imaging system. The image pixels are placed into an xy matrix and the location of the centroid is determined by the following equations

$$cntr_x = \frac{\sum_{cols} \sum_{rows} I_{ij} x_i}{\sum_{cols} \sum_{rows} I_{ij}} \tag{2.19}$$

$$cntr_y = \frac{\sum_{cols} \sum_{rows} I_{ij} y_j}{\sum_{cols} \sum_{rows} I_{ij}} \tag{2.20}$$

where $\sum_{cols}$ is the sum of each column, $\sum_{rows}$ is the sum of each row, I_{ij} is the intensity value of each pixel, x_i and y_j are the column and row designators respectively, and $cntr_{x,y}$ denote the x and y positions of the centroid in the matrix.

Once the centroid of the wavefront has been determined, its distance from the center of the image is measured in pixels. The direction from the wavefront centroid to the center of the image plane is calculated by the arc tangent function. The distance and angle values are then fed into equation 2.18 to determine the proper amount of phase shift required to move the beam to the center of the image. This phase profile is converted into a phase screen that is loaded onto the SLM, steering the beam to the center of the image plane.

2.6 Phase Holograms

Over the past few years there have been attempts to use SLMs as active imaging systems using phase holograms. Published work [11, 14] has demonstrated that phase holograms can be created and loaded onto SLMs to both shape laser beams and create images in the far field. The ideas presented in this thesis build upon the previous work to create a series of phase holograms that split a laser beam and individually steer each of its components.

Phase holograms have a high diffraction efficiency that utilize the entire face of the SLM for both beam steering and beam splitting. As will be shown in chapter 4, this has an advantage over the splitting technique utilized by the blazed grating methodology which divides the SLM into a number of subapertures, reducing the amount of area of the SLM dedicated to each beam.

The beam steering and splitting approach explored in this portion of the research is derived from a phase retrieval algorithm. The details of the algorithm can be understood using the block diagram shown in Figure 2.4. Following the methodology found in [14], development of the algorithm starts with some desired far field amplitude pattern, denoted by $P(x_f)$, where x_f is the spatial coordinate in the far field plane. The laser beam is assumed to have some constant amplitude A, which falls on the SLM. The SLM can apply a wavefront phase to the laser beam denoted by $\psi_1(x_1)$, where x_1 is the spatial coordinate in the SLM plane. The field in the far field is given by the Fourier transform of the field leaving the SLM, $Ae^{j\psi_1(x_1)}$. Physically, the Fourier transformation of this field is accomplished by a combination of fixed optics and free space propagation [6]. Mathematically, the relationship between the field leaving the SLM and the field arriving at the far field plane can be described by

$$B(x_f)e^{j\psi(x_f)} = \mathcal{F}\{Ae^{j\psi_1(x_1)}\}. \tag{2.21}$$

The goal of this phase retrieval algorithm is to determine a phase for the SLM which will yield an approximation to the desired far field amplitude $P(x_f)$. Referring to

17
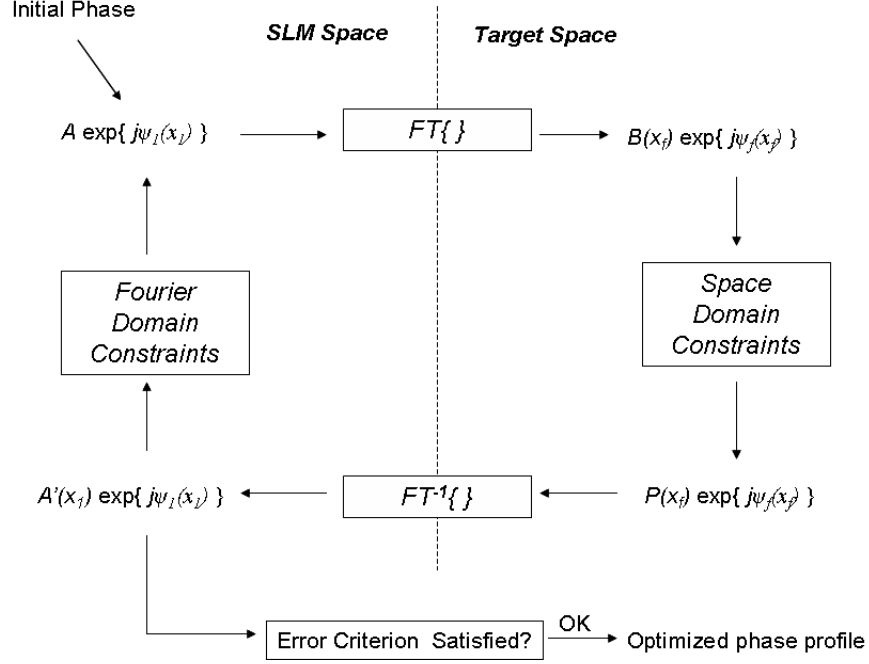
Figure 2.4:   Functional block diagram of iterative phase retreval algorithm

Figure 2.4, the process begins with a random guess at the phase $\psi(x_1)$. The field $Ae^{j\psi_1(x_1)}$ is then propagated to the far field using a fast Fourier transform to obtain the field $B(x_f)e^{j\psi_f(x_f)}$. The far field constraint is imposed by substituting the desired far field amplitude pattern $P(x_f)$ for the amplitude pattern $B(x_f)$. This modified field is then inverse Fourier transformed back to the SLM plane, yielding a field denoted by $A'(x_1)e^{j\psi_1(x_1)}$. The resulting value $\psi_1(x_1)$ is assigned to represent what is seen on the SLM for the next iteration. At this point the amplitude $A'(x_1)$ will not match the input laser field represented by A, so the SLM domain constraint is imposed by substituting A for $A'(x_1)$, completing one iteration of the process. The iterations continue until the error between the intensity distribution of the reconstructed image and the target image is within a predetermined tolerance, or a predefined number of iterations has been completed.

2.7   Computer Simulation

Computer simulation accomplished using Matlab is necessary to help understand and validate system behaviors. Since light propagation can be computationally intensive, by taking advantage of the Fourier transform nature of the Fresnel integral (Equation 2.6), the computations are greatly simplified. All experimental results are taken in the far field which allows computations for the Fourier transforms to be performed with Matlab's FFT algorithm, greatly reducing computational time.

2.8   Summary

A number of theoretical concepts have been discussed to enable a system that uses liquid crystal to steer a laser. The primary concepts covered in this chapter necessary to create the system are reviewed below.

- Diffractive beam steering devices can be implemented with an optical phased array implemented by an SLM. Phase levels and pointing angles of the array can be randomly addressed giving them great operational flexibility.

- Liquid crystals are comprised of nonlinear molecules that twist and align themselves with an applied electric field. The optical wave incident on the molecules then experiences a modified index of refraction because the average polarizability per molecule has been changed by the molecular alignment.

- A liquid crystal spatial light modulator induces a controllable phase delay on light, resulting in a steered or split beam by loading a phase map containing a predetermined amount of tilt to the system. To maximize steering efficiency, minor adverse effects of the SLM need to be compensated for prior to loading the phase screens.

- SLMs can be used in a closed loop tracking system correcting a beam back to the center of the target board (camera) by determining beam position and creating a correcting phase map.

- Phase holograms provide an alterative methodology to steering and splitting beams. Phase holograms offer high diffraction efficiency that are created though an iterative Fourier transform algorithm and loaded onto the SLM as phase maps.

- Using the same theory discussed in describing light propagation, Matlab is capable of simulating the behavior expected in the optical steering and tracking systems. This tool can aid in reducing experimentation time while simultaneously providing results to verify optical system operation.

In the following chapters, the theory discussed here will be applied to describe methodology used to design and operate the beam steering and tracking systems.

III. Methodology

3.1 Introduction

Using the theory described in Chapter II as a basis for how light interacts with a diffraction medium that can be built from liquid crystals, one can build a tool to manipulate the phase of a laser beam such that it effectively steers the beam to another direction. Given that beam steering by such a tool is possible, it is plausible to extend this idea to include the opposite effect - the stabilization of a moving beam.

The tool designed in this thesis consists of various optical element including the spatial light modulator (SLM) and an imaging camera. A computer is used to drive the SLM and imaging camera and also serves as the computational interface that controls the system in a closed loop fashion. However, before such system can be built, system limitations must be identified and a characterization study must first be performed on the SLM to measure nonlinearities between phase delay and pixel value. Following the SLM study is the procedure to operate the beam steering/tracking system and an evaluation of its performance.

3.2 System Limitations

The experiments put forth in this thesis are run on two separate systems due to a number of limiting factors. Two SLMs were used for the experiments, each with its own respective supporting equipment, details are listed in Table 3.1. Parameters in the table obviously make the $15\mu m$ SLM the device of choice for the experiments. This device is the newest of the two and is used for all the beam splitting experiments with both the phase gratings and phase holograms used as phase screens. The drawback of this device is that the BLINK software developed by Boulder Nonlinear Systems, Inc.

System Parameters	SLM 1	SLM 2
Pixel Pitch	$7\mu m$	$15\mu m$
Array Size	$512x512$	$512x512$
BLINK	Version 2.1	Version 4.2
Host PC Operating System	Windows 98	Windows XP
LabVIEW Available	Yes	No
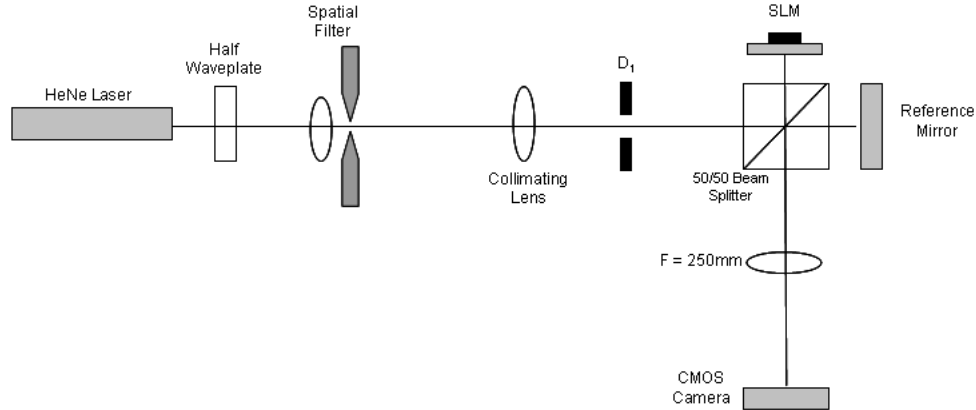
Figure 3.1:   Twyman Green Interferometer used for SLM Calibration

to run the SLM does not support continuous downloads of phase screens. BLINK requires all phase screens to be built and loaded into memory prior to operating the SLM. This limitation is not an issue when demonstrating that the SLM can split, steer, and vary the intensity of a beam. Phase screens can be pre-built and loaded into the SLM. This limitation, however, does become an issue when trying to track and stabilize an incoming beam. Beam tracking requires continuous realtime loading of phase screens.

To work around the realtime download issue with BLINK, the $7\mu m$ device was borrowed from AFRL/SNJM. Previous work with this device had included seeking ways to bypass the BLINK software and directly accessing the SLM. This was accomplished with a routine developed in LabVIEW (see Appendix B for a description of LabVIEW). Unfortunately, LabVIEW and its supporting elements for controlling the camera were only available on a Windows 98 stand-alone PC dedicated to the $7\mu m$ device. The $15\mu m$ is not backwards compatible with Windows 98 (requires Windows 2000 or better) and thus could not be transferred to the older PC. As a result, all beam tracking was done with the older device which, as results will show in Chapter 4, has severely degraded in performance with age. This degradation can also be seen in the first 30 pixel values for the calibration curve found in Figure 3.3.

3.3 Characterization of the SLM

SLMs have a couple of issues that must be taken into consideration when using them for beam steering applications. The first and foremost is that the SLM response must be measured. To insure that the SLM is creating the correct phase delay, measurements must be taken for the phase delay occurring for each gray scale value on a screen bitmap. To accomplish this, a modified Twyman-Green interferometer was set up in the lab as shown in Figure 3.1.

In the Twyman-Green interferometer, a monochromatic light source passes through a non-polarizing beam splitter such that the beam is divided into two beams of equal intensity. One of the beams illuminates the SLM, while the other illuminates a reference mirror. Both of the reflected beams are recombined at the image plane of the lens. A camera is placed in the image plane of the lens in order to record the fringes. Carefully aligning the reference mirror and the SLM such that they are nearly coplanar will create interference fringes at the image plane having an amplitude behavior described as:

$$A = \cos(\frac{Delay}{2}). \tag{3.1}$$

Relating this to intensity attenuation, the equation reduces to:

$$Delay = 2\cos^{-1}(\sqrt{I}) \tag{3.2}$$

where I is normalized between zero and one.

Calibration of the SLM is done by loading a series of solid gray bitmaps, each having a different gray value ranging from 0 to 127 and for each image recording a predetermined portion of the intensity pattern viewed by the camera. Although gray values are available from 0 to 255, the values 128 through 255 duplicate values 0 through 127 in reverse order as the liquid crystal molecules rotate through 360°. Stepping through the gray scale values will cause the fringe pattern seen in Figure 3.2 to shift across the image plane. Capturing images for each gray scale and using

23
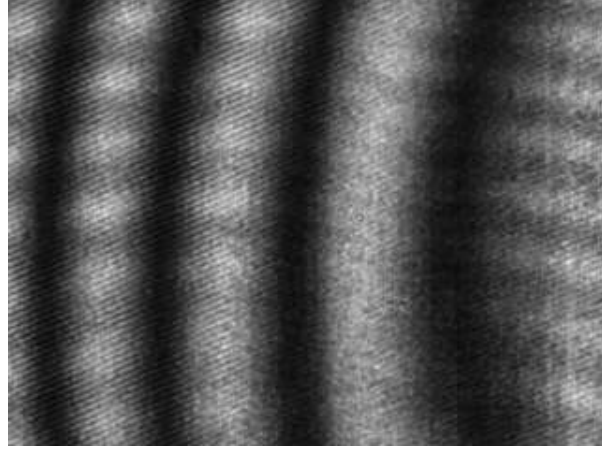
Figure 3.2:   Calibration fringes created from Twyman Green Interferometer

equation 3.2, the SLM phase delay response to SLM values can be calculated as seen in Figure 3.3.

The goal is to be able to generate a linear phase delay of 0 to 2π (known as one wave of delay) to maximize the efficiency of the blazed grating. Looking at Figure 3.3, the most linear region is from 20 to 80. However a full wave of tilt is required to maximize efficiency so the region from 5 to 70 is chosen to represent the phase screens on the $7\mu m$ pitch SLM. In this region, 256 gray values are mapped to 65 SLM values to represent phase delay in a linear fashion. A custom lookup table is created to load the proper SLM value representing the desired phase delay. The lookup table properly translates phase screens using gray values of 0 to 255 to phase delay values ranging from 0 to 2π, thus minimizing the nonlinearities in the orientational effects of the liquid crystal molecules when exposed to an electric field as discussed earlier in Chapter 2. Further examination of the data on the lower end of the curve in Figure 3.3 reveals a problem with the SLM that will be further discussed in Chapter 4. The same calibration exercise was performed for the $15\mu m$ pitch SLM device resulting in the plot found in Figure 3.4. In this case the region from 0 to 70 was chosen to represent the phase screens.

A second issue that must be taken into consideration in characterizing SLMs is the effects induced by the geometry of the device. Each pixel on the SLM array
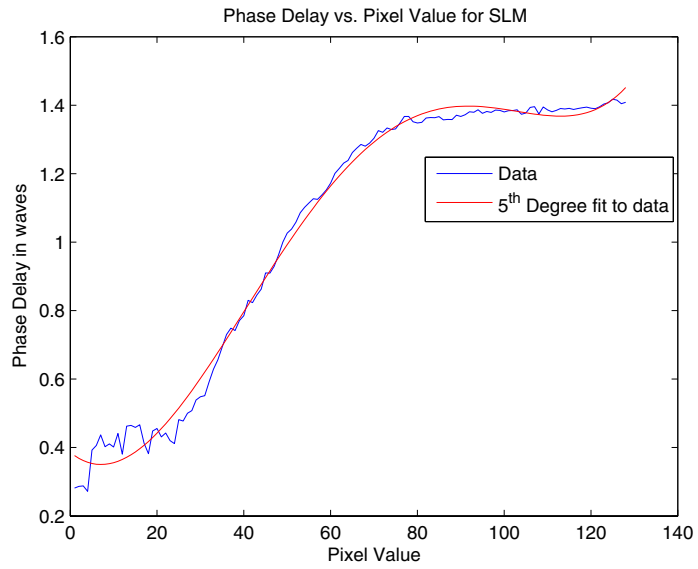
Figure 3.3:   Phase Delay vs Gray Values for $7\mu m$ pitch SLM
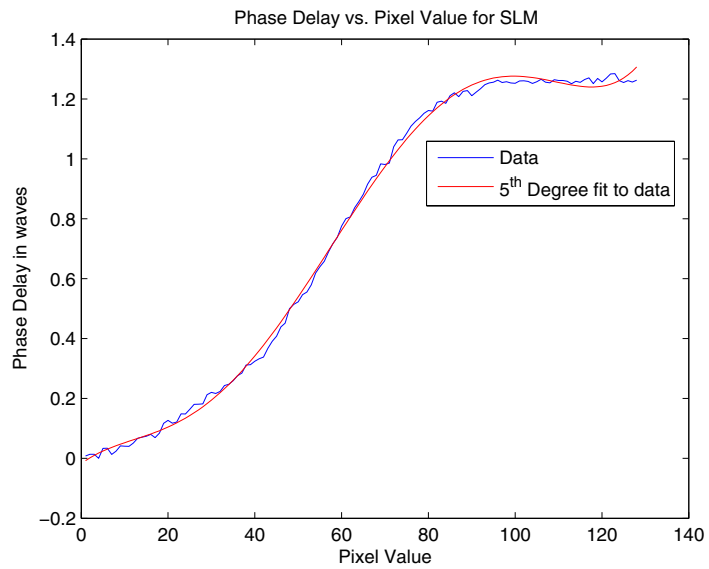
Figure 3.4:   Phase Delay vs Gray Values for $15\mu m$ pitch SLM

has a fill factor of only 83%. In between each pixel, there exists a non-reflective area. The result is the effect of a diffraction grating that reflects light into multiple orders, each containing a portion of the intensity of the incident beam. The reflected beam is focussed to a point in the far field, thus the small amount of uncontrollable light was found to have minimal effect on the overall operation of the device. However, at very small steering angles with short propagation distances, it is not possible to separate the steered light and unsteered light. Another concern is the SLM housing causes light to be scattered from the edges when the SLM is completely illuminated or in a beam tracking scenario when the beam starts to move off the device. These effects can be reduced by masking the edges of the SLM.

3.4   Optic Bench Layout

3.4.1   Beam Steering, Splitting and Intensity Variation.   The layout of the experimental system used is shown in Figure 3.5 and consists of simple optic elements, the SLM, and the camera. A Helium-Neon laser beam operating at 15mW and 632.8nm is directed into a pinhole to approximate a point source. A half wave plate is placed in the system to rotate the linear polarization from the He-Ne to line up with the polarization of the liquid crystal in the SLM. A lens collimates the light and an aperture is used to control the diameter of the collimated light so that it completely fills the area of the SLM. The SLM is loaded with a phase screen that will direct the light to the position of interest on the image plane either by simply steering the beam, or by splitting the beam into multiple beams each being steered to a predetermined location.

A single personal computer (PC) is used in the experiments for multiple purposes. First, it interfaces to a CMOS ultra camera built by Computer Modules, Inc. that captures images for evaluation of system performance. The camera has a 1280 by 1024 array with a pixel pitch of 7.5 μm. Second, the computer interfaces to the SLM via a PCI board that connects to an external Digital-to-Analog Converter (DAC) board which in turn interfaces with the SLM. A digital image of the phase screen is
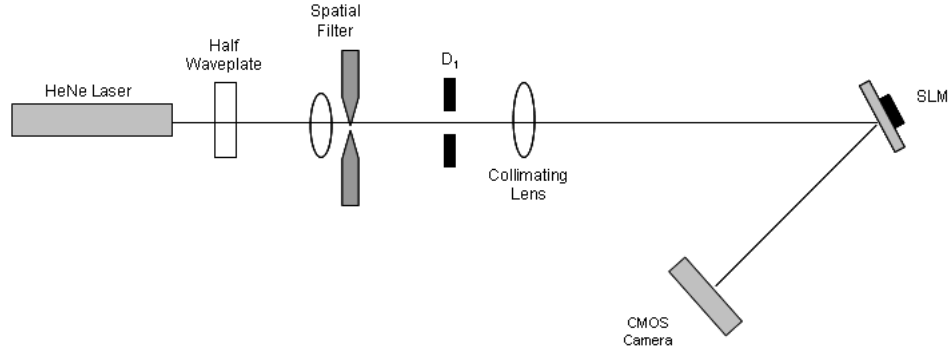
Figure 3.5:   Experimental setup for beam steering, splitting and intensity variation
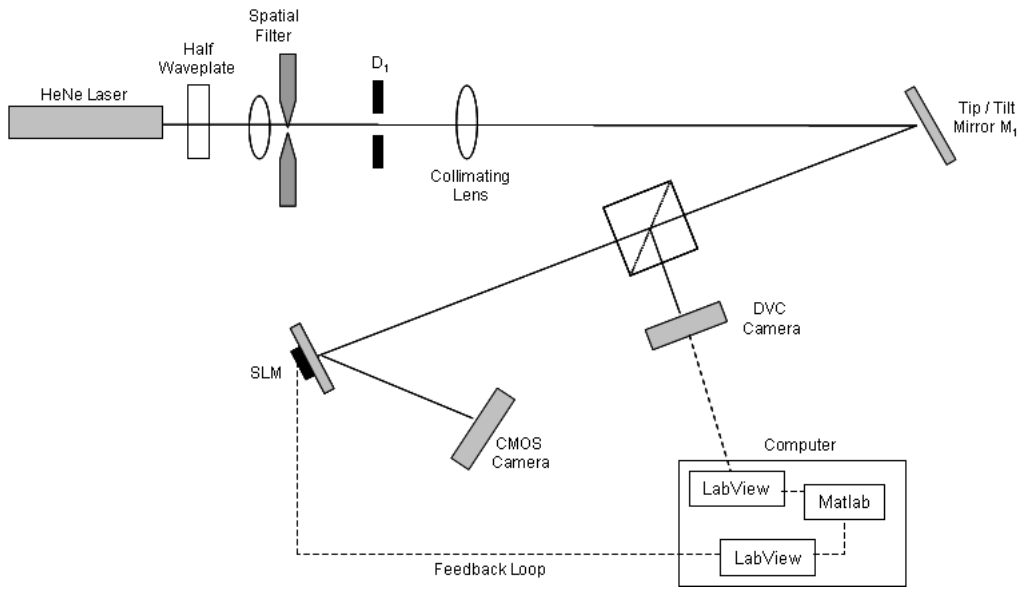
Figure 3.6:   Experimental setup for beam tracking

sent from the PCI board to the DAC where it is converted to an analog data signal that is amplified to appropriate levels for the SLM.

3.4.2 Beam Tracking.   Beam tracking and stabilization experiments require additional optics in the system as seen in Figure 3.6. A tip/tilt mirror is added to control movement of the beam allowing simulation of random movement. The beam is split with a 50/50 beam cube, passing one leg to the SLM and directing the other to a second camera (model 1310AM built by the DVC Company) that captures an image of the beam position for processing by the computer. The beam is reflected off

27

the SLM, through a lens into a second camera, recording the corrected far field beam position.

The PC uses a LabVIEW routine to control image acquisition of the uncorrected beam, generation of a correcting phase screen, and loading the phase map onto the SLM via the DAC board. LabVIEW bypasses the BLINK software, communicating directly with the DAC board. The BLINK software is limited in that it must load all phase screens to memory prior to operating the SLM. Beam tracking requires continuous download capabilities to the SLM, thus necessitating bypass of the BLINK software. Once the phase screen is passed to the SLM, it then corrects the direction of the laser to the desired point on the image plane - the center. A block diagram of the LabVIEW routine is shown in Figure 3.7 , details are examined in Appendix B.

3.5 System Operating Procedures

3.5.1 Beam Steering and Splitting.   To operate the system, a series of phase screens are produced using the Matlab routines listed and explained in Appendix A. The phase screens in the form of bitmaps are loaded onto the SLM with the BLINK software. Light passes through the system to the camera and a far field image is recorded for each phase screen.

Two techniques are examined for beam steering and beam splitting. The first uses a phase grating described in Chapter 2. To simply steer a single beam, a modulo 2π representation of tilt is applied to the system that utilizes the entire face of the SLM. An example of this type of phase screen is seen in Figure 3.8.

To split a beam into multiple parts, the $512x512$ pixels that make up the face of the SLM (and thus the phase screen) are equally divided into the number of desired beams. Each subdivision has a unique phase screen applied to steer its representative beam in the desired direction. All of the subdivision phase screens are combined into a single $512x512$ phase screen as shown in Figure 3.9 and loaded onto the SLM.
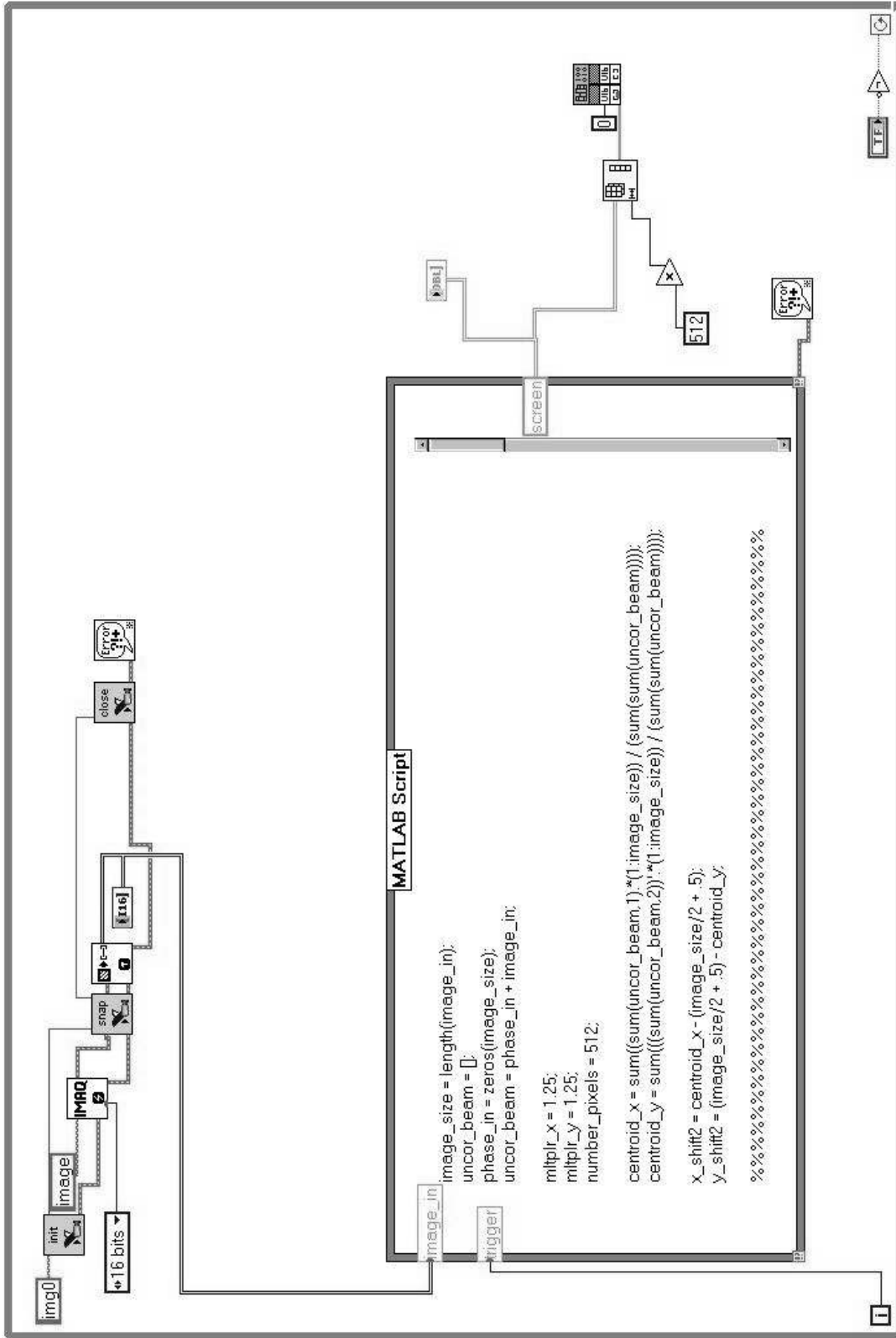
Figure 3.7:   Block diagram of the LabVIEW code
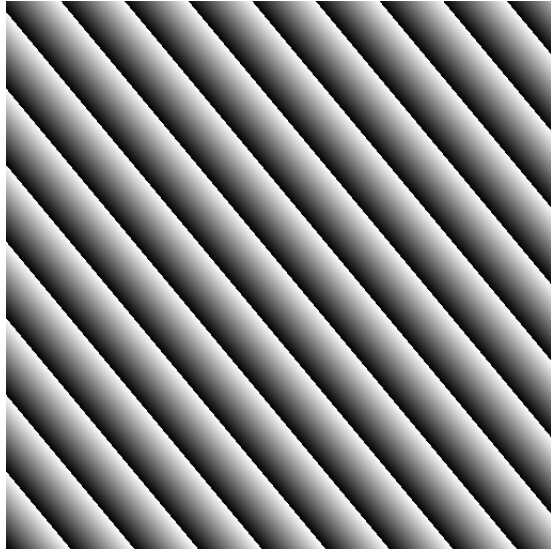
Figure 3.8:   Example phase map used to steer a beam
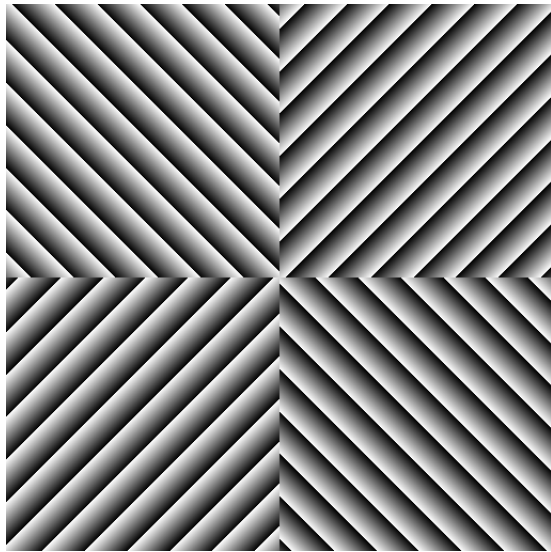
Figure 3.9:   Example phase map used to split a beam into 4 and individually steer each beam.
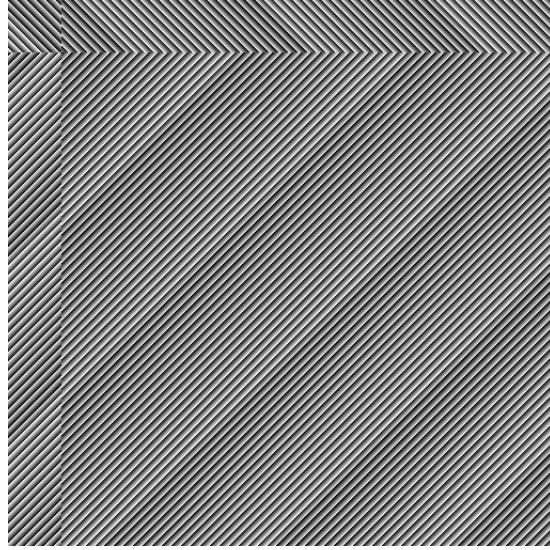
Figure 3.10:   Example phase map used to split a beam into 4 and varying the intensity of each steered beam.

At times it will be desirable to vary the amount of intensity for each of the subdivided beams. An example scenario is a forward looking aircraft relaying information simultaneously to other standoff aircraft and/or ground units. The receiving aircraft and ground units will likely be at varying distances from the transmitting aircraft, resulting in requirements for varying amounts of intensity for the transmitting laser beams. A simple way to vary the intensity is to dedicate more or less real estate of the SLM to each subdivided beam. Using a bigger portion of the SLM face allows more energy to be directed into a given direction. Figure 3.10 shows an example phase screen that divides the beam into four independent beams with varying levels of intensity.

Phase holography is the second method of beam steering and beam splitting. Similar to the phase grating technique, a Matlab routine is run to develop a phase hologram of the desired far field image. The far field image could simply be a single spot at a certain location, or a more complex image consisting of multiple spots located at predetermined locations. Utilizing the iterative phase retrieval algorithm described in Chapter 2, Matlab generates a phase hologram that is loaded onto the SLM.

31

3.5.2 Beam Tracking and Stabilization. The beam tracking and stabilization system is operated in a similar manner as the beam steering/splitting system. Rather than using Matlab to pre-generate phase screens for loading to the SLM prior to conducting the experiment as was done above, for beam tracking, the phase screens are generated in near real time based off information captured by the DVC camera. Refering to Figure 3.6, the camera sends a digital image to the PC which uses a LabVIEW program with an imbedded Matlab routine to determine how far off center the beam is by calculating the beam centroid as explained in Chapter 2. The Matlab routine then generates a correction phase screen which is passed back to LabVIEW to be loaded to the SLM and thus correcting the beam. An image of the corrected beam is captured by the CMOS camera and compared to the original image captured by the DVC camera for error analysis. The tip/tilt mirror in the system is manually adjusted to a random position to simulate the beam randomly moving across the face of the SLM. The beam correction is done in a closed loop fashion continuously running until user intervention.

As mentioned above, the goal of this experiment is to develop a system that tracks and stabilizes a beam in real time, creating phase screens on the order of 100 frames/sec. System limitations, mainly caused by the need to pass information back and forth between LabVIEW and Matlab, greatly degraded performance to a rate of 1 frame created every 16 seconds. This will be discussed further in the next 2 chapters and Appendix B.

3.5.3 Computer Simulation. Simulating both lab bench systems on a personal computer is accomplished using Matlab. Details of the Matlab routines can be found in Appendix A. Matlab scripts are generated for each scenario tested in the lab before the experiment is run. Although modeling of the lab bench setup is followed closely, the scripts are simplified to the extent that they don't consider limiting factors of the SLM such as the fill factor described above. They are however, sufficient

to gain insight to the expected behavior of the system and are used for comparison during system analysis in Chapter 4.

3.6 Summary

Two issues with the SLM exist that cause undesired effects in both the beam steering/splitting system and the beam tracking system. The first is the non-linear phase response of the SLM to gray values. This issue is resolved by measuring the response and creating a lookup table for the SLM to map gray values to phase delay in a linear fashion. The second is that the BLINK software doesn't allow for continuous downloads of phase screens during beam tracking and stabilization experiments. This issue is eliminated by using LabVIEW to directly communicate with the SLM, bypassing the BLINK software. Beam steering, splitting, and tracking can then be accomplished by loading phase screens to the SLM and recording the far field image with a camera.

IV. Results and Analysis

4.1 Introduction

Beam steering is achieved using the first optical system described in Chapter III. This system is also used to split a beam and vary the intensities of each sub-beam. The ability of this system to accurately control the desired outcome is dependant on the phase screens and the spatial light modulators ability to represent those screens. Validating the system will, therefore, be accomplished by investigating the performance of the phase screens themselves in simulation and evaluating the product of the SLM over various scenarios. Three scenarios are chosen for beam splitting and steering experiments and a fourth experiment is dedicated to beam tracking. Each scenario represents a possible real-world possibility where laser energy is used for communication purposes. For each experiment a scenario description will be given followed by an evaluation of computer simulation performance. Lab system data will then be presented and compared simulation.

4.2 Test Cases

It was mentioned in Chapter 1 that SLMs offer new capabilities to beam steering that are available in current gimballed mirror systems offering a "next generation" capability to optical communications systems. Based on interests identified by the Air Force Research Labs, four potential capabilities are examined to demonstrate the SLMs performance and flexibility in optical communications systems.

- Split a beam and individually steer each resulting beam

- Vary the intensity of the split beams

- Explore phase holography as an alternative methodology for beam splitting and beam steering

- Track and stabilize an incoming beam

4.3 Experimental Limitations

The scope of this thesis focuses only on demonstrating that an SLM can potentially be used for each scenario. Given this, the experiments are developed to run at a top level and only take into consideration basic tilt in the xy plane for steering and beam splitting. Focus and higher order abberations, can be corrected with SLMs by adding additional terms to the phase map but are neglected in this work.

4.4 Splitting and Steering Laser Beams with SLMs

The ability to split a beam into multiple parts, each capable of being individually steered to a target via nonmechanical means is the impetus behind this research. There is a desire in multiple Air Force communities: airborne platforms, space platforms, and their supporting ground elements, to have optical communication capabilities. Within this desire is the need to pass realtime information to multiple users simultaneously, of which two examples are: (1) a Predator aircraft relaying realtime imagery to fighter pilots and theater commanders for targets of opportunity; (2) Optical satellite communication links to multiple tactical users in theater, providing them secure communication capabilities due to reduced EM footprint.

SLMs can steer a beam with fairly high efficiency (according to [2] this value can be 83%), but due to physical constraints are limited to small steering angles on the order of a few $mrad$. Simply by applying a phase screen with xy tilt represented as a modulo 2π phase grating will steer a beam to any given point within the SLMs field of view. Referring back to the grating equation discussed in Chapter 2 and reprinted here for convenience,

$$d \sin \theta = m\lambda \tag{4.1}$$

where m is always taken to be 1 since we are strictly steering to the first order, the SLM has the ability to create gratings with a d value as small as 8 times it's pixel pitch. Given this, the $15\mu m$ pitch device has a maximum steering angle of $5.2mrad$ when illuminated with a $632.8nm$ HeNe laser. These angles seem insignificant at first

glance, but considering the two scenarios described above, they become noteworthy at long propagation distances. At shorter distances, multiple SLMs can be used to increase transmission angles.

Beam splitting with an SLM is traditionally done by carving the phase screen into a number (equal to the number of desired sub-beams) of equal parts. A simple example of this is shown in Figure 4.1 where the phase screen is divided into four quadrants, creating four beams in the far field. Notice from this figure that each sub-phase screen has its respective blaze grating rotated in a different direction showing that the beams can be independently steered. The upper right quadrant has a smaller d value for its grating equation causing its respective far field image to be steered further away than the others as shown in Figure 4.2. This figure shows there is a noticeable amount of light left in the zero order caused by the fill factor and diffraction effects mentioned in the previous chapters. Careful examination of this figure also shows that these effects also steer a minute amount of light to some second orders. Paying attention to just the first order beams, each of the four beams can be independently steered to any position within the $5.2mrad$ view of the SLM. Figure 4.3 is created with a Matlab simulation applying the same methodology. Notice that this figure is a mirror image of Figure 4.2. The SLM is a reflective device whereas Matlab assumes a transmissive device, thus creating the mirror image when comparing simulation to experimentation. Matlab also assumes a perfect SLM, thus eliminating the zero and second order spots. Focussing our attention on the desired splitting and movement of the beam, it can still be seen that experimentation follows simulation quite nicely.
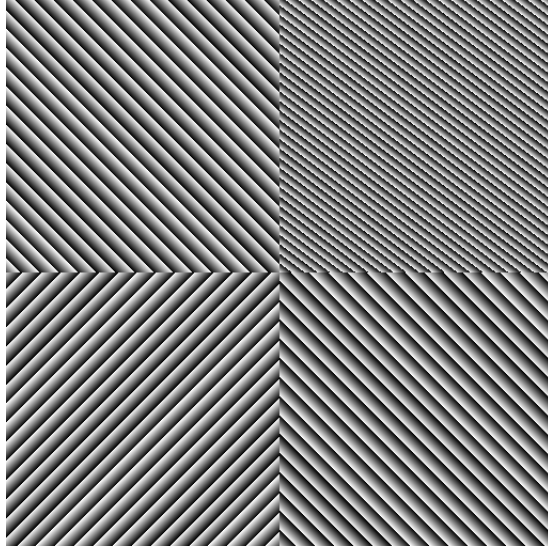
Figure 4.1:   Beam splitting and steering: Phase map using the phase grating technique to split one incoming beam into four and independently steering them in different directions.

A number of different values were used for the blaze width in combination with a number of different levels of beam splitting - all experiencing positive results. The maximum number of beams created using the phase grating technique was 16. This number was chosen solely on the fact that a $512x512$ array can easily be divided equally by a 2^n value and 16 was the biggest number that could be readily realized in a reasonable amount of time. Figure 4.4 shows the phase screen used to split the incident beam into 16 parts as seen in the far field image shown in Figure 4.5.
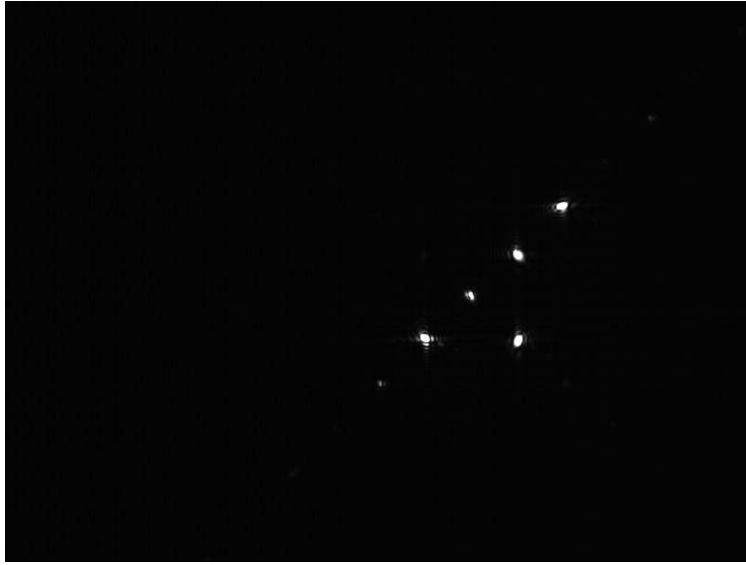
Figure 4.2:   Beam splitting and steering: Far field pattern seen by the camera. The beam is split into 4 main beams, each steered in a different direction. The residual light located at the center of the pattern is a result of diffraction effects and fill factor mentioned earlier.

Figure 4.3:   Beam splitting and steering: Similar image to Figure 4.2 created with Matlab simulation. Mirror image is cause by Matlab assuming a transmissive device.
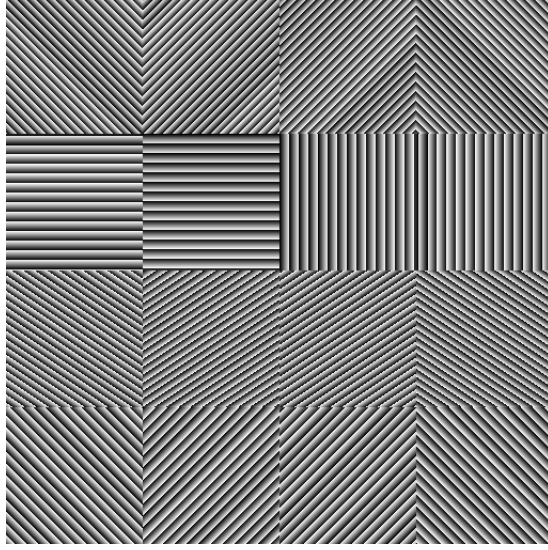
Figure 4.4:   Beam splitting and steering: Phase screen used to split a beam split into 16 parts.

It is important to keep in mind that less real estate is allocated on the SLM to each beam as the number of output beams increase. This leads to addition diffraction issues induce by the "walls" of each sub-phase screen. Previous research conducted by [13] has shown that there are **E**-field bleed-over effects between each pixel of the SLM that become pronounced in areas experiencing large resets (i.e. a modulo 2π reset). The "walls" add additional large resets to the system causing further degradation in beam quality and efficiency. A high level issue is that as the available real estate for each beam decreases the maximum value of d in the grating equation reduces affecting the ability of fine course steering.

4.5   *Shifting Power Between Split Beams*

The idea of shifting power levels between the split beams arises from the idea that there will be instances in a "one-to-many" communications link when the receiving entities are at different distances from the transmitting source, thus having different transmit power requirements.

A simple way to meet this requirement is to dedicate more real estate of the SLM to the beam requiring higher intensity. All that is happening is that more of
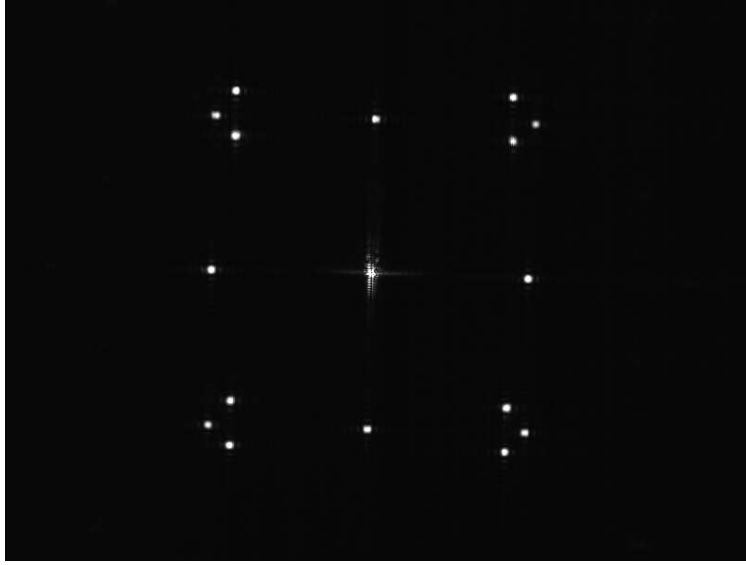
Figure 4.5:   Beam splitting and steering: Image of a single beam split into 16 beams that can be controlled individually.
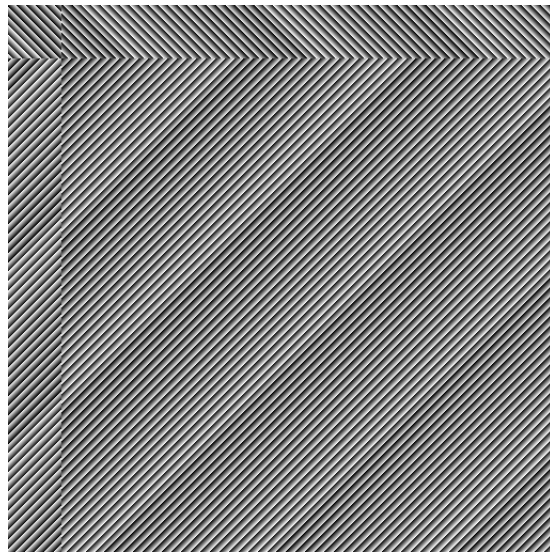
Figure 4.6:   Intensity variation: Phase screen is divided into four segments of various sizes. The segment size directly relates to the intensity level for each beam in the far field.

the incident beam is being directed in a certain direction. Figures 4.6 - 4.9 show the incident beam again being split into four sub-beams but with varying intensities. In Figure 4.7 one of the beams is almost turned off (upper right), two beams are at a medium intensity (upper and lower left), and the fourth beam (lower right) is at a high intensity saturating the camera. Applying the phase screen seen in Figure 4.8 to the same system changes the intensities to those seen in Figure 4.9.

The Matlab simulation for this experiment gave very similar results. Figure 4.10 shows the result the phase screen seen in Figure 4.8 applied under ideal conditions (again, the Matlab image is a mirror image of experimentation). As explained in Chapters 2 and 3, the ideal conditions in the simulation take away the limiting effects of the SLM such as fill factor, resulting in no energy left behind in the zeroth order. Other than this, simulation agrees quite well with experimentation.

A note worth mentioning is that this figure also shows the beams appearing to be out of focus. However, this is not a focusing issue but rather a diffraction effect due to the fact that the individual phase screens for each beam are no longer symmetrical. Although this doesn't affect the outcome of this research, this diffraction effect could be an issue in other applications. A potential way to mitigate this effect is to use the phase hologram method instead of the phase grating. However, this is beyond the scope of this thesis.

4.6 Phase Holograms

The idea of phase holograms was introduced when it was discovered that there were added diffraction issues when the beam is split a number of times. The idea is to gain greater efficiency by eliminating the added "walls" mentioned above and reducing the number of large phase resets as much as possible.

An iterative Fourier transform phase retrieval algorithm based upon work done by [11, 14] was implemented to create the phase holograms. A number of different experiments were run to find the optimum number of iterations to run in creating the

Figure 4.7:   Intensity variation: The original beam is split into four beams of varying intensity. The upper and lower left beams are at an equal medium power, the upper right beam is turned off, and the lower right beam contains the most intensity - saturating the camera in this case. SLM limitations cause some residual light to remain in the zeroth order and also to be steered to the second order.
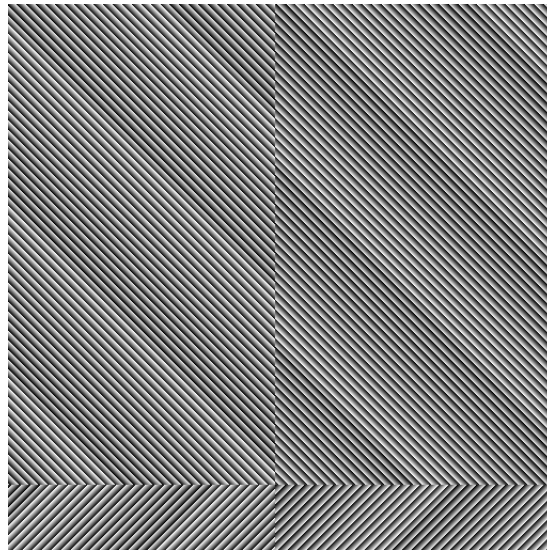
Figure 4.8:   Intensity variation: Changing the amount of SLM real estate dedicated to the beam changes the intensity of the beam in the far field.
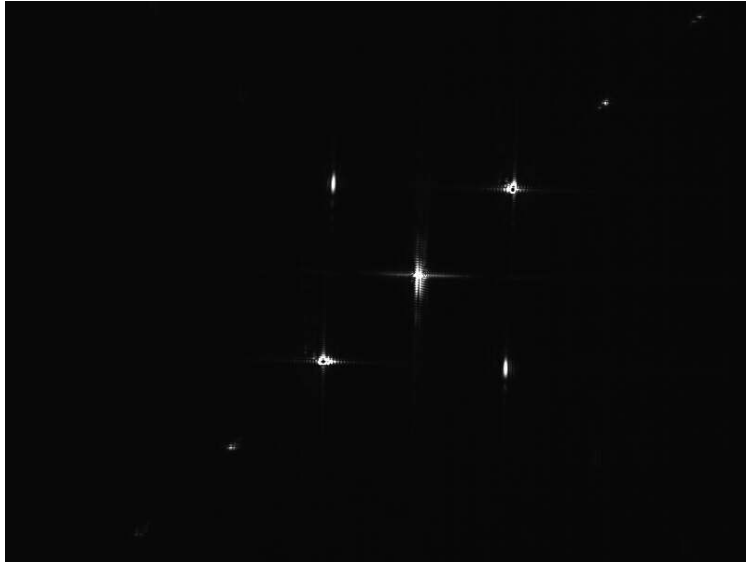
Figure 4.9: Intensity variation: Resulting variation in intensity. The upper right and lower left beams are both at a high intensity, whereas the upper left and lower right beams are at a low intensity.

Figure 4.10: Intensity variation: Matlab simulation of the scenario seen in Figure 4.9, splitting a single beam into 4 and controlling the power to each sub-beam. Two beams are at reduced power and two are at high power.

holograms. The number of iterations run for each phase hologram ranged from 2-250 with varying results. Figure 4.11 is created from 4 iterations of the phase retrieval algorithm. The resulting image created from loading this hologram as a phase screen onto an SLM is seen in Figure 4.12, splitting a beam into four sub-beams. It can be seen that the efficiency is very low as expected from only four iterations of the algorithm. Figure 4.13 is the result of letting the algorithm run for 200 iterations resulting in the far field image seen in Figure 4.14. Notice that there is little increase in the efficiency, which is not the expected result when comparing to the Matlab simulation seen in Figure 4.15.

Notice that the diameter of the split beams in these figures are much wider than the split beams shown earlier with the phase grating technique. This is due to limitations experienced in Matlab with the phase retrieval algorithm. The far field beam widths had to be at least 30 pixels in diameter for Matlab to run the algorithm which points to an error in implementing the algorithm. This is discussed further at the end of this chapter and again in Chapter 5.
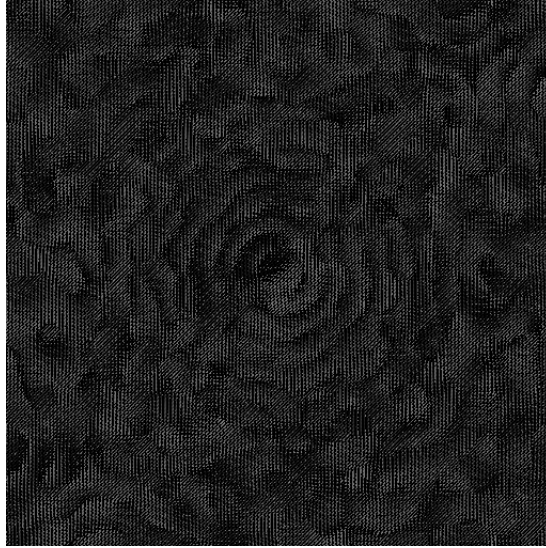
Figure 4.11: Hologram beam splitting: Phase hologram created with 4 iterations of the phase retrieval algorithm to split a beam into 4 parts.

The efficiency levels for each phase hologram were examined for both simulation and experimentation. Although simulation performed slightly lower than expected, experimentation results showed that an increase in the number of iterations did little to increase steering efficiency. Figure 4.16 shows the normalized intensity versus the number of iterations used to create the hologram for both cases. Part of this discrepancy can be attributed to the SLM, however further examination points to a potential problem in the phase retrieval process. Considering the fact that the Matlab simulations assumed a perfect SLM, it is expected that the intensity value of the spot would approach 1 with an increasing number of iterations. The suspected problem is the proper assignment of the space domain constraints. This is discussed further in Chapter 5.

The maximum number of beams that can be created with the phase hologram is unknown. The maximum number created for this research is eighteen as seen in Figures 4.17 and 4.18 - the hologram was created with 200 iterations. It can be seen that as the number of beams created increases, the intensity of each reduces as expected.
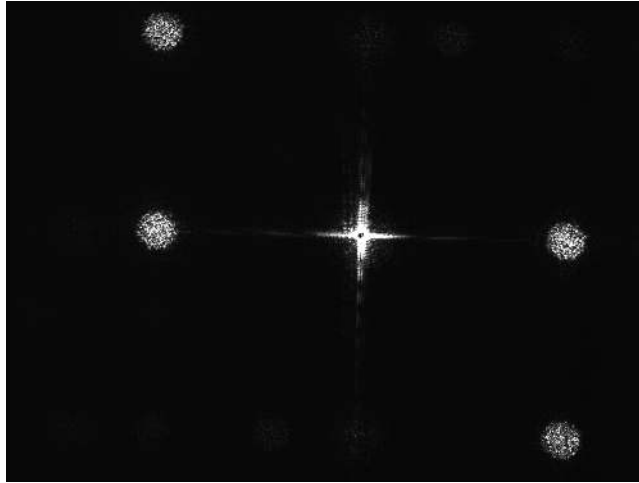
Figure 4.12: Hologram beam splitting: Far field image of the pattern created by the phase hologram created from 4 iterations of the algorithm, splitting a beam into 4 parts.
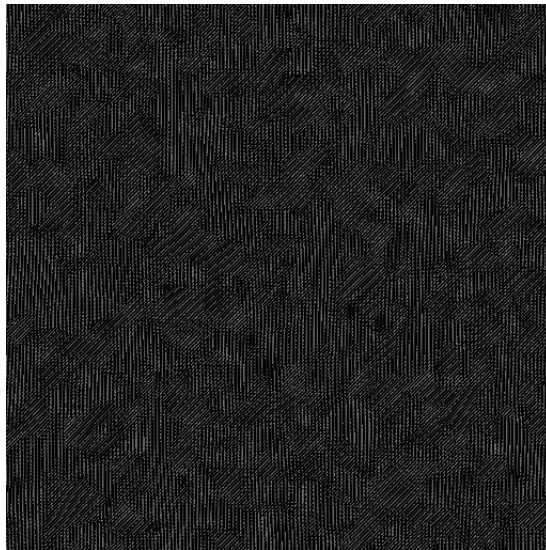
Figure 4.13: Hologram beam splitting: Phase hologram created with 200 iterations of the phase retrieval algorithm to split a beam into 4 parts.
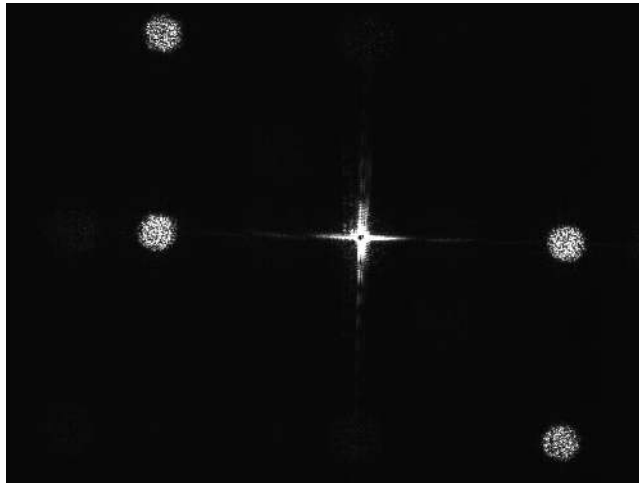
Figure 4.14: Hologram beam splitting: Far field image of the pattern created by the phase hologram created from 200 iterations of the algorithm, splitting a beam into 4 parts.
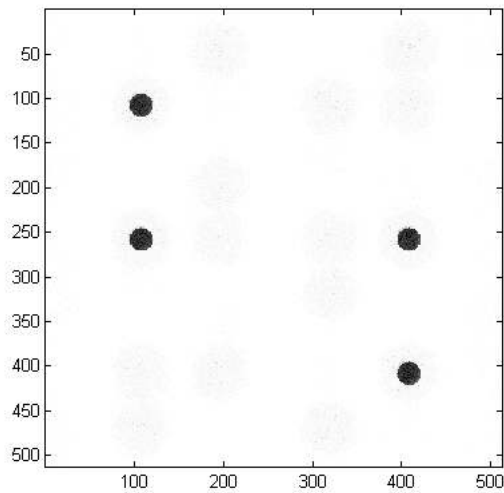
Figure 4.15: Hologram beam splitting: Matlab simulation of the far field image pattern created by the phase hologram created from 200 iterations of the algorithm, splitting a beam into 4 parts.
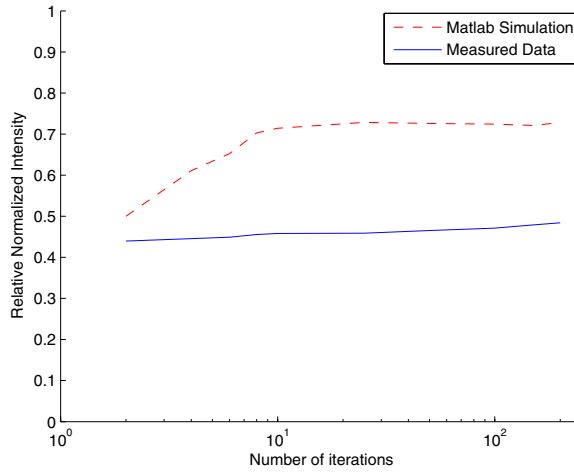
Figure 4.16:   Hologram beam splitting: Normalized average intensity for each split beam increases with additional iterations of the phase retrieval algorithm demonstrating increasing efficiency with the system. Experimental data does not increase at the pace of simulation.
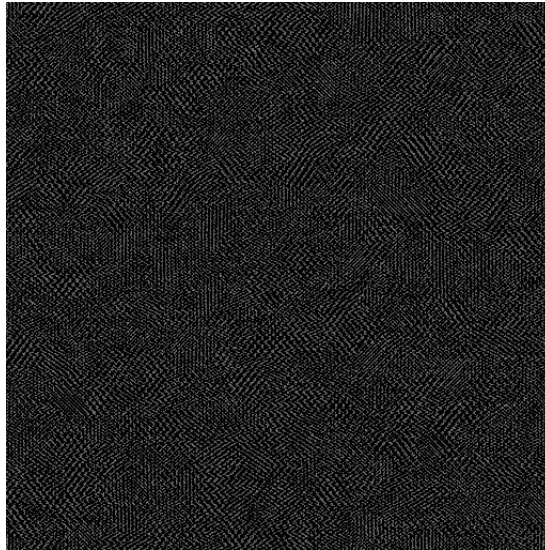
Figure 4.17:   Hologram beam splitting: Phase hologram created with 200 iterations of the phase retrieval algorithm to split a beam into 18 parts.
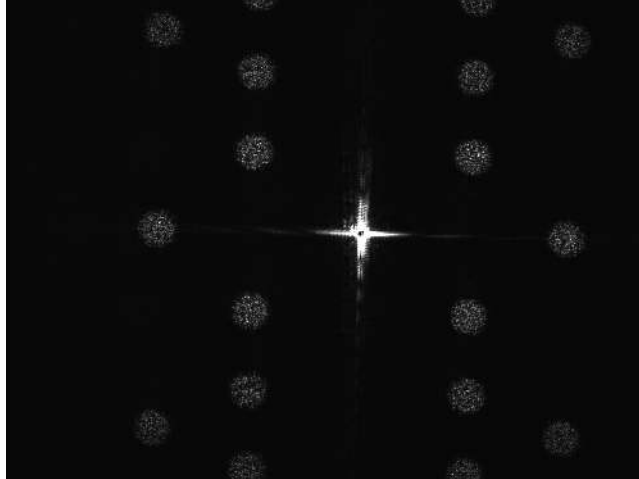
Figure 4.18:   Hologram beam splitting: Far field image of the pattern created by the phase hologram splitting a beam into 18 parts.

4.7   Closed Loop Beam Tracking and Stabilization

Closed loop beam tracking and stabilization is a capability desirable in both the optical communications realm and target tracking. The closed loop tracking system described in Chapter 3 was developed and implemented in the lab to show that using an SLM as the correcting optics is a viable option in such a system.

As mentioned previously, equipment limitations forced the use of an older device that had some age-induced problems. The lower corner of the nematic liquid crystal had a breach causing a small bubble in the device. This may have been the major contributing factor in the limited performance of the device. As a result, it was difficult to properly characterize the device and come up with a linear phase relationship between gray scale value and phase retardation. As a result, the majority of the incident beam remains in the zeroth order and the rest of the energy is distributed to multiple orders. Given these limitations, results are still sufficient to show that the system does attempt to correct the beam to the desired location.

The closed loop system takes $16sec$ to respond to a change in the incident beam angle of arrival. This can be overcome with more efficient controls and coding. The important thing to consider in this work is whether or not the SLM can return the

49
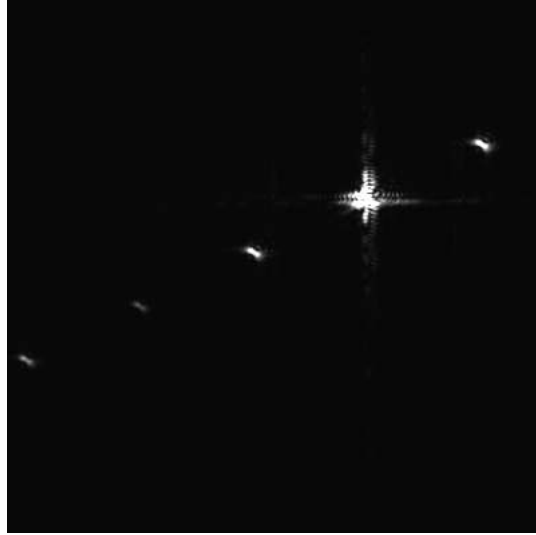
Figure 4.19: Beam Tracking: An SLM correcting a beam back to the center of the image plane. The bright spot is the uncorrected beam

beam to the desired location and thus no effort was expended on increasing the speed of the system. Figures 4.19 and 4.20 show two examples of the beam being corrected back to the center of the image plane. As mentioned above and in Chapter 3, these figures show that the damage to the SLM causes the light to steer to many orders - this is the result of inability of achieving a full wave of tilt during the calibration process. In both figures, the bright spot is the zeroth order. The spot closest to the center of each image is the corrected beam, demonstrating the success of the closed loop system. Examination of the corrected beam position error shown in Figure 4.21 shows that the given system has some issues with overshoot in the x-direction and undershoot in the y-direction. This can be compensated for in the Matlab script by adjusting the two offset variables $mltplr\_x$ and $mltplr\_y$ used to calibrate the system. This is explained further in Appendix B.

4.8 Phase Hologram vs Phase Grating

During this research is it was discovered that there are instances where the phase grating technique may not be the optimal way to split and steer laser beams. One such instance that comes to mind is when the incident beam starts to move off the face

50
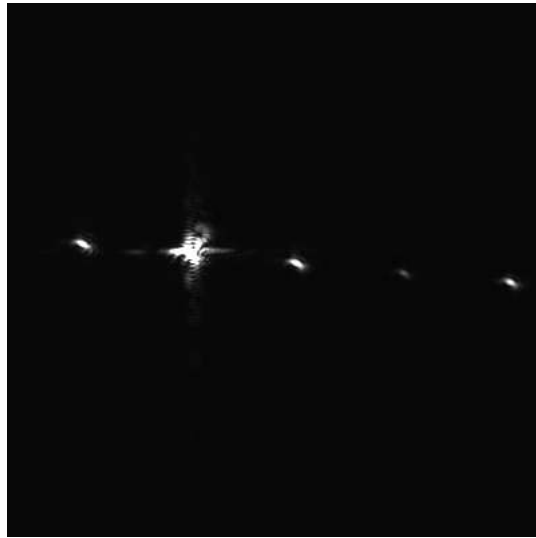
Figure 4.20:   Beam Tracking: An SLM correcting a beam back to the center of the image plane. The bright spot is the uncorrected beam
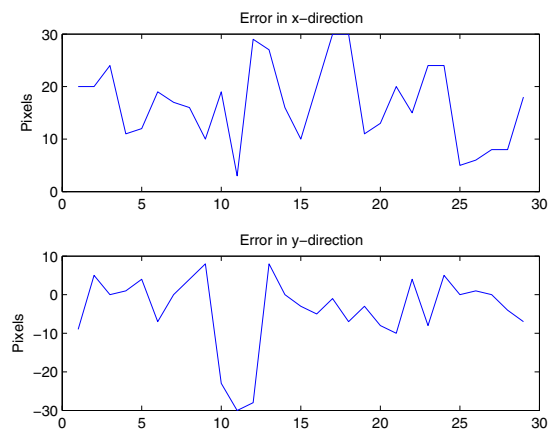
Figure 4.21:   Beam Tracking: Beam tracking error in the xy plane
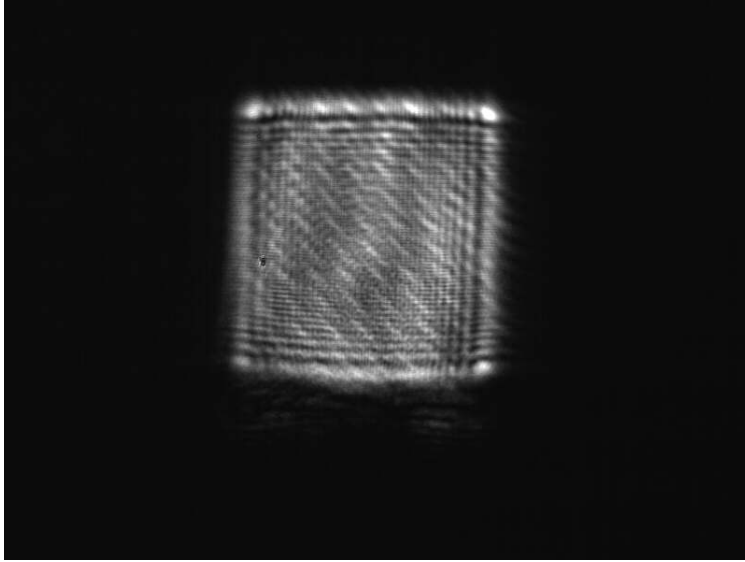
Figure 4.22:   Phase Hologram vs Phase Grating: Laser light filling the face of the SLM

of the SLM. Figure 4.22 shows the SLM completely illuminated by the SLM which is the preferred methodology for efficiency. As the beam moves off the face of the SLM as shown in Figure 4.23 problems start to occur as seen in Figure 4.24. This figure shows that a number of beams are missing from the original 16 spot pattern shown in Figure 4.5 as a result of the incident beam "missing" some of the sub-gratings. Taking the same scenario into consideration with the 18 spot image created by the phase hologram, Figure 4.25 shows that the only affect is on intensity (the contrast and brightness levels of this image have been increased by 15% in this document for printing purposes). This is significant when considering that a lost link can result in mission failure. A reduction in intensity may only result in a mission degradation.

Further showing the disadvantage of the blazed grating technique in this scenario, a phase screen was created to split the beam into 9 (greatly increasing the SLM real estate dedicated to each beam) and there was still an effect on performance as shown in Figure 4.26. Thus it goes to show that even in its current form, this is an instance where the phase hologram outshines the blazed phase grating technique.
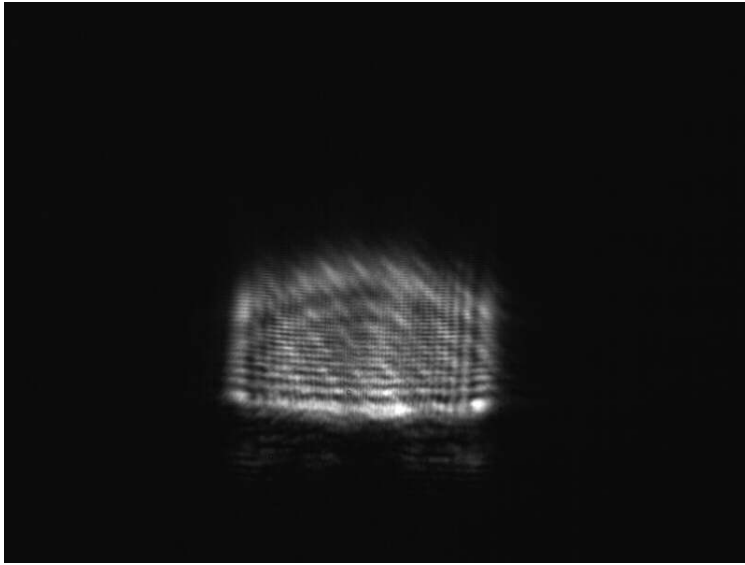
Figure 4.23:   Phase Hologram vs Phase Grating: Laser light filling only 1/2 the face
of the SLM
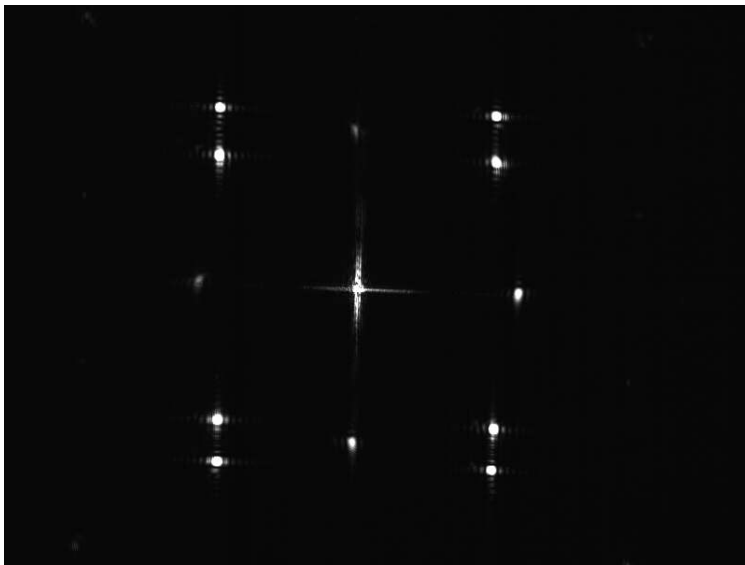
Figure 4.24:   Phase Hologram vs Phase Grating: Image found in Figure 4.5 altered
due to the beam being off center on the SLM. Some spots are lost, some lose energy
and some are affected by added diffraction.
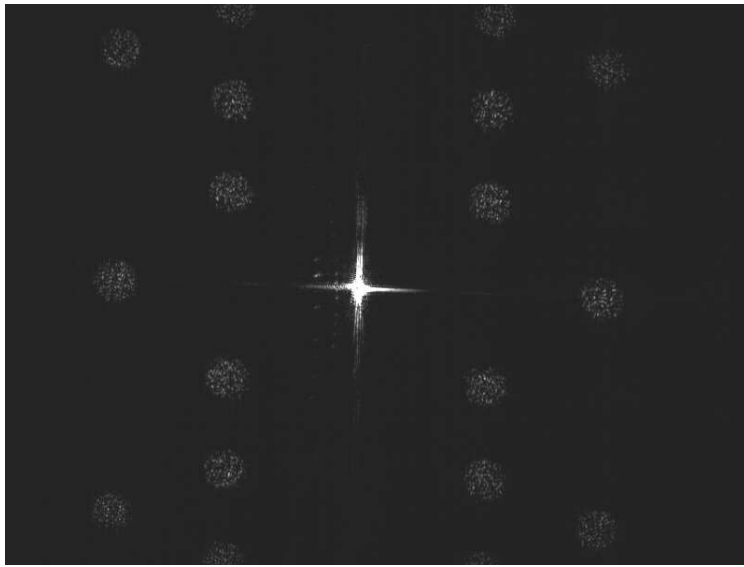
Figure 4.25:   Phase Hologram vs Phase Grating: The far field image created with a phase hologram loses intensity in each beam but all 18 original beams are present and remain focussed. (The brightness and contrast levels of this picture have been enhanced by 15% for printing of this document).
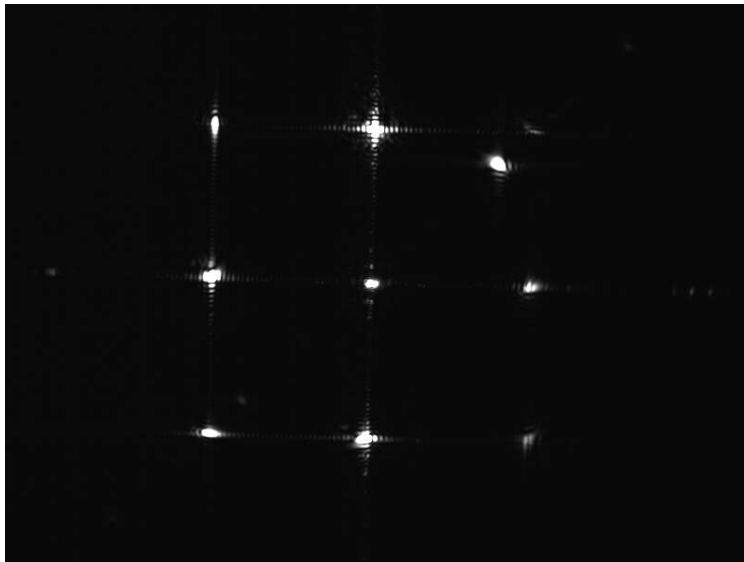
Figure 4.26:   Phase Hologram vs Phase Grating: Even a phase grating dividing a beam to 9 spots is affected by the incident beam moving off center of the SLM.

4.9   Summary

Data collected over 4 scenarios has shown that SLMs are viable candidates to replace gimballed mirror systems. Phase screens loaded to the SLMs provide the desired outcomes in the scenarios, but not without problems. Methods to mitigate these problems will be discussed in the next chapter.

V. Conclusions

5.1 Introduction

Research efforts contained in this document set out to develop and test methodologies to steer, split, and track a laser beam using SLM technology, seeking a nonmechanical alternative to gimballed mirror systems which tend to be bulky, have large power requirements, and are found to be maintenance intensive. Motivation for this research comes as the Air Force moves to high bandwidth optical communications and laser tracking devices in the air theater. The SLM uses a liquid crystal material to affect a change on the index of refraction through which the laser travels. Five objectives underlie this research. Objective one was to characterize the behavior and response of the SLM to ensure acceptable results. Second, demonstrating beam steering and splitting techniques by applying phase gratings to the SLM. Third, develop a methodology to vary the intensity levels of the split beams. The fourth objective was to demonstrate that the SLM could be used in a closed loop system to track a moving beam. The final objective involved the exploration of using phase holograms on SLMs to split and steer beams as an alternative to the phase grating methodology. An assessment of system quality can be made based on the interpretation of the collected data. By recognizing shortcomings of system performance, recommendations can be made to improve future operation.

5.2 Conclusions Drawn From Results

In this research, three scenarios were conducted for beam steering/splitting operations with a $15\mu m$ pitch SLM. A fourth scenario was conducted for beam tracking using a $7\mu m$ pitch SLM. Limitations with supporting equipment drove the need to use two different SLMs, however this did not affect the ability to demonstrate the viability of using an SLM in each scenario.

The $15\mu m$ device performed well in the beam steering and beam splitting scenarios using the blazed phase gratings. Also the SLM only has an 83% fill factor, it is able to steer the majority of light anywhere within it's field of view (radius of $5mrad$).

It was also demonstrated with this device that power can be shifted from one split beam to another by simply allocating more of the phase map (and thus the surface of the SLM) to the beam requiring more power. Results showed that the method of power shifting induces an added diffraction effect in the far field.

Phase holography was explored as a possible option for beam splitting and beam steering. The algorithm developed for this thesis proved to be quite inefficient, much lower than that of the phase grating technique. However, it was demonstrated that an incident beam could be split into at least 18 sub-beams by a single SLM using a phase hologram, paving the way for further research in this area. Another aspect of the hologram method is that when the incident beam starts to move off the face of the SLM, all beams are equally effected and slightly degrade in intensity. With the blazed grating technique, the same shift in the incident beam will cause a complete loss of some of the sub-beams, and induce a diffraction effect with some of the other beams. It was shown that a SLM using a phase hologram can maintain focuss to 18 sub-beams with as little as 60% of it's array illuminated. The same amount of illumination on an SLM using the grating technique to split will cause a complete loss to some beams and will de-focuss on others.

The $7\mu m$ SLMs performance has severely degraded with age, resulting in poor performance with the beam tracking system. The age effects caused poor efficiency in beam steering, however the concept was proven to be possible. The tracking system had a slow response time of $16sec$ and had issues with over-shoot when correcting the beam. Working with the controls aspect to speed up response time will make this a potential candidate for real world systems. The SLM has a $100ms$ load time, so the bottleneck is in the LabVIEW routine.

5.3   *Research Advances and Implications*

Some system technologies and concepts in this research have been used in unique ways. Integration of these items and ideas has led to a "first cut" at a functioning product. The use of spatial light modulators to replace gimballed mirror systems

in optical communications has been around for a few years. Until this point, the published literature had focussed on the ability to steer a beam with the inclusion of a few articles delving into beam splitting as well. The research conducted for this thesis builds upon this showing additional capabilities of the SLM including power intensity variation between split beams, using an SLM for beam tracking, and using phase holograms on an SLM to split and move beams.

The systems demonstrated in this research have positive implications for the Air Force. This research has shown that not only can the SLM replace the gimballed mirror system reducing size, weight, and power requirements, the SLM's flexibility allows for added benefits such as "one-to-many" communication capabilities with a single device. These benefits are significant to the development of next generation optical communications systems in the Air Force.

The SLM has great flexibility in its utility when working with narrowband light. There are several other applications that are worth mentioning. This research covered beam steering and beam tracking applications. Additionally, the SLM may be used as a wavefront correction device for higher order aberrations in monochromatic optical systems.

5.4   Recommendations for Future Work

There were two new applications attempted in this thesis, using phase holography on an SLM for beam splitting/steering and using an SLM for beam tracking/stablization. As neither the holography beam splitting/steering methods nor the beam tracking system performed ideally, several improvements to the current systems are available. These recommendations are all achievable using the current available technology.

Theory suggests that phase holography should have a higher performance with diffraction efficiency. The phase screens created with the iterative Fourier transform algorithm showed very low efficiency that didn't show much improvement with an

increase in the number of iterations run on the algorithm. This suggests that there is a need for some fine tuning of the algorithm. Also the masking technique used for the space domain constraints of the algorithm was quite restrictive and may have been cutting out a lot of far field energy. One recommendation is creating an adjustable masking scheme that becomes more restrictive as the number of iterations increase while maintaining conservation of energy.

AFRL/VSSV has the desire to use SLMs on communications satellites, thus requiring the SLM to have a lifetime of 15 years or more. Current SLMs have a much shorter lifespan, thus creating a need to seek methods to harden them against aging effects and expanding their longevity. It might turn out that phase gratings are more efficient for beam splitting/steering applications than phase holograms. However, at this point, this is still an unknown. One benefit that is shown in the research presented in this thesis, is that if an SLM is damaged in a way such that only half the SLM works, a beam can still be split into numerous sub-beams degraded only in intensity, resulting in a mission degradation but not mission failure. The phase grating technique experiencing the same malfunction will have a complete loss of some sub-beams in addition to degradation in others. Considering such a failure on a satellite where the SLM can't be replaced drives the need to push these ideas further.

The problems experienced with the beam tracking scenario that limited performance can be set into two categories: (1) poor performance due to a damaged SLM; and (2), algorithm performance. During this research, the focus was to get a system working well enough for proof of concept. It was shown that a closed loop optical system can use an SLM to correct and stabilize a moving beam. Performance can be enhanced in a number of ways. First, by fine-tuning and calibrating the centroid calculation algorithm in the Matlab script. The pixel size of the imaging system (the camera that grabs the image for processing in Matlab) needs to be taken into account and translated to the pixel size of the imaging system in the far field if error needs to be reduced to the sub-pixel level. A second enhancement to the system is to show the flexibility of the SLM and track two beams with a single device. This can be done by

dividing the SLM plane in half, dedicating one side to each beam. This should be a fairly trivial modification to the existing code. Building upon this, AFRL/SNJ has interest in migrating from tracking a laser beam to tracking/stabilizing an image with the SLM. Together with the conclusions of this research, the above recommendations make an image tracking device viable for a multitude of Air Force applications.

Appendix A. Matlab Phase Screen Generation and Simulation Code

A.1 Introduction

Beam steering, beam splitting, and intensity variation are all controlled mainly by loading proper phase screens to the spatial light modulator (SLM). Creating these screens is done through Matlab using several valuable functions and m-file scripts. These functions and scripts are presented here to facilitate future work and experiment repeatability. Five main programs are presented. The file "ScenarioGen.m" creates a scenario file for the beam tracking code. "BeamTracking.m" reads in the scenario file and creates a .avi output file that shows four screens; (1) the moving input beam; (2) the corrected beam; (3) the phase map created to create the beam; and (4) the xy correction error measured in pixels of the center of the image plane. "Hologram18spot.m" is an example of the iterative phase retrieval algorithm used to create the hologram phase screens for beam splitting and beam steering. "SplitScreen1mv.m" is the code that uses the blazed grating technique to simulate and create phase screens for the SLM for experiments in splitting and moving the sub-beams. "PowerSplit.m" is similar to "SplitScreen1mv.m" except that it varies the intensity of each split beam rather than moving them.

A.2 ScenarioGen.m

The following code comprises the "ScenarioGen.m" script file.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by Capt Michael J Perry, AFIT/ENP
% Updated: 8 Oct 04
%
% This function generates a series of images and associated
% phase screens that describe the path of a wandering beam
% in the far field. The output of this function will be used
% as the input to the SLM code that will keep this wandering
```

```
% beam on the center of the detector plane.
%
% The code prompts the user for the following inputs:
%       data_points(x,y)    the x-y location on the 400x400
%                           detector plane where the center of
%                           the beam is located. Valid points range
%                           from -200 to 200.
%
%       number_pixels       the size of the SLM in one dimention
%
%       num_points          the number of different beam locations
%                           in the scenario
%
% The output images are saved to a file called scenario_im.mat
% The output phase screens are saved to a file called scenario_ps.mat
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format long; number_pixels = ...
    input('Enter the size of SLM (only in 1 dimension, i.e. 512):  ');
num_points = input('Enter the number of points in the scenario:  ');
data_points = []; for x = 1:num_points
        data_points(x,1) = ...
            input('Enter distance from center in x direction:  ');
        data_points(x,2) = ...
            input('Enter distance from center in y direction:  ');
end
% call function to calibrate how much phase shift is represented by
% each pixel of the SLM
```

```matlab
[mltplr_x, mltplr_y] = calibrate_sys(number_pixels);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% create a .sif file to keep phase screens to be loaded onto the SLM
fid = fopen('G:\Thesis\sif_files\TiltSequence.sif','w');
% build image seen by phase screen and SLM (want to fill the SLM)
I0 = 1;
source = ones(number_pixels/2);
for l =
1:number_pixels/2
    for k = 1:number_pixels/2
        if sqrt(((number_pixels/4)-k)^2+((number_pixels/4)-l)^2)...
                > number_pixels/4
            source(k,l) = 0;
        else
            source(k,l) = I0;
        end
    end
end p = 2048;
pw = zeros(p);
ll = p/2-number_pixels/4+1;
ul =...
p/2+number_pixels/4;
pw(ll:ul,ll:ul) = source;
for xx = 1:num_points
    x_shift2 = data_points(xx,1);
    y_shift2 = data_points(xx,2);
    % multiply by correction factor which is how much of a
    % 2*pi phase shift each pixel represents
```

```matlab
x_phase_shift = mltplr_x * abs(x_shift2);

y_phase_shift = mltplr_y * abs(y_shift2);

row_shift = linspace(x_phase_shift,0, 512);

col_shift = (linspace(y_phase_shift,0, 512))';

phase_screen_corr = repmat(row_shift,512,1) + ...

    repmat(col_shift,1,512);

% logic to get phase screen oriented to correctly

if atan2(y_shift2,x_shift2) > 1.5708

    phi_corr = phase_screen_corr;

elseif atan2(y_shift2,x_shift2) < 0 & ...

        atan2(y_shift2,x_shift2) > -1.5708

    phi_corr = rot90(phase_screen_corr,2);

elseif atan2(y_shift2,x_shift2) <  -1.5708

        phi_corr = flipud(phase_screen_corr);

        %phi_corr = fliplr(phi_corr);

else

    phi_corr = fliplr(phase_screen_corr);

end

%create the phase screen

phi_corr2 = mod(phi_corr, 2*pi);

%change the phase screen to a format recognized by the BNS SLM and

%store it in a file to be loaded to the SLM later.

 Screen = rad2val(phi_corr2,[0:255]);

    Screen = Screen/255;

    name = ...

        ['G:\Thesis\Tilt_Sequence\SLM_Screens\TiltWaves',...

        num2str(xx) '.bmp'];

    imwrite(Screen,name,'bmp');
```

```
        fprintf(fid,'%s\n',...

            ['C:\Blink\Perry_Runs\Tilt_Sequence\SLM_Screens\TiltWaves'...

            num2str(xx) '.bmp' ', ' 'C:\Blink\LUT_Files\slm6041.LUT']);

    %calculations for the simulation portion

    phi2_corr = zeros(p);

    ll2 = p/2-number_pixels/2+1;

    ul2 = p/2+number_pixels/2;

    phi2_corr(ll2:ul2,ll2:ul2) = phi_corr2;

    pwabb_corr = pw.*exp(j*phi2_corr);

    pwim_corr = fftshift(fft2((pwabb_corr)));

    pwim2_corr = pwim_corr.*conj(pwim_corr);

    pwim3_corr = pwim2_corr(825:1224,825:1224);

    lut_im{xx} = pwim3_corr;

    lut_ps{xx} = phi_corr;

end fclose(fid); format short; savefile1 = 'scenario_im.mat';

save(savefile1, 'lut_im'); savefile2 = 'scenario_ps.mat';

save(savefile2, 'lut_ps');
```

The "ScenarioGen.m" file saves its outputs into a Matlab file so that they can be recalled later by the "BeamTracking.m" code.

A.3   BeamTracking.m

The following code comprises the "BeamTracking.m" script file.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Written by Capt Michael J Perry, AFIT/ENP

% Updated: 30 Sep 04

%

% This script reads in the scenario files built with the
```

```
% ScenarioGen.m file and tracks the moving beam
%
% The code prompts the user to name a .avi file for the output data.
% The output of this script is an .avi file that shows the code
% tracking the moving beam and the associated phase map used to
% correct the beam.  The error plots show for how far off center
% the corrected beam is.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all format long;
number_pixels = 512;     % number of pixels on the SLM in both x and y
filename3 = ...
    input('Enter filename for output data AVI including .avi extension: ','s');
aviobj3 = avifile(filename3, 'fps', 5);
%
% call function to calibrate how much phase shift is represented by
% each pixel of the SLM
[mltplr_x, mltplr_y] = calibrate_sys(number_pixels);
%
% Load scenario - lut will be the structure function that is loaded
% containing the information.  The length of lut is the number of points
% in the scenario and will drive the number of times this code is looped
load('scenario_ps.mat'); load('scenario_im.mat'); num_screens =
length(lut_im);
% set up arrays to capture error in centering of beam
error_x = []; error_y = [];
% This loop is to run the code several times to generate a avi file
% showing how the SLM code can track the movement of the beam
```

```matlab
for zz = 1:num_screens

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set up some variables...

lambda = 632.8e-9;       % wavelength of HeNe

gray_scale = 128;        % number of distinct gray scale levels

d_max = 7.68e-3;         % length and width of the SLM

pixel_size = d_max/number_pixels; uncor_beam = lut_im{zz};

% build image seen by phase screen and

% SLM (want to fill the SLM if possible)

I0 = 1; source = ones(number_pixels/2); for l = 1:number_pixels/2

   for k = 1:number_pixels/2

      if sqrt(((number_pixels/4)-k)^2+((number_pixels/4)-l)^2)...

            > number_pixels/4

         source(k,l) = 0;

      else

         source(k,l) = I0;

      end

   end

end p = 2048;

pw = zeros(p);

ll = p/2-number_pixels/4+1;

ul =

p/2+number_pixels/4;

tst = pw(ll:ul,ll:ul);

pw(ll:ul,ll:ul) =

source;

pw_image = pw(825:1224,825:1224);

 %this is to make the input image on the
```

```
%same scale as the output image

pwabb = pw.*exp(j);

 % reference point of where we want to

 % correct the beam to

pwim = fftshift(fft2((pwabb))); pwim2 = pwim.*conj(pwim); pwim3 =

pwim2(825:1224,825:1224);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% calculating centroid position of the beam

centroid_x = sum((sum(uncor_beam,1).*(1:400))...

    / (sum(sum(uncor_beam))));

centroid_y = sum(((sum(uncor_beam,2))'.*(1:400))...

    / (sum(sum(uncor_beam))));

x_shift2 = centroid_x - 200.50; y_shift2 = 200.50 - centroid_y;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% multiply by correction factor which is how much of a

% 2*pi phase shift each pixel represents.  The .22 parameter

% is an additional correction factor to get rid of a linear

% error in the calculations

x_phase_shift = (mltplr_x-.22) * abs(x_shift2); y_phase_shift =

(mltplr_y-.22) * abs(y_shift2);

% Generate the phase screen

row_shift = linspace(x_phase_shift,0, number_pixels); col_shift =

(linspace(y_phase_shift,0, number_pixels))'; phase_screen_corr =

repmat(row_shift,number_pixels,1)...

    + repmat(col_shift,1,512);

% logic to get phase screen oriented to correctly

if atan2(y_shift2,x_shift2) > 1.5708

    phi_corr = phase_screen_corr;
```

```
elseif atan2(y_shift2,x_shift2) < 0 & ...

        atan2(y_shift2,x_shift2) > -1.5708

    phi_corr = rot90(phase_screen_corr,2);

elseif atan2(y_shift2,x_shift2) <  -1.5708

        phi_corr = flipud(phase_screen_corr);

else

    phi_corr = fliplr(phase_screen_corr);

end

%make a modulo 2*pi version of the phase screen for the SLM

phi_corr = mod(phi_corr, 2*pi); new_phi = (lut_ps{zz}-phi_corr);

% zero padding for fft efficiency

phi2_corr = zeros(p); ll2 = p/2-number_pixels/2+1; ul2 =

p/2+number_pixels/2; phi2_corr(ll2:ul2,ll2:ul2) = new_phi;

% correct the beam back to the center of the display

pwabb_corr = pw.*exp(j*phi2_corr); pwim_corr =

fftshift(fft2((pwabb_corr))); pwim2_corr =

pwim_corr.*conj(pwim_corr); pwim3_corr =

pwim2_corr(825:1224,825:1224);

% checking to see if beam has been moved back to

% the center of the screen and determine how much

% error exists in both x and y

centroid2_x = sum((sum(pwim3_corr,1).*(1:400)) /...

    (sum(sum(pwim3_corr))));

centroid2_y = sum(((sum(pwim3_corr,2))'.*(1:400)) /...

    (sum(sum(pwim3_corr))));

temp_x = centroid2_x - 200.5; temp_y = 200.5 - centroid2_y; error_x

= [error_x temp_x]; error_y = [error_y temp_y];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% plotting the figures
%
% plot correction phase screen and the result of the correction
% applied to the original phase screen
h = figure(1); clf; subplot(10,10,[51:55,61:65,71:75,81:85,91:95]);
imagesc(phi_corr); axis square off colormap(flipud(gray))
title('Correcting phase screen')
subplot(10,10,[1:5,11:15,21:25,31:35,41:45]); imagesc(lut_im{zz})
axis square off colormap(flipud(gray)) title('Image of uncorrected
beam'); subplot(10,10,[6:10,16:20,26:30,36:40,46:50]);
imagesc(pwim3_corr) axis square off colormap(flipud(gray))
title('Image of corrected beam'); subplot(10,10,[66:70,76:80]);
plot(error_x) ylabel('Pixels') axis([1 num_screens -.5 .5])
title('Error in x direction') subplot(10,10,[86:90,96:100]);
plot(error_y) ylabel('Pixels') axis([1 num_screens -.5 .5])
title('Error in y direction') currentFrame3 = getframe(h); aviobj3 =
addframe(aviobj3, currentFrame3); end
%close avi files
aviobj3 = close(aviobj3);
```

Scenarios are saved as .avi files for later comparison with bench top system performance.

A.4   Hologram18spot.m

Below is the code for the Matlab script file "Hologram18spot.m."

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by Capt Michael J Perry, AFIT/ENP
% Updated: 10 Jan 05
%
```

```
% This function uses an iterative phase retreval algorithm to

% create a phase screen (only one at a time with this version)

% that splits a beam into multiple spots in the far field

%

% The following variables can be manipulated:

%       number_pixels       This is the diameter of the spot in the

%                           far field measured in pixels

%

%       number_iterations   How many times the iterative process

%                           is run

%

%       num_points          the number of different beam locations

%                           in the scenario

%

% The output images are saved to a file called scenario_im.mat

% The output phase screens are saved to a file called scenario_ps.mat

%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all

% create a .sif file to keep phase screens to be loaded onto the SLM

fid = fopen('G:\Thesis\sif_files\hologram18_200.sif','w');

% create image of the 18 spots I expect to see in far field

number_pixels = 26;      %diameter of spot

number_iterations = 200;      %number of times we loop through code

I0 = 1;

% create the spot

source = ones(number_pixels); for l = 1:number_pixels

   for k = 1:number_pixels
```

```matlab
        if sqrt(((number_pixels/2)-k)^2+ ...

                ((number_pixels/2)-l)^2) > number_pixels/2

            source(k,l) = 0;

        else

            source(k,l) = I0;

        end

    end

end

% placing the spots to build the desired far field image

% in this case 18 spots are created

p = 512; pw = zeros(p); lly = p/2-number_pixels/2+1; uly =

p/2+number_pixels/2; lly1 = p/2 - 150 - number_pixels/2+1; uly1 =

p/2 - 150 + number_pixels/2; lly2 = p/2 + 150 - number_pixels/2+1;

uly2 = p/2 + 150 + number_pixels/2; llx = p/2-number_pixels/2+1; ulx

= p/2+number_pixels/2; llx1 = p/2 - 150 - number_pixels/2+1; ulx1 =

p/2 - 150 + number_pixels/2; llx2 = p/2 + 150 - number_pixels/2+1;

ulx2 = p/2 + 150 + number_pixels/2; lly11 = p/2 - 80 -

number_pixels/2+1; uly11 = p/2 - 80 + number_pixels/2; lly22 = p/2 +

80 - number_pixels/2+1; uly22 = p/2 + 80 + number_pixels/2; llx11 =

p/2 - 120 - number_pixels/2+1; ulx11 = p/2 - 120 + number_pixels/2;

llx22 = p/2 + 120 - number_pixels/2+1; ulx22 = p/2 + 120 +

number_pixels/2; llx12 = p/2 - 180 - number_pixels/2+1; ulx12 = p/2

- 180 + number_pixels/2; llx23 = p/2 + 180 - number_pixels/2+1;

ulx23 = p/2 + 180 + number_pixels/2; llx13 = p/2 - 60 -

number_pixels/2+1; ulx13 = p/2 - 60 + number_pixels/2; llx24 = p/2 +

60 - number_pixels/2+1; ulx24 = p/2 + 60 + number_pixels/2;

pw(llx:ulx,lly1:uly1) = source; pw(llx:ulx,lly2:uly2) = source;

pw(llx1:ulx1,lly1:uly1) = source; pw(llx1:ulx1,lly2:uly2) = source;
```

```
pw(llx2:ulx2,lly1:uly1) = source; pw(llx2:ulx2,lly2:uly2) = source;

pw(llx11:ulx11,lly11:uly11) = source; pw(llx11:ulx11,lly22:uly22) =

source; pw(llx22:ulx22,lly11:uly11) = source;

pw(llx22:ulx22,lly22:uly22) = source; pw(llx12:ulx12,lly11:uly11) =

source; pw(llx12:ulx12,lly22:uly22) = source;

pw(llx23:ulx23,lly11:uly11) = source; pw(llx23:ulx23,lly22:uly22) =

source; pw(llx13:ulx13,lly11:uly11) = source;

pw(llx13:ulx13,lly22:uly22) = source; pw(llx24:ulx24,lly11:uly11) =

source; pw(llx24:ulx24,lly22:uly22) = source;

% pw_image acts as the mask used as the far field constraint

pw_image = pw; omega = fftshift(ifft2(fftshift(pw))); Amplitude = 1;

phi = ones(512); omega_prime = omega;

% Start the iterative process

for x = 1:number_iterations F_hat =

fftshift(fft2(fftshift(omega_prime)));

    F_hat_prime = F_hat./abs(F_hat);

% mask out unwanted intensities

F_hat_prime_masked = pw_image.*F_hat_prime; omega =

fftshift(fft2(fftshift(F_hat_prime_masked))); omega_prime =

Amplitude*omega./abs(omega); end

    %change the phase screen to a format recognized by the BNS SLM and

    %store it in a file to be loaded to the SLM later.

     Screen = rad2val(angle(omega),[0:255]);

        Screen = Screen/255;

        name = ...

            ['G:\Thesis\Tilt_Sequence\SLM_Screens\hologram18_200.bmp'];

        imwrite(Screen,name,'bmp');

        fprintf(fid,'%s\n',...
```

```
              ['C:\Blink\Perry_Runs\SLM_Screens\hologram18_200.bmp'...

              ', ' 'C:\Blink\LUT_Files\slm6037.LUT']);

% plot expected results to compare to lab experiment

figure imagesc(pw_image) title('desired far field image'); axis

square colormap(flipud(gray)) figure imagesc(angle(omega))

title('hologram phase screen') axis square colormap(flipud(gray))

figure imagesc(abs(F_hat)) title('output created from hologram')

axis square colormap(flipud(gray))

%close .sif file

fclose(fid);
```

The phase screens are saved as bitmaps to be loaded into the SLM.

A.5   SplitScreen1mv.m

The code for the "SplitScreen1mv.m" Matlab script is shown below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Written by Capt Michael J Perry, AFIT/ENP

% Updated: 20 Oct 04

%

% This function generates a series of images and associated phase screens

% that steer a laser beam in the far field.  This code

% takes a single beam and splits it into 4 beams, initially steering each

% of them in the same direction.  At a predetermined point, the code

% locks 3 of the beams in place while moving the 4th around the

% detector field.  This code shows the ability to steer the beams

% independently and the complete range of the steered beam

% The output of this code will be a series of phase screens

% and images showing the effects of these phase screens on a laser beam

% propagating to the far field.
```

```
%

% The code prompts the user for the following inputs:

%        data_points(x,y)     the x-y location on the 400x400

%                             detector plane where the center of

%                             the beams are located.  One set of

%                             parameters is entered for each point

%                             in the scenario.  Valid inputs are from

%                             -200 to 200 for the simulations.  Inputs

%                             ranging from -350 to 350 have been

%                             successfully made into phase screens and

%                             loaded onto the SLM in the lab.

%

%        num_points           the number of different beam locations

%                             in the scenario

%

% The output images are saved to a file called scenario_im.mat

% The output phase screens are saved to a file called scenario_ps.mat

%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

format long; number_pixels = 512; filename1 = input('Enter filename

for phase'...

    'screen only output data AVI including .avi extension: ','s');

aviobj1 = avifile(filename1, 'fps', 3); filename2 = input('Enter

filename for complete' ...

    'output data AVI including .avi extension: ','s');

aviobj2 = avifile(filename2, 'fps', 3); num_points = input('Enter

the number of points in the scenario:  '); data_points = []; for x =

1:num_points
```

```matlab
            data_points(x,1) = ...

                input('Enter distance from center in x direction:  ');

            data_points(x,2) = ...

                input('Enter distance from center in y direction:  ');

end

% call function to calibrate how much phase shift is represented by

% each pixel of the SLM

[mltplr_x, mltplr_y] = calibrate_sys(number_pixels);

%%%%%%%%%%%%%%%%%%%%%%%%%%%

% create a .sif file to keep phase screens to be loaded onto the SLM

fid = fopen('G:\Thesis\sif_files\SpltScrnSeq.sif','w');

% build image seen by phase screen and SLM (want to fill the SLM)

I0 = 1; source = ones(number_pixels); for l = 1:number_pixels

    for k = 1:number_pixels

        if sqrt(((number_pixels/2)-k)^2+((number_pixels/2)-l)^2)...

                > number_pixels/2

            source(k,l) = 0;

        else

            source(k,l) = I0;

        end

    end

end p = 2048; pw = zeros(p); ll = p/2-number_pixels/2+1; ul =

p/2+number_pixels/2; tst = pw(ll:ul,ll:ul); pw(ll:ul,ll:ul) =

source;

% The code is broken into two loops here

% the first loop breaks the beam into four and systematically

% moves all four beam out away from the center.

% The second loop keeps three of the beams still while moving
```

```matlab
% the fourth beam around the other three in an

% "around the world" fashion

for xx = 1:6

    x_shift2 = data_points(xx,1);

    y_shift2 = data_points(xx,2);

    % multiply by correction factor which is how much of a

    % 2*pi phase shift each pixel represents

    x_phase_shift = mltplr_x * abs(x_shift2);

    y_phase_shift = mltplr_y * abs(y_shift2);

    row_shift = linspace(x_phase_shift,0, 512);

    col_shift = (linspace(y_phase_shift,0, 512))';

    phase_screen_corr = repmat(row_shift,512,1) + repmat(col_shift,1,512);

    % logic to get phase screen oriented to correctly

    if atan2(y_shift2,x_shift2) > 1.5708

        phi_corr = phase_screen_corr;

    elseif atan2(y_shift2,x_shift2) < 0 & atan2(y_shift2,x_shift2) > -1.5708

        phi_corr = rot90(phase_screen_corr,2);

    elseif atan2(y_shift2,x_shift2) <  -1.5708

            phi_corr = flipud(phase_screen_corr);

            %phi_corr = fliplr(phi_corr);

    else

        phi_corr = fliplr(phase_screen_corr);

    end

    %create the phase screen

    phi_corr2 = mod(phi_corr, 2*pi);

    %split the phase screen into four parts to split the beam into four

    %parts

    section1 = phi_corr2(1:256,1:256);
```

```matlab
    section2 = rot90(section1);

    section3 = rot90(section2);

    section4 = rot90(section3);

    split_screen = [section1 section4

section2 section3];

    %change the phase screen to a format recognized by the BNS SLM and

    %store it in a file to be loaded to the SLM later.

     Screen = rad2val(split_screen,[0:255]);

        Screen = Screen/255;

         name = ...

            ['G:\Thesis\Tilt_Sequence\SLM_Screens\SplitScrn',...

            num2str(xx) '.bmp'];

        imwrite(Screen,name,'bmp');

        fprintf(fid,'%s\n',...

            ['C:\Blink\Perry_Runs\Tilt_Sequence\SLM_Screens\SplitScrn'...

            num2str(xx) '.bmp' ', ' 'C:\Blink\LUT_Files\slm6037.LUT']);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Apply the phase screen for simulation

    phi2_corr = zeros(p);

    ll2 = p/2-number_pixels/2+1;

    ul2 = p/2+number_pixels/2;

    phi2_corr(ll2:ul2,ll2:ul2) = split_screen;

    pwabb_corr = pw.*exp(j*phi2_corr);

    pwim_corr = fftshift(fft2((pwabb_corr)));

    pwim2_corr = pwim_corr.*conj(pwim_corr);

    pwim3_corr = pwim2_corr(825:1224,825:1224);

    lut_im{xx} = pwim3_corr;

    lut_ps{xx} = phi_corr;
```

```matlab
    h = figure(1);

    clf;

    subplot(10,10,[1:5,11:15,21:25,31:35,41:45]);

    imagesc(split_screen)

    axis square off

    colormap(flipud(gray))

    subplot(10,10,[6:10,16:20,26:30,36:40,46:50]);

    imagesc(pwim3_corr)

    axis square off

    colormap(flipud(gray))

    currentFrame2 = getframe(h);

    aviobj2 = addframe(aviobj2, currentFrame2);

    hh = figure(2);

    clf;

    imagesc(split_screen)

    axis square off

    colormap(flipud(gray))

    currentFrame1 = getframe(hh);

    aviobj1 = addframe(aviobj1, currentFrame1);

end

% Second loop starts here

for xx = 7:num_points

    x_shift2 = data_points(xx,1);

    y_shift2 = data_points(xx,2);

    x_phase_shift = mltplr_x * abs(x_shift2);

    y_phase_shift = mltplr_y * abs(y_shift2);

    row_shift = linspace(x_phase_shift,0, 512);

    col_shift = (linspace(y_phase_shift,0, 512))';
```

```matlab
    phase_screen_corr = repmat(row_shift,512,1) + repmat(col_shift,1,512);

    % logic to get phase screen oriented to correctly

    if atan2(y_shift2,x_shift2) > 1.5708

        phi_corr = phase_screen_corr;

    elseif atan2(y_shift2,x_shift2) < 0 & atan2(y_shift2,x_shift2) > -1.5708

        phi_corr = rot90(phase_screen_corr,2);

    elseif atan2(y_shift2,x_shift2) <  -1.5708

            phi_corr = flipud(phase_screen_corr);

            %phi_corr = fliplr(phi_corr);

    else

        phi_corr = fliplr(phase_screen_corr);

    end

    %create the phase screen

    phi_corr2 = mod(phi_corr, 2*pi);

    %split the phase screen into four parts to split the beam into four

    %parts

    section4 = rot90(rot90(rot90(phi_corr2(1:256,1:256))));

    split_screen = [section1 section4
section2 section3];

    %change the phase screen to a format recognized by the BNS SLM and

    %store it in a file to be loaded to the SLM later.

     Screen = rad2val(split_screen,[0:255]);

        Screen = Screen/255;

         name = ...

            ['G:\Thesis\Tilt_Sequence\SLM_Screens\SplitScrn',...

            num2str(xx) '.bmp'];

        imwrite(Screen,name,'bmp');

        fprintf(fid,'%s\n',...
```

```matlab
        ['C:\Blink\Perry_Runs\Tilt_Sequence\SLM_Screens\SplitScrn'...

        num2str(xx) '.bmp' ', ' 'C:\Blink\LUT_Files\slm6041.LUT']);
 %%%%%%%%%%%%%%%%%%%%%%%%%
% Apply the phase screen
phi2_corr = zeros(p);
ll2 = p/2-number_pixels/2+1;
ul2 = p/2+number_pixels/2;
phi2_corr(ll2:ul2,ll2:ul2) = split_screen;
pwabb_corr = pw.*exp(j*phi2_corr);
pwim_corr = fftshift(fft2((pwabb_corr)));
pwim2_corr = pwim_corr.*conj(pwim_corr);
pwim3_corr = pwim2_corr(825:1224,825:1224);
lut_im{xx} = pwim3_corr;
lut_ps{xx} = phi_corr;
h = figure(1);
clf;
subplot(10,10,[1:5,11:15,21:25,31:35,41:45]);
imagesc(split_screen)
axis square off
colormap(flipud(gray))
subplot(10,10,[6:10,16:20,26:30,36:40,46:50]);
imagesc(pwim3_corr)
axis square off
colormap(flipud(gray))
currentFrame2 = getframe(h);
aviobj2 = addframe(aviobj2, currentFrame2);
hh = figure(2);
clf;
```

```
    imagesc(split_screen)

    axis square off

    colormap(flipud(gray))

    currentFrame1 = getframe(hh);

    aviobj1 = addframe(aviobj1, currentFrame1);

end

%close .sif file

fclose(fid);

%close avi files

aviobj1 = close(aviobj1); aviobj2 = close(aviobj2);
```

Phase screens are saved at bitmaps to be loaded into the SLM.

A.6   PowerSplit.m

The following code comprises the "PowerSplit.m" script.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Written by Capt Michael J Perry, AFIT/ENP

% Updated: 21 Oct 04

%

% This function will split a beam into 4 beams, move

% them independantly and vary the intensity of each beam.

%

% The code prompts the user for the following inputs:

%       data_points(x,y)    the x-y location on the 400x400

%                           detector plane where the center of

%                           the beams are located.  Valid entries

%                           range from -200 to 200

%

%       number_pixels       the size of the SLM in one dimention
```

```
%

%       num_points              the number of different beam locations

%                               in the scenario

%

% The output images are saved to a file called scenario_im.mat

% The output phase screens are saved to a file called scenario_ps.mat

%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

format long; number_pixels = 512; num_points = input('Enter the

number of points in the scenario:  '); data_points1 = []; for x =

1:num_points

        data_points(x,1) = ...

            input('Enter distance from center in x direction:  ');

        data_points(x,2) = ...

            input('Enter distance from center in y direction:  ');

end

% call function to calibrate how much phase shift is represented by

% each pixel of the SLM

[mltplr_x, mltplr_y] = calibrate_sys(number_pixels);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% create a .sif file to keep phase screens to be loaded onto the SLM

fid = fopen('G:\Thesis\sif_files\PwrShftSeq.sif','w');

% build image seen by phase screen and SLM (want to fill the SLM)

I0 = 1; source = ones(number_pixels); for l = 1:number_pixels

   for k = 1:number_pixels

      if sqrt(((number_pixels/2)-k)^2+((number_pixels/2)-l)^2)...

              > number_pixels/2

         source(k,l) = 0;
```

```
        else

            source(k,l) = I0;

        end

    end

end p = 2048; pw = zeros(p); ll = p/2-number_pixels/2+1; ul =

p/2+number_pixels/2; tst = pw(ll:ul,ll:ul); pw(ll:ul,ll:ul) =

source; for xx = 1:num_points

    x_shift2 = data_points(xx,1);

    y_shift2 = data_points(xx,2);

    % multiply by correction factor which is how much of a

    % 2*pi phase shift each pixels represents

    x_phase_shift = mltplr_x * abs(x_shift2);

    y_phase_shift = mltplr_y * abs(y_shift2);

    row_shift = linspace(x_phase_shift,0, 512);

    col_shift = (linspace(y_phase_shift,0, 512))';

    phase_screen_corr = repmat(row_shift,512,1) + ...

        repmat(col_shift,1,512);

    % logic to get phase screen oriented to correctly

    if atan2(y_shift2,x_shift2) > 1.5708

        phi_corr = phase_screen_corr;

    elseif atan2(y_shift2,x_shift2) < 0 & atan2(y_shift2,x_shift2) > -1.5708

        phi_corr = rot90(phase_screen_corr,2);

    elseif atan2(y_shift2,x_shift2) <  -1.5708

            phi_corr = flipud(phase_screen_corr);

            %phi_corr = fliplr(phi_corr);

    else

        phi_corr = fliplr(phase_screen_corr);

    end
```

```matlab
%create the phase screen

phi_corr2 = mod(phi_corr, 2*pi);

%split the phase screen into four parts to split the beam into four

%parts.  For intensity variation, the values of phi_corr must

%be manually adjusted for each screen.

section1 = phi_corr2(1:450,1:250);

temp2 = rot90(phi_corr2);

section2 = temp2(1:62,1:250);

temp3 = rot90(temp2);

section3 = temp3(1:450,1:262);

temp4 = rot90(temp3);

section4 = temp4(1:62,1:262);

% The split_screen variable must be adjusted to match the

% section sizes above.

split_screen = zeros(512);

split_screen(1:450,1:250)=section1;

split_screen(451:512,1:250)=section2;

split_screen(1:450,251:512)=section3;

split_screen(451:512,251:512)=section4;

%change the phase screen to a format recognized by the BNS SLM and

%store it in a file to be loaded to the SLM later.

 Screen = rad2val(split_screen,[0:255]);

    Screen = Screen/255;

    name = ...

        ['G:\Thesis\SLM_Screens\PwrShftScrn19.bmp'];

    imwrite(Screen,name,'bmp');

    fprintf(fid,'%s\n',...

        ['C:\Blink\Perry_Runs\SLM_Screens\PwrShftScrn19.bmp'...
```

```
        ', ' 'C:\Blink\LUT_Files\slm6037.LUT']);
%%%%%%%%%%%%%%%%%%%%%%%%%%
% Apply the phase screen for comparison to lab experiment
phi2_corr = zeros(p);
ll2 = p/2-number_pixels/2+1;
ul2 = p/2+number_pixels/2;
phi2_corr(ll2:ul2,ll2:ul2) = split_screen;
pwabb_corr = pw.*exp(j*phi2_corr);
pwim_corr = fftshift(fft2((pwabb_corr)));
pwim2_corr = pwim_corr.*conj(pwim_corr);
pwim3_corr = pwim2_corr(825:1224,825:1224);
lut_im{xx} = pwim3_corr;
lut_ps{xx} = phi_corr;
h = figure(1);
clf;
subplot(10,10,[1:5,11:15,21:25,31:35,41:45]);
imagesc(split_screen)
axis square off
colormap(flipud(gray))
subplot(10,10,[6:10,16:20,26:30,36:40,46:50]);
imagesc(pwim3_corr)
axis square off
colormap(flipud(gray))
hh = figure(2);
clf;
imagesc(split_screen)
axis square off
colormap(flipud(gray))
```

```
end
```

```
%close .sif file
```

```
fclose(fid);
```

The phase screens are saved as bitmaps to be loaded into the SLM.

Appendix B. LabVIEW System Layout

B.1 Introduction

The beam tracking scenario demonstrates that a SLM can be used in a closed loop fashion to track a moving laser beam. The system is setup as described in Chapter 3 of this document and initially aligned such that the laser is centered on both cameras. Once the system is running, the tip/tilt mirror is manually adjusted in a random pattern simulating a randomly moving beam over the receiving optics (the cameras). The SLM, controlled by a LabVIEW routine, corrects for the added tilt to the laser, redirecting the beam back to the center of the CMOS camera.

B.2 LabVIEW

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as test equipment or controls devices. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data.

In LabVIEW, a new program starts by building a user interface, or front panel, with controls and indicators as shown in Figure B.1. After the user interface is built, code is added using VIs, structures and scripts to control the front panel objects. This code is contained in the block diagram of the VI as seen in Figure B.2.

B.3 System Description

B.3.1 User Interface.   The user interface of the beam tracking VI shown in Figure B.1 contains two image screens. The top image screen shows the view seen by the DVI camera simulating the receiving optics of the tracking system. The second image screen on the user interface shows the phase screen developed by the code and placed onto the SLM to correct the beam back to the center. Figure B.1 shows the beam off center and the resulting phase screen. The scaling of the image is trimmed to $512x512$ to match the array size of the SLM, making processing more efficient,
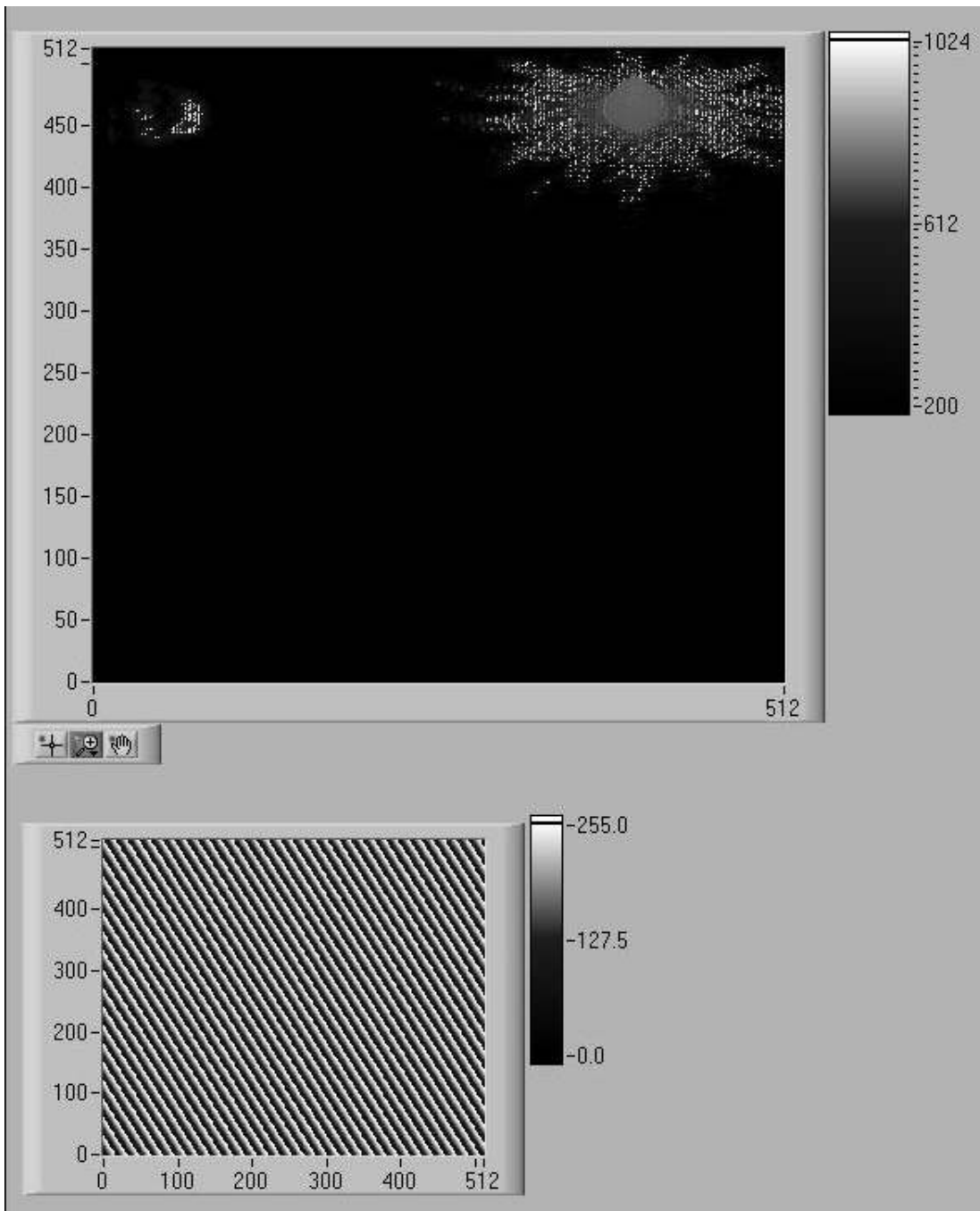
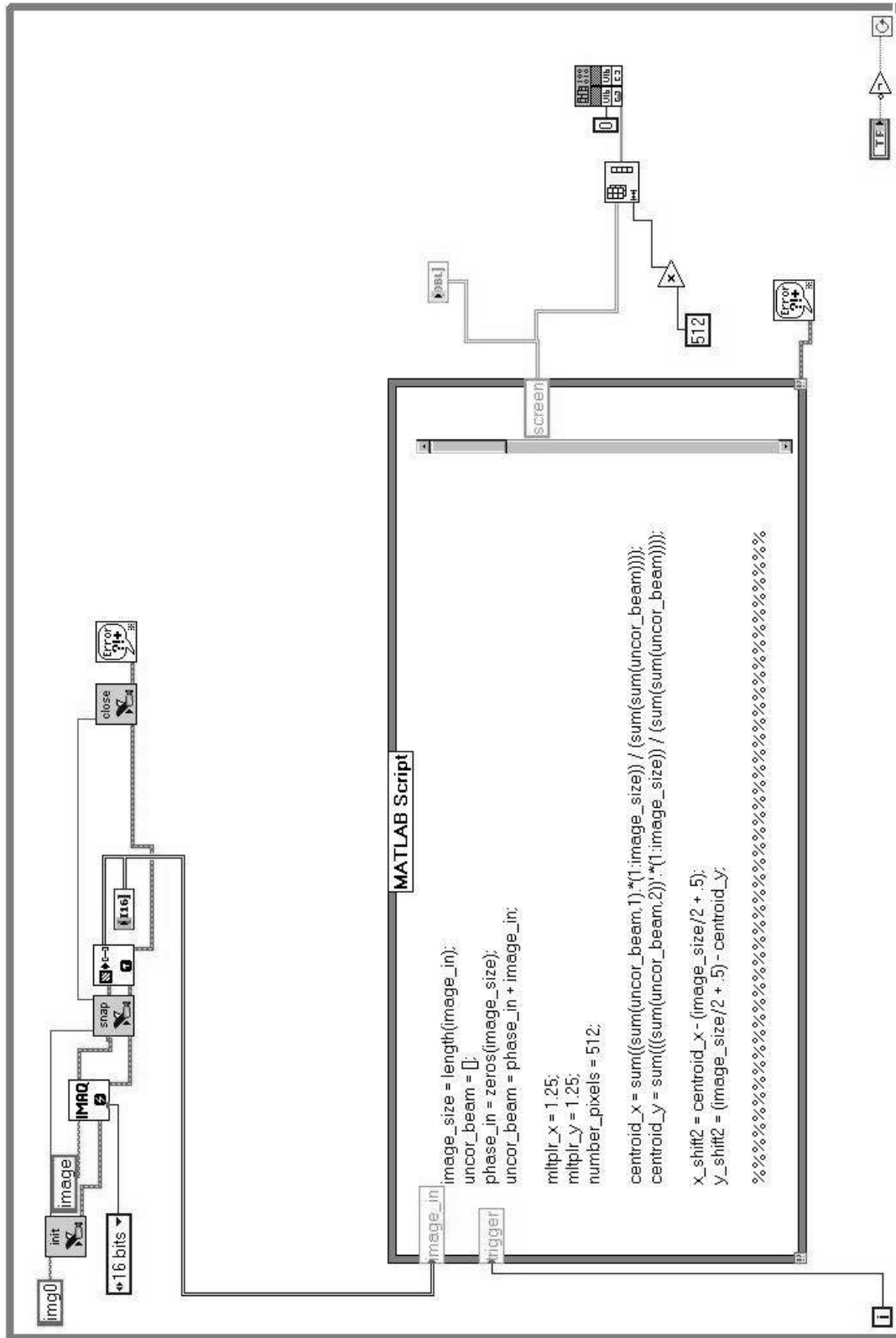Figure B.1:   User interface for LabVIEW beam tracking routine

Figure B.2: Block diagram of the LabVIEW beam tracking routine

however other image sizes can be used. Examples of images for the corrected beam taken by the CMOS camera are shown in Chapter 4.

B.3.2 Block Diagram. The block diagram shown in Figure B.2 is the main body of the VI which does the physical tracking and correcting. There are 7 other block diagram screens not shown that play a support role with the beam tracking. Routines not shown include turning on and initializing the SLM, the loading of the phase map to the SLM , and purging the SLM memory prior to shutting it down when the VI is stopped. These supporting routines were all developed under another program by AFRL/SNJM and do not play a part in the actual research.

Referring to Figure B.2, starting in the upper left corner of the figure and stepping through the routine in order of occurrence, the following describes each block:

- **img0**: T/F trigger to start routine.

- **init**: Initializes the DVC camera.

- **16 bits**: Intensity resolution of the camera.

- **image**: T/F trigger whether or not an image is present.

- **IMAQ**: VI that interfaces with and controls the camera software.

- **snap**: Captures an image from the camera.

- **T**: Translates the image into a form recognizable by Matlab.

- **close**: refreshes the camera software memory for the next picture. The camera will experience a buffer overflow if this block is not included.

- **error**: Error checking in the image capture sequence.

- **I16**: Passes the image seen by the camera to the user interface screen described above.

- **MATLAB Script**: Matlab code that takes in the image, calculates the location of the beam in reference to the center of the image and passes back a phase screen to be passed to the SLM for beam correction.

- **DBL**: Passes the image of the phase screen to the user interface described above.

The next block in the path is a VI that converts the phase screen to a format recognized by the SLM and the last block loads the phase screen onto the SLM. The remaining blocks in the figure not mentioned are necessary to keep the routine running in a continuous state until user intervention.

B.4 Code used in the Matlab script

The following code is a subsection of the scripts shown in Appendix A with just a few lines of code added to handle variable exchanges between Matlab and LabVIEW.

```
image_size = length(image_in); uncor_beam = []; phase_in =

zeros(image_size); uncor_beam = phase_in+image_in; mltplr_x=1.25;

mltplr_y=1.25; number_pixels=512; centroid_x =

sum((sum(uncor_beam,1).*(1:image_size)) /...

    (sum(sum(uncor_beam))));

centroid_y = sum(((sum(uncor_beam,2))'.*(1:image_size)) /...

    (sum(sum(uncor_beam))));

x_shift2 = centroid_x - (image_size/2 + .5); y_shift2 =

(image_size/2 + .5) - centroid_y;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x_phase_shift = (mltplr_x-.22) * abs(x_shift2); y_phase_shift =

(mltplr_y-.22) * abs(y_shift2);

% Generate the phase screen

row_shift = linspace(x_phase_shift,0, number_pixels); col_shift =

(linspace(y_phase_shift,0, number_pixels))'; phase_screen_corr =
```

```
repmat(row_shift,number_pixels,1)...

    + repmat(col_shift,1,512);

% logic to get phase screen oriented to correctly

if atan2(y_shift2,x_shift2) > 1.5708

    phi_corr = phase_screen_corr;

elseif atan2(y_shift2,x_shift2) < 0 & atan2(y_shift2,x_shift2) >

-1.5708

    phi_corr = rot90(phase_screen_corr,2);

elseif atan2(y_shift2,x_shift2) <  -1.5708

        phi_corr = flipud(phase_screen_corr);

else

    phi_corr = fliplr(phase_screen_corr);

end

%make a modulo 2*pi version of the phase screen for the SLM

phi_corr = mod(phi_corr, 2*pi);

    Screen = rad2val(split_screen,[0:255]);
```

An important point with this code involves the variables mltplr_x and mltplr_y. These two variables compensate for the pixel size of the CMOS camera and need to be adjusted if a different camera is used. Recalling the beam tracking results mentioned in Chapter 4, limited success was achieved with these variables because they were determined by trial and error. It is recommended that future work with this VI should include scripting an algorithm in the Matlab code that prompts the user to input the camera pixel size, allowing Matlab calculate the compensation factor. This will eliminate the systematic error seen in the results mentioned in Chapter 4.

Bibliography

1. BNS. "Beam Steering Using Liquid Crystals", 2001.

2. BNS. *User Guide: XY Phase Series Spatial Light Modulators Revision 2.8*, 2004.

3. Boyd, Robert W. *Nonlinear Optics.* Academic Press, San Diego, CA, second edition, 2003.

4. Brooks, Mathew. *Atmospheric Simulation Using a Liquid Crystal Wavefront Controlling Device.* Master's thesis, Air Force Institute of Technology, March 2004.

5. Emil, Hallstig, Johan Stigwall, Mikael Lindgren, and Lars Sjoqvist. "Laser beam steering and tracking using a liquid crystal spatial light modulator". *SPIE*, 5087:13–23, 2003.

6. Goodman, Joseph W. *Introduction to Fourier Optics.* McGraw Hill, second edition, 1996.

7. Goodman, Joseph W. *Statistical Optics.* John Wiley and Sons, Inc., 2000.

8. Harris, Scott. "Characterization and application of a liquid crystal beam steering device". *SPIE*, 4291:109–119, 2001.

9. Hecht, Eugene. *Optics.* Addison Wesley, fourth edition, 2002.

10. Iam-Choon, Khoo and Shin-Tson Wu. *Optics and Nonlinear Optics of Liquid Crystals.* World Scientific Publishing Co. Pte. Ltd., 1993.

11. Kyongsik, Choi, Hwi Kim, and Byoungho Lee. "Synthetic phase holograms for auto-stereoscopic image displays using a modified IFTA". *Optics Express*, 12(11):2454–2462, May 2004.

12. McManamon, Paul F. and Edward A. Watson. "Nonmechanical beam steering for passive sensors". *SPIE*, 4369:140–148, 2001.

13. Titas, Chuck. *Beam Steering.* Technical report, Bos Labs, Kent State University.

14. Welsh, Byron M., Michael C. Roggemann, and Greg L. Wilson. "Phase retrieval-based algorithms for far field beam steering and shaping". *SPIE*, 3763:11–22, July 1999.

15. Yariv, Amnon and Pochi Yeh. *Optical Waves in Crystals.* John Wiley and Sons, Inc., 1984.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
21–03–2005	Master's Thesis	Sept 2003 — Mar 2005

4. TITLE AND SUBTITLE

Using liquid crystal spatial light modulators for closed loop tracking and beam steering with phase holography

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Perry, Michael, J., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GEO/ENG/05-02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/SNJM
Attn: Mr. Robert J Feldmann
3109 Hobson Way, Bldg 622
WPAFB OH 45433-7700
DSN: 785-9614 x232

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Optical Phased Array (OPA) technology offers advantages in the reduction of size, weight, and power of optical steering devices. Nematic liquid crystal (LC) spatial light modulators (SLMs) have been studied as a potential candidate for building non-mechanical OPAs. They can steer a laser beam and split the beam into multiple beams. This thesis builds upon the prior research showing each split beam can be individually controlled, including variation in intensity. A closed loop tracking scenario shows the flexibility of the SLM by tracking and stabilizing an incoming beam. Results show that applying a phase grating to the SLM has limitations with diffraction and fringing when the SLM is divided into sub-apertures during beam splitting, forcing trade-offs in performance. An iterative Fourier transform algorithm is proposed to overcome the limitations by creating a phase hologram that steers and splits the beam without subdividing the SLM, seeking to minimize the aforementioned effects by allowing more efficient use of the LC array. Results show the beam can be split and steered by a phase hologram and is compared to simulation and the phase grating technique.

15. SUBJECT TERMS

Beam Steering, Beam Splitting, Spatial Light Modulator, SLM, Beam Tracking, Optical Phased Array, Phase Holograms

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj Matthew Goda (ENG)
U	U	U	UU	107	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255–3636, ext 4614

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18