



LIMITATIONS OF SEGMENTED WAVEFRONT CONTROL DEVICES
IN EMULATING OPTICAL TURBULENCE

THESIS

Michael D. Plourde, Captain, USAF

AFIT/GEO/ENG/08-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

LIMITATIONS OF SEGMENTED WAVEFRONT CONTROL DEVICES
IN EMULATING OPTICAL TURBULENCE

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 D. Plourde, B.S.E.E.
Captain, USAF

March 2008

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

LIMITATIONS OF SEGMENTED WAVEFRONT CONTROL DEVICES
IN EMULATING OPTICAL TURBULENCE

Michael D. Plourde, B.S.E.E.
Captain, USAF

Approved:

/signed/

March 2008

Capt Jason Schmidt, PhD (Chairman)

date

/signed/

March 2008

LtCol Steven Fiorino, PhD (Member)

date

/signed/

March 2008

Dr. Eric Magee (Member)

date

Abstract

Using a device to act as a surrogate for atmospheric turbulence in a laboratory is necessary to build and test optical systems for imaging, lidar, laser weapons, and laser communications. Liquid-crystal spatial light modulators (LC SLMs) and segmented micro-electro-mechanical-system (MEMS) deformable mirrors (DMs) are commonly used devices for altering wavefronts in order to simulate a portion of atmospheric turbulence. The best location of these devices was theoretically analyzed to obtain the broadest possible range of atmospheric conditions. It was found that two phase screens should be placed at the beginning of the optical path to achieve maximum turbulence strength for an incident plane wave. It was also revealed that a layered model for atmospheric turbulence strength can be represented by the atmospheric coherence diameter that a single segmented wavefront control device can produce. The limitations of pixelation effects on a segmented wavefront control device were investigated theoretically. The results of this analysis were then confirmed in simulation. It was found that while LC SLMs with high bandwidth have almost no adverse affects from pixelation, segmented MEMS DMs have limitations as a result of the number of mirror segments on a DM. The performance capabilities of several available commercial devices are better understood as a result of this research.

Acknowledgements

I would like to thank the Air Force Institute of Technology for providing me the opportunity to pursue an advanced degree. The experience and education gained during this process will surely be beneficial in all my future endeavors.

I would like to thank my committee members, LtCol Steven Fiorino and Dr. Eric Magee, who have helped shape this research along its successful path. I also want to sincerely thank my research advisor, Capt Jason Schmidt, for his countless contributions. His open door policy was extended many times into the evenings and weekends when he was always available to provide guidance and assistance. I would also like to commend him for his patience and understanding in helping me to fully grasp some of the complex concepts investigated in this research. I also would like to thank my fellow classmates who were always there to make me laugh and helped me truly enjoy the experience of graduate level research and studies.

None of this would have been possible without the encouragement and support of my parents. They gave me the skills necessary to succeed at any task. Most importantly, I need to offer my utmost gratitude to my best friend and editor, my wife. Her love and support during this process was unwavering and she always understood the late nights and long weekends spent locked up working. Also, her ability to get me to stop procrastinating and go back to work is the reason I was able to complete this monumental challenge. To that I say, “Thank you” and “I love you”. Last, but certainly not least, I want to thank my son who came into this world only one year ago. He is the reason I always had a smile on standby when things were the most difficult. All the hard work put forth in this research has been so that his Dad-Dad will be there on future evenings and weekends when he needs me most.

Michael D. Plourde

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
List of Figures	viii
List of Tables	x
List of Symbols	xi
List of Abbreviations	xiii
 I. Introduction	 1
1.1 Motivation	1
1.2 Goals	3
1.3 Thesis Outline	4
 II. Theory and Review	 5
2.1 Atmospheric Properties	5
2.2 Modeling the Atmosphere	9
2.2.1 Layered Model of Turbulence	9
2.2.2 Phase Screen Generation: Fourier Series Method	11
2.3 Zernike Polynomials	12
2.4 Wavefront Control Devices	17
2.4.1 Liquid-crystal Spatial Light Modulators	17
2.4.2 Noted Limitations	21
2.4.3 Deformable Mirrors	23
2.4.4 Alternative Methods	26
 III. Theoretical Analysis	 28
3.1 Phase Screen Locations	28
3.2 Minimum Atmospheric Coherence Length	31
3.2.1 Unit Analysis	33
3.2.2 Evaluation Over a Single Point in Space	34
3.2.3 Evaluation Over an Aperture	37

	Page
IV. Simulation and Analysis	49
4.1 Creating Phase Screens	49
4.2 Pixelation of Phase Screens	51
4.3 Finding Phase Variance	53
4.4 Results	54
4.4.1 Minimum Possible Atmospheric Coherence Length in Simulation	55
V. Conclusions and Future Work	59
5.1 Results and Conclusions	59
5.1.1 Limitations of Current Low-cost Wavefront Control Devices	59
5.1.2 Theoretically Emulating the Atmosphere	60
5.1.3 Emulating the Atmosphere in Simulation	61
5.1.4 Discrepancies Between Theoretical and Simulation Analysis	62
5.1.5 Practicality of Results	66
5.2 Future Work	68
Appendix A. Matlab Code to Compute Theoretical and Simulation Re- sults	70
Appendix B. Matlab Code to Create Phase Screens with the Fourier Series Method	88
Appendix C. Matlab Code to Pixelate and Compute Statistics of Phase Screens	90
Bibliography	93
Vita	96

List of Figures

Figure		Page
2.1.	Plot of common atmospheric refractive index PSDs	8
2.2.	Split-step beam propagation for two phase screens	10
2.3.	Examples of Zernike aberrations present in a turbulent wavefront	14
2.4.	Creating a turbulent wavefront by summing Zernike modes . .	15
2.5.	Residual phase of a turbulent wavefront through subtraction of Zernike modes	16
2.6.	Effects of applied voltage on molecular alignment inside a trans- missive NLC SLM	19
3.1.	Rytov number contour plot	30
3.2.	$r_{0_{\min}}$ as a function of Nyquist bandlimit for a single point on a wavefront control device	36
3.3.	Comparison of the derivation for $\sigma_{\phi}(\eta)$ to the accepted theoretical value	43
3.4.	Zernike modes required to reach investigated threshold levels .	44
3.5.	Number of pixels across a device theoretically required to reach investigated threshold levels	46
3.6.	$r_{0_{\min}}$ as a function of missing $\sigma_{\phi_{PTR}}$ for theoretical results . . .	48
4.1.	Example of a 512×512 pixel PTR phase screen	50
4.2.	Sample grid used to pixelate phase screens	52
4.3.	Examples of down-sampled PTR phase screens	56
4.4.	Plot of normalized $\sigma_{\phi_{pres}}(\eta)$ comparing simulation to theory . .	57
4.5.	Minimum atmospheric coherence length as a function of missing PTR phase standard deviation for simulation results	58
5.1.	Estimation of the filter functions resulting in discrepancies be- tween the theoretical and simulation results	64
5.2.	Discrepancy in theoretical vs. simulation results from sharp pixel edges	65

Figure		Page
5.3.	Summation plot of $\sigma_{\phi_{\text{PTR}_{\text{missing}}}}$ vs. minimum number of pixels required for several values of D/r_0	67

List of Tables

Table		Page
2.1.	First eleven Zernike polynomials and their common names . . .	13
3.1.	Threshold percentages investigated	40
3.2.	Zernike modes required to reach investigated threshold levels .	45
3.3.	Number of pixels across a device theoretically required to reach investigated threshold levels	47
4.1.	Simulation results for required number of pixels across a device	54
5.1.	Comparison of theoretical and simulation results for required number of pixels to reach investigated threshold levels	62
5.2.	Comparison of theoretical and simulation results for $r_{0_{\min}}$ of a 7×7 pixel wavefront control device.	64

List of Symbols

Symbol		Page
n	refractive index	5
$\mathbf{R}$	three-dimensional spatial coordinate (m)	5
λ	optical wavelength (μm)	5
P	pressure (mbar)	5
T	temperature (K)	5
L_0	outer scale of turbulence (m)	5
l_0	inner scale of turbulence (m)	5
$D_n(\mathbf{R})$	refractive index structure function	6
$B_n(\mathbf{R})$	refractive index covariance function	6
C_n^2	refractive index structure constant ($m^{-2/3}$)	7
$\Phi_n(\boldsymbol{\kappa})$	three-dimensional refractive index power spectral density .	7
κ	radial polar component of the three-dimensional angular spatial frequency vector (rad/m)	7
$\boldsymbol{\kappa}$	three-dimensional angular spatial frequency vector	7
r_0	atmospheric coherence diameter (m)	8
$\sigma_{\mathcal{R}}^2$	Rytov number	8
L	optical path length (m)	8
Δz_i	total length of atmosphere a phase screen represents (m) .	9
$\mathbf{c}_{n,m}$	Fourier series coefficients	12
$\phi_k(x, y)$	phase calculated with Fourier series method	12
D	aperture diameter (m)	12
$Z_{m,n}$	Zernike polynomials	12
ρ	transverse component of the two-dimensional spatial vector	13
$R_n^m(2\rho/D)$	radial functions of Zernike polynomials	13
m	azimuthal order of Zernike polynomial	13

Symbol		Page
n	radial order of Zernike polynomial	13
i	Zernike mode index number	13
$\phi(D\rho, \theta)$	phase of a wavefront	14
a_i	Zernike coefficient of the i^{th} Zernike mode	14
$W(x, y)$	aperture function for a circle of unit radius	14
$\delta_{m_i m_j}$	Kronecker delta function	16
$\Gamma(x)$	Gamma function	16
ϵ_i^2	mean square residual phase error (rad ²)	17
$F_{\max}$	Nyquist bandlimit of a segmented device	29
δ_x	individual pixel pitch	29
N	number of pixels across a segmented device	29
z_i	physical location of a phase screen	30
ν	radial polar component of the three-dimensional linear spatial frequency vector (waves/m)	32
$\boldsymbol{\nu}$	three-dimensional linear spatial frequency vector	32
$\sigma_{\phi_{\text{missing}}}$	missing phase standard deviation (waves)	32
$\%_{\text{present}}$	percentage of present $\sigma_{\phi_{\text{PTR}}}$ threshold level	33
$D_{\phi}(R)$	phase structure function (rad ²)	33
$\sigma_{\phi_{\text{PTR}}}$	total theoretical PTR phase standard deviation (waves)	39
$\sigma_{\phi_i}^2$	total phase variance present in the i^{th} Zernike mode (rad ²)	40
$P(\gamma, \kappa, z)$	function for the type of optical source	40
$\hat{F}(\gamma, \kappa)$	filter function used to represent individual Zernike modes	40
J_{n+1}	Bessel function of the first kind	41
$\mathcal{M}\{f(t); s\}$	Mellin transform	41
η	normalized linear spatial frequency	42
$\%_{\text{missing}}$	percentage of missing $\sigma_{\phi_{\text{PTR}}}$ threshold level	47

List of Abbreviations

Abbreviation		Page
AO	adaptive optics	1
DM	deformable mirror	1
LC	liquid-crystal	2
SLM	spatial light modulator	2
MEMS	micro electro-mechanical systems	2
PSD	power spectral density	7
NLC	nematic liquid crystal	17
FSM	fast steering mirror	37
PTR	piston, tip, and tilt removed	38

LIMITATIONS OF SEGMENTED WAVEFRONT CONTROL DEVICES IN EMULATING OPTICAL TURBULENCE

I. Introduction

This research aims to find the maximum equivalent atmospheric turbulence strength that can be achieved in a laboratory environment with two phase screens. It is revealed that a layered model for atmospheric turbulence strength can be represented by the atmospheric coherence diameter that a single segmented wavefront control device can produce. The limitations of pixelation effects on a segmented wavefront control device are investigated theoretically and in simulation.

1.1 Motivation

Using a device to act as a surrogate for atmospheric turbulence in a laboratory is necessary to build and test optical systems for imaging, lidar, laser weapons, and laser communications. The United States Air Force has several multi-million dollar programs that require optical propagation through the atmosphere. These include the Airborne Laser program, the Advanced Tactical Laser program, and the satellite imaging program at the Starfire Optical Range at Kirtland Air Force Base, New Mexico. Each of these programs require the propagation of light through a turbulent atmosphere. How atmospheric turbulence affects optical propagation must be fully understood before any of these programs can try to complete their respective missions. This is because the controllers need to fully understand those effects so they can be mitigated before any laser is fired.

Atmospheric turbulence effects are mitigated with the use of adaptive optics (AO). Large-scale deformable mirrors (DMs) are usually employed in these systems as real-time AO devices that correct for the effects of the atmosphere. A turbulent wavefront

is one in which the atmosphere has altered the phase of propagating light via randomly evolving distortions of the refractive index. Complex wavefront sensors in AO systems measure the amount of distorted phase added to the light by turbulence. Then DMs are used to correct for that turbulence. DMs compensate by distorting the shape of the mirror so that the conjugate of the turbulent phase is applied to the incident wavefront. When the turbulent wavefront reflects off the DM, this conjugate phase flattens the wavefront. There are many areas of research that enable this complex AO system to operate correctly at real-time speeds. Wavefront sensing, control systems, and wavefront control devices are extensively researched in order to improve this process.

Before a DM is put to work in a complicated system with real turbulence, its individual capabilities must be understood. These capabilities are first investigated in a controlled laboratory environment. Common laboratory conditions are not conducive to testing atmospheric turbulence. Therefore, a suitable turbulence surrogate must be employed. Heating elements can be used to create turbulence in a laboratory, but the turbulence they create cannot easily be controlled or reproduced. A better method of reproducing atmospheric turbulence in a laboratory is necessary to test the abilities of wavefront control devices. This research focuses on the capabilities and limitations of wavefront control devices. Liquid-crystal spatial light modulators (LC SLMs), micro electro-mechanical systems (MEMS) DMs, and static atmospheric phase wheels are small wavefront control devices that are commonly used in laboratories for recreating and correcting optical aberrations. Without first testing a device's abilities, a program is not able to use that device to its full potential. Failure of a device to perform in a complex program could result in failure of the mission and significant monetary loss.

The solution to the problem of emulating turbulence in a laboratory lies in using wavefront control devices to both produce and correct for atmospheric turbulence. If a device is capable of applying the conjugate of the turbulent phase to an incident light source, then it is also capable of applying the turbulent phase to a plane wave

light source. This means that a simple coherent laser source can be altered by a wavefront control device to recreate atmospheric turbulence. Using a wavefront control device results in repeatable and controllable atmospheric turbulence parameters for the distorted wavefront.

Another important way to inexpensively study how the atmosphere affects optical propagation in a program is to use computer simulation. Wavefront control devices in a laboratory are used to verify that these computer simulations are accurate. Without verification through experimentation, a simulation cannot be trusted to act as a surrogate for atmospheric turbulence. Thus, these two techniques are complementary.

1.2 Goals

The primary objective of this research is to identify the best locations for wavefront control devices in a laboratory experiment in order to maximize the overall turbulence they can produce. This research focuses on the locations for two wavefront control devices and a plane wave light source.

To satisfy this goal, this research investigates the limitations in emulating atmospheric turbulence of segmented wavefront control devices such as segmented MEMS DMs and LC SLMs. Another goal of this research is to identify and demonstrate the limits of these devices. This goal is addressed through an investigation of the maximum turbulence strength a segmented wavefront control device can emulate through both theoretical analysis and computer simulation. The maximum turbulence strength is a function of the Nyquist bandlimit of the segmented wavefront control device. This finding leads to an additional goal to identify the relationship between turbulence strength and the spatial frequency of a device.

1.3 Thesis Outline

This thesis meets these goals through a detailed investigation. Chapter II examines the basic theoretical properties of the atmosphere and how phase screens are used to represent atmospheric turbulence. It also discusses how different segmented wavefront control devices alter the phase of incident light and emulate turbulence. Additionally, Chapter II includes a detailed literature review that outlines limitations others have found for segmented wavefront control devices. Chapter III then looks at the laboratory setup of wavefront control devices to determine the best placement of these devices to maximize the range of turbulence conditions they can emulate. Chapter III also presents a theoretical derivation of a numerical solution for the maximum possible amount of turbulence as a function of the number of pixels on a segmented wavefront control device. Chapter IV presents a complementary analysis using computer simulation with the **Matlab**[®] computer program from The MathWorks Inc. Finally, Chapter V gives a comparison between the theoretical and simulation results. This chapter then provides concluding remarks as well as possible extensions of this work in future research.

II. Theory and Review

This work builds on that of previous researchers, moreover, this section reviews what others have done in the field of atmospheric propagation using SLMs. It begins with basic theory concerning atmospheric turbulence and then explores other important and relevant areas that are the foundations of this research.

2.1 *Atmospheric Properties*

Light traveling in a vacuum can easily be modeled using the Fresnel propagation integral. [13] However, when light travels through the Earth's atmosphere, it is affected by the turbulence present. Understanding how that turbulence alters the path of light traveling through it has been the subject of many publications. One of the simplest ways to understand the properties of atmospheric turbulence is to view the refractive index of the atmosphere as a random process. [1] Random variations in the temperature and pressure of the Earth's atmosphere spatially and temporally alter the atmosphere's index of refraction n for a given point $\mathbf{R}$ in space according to

$$n(\mathbf{R}) = 1 + 77.6 \times 10^{-6} (1 + 7.52 \times 10^{-3} \lambda^{-2}) \frac{P(\mathbf{R})}{T(\mathbf{R})}, \quad (2.1)$$

where λ is the optical wavelength in micrometers, P is the pressure in millibars, and T is the temperature in Kelvins. [1] It is impossible to know the refractive index of the atmosphere at all points in space at every moment in time. Therefore, a statistical analysis is needed to best represent the atmosphere. [33]

This analysis is based on utilizing turbulent eddies (i.e., pockets of air with highly correlated indices of refraction). [31] Turbulence can be defined as randomly distributed eddies of varying sizes and temperatures. How light travels through the changes in index of refraction due to turbulent eddies in its path defines how it is affected by atmospheric turbulence. The outer scale L_0 is the average size of the largest eddies. Likewise, the inner scale l_0 is the average size of the smallest eddies. As inertial forces act on the atmosphere, the larger eddies break up into smaller ones and a continuous distribution of eddy sizes can be found between the outer and inner

scales. This range of eddy sizes is known as the inertial subrange. When eddies break down smaller than l_0 , they are considered too small to follow the statistics of turbulent flow and turbulence gives way to laminar flow. [1, 33]

The simplest method of understanding the random fluctuations in index of refraction is to use its structure function $D_n(\mathbf{R})$. This function can be used to describe the mean-square difference in index of refraction between two points separated by a distance $\mathbf{R}$ in space. The structure function is defined by

$$D_n(\mathbf{R}_1, \mathbf{R}_2) = \langle [n(\mathbf{R}_1) - n(\mathbf{R}_2)]^2 \rangle, \quad (2.2)$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are vectors describing points in space and $\langle \cdot \rangle$ is an ensemble average. [1] Assuming that refractive index fluctuations maintain stationary increments further simplifies this equation. If the field is statistically homogenous, the covariance function $B_n(\mathbf{R})$ given by

$$B_n(\mathbf{R}_1, \mathbf{R}_2) = \langle n_1(\mathbf{R}_1)n_1(\mathbf{R}_1 + \mathbf{R}) \rangle, \quad (2.3)$$

becomes independent of spatial position and is only dependent on the distance between the two points of interest: $\mathbf{R} = \mathbf{R}_1 - \mathbf{R}_2$. If the random field is also isotropic, the covariance function becomes dependent only on the scalar distance $R = |\mathbf{R}_1 - \mathbf{R}_2|^2$ between the two points. The structure function is related to the covariance function by

$$D_n(R) = 2[B_n(0) - B_n(R)]. \quad (2.4)$$

When these assumptions are combined with Kolmogorov's dimensional analysis, the structure function is reduced to

$$D_n(R) = C_n^2 R^{2/3} \quad \text{for} \quad l_0 \ll R \ll L_0, \quad (2.5)$$

where C_n^2 is the refractive index structure constant and has units of $\text{m}^{-2/3}$. This parameter defines the strength of the fluctuations in the refractive index and is generally referenced as the strength of the turbulence. [1,31]

The power spectral density (PSD) of the random changes in the atmosphere's index of refraction $\Phi_n(\boldsymbol{\kappa})$ also characterizes the statistical distribution of the size and number of turbulent eddies along the optical path. The Wiener-Khintchine theorem, given by

$$\Phi_n(\kappa) = \int_{-\infty}^{\infty} B_n(R) e^{-j2\pi\kappa R} dR, \quad (2.6)$$

states that the power spectrum is the Fourier transform of the covariance of the refractive index variation from Eq. (2.3). [1] This theorem and Eq. (2.4) can be combined, leading to an expression for the refractive index structure function in terms of the PSD, given by

$$D_n(R) = 8\pi \int_0^{\infty} \kappa^2 \Phi_n(\kappa) \left[1 - \frac{\sin(\kappa R)}{\kappa R} \right] d\kappa, \quad (2.7)$$

where κ is the radial polar component of the three-dimensional angular spatial frequency vector $\boldsymbol{\kappa}$, measured in rad/m. The 2/3 power-law shown in Eq. (2.5) substituted into Eq. (2.7) leads to

$$\Phi_n(\kappa) = 0.033 C_n^2 \kappa^{-11/3} \quad \text{for} \quad 1/L_0 \ll \kappa \ll 1/l_0. \quad (2.8)$$

Eq. (2.8) is more commonly known as the Kolmogorov power-law spectrum and is only valid over the inertial subrange. Other spectra were introduced in order to increase the valid range for the PSD. These include the von Kármán, Tatarskii, and modified spectrum among others. The modified von Kármán spectrum is given by

$$\Phi_n(\kappa) = 0.033 C_n^2 (\kappa^2 + \kappa_0^2)^{-11/6} \exp\left(-\frac{\kappa^2}{\kappa_m^2}\right) \quad \text{for} \quad 0 \leq \kappa < \infty, \quad (2.9)$$

where $\kappa_0 = 2\pi/L_0$ and $\kappa_m = 5.92/l_0$. [1] Four common PSDs are shown as a function of spatial frequency in Fig. 2.1. For simplicity and proof of concept, this research will

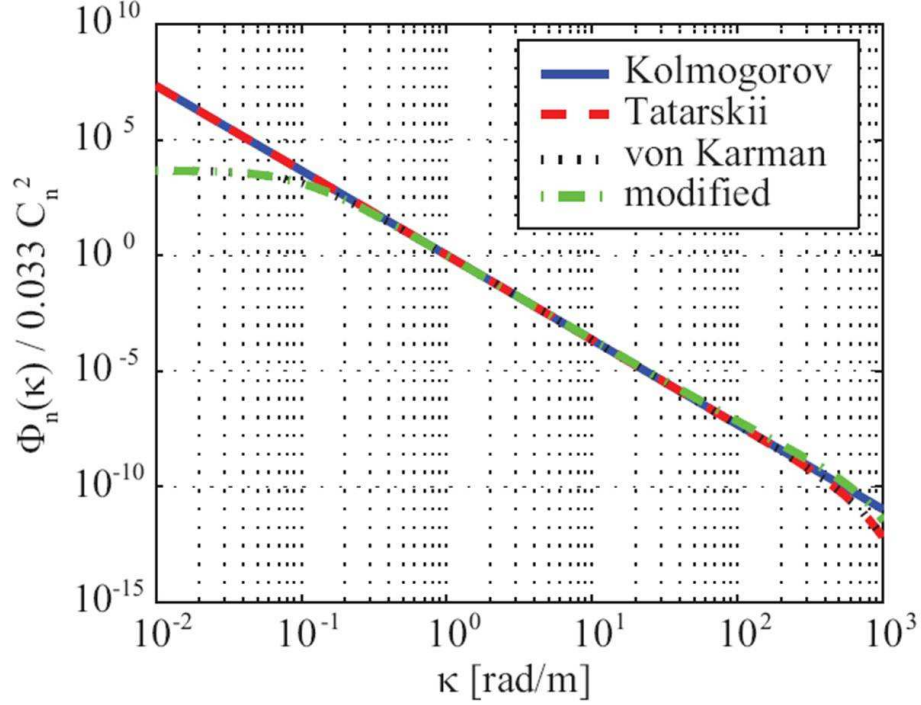


Figure 2.1: Common atmospheric refractive index PSD models given as a function of spatial frequency. This image is reproduced with the consent of Schmidt [33]

use the Kolmogorov power-law spectrum from Eq. (2.8) to represent the statistical properties of the atmosphere.

Other important atmospheric parameters are the Fried parameter r_0 and the Rytov number $\sigma_{\mathcal{R}}^2$ which are defined by

$$r_0 = \left[0.423k^2 \int_0^L C_n^2(z) dz \right]^{-3/5} \quad (2.10)$$

$$\sigma_{\mathcal{R}}^2 = 0.5631k^{7/6} \int_0^L C_n^2(z)(L-z)^{5/6} dz, \quad (2.11)$$

where λ is the optical wavelength and $k = 2\pi/\lambda$. Equations (2.10) and (2.11) are both valid for a plane wave propagating from $z = 0$ to $z = L$, where L is the optical path length measured in meters. [1] The Fried parameter defines the roll off of the volume underneath the optical transfer function and is a measure of the coherence width of the atmosphere. The Rytov number is a measure of the scintillation in light

as it travels through the atmosphere. [1, 12] It is equivalent to log-amplitude variance under weak turbulence conditions. The Fried parameter and Rytov number are direct measures of the phase and amplitude properties of an optical wave.

2.2 *Modeling the Atmosphere*

2.2.1 *Layered Model of Turbulence.*

Modeling the atmosphere would be easy if it were possible to use a simple statistical model. Unfortunately this is not usually an option because the equations are often analytically intractable for the most general theories. This is especially true in the case of modeling complex AO systems. [31] The most common solution to this problem is to break the turbulent optical path into a finite number of discrete layers. Under certain constraints, a layered model is valid for an incident plane wave. [31]

Each layer is a thin phase screen that represents a much thicker volume of atmospheric turbulence. As light travels through a phase screen, it accumulates the turbulence statistics associated with a long propagation through turbulence. The phase screen must be much thinner than the section of the total propagation distance it represents. To propagate a wavefront through a simulated turbulence path using phase screens, the light is first propagated from the source to the first screen as if it were traveling through a vacuum. Next, the complex transmittance function of the first phase screen is multiplied to the incident wavefront. The wavefront continues propagating to the next phase screen as if through a vacuum, and this process repeats until the wavefront reaches the target. This is known as the split-step beam-propagation method and is shown for a two phase screen system in Fig. 2.2. [33]

The split-step beam-propagation approach allows for simulation of a broad and continuous atmosphere in a discrete method. The i^{th} layer in the model can be theoretically defined by the structure parameter $C_{n_i}^2$, the height above the ground z_i , and the thickness Δz_i of the layer of atmospheric turbulence it represents. Values for these parameters are chosen so that the low-order statistical moments m for the

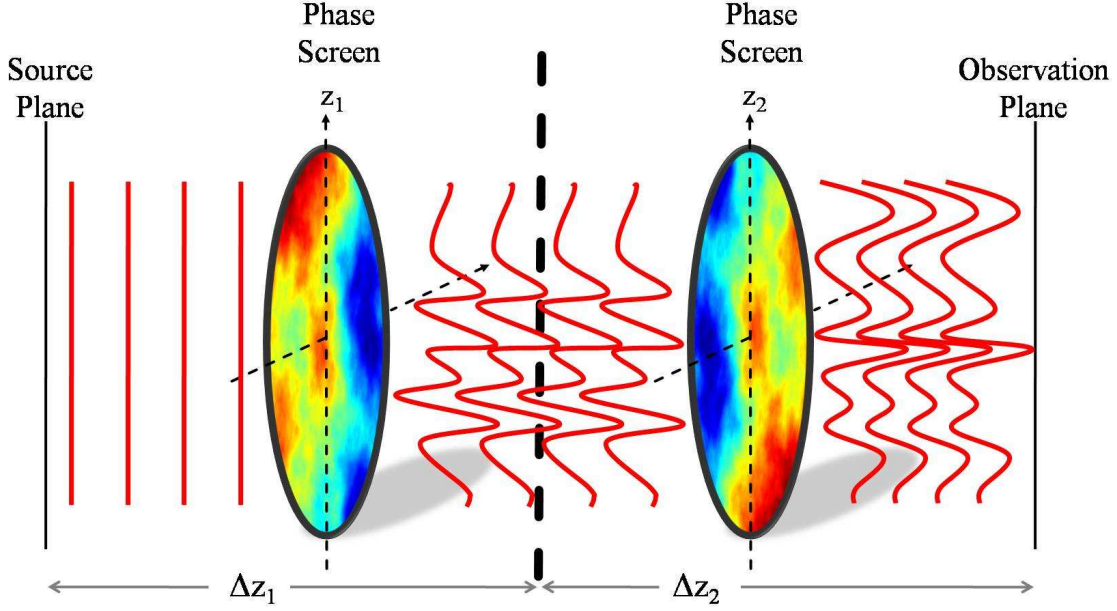


Figure 2.2: Split-step beam propagation method using two phase screens. Plane-wave light is incident from the left and propagates through vacuum to the first phase screen. The light is distorted by the first screen and then propagates through vacuum to the second screen. The second screen further distorts the light and it then travels through vacuum to the observation plane. The resultant wavefront at the observation plane has the same turbulence characteristics as light that has traveled continuously through a turbulent atmosphere.

continuous model match those for the layered model

$$\int_0^L C_n^2(z')(z')^m dz' = \sum_{i=1}^N C_{n_i}^2 z_i^m \Delta z_i, \quad (2.12)$$

where L is the total propagation distance and N is the number of phase screens used to represent L . [31] This allows for important atmospheric parameters to be computed with a discrete sum. [33] The Fried parameter r_0 and Rytov number σ_R^2 are represented continuously by Eqs. (2.10) and (2.11), respectively, and can be computed for a layered turbulence model using

$$r_0 = \left[0.423 k^2 \sum_i C_{n_i}^2 \Delta z_i \right]^{-3/5} \quad (2.13)$$

$$\sigma_R^2 = 0.5631 k^{7/6} \sum_i C_{n_i}^2 (L - z_i)^{5/6} \Delta z_i. \quad (2.14)$$

An effective coherence diameter r_{0_i} can be found for the i^{th} turbulent layer by combining terms in Eq. (2.13) and is given by

$$r_{0_i} = [0.423k^2C_{n_i}^2\Delta z_i]^{-3/5}. \quad (2.15)$$

The coherence diameter for each layer can then be combined to find the overall Fried parameter for the entire length of atmosphere being represented and is given by

$$r_0 = \left(\sum_{i=1}^N r_{0_i}^{-5/3} \right)^{-3/5}. \quad (2.16)$$

The Fried parameter is often used in addition to the structure function to define the strength of the turbulence in a phase screen. [33] Using a similar analysis, the overall Rytov number, given by

$$\sigma_{\mathcal{R}}^2 = 1.331k^{-5/6} \sum_{i=1}^N r_{0_i}^{-5/3} (L - z_i)^{5/6}, \quad (2.17)$$

can also be written in terms of the individual coherence diameters for each layer. As also seen in the continuous model defined in Section 2.1, it should be noted that the above equations are only valid for a propagating plane wave traveling from $z = 0$ to $z = L$. [1]

2.2.2 Phase Screen Generation: Fourier Series Method.

Phase screens are created using a computer generated two-dimensional array of random numbers that have been manipulated into a grid of phase values with the same properties as the atmosphere they represent. The method used in this research to manipulate those random values and create phase screens is the Fourier series approach.

The first step to creating phase screens using the Fourier series method is to choose the PSD for the turbulence desired based on L_0 , l_0 , r_0 , and the linear spatial frequency (f_x, f_y) . [7] The PSD must be sampled at a fast rate in the low frequency

region and less often in the high frequency region in order to adequately sample the spectrum. This gives more weight to the lower order aberrations that are responsible for most of the turbulence. [21]

Next, Fourier series coefficients are found for the spatial frequency region of interest. This is done by randomly drawing Gaussian variables with the appropriate variance given by the PSD. The Fourier series coefficients are denoted by $\mathbf{c}_{n,m}$. Each has circular Gaussian statistics with variance corresponding to the sampled PSD. Each randomly drawn array of the complex coefficients represents a unique random phase. A Fourier series phase $\phi_k(x, y)$ is calculated by

$$\phi_k(x, y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \mathbf{c}_{n,m} e^{j2\pi(f_{xn}x + f_{ym}y)}, \quad (2.18)$$

where (x, y) are the spatial coordinates of the phase screen and (f_{xn}, f_{ym}) are the spatial frequency components. One of the main benefits to this approach is that the phase screen is defined for all points in space. The coefficients have to be determined only once and then using the spatial frequency limits, the phase screen can be synthesized anywhere in space. [21]

2.3 Zernike Polynomials

While any basis (e.g. Fourier series) can be used to represent aberrations in a phase screen, the Zernike series is selected for this research. This is because it is orthogonal over a circular aperture. Random aberrations that result from light traveling through turbulence can be represented by a weighted sum of Zernike polynomials. [31] Zernike modes are two-dimensional polynomials that are represented here as orthonormal on an aperture of diameter D . The Zernike polynomials $Z_{m,n}$ are given in polar coordinates by

$$Z_{m,n}(\rho, \theta)_x = \sqrt{n+1} R_n^m(2\rho/D) \sqrt{2} \cos(m\theta) \quad (2.19)$$

$$Z_{m,n}(\rho, \theta)_y = \sqrt{n+1} R_n^m(2\rho/D) \sqrt{2} \sin(m\theta) \quad (2.20)$$

Table 2.1: First eleven Zernike polynomials and their common names.

Mode (i)	m	n	$Z_{m,n}(\rho, \theta)$	Name
1	0	0	1	piston
2	1	1	$2(2\rho/D) \cos(\theta)$	x tilt
3	1	1	$2(2\rho/D) \sin(\theta)$	y tilt
4	0	2	$\sqrt{3}[2(2\rho/D)^2 - 1]$	defocus
5	2	2	$\sqrt{6}(2\rho/D)^2 \cos(2\theta)$	x primary astigmatism
6	2	2	$\sqrt{6}(2\rho/D)^2 \sin(2\theta)$	y primary astigmatism
7	1	3	$\sqrt{8}[3(2\rho/D)^3 - 2(2\rho/D)] \cos(\theta)$	x primary coma
8	1	3	$\sqrt{8}[3(2\rho/D)^3 - 2(2\rho/D)] \sin(\theta)$	y primary coma
9	3	3	$\sqrt{8}(2\rho/D)^3 \cos(3\theta)$	x trefoil
10	3	3	$\sqrt{8}(2\rho/D)^3 \sin(3\theta)$	y trefoil
11	0	4	$\sqrt{5}[6(2\rho/D)^4 - 6(2\rho/D)^2 + 1]$	primary spherical

for $m \neq 0$, and

$$Z_{0,n}(\rho, \theta) = \sqrt{n+1} R_n^0(2\rho/D) \quad (2.21)$$

for $m = 0$, where ρ is the transverse component of the two-dimensional spatial vector R . The radial functions $R_n^m(2\rho/D)$ used in these equations are defined by

$$R_n^m(2\rho/D) = \sum_{q=0}^{(n-m)/2} \frac{(-1)^q}{q!} \frac{(n-q)!(2\rho/D)^{n-2q}}{[(n+m)/2 - q]! [(n-m)/2 - q]!}, \quad (2.22)$$

where the azimuthal order m and radial order n are non-negative integers that satisfy $m \leq n$ and $n-m = \text{even}$. [31,32] These polynomials can be found for any combination of (m, n) , however, the low-order modes represent the most common aberrations that affect optical systems. Zernike modes are sometimes referenced by their mode index number i . Some common low-order Zernike modes are given in Table 2.1, and a visual example of some common modes present in a turbulent phase screen is given in Fig. 2.3.

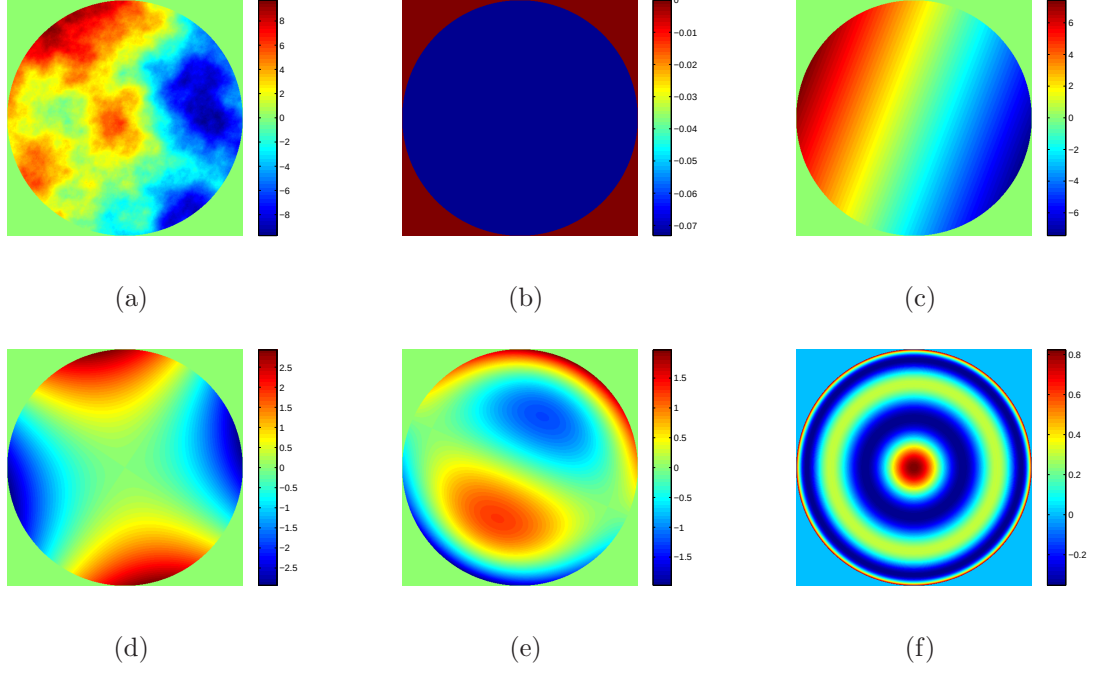


Figure 2.3: Examples of Zernike aberrations present in a turbulent wavefront with the following modes present: (a) all Zernike modes (b) piston (c) X and Y tilt (d) X and Y astigmatism (e) X and Y coma (f) tertiary spherical.

The phase of a wavefront $\phi(D\rho, \theta)$ can be represented as a sum of all the Zernike modes in the wave with

$$\phi(D\rho, \theta) = \sum_{i=1}^N a_i Z_i(\rho, \theta), \quad (2.23)$$

where a_i is the weight associated with the i^{th} Zernike polynomial and N is the number of summed Zernike modes. The expression used to find the weights for individual Zernike modes in cartesian coordinates within a given wavefront $\phi(x, y)$ is

$$a_i = \frac{\iint_{-\infty}^{\infty} W(x, y) \phi(x, y) Z_i(x, y) dx dy}{\iint_{-\infty}^{\infty} W(x, y) Z_i^2(x, y) dx dy} \quad (2.24)$$

where $W(x, y)$ is the aperture function for a circle of unit radius. [31]

Equation (2.23) is verified in Fig. 2.4 which shows that as N increases, the phase of the wavefront approaches the complete turbulent phase with all modes included. Figure 2.4(f) is the turbulent wavefront created by summing the first 37 Zernike

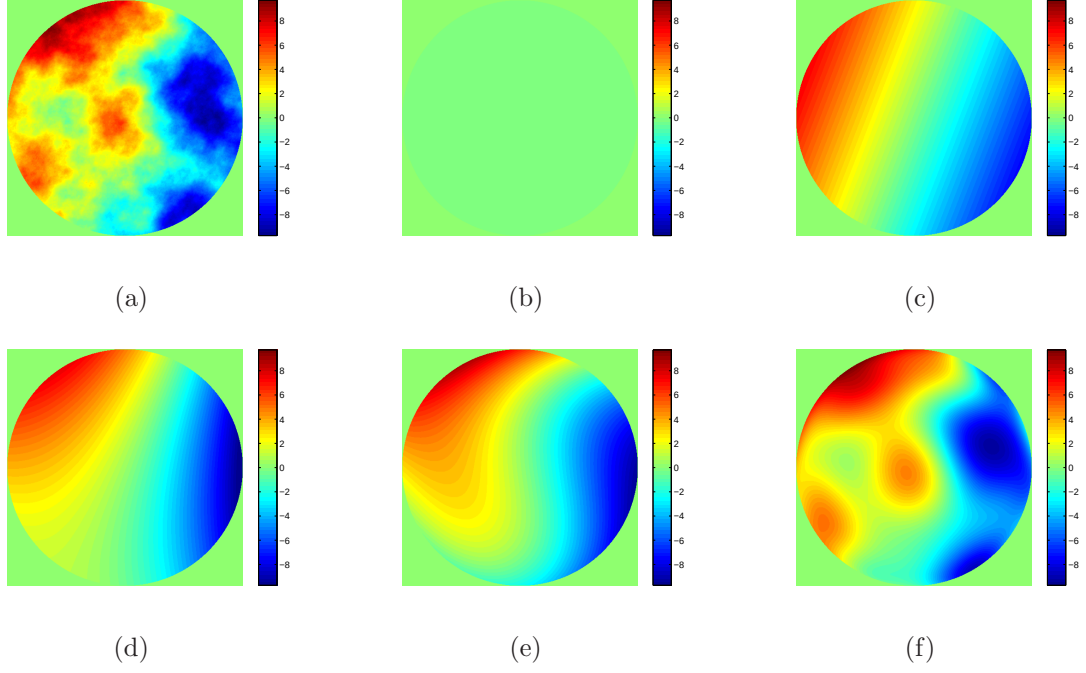


Figure 2.4: Turbulent wavefront phase with the following Zernike modes present: (a) all Zernike modes (b) piston only (c) piston through X and Y tilt (d) piston through X and Y astigmatism (e) piston through X and Y coma (f) piston through tertiary spherical. All subfigures are plotted on the same scale.

modes. It can be seen that Fig. 2.4(f) begins to resemble Fig. 2.4(a) in which all Zernike modes are present. Similarly, Eq. (2.23) can be used to show that subtracting Zernike modes from a wavefront leaves only the residual phase. As that phase approaches to zero, the wavefront loses its turbulence and becomes a plane wave. Figure 2.5 presents an example of this phenomenon. It can be seen that Fig. 2.5(f) is approaching a non-aberrated wavefront because it has lost nearly all of its phase and it resembles an AO compensated wavefront.

To synthesize a phase screen using the Zernike series, the coefficients are computed beginning with random draws from the standard normal probability density function. These are then transformed to have the covariance given by Eqs. (2.25) and (2.26). An expression derived from the Kolmogorov power spectrum for the covariance be-

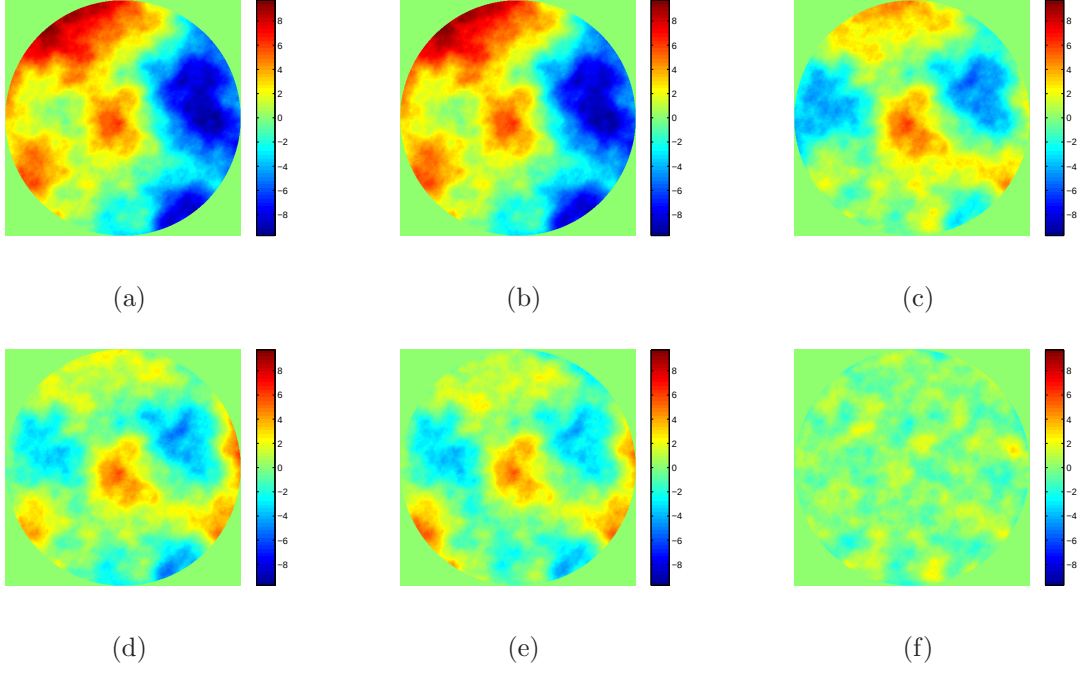


Figure 2.5: Residual phase of a turbulent wavefront with the following Zernike modes removed: (a) none (b) piston only (c) piston through X and Y tilt (d) piston through X and Y astigmatism (e) piston through X and Y coma (f) piston through tertiary spherical. All subfigures are plotted on the same scale.

tween the Zernike coefficients is given by

$$\begin{aligned} \langle a_i a_j \rangle = & 0.0072 \left(\frac{D}{r_0} \right)^{5/3} (-1)^{(n_i+n_j-2m_i)/2} [(n_i+1)(n_j+1)]^{1/2} \pi^{8/3} \delta_{m_i m_j} \\ & \times \frac{\Gamma(14/3) \Gamma[(n_i+n_j-5/3)/2]}{\Gamma[(n_i-n_j+17/3)/2] \Gamma[(n_j-n_i+17/3)/2] \Gamma[(n_i+n_j+23/3)/2]}, \end{aligned} \quad (2.25)$$

for $i - j = \text{even}$, and

$$\langle a_i a_j \rangle = 0 \quad (2.26)$$

for $i - j = \text{odd}$. [25, 31] In Eq. (2.25) $\delta_{m_i m_j}$ is the Kronecker delta function and $\Gamma(x)$ is the Gamma function given by [1]

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt, \quad \text{Re}(x) > 0. \quad (2.27)$$

An approximation for the mean square residual phase error ϵ_i^2 when modes 1 through i are removed, was found by Noll [25] and is given in rad^2 by

$$\epsilon_i^2 \approx 0.2944i^{-\sqrt{3}/2}(D/r_0)^{5/3}. \quad (2.28)$$

Equation (2.28) is only valid for higher order modes ($i > 10$), so in his article, Noll [25] presents the results from a laborious technique performed by Fried to calculate the residual phase found in the lower order modes.

2.4 *Wavefront Control Devices*

Thus far, wavefront control has only been discussed abstractly. In practice, there are two ways that devices alter the phase of a wavefront. They can either alter the index of refraction that the wavefront travels through or change the distance traveled by the light. LC SLMs use the property of birefringence to alter the index of refraction in the SLM. [13] In contrast, DMs use tiny actuators to alter the shape of the wavefront so that the physical distance traveled by light is modified.

2.4.1 *Liquid-crystal Spatial Light Modulators.*

Atmospheric turbulence is characterized by the change in index of refraction of the atmosphere. LC materials have been utilized in order to duplicate this effect and create phase screens. LC devices are commonly used in laboratory environments for adaptive optics because they are nonmechanical, inexpensive, and highly reliable. LC materials share properties of both liquids and solids. These elliptically-shaped molecules that compose LC materials are not bound to each other in a rigid formation, so they are able to rotate or slide when an electrical or mechanical force is applied. LC materials come in a variety of meso-phases; however, the one of interest to this research is the nematic liquid crystal (NLC). The molecules in a NLC tend to align in parallel with their centers randomly distributed throughout the crystal. [13]

The molecules in these devices have positive uniaxial birefringent characteristics that rotate when an electric field is applied. Light traveling through the elliptical molecules of a NLC SLM experiences an altered index of refraction based on the alignment of the molecules. The NLC SLM is composed of several layers. Linearly polarized incident light first passes through a transparent glass cover, then through a transparent electrode, and then into the NLC material. The incident light obtains a phase delay as it travels through the NLC material and then reflects from an array of electrodes before traveling back out along the same path. The double pass through the NLC material causes a phase delay corresponding to the electric field applied to that area of the device and the reflective electrode array subdivides the NLC SLM into an array of independently controlled elements. Each element can impart a different phase delay on the incident wavefront. Large XY arrays of control circuitry behind the NLC SLM allow for hundreds of thousands of controllable phase alterations in current commercial devices. [13,33] Figure 2.6 provides an example of how an applied voltage alters the molecular alignment inside a transmissive NLC SLM. [33]

2.4.1.1 Practical issues for adaptive optics.

The ability of a NLC SLM to impart many controllable phase distortions on a single wavefront makes it useable as a phase screen. In 1998, Dayton et al. [9] were among the first to use a NLC SLM as a turbulent phase screen. That group used a NLC device created by Meadowlark Optics Inc. with an 11 mm active area composed of 127 cells arranged hexagonally. It was found that the device was capable of producing up to 36 Zernike modes and was thus capable of providing good approximations for Kolmogorov phase fronts with $D/r_0 < 10$. These measurements were analyzed using the structure function of the altered wavefront and the error in Zernike mode representation. It was found that spatial quantization limited the order of spatial correction that could be achieved. In addition to Zernike mode representation, Dayton et al. showed the ability of a NLC SLM to compensate time evolving phase screens as an adaptive corrector. One year earlier, Love [20] was able to produce 15 Zernike

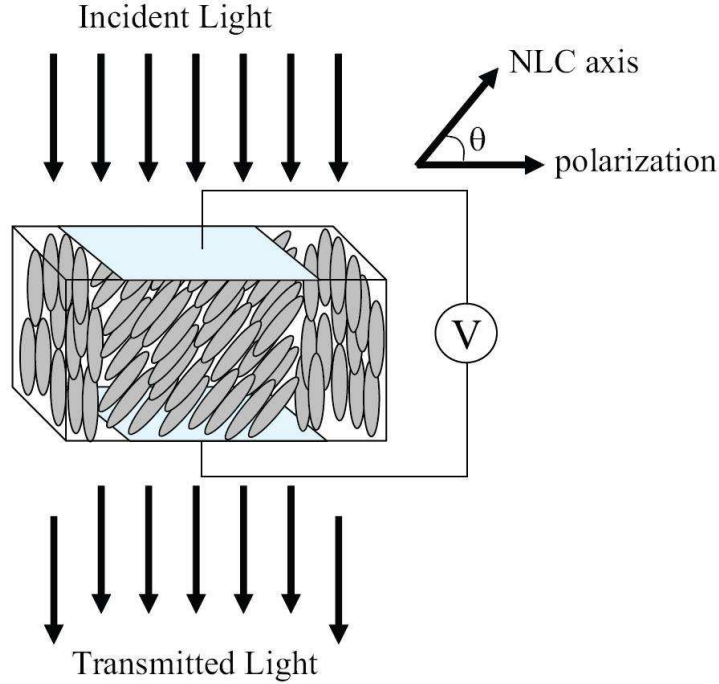


Figure 2.6: Diagram of using an applied voltage to control the birefringence inside a NLC SLM. The molecules rotate with respect to the applied voltage on the device. Incident light travels from the top and passes through the transparent electrode before hitting the rotated NLC molecules. The rotated molecules in the middle of the schematic cause a phase delay in the transmitted light with respect to the edges because the material is uniaxial positive. This image is reproduced with the consent of Schmidt [33]

modes on a 69 pixel device that was also made by Meadowlark Optics Inc. His work proved that quality optical devices could be made using NLC SLMs. It is common for today's NLC SLMs to have at least 512×512 pixels.

Brooks [7] and Phillips [27] further demonstrated the abilities of NLC SLMs. Both researchers used a 512×512 pixel NLC SLM created by Boulder Nonlinear Systems Inc. with a 7.68 mm active area. Brooks was able to demonstrate that atmosphere-like aberrations could be added to an optical system via a NLC SLM. He was one of the first to perform atmospheric simulation using one of these devices. Phillips was able to determine that a system could be created using multiple SLMs to independently control the Fried parameter, isoplanatic angle, and the Rytov variance of the simulated atmosphere. Both researchers demonstrated that experimental values could be scaled appropriately to show good correlation with the theoretical and

simulation values. Brooks created a probability density function of irradiance from all phase screen realizations and compared it to the theoretically expected log-normal distribution to verify the match. He found that even though the error between the experimental and theoretical cases was fairly high, the intensity distributions still had the form and properties of log-normal curves. Phillips also performed an intensity evaluation to verify that the Rytov variance could be adequately modeled using only two phase screens. He found that the system creates higher Rytov variances than expected for weak turbulence and lower variations than expected for the opposite case. The error in the experimental simulation ranged from 2% for nominal turbulence to 75% for very weak turbulence. This was attributed to the SLM being driven to its limits (in terms of applied voltage) for the very strong and very weak turbulence cases, thereby causing significant error. Brooks also performed a phase evaluation to further validate the abilities of the NLC SLM. The phase of the wavefront in the observation plane was compared to the average structure function of each scenario over many realizations with the analytic Kolmogorov structure function. Brooks found experimental phase measurement errors from 15.7-26.6% which were determined to be average quality results. Despite the large error, it was shown that the experimental structure function still adhered to the 5/3 power law. [7, 27]

One of the most practical issues for NLC SLMs in adaptive optics is the problems that can come from not properly calibrating the device. Both Brooks [7] and Phillips [27] mentioned that diffraction from the square apertures of the SLMs may have been a significant contributing factor to the errors seen in the intensity and phase evaluations. However, poor calibration of the LC SLMs was not presented as a large source of error in either paper. In his paper, Schmidt [35] discusses how improper calibrations can lead to significant error.

Schmidt used the same LC SLM as Brooks [7] and Phillips [27]. This time, it was used to show the effects of proper calibration on the wavefront in the observation plane. He showed that a phase-to-command calibration method could be improved upon by introducing a quarter-wave plate into the optical path and performing a

two-step calibration. First, the irradiance was measured without the wave plate, and then it was measured with the wave plate. This extra step preserved the sign of the derivative of the measured phase and allowed for a standard phase unwrapping program to compute the true physical phase written onto the SLM. In addition to improving the method of calibration of the phase loaded onto each pixel of the SLM, this was the first time a theoretical analysis of the amplitude modulation phase-to-command-value calibration was shown. [35]

Schmidt [35] also developed a method of calibrating the SLM for the static aberration inherent in every device due to a slightly warped backplane formed during the manufacturing process. Most SLMs are found to have one to three peak to valley waves of static aberration. The static aberration in the SLM described in Schmidt’s [35] paper was measured using a Twyman-Green interferometer and compensated by commanding a nominal phase map onto the device. The true benefits of having a properly calibrated device were then demonstrated in the paper. [35] In another article, Hart [16] even showed that static aberrations on SLMs can be characterized and removed using simple and widely available hardware without requiring expensive aberration sensing equipment. Any experiment using a SLM should ensure proper calibration before analyzing any results.

2.4.2 Noted Limitations.

Several limitations exist in the use of NLC SLMs as adaptive optics. The NLC material itself is wavelength dependent. Therefore, different incoming wavefronts will see different indices of refraction and optical path lengths when they hit the device. NLC SLMs are manufactured to operate within specific wavelength regions. These devices also only operate correctly with linearly polarized light. Bold et al. [6] address this issue with a discussion of how a SLM can be coupled with a 1/4-wave plate to provide uniform modulation of unpolarized light. SLMs also suffer a decrease in signal throughput and an increase in diffraction artifacts as a result of the opaque transistors

surrounding each pixel. [3] However, these effects can be compensated to some degree through proper calibration of the device. [16]

The most prominent issue that plagues NLC SLMs is the relaxation time of the material. When a phase is electronically commanded to the device, it can respond extremely fast. However, the molecules in the NLC material take comparatively much longer to relax back to their original state after that command is removed. Current devices have response times on the order of several milliseconds. Serati et al. [36] present theory proving the possibility of sub-millisecond response times for NLC SLMs. Although close, manufacturers still have not been able to create devices that are capable of such frame rates. There is much work in high-speed LC materials in which dual-frequency devices have been moderately successful. [8, 24, 29, 36] This limitation constricts SLMs to a laboratory environment and prevents them from being used in real-time high speed AO applications.

Litvin et al. [19] and Bagwell et al. [3] also discuss how digitization is another limiting constraint of SLM devices. Most current SLMs have eight bits of phase per 2π of phase. The desired analog phase value for a pixel must be converted to a number from 0 to 255 in order to be electronically commanded to the device. The polarity independence of NLC molecular rotation further limits this range to seven bits or 128 programmable levels. This restriction exceeds the spatial resolution of the pixels on most devices and leads to sampling as the limiting constraint when using NLC SLMs. [3, 19]

2.4.2.1 Other uses.

Liquid-crystal SLMs can be used in a vast array of other applications. They can be used for AO to both simulate atmospheric turbulence and correct for it. It has also been shown that SLMs can be used to steer a laser beam. Recently, Linnenberger et al. [18] showed that LC devices can be used to steer a laser beam up to ± 6.95 degrees. Hällstig et al. [14, 15] not only discusses how using LC SLMs to alter the phase of a wavefront can induce a deflection that acts to steer a laser beam, but goes on to show

that a SLM can be used to alter the shape of the beam itself. Schmidt [34] expands on beam shaping through the use of a phase retrieval-based wavefront control to shape a laser beam into a turbulence-distorted beam. This effectively eliminates the need for multiple phase screens in a layered turbulence model and allows LC SLMs to model bulk atmospheric turbulence without moving parts. Bagwell et al. [3] use SLMs to act as lenses and even demonstrate nonmechanical zoom and enhanced multi-spectral imaging systems using liquid-crystal devices.

2.4.3 Deformable Mirrors.

The DM is another widely used device to correct for aberrations in a wavefront. As stated previously, DMs use actuators located behind the mirror surface to alter the shape of the wavefront so that the physical distance traveled by light is modified. Changing the length of the optical path creates a phase delay at the observation plane. The DM acts as a phase screen capable of commanding many phase distortions on a single wavefront when the light is spread over many actuators. DM actuators are capable of providing several micrometers of stroke with nanometer precision. [4] DMs are the preferred device used in real-time astronomical applications due to their fast operating speed. [11] However, such DMs are often too expensive to use in a laboratory environment. Instead, a smaller and more cost efficient device was needed for use in research applications. MEMS DMs provide a solution by combining technology from the fields of micro-instruments, adaptive optics, and controls to form a unique mirror assembly. [23] These mirrors have the advantages of being simple to manufacture, inexpensive, lightweight, and integrable with drive and sensing electronics. [23] They also consume very little power, are capable of being driven at KHz frame rates, and, unlike SLMs, most DMs are wavelength-independent due to the reflective properties of the mirror surface. [5, 11] The three main types of MEMS DMs in use today are continuous mirror, segmented mirror with piston motion only, and segmented mirror with tip/tilt motion DMs. [5] MEMS DMs commonly have 5×5 to 12×12 pixels/mirror

segments and the more expensive models have as many as 32×32 pixels/mirror segments.

2.4.3.1 Continuous DMs.

Continuous facesheet membrane DMs are created by stretching a thin mirror membrane over a circular silicon frame. The shape of the membrane is then altered through the use of electrostatic forces. The first continuous membrane MEMS DM had nine actuators and was designed, manufactured, and tested in the end of the 20th century at Boston University. [4]. Another method of creating a continuous facesheet DM is to use a thin glass faceplate that is controlled with piezoelectric actuators. Xinetics Inc. is one company that manufactures this type of device.

A continuous surface offers several advantages over other mirror types. The surfaces of both the membrane and thin glass continuous facesheet DMs ensure smooth and continuous phase variations over the entire mirror. A smooth surface causes almost no diffraction in the reflected beam. Also, there is no loss of optical intensity due to fill factors present in an array like the NLC SLM. These advantages do not come without drawbacks to the continuous DM. For example, the continuous surface causes interactuator coupling that complicates adaptive optic control between the actuators. This coupling is known as the influence function and is a compromise of the system. An influence function of approximately ten percent is typical between adjacent actuators. [5] This influence function causes a tension or compression on the mirror surface that can cause adverse affects in MEMS devices.

Fernández and Artal [11] were able to use a 37 element continuous MEMS DM manufactured by OKO Technologies as an adaptive optical corrector. They successfully reproduced the first 21 Zernike modes with good accuracy using the mirror. Unfortunately, the limited range of control voltages allowed some of the higher order modes to only be producible over a very small range. Despite this limitation, it was a successful demonstration of adaptive wavefront control using a continuous facesheet MEMS DM. [11] Doble and Williams [10] are working to apply continuous

MEMS DMs in the field of optometry. They discuss how glasses and contact lenses are static devices that can only correct low-order aberrations such as defocus and astigmatism. Doble and Williams present the idea that continuous MEMS devices can be used as adaptive optics for real-time complete correction of human vision. In order to make this concept a reality, it is necessary to continue to improve the MEMS devices that are commercially available. The challenge lies in manufacturing a compact and reliable DM with large stroke and actuator count at a low cost. [10]

2.4.3.2 Segmented DMs.

Another type of MEMS DM is the segmented DM. This type of mirror comes in two different versions. The first type is the segmented MEMS DM with piston-only motion. This DM is created with tiny mirror segments at the end of each actuator which are completely independent of each other. This mirror has no interactor coupling between the mirrors, however undesirable diffraction effects are introduced due to the gaps between adjacent mirror segments. Also, this mirror does not allow for spatially continuous phase modulation across the surface of the mirror. [5] The benefit to this mirror is that it allows for relief of the interactor stress prominent in continuous facesheet DMs. Furthermore, it also provides the most mirror segments over any device due to the one-to-one ratio of segments to actuators.

The second type of segmented MEMS DM is a hybrid between the previous alternatives. A segmented DM with tip/tilt motion has mirror segments over three actuators. Sometimes these actuators are shared between segments. In either case, this device allows for the optical phase to be closely matched at the interface between adjacent segments. [5] There are some diffraction and interactor coupling issues associated with this device, but each is an acceptable compromise between the other two types of MEMS DMs. Perreault et al. [26] were able to use segmented DMs with tip/tilt motion to achieve phase modulation in an AO system. Their research showed significant reduction in the wavefront phase error after correction. The Strehl ratio of the wavefront increased from 0.0034 to 0.1950 when the aberrated wavefront

was compensated. [26] This Strehl ratio is a performance metric used in imaging. Strehl ratios close to one represent a near-diffraction limited beam with almost no adverse turbulence affects. The segmented mirrors relieved problems caused by stress on a continuous membrane and still maintained a fill factor of 98.6%. This type of mirror also maintained phase continuity between segments through actuator sharing of adjacent mirror segments. [26]

Lee et al. [17] were able to show correction near the diffraction limit for spherical aberration using a segmented MEMS DM. The group corrected an aberrated wavefront with a 0.7 meter radius of curvature within two percent of the predicted performance. They were able to achieve these results by coupling the segmented DM with a refractive lenslet array. The lenslet improved the effective fill factor of the DM by focusing the incident light onto the centers of the mirror segments. A DM with 128 individual elements on a 12×12 square grid was used for this research. The combination of the lenslet and the segmented DM behaved as a phase-only modulator for mirror deflections much smaller than the focal length of the lenslet. [17]

The side effects associated with the type of MEMS DM used in an experiment must be considered during research. The number of actuators, the motion of resolution of each mirror segment, the control bandwidth of each actuator, and the actuator stroke are the most important variables that must be understood when using MEMS DMs in adaptive optics applications. [5]

2.4.4 Alternative Methods.

LC SLMs and DMs are not the only way to alter a wavefront and simulate atmospheric turbulence. Probably the most common method is a rotating phase wheel atmospheric turbulence simulator. This device uses two simple static phase plates to act as turbulence layers. The phase plates are created based on the concept of near-index matching. When they are rotated, the light experiences a randomly evolving index of refraction. Most of the important atmospheric parameters can be manipulated by changing the relative positions and rotation speeds of the phase plates

when two static phase plates are put together in the same optical system. [30] This is not an ideal method of turbulence creation because it takes considerably more time to make simple changes to the atmosphere than the other methods discussed. [22] As evidenced in any educational AO laboratory, obtaining desired values for the atmospheric turbulence parameters requires tedious calculations and movement of the phase wheels. Phase wheels are often repositioned by hand which is not as precise as an electronically addressed wavefront control device such as a SLM or DM. SLMs and DMs are much more reproducible because they are electronically addressed. SLMs also benefit from being nonmechanical in that they do not have any moving parts that require maintenance. Phase wheel systems can also be cumbersome and usually require several meters of space on an optical bench, whereas SLMs and MEMS DMs are compact and require only a few centimeters of space in a laboratory.

III. Theoretical Analysis

This chapter conveys the theoretical approach and methodology used to address the main ideas of this research. First, a theoretical understanding of setting up two phase screens for atmospheric simulation is presented and discussed. Then, a detailed analysis of the minimum atmospheric coherence diameter as a function of spatial frequency is presented and discussed.

3.1 Phase Screen Locations

This section investigates the strongest possible optical turbulence that can be emulated by two wavefront control devices in a laboratory. Previous work by Phillips et al. [27] concluded that a broad range of turbulence parameters could be represented using only two LC SLMs as phase screens. Two phase screens are commonly used in a laboratory environment due to cost limitations and their ability to represent a broad range of atmospheric conditions. This research quantifies the maximum strength of turbulence two phase screens are capable of representing.

The expression for atmospheric coherence diameter is given in a discrete form by Eq. (2.16). When this equation is expanded for two phase screens it becomes

$$r_0 = \left(r_{0_1}^{-5/3} + r_{0_2}^{-5/3} \right)^{-3/5}. \quad (3.1)$$

It is only dependent on the atmospheric coherence diameters of the individual phase screens. The value r_0 represents the distance between two coherent points in the atmosphere. Therefore, turbulence is at a maximum strength when that distance is small. In the equation above, r_{0_1} and r_{0_2} are independent of each other and thus, the minimum possible overall atmospheric coherence length occurs when both phase screen r_0 values are at a minimum. Assuming both phase screens are created with the same type of device, this limit is the same for each screen. Under these constraints, Eq. (3.1) simplifies to

$$r_0 = 2^{-3/5} r_{0_{\min}}(F_{\max}), \quad (3.2)$$

where $F_{\max}$ is the Nyquist bandlimit of the pixelated wavefront control device used as a phase screen, and $r_{0_{\min}}$ is the minimum possible value for atmospheric coherence length that can be represented with each phase screen. A segmented wavefront control device has a Nyquist bandlimit corresponding to

$$F_{\max} = \frac{1}{2\delta_x}, \quad (3.3)$$

where δ_x is the width of each individual pixel. [13] The number of pixels N across a device can be found with

$$N = \frac{D}{\delta_x} = 2DF_{\max}, \quad (3.4)$$

where D is the diameter of the active area of the device. Using these equations, it is easy to see that $r_{0_{\min}}$ is both a function of the spatial frequency of the device as well as the number of pixels across the device.

The other atmospheric parameter of interest is the Rytov number, which is also known as the log-amplitude variance under weak turbulence conditions. The equation for this value was given for a layered turbulence model by Eq. (2.17) and can be written for two phase screens by

$$\sigma_{\mathcal{R}}^2 = 1.331k^{-5/6} \left[r_{0_1}^{-5/3} (L - z_1)^{5/6} + r_{0_2}^{-5/3} (L - z_2)^{5/6} \right]. \quad (3.5)$$

Using the same rationale as before, the individual coherence diameters both become $r_{0_{\min}}$ and the expression above reduces to

$$\sigma_{\mathcal{R}_{\max}}^2 = 1.331k^{-5/6} r_{0_{\min}}^{-5/3} L \left[\left(1 - \frac{z_1}{L}\right)^{5/6} + \left(1 - \frac{z_2}{L}\right)^{5/6} \right], \quad (3.6)$$

for the maximum possible Rytov number. Creating linear arrays of values for the variables for z_1/L and z_2/L leads to a contour plot for

$$\sigma_{\mathcal{R}_{\max}}^2 \left(\frac{r_{0_{\min}}^{5/3}}{L\lambda^{5/6}} \right) = 1.331 (2\pi)^{-5/6} \left[\left(1 - \frac{z_1}{L}\right)^{5/6} + \left(1 - \frac{z_2}{L}\right)^{5/6} \right] \quad (3.7)$$

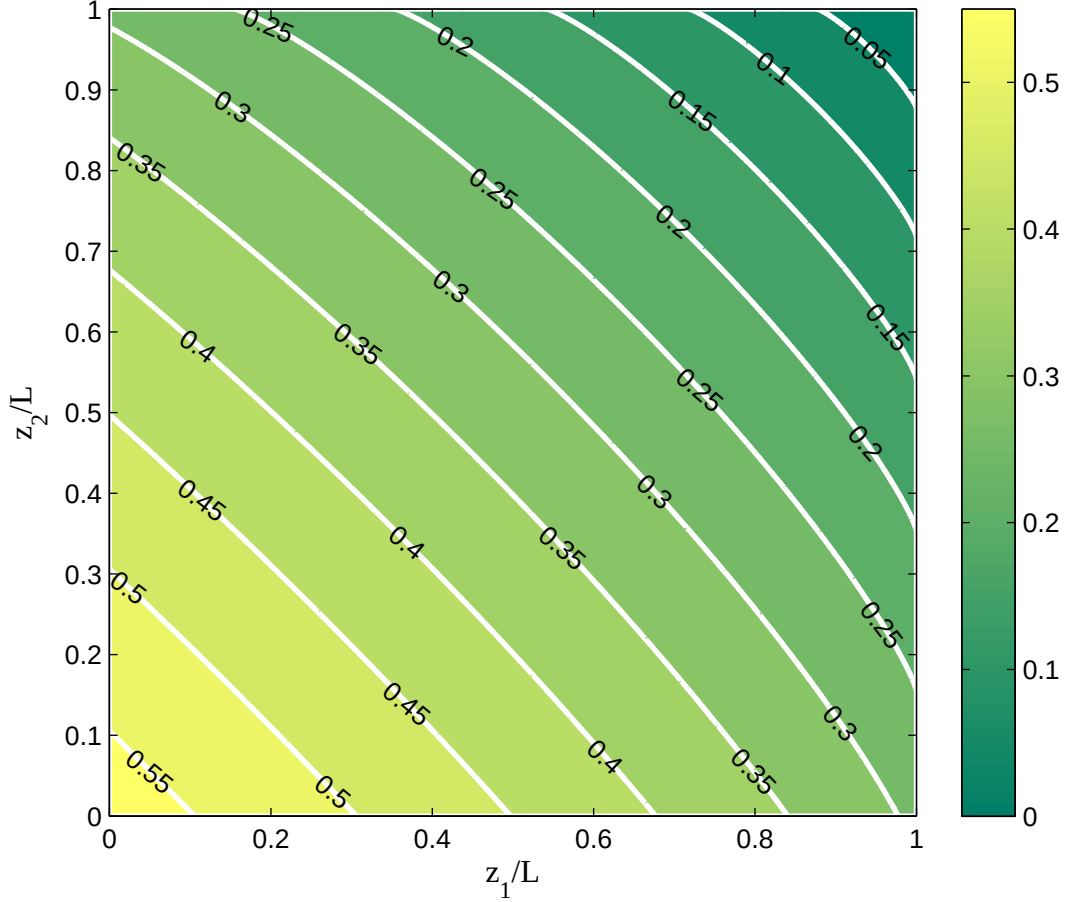


Figure 3.1: Maximum normalized Rytov number $\sigma_{\mathcal{R}_{\max}}^2 \left(\frac{r_{0_{\min}}^{5/3}}{L\lambda^{5/6}} \right)$ given as a function of different phase screen locations for a two-screen setup.

which is shown in Fig. 3.1. This equation shows that the maximum Rytov number is a function of $r_{0_{\min}}$, total propagation distance, wavelength, and also the location of the phase screens for a two-screen setup. Most importantly, Fig. 3.1 shows that the maximum value for the Rytov number can be found when both phase screens are as close as possible to the beginning of the propagation distance. This is possible because Eq. (3.6) does not depend on the total length of atmosphere each phase screen represents Δz_i , but only on the physical position of each phase screen z_i . Even though the theoretical setup for the most turbulent atmosphere requires both phase screens to be placed at the very beginning of the optical path, the physical size of each wavefront control device limits their placement. This physical limitation may be

insurmountable to achieve the maximum possible turbulence, but Fig. 3.1 provides an estimate for the maximum Rytov number obtainable given any laboratory setup for a two phase screen system.

3.2 *Minimum Atmospheric Coherence Length*

The previous discussion on phase screen locations has revealed that the minimum possible overall value for r_0 and the maximum possible overall value for $\sigma_{\mathcal{R}}^2$ are functions of $r_{0_{\min}}$. In order to truly identify the maximum turbulence that can be created in a laboratory with two phase screens, it is essential to look into the absolute minimum value for the atmospheric coherence length of a phase screen. As previously stated, this value is a function of the Nyquist bandlimit of the segmented wavefront control device being used. Equation (3.4) shows there is a direct relationship between the Nyquist bandlimit and the number of pixels across a device. Therefore, an investigation is required to determine the relationship for $r_{0_{\min}}$ as a function of the number of pixels across a device.

The initial method used for finding the minimum atmospheric coherence diameter in terms of spatial frequency is to begin with the refractive index PSD, convert it to overall phase variance, and then solve that for r_0 . The first PSD investigated is the Tatarskii spectrum as given by

$$\Phi_n(\kappa) = 0.033C_n^2\kappa^{-11/3}\exp\left(-\frac{\kappa^2}{\kappa_m^2}\right) \quad \text{for} \quad 0 \leq \kappa \ll 1/l_0, \quad (3.8)$$

where $\kappa_m = 5.92/l_0$. [1] This spectrum is used because it includes the inner scale l_0 which limits the PSD at higher spatial frequencies. When written in terms of atmospheric coherence diameter, Noll [25] shows how this equation reduces to

$$\Phi(\nu) = 0.023r_0^{-5/3}\nu^{-11/3}\exp\left[-\left(\frac{l_0}{0.942}\right)^2\nu^2\right] \quad \text{for} \quad 0 \leq \nu \ll 1/l_0, \quad (3.9)$$

where ν is the radial polar component of the three-dimensional linear spatial frequency vector $\boldsymbol{\nu}$ and is measured in waves/meter. The phase PSD is not the most convenient parameter to evaluate so the overall phase variance associated with the wavefront is used instead. Phase variance can be found by integrating the PSD and is given by

$$\sigma_{\phi_{\text{total}}}^2 = 2\pi \int_0^\infty \Phi(\nu) \nu d\nu. \quad (3.10)$$

This equation gives the phase variance (measured in rad^2) for all spatial frequencies from zero to infinity. This research, however, is interested in identifying how limitations in spatial frequency effect the range of turbulence parameters. In order to represent this constraint, Eq. (3.10) can only be evaluated from F_{max} to infinity. Constraining the limits of integration allows one to find only the phase variance not simulated by the segmented wavefront control device. The overall range of representable phase variance by the device can be found with

$$\sigma_{\phi_{\text{device}}}^2 = 2\pi \int_0^{F_{\text{max}}} \Phi(\nu) \nu d\nu \quad (3.11)$$

$$= 2\pi \int_0^\infty \Phi(\nu) \nu d\nu - 2\pi \int_{F_{\text{max}}}^\infty \Phi(\nu) \nu d\nu \quad (3.12)$$

$$= \sigma_{\phi_{\text{total}}}^2 - \sigma_{\phi_{\text{missing}}}^2. \quad (3.13)$$

In order to evaluate r_0 in realistic terms and remove extraneous variables, a level of fidelity must be established. The level of fidelity can best be created by setting an error threshold of missing phase variance that is acceptable as a result of using a segmented wavefront control device. This error threshold is a level of overall missing phase standard deviation $\sigma_{\phi_{\text{missing}}}$ (measured in waves) that is acceptable to the researcher and is represented by

$$\sigma_{\phi_{\text{error}}} = \frac{1}{2\pi} \sqrt{2\pi \int_{F_{\text{max}}}^\infty \Phi(\nu) \nu d\nu}. \quad (3.14)$$

When measured in percentage of phase standard deviation missing, this level is $1 - (\%_{\text{present}})(\sigma_{\phi_{\text{total}}})$, where $\%_{\text{present}}$ is the threshold level determined by the percentage of the overall phase standard deviation that the wavefront control device must be capable of emulating. This research investigates the threshold of percent phase standard deviation present at the 83%, 90%, 95%, 96%, 97%, 98%, and 99% levels. This range should cover the needs of most researchers while providing a detailed outlook on how the varying requirements of researchers alter the numbers and types of devices that will satisfy those needs. The terms “phase error” and “missing phase” will be used interchangeably throughout this investigation.

3.2.1 Unit Analysis.

Now that a threshold has been established, it is instructive to perform a unit analysis of the equations that are to be manipulated. First, it will be verified that Eq. (3.10) actually yields phase variance in rad^2 . This analysis begins with the phase structure function $D_\phi(R)$ which is given by Noll [25] to be

$$D_\phi(R) = 6.88 \left(\frac{R}{r_0} \right)^{5/3}, \quad (3.15)$$

and is measured in rad^2 . This function is related to the Wiener-Khintchine theorem given in Eq. (2.6) by [25]

$$D_\phi(R) = 2 \iint_{-\infty}^{\infty} \Phi(\nu) [1 - \cos(2\pi\nu \cdot R)] d^2\nu. \quad (3.16)$$

This equation can be used to determine the units of the phase PSD $\Phi(\nu)$. When represented as units, Eq. (3.16) becomes

$$\text{rad}^2 = \text{Units}(\Phi(\nu)) [1 - \cos(\text{m/m})] \left(\frac{\text{waves}}{\text{m}} \right)^2,$$

where the units in the cosine divide out. This results in the units of the PSD given by

$$\text{Units}(\Phi(\nu)) = \text{rad}^2 \left(\frac{\text{m}}{\text{waves}} \right)^2.$$

A simple substitution of this expression into Eq. (3.10) yields

$$\begin{aligned} \sigma_{\phi_{\text{total}}}^2 &= 2\pi \int_0^\infty \Phi(\nu) \nu d\nu \\ \text{Units}(\sigma_{\phi_{\text{total}}}^2) &= 2\pi \int_0^\infty \left[\text{rad}^2 \left(\frac{\text{m}}{\text{waves}} \right)^2 \right] \left(\frac{\text{waves}}{\text{m}} \right) \left(\frac{\text{waves}}{\text{m}} \right) \\ &= \text{rad}^2. \end{aligned}$$

Thus confirming that Eq. (3.10) yields phase variance in rad^2 .

In order to compare the phase variance to the threshold value for phase standard deviation, these units are converted to waves. The conversion from units of rad^2 to waves is given by

$$\text{waves} = \frac{\sqrt{\text{rad}^2}}{2\pi}.$$

The units of these equations must be carefully tracked so there is no confusion in the final results. For this research, phase variance is given in rad^2 and phase standard deviation is always measured in waves.

3.2.2 Evaluation Over a Single Point in Space.

With the unit verification complete, it is possible to move forward and evaluate the atmospheric coherence diameter as a function of the number of pixels across a wavefront control device. Equations (3.9) and (3.14) can be combined to form

$$\sigma_{\phi_{\text{error}}}^2 (2\pi)^2 = 2\pi \int_{F_{\text{max}}}^\infty 0.023 r_0^{-5/3} \nu^{-11/3} \exp \left[- \left(\frac{l_0}{0.942} \right)^2 \nu^2 \right] \nu d\nu. \quad (3.17)$$

A simple change of variables must be performed to reduce this equation to

$$4\pi \sigma_{\phi_{\text{error}}}^2 = 0.023 r_0^{-5/3} \left(\frac{l_0}{0.942} \right)^{5/3} \int_{f(F_{\text{max}})}^\infty f^{-11/6} e^{-f} df, \quad (3.18)$$

where $f = (l_0/0.942)^2 \nu^2$ and $f(F_{\max}) = (l_0/0.942)^2 F_{\max}^2$. The integral portion of this equation can be recognized as an upper incomplete gamma function which further reduces the equation to

$$4\pi\sigma_{\phi_{\text{error}}}^2 = 0.023r_0^{-5/3} \left(\frac{l_0}{0.942}\right)^{5/3} \Gamma\left(-\frac{5}{6}, f(F_{\max})\right). \quad (3.19)$$

An incomplete gamma function cannot be evaluated with a negative value in its argument, so this was transformed using a simple identity from Arfken and Weber [2] given by

$$\Gamma(a-1, x) = a\Gamma(a, x) + x^a e^{-x}. \quad (3.20)$$

Using this identity to evaluate Eq. (3.19) leads to

$$4\pi\sigma_{\phi_{\text{error}}}^2 = r_0^{-5/3} \left[0.023 \left(\frac{l_0}{0.942}\right)^{5/3} \right] \left\{ \frac{1}{6} \Gamma\left(\frac{1}{6}, f(F_{\max})\right) + [f(F_{\max})]^{1/6} \exp[-f(F_{\max})] \right\} \quad (3.21)$$

When the expression for $f(F_{\max})$ is substituted back into this equation and it is solved for r_0 , it becomes

$$r_{0_{\min}} = \left(\frac{4\pi\sigma_{\phi_{\text{error}}}^2}{0.023 \left(\frac{l_0}{0.942}\right)^{5/3} \left\{ \frac{1}{6} \Gamma\left(\frac{1}{6}, \left(\frac{l_0}{0.942}\right)^2 F_{\max}^2\right) + \left(\frac{l_0}{0.942}\right)^{1/3} F_{\max}^{1/3} \exp\left[-\left(\frac{l_0}{0.942}\right)^2 F_{\max}^2\right] \right\}} \right)^{-3/5}. \quad (3.22)$$

This equation provides the sought after expression for the minimum atmospheric coherence length in terms of the Nyquist bandlimit of a segmented wavefront control device. It can be manipulated to express $r_{0_{\min}}$ as a function of the number of pixels across a device using the relationship between N and the Nyquist bandlimit given by Eq. (3.4). When realistic values are chosen for the phase variance error, inner scale, and overall aperture diameter, Eq. (3.22) can be plotted as shown in Fig. 3.2. This figure shows that Eq. (3.22) provides the minimum possible atmospheric coherence diameter as the Nyquist bandlimit of a device increases.

Unfortunately there is no obvious practical application of Eq. (3.22) at this

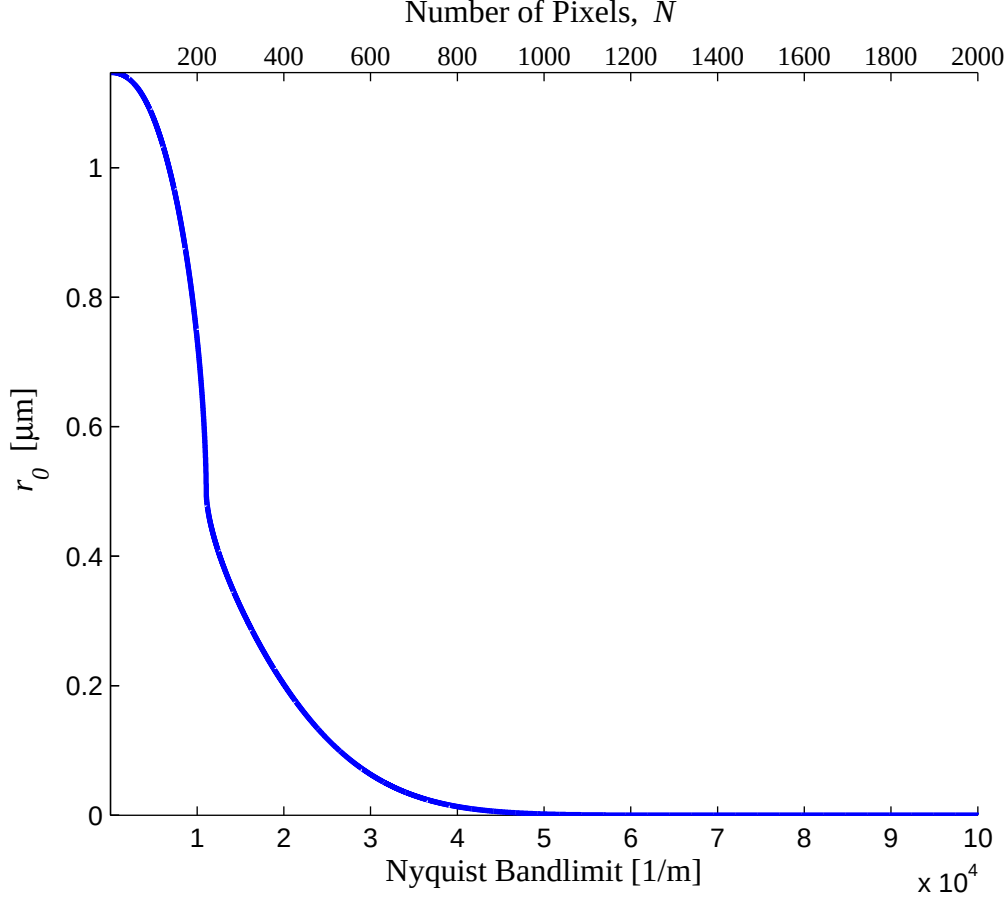


Figure 3.2: Minimum possible r_0 value as a function of the Nyquist bandlimit of a wavefront control device for $D = 10$ mm, $l_0 = 45\mu m$, and $\sigma_{\phi_{\text{error}}} = 1/10$ waves.

point in time. The amount of phase standard deviation missing as a result of using a pixelated device is always given as the total phase standard deviation error over the entire active aperture of the device. Regrettably, Eq. (3.22) only gives the relationship for a single point on that aperture. In order to make the value obtained from this equation useful, a complicated correlation matrix would have to be created for the phase at every point on the aperture. Currently, methods do not allow the creation of such an autocorrelation matrix. However, future research is anticipated to provide the necessary matrix. The analytic expression in Eq. (3.22) cannot be verified until future research comes to fruition. Therefore, another method must be found for the relationship between $r_{0_{\text{imin}}}$ and F_{max} which is investigated in Section 3.2.3.

3.2.3 Evaluation Over an Aperture.

The search for a method to evaluate the total phase variance over an aperture of diameter D ends with the use of Zernike modes. As stated in Section 2.3, a Zernike polynomial can be written as a function of the active aperture of the wavefront control device. Also, the total phase of a wavefront can be represented as a weighted sum of Zernike modes. Similarly, the total phase variance of a wavefront over an aperture can be represented as a sum of the phase variance associated with each Zernike mode over the aperture, as demonstrated in Eq. (2.23). [32]

The overall missing phase variance of a segmented wavefront control device stems from its limitations on two fronts. First, the Nyquist bandlimit of the device limits the overall phase it can represent. Second, in order to analytically represent missing phase variance as a function of spatial frequency, the number of Zernike polynomials that can be added to a device must also be limited. Zernike modes can reproduce true turbulence when they are summed from the first mode (piston) to infinity. [25] However, the best way to reproduce turbulence as a function of spatial frequency is to write it as a finite sum of Zernike modes. The modes excluded from this sum represent the part of the total missing phase variance from finite modes. An analytical expression for $r_{0_{\min}}$ as a function of Nyquist bandlimit can not be derived for the Zernike mode approach. A numerical approach works, but only for a finite number of Zernike modes. This research only focuses on the higher order modes that a device can emulate. Piston, tip, and tilt can all be easily represented in turbulence with the use of a fast steering mirror (FSM). Tip and tilt represent approximately 87% of the total phase variance in a turbulent path. [32] This causes much of the throw of the actuators on a complex device such as a MEMS DM to be wasted on producing the first three modes and limits the total range available to represent the higher order modes where i is equal to four or greater. For this reason, only the phase variance associated with higher order modes has been investigated and it is assumed that all of the phase from piston, tip, and tilt are represented with another device such as a FSM.

3.2.3.1 Fidelity Levels.

A level of fidelity must be chosen to determine the amount of overall phase variance missing from the combined effects of pixelation and finite modes. This level represents an acceptable limit to fidelity. Once this total fidelity level is chosen, it is budgeted between the amount missing due to finite modes and the amount missing from the Nyquist bandlimit of the device. Since the amount of phase standard deviation missing from pixelation is dependent on the number of modes, the level from finite modes must be chosen first. For example, if the researcher determines that it is acceptable to represent only 95% of the total phase standard deviation, the amount present by having finite modes will be 97.5% of the total. Then the amount representable by using a pixelated device will be 97.5% of the phase standard deviation left after losing some due to finite modes. These values are given by

$$\begin{aligned} \% \text{ of Total Present} &= (\% \text{ of Total Present due to Finite Modes}) \\ &\times (\% \text{ of Modes Present due to Pixelation}). \end{aligned} \quad (3.23)$$

In an effort to simplify these two fidelity levels, the percentage of the total phase standard deviation present due to finite modes was set equal to the percentage present after pixelating the total mode phase standard deviation. This reduces the above equation to

$$\% \text{ of Total Present} = (\% \text{ Present due to Modes or Pixelation})^2. \quad (3.24)$$

An example is given to clarify this confusing concept. According to Sasiela [32], the total amount of phase standard deviation that exists in a wavefront after piston, tip, and tilt have been removed (PTR) is given by

$$\sigma_{\phi_{\text{PTR}}} = \frac{\sqrt{0.134}}{2\pi} \left(\frac{D}{r_0} \right)^{5/6} \approx 0.05826 \left(\frac{D}{r_0} \right)^{5/6}, \quad (3.25)$$

as measured in waves. When D/r_0 is normalized out of this equation, the total PTR phase standard deviation $\sigma_{\phi_{\text{PTR}}}$ of a wavefront is approximately 0.058260 waves. If the researcher can accept a total loss of 5% of phase standard deviation over an aperture, that leaves $(0.95)(0.058260) \approx 0.055347$ waves of phase standard deviation that must be represented with the wavefront control device. The percentage present after only a finite number of modes has been represented can be found using Eq. (3.24) and is given as $(0.95)^{1/2} \approx 0.975$. This leads to $(0.975)(0.058260) \approx 0.056804$ waves of total phase standard deviation present after summing to a finite number Zernike modes. The total phase standard deviation of that value present after considering the Nyquist bandlimit of the device is also 97.5% as given by the fidelity level equations. This leads to $(0.975)(0.056804) \approx 0.0553839$ waves of total PTR phase standard deviation that a device can represent after losing some to finite modes and some to the spatial frequency of the device. This value is approximately 95% of the total PTR phase standard deviation of a wavefront as given by Sasiela [32] in Eq. (3.25). Therefore, this example shows how levels of fidelity are evaluated in this research. A total loss of 5% can be found by losing 2.5% of the total phase standard deviation present due to finite modes and then losing another 2.5% of that due to the Nyquist bandlimit of a wavefront control device.

This example and analysis method leads to the threshold levels shown in Table 3.1 for both finite modes and pixelation of a device, given a threshold for the overall normalized phase standard deviation present in a device. Several common threshold levels are shown in this table as well as the 83% threshold level which is also in all subsequent tables in this research to support an example in Chapter IV.

3.2.3.2 Phase Standard Deviation from Finite Zernike Modes.

It has been shown in Section 3.2.3.1 that the total phase standard deviation missing depends on the total number of Zernike modes needed to reach a certain fidelity level of total phase standard deviation present after removing piston, tip, and tilt. To find this, an equation is needed for the total phase variance present in a given

Table 3.1: Threshold percentages for finite modes and pixelation of a device given a threshold for the overall normalized phase standard deviation present in a device.

Percentage of total $\sigma_{\phi_{PTR}}$ present	Percentage present from finite modes $\%_{\text{pres}_{\text{modes}}}$	Percentage present from pixelation $\%_{\text{pres}_{\text{pixels}}}$
99%	99.5%	99.5%
98%	99.0%	99.0%
97%	98.5%	98.5%
96%	98.0%	98.0%
95%	97.5%	97.5%
90%	94.9%	94.9%
83%	91.1%	91.1%

Zernike mode $\sigma_{\phi_i}^2$. Fortunately, Roggemann and Welsh [31] offer this expression as

$$\sigma_{\phi_i}^2 = \overline{a_i^2} = 4.5 \times 10^{-4} \pi^{8/3} \left(\frac{D}{r_0} \right)^{5/3} (-1)^{(n-m)} \alpha \frac{\Gamma\left(\frac{14}{3}\right) \Gamma\left(\beta - \frac{11}{6}\right)}{\left[\Gamma\left(\frac{17}{6}\right)\right]^2 \Gamma\left(\beta + \frac{17}{6}\right)}, \quad (3.26)$$

where $\overline{a_i^2}$ is the variance of the Zernike coefficients, i is the Zernike mode in question, $\beta = n + 1$, and $\alpha = 16\beta$. Recall from Section 2.3 that n and m are directly correlated to the i^{th} Zernike mode, as referenced in Table 2.1. Equation (3.26) does not allow for the next step in the process. Therefore, an equation is needed that provides the phase variance present in a Zernike mode as a function of spatial frequency. Sasiela [32] gives such an equation with

$$\sigma_{\phi_i}^2 = 0.2073 k^2 \int_0^L dz C_n^2(z) \int d\kappa f(\kappa) \cos^2[P(\gamma, \kappa, z)] \hat{F}(\gamma\kappa), \quad (3.27)$$

where $P(\gamma, \kappa, z)$ specifies the type of optical source and is zero due to no actual propagation taking place. Further, $f(\kappa) = \kappa^{-11/3}$, as given by the Kolmogorov spectrum, and $\hat{F}(\gamma, \kappa)$ is a filter function used to represent the individual Zernike modes given

by

$$\hat{F}(\gamma\kappa) = \begin{cases} F_{m,n}(\kappa)_x \\ F_{m,n}(\kappa)_y \\ F_{0,n}(\kappa) \end{cases} = (n+1) \left[\frac{2J_{n+1}\left(\frac{D}{2}\kappa\right)}{\frac{D}{2}\kappa} \right]^2 \begin{cases} 2\cos^2(m\phi) \\ 2\sin^2(m\phi) \\ 1 \end{cases}, \quad (3.28)$$

where m and n are the azimuthal and radial orders, respectively, and J_{n+1} is a Bessel function of the first kind. This analysis requires the use of the Kolmogorov spectrum and discontinues a PSD that is dependent on the inner scale because an analytic solution cannot be found for the total phase variance in a given mode with these more complex equations. Equation (3.27) can be written in terms of r_0 after substituting Eq. (2.15). Also, when all modes are being considered $F_{m,n}(\kappa)_x$ and $F_{m,n}(\kappa)_y$ can be combined so that the trigonometric functions drop out due to the identity

$$\sin^2(x) + \cos^2(x) = 1. \quad (3.29)$$

The result is a simpler equation for the total phase variance present in each mode as a function of r_0 and spatial frequency given by

$$\sigma_{\phi_i}^2 = 0.98014\pi D^{-2} r_0^{-5/3} \alpha \int_0^\infty \left[J_\beta \left(\frac{D}{2}\kappa \right) \right]^2 \kappa^{-14/3} d\kappa, \quad (3.30)$$

where once again $\beta = n + 1$ and $\alpha = 16\beta$.

Before investigating how this equation can lead to a relationship between the minimum atmospheric coherence diameter and the Nyquist bandlimit of a device, it is important to verify Eq (3.30). If a device is continuous, the limits of integration shown in Eq. (3.30) are correct and they represent the total phase variance present in the i^{th} Zernike mode. This value should be equal to that given by Eq. (3.26) which implicitly includes all spatial frequencies. In order to compare these two equations, an analytic form must be found for Eq. (3.30). The help of Mellin transforms $\mathcal{M}\{f(t); s\}$ can be used to obtain such a solution. Poularikas [28] gives the scaling property of a

Mellin transform as

$$\mathcal{M}\{f(at); s\} = \int_0^\infty f(at)t^{s-1}dt = a^{-s} \int_0^\infty f(x)x^{s-1}dx = a^{-s}F(s). \quad (3.31)$$

Sasiela [32] makes this property useful with the Mellin transform pair for the square of a Bessel function of the first kind which is given by

$$\mathcal{M}\{J_\beta^2(x)\} = \frac{1}{2\sqrt{\pi}} \frac{\Gamma\left(\frac{s}{2} + \beta\right) \Gamma\left(\frac{1}{2} - \frac{s}{2}\right)}{\Gamma\left(\beta + 1 - \frac{s}{2}\right) \Gamma\left(1 - \frac{s}{2}\right)} \quad (3.32)$$

Using the relationships expressed in Eqs. (3.31) and (3.32) to simplify Eq. (3.30) leads to

$$\sigma_{\phi_i}^2 = 0.49007(2)^{-11/3} \sqrt{\pi} \left(\frac{D}{r_0}\right)^{5/3} \frac{\Gamma\left(\frac{14}{6}\right) \Gamma\left(\beta - \frac{11}{6}\right)}{\Gamma\left(\frac{17}{6}\right) \Gamma\left(\beta + \frac{17}{6}\right)}, \quad (3.33)$$

which is another analytic expression for the total phase variance present in the i^{th} Zernike mode. The difference is that Eq. (3.33) is derived from an equation given as a function of spatial frequency. This expression looks similar to that given by Roggemann and Welsh in Eq. (3.26). However, to verify that both expressions yield the same values, they can be normalized so that D/r_0 is removed from the equations and plotted on the same graph. This graph is shown in Fig. 3.3 and it is visually apparent that Eqs. (3.26) and (3.33) yield the same values. Now that a method for expressing the total phase standard deviation as a function of modes has been presented, the number of modes corresponding to the thresholds defined in Table 3.1 can be found. These modes are shown in Fig. 3.4 and Table 3.2.

3.2.3.3 Phase Standard Deviation from Finite Number of Pixels.

An expression for phase standard deviation as a function of spatial frequency can be investigated with the number of modes known for each fidelity threshold of interest to this research. The verification of Eq. (3.33) implies that the equation it was derived from is also correct. Equation (3.30) must be normalized so that D/r_0 can be removed from the integral. This is accomplished by converting the spatial frequency variable κ to a dummy variable for normalized linear spatial frequency η ,

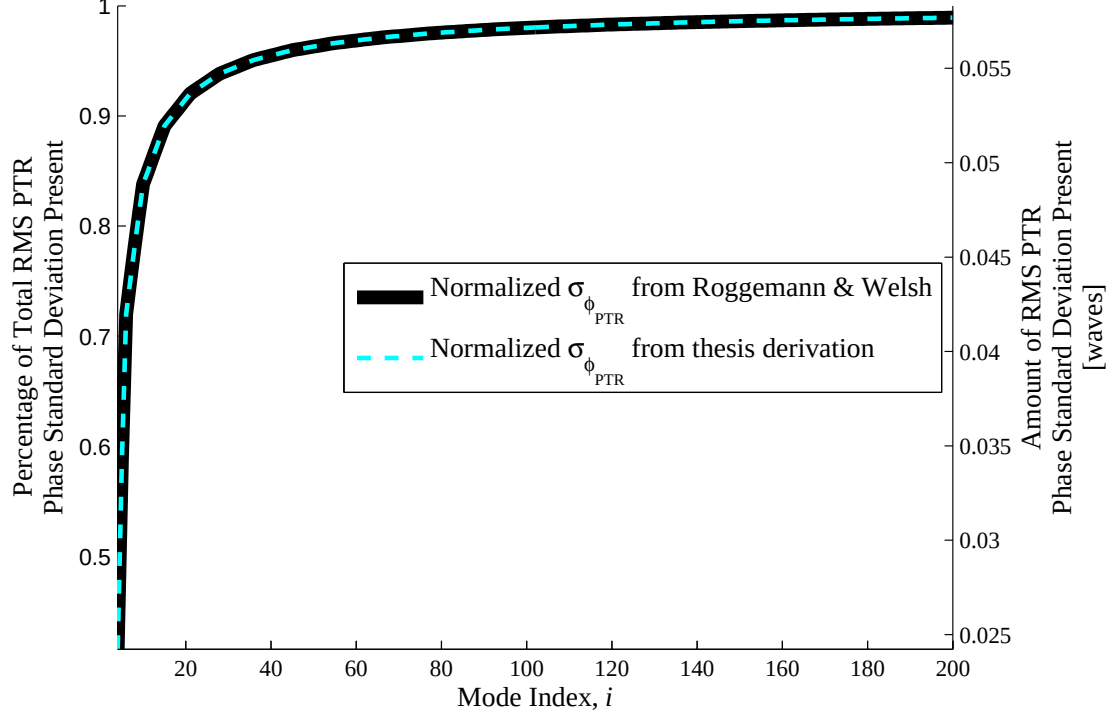


Figure 3.3: Comparison of total normalized phase standard deviation as a function of modes equations from Roggemann and Welsh and this research's derivation

where

$$\eta = \kappa \frac{D}{2}. \quad (3.34)$$

This substitution allows the phase variance present in each mode to be given by

$$\sigma_{\phi_i}^2 = 0.98014(2)^{-11/3} \pi \alpha \left(\frac{D}{r_0} \right)^{5/3} \int_0^\infty [J_\beta(\eta)]^2 \eta^{-14/3} d\eta, \quad (3.35)$$

which is only a function of the variable η and is directly related to the total number of pixels on a device when D/r_0 is normalized out of the equation. The expression

$$\eta_{\max} = N \frac{\pi}{2} \quad (3.36)$$

converts the total number of pixels on a device to the maximum value of the variable η to determine how much phase variance is present on a device with a limited number

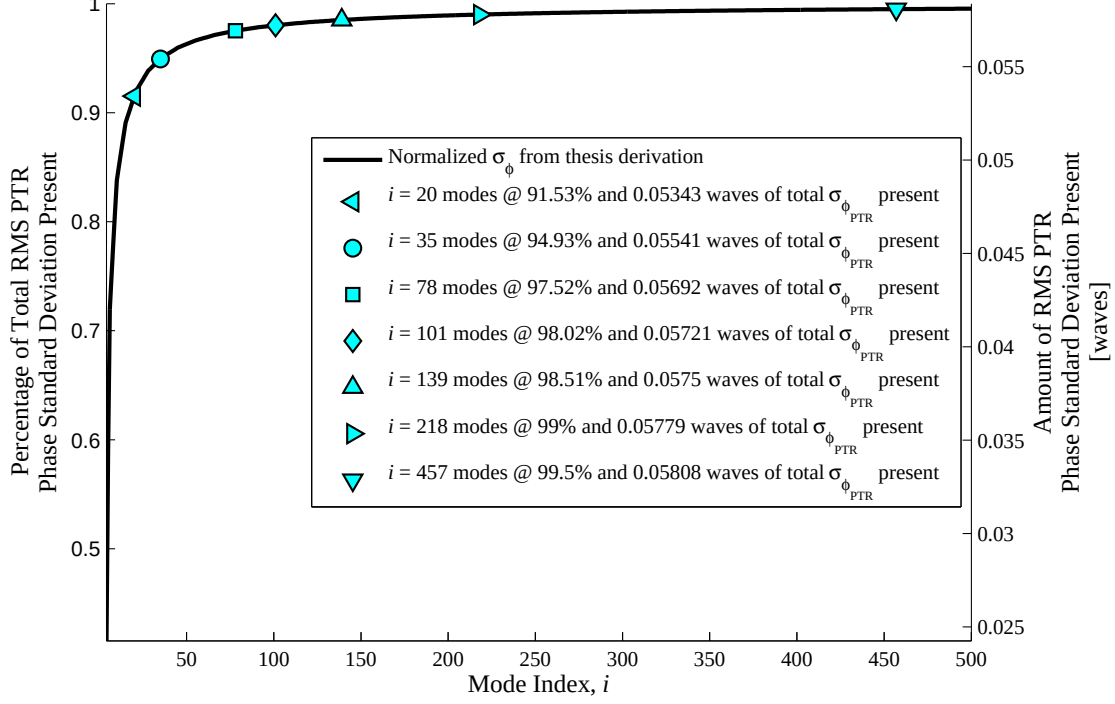


Figure 3.4: Number of modes required to reach a variety of thresholds of total phase standard deviation present after using a discrete number of Zernike modes.

of pixels. Substituting this into the upper limit of integration in Eq. (3.35) gives

$$\sigma_{\phi_i}^2(\eta) = 0.98014(2)^{-11/3}\pi\alpha\left(\frac{D}{r_0}\right)^{5/3}\int_0^{\eta_{\max}}[J_\beta(\eta)]^2\eta^{-14/3}d\eta, \quad (3.37)$$

which is an expression for the total amount of phase variance present in a given mode due to the finite pixels on a wavefront control device.

Finally, the amount of phase variance present on a segmented wavefront control device with a limited number of pixels after a finite number of higher order Zernike modes have been summed can be expressed by

$$\sigma_{\phi_{\text{pixels}}}^2(\eta) = \sum_{i=4}^{\text{total modes present}} \sigma_{\phi_i}^2(\eta). \quad (3.38)$$

Table 3.2: Number of modes required to reach certain thresholds of total phase standard deviation present after using a discrete number of Zernike modes.

Percentage of total $\sigma_{\phi_{PTR}}$ present	Percentage present from finite modes	Number of modes required to reach threshold
99%	99.5%	457
98%	99.0%	218
97%	98.5%	139
96%	98.0%	101
95%	97.5%	78
90%	94.9%	35
83%	91.1%	20

Earlier in this section, it was stated that piston, tip, and tilt are not included in this sum because they are easy to include with another device and interrupt the true number of higher order modes that a complex wavefront control device can produce.

Figure 3.5 shows the plot of Eq. (3.38). This plot shows the total normalized phase standard deviation present as a function of spatial frequency for several levels of fidelity after limiting the number of Zernike modes present on a wavefront. Since different threshold levels require a different number of Zernike modes to be summed in Eq. (3.38), the plots are slightly different for each threshold level. The steepest slope is seen at the lowest threshold level, where there are the fewest Zernike polynomials summed. This is intuitive because fewer modes mean a lower total phase standard deviation present, and therefore, it takes fewer η samples to sum to that number. The number of pixels required for a wavefront control device for each threshold level shown in Table 3.1 are given in Table 3.3. Figures 3.3, 3.4, and 3.5 are the result of the **Matlab**® code shown in Appendix A.

The results shown in Table 3.3 provide the total number of modes and number of pixels a device must have to successfully create a certain percentage of the turbulence associated with all higher order Zernike modes. It suggests that to recreate the atmosphere from all higher order modes with only a five percent loss, a wavefront

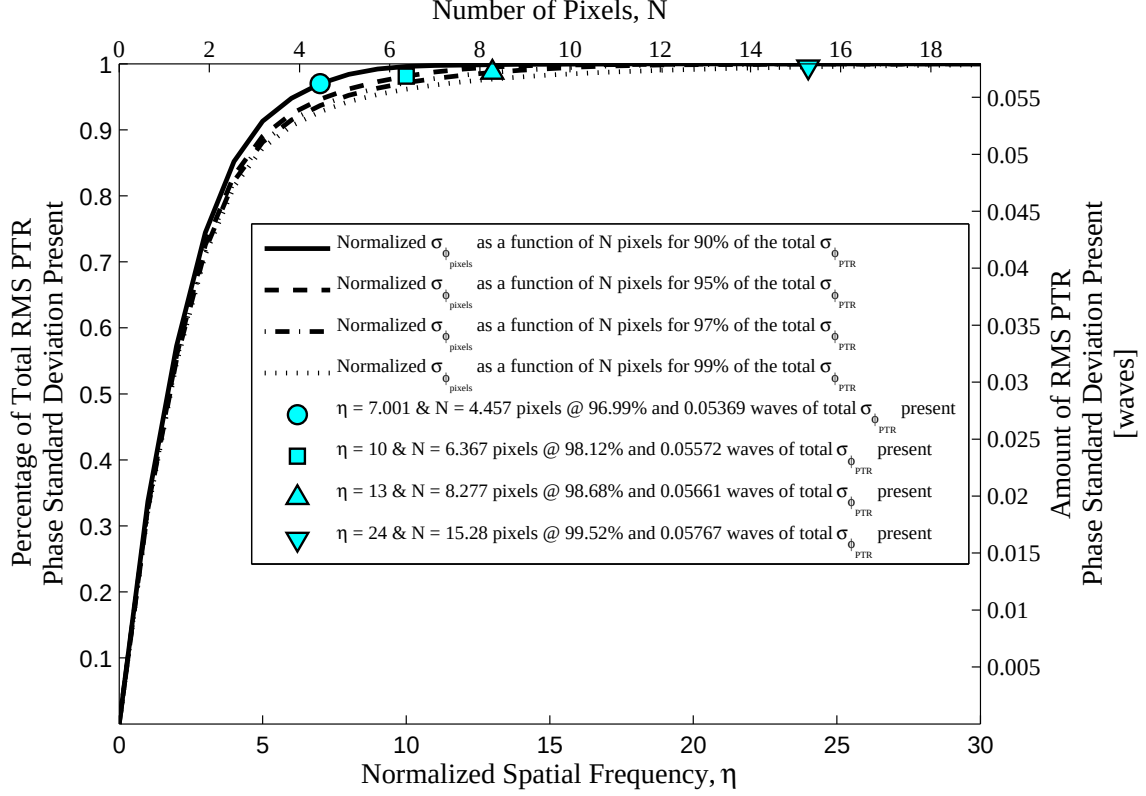


Figure 3.5: Number of pixels required to reach a variety of threshold levels of total phase standard deviation present after using a Nyquist bandlimited device and only summing over a discrete number of Zernike modes.

control device with only six or seven pixels across would be necessary. This value seems extremely low. Thus it is important to verify these results with simulation.

3.2.3.4 Minimum Theoretical Atmospheric Coherence Length.

Simulation verification is shown in Chapter IV; however, it is first necessary to complete the analysis of finding the minimum possible value for r_0 that can be represented by all higher order modes with a specific wavefront control device. This expression comes from using Eq. (3.25) for a specific device. The minimum r_0 is the value at which the achieved phase standard deviation is close enough to the desired phase standard deviation. Close enough is defined by the desired fidelity limits previously described in Section 3.2.3.1. This value should not be confused with the effective r_0 of the turbulence on the device. Standard MEMS DMs have five to

Table 3.3: Number of pixels theoretically required to reach certain threshold levels of total phase standard deviation present after using a discrete number of Zernike modes.

Percentage of total $\sigma_{\phi_{\text{PTR}}}$ present	Finite mode $\%_{\text{threshold}}$	# of modes required	Finite pixel $\%_{\text{threshold}}$	η_{max}	# of pixels required
99%	99.5%	457	99.5%	24	15.28
98%	99.0%	218	99.0%	16	10.19
97%	98.5%	139	98.5%	13	8.28
96%	98.0%	101	98.0%	11	7.00
95%	97.5%	78	97.5%	10	6.37
90%	94.9%	35	94.9%	7	4.46
83%	91.1%	20	91.1%	6	3.82

twelve mirror segments across the device, so an example is presented using a seven pixel device to better understand this concept.

A 7×7 pixel device is able to represent 96% of the total PTR phase standard deviation according to Table 3.3. Therefore, four percent of the total is missing when a device with only seven pixels across is used. The minimum possible value for r_0 that this device can represent is a function of how many waves of phase standard deviation a researcher is willing to accept as missing. This expression is given by

$$\sigma_{\phi_{\text{PTR}_{\text{missing}}}} = \%_{\text{missing}} \left(\frac{\sqrt{0.134}}{2\pi} \right) \left(\frac{D}{r_{0_{\text{min}}}} \right)^{5/6}, \quad (3.39)$$

where $\sigma_{\phi_{\text{PTR}_{\text{missing}}}}$ is the number of waves of missing PTR phase standard deviation that a researcher can accept and $\%_{\text{missing}}$ is the percentage of missing PTR phase standard deviation as a result of the number of pixels on a device. Solving Eq. (3.39) for $r_{0_{\text{min}}}$ yields

$$r_{0_{\text{min}}} = D \left[\frac{\sqrt{0.134}}{2\pi} \left(\frac{\%_{\text{missing}}}{\sigma_{\phi_{\text{PTR}_{\text{missing}}}}} \right) \right]^{6/5}. \quad (3.40)$$

Equation (3.40) indirectly relates the minimum atmospheric coherence length to the number of pixels across a device through the use of Eqs. (3.36), (3.37), and (3.38). These equations all lead to a value for $\%_{\text{missing}}$ for a specific number of pixels across

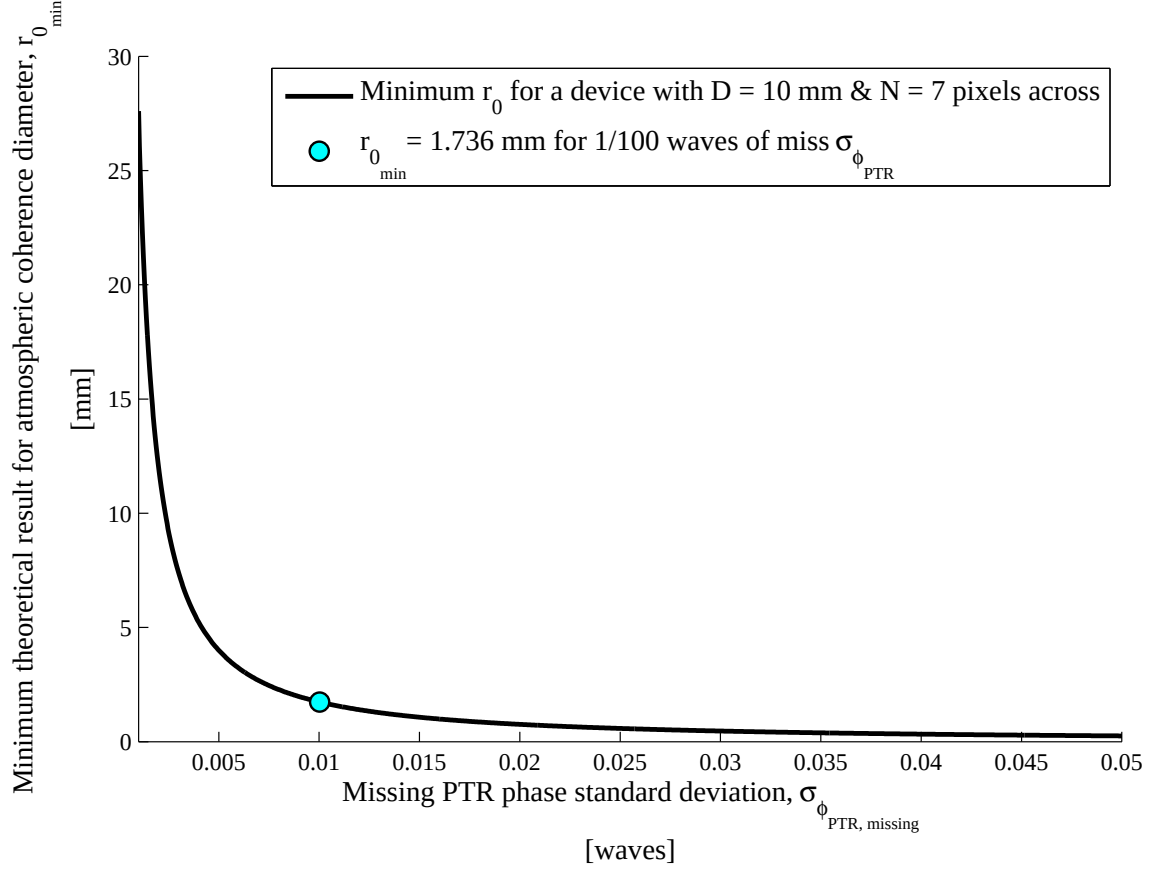


Figure 3.6: Minimum atmospheric coherence length as a function of missing PTR phase standard deviation for theoretical results for a 7×7 pixel device with an aperture diameter $D = 10$ mm.

a device which, in turn, leads to the minimum possible $r_{0\min}$ value for that device. In the example, $N = 7$ and $\%_{\text{missing}} = 4\%$. Equation (3.40) is shown as a function of $\sigma_{\phi_{\text{PTR,missing}}}$ in Fig. 3.6 when the aperture diameter is set arbitrarily as $D = 10$ mm. Figure 3.6 shows that, in theory, a 7×7 pixel device with an aperture of $D = 10$ mm can represent a minimum value for $r_0 = 1.736$ mm when the researcher is willing to have 1/100 waves of missing PTR phase standard deviation. An even smaller value for $r_{0\min}$ can be obtained if the researcher can accept a larger amount of missing PTR phase standard deviation. These results show that the maximum strength of turbulence that a phase screen can represent is dependent on the requirements of the researcher. The results gained from this analysis will be discussed in Chapter V.

IV. Simulation and Analysis

This chapter discusses how the results from the previous chapter were verified with computer simulation. It begins with the creation and pixelation of phase screens and then moves on to how phase variance is extracted from a computer model. Finally, the results from the simulation are presented.

4.1 *Creating Phase Screens*

The first step in simulating the theoretical results obtained in the previous chapter is to create atmospheric phase screens. The goal of these simulations is to verify the number of pixels across a device required to represent a certain threshold level of phase standard deviation present in the emulated turbulence over an aperture. This goal is limited to the characteristics of a phase screen on a single wavefront control device. This eliminates all need for simulating optical propagation through the atmosphere. All that is necessary in this verification is to create a random number of phase screens, compute the statistics of those screens, and compare those to the theoretical values calculated in the previous chapter.

One of the most accepted ways to create a turbulent phase screen in simulation is to use the Fourier series method discussed in Section 2.2.2. This method allows for near-continuous phase screens that incorporate a large number of Fourier series coefficients, which produces more realistic atmospheric phase screens than is possible by summing a finite number of Zernike modes. The Fourier series method produces phase screens in simulation that are similar to recreating phase screens by summing over a near-infinite number of Zernike modes.

This research capitalized on previous work performed by Dr. Eric Magee of MZA Associates, Inc. and used several of his `Matlab`[®] functions to create the necessary phase screens. Appendix B contains the `Matlab`[®] file written to create phase screens via the Fourier series method. This listing creates 250 random phase screens over an aperture of unit radius all with values for $D/r_0 = 5$. It uses the Kolmogorov turbulence spectrum discussed in Section 2.1 to produce screens with 128 Fourier

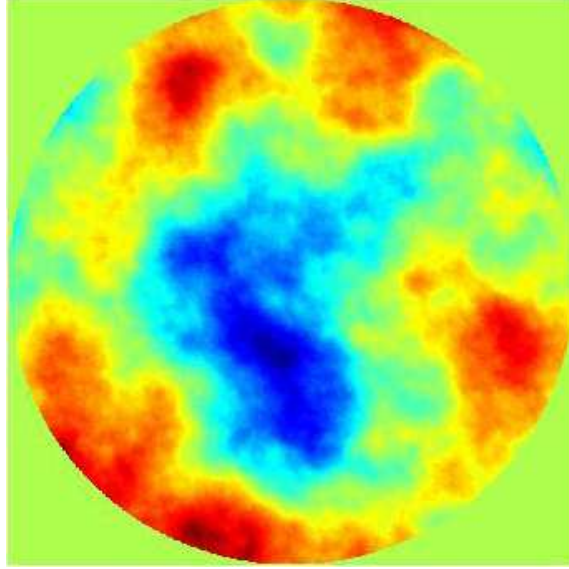


Figure 4.1: Random individual realization of a piston/tilt removed phase screen created using the Fourier series method.

series coefficients. A large number of realizations is needed to accurately calculate the statistics of the phase screens. Two hundred and fifty random phase screen realizations offer a large enough sample base to provide good statistical values of the phase variance. D/r_0 can be set to any number because it is eventually normalized out of the simulation results. Enough Fourier series coefficients are needed to create realistic turbulence in the phase screens. The value chosen allows for realistic near-continuous phase screens without taking too much computational effort. First, this listing calls the **Matlab**[®] function `mkPSD_thesis.m` which creates the individual power spectral density realizations for the phase screens to be created with the Fourier series method. Next, Listing B.1 calls the **Matlab**[®] function `FSCoeff_thesis.m` which calculates the Fourier series coefficients from the individual PSD realizations. After that, the listing calls the **Matlab**[®] function `FSScreenReal_thesis.m` which actually generates the individual phase screen realizations. The three **Matlab**[®] functions called were all provided by Dr. Magee and may be available upon request via email at emagee@mza.com.

After an individual phase screen has been created, Listing B.1 calculates the Zernike coefficients for piston, tip, and tilt. This allows the program to then remove the first three Zernike modes from the phase screen realization. It was shown in Section 2.3 that Zernike polynomials can be removed from a phase screen through simple subtraction due to each mode’s orthogonality over a circular aperture. A PTR phase screen is necessary because this research focuses on the ability of a wavefront control device to create atmosphere with only the higher order modes included. A FSM can easily reproduce the effects of the first three Zernike modes as previously discussed in Section 3.2.3. The final result of this `Matlab`[®] code is 250 random realizations of piston/tilt removed phase screens. One of these realizations is shown in Fig. 4.1. This figure is representative of all 250 realizations since each one is created randomly.

4.2 Pixelation of Phase Screens

The next step in the simulation process is to down-sample each phase screen. Limitations in `Matlab`[®]’s processing capabilities limit the size of each phase screen to 512×512 pixels. This limitation means that the near continuous sample shown in Fig. 4.1 is actually a 512×512 pixelated phase screen. In order to find the effect of pixelation on the standard deviation across many realizations of a phase screen, it is necessary to create an array of pixelated phase screens for which to find the statistics.

The `Matlab`[®] code that down-samples the phase screen realizations can be viewed in Appendix C. It is essentially an averaging function that down-samples the phase screen by

$$N_{\text{new}} = \frac{N_{\text{orig}}}{n}, \quad (4.1)$$

where N_{new} is the number of pixels desired across the down-sampled phase screen, N_{orig} is the number of pixels across the original phase screen, and n is the value that satisfies this equation. The `Matlab`[®] code maps the original phase screen onto a $N_{\text{new}} \times N_{\text{new}}$ grid with each square inside the grid composed of $n \times n$ pixels of the

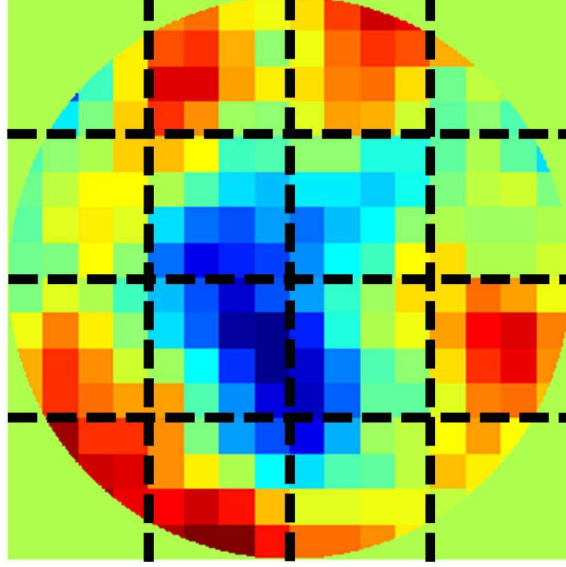


Figure 4.2: $N_{\text{new}} \times N_{\text{new}} = 4 \times 4$ grid showing each square of $n \times n = 16$ pixels that are averaged to down-sample a $N_{\text{orig}} \times N_{\text{orig}} = 16 \times 16$ pixel phase screen.

original phase screen. This grid is shown in Fig. 4.2 for an example where $N_{\text{orig}} = 16$ is down-sampled to $N_{\text{new}} = 4$ with a $n \times n = 4 \times 4$ grid. Next, each pixel in the $n \times n$ square is replaced with the average of all the pixels inside the square. Repeating this method at every square on the grid leads to a phase screen that is actually N_{orig} pixels across, but has been down-sampled to simulate a screen that is only N_{new} pixels across. Each down-sampled screen remains $N_{\text{orig}} \times N_{\text{orig}}$ pixels which enables it to better map to the matrix sizes of the other variables in the code. Also, keeping the pixelated screens the original size enables them to easily be placed on a SLM in the future if experimental verification of these results is pursued.

For this research $N_{\text{orig}} = 512$ pixels, and n is a power of two ranging between 1 and 512. N_{orig} must be divisible by n so that N_{new} is an integer, because the number of pixels cannot be fractional. This method yields ten differently pixelated phase screens (each with 250 realizations) ranging from 1×1 pixels to 512×512 pixels. An example of one of the down-sampled screens is shown in Fig. 4.3, and the original 512×512 pixel screen is shown in Fig. 4.1.

4.3 Finding Phase Variance

The next step is to find the phase variance across the 250 realizations of each of the ten levels of pixelated phase screens. This value is obtained by calculating the variance of the Zernike coefficient a_i for each high order mode represented across all 250 phase screen realizations. The number of modes required for a given threshold of phase standard deviation desired to be present in the emulated atmosphere can be found from Table 3.2. Once the number of modes is determined, the Zernike coefficient for each mode can be found for an individual phase screen realization with Eq. (2.24). As shown in Eq. (3.26), the phase variance of the i^{th} Zernike mode is found by calculating the statistical variance of the Zernike coefficients over every phase screen realization for that mode. The total phase variance present now becomes the sum of the variances for each mode. The **Matlab**[®] listing in Appendix C computes this phase variance for each of the ten levels of pixelation for $i = 4$ to $i = 139$ Zernike modes.

The simulation analysis was limited to the 97% threshold level of total phase standard deviation present in the emulated turbulence due to the memory limitations in **Matlab**[®]. In order for the listing in Appendix C to run in a reasonable amount of time, the Zernike polynomials had to be pre-computed. This created a variable in **Matlab**[®] that was $512 \times 512 \times i$ in size, where i is the number of modes used. This variable became too large for **Matlab**[®] to keep stored and continue running when the number of modes pre-computed reached the 98% threshold level in Table 3.2. It was possible to find these values in **Matlab**[®] by computing each Zernike polynomial inside of a loop and then clearing the variable; however, the execution time for this was infeasible. It would require over two weeks to run the program for all ten pixelation levels out to the 99% threshold value of total phase standard deviation present. A supercomputer could have been accessed to speed up this process; however, the results obtained from the lower threshold levels were sufficient to forgo pursuing the phase variance values for the 98 and 99% threshold levels.

Table 4.1: Simulation results for the number of pixels required to reach certain threshold levels of total phase standard deviation present after using a discrete number of Zernike modes.

Percentage of total $\sigma_{\phi_{PTR}}$ present	Finite mode % _{threshold}	# of modes required	Finite pixel % _{threshold}	$\eta_{\max}$	# of pixels required
99%	99.5%	457	99.5%	<i>N/A</i>	<i>N/A</i>
98%	99.0%	218	99.0%	<i>N/A</i>	<i>N/A</i>
97%	98.5%	139	98.5%	40	25.47
96%	98.0%	101	98.0%	33	21.01
95%	97.5%	78	97.5%	28	17.83
90%	94.9%	35	94.9%	17	10.82
83%	91.1%	20	91.1%	11	7.00

4.4 Results

The final step is to plot the results. These plots are the result of the **Matlab**[®] code shown in Appendix A. Figure 4.4 displays the theoretical and simulation plots of normalized phase standard deviation. This is shown as a function of the number of pixels across a device as a result of having only 95% of the total phase standard deviation present. This is due to finite Zernike modes and a limited spatial frequency. This phase variance is normalized because the $D/r_0 = 5$ value was easy to remove from the overall phase variance after it was computed.

Figure 4.4 shows that the simulation results match closely with the theoretical values but are not quite the same. This discrepancy is discussed in Chapter V. There are only ten data points from which to plot the simulation results due to only obtaining the phase standard deviation values at ten pixelation levels of the phase screen realizations. The **interp** command in **Matlab**[®] is used to create the continuous simulation plot line in the figure. This command conducts a piecewise cubic Hermite interpolation of the ten simulation data points. This interpolation allows for an estimation of the plot values to be obtained for every value of η . Figure 4.4 only shows the results for the overall 95% threshold case; however, the other cases from the theoretical analysis shown in Table 3.3 are presented for the simulation analysis

in Table 4.1. As previously stated, the 98 and 99% cases could not be evaluated in simulation due to the limitations of the memory in `Matlab`[®]. These simulation results are compared to the theoretical analysis results in Chapter V.

4.4.1 Minimum Possible Atmospheric Coherence Length in Simulation.

The example in Section 3.2.3.4 describing the minimum possible value of atmospheric coherence diameter that a device can produce is revisited. This allows for a comparison between the simulation and theoretical results. In this example, a 7×7 pixel device is theoretically able to represent 96% of the total PTR phase standard deviation. However, this device can only represent 83% of the total PTR phase standard deviation according to the simulation results in Table 4.1. Therefore, seventeen percent of the total is missing when a device with only seven pixels across is used in simulation. The minimum possible value for r_0 that this device can represent is given in Eq. (3.40). In the example for the simulation results, $N = 7$ and $\%_{\text{missing}} = 17\%$. Equation (3.40) is shown as a function of $\sigma_{\phi_{\text{PTR}_{\text{missing}}}}$ in Fig. 4.5 when an arbitrary value for the aperture diameter is set as $D = 10$ mm. Figure 4.5 shows that, in simulation, a 7×7 pixel device with an aperture of $D = 10$ mm can represent a minimum value for $r_0 = 9.856$ mm when the researcher is willing to allow 1/100 waves of missing PTR phase standard deviation. An even smaller value for $r_{0_{\text{min}}}$ can be obtained if the researcher can accept a larger amount of missing PTR phase standard deviation. These results show that the maximum amount of turbulence that a phase screen can represent is dependent on the requirements of the researcher. The results gained from this example are discussed in Chapter V and compared to the theoretical results obtained in Section 3.2.3.4.

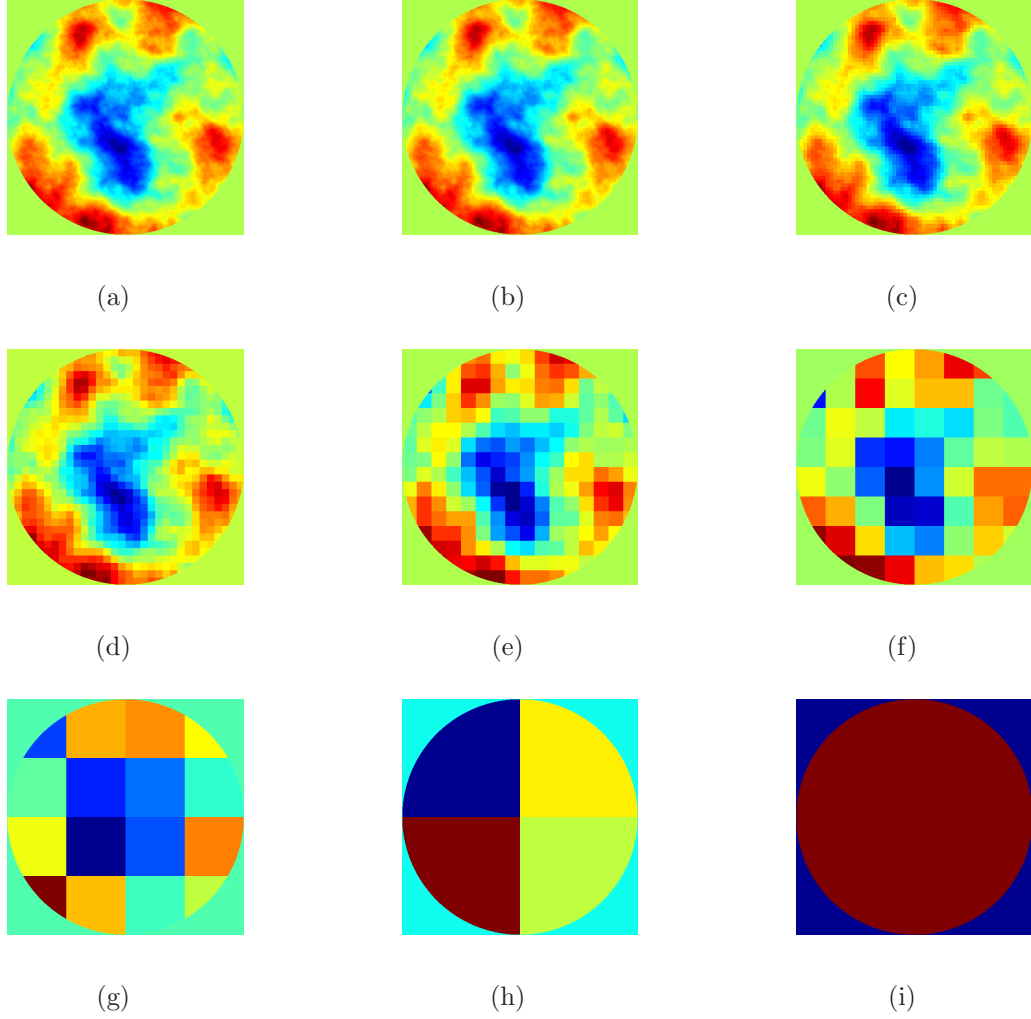


Figure 4.3: Random individual realization of a piston/tilt removed phase screen created using the Fourier series method and then down-sampled to: (a) 256×256 pixels (b) 128×128 pixels (c) 64×64 pixels (d) 32×32 pixels (e) 16×16 pixels (f) 8×8 pixels (g) 4×4 pixels (h) 2×2 pixels (i) 1×1 pixels

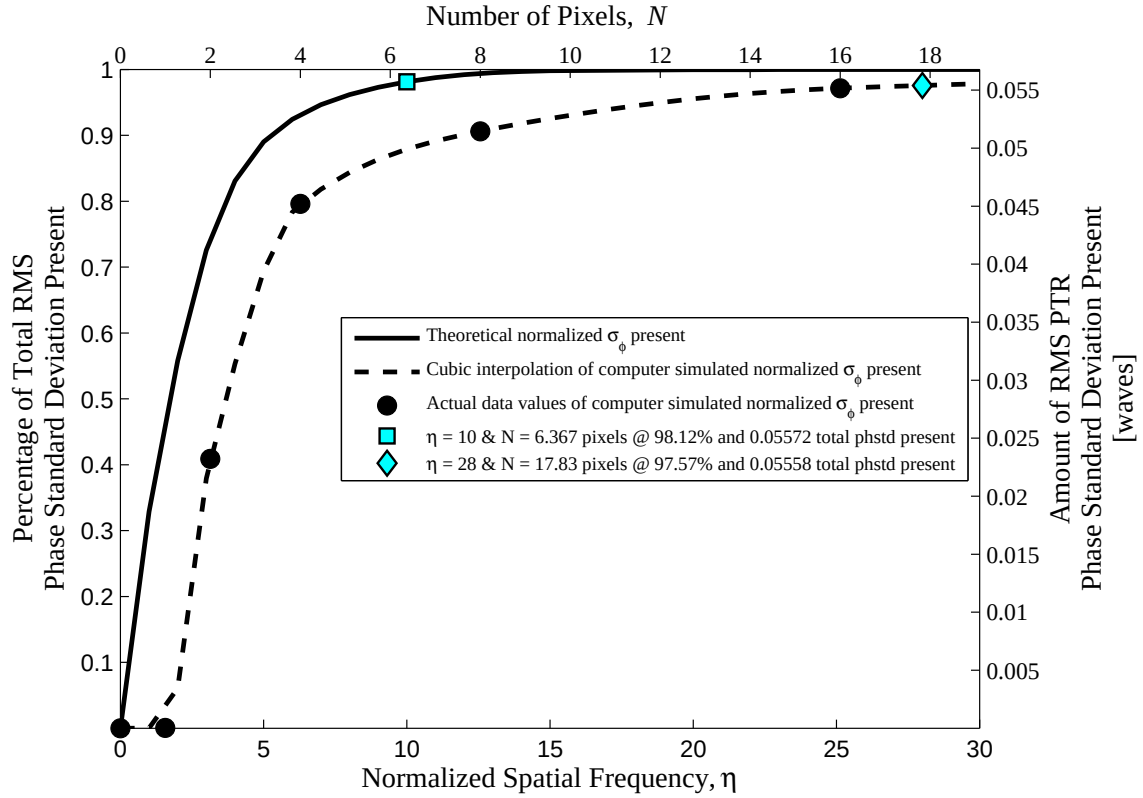


Figure 4.4: Normalized phase standard deviation present as a function of the number of pixels across a wavefront control device after only having 95% of the total phase standard deviation present due to finite Zernike modes and a limited spatial frequency.

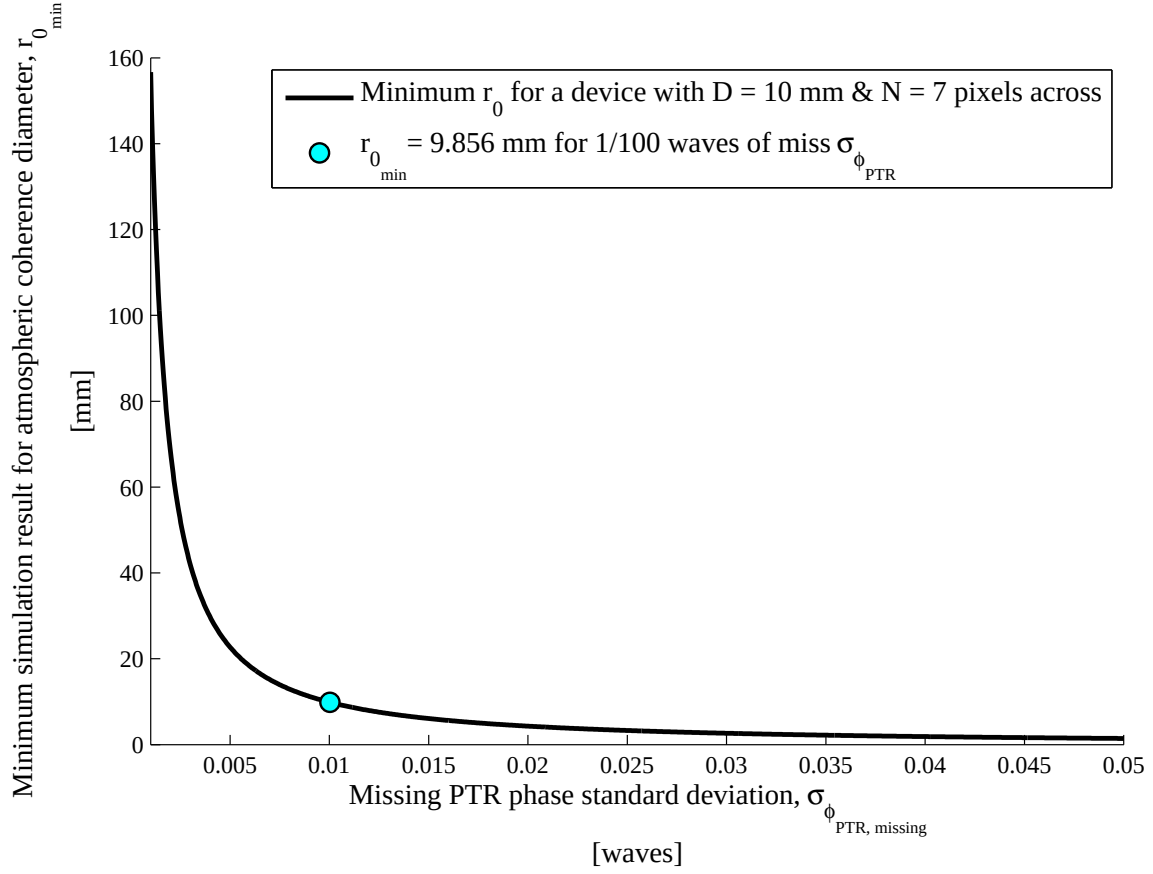


Figure 4.5: Minimum atmospheric coherence length as a function of missing PTR phase standard deviation for simulation results for a 7×7 pixel device with an aperture diameter $D = 10$ mm.

V. Conclusions and Future Work

This chapter revisits the key results of this research and presents overall conclusions. Further, it addresses the main challenges faced during this research and how well the initial goals were satisfied. Finally, this chapter presents some recommendations for extensions to this research as well as future work that could be pursued.

5.1 *Results and Conclusions*

5.1.1 *Limitations of Current Low-cost Wavefront Control Devices.*

Recall, Chapter II presented a review of relevant theory and literature for this research. This review identified several key limitations of segmented wavefront control devices already researched and highlighted knowledge gaps that are bridged in this thesis. It was found that there are two main types of segmented wavefront control devices, each with its own advantages and disadvantages. These two types, NLC SLMs and MEMS DMs, are both compact, inexpensive, electronically addressable, highly reproducible, and suited for AO research and turbulence emulation in a laboratory environment.

NLC SLMs have much larger arrays of independently controllable elements (at least 512×512 pixels) without any moving parts. Disadvantages of NLC SLMs are that they have slow relaxation times and, because of this, they cannot currently be used in real-time AO systems. NLC SLMs are also wavelength and polarization dependent devices and have digitization limitations with only 128 programmable levels. MEMS DMs have the advantage of extremely high frame-rates and can be used in real-time AO systems. Segmented MEMS DMs do not have interactuator coupling, but are limited by diffraction effects. Using lenslets to focus the incident light in the center of each mirror segment has been shown to mitigate these adverse effects.

This review did not reveal any in-depth research on the effects of pixelation on abilities of NLC SLMs and MEMS DMs. This thesis research bridged that knowledge

gap through an investigation of the limitations of Nyquist bandwidth as a result of pixelation for segmented wavefront control devices.

5.1.2 *Theoretically Emulating the Atmosphere.*

Chapter III has examined the theory of layered turbulence to uncover the maximum turbulence that could be emulated with only two phase screens. It was desired to find the minimum possible atmospheric coherence diameter and maximum possible Rytov number that could be achieved with two phase screens. This research uncovered that both of these important atmospheric turbulence parameters were functions of the minimum possible atmospheric coherence diameter that a single segmented wavefront control device could produce. Additionally, it was found that the Rytov number is at a maximum when both phase screens in a system are at the beginning of the optical propagation path. This is difficult in a laboratory due to the physical size of wavefront control devices. Therefore, Fig. 3.1 presents a contour plot identifying the maximum possible Rytov number a system could achieve based on the position of each phase screen.

Another important result of the theoretical analysis was that minimum atmospheric coherence diameter is a function of Nyquist frequency which is directly related to the number of pixels across a wavefront control device. An expression, given by Eq. (3.22), was found for the relationship between $r_{0_{\min}}$ and the Nyquist bandlimit of a device at a single point on a device. This presented the first of many challenges in this research because without a detailed correlation matrix of every point on a wavefront control device, this relationship does not have a practical application.

Phase standard deviation threshold levels and Zernike modes were pursued to find a more practical relationship of atmospheric coherence diameter to the number of pixels on a device over the entire aperture of the device. An expression for the total amount of PTR phase standard deviation present in a Zernike mode as a function of normalized spatial frequency was developed and is shown in Eq. (3.37). This result was confirmed to be accurate through comparison with a similar expression given by

Roggemann and Welsh [31] that assumed infinite bandwidth. **Matlab**[®] was then used to perform a numeric analysis to find the total percentage of PTR phase standard deviation a device can represent as a function of the number of pixels across the device. Finally, an example was presented in Section 3.2.3.4 to show how this relationship can lead to the minimum possible atmospheric coherence diameter a single wavefront control device can represent. This representation is a function of the number of waves of missing PTR phase standard deviation a researcher is willing to accept.

This roundabout numeric method for finding an expression for the minimum coherence diameter of a device is necessary because it is not possible to derive a direct analytic expression. The results from this numeric analysis can be substituted into Eqs. (3.2) and (3.6) to obtain the minimum overall r_0 and maximum Rytov number for a two phase screen system. This can yield the true limitations in the strength of turbulence a two phase screen system can emulate when using segmented wavefront control devices as phase screens. The results depend on the acceptable limitations of the researcher and the specific wavefront control device used.

5.1.3 Emulating the Atmosphere in Simulation.

Chapter IV provides a confirmation of the results obtained theoretically through **Matlab**[®] simulation. Random realizations of PTR phase screens were created so that the phase standard deviation over those screens could be calculated. These realizations were then individually down-sampled to several pixelation levels. An expression for the amount of PTR phase standard deviation present in a simulated device as a function of the number of pixels across that device was then interpolated from the simulation results using **Matlab**[®]. The simulation was unfortunately limited to the 97% threshold level of total PTR phase standard deviation because of the limitations in **Matlab**[®]'s memory capability. This limitation prevented a comparison between the 98 and 99% threshold level results from theory and simulation. Finally, the same example pursued in Chapter III was revisited for the simulation results.

Table 5.1: Comparison of theoretical and simulation results for the number of pixels required to reach certain threshold levels of total phase standard deviation present after using a discrete number of Zernike modes.

% of total $\sigma_{\phi_{\text{PTR}}}$ present	Finite mode & pixel % _{thresh}	# of modes required	Pixels required in theory	Pixels required in simulation
99%	99.5%	457	15.28	N/A
98%	99.0%	218	10.19	N/A
97%	98.5%	139	8.28	25.47
96%	98.0%	101	7.00	21.01
95%	97.5%	78	6.37	17.83
90%	94.9%	35	4.46	10.82
83%	91.1%	20	3.82	7.00

5.1.4 Discrepancies Between Theoretical and Simulation Analysis.

The numerical evaluation and **Matlab**[®] simulations disagree about the amount of PTR phase standard deviation present in a device. The plot of this bandlimited PTR phase standard deviation shown in Fig. 4.4 shows a discrepancy between the two different methods. This discrepancy can also be visualized in the last two columns of Table 5.1 which is a combination of the results from Tables 3.3 and 4.1. This may appear to be large at low spatial frequencies, but it can be explained quite simply. Figure 4.4 shows that a 7×7 pixel device can theoretically represent 96% of the total PTR phase standard deviation present in the atmosphere. This is in contrast to the simulation, where it would take a 21×21 pixel device to represent the same amount of atmospheric turbulence. Table 5.1 further shows that, in simulation, a 7×7 pixel device is only capable of representing 83% of the total PTR phase standard deviation present in the atmosphere.

Part of the discrepancy between the theoretical and simulation results can be justified by looking at how each method filters out the threshold levels. The theoretical analysis numerically evaluates an integral that goes from zero to infinity when all the PTR phase standard deviation is present. This analysis simply cuts off the integral at a maximum normalized spatial frequency η_{max} to find the amount present up to a

certain number of pixels. This acts as a rectangle filter function on the equation and is shown by

$$\sigma_{\text{PTR}_{\text{pres}}} \propto \int_0^{\eta_{\text{max}}} f(\eta) d\eta \quad (5.1)$$

$$\propto \int_0^{\infty} f(\eta) \text{rect} \left(\frac{\eta}{\eta_{\text{max}}} - \frac{1}{2} \right) d\eta, \quad (5.2)$$

where $f(\eta)$ is a function of normalized spatial frequency and the rectangle function filters the integral so that it is only evaluated from zero to η_{max} . This filter function causes the sharp drop seen in the theoretical results shown in Fig. 4.4. The simulation method does not use a rectangle function to filter out the unwanted phase standard deviation. It merely calculates the phase standard deviation of random realizations of phase screens at different pixelation levels. It is shown in Fig. 4.4 that as the number of pixels on a device increases, the difference between the simulation and theoretical results decreases. This means that the discrepancy at low normalized spatial frequencies is caused by the pixelation process. In other words, the pixelation process acts as a filtering function that removes more phase standard deviation than the rectangle function described in Eq. (5.1). Therefore, it takes more pixels to make up for the phase lost from down-sampling the phase screen.

This result can be validated by considering the down-sampling process. It is simply an averaging procedure of the phase over large sections of the aperture. This averaging procedure has adverse affects that reduce the amount of PTR phase standard deviation calculated across many phase screen realizations. It is clear that an increase in the number of pixels that are averaged together leads to an increase in the discrepancy between the theoretical and simulation results of PTR phase standard deviation. The derivation of the actual shape of the filter function from pixelation of the phase screens is left for future research. However, a visual approximation of these filter functions is shown in Fig. 5.1. Figure 5.1 explains part of the reason for why pixelation results are different than the theoretical results. This process also explains some of the discrepancy between the examples at the end of Chapters III and IV

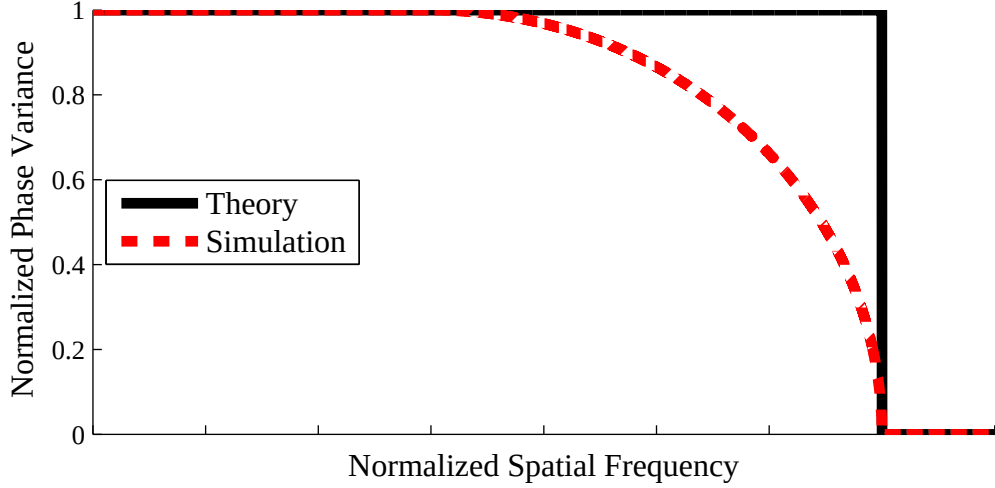


Figure 5.1: Estimation of the filter functions applied in theory and in simulation that result in discrepancies between the results of the two different methods at low spatial frequencies.

Table 5.2: Comparison of theoretical and simulation results for $r_{0_{\min}}$ of a 7×7 pixel wavefront control device.

Analysis method	Number of pixels	$r_{0_{\min}}$ for $D = 10$ mm and $1/100$ waves of $\sigma_{\phi_{\text{PTR}_{\text{missing}}}}$	$D/r_{0_{\min}}$
Theory	7×7	1.736 mm	5.759
Simulation	7×7	9.856 mm	1.015

whose results are shown in Table 5.2. The example shows that a 7×7 pixel device can theoretically produce a minimum value of atmospheric coherence length that is over five times smaller than the same device can produce in simulation.

Another explanation for part of the discrepancy between the theoretical results and the simulation results is the sharp edges of the pixels. Sharp variations between values at the edges of two pixels cause fluctuations in frequency space. The edges of the pixels act as rectangle functions that become sinc functions when transformed into the frequency domain. The cutoff frequency for each pixel is twice that of the theoretical cutoff $F_{\max}$ given by Eq. (3.3). This leads to the effect shown in Fig. 5.2 in which the sinc function causes some of the desired phase variance to be missed and includes some unwanted phase variance from higher spatial frequencies. It is unclear

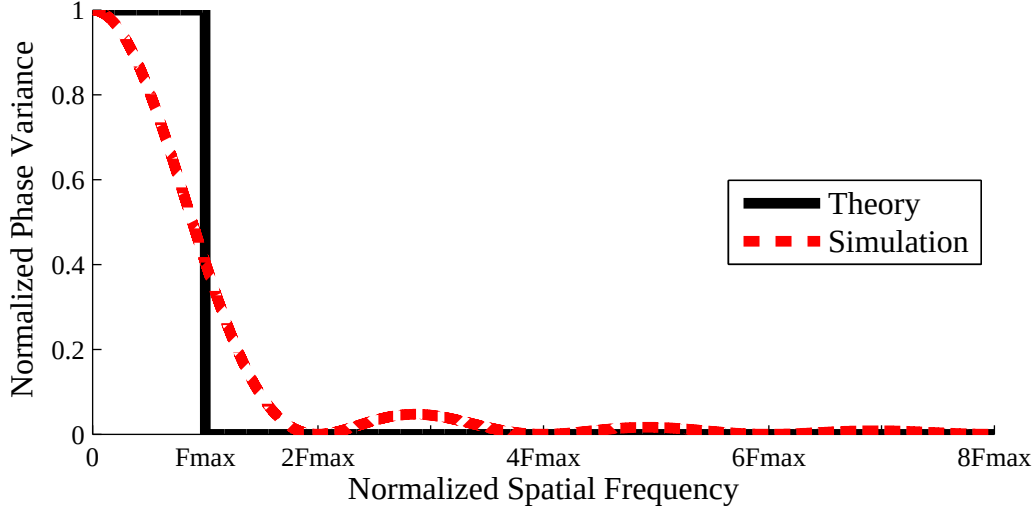


Figure 5.2: Discrepancy in theoretical vs. simulation results from sharp pixel edges on a wavefront control device.

how much is truly added or missing as a result of this phenomenon. This effect could be included to make the theoretical approach more realistic; however, it would require knowledge of the correlation matrix between all of the pixels on a wavefront control device. This is left as a topic for future research. Figure 5.2 shows that the phase variance present in a pixelated phase screen is not entirely representative of the atmospheric turbulence being emulated.

The expected results in an actual laboratory should be closer to those of the simulation than the theory developed in this research. This is due to down-sampling being required when a phase screen is loaded onto a wavefront control device so that it matches the number of pixels on that device. Therefore, the pixelation filtering process also takes place in a laboratory environment so that phase screens can be loaded onto wavefront control devices. The number of pixels across a device directly affects the amount of phase that is lost in this process. A 512×512 NLC SLM would not experience nearly as much of a loss in phase as a standard 25 to 144-mirror segmented MEMS DM. However, the theory and simulation results agree for large N . According to Fig. 4.4 a standard NLC SLM should be able to produce approximately the same minimum atmospheric coherence length in theory as in simulation. Therefore, the

actual device in a laboratory should also meet these expectations. A MEMS DM with only a few mirror segments is limited by the down-sampling procedure used to map a phase screen to the device. It is possible that a more complex method of down-sampling phase screens would yield simulation results closer to those of the theory. These filter functions explain the difference between the theoretical and simulation results in this research.

5.1.5 Practicality of Results.

The results of this thesis can be brought together in one simple plot for the minimum number of pixels across a wavefront control device for a given amount of missing RMS PTR phase standard deviation. This plot is representative of the worst possible turbulence conditions and is shown in Fig. 5.3. Figure 5.3 was created from the simulation results for the 97% case using 139 Zernike modes. The simulation results were used because they are more realistic than the theoretical results. This plot was limited to only 139 Zernike modes due to the `Matlab`[®] memory limitations explained in Section 4.3. Figure 5.3 was created using Eq. (3.39) to find the total $\sigma_{\phi_{\text{PTR}_{\text{missing}}}}$ for a variety of D/r_0 values. The values for $\%_{\text{missing}}$ were converted to total pixels across a wavefront control device through a plot similar to that found in Fig. 4.4.

The practical application of these results can be shown with an example. A given laboratory setup uses a laser operating at a wavelength of $\lambda = 500$ nm and has a total propagation distance of $L = 2$ m. The experiment in question requires two wavefront control devices each with an aperture diameter of $D = 10$ mm. The researcher wants to know the minimum number of pixels a wavefront control device must have to represent a value of $D/r_0 = 10$ and only be missing 1/20 or 1/100 waves of RMS PTR phase standard deviation. According to Fig 5.3, a 7×7 pixel device is necessary to represent $D/r_0 = 10$ to within 1/20 waves and a 30×30 pixel device is required to represent $D/r_0 = 10$ to within 1/100 waves of RMS PTR phase standard deviation. The researcher also wants to know the maximum Rytov number this setup

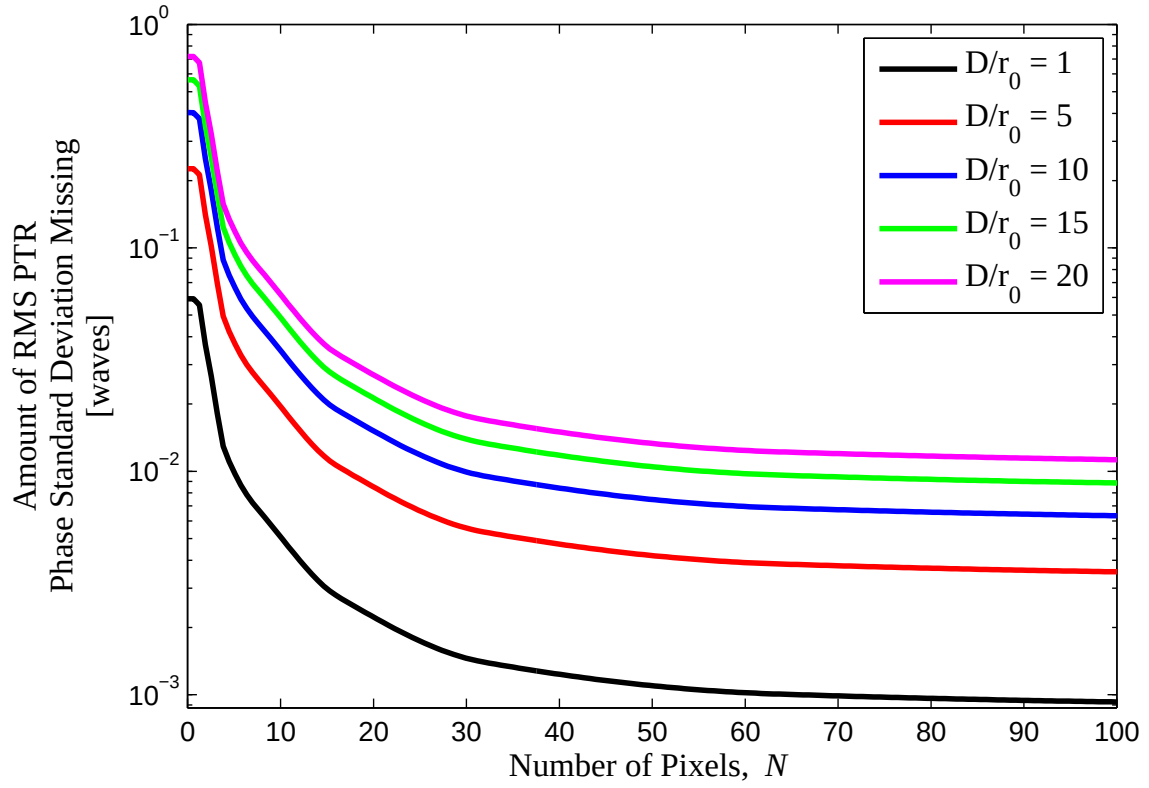


Figure 5.3: Total RMS PTR phase standard deviation missing $\sigma_{\phi_{\text{PTR missing}}}$ versus the minimum number of pixels across a wavefront control device required to represent given values of D/r_0 .

can emulate. The desired $D/r_0 = 10$ constraint yields $r_{0\min} = 1$ mm for $D = 10$ mm. To achieve the maximum possible Rytov number, both wavefront control devices are placed at the very beginning of the optical path. This yields a slightly unrealistic value of $z = 0$ for the position of both phase screens. The resulting value for the maximum possible Rytov number for this scenario when all of these parameters are substituted into Eq. (3.6) is $\sigma_{\mathcal{R}_{\max}}^2 = 0.646$.

In conclusion, NLC SLMs are rarely adversely affected by their Nyquist bandlimit, but segmented MEMS DMs do experience limitations in the amount of atmospheric turbulence they can reproduce as a result of their number of mirror segments. However, both devices are capable of reproducing the total desired PTR phase standard deviation with high levels of fidelity. The requirements of each individual research experiment should determine if the abilities of a particular device are sufficient.

These results were much more promising than initially anticipated. It was expected that there would be severe limitations in the abilities of all segmented devices. This was not the case, and it turns out that only segmented MEMS DMs with very few mirror segments are affected by the Nyquist bandlimit of the device. The other advantages and disadvantages of both devices should be weighed when considering which type of wavefront control device is best suited for a laboratory experiment.

5.2 *Future Work*

This research is complete, but there are many avenues that may be investigated to further this investigation. First, the results of the theoretical and simulation analysis presented can be confirmed experimentally. Time did not allow for any experiments; however, such an investigation should be carried out for a variety of devices with different Nyquist bandlimits to verify the effects of pixelation on a phase screen device.

Also, an investigation into the autocorrelation matrix between individual points over an aperture can be pursued. A better understanding of how single points of an

aperture are related will yield an analytic expression for the minimum atmospheric coherence diameter as a function of Nyquist bandlimit for a wavefront control device. An analytic approach would not have the limitations from finite Zernike modes that were revealed in the numerical analysis in this research.

Another extension of this research is the investigation of how different quantities of phase screens affect the overall atmospheric properties that can be represented in a laboratory. Along the same line of research, the best locations for phase screens in a laboratory can be determined for different types of optical sources. This research only looked at a plane wave, but a spherical wave would be interesting as well.

New designs of segmented wavefront control devices could also be researched. The pixels on NLC SLMs and MEMS DMs have sharp edges. These edges impart extraneous high frequencies into the phase screen. It is possible that a spatial filter inside a 4-f system could make a pixelated system appear continuous and mitigate these effects.

The simulation analysis can also be extended along several paths. A supercomputer can be utilized to carry out the simulation results to higher threshold levels. Also, an investigation could be mounted to fully understand how the pixelation process filters the phase standard deviation results. This could lead into work on improving the down-sampling procedure to mitigate the adverse effects down-sampling has on the simulation results. A more complex method of pixelating a phase screen could yield better simulation results.

Finally, future work could be done to compare the capabilities of segmented wavefront control devices with those of the more traditional atmospheric phase wheels. An investigation in how wavefront sensors are affected by segmented wavefront control devices would also be of interest. These future investigations could truly uncover the full range of limitations and capabilities of segmented wavefront control devices.

Appendix A. Matlab Code to Compute Theoretical and Simulation Results

Listing A.1: Matlab[®] file to compute theoretical and simulation results.

```
% Theoretical and Simulation Analysis
clc;
format compact;
warning off all;
5
T = true;
F = false;

% Set Flags
10 include_Dr0 = F;

% Define user changeable variables
eta_max = 1e4; % maximum extent of normalized spatial freq
D_over_r0 = 40; % value for D/r0 if it is included
15
% Uncomment the percent threshold level to be used for the amount ...
% of phase
% standard deviation present after finite modes are considered
% mode_percent_pres_THRESH = 0.911; % 83% total percentage ...
% of waves present
% mode_percent_pres_THRESH = 0.949; % 90% total percentage ...
% of waves present
20 mode_percent_pres_THRESH = 0.975; % 95% total percentage of ...
% waves present
% mode_percent_pres_THRESH = 0.980; % 96% total percentage ...
% of waves present
% mode_percent_pres_THRESH = 0.985; % 97% total percentage ...
% of waves present
% mode_percent_pres_THRESH = 0.990; % 98% total percentage ...
% of waves present
% mode_percent_pres_THRESH = 0.995; % 99% total percentage ...
% of waves present
25
% ALSO CHANGE:
% SIM_PTR_stdev_pres_after_pixel_percent(Y,:)
% PLOURDE_percent_pres_simulation_pos_XX
% PLOURDE_percent_pres_simulation_value_XX
30 % PLOURDE_actual_pres_simulation_value_XX
% cor_eta_PLOURDE_percent_pres_simulation_XX
% cor_Z_mode_PLOURDE_percent_pres_mode_equation_XX
% in LINES 447 - 452
% to correct value for XX and Y
35
% Set the percentage of waves allowed missing from finite number ...
% of pixels and modes
pixel_percent_pres_THRESH = mode_percent_pres_THRESH;
```

```

40 % Set plot limits for later
    xmin_modes = 4;
    xmax_modes1 = 200;
    xmax_modes2 = 500;

45    xmax_eta = 30;
        xmax_pixels = 2*xmax_eta/pi;

% Set up Zernike modes to be used
% Load Zernike index files (containing n, m,& J values for ...
each Zernike mode
50    load('zernike_index.mat');
        load('zernike_index_no_trig.mat');

% Remove piston and tilt modes from the index files
    zernike_index(1:3,:) = [];
55    zernike_index_no_trig(1:2,:) = [];

% Set up Zernike modes to be used
    n = zernike_index(:,1);
    m = zernike_index(:,2);
60    J = zernike_index(:,3);
        nu = n + 1;
        alpha = 16.*nu;

    n_trig = zernike_index_no_trig(:,1);
65    m_trig = zernike_index_no_trig(:,2);
    J_trig = zernike_index_no_trig(:,3);
        nu_trig = n_trig + 1;
        alpha_trig = 16.*nu_trig;

70 % Set up the normalized spatial frequency dummy variable for ...
integration
    eta_min = eps;
    eta_numpts = eta_max;
    eta_space = (eta_max - eta_min)/eta_numpts;
    eta = linspace(eta_min, eta_max, eta_numpts);

75    D_Fx = eta./pi; % spatial frequency
    D_over_delta_pix = 2*eta./pi; % number of pixels across for ...
        a given point in eta

80 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mode Equations Defined in Roggemann and Welsh %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Define Roggemann Mode Equation
85 for idx1 = 1:length(zernike_index);

    % Compare to Roggemann phvar & stdev equations

```

```

mode_phvar_ALL_Rogg(idx1,:) = 4.5e-4*pi^(8/3).*(-1).^(n(...
    idx1) - m(idx1)).*alpha(idx1).*...
    ((gamma(14/3).*gamma(nu(idx1) - 11/6))./(gamma(17/6))...
    ^2.*gamma(nu(idx1) + 17/6)));
90 end

% Set up an if statement to include D/r0 functions
if include_Dr0
    % Find the D/r0 weighted phase variance present as a ...
    function of the number of modes
95     cum_PTR_phvar_pres_Rogg = D_over_r0^(5/3).*cumsum(...
        mode_phvar_ALL_Rogg);    %[rad^2]
else
    % Find the normalized phase variance present as a function...
    of the number of modes
    cum_PTR_phvar_pres_Rogg = cumsum(mode_phvar_ALL_Rogg)...
        ;    %[rad^2]
end
100

% Convert to phase standard deviation Present in waves
cum_PTR_phstdev_pres_Rogg = sqrt(cum_PTR_phvar_pres_Rogg)./(2*...
    pi);    %[waves]

% Find Missing phase standard deviation in waves
105     cum_PTR_phstdev_miss_Rogg = max(cum_PTR_phstdev_pres_Rogg) - ...
        cum_PTR_phstdev_pres_Rogg;    % [waves]

% Find the percentage of total PTR phase standard deviation ...
present
cum_PTR_phstdev_pres_Rogg_percent = cum_PTR_phstdev_pres_Rogg...
    ./max(cum_PTR_phstdev_pres_Rogg);

110
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mode Equations Theoretically Derived by Plourde %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

115 % Define the constants to be used in the equations below
    C_theory = (0.2073/0.423)*2^(-11/3)*sqrt(pi);

% Define Plourde Mode Equation
    for idx1 = 1:length(zernike_index);
120
        % Define the analytic phvar & stdev equations
        mode_phvar_wh_theory_ALL(idx1,:) = C_theory.*alpha(...
            idx1).*((gamma(nu(idx1) - 11/6).*gamma(14/6))./(...
            gamma(nu(idx1) + 17/6).*gamma(17/6)));
    end

125 % Set up an if statement to include D/r0 functions
    if include_Dr0

```

```

        % Find the D/r0 weighted phase variance present as a ...
        function of the number of modes
        cum_PTR_phvar_pres_theory = D_over_r0^(5/3).*cumsum(...
            mode_phvar_wh_theory_ALL);           %[rad^2]
    else
130        % Find the normalized phase variance present as a function...
        of the number of modes
        cum_PTR_phvar_pres_theory = cumsum(...
            mode_phvar_wh_theory_ALL);           %[rad^2]
    end

    % Convert to phase standard deviation Present in waves
135    cum_PTR_phstdev_pres_PLOURDE = sqrt(cum_PTR_phvar_pres_theory)...
        ./(2*pi);           %[waves]

    % Convert to percentage of total phase standard deviation present
    cum_PTR_phstdev_pres_PLOURDE_percent = ...
        cum_PTR_phstdev_pres_PLOURDE ./max(...
            cum_PTR_phstdev_pres_PLOURDE);

140    % Find positions of and modes associated with the various ...
    percentage threshold levels
    PLOURDE_percent_pres_mode_equation_pos_99 = min(find(...
        cum_PTR_phstdev_pres_PLOURDE_percent >= 0.995));
    PLOURDE_percent_pres_mode_equation_value_99 = ...
        cum_PTR_phstdev_pres_PLOURDE_percent(...
            PLOURDE_percent_pres_mode_equation_pos_99);
    PLOURDE_actual_pres_mode_equation_value_99 = ...
        cum_PTR_phstdev_pres_PLOURDE(...
            PLOURDE_percent_pres_mode_equation_pos_99);
    cor_Z_mode_PLOURDE_percent_pres_mode_equation_99 = J(...
        PLOURDE_percent_pres_mode_equation_pos_99);

145    PLOURDE_percent_pres_mode_equation_pos_98 = min(find(...
        cum_PTR_phstdev_pres_PLOURDE_percent >= 0.990));
    PLOURDE_percent_pres_mode_equation_value_98 = ...
        cum_PTR_phstdev_pres_PLOURDE_percent(...
            PLOURDE_percent_pres_mode_equation_pos_98);
    PLOURDE_actual_pres_mode_equation_value_98 = ...
        cum_PTR_phstdev_pres_PLOURDE(...
            PLOURDE_percent_pres_mode_equation_pos_98);
    cor_Z_mode_PLOURDE_percent_pres_mode_equation_98 = J(...
        PLOURDE_percent_pres_mode_equation_pos_98);

150    PLOURDE_percent_pres_mode_equation_pos_97 = min(find(...
        cum_PTR_phstdev_pres_PLOURDE_percent >= 0.985));
    PLOURDE_percent_pres_mode_equation_value_97 = ...
        cum_PTR_phstdev_pres_PLOURDE_percent(...
            PLOURDE_percent_pres_mode_equation_pos_97);
    PLOURDE_actual_pres_mode_equation_value_97 = ...
        cum_PTR_phstdev_pres_PLOURDE(...
            PLOURDE_percent_pres_mode_equation_pos_97);

```

```

cor_Z_mode_PLOURDE_percent_pres_mode_equation_97 = J(...
    PLOURDE_percent_pres_mode_equation_pos_97);
155
PLOURDE_percent_pres_mode_equation_pos_96 = min(find(...
    cum_PTR_phstdev_pres_PLOURDE_percent >= 0.980));
PLOURDE_percent_pres_mode_equation_value_96 = ...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
    PLOURDE_percent_pres_mode_equation_pos_96);
PLOURDE_actual_pres_mode_equation_value_96 = ...
    cum_PTR_phstdev_pres_PLOURDE(...
    PLOURDE_percent_pres_mode_equation_pos_96);
cor_Z_mode_PLOURDE_percent_pres_mode_equation_96 = J(...
    PLOURDE_percent_pres_mode_equation_pos_96);
160
PLOURDE_percent_pres_mode_equation_pos_95 = min(find(...
    cum_PTR_phstdev_pres_PLOURDE_percent >= 0.975));
PLOURDE_percent_pres_mode_equation_value_95 = ...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
    PLOURDE_percent_pres_mode_equation_pos_95);
PLOURDE_actual_pres_mode_equation_value_95 = ...
    cum_PTR_phstdev_pres_PLOURDE(...
    PLOURDE_percent_pres_mode_equation_pos_95);
cor_Z_mode_PLOURDE_percent_pres_mode_equation_95 = J(...
    PLOURDE_percent_pres_mode_equation_pos_95);
165
PLOURDE_percent_pres_mode_equation_pos_90 = min(find(...
    cum_PTR_phstdev_pres_PLOURDE_percent >= 0.949));
PLOURDE_percent_pres_mode_equation_value_90 = ...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
    PLOURDE_percent_pres_mode_equation_pos_90);
PLOURDE_actual_pres_mode_equation_value_90 = ...
    cum_PTR_phstdev_pres_PLOURDE(...
    PLOURDE_percent_pres_mode_equation_pos_90);
cor_Z_mode_PLOURDE_percent_pres_mode_equation_90 = J(...
    PLOURDE_percent_pres_mode_equation_pos_90);
170
PLOURDE_percent_pres_mode_equation_pos_83 = min(find(...
    cum_PTR_phstdev_pres_PLOURDE_percent >= 0.911));
PLOURDE_percent_pres_mode_equation_value_83 = ...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
    PLOURDE_percent_pres_mode_equation_pos_83);
PLOURDE_actual_pres_mode_equation_value_83 = ...
    cum_PTR_phstdev_pres_PLOURDE(...
    PLOURDE_percent_pres_mode_equation_pos_83);
cor_Z_mode_PLOURDE_percent_pres_mode_equation_83 = J(...
    PLOURDE_percent_pres_mode_equation_pos_83);
175
% Plot Plourde mode equation vs Rogg mode equations to verify they...
    are the same
figure(1); clf;
    hold on;

```

```

plot(J, cum_PTR_phstdev_pres_Rogg_percent, 'k-', 'linewidth'...
    , 9);
180 plot(J, cum_PTR_phstdev_pres_PLOURDE_percent, 'c--', '...
    linewidth', 3);
    hold off;
h_1 = legend('Normalized \sigma_{\phi_{PTR}} from Roggemann & ...
    Welsh',...
        'Normalized \sigma_{\phi_{PTR}} from thesis ...
        derivation', 'location', 'E');
% h_t = title('\bf Percentage of Normalized Phase Std ...
Deviation Present as a Function of Zernike Mode Index \rm')
185 % 'Total Phase Std Deviation Expressed as a Fraction ...
of PTR ';...
% 'after including the first \iti\rm Zernike modes in ...
representing the atmosphere'; ' ';...
% ['For error from having finite modes = ' num2str...
(100*mode_percent_pres_THRESH,'%6.4g') '% of total Phase Std ...
Deviation Present']});
ax_left = gca;
    xlim([xmin_modes xmax_modes1]); ylim([min(...
        cum_PTR_phstdev_pres_Rogg_percent) max(...
        cum_PTR_phstdev_pres_Rogg_percent)]);
190 h_y1 = ylabel({'Percentage of Total RMS PTR'; 'Phase ...
    Standard Deviation Present'});
ax_right = axes('Position',get(ax_left,'Position'),'...
YAxisLocation','right','Color','none','YColor','k');
    xlim([xmin_modes xmax_modes1]); ylim([min(...
        cum_PTR_phstdev_pres_Rogg) max(...
        cum_PTR_phstdev_pres_Rogg)]);
    h_x2 = xlabel('Mode Index, \iti\rm');
    h_y2 = ylabel({'Amount of RMS PTR'; 'Phase Standard ...
    Deviation Present'; '[waves]'});
195 set(gca, 'FontName' , 'Times New Roman');
set([h_x2, h_y1, h_y2, h_1], 'FontName' , 'Times New Roman'...
    , 'fontsize', 16);

% Plot values of percent threshold levels against the mode index ...
on the Plourde mode equations plot
figure(2); clf;
200 hold on;
plot(J, cum_PTR_phstdev_pres_PLOURDE_percent, 'k-', 'linewidth...
    , 4);
plot(J(PLOURDE_percent_pres_mode_equation_pos_83),...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
        PLOURDE_percent_pres_mode_equation_pos_83),'k<', '...
    linewidth', 2,...
    'Markersize', 14, 'MarkerFaceColor', 'c', '...
    MarkerEdgeColor', 'k', 'EraseMode', 'none');
205 plot(J(PLOURDE_percent_pres_mode_equation_pos_90),...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
        PLOURDE_percent_pres_mode_equation_pos_90),'ko', '...
    linewidth', 2,...

```

```

        'Markersize', 14, 'MarkerFaceColor', 'c', '...
        MarkerEdgeColor', 'k', 'EraseMode', 'none');
plot(J(PLOURDE_percent_pres_mode_equation_pos_95),...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
        PLOURDE_percent_pres_mode_equation_pos_95),'ks', '...
        linewidth', 2,...
210     'Markersize', 14, 'MarkerFaceColor', 'c', '...
        MarkerEdgeColor', 'k', 'EraseMode', 'none');
plot(J(PLOURDE_percent_pres_mode_equation_pos_96),...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
        PLOURDE_percent_pres_mode_equation_pos_96),'kd', '...
        linewidth', 2,...
        'Markersize', 14, 'MarkerFaceColor', 'c', '...
        MarkerEdgeColor', 'k', 'EraseMode', 'none');
215 plot(J(PLOURDE_percent_pres_mode_equation_pos_97),...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
        PLOURDE_percent_pres_mode_equation_pos_97),'k^', '...
        linewidth', 2,...
        'Markersize', 14, 'MarkerFaceColor', 'c', '...
        MarkerEdgeColor', 'k', 'EraseMode', 'none');
plot(J(PLOURDE_percent_pres_mode_equation_pos_98),...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
        PLOURDE_percent_pres_mode_equation_pos_98),'k>', '...
        linewidth', 2,...
        'Markersize', 14, 'MarkerFaceColor', 'c', '...
        MarkerEdgeColor', 'k', 'EraseMode', 'none');
220 plot(J(PLOURDE_percent_pres_mode_equation_pos_99),...
    cum_PTR_phstdev_pres_PLOURDE_percent(...
        PLOURDE_percent_pres_mode_equation_pos_99),'kv', '...
        linewidth', 2,...
        'Markersize', 14, 'MarkerFaceColor', 'c', '...
        MarkerEdgeColor', 'k', 'EraseMode', 'none');
    hold off;
    h_1 = legend('Normalized \sigma_{\phi} from thesis derivation...
        ,...
225 ['\iti\rm = ' ...
num2str(cor_Z_mode_PLOURDE_percent_pres_mode_equation_83,...
'%6.4g') ' modes @ ' ...
num2str(100*PLOURDE_percent_pres_mode_equation_value_83,...
'%6.4g') '% and ' ...
230 num2str(PLOURDE_actual_pres_mode_equation_value_83,...
'%6.4g') ' waves of total \sigma_{\phi_{PTR}} present'],...
['\iti\rm = ' ...
num2str(cor_Z_mode_PLOURDE_percent_pres_mode_equation_90,...
'%6.4g') ' modes @ ' ...
235 num2str(100*PLOURDE_percent_pres_mode_equation_value_90,...
'%6.4g') '% and ' ...
num2str(PLOURDE_actual_pres_mode_equation_value_90,...
'%6.4g') ' waves of total \sigma_{\phi_{PTR}} present'],...
['\iti\rm = ' ...
240 num2str(cor_Z_mode_PLOURDE_percent_pres_mode_equation_95,...
'%6.4g') ' modes @ ' ...

```

```

num2str(100*PLOURDE_percent_pres_mode_equation_value_95,...
'%6.4g') '% and ' ...
245 num2str(PLOURDE_actual_pres_mode_equation_value_95,...
'%6.4g') ' waves of total \sigma_{\phi_{PTR}} present'],...
['\iti\rm = ' ...
num2str(cor_Z_mode_PLOURDE_percent_pres_mode_equation_96,...
'%6.4g') ' modes @ ' ...
250 num2str(100*PLOURDE_percent_pres_mode_equation_value_96,...
'%6.4g') '% and ' ...
num2str(PLOURDE_actual_pres_mode_equation_value_96,...
'%6.4g') ' waves of total \sigma_{\phi_{PTR}} present'],...
['\iti\rm = ' ...
255 num2str(cor_Z_mode_PLOURDE_percent_pres_mode_equation_97,...
'%6.4g') ' modes @ ' ...
num2str(100*PLOURDE_percent_pres_mode_equation_value_97,...
'%6.4g') '% and ' ...
num2str(PLOURDE_actual_pres_mode_equation_value_97,...
'%6.4g') ' waves of total \sigma_{\phi_{PTR}} present'],...
260 ['\iti\rm = ' ...
num2str(cor_Z_mode_PLOURDE_percent_pres_mode_equation_98,...
'%6.4g') ' modes @ ' ...
num2str(100*PLOURDE_percent_pres_mode_equation_value_98,...
'%6.4g') '% and ' ...
265 num2str(PLOURDE_actual_pres_mode_equation_value_98,...
'%6.4g') ' waves of total \sigma_{\phi_{PTR}} present'],...
['\iti\rm = ' ...
num2str(cor_Z_mode_PLOURDE_percent_pres_mode_equation_99,...
'%6.4g') ' modes @ ' ...
270 num2str(100*PLOURDE_percent_pres_mode_equation_value_99,...
'%6.4g') '% and ' ...
num2str(PLOURDE_actual_pres_mode_equation_value_99,...
'%6.4g') ' waves of total \sigma_{\phi_{PTR}} present'],...
'location', 'E');
275 % h_t = title('Percentage of Normalized Phase Std Deviation ...
Present as a Function of Zernike Mode Index')
% 'Total Phase Std Deviation Expressed as a Fraction ...
of PTR '];...
% 'after including the first \iti\rm Zernike modes in ...
representing the atmosphere'; ' '];...
% ['For error from having finite modes = ' num2str...
(100*mode_percent_pres_THRESH,'%6.4g') '% of total Phase Std ...
Deviation Present']});
ax_left = gca;
280 xlim([xmin_modes xmax_modes2]); ylim([min(...
cum_PTR_phstdev_pres_PLOURDE_percent) max(...
cum_PTR_phstdev_pres_PLOURDE_percent)]);
h_y1 = ylabel({'Percentage of Total RMS PTR'; 'Phase ...
Standard Deviation Present'});
ax_right = axes('Position',get(ax_left,'Position'),'...
YAxisLocation','right','Color','none','YColor','k');

```

```

        xlim([xmin_modes xmax_modes2]); ylim([min(...
            cum_PTR_phstdev_pres_PLOURDE) max(...
            cum_PTR_phstdev_pres_PLOURDE)]);
        h_x2 = xlabel('Mode Index, \iti\rm');
285     h_y2 = ylabel({'Amount of RMS PTR'; 'Phase Standard ...
        Deviation Present'; '[waves]'});
        set(gca, 'FontName', 'Times New Roman');
        set([h_x2, h_y1, h_y2, h_l], 'FontName', 'Times New Roman'...
            , 'fontsize', 16);

290 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % PLOURDE Mode Simulation %
    % (numerical analysis of pixels as a %
    % function of normalized spatial frequency) %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
295 % Define the percentage threshold modes of interest
    thresh_modes = [cor_Z_mode_PLOURDE_percent_pres_mode_equation_90...
        ...
        cor_Z_mode_PLOURDE_percent_pres_mode_equation_95...
        ...
        cor_Z_mode_PLOURDE_percent_pres_mode_equation_96...
        ...
300     cor_Z_mode_PLOURDE_percent_pres_mode_equation_97...
        ...
        cor_Z_mode_PLOURDE_percent_pres_mode_equation_98...
        ...
        cor_Z_mode_PLOURDE_percent_pres_mode_equation_99]

    % Define the constants to be used in the equations below
305     C_sim = (0.2073/0.423)*2*pi*2^(-11/3);

    % Iterate over each threshold percentage level of interest
    for idx_thresh = 1:length(thresh_modes)

310     % Find the number of corresponding modes needed for the ...
        zernike_index_no_trig
        N_modes_trig(idx_thresh) = min(find(J_trig >= thresh_modes...
            (idx_thresh)));

        % Define Plourde Mode Simulation
        for idx2 = 1:N_modes_trig(idx_thresh);
315
            % Define the inside of the numeric integral
            if m_trig(idx2) == 0;
                SIM_mode_phvar_wh_ALL(idx2,:) = C_sim.*...
                    alpha_trig(idx2).*(besselj(nu_trig(idx2), ...
                        eta)).^2.*eta.^(-14/3);
            else

```

```

320         SIM_mode_phvar_wh_ALL(idx2,:) = 2*C_sim.*...
            alpha_trig(idx2).*(besselj(nu_trig(idx2), ...
            eta)).^2.*eta.^(-14/3);

            end

            % Integrate the above equation over eta and find ...
            numeric phvar & stdev
            SIM_mode_cumsum_phvar_wh_ALL(idx2,:) = cumsum(...
            SIM_mode_phvar_wh_ALL(idx2,:)).*eta_space; ...
            % [rad^2]

325     end

    % Set up an if statement to include D/r0 functions
    if include_Dr0
        % Find D/r0 weighted Simulated PTR phvar present in ...
        terms of spatial frequency after modes have been ...
        removed
330         SIM_PTR_phvar_pres_after_pixel(idx_thresh,:) = ...
            D_over_r0^(5/3).*sum(...
            SIM_mode_cumsum_phvar_wh_ALL);    % [rad^2]
    else
        % Find normalized Simulated PTR phvar present in terms...
        of spatial frequency after modes have been removed
        SIM_PTR_phvar_pres_after_pixel(idx_thresh,:) = sum...
            (SIM_mode_cumsum_phvar_wh_ALL);    % [rad^2]
    end

335     end

    % Convert to Simulated PTR Phase Std Dev present in terms of ...
    spatial frequency after modes have been removed
    SIM_PTR_stdev_pres_after_pixel(idx_thresh,:) = sqrt(...
        SIM_PTR_phvar_pres_after_pixel(idx_thresh,:))./(2*pi); ...
    % [waves]

    % Find maximum value of Simulated PTR phase STD Dev present ...
    after pixel have been removed
340     SIM_PTR_stdev_pres_after_pixel_value(idx_thresh,:) = max(...
        SIM_PTR_stdev_pres_after_pixel(idx_thresh,:)); % [waves...
    ]

    % Convert to percentage of total phase standard deviation ...
    present after pixelation effects
    SIM_PTR_stdev_pres_after_pixel_percent(idx_thresh,:) = ...
        SIM_PTR_stdev_pres_after_pixel(idx_thresh,:)./max(...
        SIM_PTR_stdev_pres_after_pixel(idx_thresh,:));

    end

345     % Find positions of and number of pixels associated with the ...
    various percentage threshold levels
    PLOURDE_percent_pres_simulation_pos_90 = min(find(...
        SIM_PTR_stdev_pres_after_pixel_percent(1,:) >= 0.949));

```

```

PLOURDE_percent_pres_simulation_value_90 = ...
    SIM_PTR_stdev_pres_after_pixel_percent(1, ...
    PLOURDE_percent_pres_simulation_pos_90);
PLOURDE_actual_pres_simulation_value_90 = ...
    SIM_PTR_stdev_pres_after_pixel(1, ...
    PLOURDE_percent_pres_simulation_pos_90);
350 cor_eta_PLOURDE_percent_pres_simulation_90 = eta(...
    PLOURDE_percent_pres_simulation_pos_90);
cor_pixels_PLOURDE_percent_pres_simulation_90 = ...
    D_over_delta_pix(PLOURDE_percent_pres_simulation_pos_90);

PLOURDE_percent_pres_simulation_pos_95 = min(find(...
    SIM_PTR_stdev_pres_after_pixel_percent(2,:) >= 0.975));
PLOURDE_percent_pres_simulation_value_95 = ...
    SIM_PTR_stdev_pres_after_pixel_percent(2, ...
    PLOURDE_percent_pres_simulation_pos_95);
355 PLOURDE_actual_pres_simulation_value_95 = ...
    SIM_PTR_stdev_pres_after_pixel(2, ...
    PLOURDE_percent_pres_simulation_pos_95);
cor_eta_PLOURDE_percent_pres_simulation_95 = eta(...
    PLOURDE_percent_pres_simulation_pos_95);
cor_pixels_PLOURDE_percent_pres_simulation_95 = ...
    D_over_delta_pix(PLOURDE_percent_pres_simulation_pos_95);

PLOURDE_percent_pres_simulation_pos_96 = min(find(...
    SIM_PTR_stdev_pres_after_pixel_percent(3,:) >= 0.980));
360 PLOURDE_percent_pres_simulation_value_96 = ...
    SIM_PTR_stdev_pres_after_pixel_percent(3, ...
    PLOURDE_percent_pres_simulation_pos_96);
PLOURDE_actual_pres_simulation_value_96 = ...
    SIM_PTR_stdev_pres_after_pixel(3, ...
    PLOURDE_percent_pres_simulation_pos_96);
cor_eta_PLOURDE_percent_pres_simulation_96 = eta(...
    PLOURDE_percent_pres_simulation_pos_96);
cor_pixels_PLOURDE_percent_pres_simulation_96 = ...
    D_over_delta_pix(PLOURDE_percent_pres_simulation_pos_96);

365 PLOURDE_percent_pres_simulation_pos_97 = min(find(...
    SIM_PTR_stdev_pres_after_pixel_percent(4,:) >= 0.985));
PLOURDE_percent_pres_simulation_value_97 = ...
    SIM_PTR_stdev_pres_after_pixel_percent(4, ...
    PLOURDE_percent_pres_simulation_pos_97);
PLOURDE_actual_pres_simulation_value_97 = ...
    SIM_PTR_stdev_pres_after_pixel(4, ...
    PLOURDE_percent_pres_simulation_pos_97);
cor_eta_PLOURDE_percent_pres_simulation_97 = eta(...
    PLOURDE_percent_pres_simulation_pos_97);
cor_pixels_PLOURDE_percent_pres_simulation_97 = ...
    D_over_delta_pix(PLOURDE_percent_pres_simulation_pos_97);
370 PLOURDE_percent_pres_simulation_pos_98 = min(find(...
    SIM_PTR_stdev_pres_after_pixel_percent(5,:) >= 0.990));

```

```

PLOURDE_percent_pres_simulation_value_98 = ...
    SIM_PTR_stdev_pres_after_pixel_percent(5, ...
    PLOURDE_percent_pres_simulation_pos_98);
PLOURDE_actual_pres_simulation_value_98 = ...
    SIM_PTR_stdev_pres_after_pixel(5, ...
    PLOURDE_percent_pres_simulation_pos_98);
cor_eta_PLOURDE_percent_pres_simulation_98 = eta(...
    PLOURDE_percent_pres_simulation_pos_98);
375 cor_pixels_PLOURDE_percent_pres_simulation_98 = ...
    D_over_delta_pix(PLOURDE_percent_pres_simulation_pos_98);

PLOURDE_percent_pres_simulation_pos_99 = min(find(...
    SIM_PTR_stdev_pres_after_pixel_percent(6,:) >= 0.995));
PLOURDE_percent_pres_simulation_value_99 = ...
    SIM_PTR_stdev_pres_after_pixel_percent(6, ...
    PLOURDE_percent_pres_simulation_pos_99);
PLOURDE_actual_pres_simulation_value_99 = ...
    SIM_PTR_stdev_pres_after_pixel(6, ...
    PLOURDE_percent_pres_simulation_pos_99);
380 cor_eta_PLOURDE_percent_pres_simulation_99 = eta(...
    PLOURDE_percent_pres_simulation_pos_99);
cor_pixels_PLOURDE_percent_pres_simulation_99 = ...
    D_over_delta_pix(PLOURDE_percent_pres_simulation_pos_99);

% Plot results of theoretical numerical analysis of phase standard...
% deviation a a function of the number of pixels on a device
figure(3); clf;
385 hold on;
plot(eta, SIM_PTR_stdev_pres_after_pixel_percent(1,:), 'k-', '...
    linewidth', 4);
plot(eta, SIM_PTR_stdev_pres_after_pixel_percent(2,:), 'k--', ...
    'linewidth', 4);
plot(eta, SIM_PTR_stdev_pres_after_pixel_percent(4,:), 'k-.', ...
    'linewidth', 4);
plot(eta, SIM_PTR_stdev_pres_after_pixel_percent(6,:), 'k:', '...
    linewidth', 4);
390 plot(eta(PLOURDE_percent_pres_simulation_pos_90),...
    SIM_PTR_stdev_pres_after_pixel_percent(1, ...
    PLOURDE_percent_pres_simulation_pos_90),'ko', '...
    linewidth', 2,...
    'Markersize', 14, 'MarkerFaceColor', 'c', '...
    MarkerEdgeColor', 'k', 'EraseMode', 'none');
plot(eta(PLOURDE_percent_pres_simulation_pos_95),...
    SIM_PTR_stdev_pres_after_pixel_percent(2, ...
    PLOURDE_percent_pres_simulation_pos_95),'ks', '...
    linewidth', 2,...
395 'Markersize', 14, 'MarkerFaceColor', 'c', '...
    MarkerEdgeColor', 'k', 'EraseMode', 'none');
plot(eta(PLOURDE_percent_pres_simulation_pos_97),...
    SIM_PTR_stdev_pres_after_pixel_percent(4, ...
    PLOURDE_percent_pres_simulation_pos_97),'k^', '...
    linewidth', 2,...

```

```

'Markersize', 14, 'MarkerFaceColor', 'c', '...
  MarkerEdgeColor', 'k', 'EraseMode', 'none');
plot(eta(PLOURDE_percent_pres_simulation_pos_99),...
400 SIM_PTR_stddev_pres_after_pixel_percent(6, ...
  PLOURDE_percent_pres_simulation_pos_99),'kv', '...
  linewidth', 2,...
'Markersize', 14, 'MarkerFaceColor', 'c', '...
  MarkerEdgeColor', 'k', 'EraseMode', 'none');
hold off;
h_1 = legend('Normalized \sigma_{\phi_{pixels}} as a function ...
of N pixels for 90% of the total \sigma_{\phi_{PTR}}',...
  'Normalized \sigma_{\phi_{pixels}} as a function ...
    of N pixels for 95% of the total \sigma_{\phi_{...
      PTR}}',...
405 'Normalized \sigma_{\phi_{pixels}} as a function ...
    of N pixels for 97% of the total \sigma_{\phi_{...
      PTR}}',...
  'Normalized \sigma_{\phi_{pixels}} as a function ...
    of N pixels for 99% of the total \sigma_{\phi_{...
      PTR}}',...
['\eta = ' num2str(...
  cor_eta_PLOURDE_percent_pres_simulation_90, '%6.4g'...
  )...
' & N = ' num2str(...
  cor_pixels_PLOURDE_percent_pres_simulation_90, '...
  %6.4g')...
' pixels @ ' num2str(100*...
  PLOURDE_percent_pres_simulation_value_90, '%6.4g')...
  ...
410 '% and ' num2str(...
  PLOURDE_actual_pres_simulation_value_90, '%6.4g') '...
  waves of total \sigma_{\phi_{PTR}} present'],...
['\eta = ' num2str(...
  cor_eta_PLOURDE_percent_pres_simulation_95, '...
  %6.4g')...
' & N = ' num2str(...
  cor_pixels_PLOURDE_percent_pres_simulation_95, ...
  '%6.4g')...
' pixels @ ' num2str(100*...
  PLOURDE_percent_pres_simulation_value_95, '%6.4...
  g')...
'% and ' num2str(...
  PLOURDE_actual_pres_simulation_value_95, '%6.4g...
  ') ' waves of total \sigma_{\phi_{PTR}} present...
  '],...
415 ['\eta = ' num2str(...
  cor_eta_PLOURDE_percent_pres_simulation_97, '%6.4g')...
  ...
' & N = ' num2str(...
  cor_pixels_PLOURDE_percent_pres_simulation_97, '...
  %6.4g')...

```

```

        ' pixels @ ' num2str(100*...
        PLOURDE_percent_pres_simulation_value_97, '%6.4g')...
        ...
    '% and ' num2str(...
    PLOURDE_actual_pres_simulation_value_97, '%6.4g') '...
    waves of total \sigma_{\phi_{PTR}} present'],...
    ['\eta = ' num2str(...
        cor_eta_PLOURDE_percent_pres_simulation_99, '...
        %6.4g')...
420     ' & N = ' num2str(...
        cor_pixels_PLOURDE_percent_pres_simulation_99, ...
        '%6.4g')...
        ' pixels @ ' num2str(100*...
        PLOURDE_percent_pres_simulation_value_99, '%6.4...
        g')...
        '% and ' num2str(...
        PLOURDE_actual_pres_simulation_value_99, '%6.4g...
        ') ' waves of total \sigma_{\phi_{PTR}} present...
        '],...
        'location', 'E');
%     h_t = title({'Percentage of Finite Mode Limited Normalized ...
Phase Std Deviation Present'; 'as a Function of Number of Pixels...
across a Wavefront Control Device';...
425 %         ' '; ' '; ' '});
%         axpos = get(gca, 'pos');
%         extent = get(h_t, 'extent');
%         set(gca, 'pos', [axpos(1) axpos(2) axpos(3) axpos(4) -(4/5)...
*extent(4)])
ax_left = gca;
430     xlim([0 xmax_eta]); ylim([min(min(...
        SIM_PTR_stdev_pres_after_pixel_percent)) max(max(...
        SIM_PTR_stdev_pres_after_pixel_percent))]);
    h_x1 = xlabel('Normalized Spatial Frequency, \eta');
    h_y1 = ylabel({'Percentage of Total RMS PTR'; 'Phase ...
        Standard Deviation Present'});
    ax_right = axes('Position', get(ax_left, 'Position'), '...
    XAxisLocation', 'top', 'YAxisLocation', 'right', 'Color', '...
    none', 'YColor', 'k');
    xlim([0 xmax_pixels]); ylim([min(min(...
        SIM_PTR_stdev_pres_after_pixel)) max(max(...
        SIM_PTR_stdev_pres_after_pixel))]);
435     h_x2 = xlabel({'Number of Pixels, N'});
    h_y2 = ylabel({'Amount of RMS PTR'; 'Phase Standard ...
        Deviation Present'; '[waves]'});
    set(gca, 'FontName', 'Times New Roman');
    set([h_x1, h_x2, h_y1, h_y2, h_1], 'FontName', 'Times New ...
        Roman', 'fontsize', 16);

440     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% COMPUTER SIMULATION RESULTS AND ANALYSIS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

445 % MANUALLY SET the FOLLOWING VALUES to correspond with desired ...
    threshold level
    % This is done for comparison with theoretical results
    SIM_PTR_stdev_pres_after_pixel_percent = ...
        SIM_PTR_stdev_pres_after_pixel_percent(2,:);
    PLOURDE_percent_pres_simulation_pos = ...
        PLOURDE_percent_pres_simulation_pos_95;
    PLOURDE_percent_pres_simulation_value = ...
        PLOURDE_percent_pres_simulation_value_95;
450 PLOURDE_actual_pres_simulation_value = ...
        PLOURDE_actual_pres_simulation_value_95;
    cor_eta_PLOURDE_percent_pres_simulation = ...
        cor_eta_PLOURDE_percent_pres_simulation_95;
    cor_Z_mode_PLOURDE_percent_pres_mode_equation = ...
        cor_Z_mode_PLOURDE_percent_pres_mode_equation_95;

    % Calculate values to use in plots
455 cor_pixels_PLOURDE_percent_pres_simulation = D_over_delta_pix(...
        PLOURDE_percent_pres_simulation_pos);

    % Load the necessary files created from the computer simulation
    load('Thesis\Matlab\mat Files\HO_computer_SIM_results.mat')

460 % Find the correct standard deviation values that match the number...
    of modes used for the threshold level of interest
    Computer_SIM_phstdev_pres_after_modes_and_pixel(2:11,2) = ...
        HO_norm_cum_phstdev_a_mode_ALL(:,...
        cor_Z_mode_PLOURDE_percent_pres_mode_equation-3);

    % Add a zero pixelation value that to provide more data points ...
    for the interpolation
    Computer_SIM_phstdev_pres_after_modes_and_pixel(1,2) = 0;

465 % Set up an if statement to include D/r0 functions
    if include_Dr0
        % Find D/r0 weighted Comp SIM PTR phstdev present in terms...
        of spatial frequency after modes have been removed
        comp_SIM_value_pres = D_over_r0^(5/6).*...
            Computer_SIM_phstdev_pres_after_modes_and_pixel...
            (:,2); % [waves]
470 else
        % Find normalized Comp SIM PTR phstdev present in terms of...
        spatial frequency after modes have been removed
        comp_SIM_value_pres = ...
            Computer_SIM_phstdev_pres_after_modes_and_pixel...
            (:,2); % [waves]
    end

475 % Create array of values so interpolation goes out to the correct ...
    number of data points

```

```

comp_SIM_pixels = ...
    Computer_SIM_phstdev_pres_after_modes_and_pixel(:,1);
comp_SIM_eta = (pi/2).*comp_SIM_pixels;

% Interpolate the amount present to work for all values of eta
480 comp_SIM_value_pres_INTERP = interp1(comp_SIM_eta(1:11), ...
    comp_SIM_value_pres(1:11), eta, 'cubic'); % [waves]

% Convert to percentage of total phase standard deviation present ...
    after pixelation effects
    comp_SIM_percent_pres = comp_SIM_value_pres./max(...
        comp_SIM_value_pres);
    comp_SIM_percent_pres_INTERP = comp_SIM_value_pres_INTERP./max...
        (comp_SIM_value_pres_INTERP);
485

% Find positions of and number of pixels associated with the ...
    various percentage threshold levels
    COMP_percent_pres_simulation_pos = min(find(...
        comp_SIM_percent_pres_INTERP >= pixel_percent_pres_THRESH))...
        ;
    COMP_percent_pres_simulation_value = ...
        comp_SIM_percent_pres_INTERP(...
            COMP_percent_pres_simulation_pos);
    COMP_actual_pres_simulation_value = comp_SIM_value_pres_INTERP...
        (COMP_percent_pres_simulation_pos);
490 cor_eta_COMP_percent_pres_simulation = eta(...
        COMP_percent_pres_simulation_pos);
    cor_pixels_COMP_percent_pres_simulation = D_over_delta_pix(...
        COMP_percent_pres_simulation_pos);

% Plot computer simulation vs. theoretical results for phase std ...
    dev present after pixelation effects as a function of the ...
    number of pixels on a device
    figure(4); clf;
495 hold on;
    plot(eta, SIM_PTR_stdev_pres_after_pixel_percent, 'k-', '...
        linewidth', 4);
        plot(eta, comp_SIM_percent_pres_INTERP, 'k--', '...
            linewidth', 4);
        plot(comp_SIM_eta, comp_SIM_percent_pres, 'ko',...
            'MarkerSize', 14, 'MarkerFaceColor', 'k', '...
                MarkerEdgeColor', 'k', 'EraseMode', 'none'...
            );
500 plot(eta(PLOURDE_percent_pres_simulation_pos),...
        SIM_PTR_stdev_pres_after_pixel_percent(...
            PLOURDE_percent_pres_simulation_pos),'ks'...
            , 'linewidth', 2,...
            'MarkerSize', 14, 'MarkerFaceColor', 'c', '...
                MarkerEdgeColor', 'k', 'EraseMode', 'none'...
            );
    plot(eta(COMP_percent_pres_simulation_pos),...

```

```

        comp_SIM_percent_pres_INTERP(...
            COMP_percent_pres_simulation_pos),'kd', '...
            linewidth', 2,...
505         'Markersize', 14, 'MarkerFaceColor', 'c', '...
            MarkerEdgeColor', 'k', 'EraseMode', 'none'...
        );

        hold off;
h_1 = legend('Theoretical normalized \sigma_{\phi} present...
',...
    'Cubic interpolation of computer simulated ...
    normalized \sigma_{\phi} present',...
    'Actual data values of computer simulated ...
    normalized \sigma_{\phi} present',...
510    ['\eta = ' num2str(...
        cor_eta_PLOURDE_percent_pres_simulation, '...
        %6.4g')...
    ' & N = ' num2str(...
        cor_pixels_PLOURDE_percent_pres_simulation...
        , '%6.4g')...
    ' pixels @ ' num2str(100*...
        PLOURDE_percent_pres_simulation_value, '...
        %6.4g')...
    '% and ' num2str(...
        PLOURDE_actual_pres_simulation_value, '%6.4...
        g') ' total phstd present'],...
    ['\eta = ' num2str(...
        cor_eta_COMP_percent_pres_simulation, '%6.4g')...
        ...
515    ' & N = ' num2str(...
        cor_pixels_COMP_percent_pres_simulation, '%6.4g...
        ')...
    ' pixels @ ' num2str(100*...
        COMP_percent_pres_simulation_value, '%6.4g')...
    '% and ' num2str(COMP_actual_pres_simulation_value...
        , '%6.4g') ' total phstd present'], 'location'...
        , 'E');

%         h_t = title({'Normalized phase standard deviation ...
present as a function of the number of pixels across a ...
wavefront control device'; ' ');...
%         'after only having 95% of the total \...
phi_{PTR} present due to finite Zernike modes and limited ...
spatial frequency';...
520 %         ' '; ' ');
%         axpos = get(gca,'pos');
%         extent = get(h_t,'extent');
%         set(gca,'pos',[axpos(1) axpos(2) axpos(3)
%         axpos(4)-(5/6)*extent(4)])
525 ax_left = gca;
        xlim([0 xmax_eta]); ylim([min(min(...
            SIM_PTR_stdev_pres_after_pixel_percent)) max(max(...
            SIM_PTR_stdev_pres_after_pixel_percent))]);
        h_x1 = xlabel('Normalized Spatial Frequency, \eta');

```

530

```

        h_y1 = ylabel({'Percentage of Total RMS'; 'Phase ...
            Standard Deviation Present'});
    ax_right = axes('Position',get(ax_left,'Position'),'...
        XAxisLocation', 'top', 'YAxisLocation', 'right', 'Color...
        ', 'none', 'YColor', 'k');
    xlim([0 xmax_pixels]); ylim([min(min(...
        SIM_PTR_stdev_pres_after_pixel)) max(max(...
        SIM_PTR_stdev_pres_after_pixel))]);
    h_x2 = xlabel('Number of Pixels, N');
    h_y2 = ylabel({'Amount of RMS PTR'; 'Phase Standard ...
        Deviation Present'; '[waves]'});
    set(gca, 'FontName'    , 'Times New Roman');
    set([h_x1, h_x2, h_y1, h_y2, h_l], 'FontName'    , 'Times ...
        New Roman', 'fontsize', 16);

```

Appendix B. Matlab Code to Create Phase Screens with the Fourier Series Method

Listing B.1: Matlab[®] file to create phase screens using the Fourier series method.

```
% Fourier Series Phase Screen Creator
clear all; clc;

% load path for locations of functions called
5   addpath('Thesis\Matlab');

% Define important parameters
numscreens = 250; % number of random phase screens to create
N = 512; % number of screen grid points
10  D = 2; % aperture diameter [m]
D_over_r0 = 5 % strength of turbulence to be created
    idxStr_Dr0 = sprintf('%02d', D_over_r0);

% Set the limits of integration
15  eta_min = eps;
    eta_max = 1e4;
    eta_numpts = 1e4;
    eta_space = (eta_max - eta_min)/eta_numpts;
    eta = linspace(eta_min, eta_max, eta_numpts);
20

% Calculate other important parameters
r0 = D./D_over_r0; % atmospheric coherence diameter [m]
deltax = D/N; % grid spacing
Fx = eta./(D*pi); % linear spatial frequency
25

% Create a circular aperture of unit radius
x = linspace(-D/2,D/2, N);
[x y] = meshgrid(x,x);
[theta rho] = cart2pol(x,y);
30  circ_ap = rho < D/2;
    r = (2.*rho)./D;

% Define the piston and tilt Zernike mo36des to be removed
Z_piston = 1;
35  Z_xtilt = 2.*r.*cos(theta);
    Z_ytilt = 2.*r.*sin(theta);

% PSD setup
T.lo = 0; % inner scale [m]
40  T.Lo = inf; % outer scale [m]
    T.Model = 'FHill'; % spectral model

T.fmin = min(Fx); % min spatial frequency [1/m]
T.fmax = max(Fx); % max spatial frequency [1/m]
45  T.N = 2^7; % number of FS coefficients
    T.r0 = r0;
```

```

% loop over realizations
phi = zeros(N, N);
50 for idx_num = 1 : numscreens
    idxStr_num = sprintf('%03d', idx_num);

    PSD = mkPSD_thesis(T);           % create the PSD
    C = FSCoeff_thesis(PSD);         % generate coefficients
55 x = (-N/2 : N/2-1) * deltax; % position coordinates [m]
    y = x;
    S = FSscrnREAL_thesis(x, y, C); % generate the phase ...
        screens
    phi(:, :) = S.g-mean(mean(S.g));

60 % Calculate zernike coefficients to for Piston, Tip, and ...
    Tilt
    a_piston = sum(sum(circ_ap.*phi(:, :).*Z_piston))./sum(...
        sum(circ_ap.*Z_piston.^2));
    a_xtilt = sum(sum(circ_ap.*phi(:, :).*Z_xtilt))./sum(...
        sum(circ_ap.*Z_xtilt.^2));
    a_ytilt = sum(sum(circ_ap.*phi(:, :).*Z_ytilt))./sum(...
        sum(circ_ap.*Z_ytilt.^2));

65 % Remove Piston, Tip, and Tilt from the phase screens
    phi_PTR(:, :) = phi(:, :) - (a_piston.*Z_piston) - (...
        a_xtilt.*Z_xtilt) - (a_ytilt.*Z_ytilt);

% Save important variables for future use
    save(['Thesis\Matlab\mat Files\thesis_Dr0_' idxStr_Dr0...
        '_screens_' idxStr_num '.mat'], ...
70 'phi', 'phi_PTR', 'D', 'D_over_r0', 'circ_ap', 'r'...
        , 'theta', 'numscreens');

end

```

Appendix C. Matlab Code to Pixelate and Compute Statistics of Phase Screens

Listing C.1: Matlab[®] file to down-sample and compute statistics of phase screen realizations.

```

% Fourier Series Phase Screen Evaluator
clear all; clc;
format compact;

5 % load path for locations of functions called
    addpath('Thesis\Matlab\Final Thesis Files');

% Define important parameters
    numscreens = 250; % number of random phase screens to create
10 N = 512; % number of screen grid points
    D = 2; % aperture diameter [m]
    D_over_r0 = 5 % strength of turbulence to be created
        idxStr_Dr0 = sprintf('%02d', D_over_r0);
    cor_Z_mode_PLOURDE_percent_pres_mode_equation = 139; % ...
        number of modes required for threshold level

15 % Create array of values for N_new
    pixels = [512 256 128 64 32 16 8 4 2 1];

% Create a circular aperture of unit radius
20 x = linspace(-D/2,D/2, N);
    [x y] = meshgrid(x,x);
    [theta rho] = cart2pol(x,y);
    circ_ap = rho < D/2;
        r = (2.*rho)./D;

25 % Pre-compute all of the Zernike polynomials using eval_zernike ...
    function
        for idx_mode = 4 : ...
            cor_Z_mode_PLOURDE_percent_pres_mode_equation
                Z_mode(:, :, idx_mode) = eval_zernike(idx_mode, r, theta);
                denom(idx_mode) = sum(sum(circ_ap.*Z_mode(:, :, idx_mode)...
                    .^2));
30 Z_mode(:, :, idx_mode) = circ_ap.*Z_mode(:, :, idx_mode);
        end

% Iterate over all pixelation levels
for idx_pixels = 1:length(pixels)
35 idxStr_pixels = sprintf('%03d', pixels(idx_pixels));
    iter = pixels(idx_pixels);
    idxStr_iter = sprintf('%03d', iter)

% Iterate over all random phase screens
40 for idx_screen = 1 : numscreens
    idxStr_screen = sprintf('%03d', idx_screen);

```

```

% Load each PTR phase screen realization
load(['Thesis\Matlab\mat Files\thesis_Dr0_' idxStr_Dr0...
    '_screens_' idxStr_screen '.mat'],...
45     'phi_PTR');
if pixels(idx_pixels) ~= 512;
    % Down-sample from N to N/div
    pix = pixels(idx_pixels) % how many pixels you ...
        want to end up in each row
    N = length(phi_PTR); % how many pixels you ...
        are starting with in each row
50     div = N/pix ; % the reduction in ...
        pixels

    for idx = 1:N/div; % create a matrix to ...
        pull from
        odd(idx) = count;
        count = count + div;
55     end

    for idx2 = 1:N/div
        for idx = 1:N/div
            odd_val(idx2,idx) = mean(mean(phi_PTR(odd(...
                idx2):odd(idx2)+(div-1),odd(idx):odd(...
                idx)+(div-1)))));
60

            % Create pixelated PTR phase screen ...
            realization
            phi_PTR_pixellated(odd(idx2):odd(idx2)...
                +(div-1), odd(idx):odd(idx)+(div-1)...
                ) = odd_val(idx2,idx);

            end
        end
65     end

% Calculate Zernike coefficients for the pixelated PTR ...
phase screen realization
for idx_mode = 4 : ...
    cor_Z_mode_PLOURDE_percent_pres_mode_equation
70     if pixels(idx_pixels) ~= 512;
        a_mode(idx_mode, idx_screen) = sum(sum(...
            phi_PTR_pixellated(:,:).*Z_mode(:,:, ...
            idx_mode)))./denom(idx_mode);

        else
            a_mode(idx_mode, idx_screen) = sum(sum(...
                phi_PTR(:,:).*Z_mode(:,:, idx_mode)))./...
                denom(idx_mode);

            end
        end
75     end

end

% Compute the statistics of the PTR phase screen realizations

```

```

% Find variance in each Zernike mode
phvar_a_mode = var(a_mode,0,2);
80
% Find HO total phase variance up to each Zernike mode
HO_cum_phvar_a_mode = cumsum(phvar_a_mode);

% Convert to phase standard deviation
85
HO_cum_phstdev_a_mode = sqrt(HO_cum_phvar_a_mode)...
./(2*pi);

% Remove r_0 dependence from the phase standard deviation
HO_norm_cum_phstdev_a_mode = HO_cum_phstdev_a_mode./...
D_over_r0.^(5/6);

90
% Find total HO phase std dev up to each Zernike mode for ...
each pixelation level
HO_norm_cum_phstdev_a_mode_ALL(idx_pixels+1,:) = ...
HO_norm_cum_phstdev_a_mode;

% Find maximum possible phase standard deviation for ...
simulation and show on main screen
phstdev_pres_after_modes_and_pixel = ...
HO_norm_cum_phstdev_a_mode(...
cor_Z_mode_PLOURDE_percent_pres_mode_equation)
95
total_phstdev_pres_after_pixel = max(...
HO_norm_cum_phstdev_a_mode)

% Create array of pixelation levels and the associated ...
simulation phase standard deviation values
Computer_SIM_phstdev_pres_after_modes_and_pixel(...
idx_pixels+2,1) = iter;
Computer_SIM_phstdev_pres_after_modes_and_pixel(...
idx_pixels+2,2) = ...
phstdev_pres_after_modes_and_pixel;
100

% Create sample PTR phase screen realizations for figures
phi_PTR_samp(:,:,idx_pixels+1) = phi_PTR;
phi_PTR_pixellated_samp(:,:,idx_pixels+1) = ...
phi_PTR_pixellated;

105
% Save important variables for future use
save('Thesis\Matlab\mat Files\HO_computer_SIM_results.mat'...
,...
'Computer_SIM_phstdev_pres_after_modes_and_pixel', '...
phi_PTR_samp',...
'phi_PTR_pixellated_samp', '...
HO_norm_cum_phstdev_a_mode_ALL')

end

```

Bibliography

1. Andrews, Larry C. and Ronald L. Phillips. *Laser Beam Propagation through Random Media*. SPIE Press, Bellingham, WA, second edition, 2005.
2. Arfken, George B. and Hans J. Weber. *Mathematical Methods for Physicists*. Academic Press, New York, NY, fourth edition, 1995.
3. Bagwell, Bret E., David V. Wick, Robert Batchko, Justin D. Mansell, Ty Martinez, Sergio R. Restaino, Don M. Payne, Jamie Harriman, Steve Serati, Gary Sharp, and Jim Schwiegerling. “Liquid Crystal Based Active Optics”. *SPIE*, 6289:0801–0812, 2006.
4. Bifano, Thomas G., Raji Krishnamoorthy Mali, John Kyle Dorton, Julie Perreault, Nelsimar Vandelli, Mark N. Horenstein, and David A. Castanon. “Continuous-membrane Surface-micromachined Silicon Deformable Mirror”. *Optical Engineering*, 36(5):1354–1360, May 1997.
5. Bifano, Thomas G., Julie Perreault, Raji Krishnamoorthy Mali, and Mark N. Horenstein. “Microelectromechanical Deformable Mirrors”. *IEEE Journal of Selected Topics in Quantum Electronics*, 5(1):83–89, Jan/Feb 1999.
6. Bold, G.T., T.H. Barnes, J. Gourlay, R.M. Sharples, and T.G. Haskell. “Practical Issues for the Use of Liquid Crystal Spatial Light Modulators in Adaptive Optics”. *Optics Communications*, 148:323–330, Mar 1998.
7. Brooks, Matthew R. and Matthew E. Goda. “Atmospheric Simulation Using a Liquid Crystal Wavefront Controlling Device”. *SPIE*, 5553:258–268, 2004.
8. Dayton, David, John Gonglewski, Sergio Restaino, Jeffrey Martin, James Phillips, Mary Hartman, Stephen Browne, Paul Kervin, Joshua Snodgrass, Nevin Heimann, Michael Shilko, Richard Pohle, Bill Carrion, Clint Smith, and Daniel Thiel. “Demonstration of New MEMS and Liquid Crystal Adaptive Optics on Bright Astronomical Objects and Satellites”. *Optics Express*, 10(25):1508–1519, Dec 2002.
9. Dayton, David C., Stephen L. Browne, Steven P. Sandven, John D. Gonglewski, and Alexis V. Kudryashov. “Theory and Laboratory Demonstrations on the Use of a Nematic Liquid-Crystal Phase Modulator for Controlled Turbulence Generation and Adaptive Optics”. *Applied Optics*, 37(13):5579–5589, Aug 1998.
10. Doble, Nathan and David R. Williams. “The Application of MEMS Technology for Adaptive Optics in Vision Science”. *IEEE Journal of Selected Topics in Quantum Electronics*, 10(3):629–635, May/Jun 2004.
11. Fernandez, Enrique J. and Pablo Artal. “Membrane Deformable Mirror for Adaptive Optics: Performance Limits in Visual Optics”. *Optics Express*, 11(9):1056–1069, May 2003.

12. Goodman, Joseph W. *Statistical Optics*. John Wiley & Sons, Inc., New York, NY, first edition, 1985.
13. Goodman, Joseph W. *Introduction to Fourier Optics*. McGraw Hill, New York, NY, second edition, 1996.
14. Hallstig, Emil, Lars Sjoqvist, and Mikael Lindgren. “Characterization of a Liquid Crystal Spatial Light Modulator for Beam Steering”. *SPIE*, 4632:187–196, 2002.
15. Hallstig, Emil, Johan Stigwall, Mikael Lindgren, and Lars Sjoqvist. “Laser Beam Steering and Tracking using a Liquid Crystal Spatial Light Modulator”. *SPIE*, 5087:13–23, 2003.
16. Hart, Nathaniel W., Michael C. Roggemann, Alex Sergeyev, and Timothy J. Schulz. “Characterizing Static Aberrations in Liquid Crystal Spatial Light Modulators Using Phase Retrieval”. *Optical Engineering*, 46(8):660101–660107, Aug 2007.
17. Lee, Max K., William D. Cowan, and Byron M. Welsh. “Aberration-correction Results from a Segmented Microelectromechanical Deformable Mirror and a Refractive Lenslet Array”. *Optics Letters*, 23(8):645–647, April 1998.
18. Linnenberger, Anna, Steve Serati, and Jay Stockley. “Advances in Optical Phased Array Technology”. *SPIE*, 6304:0T01–0T09, 2006.
19. Litvin, Igor A., Liesl Burger, Mapula P. De Gama, Ally Mathye, and Andrew Forbes. “Laser Beam Shaping Limitations for Laboratory Simulation of Turbulence Using a Phase-Only Spatial Light Modulator”. *SPIE*, 6663:0R01–0R07, 2007.
20. Love, Gordon D. “Wave-Front Correction and Production of Zernike Modes with a Liquid-Crystal Spatial Light Modulator”. *Applied Optics*, 36(7):1517–1524, Mar 1997.
21. Magee, Eric P. *Phase Screens for Long Time Series Wave Optics Simulations*. Technical report, Mission Research Corporation, May 2003.
22. Mantravadi, Samuel V., Troy A. Rhoadarmer, and Robert S. Glas. “Simple Laboratory System for Generating Well-Controlled Atmospheric-Like Turbulence”. *SPIE*, 5553:290–300, 2004.
23. Miller, Linda M., Michael L. Agronin, Randall K. Bartman, William J. Kaiser, Thomas W. Kenny, Robert L. Norton, and Erika C. Vote. “Fabrication and Characterization of a Micromachined Deformable Mirror for Adaptive Optics Applications”. *SPIE*, 1945:421–430, 1993.
24. Naumov, A.F. and G.D. Love. “Perspectives of Liquid Crystals for Adaptive Optics Applications”. *SPIE*, 4124:57–65, 2000.
25. Noll, Robert J. “Zernike Polynomials and Atmospheric Turbulence”. *J. Optical Society of America*, 66(3):207–211, Mar 1976.

26. Perreault, Julie, Thomas G. Bifano, B. Martin Levine, and Mark N. Horenstein. "Adaptive Optic Correction Using Microelectromechanical Deformable Mirrors". *Optical Engineering*, 41(3):561–566, Mar 2002.
27. Phillips, James D., Matthew E. Goda, and Jason D. Schmidt. "Atmospheric Turbulence Simulation Using Liquid Crystal Spatial Light Modulators". *SPIE*, 5894:0601–0611, 2005.
28. Poularikas, Alexander D. *The Handbook of Formulas and Tables for Signal Processing*. CRC Press LLC, Boca Raton, FL, 1999.
29. Restaino, Sergio R., Jeff T. Baker, and Don Payne. "New Liquid Crystal Devices for Adaptive Optics". *SPIE*, 4493:41–45, 2002.
30. Rhoadarmer, Troy A. and Roger P. Angel. "Low-Cost, Broadband Static Phase Plate for Generating Atmosphericlike Turbulence". *Applied Optics*, 40(18):2946–2955, Jun 2001.
31. Roggemann, Michael C. and Byron M. Welsh. *Imaging Through Turbulence*. CRC Press, Inc., New York, NY, first edition, 1996.
32. Sasiela, Richard J. *Electromagnetic Wave Propagation in Turbulence, Evaluation and Application of Mellin Transforms*. Springer-Verlag, New York, NY, first edition, 1994.
33. Schmidt, Jason D. *Free-Space Optical Communications Performance Enhancement by Use of a Single Adaptive Optics Correcting Element*. Ph.D. thesis, University of Dayton, 2006.
34. Schmidt, Jason D., Matthew E. Goda, and Bradley D. Duncan. "Emulating Bulk Turbulence with a Liquid-Crystal Spatial Light Modulator". *SPIE*, 6306:0O01–0O12, 2006.
35. Schmidt, Jason D., Matthew E. Goda, and Bradley D. Duncan. "Aberration Production Using a High-Resolution Liquid-Crystal Spatial Light Modulator". *Applied Optics*, 46(13):2423–2433, May 2007.
36. Serati, Steve, Xiaowei Xia, Owais Mughal, and Anna Linnenberger. "High-Resolution Phase-Only Spatial Light Modulators With Sub-Millisecond Response". *SPIE*, 5106:138–144, 2003.

Vita

Michael Dominic Plourde was born at Chanute AFB, Illinois and grew up at various Air Force Bases throughout the contiguous United States. He graduated as valedictorian of 250 students at South River High School in Edgewater, Maryland. He earned a Bachelor of Science degree in Electrical Engineering from the University of South Florida in Tampa, Florida where he also completed R.O.T.C. training and was commissioned as a Second Lieutenant in the United States Air Force. He then worked as an electronic countermeasures engineer at the Sensors Directorate of the Air Force Research Laboratory at Wright-Patterson AFB, Ohio, where he was promoted to First Lieutenant. From there, he was competitively selected to attend the Air Force Institute of Technology where he pursued a Master of Science degree in Electrical Engineering with a concentration in Electro-Optics and was promoted to the rank of Captain. Upon completion of his degree, he will apply his newfound knowledge while working on the Airborne Laser Program at Kirtland AFB, New Mexico.

Permanent address: 2950 Hobson Way
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

REPORT DOCUMENTATION PAGE					Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE (DD-MM-YYYY) 27-03-2008		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) Aug 2006 — Mar 2008		
4. TITLE AND SUBTITLE Limitations of Segmented Wavefront Control Devices in Emulating Optical Turbulence				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Plourde, Michael D., Captain, USAF				5d. PROJECT NUMBER ENG 08-295		
				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 DSN:785-3636				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GEO/ENG/08-02		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/RDS (AFMC) Darryl Sanchez, PhD 3550 Aberdeen Ave SE Kirtland Air Force Base, NM 87117-5776 505-846-7209 (DSN:246-7209), email: darryl.sanchez@kirtland.af.mil				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 Using a device to act as a surrogate for atmospheric turbulence in a laboratory is necessary to build and test optical systems for imaging, lidar, laser weapons, and laser communications. Liquid-crystal spatial light modulators (LC SLMs) and segmented micro-electro-mechanical-system (MEMS) deformable mirrors (DMs) are commonly used devices for altering wavefronts in order to simulate a portion of atmospheric turbulence. The best location of these devices was theoretically analyzed to obtain the broadest possible range of atmospheric conditions. It was revealed that a layered model for atmospheric turbulence strength can be represented by the atmospheric coherence diameter that a single segmented wavefront control device can produce. The limitations of pixelation effects on a segmented wavefront control device were investigated theoretically. The results of this analysis were then confirmed in simulation. The performance capabilities of several available commercial devices are better understood as a result of this research.						
15. SUBJECT TERMS Adaptive Optics, Optical Atmospheric Turbulence, Deformable Mirror, Spatial Light Modulator						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jason D. Schmidt, Capt, USAF (ENG)	
U	U	U	UU	111	19b. TELEPHONE NUMBER (include area code) 937-255-3636 x7224, jason.schmidt@afit.edu	