

Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

3-2010

Phase Unwrapping in the Presence of Strong Turbulence

Casey J. Pellizzari

Follow this and additional works at: <https://scholar.afit.edu/etd>



Part of the [Signal Processing Commons](#)

Recommended Citation

Pellizzari, Casey J., "Phase Unwrapping in the Presence of Strong Turbulence" (2010). *Theses and Dissertations*. 1760.

<https://scholar.afit.edu/etd/1760>

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.



**PHASE UNWRAPPING
IN THE PRESENCE OF
STRONG TURBULENCE**

THESIS

Casey J. Pellizzari, Captain, USAF
AFIT/GE/ENG/10-23

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS 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.

AFIT/GE/ENG/10-23

PHASE UNWRAPPING
IN THE PRESENCE OF
STRONG 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

Casey J. Pellizzari, BSEE
Captain, USAF

March 2010

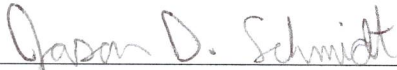
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AFIT/GE/ENG/10-23

PHASE UNWRAPPING
IN THE PRESENCE OF
STRONG TURBULENCE

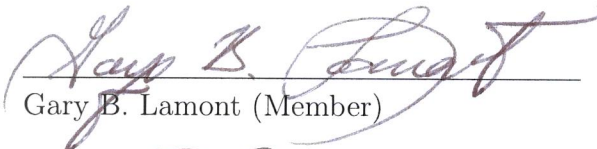
Casey J. Pellizzari, BSEE
Captain, USAF

Approved:




Jason D. Schmidt (Chairman)

18 Feb 10
Date



Gary B. Lamont (Member)

18 FEB 2010
Date



William P. Baker (Member)

18 Feb 2010
Date

Abstract

Phase unwrapping in the presence of branch points using a least-squares wavefront reconstructor requires the use of a Postprocessing Congruence Operation (PCO). This ensures that the unwrapped output is congruent to the wrapped input. 2π discontinuities known as branch cuts in the unwrapped phase are altered by the addition of a constant parameter h to the rotational component when applying the PCO. Past research has shown that selecting a value of h to minimize the proportion of irradiance in the pupil plane adjacent to branch cuts helps to maximize performance of adaptive-optics (AO) systems in strong turbulence. Recent non-optimal implementations of the PCO have accomplished this in part. In continuation of this objective, this research focuses on optimizing the PCO while accounting for the cumulative effects of the integral control law to improve AO performance in strong turbulence. Several optimizations are developed and compared in closed-loop AO using wave-optics simulations. The most successful optimization is shown to significantly reduce the normalized variance of the Strehl ratio across a wide range of turbulence strengths and frame rates, including decreases of up to 25 percent when compared to a non-optimized PCO algorithm. AO systems which depend on high, steady Strehl ratio values serve to benefit from these algorithms when operating in the presence of branch points.

Acknowledgements

I would like to sincerely thank Maj Jason Schmidt for his support which was critical in the development of this research. Maj Schmidt always exhibited a level of dedication and willingness to assist me unparalleled by his peers. I can only hope that some of his professionalism and expert research methodologies have rubbed off on me. I would also like to thank my research committee members Dr. Gary Lamont and Dr. William Baker for taking me on as a student even though the topic was not in their area of expertise. The time they sacrificed to attend my research briefings and provide feedback was extremely valuable and allowed me to obtain outside perspectives on the problem. In addition, I would like to thank the OPTECS research group for their mentorship along the way, and for setting the bar high. Finally, I am most grateful for my wife who has sacrificed just as much as I have so that I might be successful in my academics.

Casey J. Pellizzari

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
List of Figures	viii
List of Tables	xi
I. Introduction	1
1.1 Problem Statement and Hypothesis	2
1.2 Thesis Overview	3
II. Background and Related Research	5
2.1 Conventional Adaptive Optics	5
2.1.1 Atmospheric Turbulence	5
2.1.2 Wave-front Measurement	11
2.1.3 Wave-front Correction	16
2.1.4 Wave-front Reconstruction	18
2.1.5 Wrapped Phase	20
2.1.6 AO Control	23
2.1.7 Metrics of Performance	25
2.2 Branch Points and Branch Cuts	27
2.2.1 Cause and Behavior of Branch Points and Cuts	27
2.2.2 Problem in Adaptive Optics	30
2.2.3 Phase Decomposition	32
2.2.4 Branch-Point Detection	35
2.3 Unwrapping Methods	37
2.3.1 Path-Following Algorithms	39
2.3.2 Regional Algorithms	41
2.3.3 Global Algorithms	42
2.3.4 Hybrid Algorithms	53
2.4 Chapter Summary	55
III. Simulation Environment and Algorithm Design	56
3.1 Simulation Environment	56
3.1.1 Atmosphere	56

	Page
3.1.2 AO System	64
3.1.3 Parameter Exploration	68
3.2 Algorithm Design	81
3.2.1 Optimization Attempts	81
3.2.2 Key Algorithms	85
3.2.3 Algorithm Comparison	91
3.3 Chapter Summary	95
IV. Results and Analysis	96
4.1 IWCL	96
4.2 Frame-to-Frame variations in optimal values of h	98
4.3 Strehl Ratio	100
4.3.1 Frame Rate	102
4.3.2 Turbulence Strength	105
4.3.3 Benign and Challenging Conditions	106
4.3.4 Strehl Ratio CDF	107
4.4 Chapter Summary	108
V. Conclusions and Recommendations	110
5.1 Summary	110
5.2 Challenges Overcome	111
5.3 Key Results	113
5.4 Recommendations	115
5.5 Chapter Summary	117
A. Appendix A - MATLAB [®] Code	118
Bibliography	140
Vita	144

List of Figures

Figure		Page
1.	Plane wave incident upon atmosphere	6
2.	Shack-Hartman lenslet array	12
3.	Relationship between tilt and beam displacement	14
4.	Self Referencing Interferometer	14
5.	Phase conjugation with a deformable mirror	17
6.	Behavior of coupled actuators.	17
7.	Kolmogorov phase Power Spectral Density model	19
8.	Fried reconstruction geometry	19
9.	Wrapped and unwrapped phase	21
10.	\tan^{-1} behavior	21
11.	Open- and closed-loop adaptive optics	24
12.	Two-dimensional view of branch points and cuts	29
13.	Three-dimensional view of branch points and cuts	29
14.	Path-dependent unwrapping	31
15.	Unwrapped phase with errors due to branch points	31
16.	Gradient plots of a rotational and irrotational phase	34
17.	Example of Helmholtz decomposition	34
18.	Circulation method of branch-point detection	36
19.	Rotation of the gradient for branch-point detection	38
20.	Alternate method of branch-point detection	38
21.	Example of poorly placed branch cuts	41
22.	Periodicity used for a Fast Fourier Transform algorithm	45

Figure	Page
23.	Complex-exponential reconstruction grid reduction 49
24.	Non-LS phase altered by PCO 54
25.	Example Kolmogorov phase screen 58
26.	Simulation geometry 58
27.	Bandlimited point source for computer simulation 59
28.	Sampling analysis for computer simulation 61
29.	Example of point-source propagation in vacuum 62
30.	Example of point-source propagation in turbulence 62
31.	Structure function validation of simulations 63
32.	PSF validation of simulations 63
33.	MTF validation of simulations 64
34.	Simulation architecture for closed-loop observations 65
35.	AO system alignment 67
36.	Spatial filtering and down-sampling of SRI output 69
37.	Simulation architecture for open-loop observations 70
38.	Strehl ratio and IWCL vs. h 70
39.	Open-loop PDF for optimal h 72
40.	Optimal h vs. time 72
41.	Example of IWCL vs. h for closed-loop simulations 73
42.	Simulation parameter space 74
43.	Closed-loop PDF's for optimal h by turbulence strength 75
44.	IWCL vs. h for weak turbulence 76
45.	Closed-loop PDF's for optimal h by frame rate 78

Figure	Page
48. Effects of sudden changes in h	80
49. Phase-value histograms and IWCL	82
50. Distribution of <i>seeds</i> for Opt6	86
51. Additional examples of IWCL vs. h for closed-loop simulations	89
52. Golden Ratio Search	90
53. IWCL difference between LSPV+1 and LSPV+200	99
54. Typical Strehl ratio results for Xphase	103
55. Importance of IWCL as a function of frame rate	105
56. Maximizing correlation as a function of frame rate	106
57. Comparison of CDF's for Xphase, Opt2, and Opt6	109
58. Comparison of CDF's for PCO algorithms	109

List of Tables

Table		Page
1.	Atmospheric parameters being used for computer simulation	60
2.	Descriptions of optimized algorithms.	83
3.	Descriptions of optimized algorithms (cont.).	84
4.	Reduction of mean IWCL	99
5.	Average variances of h_{diff}	101
6.	Normalized Strehl Ratio Variance	103

PHASE UNWRAPPING
IN THE PRESENCE OF
STRONG TURBULENCE

I. Introduction

On 31 Jan. 2010, the Missile Defense Agency (MDA) conducted a test which exposed vulnerabilities in the Ground-Based Midcourse Defense (GMD) element of the U.S. Ballistic Missile Defense System (BMDS). A ground-based kill vehicle launched from Vandenberg Air Force Base failed to intercept and destroy a target missile launched from the Marshall Islands [30]. As illustrated by the test, intercepting a missile moving at speeds of approximately 4,000 mph is extremely difficult. Senior leaders have called upon directed energy (DE) weapons as a way to revolutionize military engagements such as missile defense [46]. Three days after the failed MDA test, the value of DE weapons was demonstrated when the Airborne Laser (ABL) successfully engaged and destroyed a missile while still in boost phase [25]. The speed-of-light delivery (670 million mph) and precision capabilities of DE weapons make them ideal for such operational scenarios. Unfortunately, high-energy laser weapons such as the one used by ABL interact with the turbulent atmosphere, significantly reducing power on target. Overcoming the challenges of strong atmospheric turbulence during horizontal propagation (HP) continues to be a key area of interest within the DE community and supports high-level doctrine. Joint Vision 2020 states “The joint force of 2020 must be prepared to “win” across the full range of military operations...” [1]. In fiscal year (FY) 2007, the Scientific Advisory Board issued a HP challenge to advance long-range, near-horizontal path atmospheric compensation ca-

pabilities. Adaptive optics (AO) systems are critical for HP in these most challenging conditions.

Wave-front sensing is an important part of AO. An interferometric wave-front sensor like the self-referencing interferometer (SRI) can be used to measure the real and imaginary parts of a complex field to compute the principal value of the phase. Prior to being applied to a deformable mirror (DM), the measured phase must be unwrapped. AO systems using conventional unwrappers perform adequately when the measured phase of a distorted optical field is continuous across the aperture. However, when an optical field is propagated through strong turbulence such as that encountered by ABL or future tactical laser systems, nulls in amplitude occur, leading to branch points in the phase [15]. It has been shown that branch points and the 2π phase discontinuities known as branch cuts connecting them, can degrade the performance of AO systems using a least-squares (LS) wave-front reconstructor [15]. This is due to the fact that a LS unwrapping algorithm cannot detect branch points and branch cuts. Therefore, it produces a reconstructed phase which is missing information and is not congruent (modulo- 2π -equivalent) to the wrapped input [36]. A Postprocessing Congruence Operation (PCO) can be applied to the output of the LS reconstructor to produce an unwrapped phase congruent to the input [17]. For each phase that must be unwrapped, the PCO can produce different solutions depending on the parameters being used. This research focuses on real-time optimization of the PCO in an attempt to maximize AO performance.

1.1 Problem Statement and Hypothesis

Department of Defense (DoD) applications of AO require its use in all conditions, including strong turbulence. Traditional phase-unwrapping techniques fail under these circumstances leading to the requirement for alternative methods. Use of a

LS unwrapper with a PCO can improve system performance in strong turbulence if the PCO parameter is optimized in real time to minimize IWCL as well as mitigate the effects of the integral-control law. The goal of this research is to develop an optimal phase-unwrapping algorithm using a LS unwrapper with a PCO to improve the performance of an AO system when compared to conventional unwrapping techniques in strong turbulence. Specifically, the objectives include:

1. validate the correlation between Strehl ratio, a metric for AO performance, and Irradiance Weighted Cut Length (IWCL), a measure of irradiance adjacent to branch cuts,
2. determine the statistical relationship between the PCO parameter value and IWCL,
3. determine effects of integral-control law when using a PCO unwrapper,
4. develop an unwrapping algorithm which utilizes the above relationship to maximize Strehl ratio mean while minimizing its variance, and
5. use wave-optics simulations to compare IWCL and Strehl ratio performances of the algorithm to conventional methods on a closed-loop AO system.

1.2 Thesis Overview

Chapter II provides an introduction to AO, phase unwrapping, and branch points and cuts. Several branch-point-tolerant-phase unwrappers proposed in current literature are also discussed. The simulation environment used to meet the objectives of Sec. 1.2, along with the design of key PCO optimizations are presented in Ch. III. Chapter IV discusses the results and analysis of the simulations developed in Ch. III. Finally, Ch. V summarizes the efforts of this research and highlights the challenges

overcome, key results, and the contributions and new knowledge gained. In addition, it presents ideas for future efforts intended to continue research of phase unwrapping and AO compensation in the presence of strong turbulence.

II. Background and Related Research

This chapter introduces the basic concepts and components of conventional AO systems. Branch-point and branch-cut theory is discussed along with its application to AO. Various existing unwrapping methods are described including path-following, regional, global, and hybrid algorithms. This information provides the background necessary to develop and compare effective phase-unwrapping algorithms.

2.1 Conventional Adaptive Optics

2.1.1 Atmospheric Turbulence.

Temperature and density fluctuations in the atmosphere lead to unstable air masses called eddies, which occur in a continuum of sizes [2]. The change in index of refraction from one eddy to another causes light to bend along a given path. Wavefronts passing through this inhomogeneous medium are distorted unevenly across the beam. Figure 1 shows how a plane wave is altered by the atmosphere. Since AO systems attempt to correct these distortions, it is necessary to model atmospheric behavior prior to the design of such a system.

Based primarily on physical insight, Kolmogorov laid the foundation of turbulence theory by analyzing the velocity fluctuations in the atmosphere [2]. The idea of scales is devised to categorize eddies by their size. The inertial subrange refers to the range of eddy sizes in which fully developed turbulent flow takes place, l_0 being the smallest and L_0 the largest. By assuming that eddies in the inertial subrange are locally statistically homogeneous and isotropic, Kolmogorov was able to develop a statistical model of the turbulent flow velocity in the form of a structure function. Because turbulent flow is not a stationary random process, the covariance is not a meaningful quantity. Rather, for a process that has stationary increments, the structure function

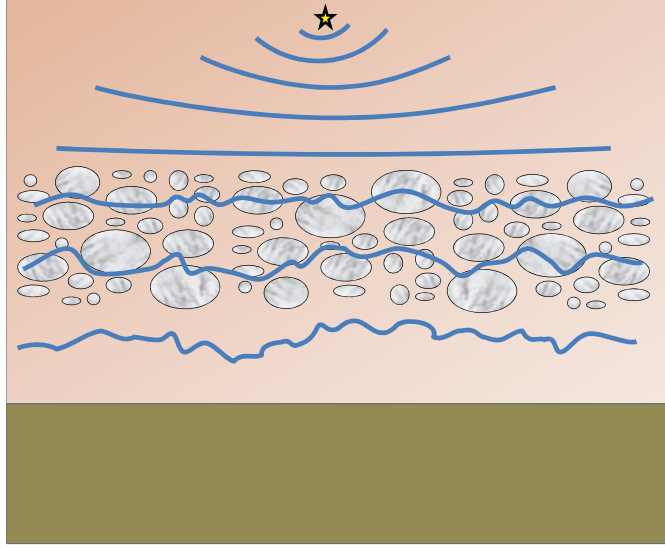


Figure 1. After propagating from a distant source, the wave-front incident upon the atmosphere is nearly planar.

is more appropriate [2]. For the spatially random process $x(\mathbf{R})$, the structure function is defined as

$$D_x(\mathbf{R}_1, \mathbf{R}_2) = \langle [x(\mathbf{R}_1) - x(\mathbf{R}_2)]^2 \rangle, \quad (1)$$

where $\langle . \rangle$ represents the ensemble average. Obukhov (and independently Corrsin) were able to relate Kolmogorov's turbulent flow model to temperature fluctuations, from which the index-of-refraction fluctuations could be determined [45]. For statistically homogeneous and isotropic turbulence, the index of refraction structure function is

$$D_n(R) = \begin{cases} C_n^2 l_0^{-4/3} R^2, & 0 \leq R \ll l_0, \\ C_n^2 R^{2/3}, & l_0 \ll R \ll L_0, \end{cases} \quad (2)$$

where R is the scalar separation between two points and C_n^2 is the index of refraction structure constant in units of $\text{m}^{-2/3}$ [2]. Several C_n^2 models have been developed which provide measures of the turbulence strength based primarily on altitude. The Power Spectral Density (PSD) shows how the power of a random process is distributed with respect to frequency. From the structure function, the three-dimensional PSD of the

atmosphere Φ_n can be found based on the relationship

$$\Phi_n(\kappa) = \frac{1}{4\pi^2\kappa^2} \int_0^\infty \frac{\sin(\kappa R)}{\kappa R} \frac{d}{dR} \left[R^2 \frac{d}{dR} D_n(R) \right] dR, \quad (3)$$

where κ is the angular spatial frequency [2]. Within the inertial subrange, the index-of-refraction PSD given by Eq. (3) evaluates to

$$\Phi_n(\kappa) = 0.033C_n^2\kappa^{-11/3}, \quad 1/L_0 \ll \kappa \ll 1/l_0. \quad (4)$$

To determine the PSD outside this range, one can assume $L_0 = \infty$ and $l_0 = 0$ or use a more sophisticated spectrum model such as Tatarskii, von Kármán, or the *modified atmospheric spectrum*. These models account for physics outside the inertial subrange.

To relate index of refraction fluctuations to optical field fluctuations, one must start with the governing partial differential equation for a scalar optical field in a vacuum given by

$$\nabla^2 U_0 + k^2 U_0 = 0, \quad (5)$$

where ∇^2 represents the Laplacian operator, U_0 is the field, and $k = 2\pi/\lambda$ is the optical wave number wavelength λ . When the optical wave propagates through a random medium, the stochastic wave equation is needed, which is given by

$$\nabla^2 U + k^2 n^2(\mathbf{R}) U = 0, \quad (6)$$

where $n(\mathbf{R})$ is the index of refraction as a function of position [45]. The index of refraction can be written as $n(\mathbf{R}) = 1 + n_1(\mathbf{R})$, where 1 is the index of refraction in a vacuum, and $n_1(\mathbf{R})$ is the perturbation from the vacuum case [2]. In the case of weak turbulence, the assumption $|n_1(\mathbf{R})| \ll 1$ can be used to approximate $n^2(\mathbf{R}) \approx$

$1 + 2n_1(\mathbf{R})$ in Eq. (6), giving

$$\{\nabla^2 + k^2 [1 + 2n_1(\mathbf{R})]\}U(\mathbf{R}) = 0. \quad (7)$$

In weak turbulence $1 + 2n_1(\mathbf{R}) \approx 1$, making Eq. (7) very close to Eq. (5). This indicates that the effects on the field due to the index of refraction perturbations are only slightly different from the vacuum case. To compute statistical moments of the field, an approximate solution to Eq. (7) is needed. One such method is the Rytov approximation which assumes the solution has the form

$$U(\mathbf{R}) = U(\mathbf{r}, L) = U_0(\mathbf{r}, L) \exp[\psi(\mathbf{r}, L)], \quad (8)$$

where ψ is a complex phase perturbation due to turbulence, U_0 is the vacuum field, and \mathbf{r} and L are, respectively, the cylindrical coordinates for radial location and propagation distance [2]. The complex phase perturbation takes the form

$$\psi(\mathbf{r}, L) = \psi_1(\mathbf{r}, L) + \psi_2(\mathbf{r}, L) + \dots, \quad (9)$$

where ψ_1 and ψ_2 are the first- and second-order complex phase perturbations, respectively. Using this solution, statistical moments of the perturbation can be computed. Finally, the method of cumulants can be used to compute moments of the field. The second-order moment of optical field U at propagation distance L , given for points \mathbf{r}_1 and \mathbf{r}_2 , is known as the mutual coherence function (MCF) represented by [2]

$$\Gamma_{12}(\mathbf{r}_1, \mathbf{r}_2, L) = \langle U(\mathbf{r}_1, L) U^*(\mathbf{r}_2, L) \rangle. \quad (10)$$

The MCF is a measure of the spatial coherence of two points in the observation plane. When the two points are at the same location, they are perfectly correlated. As they

are moved apart, the degree of correlation is reduced and the source is said to have limited spatial coherence [18]. This limited spatial coherence is caused by both the propagation geometry and the atmospheric turbulence. It can be computed from the modulus of the normalized MCF known as the *degree of coherence* (DOC), given by

$$\gamma(\mathbf{r}_1, \mathbf{r}_2, L) = \frac{|\Gamma_{12}(\mathbf{r}_1, \mathbf{r}_2, L)|}{\sqrt{\Gamma_{12}(\mathbf{r}_1, \mathbf{r}_1, L) \Gamma_{12}(\mathbf{r}_2, \mathbf{r}_2, L)}} \quad (11)$$

$$= \exp\left[-\frac{1}{2}D(\mathbf{r}_1, \mathbf{r}_2, L)\right], \quad (12)$$

where $D(\mathbf{r}_1, \mathbf{r}_2, L)$ is the wave structure function (WSF) [2]. For the case of a spherical wave, the WSF in polar coordinates is given by

$$D_{sp}(\rho, L) = 8\pi^2 k^2 \int_0^L \int_0^\infty \kappa \Phi_n(\kappa) \left\{1 - J_0\left[\left(1 - \frac{z}{L}\right) \kappa \rho\right]\right\} d\kappa dz, \quad (13)$$

where ρ is the radius from the axis of propagation, L is the propagation distance, $k = 2\pi/\lambda$ is the wave number given, κ is the spatial frequency, z is the position along the propagation axis, and J_0 is a Bessel function of the first kind, order zero [2]. Using the von Kármán spectrum, Eq. (13) reduces to

$$D_{sp}(\rho, L) = \begin{cases} 1.09 C_n^2 k^2 L l_0^{-1/3} \rho^2 [1 - 0.72 (\kappa_0 l_0)^{1/3}], & \rho \ll l_0, \\ 1.09 C_n^2 k^2 L \rho^{5/3} [1 - 0.72 (\kappa_0 \rho)^{1/3}], & \rho \gg l_0, \end{cases} \quad (14)$$

where C_n^2 is assumed to remain constant along the propagation path.

The point at which the DOC reduces to $1/e$ is known as the spatial coherence radius ρ_0 . Setting Eq. (11) equal to $1/e$ and assuming an infinite outer scale, the

spherical wave coherence radius ρ_{sp} can be found to be

$$\rho_{sp} = \begin{cases} \left(0.55C_n^2k^2Ll_0^{-1/3}\right)^{-1/2}, & \rho_{sp} \ll l_0, \\ (0.55C_n^2k^2L)^{-3/5}, & l_0 \ll \rho_{sp} \ll L_0. \end{cases} \quad (15)$$

Apertures with a radius larger than ρ_0 experience a breakdown in the coherence of the light [47]. Fried's parameter, or spatial coherence diameter r_0 is more commonly used, and in this context $r_0 = 2.1\rho_0$. When imaging with a telescope, increasing the aperture diameter larger than r_0 produces a minimal gain in resolution [41].

Since the atmosphere is dynamically changing, it is necessary to quantify this rate of change when designing AO systems. The Greenwood frequency is a measure of this change and is given by

$$f_G = 0.2549 \left[k^2 \int_0^L C_n^2(z)v^{5/3}(z)dz \right]^{3/5}, \quad (16)$$

where v is the velocity perpendicular to the optical axis [44]. It is defined as the AO system control bandwidth at which the residual phase variance is one radian squared.

Finally, in strong turbulence the propagated beam suffers amplitude fluctuations which lead to branch points. A measurement of amplitude fluctuation (scintillation) over a given path is called the log-amplitude variance and is given by [2]

$$\sigma_\chi^2 = 0.5361k^{7/6}L^{5/6} \int_0^L C_n^2\left(\frac{z}{L}\right)^{5/6} \left(1 - \frac{z}{L}\right)^{5/6} dz, \quad (17)$$

which is often referred to as the Rytov number \mathcal{R} . It is important to note that the *Rytov approximation* is only valid in weak turbulence [44]. Rytov theory predicts that amplitude fluctuations increase with increasing turbulence. In practice, the amplitude fluctuations saturate in strong turbulence. A more sophisticated propagation theory is needed in this region, however Rytov theory does typically predict phase disturbances

accurately enough to solve practical problems [44]. Even though the Rytov number is not equivalent to log-amplitude variance in strong turbulence, it is a very commonly used measure of turbulence strength.

Atmospheric turbulence is often considered to be a linear system in practical applications, and when working with linear systems it is useful to observe the impulse response. In optics, the spatial impulse response is known as the point spread function (PSF) [19]. The PSF determines how well a system (may include the turbulence) can image a point source. Ideally it would be represented by a Dirac delta function in the image plane. However, due to the limits of diffraction and the distortion caused by the atmosphere, the PSF has finite width. The frequency response of the system can be analyzed by considering the modulus of the PSF's Fourier transform, known as the modulation transfer function (MTF). The long-exposure MTF for atmospheric effects in the image plane of a lens is given by

$$\mathcal{H}_{LE}(\nu) = \exp \left[-3.44 \left(\frac{\lambda f \nu}{r_0} \right)^{5/3} \right], \quad (18)$$

where ν is the spatial frequency, λ is the wavelength of the optical field, f is the focal length of the lens, and r_0 is the spatial coherence diameter [18]. The term “long exposure” indicates that the integration time being used to obtain the image is long compared to the speed of fluctuations in the field caused by the atmosphere. Knowledge of the atmospheric PSF and MTF allows turbulence effects to easily be considered as a single element in a linear system [2].

2.1.2 Wave-front Measurement.

Wave-front measurement refers to determining the phase distortions on a nearly collimated beam at the exit pupil of a telescope. The direct sinusoidal oscillations of optical waves are too rapid to measure directly [47]. Only the squared magnitude of

the field known as the irradiance can be observed over many oscillations of the wave field. There have been many methods devised and implemented to recover phase information from a beam's irradiance. This section discusses only two such methods, an indirect method and a direct method. An indirect measurement determines the wave-front phase by measuring some related attribute such as localized tilt or slope of an incoming wave-front and from that computes the phase. As indicated in the name, a direct measurement measures principal value of the phase directly from the irradiance with almost no additional computation [16].

2.1.2.1 Shack-Hartmann Wave-front Sensor.

A Shack-Hartman (SH) Wave-front Sensor (WFS) is an indirect wave-front measurement device. It divides the incident wave-front into sections by passing it through an array of subapertures. Each subapertures contains a lens which focuses that portion of the beam on a focal-plane array as depicted in Fig. 2.

If the light passing through a subaperture is tilted, it focuses to a spot off axis. A SH WFS then must determine how far the spot has shifted. A typical example of this process is called centroiding [11]. Usually, this is accomplished using a center-of-mass

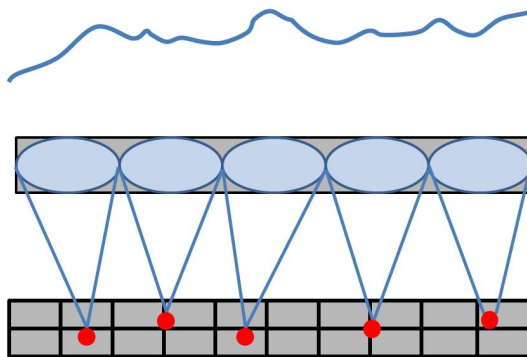


Figure 2. Shack-Hartman (SH) wave-front Sensor (WFS) lenslet array.

estimator. The displacements in both the x and y directions are given by

$$S_x = \frac{\sum \sum x_i I(x_i, y_j)}{\sum \sum I(x_i, y_j)}, \quad S_y = \frac{\sum \sum y_j I(x_i, y_j)}{\sum \sum I(x_i, y_j)}, \quad (19)$$

where $I(x_i, y_j)$ is the measured irradiance at pixel (x_i, y_j) [11]. Other methods of determining displacement of the focal spot include correlation and quad-cell measurements.

Once the displacement has been determined, it must be related to the slope of the incident wave-front. Figure 3 shows the relationship between a tilted wave-front and the corresponding focal point displacement.

The angle of incidence θ_i and the associated phase due the tilt ϕ_{tilt} are given by

$$\theta_i = \tan^{-1} \left(\frac{S_{x,y}}{f} \right), \quad (20)$$

$$\phi_{tilt} = \frac{kD}{4} \theta_i, \quad (21)$$

where f is the focal length of the lens being used in the subaperture, k is the wave number, and D is the diameter of the lens. Once ϕ_{tilt} is known across each subaperture in both the x and y directions, a reconstruction algorithm can integrate them together to form a single wave-front which is then used to control the DM. Advantages of using a SH WFS include a large dynamic range over which distortions can be measured, low sensitivity to chromaticity, and a high Technology Readiness Level (TRL). Disadvantages include sensitivity to scintillation and high propagation of noise [11].

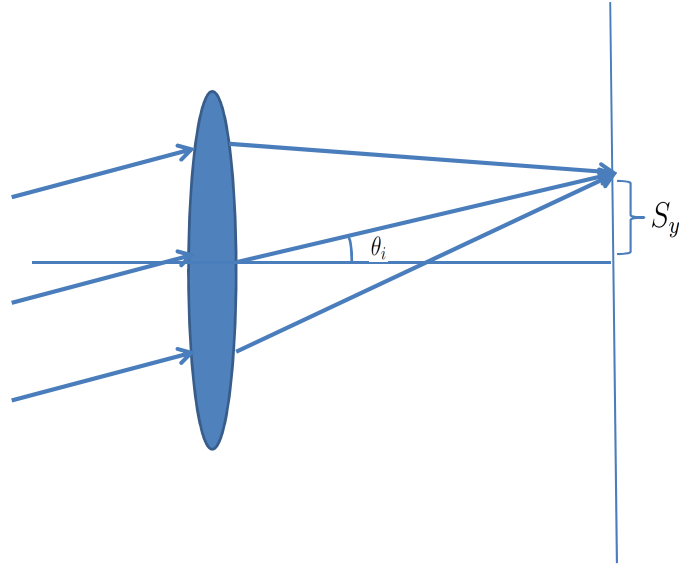


Figure 3. Relationship between tilt of incident beam and displacement of focal spot.

2.1.2.2 Self Referencing Interferometer.

A SRI directly measures the phase of an incident wave-front. The term “self referencing” comes from the fact that an SRI splits light in two, and spatially filters one leg. This creates a plane-wave reference from which the distortions in the other beam can be compared via an interference pattern, hence the “interferometer” in the name. Figure 4 shows how this is accomplished.

Once split, the incident wave-front is focused down into a single-mode fiber. Only

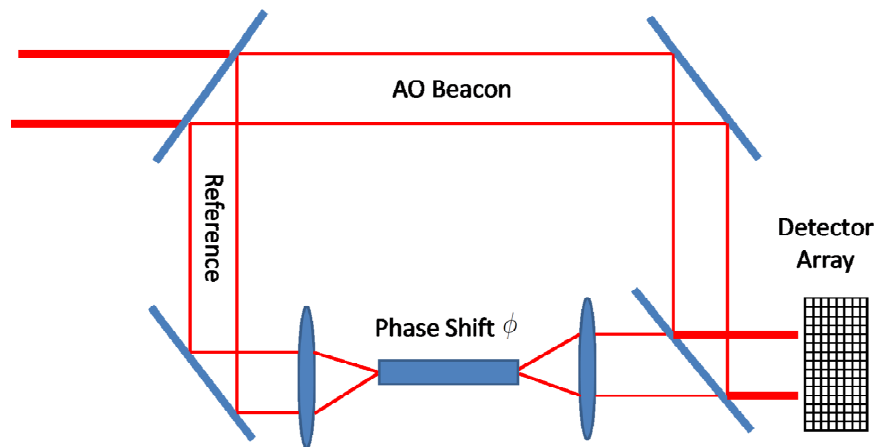


Figure 4. Self Referencing Interferometer

the low-frequency content near the optical axis is coupled into the fiber. This corresponds to the DC component. Once in the fiber, the optical wave is phase shifted. The phase-shifted beam emerges from the fiber with a spherical phase which is then collimated by a lens. It then interferes with the original distorted wave-front. By the principle of superposition, the total field U_t incident on the detector is given by

$$U_t = U_i + U_r = A_i e^{j\phi_i} + A_r e^{j\phi_r}, \quad (22)$$

where the subscript r represents the reference plane wave and the subscript i represents the incident distorted wave-front. In Eq. (22), A is the amplitude of each beam, ϕ_i is the phase of U_i , and ϕ_r is the phase of U_r after a phase shift has been applied in the fiber. The total irradiance I_t , which is what can be physically measured, is given by

$$I_t = U_t U_t^* = (A_i e^{j\phi_i} + A_r e^{j\phi_r}) (A_i e^{-j\phi_i} + A_r e^{-j\phi_r}), \quad (23)$$

$$I_t = A_i^2 + A_r^2 + A_i A_r e^{j(\phi_i - \phi_r)} + A_i A_r e^{-j(\phi_i - \phi_r)}, \quad (24)$$

$$I_t = A_i^2 + A_r^2 + 2A_i A_r \cos(\phi_i - \phi_r), \quad (25)$$

where U_t^* is the complex conjugate of U_t and A_i and A_r are the field amplitudes of the incident and reference beams, respectively. The purpose of balancing the split between the signal and reference legs is to ensure that $A_i = A_r = A$. When this is done, Eq. (25) simplifies to

$$I_t = 2A^2 [1 + \cos(\phi_i - \phi_r)]. \quad (26)$$

When $\phi_i - \phi_r = n2\pi$ ($n = 0, 1, 2, \dots$), the irradiance is maximized. For $\phi_i - \phi_r = 2(n + 1/2)\pi$ ($n = 0, 1, 2, \dots$), the irradiance is zero. Typically, an SRI measures the irradiance for four different values of ϕ_r . This can be done sequentially with one split

or on four different regions of the same camera simultaneously with several splits. From the four interferograms I_1 , I_2 , I_3 , and I_4 , corresponding to $\phi_r = 0, \pi/2, \pi$, and $3\pi/2$, respectively, ϕ_w the principal value of ϕ_i is determined by

$$\phi_w = \text{Tan}^{-1}(I_4 - I_2, I_1 - I_3), \quad (27)$$

where $\text{Tan}^{-1}(y, x)$ is a four-quadrant inverse tangent operator that returns the principal value in the range $(-\pi, \pi]$. The process of unwrapping ϕ_w is discussed in Sec. 2.1.5. A major benefit of using an SRI is its theoretical immunity to scintillation [40]. Since it measures phase directly, it is less susceptible to nulls in the irradiance which significantly degrade the ability of a SH WFS to measure subaperture tilts. The drawbacks of using an SRI are that the dynamic range that can be measured is much smaller than the SH WFS and the SRI is outperformed by the SH WFS at low resolution in weak turbulence.

2.1.3 Wave-front Correction.

Typical AO systems apply corrections to an aberrated wave-front by rapidly adjusting the figure of a mirror. Figure 5 shows how a mirror can apply a conjugate shape to an incident wave-front, resulting in the reflection of a flat wave.

There are many types of DM's that are used in AO. Segmented mirrors are made of many small adjacent mirrors working independently. Actuators are used to raise or lower segments, and some mirrors can even tip and tilt the segments. Continuous face-sheet mirrors are more common and are made of one single mirrored surface with actuators attached to the back. Since all the actuators are connected to the same continuous face-sheet, their movements are coupled. The position of one actuator can effect the shape of the mirror at the actuators nearby as shown in Fig. 6. The influence function describes this interdependency between actuators, and determines

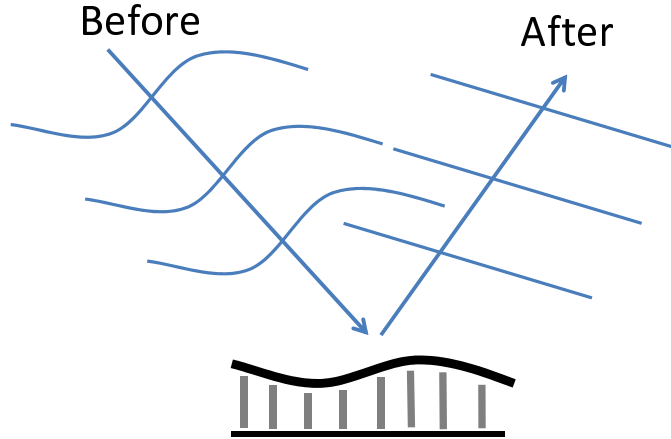


Figure 5. Phase conjugation with a Deformable Mirror (DM)

how well a DM can compensate an aberrated wave-front [21]. Assuming that the influence functions are independent allows the DM surface to be treated as a linear sum of influence functions, which can be used to predict performance [33]. Correcting a high-power laser requires a thick face sheet. This means that there is more coupling and the mirror's ability to compensate is decreased in areas where there are sharp changes in the phase.

With any type of deformable mirror, the dynamic range (stroke limit) of the actuators is important. It limits the magnitude of corrections that can be applied. To help illustrate the problem, Fig. 7 shows the Kolmogorov phase PSD (spatial frequencies). It can be seen that most of the power lies in low-frequency aberrations. One such aberration is tilt. This means that a DM might use much of its actuator throw correcting tilt, which leaves little available for the higher-order aberrations. To take this burden off of the DM, many AO configurations correct for low- and high-

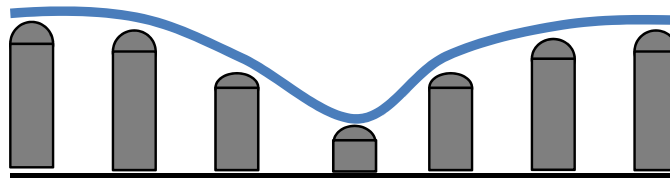


Figure 6. Behavior of coupled actuators. The center actuator is fully pulled down which pulls down its neighbors.

order aberrations separately. A typical example is the use of a Fast Steering Mirror (FSM). A separate wave-front sensor can be used to monitor the tilt across the entire incoming beam, just as a SH WFS does for small sections of a wave-front. The data are then fed into a dedicated control system which controls the angular tilt of the FSM, thereby removing much of the aberrations from the wave-front.

2.1.4 Wave-front Reconstruction.

Wave-front reconstruction is required with indirect wave-front sensing and consists of transforming the output of the sensor into commands for the DM in a timely and accurate manner. Measurements from each type of WFS are different, but in most cases, the output can not directly command a DM. Zonal reconstruction involves relating the geometry of the WFS measurement to the geometry of the actuators. Typically, wave-front gradients are integrated, resulting in a large system of linear equations. It is generally necessary to have more wave-front gradient measurements than actuators to command for solving an overdetermined system of equations [47]. One very common geometry for this type of reconstruction is the Fried geometry as shown for a single subaperture in Fig. 8. The slopes are measured in the x and y directions across the subaperture. These slopes are then translated into phase estimates at the corners, which are used to control actuators at the corresponding locations.

The measured slopes are related to the phase by

$$\mathbf{s} = \mathbf{G}\boldsymbol{\phi} \tag{28}$$

where \mathbf{s} is a vector containing the x- and y-slopes, $\boldsymbol{\phi}$ is the phase vector containing commands for the DM, and \mathbf{G} is the geometry matrix [34]. We need to solve for the phases $\boldsymbol{\phi}$, so that $\boldsymbol{\phi} = \mathbf{G}\mathbf{s}$. There are many ways to compute \mathbf{G} , such as

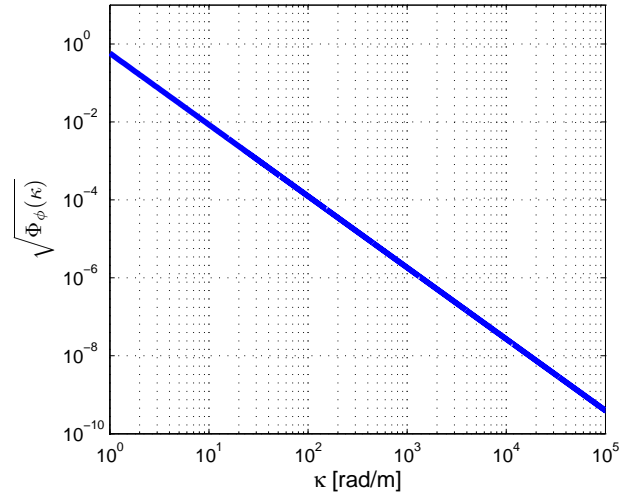


Figure 7. Kolmogorov phase Power Spectral Density (PSD) model. The symbol κ represents the radial component of the angular spatial frequency.

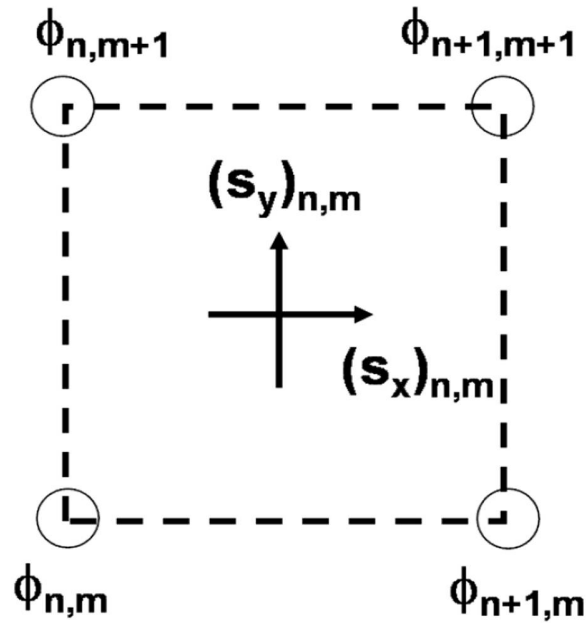


Figure 8. Fried reconstruction geometry. S_y and S_x are the slopes at location (n,m) .

minimum-variance and maximum *a posteriori* methods [41]. Since the system is over-determined, a pseudo-inverse of \mathbf{G} can be used to solve for ϕ , which is one of the simplest methods. Given the large number of measurements and unknowns (sometimes over 1000), this can be a computationally intensive task which must execute very quickly. Since \mathbf{G} is sparse, computing the pseudo-inverse is manageable with fast methods like sparse Cholesky decomposition. [12].

2.1.5 Wrapped Phase.

Direct wave-front sensing results in wrapped phase which is confined to the range $(-\pi, \pi]$. An example is the SRI output given by Eq. (27). Wrapping is a non-linear process which essentially adds $2n\pi$, where n is an integer, to each point in the phase so that the resulting value lies in the range $(-\pi, \pi]$ [17]. Figure 9 shows an example of how a continuous phase maps to its wrapped form.

Wrapped phase ϕ_w is given by

$$\psi = \text{Arg}(e^{i\phi}), \quad (29)$$

where ϕ is unwrapped phase. The right hand side of Eq. (29) is also known as the wrapping operator $W(\phi)$. Substituting $U = \exp(i\phi)$ into Eq. (29) gives

$$\phi_w = \text{Tan}^{-1}[\text{Im}(U), \text{Re}(U)], \quad (30)$$

where $\text{Im}(U)$ and $\text{Re}(U)$ are the imaginary and real parts, respectively of field U . The phase resulting from the wrapping operation is called the principal value. As shown in Fig. 10, when the Tan^{-1} is taken for an angle just before the third quadrant and just after, there is a 2π difference. It is this discontinuity that manifests itself as wrapping cuts across a wrapped phase, as shown in Fig. 9.

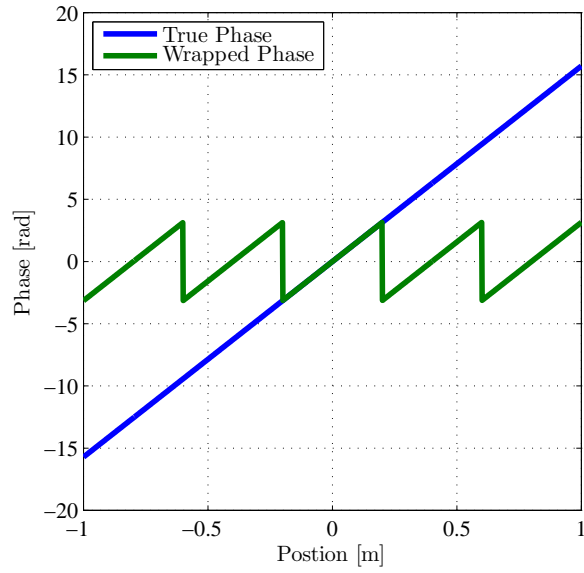


Figure 9. Example of wrapped and unwrapped phase for a tilted wave-front.

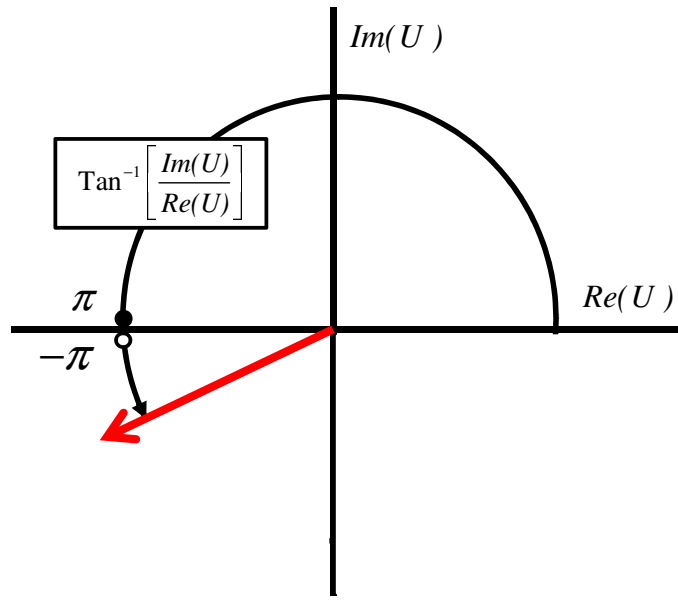


Figure 10. Tan^{-1} behavior when crossing between second and third quadrant.

An SRI directly outputs wrapped phase, while phase gradients measured by a SH WFS must be wrapped prior to reconstruction. In either case, when reconstructing a wave-front, one must unwrap the phase before applying commands to a DM. There are many methods of unwrapping depending on the application. For example, in calendar year 2000, over 200 journal articles had been published on two dimensional phase unwrapping [20]. All methods are based on integrating the gradients across the aperture [17]. This requires that any integration path taken between two points result in the same value which is known as path independence. If the integral around every simple closed-loop contour is zero:

$$\oint \varphi(\mathbf{r}) d\mathbf{r} = 0, \quad (31)$$

where $\varphi(\mathbf{r})$ is the phase evaluated at \mathbf{r} , then path independence exists [17]. Under normal circumstances this is not a problem. However, branch points present problems for this approach. Different integration paths result in different unwrapped phases. Section 2.2.2 discusses this in more detail.

Another problem to consider when unwrapping is the effect of noise. The relationship between the phase due to noise prior to unwrapping and that after unwrapping is highly non-linear [4]. Balmer *et al.* showed that when tilt is present, estimating the gradient by wrapping adjacent phase differences in the presence of noise violates Eq. (31). Some unwrapping methods assume noise to be a zero-mean, Gaussian process, however if the gradient estimate is wrapped, the noise term no longer has a mean of zero. The Probability Density Function (PDF) of the true phase difference between adjacent samples is Gaussian, but with a mean equal to their difference [4].

2.1.6 AO Control.

In an AO system, the link between the sensor and the corrector is the controller. AO utilizes closed-loop control for stable performance in the presence of stochastic aberrations and noise. In an open-loop system, the WFS measures the light entering prior to correction. Conversely, a closed-loop configuration senses light after it has been corrected by the DM. This creates a feedback loop which provides the system with a way of measuring how well it is performing. Figure 11 provides a diagram of each concept.

The distorted input beam reflects off the DM which applies a correction to it. In closed-loop AO, a WFS senses the residual distortions in the beam after the DM has applied a correction which is also called the error. The computer then reconstructs the wave-front and applies new commands to the DM in an attempt to make the wave-front as flat as possible. The DM commands at time step t_k are given by

$$\phi_{DM}(t_k) = \alpha\phi_{DM}(t_{k-1}) - \beta\phi_{error}(t_{k-1}), \quad (32)$$

which are a weighted combination of the previous commands $\phi_{DM}(t_{k-1})$ as well as the wave-front reconstructed from the error $\phi_{error}(t_{k-1})$. Equation (32) is referred to as a proportional-plus-integral (PI) control law. The proportionality of new to previous commands can be altered by adjusting the gain parameters α and β until an optimal, stable performance is reached. The parameter α controls how much of the previous commands are reused in the next frame, and is typically equal to one. The parameter β determines how much of the error is compensated in each frame, with typical values of 0.3 or 0.4. A low β value causes the system to be slow to reduce error caused by changes in the atmosphere, while a system with a high β is quick to respond to new changes but may lead to instabilities caused by being over reactive. A careful balance

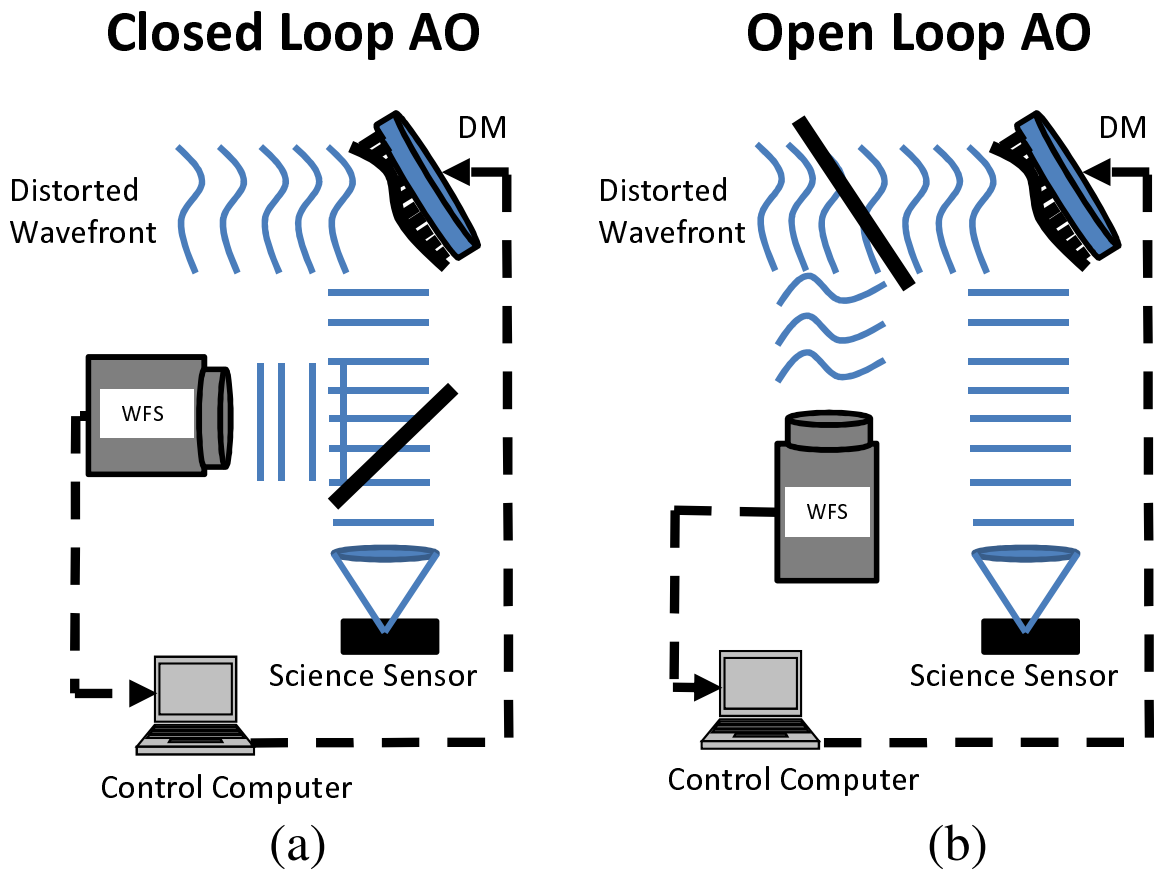


Figure 11. Examples of (a) open- and (b) closed-loop high-order adaptive optics.

of the gain parameters must be obtained to optimize performance. The rate at which the system as a whole can perform this loop is known as its frame rate. Integration time on the WFS, complexity of reconstruction, computational speed, and actuator response can all affect the frame rate. Typically though, the WFS integration time has the largest impact on frame rate.

2.1.7 Metrics of Performance.

Measuring the residual phase after compensation is the most direct indication of performance for an AO system. Ideal compensation would perfectly conjugate an aberrated beam, resulting in a flat wave-front. Any variance in the compensated wave-front is undesirable. Wave-front variance can be computed by

$$\sigma^2 = \frac{\int \int P(x, y) [\phi(x, y) - \phi_m(x, y)]^2 dx dy}{\int \int P(x, y) dx dy} \quad (33)$$

where $\phi(x, y)$ is the wave-front phase, $\phi_m(x, y)$ is its mean, and $P(x, y)$ is the pupil function which represents the aperture [47]. The *wave-front error* is defined as the square root of the wave-front variance. The on-axis peak irradiance for an un-aberrated optical field originating from a point source and passing through an aperture of diameter D , calculated in the region of Fraunhofer diffraction (far field) is given by

$$I_0 = P \frac{\pi D^2}{4\lambda^2 R^2}, \quad (34)$$

where P is the optical power in watts passing through the aperture, λ is the wavelength, and R is the radius of the converging spherical wave at the aperture measured from the image plane. [47; 6]. When a small wave-front error is present (about 1/6

wave) the on-axis irradiance can be given by

$$I = P \frac{\pi D^2}{4\lambda^2 R^2} \left[1 - \left(\frac{2\pi}{\lambda} \right)^2 \sigma^2 \right]. \quad (35)$$

From Eq. (35), it can be seen that the terms in brackets represent the ratio of the reduction in irradiance from an increase in wave-front error. For this reason it is called the Strehl ratio and is one of the most common measures of performance for imaging-systems. Since the Strehl ratio is equal to the first two terms in the expansion of $\exp(-x^2)$, it is often approximated by [47]

$$S \approx \exp \left[- \left(\frac{2\pi\sigma}{\lambda} \right)^2 \right]. \quad (36)$$

When no aberrations are present, $S = 1$. As the wave-front variance increases, the Strehl approaches zero. The field-estimation Strehl ratio refers to a convenient method of computing Strehl ratio which does not require propagation to a focal plane, given by

$$S = \frac{|\int \int U(x, y) dx dy|^2}{A \int \int |U(x, y)|^2 dx dy}, \quad (37)$$

where $U(x, y)$ is the field across the aperture area A . When working with a discretely sampled field, Eq. (37) can be computed by

$$S = \frac{|\text{mean}[U(x, y)]|^2}{\text{mean}[|U(x, y)|^2]}, \quad (38)$$

Phase-unwrapping algorithms can be compared examining their effect on Strehl ratio under identical circumstances in a closed-loop AO system. It is desired to have a high, steady Strehl ratio without large, sudden drops in value. One way to measure how well an AO system does this is to examine the Strehl ratio variance, normalized by the mean squared. This normalized variance is used in this research to compare

algorithms.

2.2 Branch Points and Branch Cuts

As mentioned in Sec. 2.1.5, most AO systems require phase to be unwrapped for controlling continuous DM's. Branch points and branch cuts present difficulties in unwrapping and degrade performance. For this reason, there has been a significant amount of research focused on mitigating their effects. This section discusses the cause and behavior of branch points and cuts, their relevance to AO, and finally how to detect them.

2.2.1 Cause and Behavior of Branch Points and Cuts.

Strong turbulence causes amplitude fluctuations characterized by a large Rytov number. As a result, the distorted wave has nulls in its irradiance [14]. It has been shown that a zero-amplitude point in the observed field forces the phase to be a non-single-valued function [50]. This means that when integrating the phase gradient around an arbitrarily small circuit in a counter-clockwise direction centered on a branch point, a non-zero value results. The sign of this closed-contour integral is called the branch point's charge. A branch point must be connected to an oppositely charged branch point by a 2π discontinuity called a branch cut [14]. Branch cuts are artificially determined lines where the discontinuity has been forced to reside which compensates for the non-zero curl of the phase slope around branch points [49]. The discontinuity of the cut is the opposite 2π multiple encountered from the branch point. Also, cuts may connect a branch point to another branch point outside the aperture, as shown in Fig. 12. In this case, a branch cut begins at the branch point and connects to the edge of the aperture. In the three-dimensional view shown in Fig. 13, the discontinuities of the branch cut can be observed. It has been shown

that the location of branch points are fixed but the cuts which connect them can be altered [14]. Properly placed branch cuts serve as barriers which prevent unwrapping paths from crossing, thereby allowing path-independent unwrapping [23; 43].

Branch points are also referred to as residues in an analogy to the residues from complex-variable contour integration, although they are not quite identical concepts [17]. In both cases, residues are the b_1 coefficients of the Laurent series for a function of a complex variable [17; 26]. The Laurent series is a tool which provides a series representation of a function when it is not analytical at some point. When this is the case, a Taylor series representation does not exist [17]. In complex variable contour integration, Cauchy's residue theorem is

$$\oint f(z)dz = 2\pi j \times \sum \text{Res}f(z), \quad (39)$$

where z is a complex variable and the sum of residues enclosed by the contour can take on non-integer values. The residue theorem for phase unwrapping is given by

$$\oint \nabla\varphi(\mathbf{r})d\mathbf{r} = 2\pi \times \sum Q_i, \quad (40)$$

where $\varphi(\mathbf{r})$ is the phase function evaluated at \mathbf{r} , Q_i are the charges of the branch points within the contour, and the sum of enclosed phase residue charges is restricted to integer values. If the sum of phase residues' charges enclosed is balanced (net charge of zero), Eq. (31) is satisfied and path-independent unwrapping is possible [17]. By connecting oppositely charged branch points and not allowing unwrapping paths to cross, branch cuts force any closed path to encompass a net charge of zero [17].

When addressing branch points and cuts, it is helpful to understand their statistical behavior. It has been shown that the probability density function of optimal branch cut lengths (shortest possible) needed to unwrap a phase is Gaussian [24].

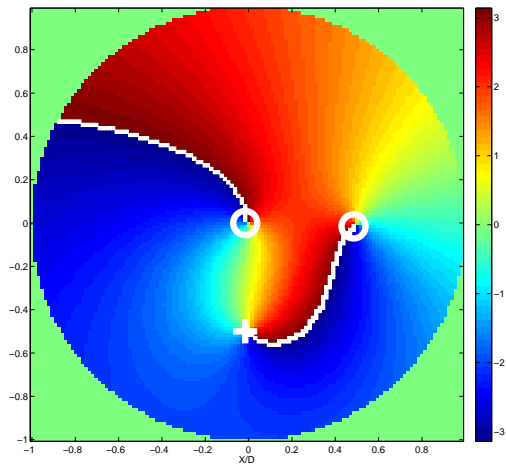


Figure 12. Two-dimensional view of a rotational phase with branch points and cuts. X's mark positive branch points, and O's mark negative branch points.

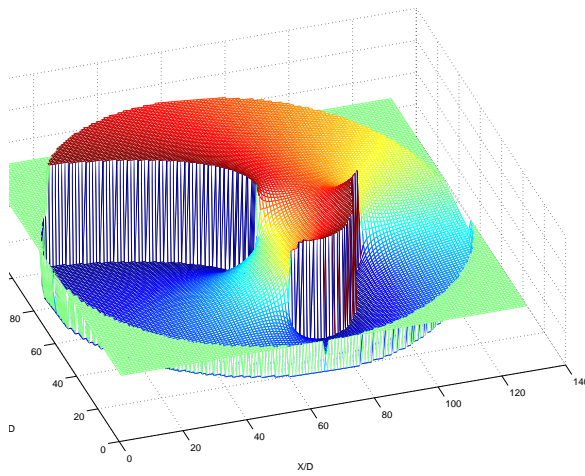


Figure 13. Three-dimensional view of Fig. 12.

Also, the density at which branch points occur in the observation aperture, with respect to increasing turbulence strength is non-linear. In this context, “density” refers to the quantity per unit area in the aperture. Voitsekhovich *et al.* explored the relationship between branch point density and other parameters such as propagation distance, wavelength, and scale sizes [50]. It has been shown that as the propagation distance increases, thereby increasing the turbulence strength, the increase in branch points can be divided into four regions. The first is the weak turbulence region in which branch points are rare. The second is an intermediate region between weak and strong turbulence in which the branch point density grows rapidly. Next, in the region of strong turbulence, the density begins to slow but still increases in a non-linear way. Finally, in the region in which the turbulence is considered saturated, the number of branch points grows linearly. It is also shown that longer wavelengths experience fewer branch points due to less scattering from atmospheric inhomogeneities.

2.2.2 Problem in Adaptive Optics.

There are two main problems that occur in AO when branch points are present, erroneous unwrapping and difficulty conjugating discontinuities in the phase. Branch-point-tolerant unwrapping algorithms which address these two issues can lead to significant improvements in the Strehl ratio and hold promise for future AO systems [42].

The erroneous effects of branch points when using simple unwrapping techniques extend over a wide region in the aperture [15]. As previously described, different unwrapping paths lead to different phase values for the same point, each equally correct. Figure 14 shows how a branch point in the phase can lead to two different values at point B when integrating the gradient from point A. The red path experiences two jumps of 2π , while the green path experiences only one. The two computed phase values cannot be the same.

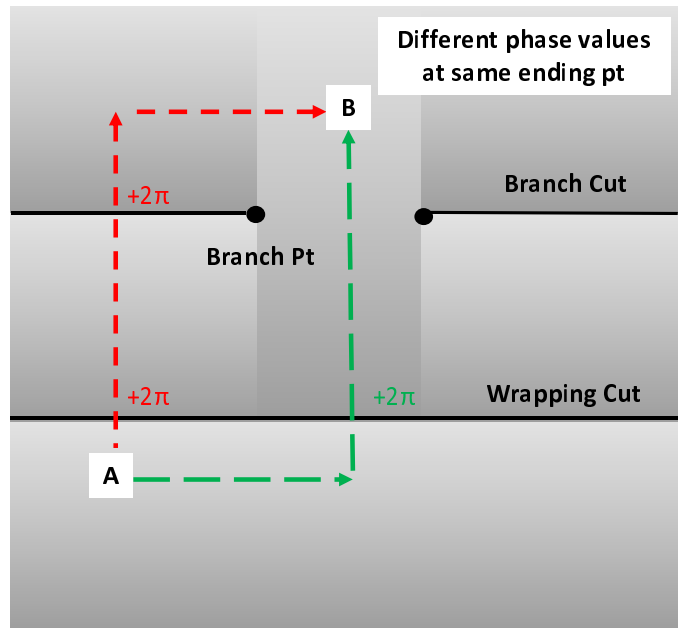


Figure 14. Example of how two unwrapping paths can result in different values at point B when branch points are present.

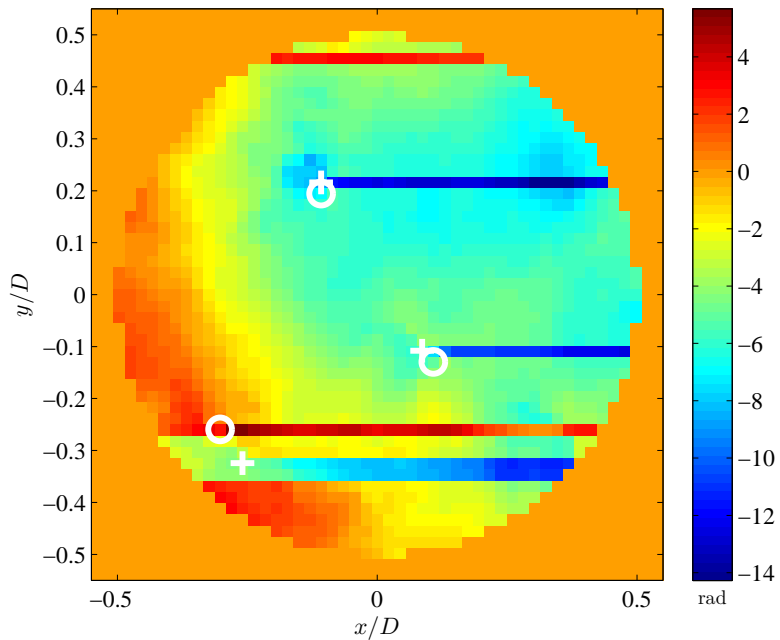


Figure 15. Example of phase with branch points unwrapped using a simple unwrapper. X's mark positive branch points and O's negative.

Figure 15 shows an example of a phase with branch points unwrapped using a simple technique known as Itoh’s method (see Sec. 2.3.1.1). The streaking is indicative of branch points corrupting the unwrapped phase. One of the most common unwrapping methods is the least-squares (LS) algorithm which is discussed in detail in Sec. 2.3.3.2. Fried showed that effects of branch points are transparent to the LS algorithm [15]. This means that a LS reconstructed wave-front does not contain the phase that contains branch point effects. For this reason, the un-reconstructed part is referred to as the non-LS phase. This means that when compared to the original phase incident on the system, the error is the non-LS portion. Even if the reconstructed phase did contain the non-LS component, the continuous-face sheet DM would have difficulty matching the branch cuts. Sharp changes in phase can not be conjugated well by coupled actuators. If these cuts occur in areas of low signal, the effect on the AO system is minimized [14]. This requires that the cuts be short in length and in areas of low irradiance.

2.2.3 Phase Decomposition.

A vector function $\mathbf{F}(\mathbf{r})$ describing a field is considered irrotational if and only if [17]

$$\nabla \times \mathbf{F}(\mathbf{r}) = 0. \quad (41)$$

When this is true, the function can be described by

$$\mathbf{F}(\mathbf{r}) = \nabla\varphi(\mathbf{r}), \quad (42)$$

where the right hand side of Eq. (42) is the gradient of scalar function $\varphi(\mathbf{r})$. If a vector function is the gradient of a scalar function, it does not have a curl component [26] and neither does the scalar function $\varphi(\mathbf{r})$ [17]. For wave-fronts without a curl component,

path-independent unwrapping is possible. This is not the case when

$$\nabla \cdot \mathbf{F}(\mathbf{r}) = 0, \quad (43)$$

which means that the vector function is rotational. If the divergence and the curl of $\mathbf{F}(\mathbf{r})$ are specified everywhere in the region of interest, $\mathbf{F}(\mathbf{r})$ can be expressed as a sum of an irrotational vector function and a rotational vector function which is known as the Helmholtz decomposition theorem [17]. Figure 16 shows the gradients of an irrotational and rotational phase. Circulation is clearly evident in the rotational phase slope shown in plot (a), while the irrotational phase slope in plot (b) shows no circulation.

It is this decomposition which Fried originally proposed be used to find the non-LS phase [15]. Since branch points and cuts only occur in the rotational component of the phase, and since the LS unwrapper only reconstructs the irrotational component, he suggested subtracting the LS phase from the original measured phase to obtain the rotational, or non-LS component. Figure 17 provides an example of how the measured phase can be split into rotational and irrotational components. The non-LS portion can then be added to the LS component to reduce the reconstruction error. Ghiglia and Pritt also proposed a method which utilizes this decomposition [17]. They developed a method to construct the rotational component from just the location and sign of the branch points. By taking the phase from a single branch point at the origin and centering it at the location of each branch point of a similar sign, a superposition can be made which is a close approximation to the rotational component. Since the position of the branch point is only known to be within a given boundary, error is introduced by the phase additions. This error only shows up as an irrotational component, so it can be removed via the Helmholtz decomposition. The result is the rotational field which is measured, plus an arbitrary constant.

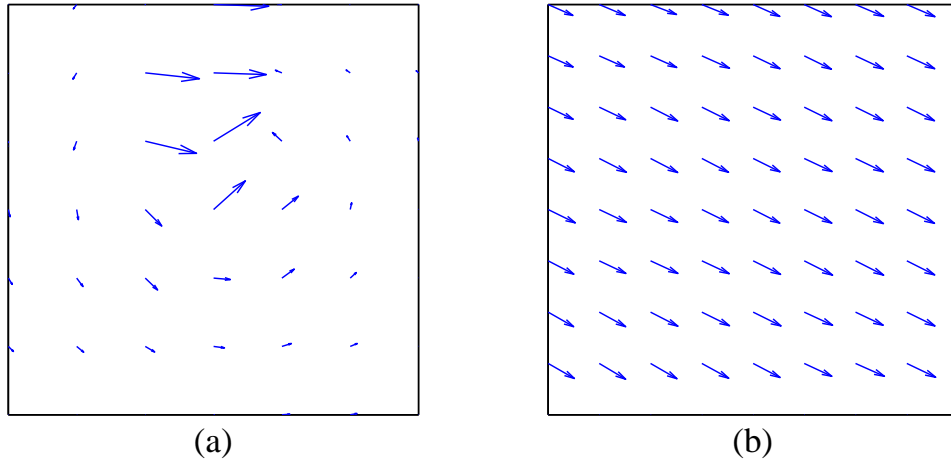


Figure 16. Gradient plots of a (a) rotational and (b) irrotational phase component. The rotational phase contains a branch point at its center.

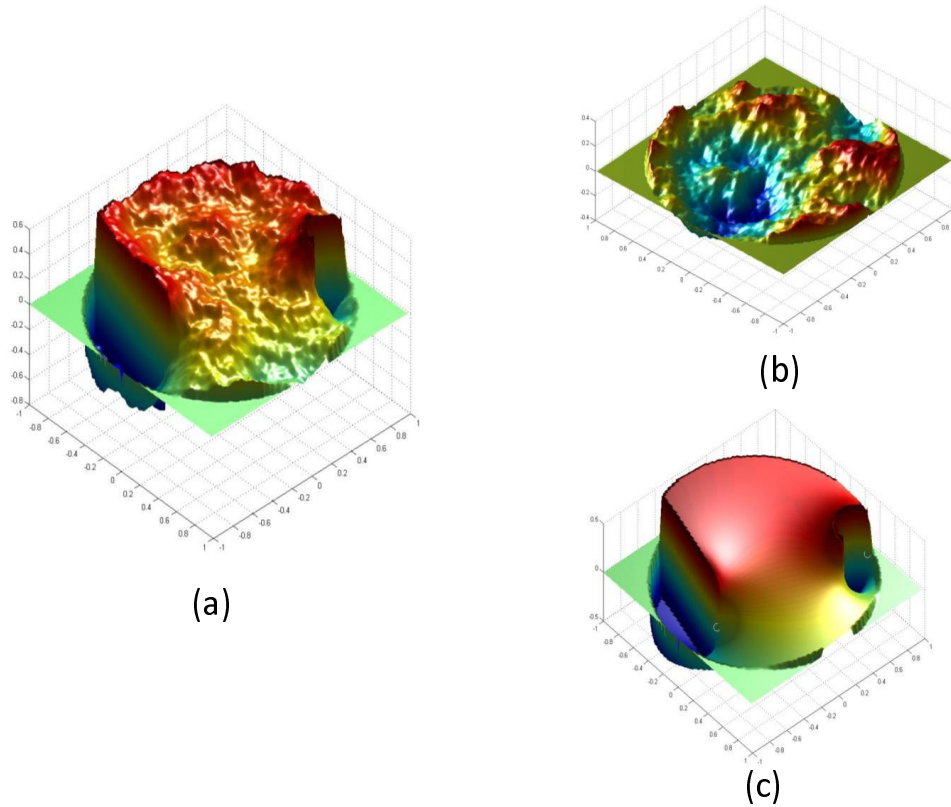


Figure 17. Example of the Helmholtz decomposition using a LS unwrapper, (a) wrapped input, (b) LS component, (c) non-LS component.

2.2.4 Branch-Point Detection.

Some unwrapping methods begin by detecting branch points. One of the most common techniques being used to detect branch points is referred to as the circulation method. Since branch points rarely, if ever fall on a sample point in a discretely sampled grid [42], the circulation method integrates the phase derivatives around each 2×2 pixel combination to determine if a branch point exists within that area. This amounts to computing the sum of the wrapped phase differences around a 2×2 region [17]. Figure 18 shows one loop around point P_1 in a 2×2 array. The phase differences between adjacent samples are computed in both the x - and y -directions and wrapped. Next, the wrapped differences are summed according to

$$\sum W [\Delta\phi (P_1)] = \Delta y_1 + \Delta x_2 - \Delta y_2 - \Delta x_1, \quad (44)$$

where Δx and Δy are the x and y wrapped phase differences. If a non-zero value is obtained from the loop, a branch point is present somewhere in that region. This is a direct consequence of Eq. (40). A counterclockwise loop that results in a positive value, indicates a positive branch point and a negative value, a negative branch point [14]. It is important to note that branch-point detection can be difficult when noise is present. When the noise appears as a rotational component in the phase with the opposite sign as a branch point in the same 2×2 region, a closed-loop integral does not evaluate to an integer multiple of 2π . Branch-point detection methods which only count integer values may not find branch points in loops with noise.

Le Bigot and Wild [27; 51] proposed an alternate method to detect branch points which they claimed is unambiguous in the presence of measurement noise [51]. Their technique converts the non-LS component of the phase into a “potential” function in which the branch points become easily recognized singularities, appearing as peaks

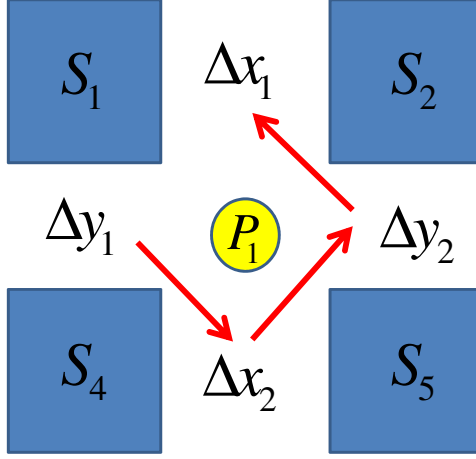


Figure 18. Circulation method of branch-point detection. Each box represents a discrete sample. The differences between adjacent samples are represented by Δx and Δy .

and valleys in a three-dimensional contour plot [27]. The potential function is similar to Fried's Hertz potential (see Sec. 2.3.3.5) in creating logarithmically divergent peaks and valleys. Creation of the potential function is accomplished with one matrix multiplication by which the measured gradients are rotated by $\pi/2$. The potential \mathbf{h} is given by

$$\mathbf{h} = \mathbf{W}^{-1}\mathbf{s}, \quad (45)$$

where \mathbf{s} is a vector containing the measured slopes. \mathbf{W}^{-1} is the pseudo-inverse of

$$\mathbf{W} = \begin{bmatrix} \mathbf{G}_y \\ -\mathbf{G}_x \end{bmatrix}, \quad (46)$$

where \mathbf{G}_x and \mathbf{G}_y are the components of the geometry matrix \mathbf{G} which act on the x and y slopes respectively. The effect of Eq. (45) can be seen in Fig. 19. The rotation of the gradient around a branch point in Fig. 19 (a) becomes divergent when Eq. (45) is applied, as shown in Fig. 19 (b). Figure 20 (a) shows the contour of a simple phase function with two oppositely signed branch points. The corresponding potential \mathbf{h} is shown in Fig. 20 (b) and has a peak at the location of the positive branch point and a

valley at the negative branch point. Other phase functions tested using this method produced potentials which are more ambiguous in their peak and valley locations. Figure 20 (b) also shows a waffle-like pattern which appears in the potential, adding more confusion to the branch point locations. It may be possible to spatially filter the potential to remove the waffle pattern if one chose to use this method of detection.

2.3 Unwrapping Methods

Phase unwrapping is being used in many fields including AO, synthetic aperture radar (SAR), medical imaging, coherent imaging, and speckle interferometry. Each of these fields operates under different constraints, so an unwrapping algorithm which works well for one may not be suitable for others. As mentioned previously, there have been many articles written on two-dimensional phase-unwrapping. Most unwrapping algorithms can be categorized as path-following, regional/local algorithms, global algorithms, or as a hybrid technique. Sorting through the numerous options to choose an appropriate algorithm can be a daunting task. Specific requirements of AO can be used to simplify this task. One of the most important requirements is computational speed. Typical AO systems must operate with kilohertz frame rates, so any unwrapping process must be able to keep up with this rate. Accuracy is also an important factor. An unwrapped phase must be modulo- 2π -equivalent to the measured phase. Finally, since the purpose of AO is to improve wave-front quality, an algorithm's ability to improve the Strehl ratio must be considered. As stated previously, minimizing cut length and moving them to areas of low irradiance improves the Strehl ratio [48]. Unwrappers that simply reduce cut length but do not consider irradiance may not be adequate for AO. The remainder of Chap. II is dedicated to explaining various types of unwrappers and analyzing their applicability to AO in strong turbulence.

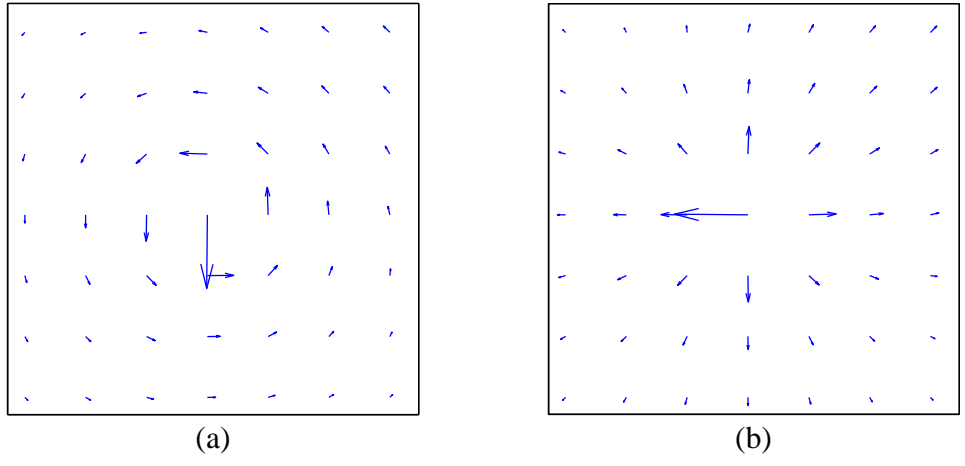


Figure 19. Le Bigot and Wild's method of branch-point detection rotates the gradient of the rotational component (a) turning the vortices into converging and diverging peaks and valleys (b).

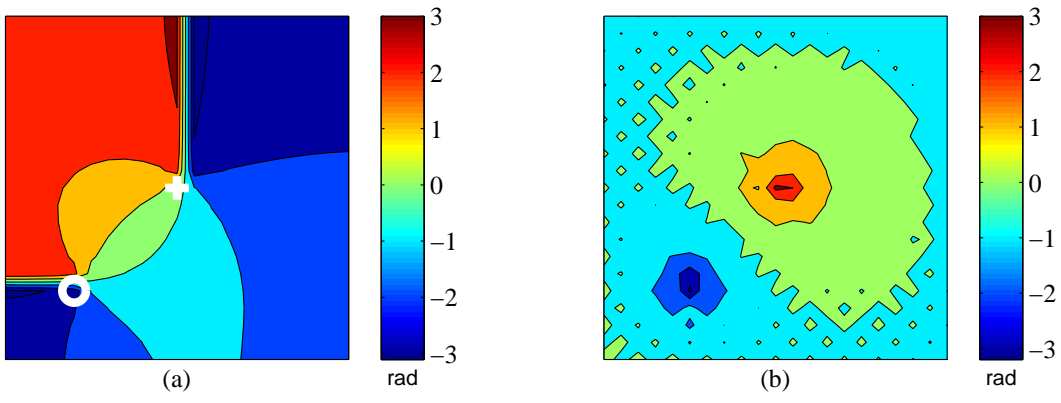


Figure 20. Example of Le Bigot and Wild's branch point detection method applied to a phase function (a) with two branch points indicated by the + (positive) and the o (negative). The potential function created (b) by rotating the gradient creates peaks and valleys at the location of branch points.

2.3.1 Path-Following Algorithms.

2.3.1.1 Path-Dependent.

Path-dependent unwrappers, also known as flood-fill unwrappers employ the unwrapping technique explained in Sec. 2.1.5. They are fast and simple but cannot handle noise or branch points [28]. One such algorithm is the 2-D Itoh's Method [17]. It begins by taking the value of the first grid point in the array (top left) which becomes the reference. Next it integrates the differences down the first column to find the phase at each point. As a result, the first grid point (or sample) in each row is unwrapped. The algorithm then uses the same method to unwrap along each row. As long as the phase does not change more than π radians, this method reconstructs the true phase. However, when this is not the case, the wrapped phase differences no longer represent the true phase gradient and so the method fails to accurately unwrap [17]. Figure 15 is created using Itoh's method to unwrap a wave-front which contained branch points.

2.3.1.2 Residue Compensation.

Residue, or branch-point compensation methods seek branch points and generate branch cuts to allow path-independent unwrapping. The cuts balance the phase residues so that any closed path encloses a net charge of zero [17]. These methods are generally computationally efficient but not robust [28]. Since they require the detection of branch points, they are only as good as the detection method being used. A very well known residue-compensation algorithm is the Goldstein, Zebker, and Werner algorithm, often referred to as just Goldstein [17; 37]. First, the array is searched one 2×2 loop at a time until it finds a branch point. When one is found, a 3×3 pixel box is centered on the top left sample from the 2×2 loop. This 3×3 box is searched for other branch points. If none are found, it makes the box bigger and

continues to search. If a branch point is found, the two are connected. If the two balance, they are marked as “balanced”. Otherwise, branch points are continued to be connected until the web is balanced. When an aperture edge is encountered, it is connected to the branch point(s). Once the branch cuts are in place, the algorithm uses a path-following unwrapping technique which is not allowed to cross the branch cuts. Overall, this method tends to create short cuts and is said to be very fast. However, there are several problems that occur when using this method [17]. First, webs of branch cuts can isolate a region from any unwrapping path as illustrated in Fig. 21. Secondly, the algorithm does not consider the pixel quality when unwrapping. This means that there is no consideration as to the irradiance when placing cuts. Finally, Ghiglia and Pritt showed that in their results the algorithm is slow relative to other unwrapping algorithms. The lack of accuracy and speed make this method a poor choice for AO.

Another type of residue-compensation algorithm is the nearest-neighbor unwrappers [10; 17; 43]. Each branch point is connected to the closest oppositely charged branch point. This is very similar to Goldstein’s algorithm with the exception that branch points can only be connected by one branch cut. This prevents webs of branch cuts from forming. As with Goldstein’s algorithm, a nearest-neighbor approach does not consider the irradiance under the branch cut, and therefore is unlikely to maximize AO system performance.

2.3.1.3 Quality-Guided.

Quality-guided path-following methods unwrap the highest-quality samples first. Sample quality is determined by the difference between that sample and its neighbors. Those with small differences have high quality [37]. Once a quality map has been developed, a flood-fill technique is used to unwrap along a path whose quality does

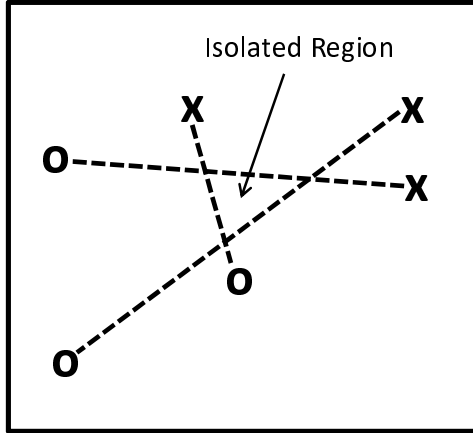


Figure 21. Example of how poorly placed branch cuts can isolate a region from being unwrapped.

not fall below a predetermined threshold [17]. This type of algorithm does not identify branch points, nor does it lay branch cuts. It attempts to avoid those areas altogether. Results presented by Ghiglia and Pritt show that their quality-guided method is about six times slower than the Goldstein algorithm, although it did yield better performance [17]. They also showed that it failed when noise is added due to a poor quality map. Herraes *et al.* proposed a novel “Fast two-dimensional phase-unwrapping algorithm” based on sorting by reliability [28]. They claimed that this method could unwrap a 512×512 array in half a second. Although this may be an improvement over traditional quality-guided methods, it may still be too slow for AO applications.

2.3.2 Regional Algorithms.

Regional algorithms separate an image into regions and process them separately. They provide a compromise between robustness and computational requirements [28]. One such regional unwrapping technique that has been proposed by Herraes *et al.*, is the “Robust, simple, and fast algorithm for phase-unwrapping” [22]. This method attempts to divide and conquer the unwrapping process. It divides an array into four

regions, and then subdivides each one into four smaller regions. It continues this process until it is left with a 2×2 grid which it unwraps using a flood-fill technique. By unwrapping each section independently, the error from a particular branch point can be isolated. The algorithm then uses a technique to stitch the whole phase surface back together again. Although this process is intended for real-time implementation, it is recursive and may still be too slow for AO applications. However, the possibility of parallel processing the separate sections could be used to speed it up. Finally, one interesting thing to note about the research conducted by Herraes *et al.*, is that this is the only algorithm encountered in this research which splits a wave-front up and then pieces it back together. If another algorithm needed to be applied differently to separate sections of the phase, this technique would be useful.

2.3.3 Global Algorithms.

Global algorithms minimize a global function which depends on the application. Examples include minimizing the overall branch-cut length, the irradiance under a branch cut, or the error of an estimated phase when compared with the true phase. Global algorithms attempt to reach the best solution that exists, and for this reason are considered robust but computationally intensive [28]. This section discusses the following global algorithms: minimum-norm, and “other” which includes genetic, simulated annealing, and Flynn’s minimum discontinuity. Finally, a post-processing step required by all global techniques is discussed.

2.3.3.1 Minimum-Norm Algorithms of the General Form.

A minimum-norm algorithm imposes constraints on a desired solution to make the local derivatives of the unwrapped phase match the measured derivatives “as closely as possible” [17]. Written as L^p -norm, p can be chosen depending on the type of

solution desired. Minimum-norm algorithms minimize a cost function, which in the case of phase unwrapping is a function of the slope discrepancy. Slope discrepancy refers to the difference between the actual slopes of the wave-front and the the slopes produced by the geometry matrix times the actuator commands [difference between left and right side of Eq. (30)]. The general L^p -norm solution minimizes the cost function ϵ^p given by

$$\epsilon^p = \sum_{i=0}^N |\text{slope discrepancy}|^p, \quad (47)$$

where N is the total number of slopes considered.

2.3.3.2 Minimum-norm Algorithms of the Least-Squares Form.

One of the most common minimum-norm algorithms is of the least-squares form, meaning that $p = 2$, so that the square of the magnitude of all the differences is minimized. Roggemann claims that least-squares for phase reconstruction is most widely used in AO for three reasons [42]. The first is that the development of AO has mainly been focused on astronomical applications where the turbulence is in the near-field and phase effects dominate over amplitude fluctuations. Secondly, LS does not require a statistical model and does not require constant monitoring of turbulence conditions [42], as do techniques such as Minimum Variance Unbiased (MVU) models [34]. Finally, under weak turbulence conditions, LS algorithms tend to reduce noise since they average all the possible paths to estimate the phase at one point.

In contrast to other algorithms, LS solutions are not found by adding integer values of 2π to the wrapped phase [32]. This means that the difference between the LS unwrapped phase and the wrapped phase can have a value which is not an integer multiple of 2π . More importantly, the LS algorithm only reconstructs the irrotational component of the phase which leaves out the rotational component [15]. This tends

to result in an overly smoothed reconstructed phase [42]. LS unwrappers always underestimate the local average phase slope when branch points are present [17]. This is because branch points and cuts are only present in the rotational component, and the LS unwrapper is blind to their effects [15].

LS phase unwrapping can be unweighted or weighted. Unweighted LS unwrapping amounts to solving partial differential equations in the form of a discrete version of Poisson's equation [17]. The unweighted least-squares solution to an over-determined set of linear equations

$$\mathbf{G}\boldsymbol{\phi} = \mathbf{s}, \quad (48)$$

is given by

$$\mathbf{G}^T\mathbf{G}\boldsymbol{\phi} = \mathbf{G}^T\mathbf{s}, \quad (49)$$

$$(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{G}\boldsymbol{\phi} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{s}, \quad (50)$$

$$\boldsymbol{\phi}_{LS} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{s}, \quad (51)$$

where $\boldsymbol{\phi}$ is an $MN \times 1$ solution vector containing phase, \mathbf{s} is the $(2MN - M - N) \times 1$ wrapped phase differences, and \mathbf{G} is a geometry matrix which converts phase to wrapped phase differences [49]. The T superscript represents the transpose, and as discussed in Sec. 2.1.4, the inverse operation is actually the pseudo-inverse. $\mathbf{G}^T\mathbf{G}$ performs the discrete Laplacian operation [17]. One method being used to solve a partial differential equation in the form of a discrete version of Poisson's equation is a Fast Fourier Transform (FFT). These algorithms stitch together mirror images of the phase to create periodicity, as shown in Fig. 22. When the Poisson's equation is restricted to a periodic function, it has a unique solution, and boundary equations are not required [17]. It can then easily be solved using Fourier-transform techniques. There are many other methods to find the least-squares solution. For a more exhaustive discussion see Ref. [17].

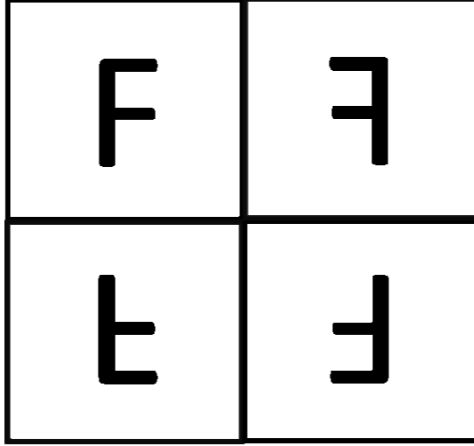


Figure 22. Example of periodicity created to use a Fast Fourier Transform (FFT) algorithm to solve an unweighted least squares phase-unwrapping problem.

Weighted LS algorithms use weights or quality maps to avoid integrating around branch points [17]. They are still affected by branch points, although to a lesser degree than the unweighted LS [36]. The weighted least-squares solution to an overdetermined set of linear equations given by Eq. (48) is

$$WG\phi = Ws, \quad (52)$$

$$G^T W^T W G \phi = G^T W^T W s, \quad (53)$$

where W is a matrix of weights. To simplify notation, let

$$Q = G^T W^T W G, \quad (54)$$

$$c = G^T W^T W s, \quad (55)$$

which gives

$$Q\phi = c. \quad (56)$$

The vector c contains the weighted phase differences with the discrete Laplacian oper-

ation applied to them [17]. Unfortunately, Eq. (56) cannot be solved using unweighted LS techniques such as an FFT-based algorithm. It requires an iterative process to solve, and is therefore more computationally intensive and requires more time than unweighted LS. Two methods that can be used to solve weighted LS problems are the Picard method and Preconditioned Conjugate Gradient (PCG). Since the Picard method requires many iterations, it is impractical and is not discussed further here. PCG is a faster way of solving sparse linear-algebra equations [34]. The preconditioning refers to a step in which the reconstruction matrix is shaped closer to an identity matrix. This is done by solving for the unweighted LS solution and using it as an estimate for the PCG solution. In each iteration, the algorithm searches for the solution in a conjugate (perpendicular) direction [34]. This method requires N iterations for a $N \times N$ matrix [17].

2.3.3.3 Multigrid Algorithms and the Complex Exponential Reconstructor.

Multigrid algorithms are used for solving partial differential equations (such as in LS unwrapping) on large grids [17]. They are based on the application of Gauss-Seidel relaxation schemes which can extract high-frequency content (local smoothing), leaving the low-frequency or global information [17]. Gauss-Seidel schemes are slow to converge and therefore can only be practically applied to small grid sizes. Phase-unwrapping multigrid algorithms use a process called *restriction* to transfer the unwrapping problem to a coarser grid which allows the application of Gauss-Seidel relaxation schemes to become practical [17]. The lower sampling rate of the coarse grid increases the spatial frequency of residual error contained in the phase data. Relaxation then smooths the error, leaving a smooth surface structure which can be transferred back to the fine grid as global information in a process called

prolongation [17].

One such application of the multigrid technique is the Complex Exponential Reconstructor (CER) which is designed to reconstruct the LS phase as well as the non-LS phase [38]. It is a non-linear recursive algorithm that consist of three steps, reduce (*restriction*), solve, and rebuild (*prolongation*) [13]. The addition of phase and phase differences is done by multiplying complex exponentials to eliminate errors introduced by wrapping of the phase [31]. The differential phasor Δu is related to the corresponding phase difference $\Delta\phi$ by

$$\Delta u = \exp(i\Delta\phi), \quad (57)$$

and the phasor u is related to the phase ϕ by

$$u = \exp(i\phi). \quad (58)$$

It can be seen from Eq. (57) that changes of 2π in the phase differences do not affect the differential phasors, which is why they are not affected by 2π discontinuities such as branch cuts.

During the reduce step, the differential phasors are calculated, and the input grid size is reduced by almost half. To illustrate this, Fig. 23 shows how a 5×5 grid is reduced to 3×3 . The new differential phasors are calculated between two points by summing along the three paths represented by solid and dashed arrows and then averaging. This process is repeated until only a 2×2 grid remains, at which point the reduce step is complete. The solve step consists of reconstructing the phasors at the corners of the 2×2 grid using a LS algorithm [38]. Finally, the rebuild step uses the reconstructed phasors along with the differential phasors found in the reduce step to reconstruct phasors on a slightly finer grid. This process is repeated until the original

grid size is reached. In general, CER outperform LS unwrappers in the presence of branch points and branch cuts since they reconstruct the non-LS phase. However, CER performance is still degraded in strong scintillation since it does not completely account for the effects of scintillation on the WFS [38].

2.3.3.4 Other.

There are many other global phase-unwrapping algorithms which have been developed for SAR, where the time requirements of AO are non-existent and unwrapping time can be sacrificed to achieve the best solution possible. Most of these algorithms only attempt to minimize the overall length of branch cuts and do not consider the irradiance under the cuts. They treat phase unwrapping similar to the traveling-salesman problem (TSP). The TSP is a theoretical combinatorial optimization problem [43]. It is summarized as a traveling salesman that must visit N cities in the shortest route, not passing through any city twice, and must return home. The complexity of the problem grows exponentially as N increases, and it takes N iterations to look at every possible solution [43].

Genetic Algorithms (GA's) are useful when it is desired to minimize a large function with many local minima [43]. A GA uses artificial intelligence which mimics evolution of genes. It uses techniques based on natural selection, crossover, and mutation to ensure that the problem space continues to be searched for better solutions [43]. As with the principles they are based on, GA's converge on a solution slowly and are not suitable for AO.

Simulated annealing is a technique for solving TSP problems and is based on the physical process being used to remove internal strains from solids [10]. This method does provide a correct solution, but when a large number of branch points are present, it is very slow and can become impractical. Cusack *et al.* explains that a

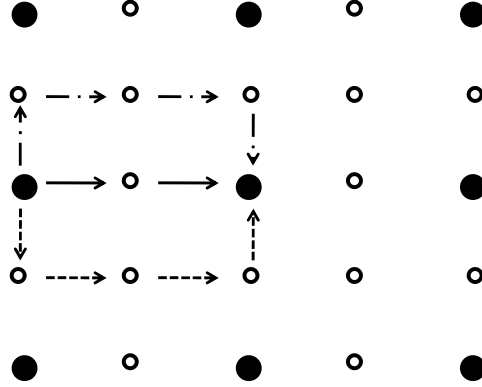


Figure 23. The reduction of a 5×5 grid to 3×3 . The the solid dots represent points that remain in the reduced grid. Solid and dashed arrows represent the different paths being used determine the differential phasors for the coarser grid. The differential phasors of the fine grid are summed and averaged along these paths.

map containing 3000 branch points would take 24 hours to unwrap with this method on a two-computer SPARCstation setup in 1995 [10]. Although AO applications do not experience this many branch points, and computers are much faster now, the example illustrates that this is not a quick algorithm.

Another global phase-unwrapping algorithm is Flynn's minimum-discontinuity approach described by Ghiglia and Pritt [17]. This method first identifies the fringe lines in the wrapped phase and adds integer values of 2π to regions separated by the fringes, which minimizes the discontinuities. This algorithm finds the actual solution of minimum discontinuity. It is a slow process which does not work well when there are an unbalanced number of branch points in the phase. It also does not consider the irradiance. For these reasons, it should not be applied to AO.

2.3.3.5 Post Processing Congruence Operations.

Global phase-unwrapping techniques require a congruence operation to ensure that $\exp(j\phi) = \exp(j\phi_w)$, where ϕ is the unwrapped phase and ϕ_w is wrapped phase [32]. A congruence operation computes the LS solution and forms a key of integers that are multiplied by 2π and then added to the wrapped phase. This makes the surface

congruent, or modulo- 2π -equivalent, to the wrapped input [36]. The congruence operation is defined as

$$\phi_c = \phi_{LS} + W[\phi_w - \phi_{LS}], \quad (59)$$

where ϕ_c is the unwrapped phase which is congruent to the wrapped phase ϕ_w , and ϕ_{LS} is the phase computed by the LS unwrapper [36]. $W[\cdot]$ represents the wrapping function.

The congruence operation is insufficient when noise is present for the following reasons [17]. First consider the no-noise case where

$$\phi_{LS} = \phi_w + n\pi. \quad (60)$$

Substituting into Eq. (59) and reducing gives

$$\phi_c = \phi_w + n\pi + W[\phi_w - (\phi_w + n\pi)] \quad (61)$$

$$\phi_c = \phi_w + n\pi + W[-n\pi]. \quad (62)$$

Equation (62) shows that value of n determines ϕ_c . When n is even, $W[-n\pi] = 0$ and

$$\phi_c = \phi_w + n\pi. \quad (63)$$

If n is odd, there are two possible values for ϕ_c . For n odd and negative,

$$\phi_c = \phi_w + (n - 1)\pi. \quad (64)$$

For n odd and positive,

$$\phi_c = \phi_w + (n + 1)\pi. \quad (65)$$

In a continuous phase function, the value of n changes from sample-to-sample fairly

smoothly. This means there are rarely cases where the difference between two regions is 2π . Now consider the case where a zero-mean Gaussian noise ϵ is present so that $\phi_{LS} = \phi_w + n\pi + \epsilon$. Substituting into Eq. (59) gives

$$\phi_c = \phi_w + n\pi + \epsilon + W[\phi_w - (\phi_w + n\pi + \epsilon)], \quad (66)$$

$$\phi_c = \phi_w + n\pi + \epsilon + W[-n\pi - \epsilon]. \quad (67)$$

Now, the value of n and ϵ determines ϕ_c . Regardless of the value of ϵ , when n is even, $W[-n\pi - \epsilon] = -\epsilon$ and $\phi_c = \phi_w + n\pi$. The value of ϵ plays a much larger role when we consider the case where n is odd. For n odd and either positive or negative,

$$\phi_c = \begin{cases} \phi_w + (n+1)\pi, & \epsilon > 0, \\ \phi_w + (n-1)\pi, & \epsilon < 0. \end{cases} \quad (68)$$

When two samples have oppositely signed noise values, there is a 2π discontinuity between them. If a zero-mean Gaussian PDF is assumed for the noise, many discontinuities occur in areas where n is odd since there is an equal chance of both positive and negative noise. To minimize the number of discontinuities, a constant value h in the range $[0, 2\pi)$ should be added to the non-LS component before and after wrapping so that $\phi_c = \phi_{LS} + h + W[\phi_w - \phi_{LS} - h]$ [36]. The value of h must be chosen to obtain the desired results.

Phase-unwrapping algorithms have been developed which utilize some or all of the PCO. When using a LS reconstructor, the PCO is required to incorporate the rotational component into the reconstructed phase and reduce the error. One of the first to propose such a method was Fried. Early on, Fried proposed finding the “hidden phase” (non-LS phase) by subtracting the LS unwrapped phase from the wrapped phase. This provided the missing data which could be added back to the

LS phase [15]. Fried’s method is very similar to the PCO, although there is no suggestion of wrapping the “hidden phase” before adding it to the LS component. Later, he derived an analytic formula for the hidden phase given by

$$\phi_{hid}(\mathbf{r}) = Im \left\{ \log \frac{\prod_{k=1}^K (x - x_k) + i(y - y_k)}{\prod_{k=1}^K (x - x'_k) + i(y - y'_k)} \right\}, \quad (69)$$

where x and y are the cartesian coordinates of $\phi_{hid}(\mathbf{r})$, x_k and y_k are the coordinates of the k^{th} positive branch point, and x'_k and y'_k are the coordinates of the k^{th} negative branch point [13]. Fried called this algorithm SmoothPhase [13]. Use of Eq. (69) requires knowledge of branch-point locations, which as previously discussed, can be difficult to determine. Arrasmith’s research [3] essentially implemented this equation in a PCO algorithm and evaluated its performance when zero-mean Gaussian noise is present. He showed that the algorithm worked well when the noise PDF had standard deviations of $\pi/3$ and $\pi/2$.

Another phase-unwrapping algorithm, which more closely follows the PCO, is Venema and Schmidt’s least-squares principal-value plus four (LSPV+4) [48; 49]. This method focuses on selecting the optimum value of h when computing ϕ_c , with one exception. It does not add h to the wrapped non-LS component before adding it to the LS unwrapped phase. That is,

$$\phi_c = \phi_{LS} + W [\phi_w - \phi_{LS} - h]. \quad (70)$$

Altogether, this algorithm evaluates Eq. (70) for four different values of h and chooses the unwrapped phase with the lowest irradiance around the branch cuts. Figure 24 shows how the parameter h affects the branch cuts in the non-LS phase. The branch points remain in the same location but the branch cuts change. Since h is periodic,

$h = 0$ in Fig. 24(a) is identical to $h = 1$ in Fig. 24(i). In an ideal situation, application of the PCO would use the parameter h which gave the best Strehl ratio. Unfortunately, the effect of h on Strehl ratio is not available until after the unwrapped phase has been applied to the mirror and the field conjugated. As previously mentioned, branch cuts that occur in areas of low irradiance minimize negative effects on the AO system. Venema and Schmidt used this concept to develop another metric which is directly accessible by the PCO, normalized cut length. This is the name given to the integral of the field irradiance along any phase cuts, divided by the average irradiance of the field [48]. They showed that there is a high anti-correlation between normalized cut length and Strehl ratio (-0.99). Simulations conducted compared LSPV+4 to various unwrappers including Fried’s SmoothPhase. Results showed that under strong turbulence (0.8 log-amplitude variance), LSPV+4 gave the best performance at a reasonable computational speed.

2.3.4 Hybrid Algorithms.

With so many phase-unwrapping methods to choose from and many applications, hybrid approaches which mix and match various techniques are common. To ensure a thorough review, a couple are mentioned in this section. The first is the hybrid genetic algorithm [43], which combines both global and local methods to solve the TSP. The focus of this algorithm is simply minimizing the cut length. It does not try to avoid areas of high irradiance when placing branch cuts. Unfortunately this algorithm inherits its speed from the global portion, and is not quick enough for AO.

The mask cut algorithm described by Ghiglia and Pritt [17] is a hybrid between quality-guided path-following and residual compensation techniques. It combines the advantages of Goldstein’s algorithm and the quality-guided methods. First, it places branch cuts to serve as unwrapping barriers. Then, it unwraps using pixel quality

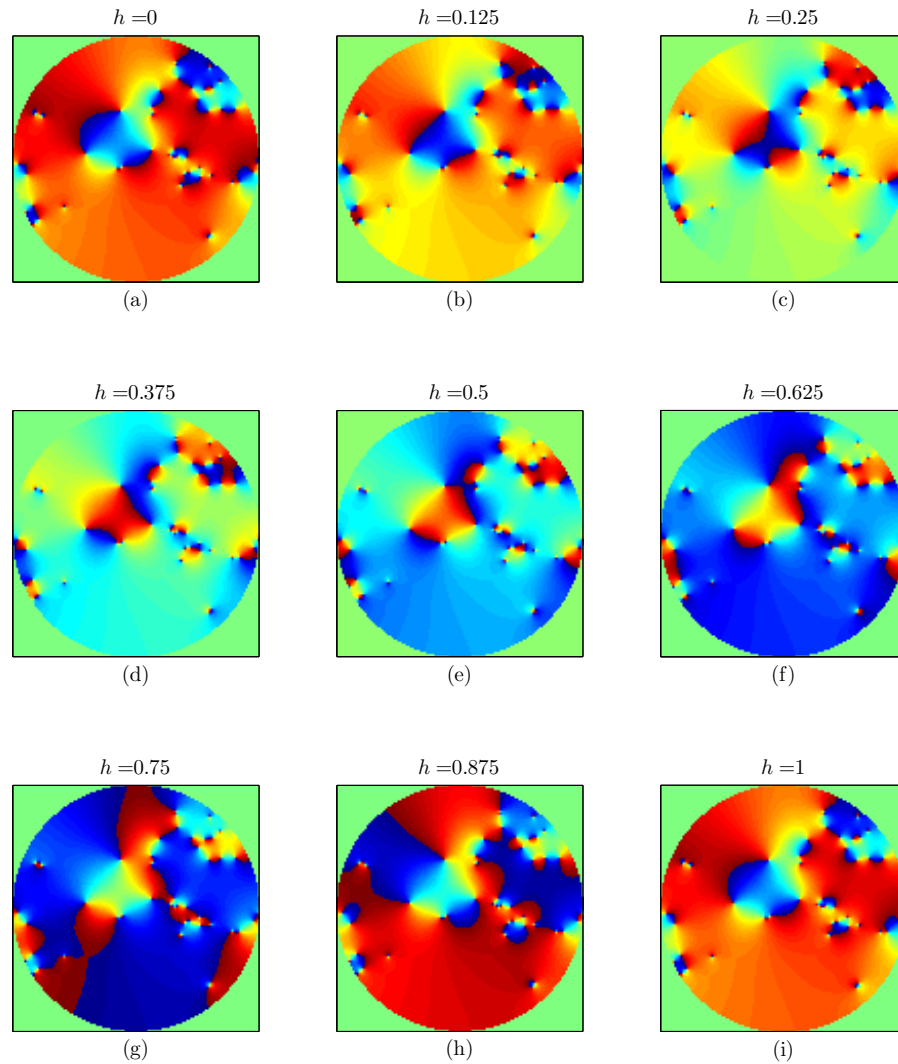


Figure 24. The wrapped non-LS phase is altered by the value of h chosen when applying the PCO. Values of h and phase are given in waves where blue represents zero and red represents one-wave.

to guide the path. Unfortunately, it took over eight times longer than the Goldstein method, and it did not work well when noise is present [17].

2.4 Chapter Summary

In summary, light propagating through strong turbulence experiences nulls in intensity which result in branch points. These branch points require special consideration when unwrapping the phase of the complex optical field. Regional, global, and hybrid phase-unwrapping algorithms have been presented which attempt to mitigate the effects of branch points when unwrapping phase. Most are computationally intensive and cannot be used in real time AO. LSPV+4 is a proposed implementation of a PCO algorithm which is among the fastest methods and has been shown through simulations to be effective. It does not find the lowest IWCL possible, but rather chooses the best of four cases. An optimized PCO unwrapping algorithm may result in lower IWCL and higher Strehl ratios. Chapter III explores the PCO parameter space and discusses the development of several optimizations.

III. Simulation Environment and Algorithm Design

This chapter discusses the methodology for achieving the objectives stated in Sec. 1.1. It describes the setup and validation of the simulation environment, the unwrappers which the new algorithms are compared against, and the methods of comparison. Information obtained from exploration of the parameter space is also presented. Finally, various optimized algorithms, along with the optimization techniques are described.

3.1 Simulation Environment

To study, develop, and compare phase-unwrapping algorithms for improving AO performance in the presence of strong turbulence, it is necessary to simulate both atmospheric propagation and a complete AO system. WaveProp and AOTools are two MATLAB[®] toolboxes that are selected for this task based on their ease of use and extensibility. AOTools is a package of functions and graphical user interfaces (GUI) for analyzing tasks related to AO systems [8]. WaveProp is a package of classes and functions that simulate AO components like DMs and wave-front sensors. [9]. Both toolboxes are provided to AFIT by the Optical Sciences Company (tOSC).

3.1.1 Atmosphere.

A common approach to simulating effects of atmospheric turbulence is to treat the turbulence as a number of discrete layers [45]. Each layer is modeled as a phase screen which adds phase delay to an incident field. Alternating steps of free-space diffraction and phase-screen accumulation represent the effects of propagating through an extended volume of turbulence. Phase screens are created by two-dimensional arrays of computer-generated random numbers, representing phase values which have

the desired spatial statistics based on a given turbulence model [45]. Screens based on the Kolmogorov turbulence model are used in this research. An example is shown in Fig. 25. To apply the desired temporal statistics to the propagated field based on platform and/or wind velocities, phase screens can be shifted transverse to the optical axis as a function of time. This concept is based on the Taylor frozen-turbulence hypothesis. The velocity at which the screens are shifted is calculated to ensure the propagated field has the correct Greenwood frequency, as described in Sec. 2.1.1.

The first step in developing a simulation to research phase-unwrapping algorithms is to consider a propagation geometry that results in branch points. To induce scintillation in the field, horizontal propagation of a point source is modeled at a wavelength of $1.06 \mu\text{m}$ with constant turbulence. Point sources are commonly used in literature as an ideal representation of the beacon observed by an AO system. A modeled point source is propagated 60 km to a 1 m aperture in the observation plane as shown in Fig. 26. An ideal point source is given by a Dirac delta function and therefore would have a constant spectrum spanning all spatial frequencies. Computer simulation of a point source requires that it be bandlimited [45]. WaveProp creates a bandlimited point source by back-propagating a square shape from the observation plane to the source plane, creating a shape similar to a two-dimensional sinc. This gives the source spherical-wave properties such as uniform intensity and parabolic phase in a finite region of the observation plane. The point source modeled by WaveProp for this research is shown in Fig. 27.

The next important step in simulating atmospheric propagation is to ensure that the source and observation planes are sampled adequately. Since wave-optics simulations are based on Discrete Fourier Transforms (DFT), the Nyquist sampling criterion must be satisfied to avoid aliasing [45]. Using a technique developed by Schmidt [45], the simulation geometry and turbulence strength are considered to produce a contour

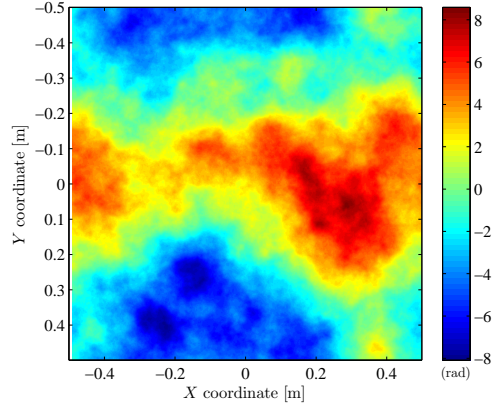


Figure 25. An example of a phase screen generated using Kolmogorov turbulence model. The side length is 1 m and $r_0 = 0.1$ m.

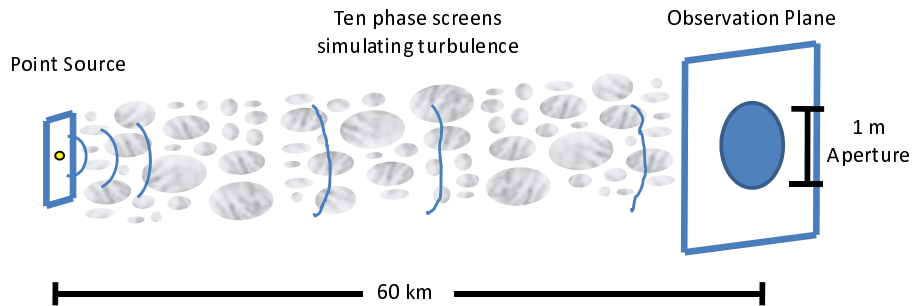


Figure 26. Simulation geometry showing a point source propagating 60 km through ten phase screens and being collected by a 1 m aperture in the observation plane.

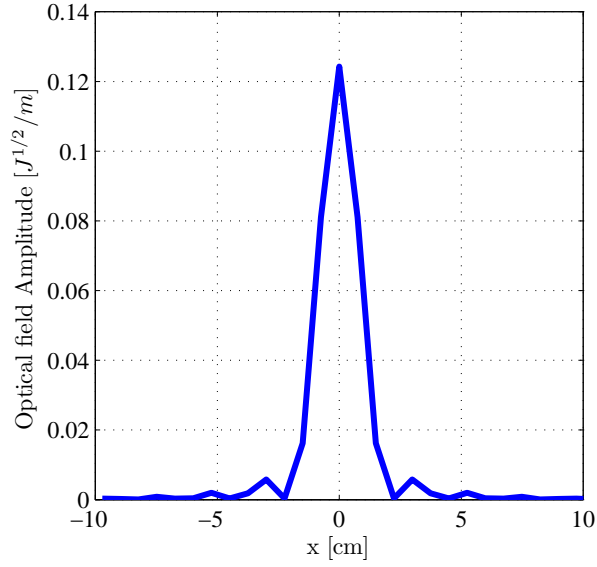


Figure 27. A $y = 0$ slice through the optical field of a bandlimited point source for use in computer simulations.

plot, allowing easy selection of the grid size and spacing needed for accurate results. Figure 28 shows the plot being used for this research. The ratios D_1/δ_1 and D_n/δ_n represent the number of grid points across the region of interest in the source and observations planes, respectively. The sampling constraints are represented by the contour lines which indicate the grid size N ($N = 2^n$, where n is the contour value), and the dashed line which indicates the minimum number of samples across the regions of interest needed to prevent aliasing. To avoid lengthy computations, it is desired to have a maximum grid size of 1024 points per side. This requires a point be chosen below the $n = 10$ contour line (but above the dashed line). The point selected is shown in Fig. 28 and corresponds to approximately four points across the central lobe of the model point source and 120 points across the aperture in the observation plane. It is important to note that the sampling analysis is conducted for the highest turbulence strength simulated (most restrictive case). In addition to satisfying the geometric constraints, the use of DFT for propagation requires that one adequately sample the quadratic phase factor that appears in the transform [45]. Schmidt's tech-

nique considers this requirement, resulting in a constraint on the maximum distance possible for a single propagation Δz_{max} . Dividing the total propagation distance by Δz_{max} gives the minimum number of partial propagations required. For this research, that is found to be four. To err on the side of caution, ten partial propagations are being used, requiring ten phase screens, a number commonly used in literature to model horizontal propagation.

Once the geometry is determined, phase screens are generated, and atmospheric models are developed using WaveProp. In all, five different models are developed to simulate turbulence of various strengths as shown in Table 1. The models are developed using constant C_n^2 values corresponding to the desired Rytov numbers. A constant 5 mph wind is added, resulting in the Greenwood frequencies shown in Table 1.

The final step in developing the atmospheric propagation model is validation. To ensure the geometry between the source and aperture is modeled correctly, a vacuum (no turbulence) propagation is conducted. As expected, Fig. 29 shows a uniform irradiance and phase in the telescope’s entrance pupil. The phase has been collimated to remove the phase factor associated with a spherical wave. Next, a simulation is conducted with the turbulence included. Figure 30 shows how turbulence affects the field irradiance and phase for a Rytov number of 0.8. At this turbulence

Table 1. Atmospheric parameters being used for computer simulation. Five different cases are considered from weak to very strong turbulence strengths, relative to the scintillation encountered.

Turbulence	Rytov number	r_0 [cm]	f_G [Hz]
Low	0.2	22.8	16.8
Moderate	0.4	15.1	25.5
Moderate/High	0.6	11.8	32.5
High	0.8	9.9	38.7
Very High	1.0	8.5	45.0

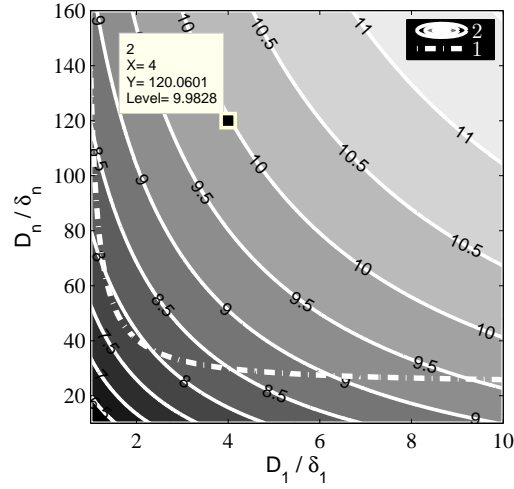


Figure 28. Graphical tool being used for sampling analysis in this research. D_1/δ_1 and D_n/δ_n represent the number of grid points across the region of interest in the source and observations planes, respectively. The sampling constraints are represented by the contour lines which indicate the grid size N ($N = 2^n$, where n is the contour value) and the dashed line indicates the minimum number of samples across the regions of interest needed to prevent aliasing. The data marker shows the point chosen.

strength, one expects to see high scintillation and phase distortion, which is the case in Fig. 30. Although a visual inspection of Fig. 30 indicates that the turbulence model is working, more involved formal techniques are used to validate the results. This includes comparing the theoretical and observed wave structure functions, PSF's, and MTF's.

Figure 31 shows the theoretical and observed phase structure functions for Rytov numbers of 0.4 and 1.0. The simulated structure function is averaged over 20 propagations, and in general agrees with theory. Figure 32 shows the simulated PSF for the turbulence model matches the theoretical PSF based on the turbulence parameters. The final validation technique considers the frequency response of the turbulence model by examining the MTF. Figure 33 shows the theoretical and average observed MTF for Rytov numbers of 0.4 and 1.0. It also shows how they compare with the diffraction limit of the aperture.

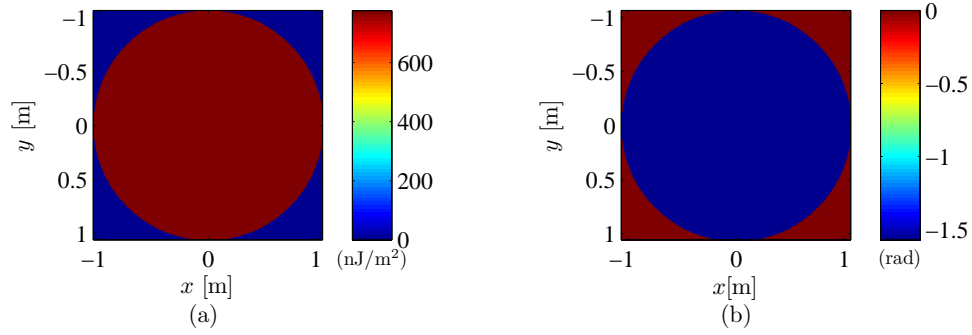


Figure 29. Vacuum (no turbulence) simulation of a point source with uniform irradiance (a) and a uniform collimated phase (b).

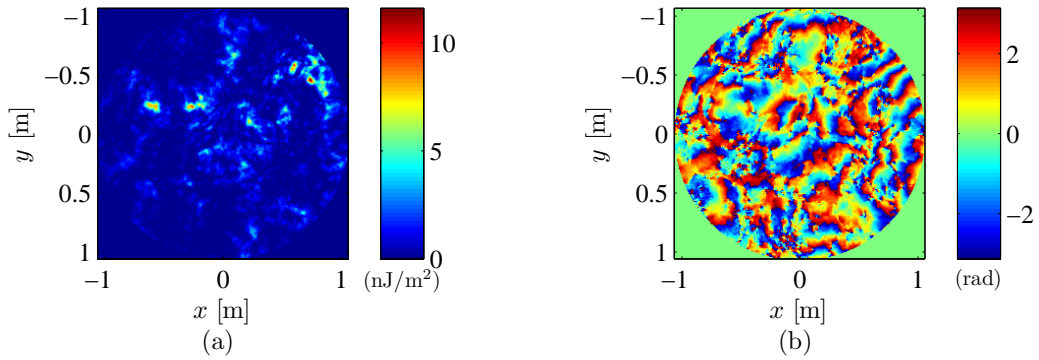


Figure 30. Example propagation of a point source in turbulence, with a Rytov number of 0.8. The irradiance (a) is highly scintillated and the phase (b) is distorted.

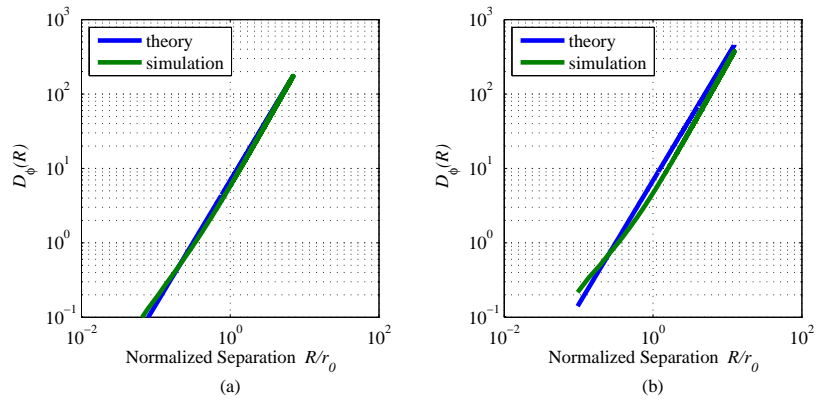


Figure 31. Theoretical and simulated structure functions in the observation plane averaged over 20 propagations for Rytov numbers of 0.4 (a) and 1.0 (b).

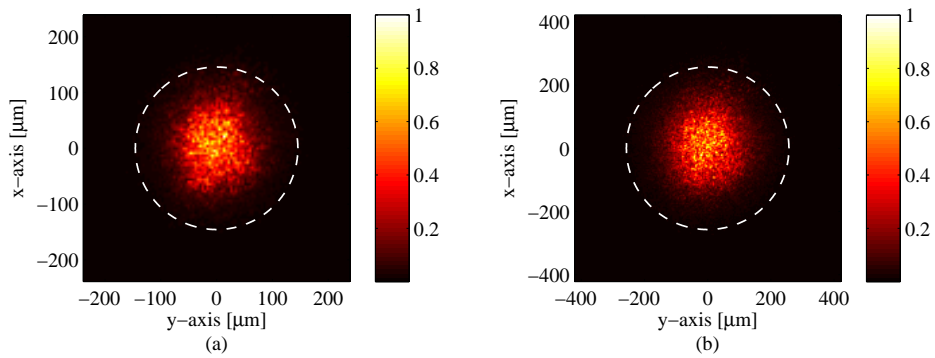


Figure 32. Theoretical (dashed white circle) and simulated ('hot' fill) PSFs averaged over 20 propagations for Rytov numbers of 0.4 (a) and 1.0 (b).

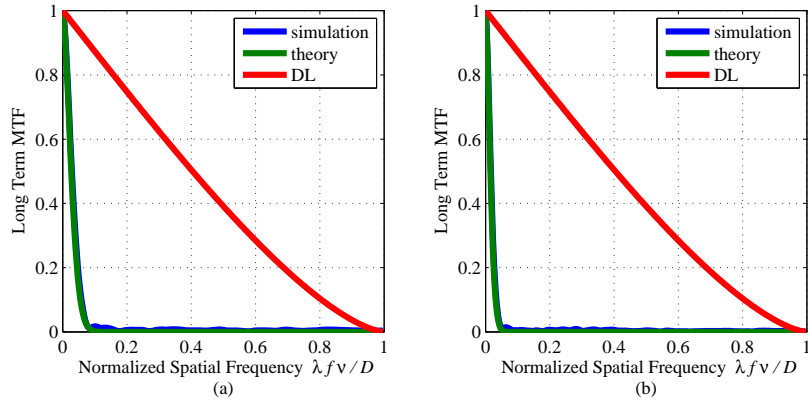


Figure 33. Theoretical and simulated MTFs averaged over 20 propagations for Rytov numbers of 0.4 (a) and 1.0 (b). The red lines represent the diffraction limit, and the x-axis has been normalized by the width of the diffraction-limited MTF.

3.1.2 AO System.

Once the turbulence model is developed and validated, an AO system is modeled. The first element in the system is a telescope which collimates and demagnifies the incident field from the 1 m aperture to a beam of width 2 cm. This allows the entire beam to be measured and corrected by small detectors and mirrors. Figure 34 shows a diagram of the closed-loop system receiving light from the telescope exit pupil. The FSM is controlled by the tracking sensor and processor. They measure the residual tilt of the beam after reflection from the DM and apply a linear control law to the output. The control law coefficients being used are $\alpha = 0.995$ and $\beta = 0.5$.

Modeling a DM typically requires interpolating low-resolution DM commands to a high-resolution grid size, equal to that of the optical field being compensated. An influence function (IF) may be applied to represent the coupling of actuators either before or as part of the interpolation process, as discussed in Sec. 2.1.3. In addition,

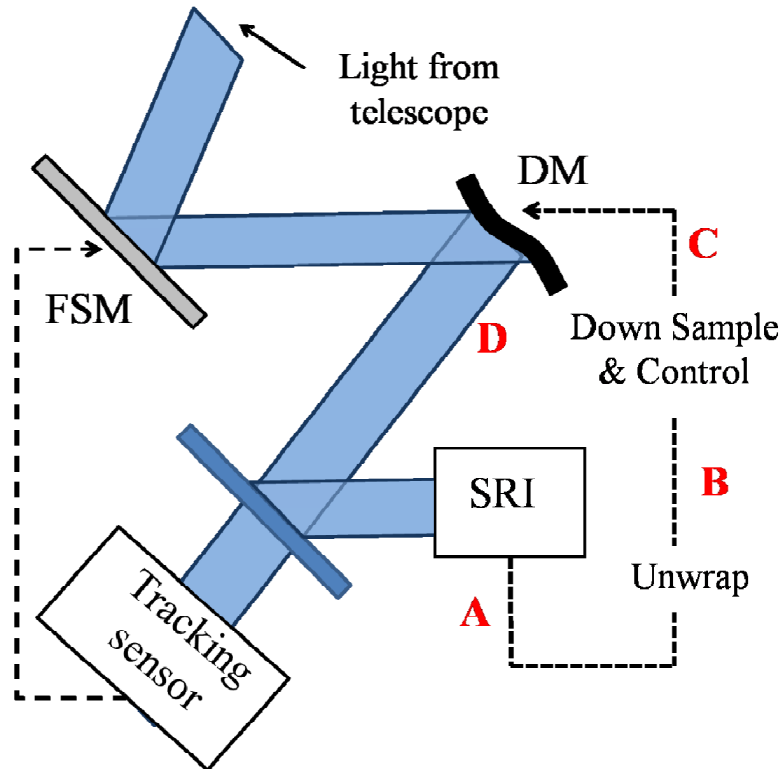


Figure 34. Closed-loop simulation architecture for studying phase-unwrapping algorithms. Light from the telescope enters the AO system, and the SRI senses the wave-front compensated by the FSM and DM. Wrapped phase from the SRI (A) is unwrapped (B), then down-sampled and applied to the controller. The commands are then sent to the DM (C). Cut length and other metrics are computed at Pt. (B), and the Strehl ratio is computed at Pt. (D).

DM commands which exceed the dynamic range of the mirror being modeled should be set to the stroke limit. The higher resolution grid represents the phase delay imparted by the DM to the incident field. In this research, a DM is modeled to represent a 22×22 -actuator, continuous-surface DM. WaveProp's linear interpolation method is used with a stroke limit of $7 \mu\text{m}$ and 20% coupling between adjacent actuators, representative of commonly available high-speed DM's. The coupling is important as it reflects the realistic inability of a continuous-surface DM to conjugate sharp discontinuities in the phase, such as branch cuts. Prior to interpolating, WaveProp accounts for the effects of coupling by convolving the DM commands with a 3×3 coupling impulse function. Only 20 actuators spanned the 2 cm beam width, while the other rows and columns fall just outside. This is done to avoid experimentally observed edge effects resulting from WaveProp interpolating the phase of the DM between actuators. For edge actuators inside the width of the beam, WaveProp cannot accurately compute the phase beyond those points. Figure 35 shows the alignment relationship between the telescope exit pupil and the DM actuators.

The next element in Fig. 34 is the 43×43 -subaperture spatial SRI for sensing high-order aberrations. Since a linear-filter control law is utilized, a high-resolution SRI with at least two subapertures per actuator is needed [5]. This configuration avoids unsensed 2π differences between actuators and has been shown to operate effectively in strong turbulence [39]. Figure 35 shows how the SRI subapertures map to the DM actuators. The phase at each actuator is determined by the eight subapertures around it and the one directly in line with the actuator. Based on Barchers' [5] research, the appropriate kernel for reducing the high-resolution SRI measurement to

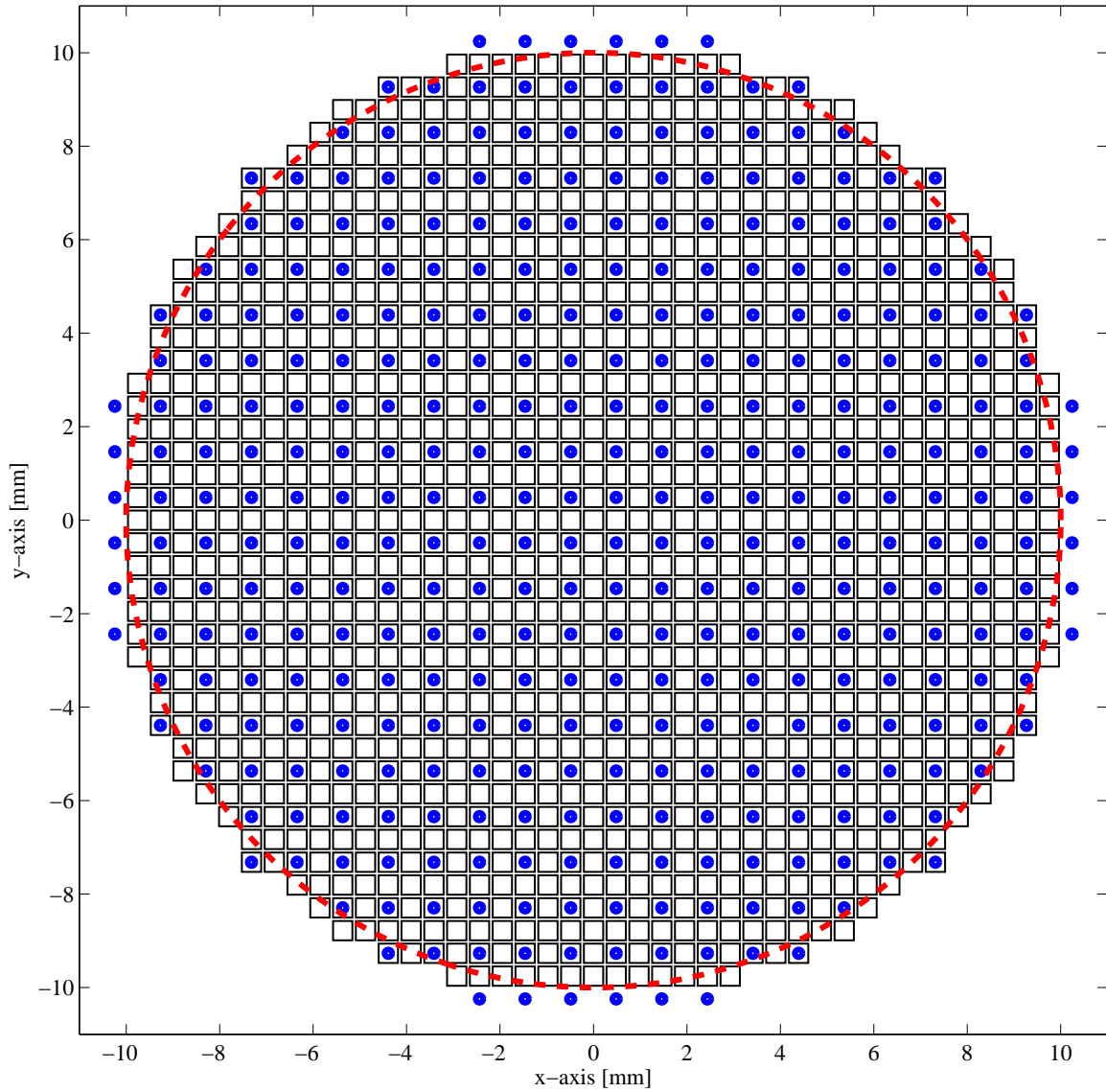


Figure 35. Alignment relationships between the telescope exit pupil (red dashed circle), controllable DM actuators (blue dots), and active SRI subapertures (black squares).

actuator resolution is given by

$$\begin{bmatrix} 0.0625 & 0.1250 & 0.0625 \\ 0.1250 & 0.2500 & 0.1250 \\ 0.0625 & 0.1250 & 0.0625 \end{bmatrix}. \quad (71)$$

This kernel is convolved with the output of the SRI, and then every other sample is picked out for the DM actuator grid. Figure 36 provides a visualization of this process. To avoid filtering the edges of the phase with the zero-valued samples outside the aperture, the phase is interpolated out to the edge of the array. It is important to note that the wrapped output of the SRI is unwrapped prior to applying the spatial filter and down-sampling. It is at this point in the simulation where different unwrapping algorithms are interchanged to see how they affect system performance. Once the phase is unwrapped, filtered, and down-sampled, a linear control law is applied to produce DM commands. The control law coefficients are $\alpha = 1$ and $\beta = 0.4$. Validation of the AO system consists of applying a vacuum-propagated field over a period of time to ensure the Strehl ratio of the corrected field reaches a value of one. In addition, a model point source is propagated through very weak turbulence. The AO system is able to achieve a Strehl ratio of approximately one.

3.1.3 Parameter Exploration.

Once the turbulence and AO system models are developed and tested, the parameter space for PCO optimization can be explored. A key characteristic of any PCO algorithm is its ability to distinguish a value of h that maximizes system performance. As previously mentioned, Venema and Schmidt focus on minimizing normalized cut length in their algorithm since Strehl ratio is not available during unwrapping [48]. This research uses a similar concept called irradiance-weighted cut length (IWCL).

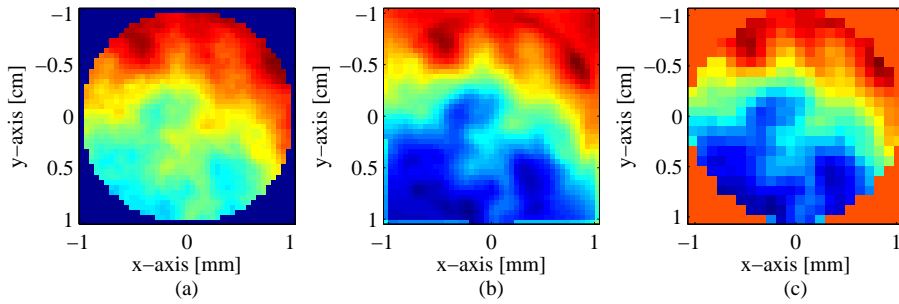


Figure 36. Conversion of SRI high-resolution output to actuator resolution. The wrapped output from the SRI (a), is interpolated out to the array edge and convolved with a kernel (b), then down-sampled to actuator resolution (c).

IWCL is the fraction of irradiance in the pupil adjacent to any branch cuts.

When choosing h in Eq. (70) to minimize IWCL and correspondingly maximize Strehl ratio, it is necessary to explore IWCL as an objective function of h . Prior to carrying out closed-loop simulations, an open-loop simulation is created for this purpose, and its diagram is shown in Fig. 37. Independent realizations of a propagated field are applied to the system 1,000 times. For each realization, the value of h is adjusted through a one-wave range (one wave or wavelength is equal to a change in phase of 2π radians) to compute the IWCL and Strehl ratio. Figure 38 shows typical results for one realization. Although this realization does not exhibit perfect anti-correlation, it does show that minimizing IWCL is a sound approach to maximizing Strehl ratio. Another important observation is that the relationship between h , IWCL, and Strehl ratio is periodic. The results are almost identical for values of h separated by one wave. This is beneficial when trying to optimize the PCO since the parameter space of h can be narrowed to a range of one wave.

A process similar to that shown in Fig. 37 is repeated 5,000 times for three different Rytov numbers between 0.6 and 1.0 at frame rates of 3 kHz and 10 kHz. The result is 15,000 different realizations ranging in turbulence strength at each frame

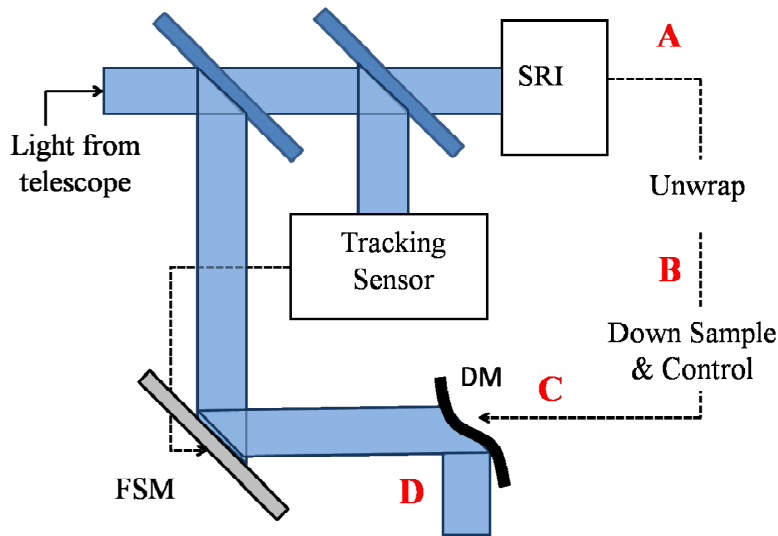


Figure 37. Open-loop simulation architecture for determining the relationship between h , Strehl ratio, and IWCL. Light from the telescope enters the AO system and the SRI senses the uncompensated wave-front. Wrapped phase from the SRI (A) is unwrapped (B), then down-sampled and applied to the controller. The commands are then sent to the DM (C) which conjugates the same field that is sensed by the SRI. IWCL is computed at Pt. (B), and the Strehl ratio is computed at Pt. (D).

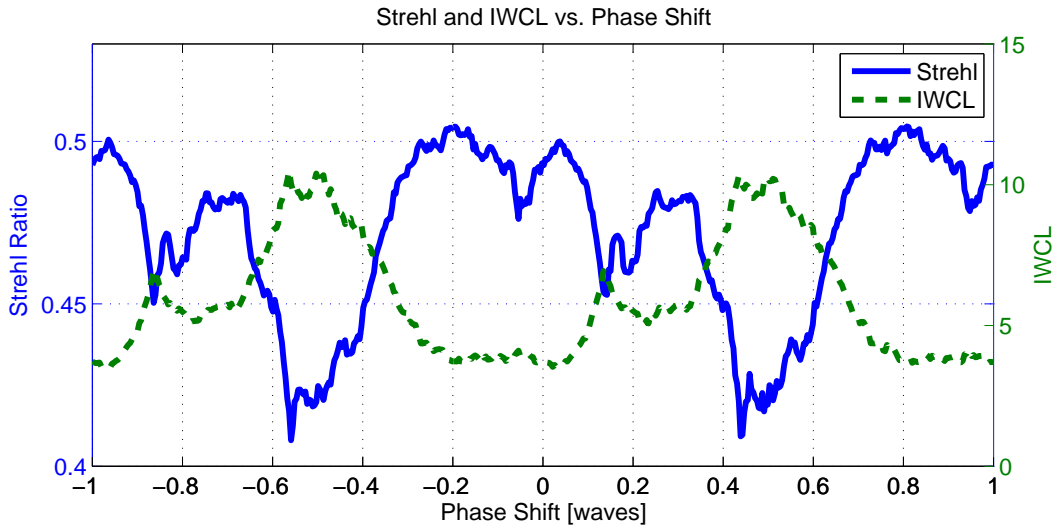


Figure 38. Strehl ratio and IWCL as a function of phase shift h for one realization with a Rytov number of 0.8.

rate for which data of h versus IWCL is analyzed. For each realization, the value of h corresponding to the lowest IWCL is recorded. Figure 39 shows the Probability Density Functions (PDF) for the optimal value of h over the 15,000 realizations at (a) 3 kHz, and (b) 10 kHz. Since both PDF's are fairly uniform, any optimization of the PCO for open-loop AO needs to search a whole period for the best value of h .

To fully understand the relationship between h and IWCL, one must also observe what happens in closed-loop AO. In Fig. 34, a LSPV+200 phase-unwrapping algorithm is used in closed-loop to evaluate 200 different values of h in the range $(-0.5, 0.5]$ waves. The values of h corresponding to the lowest IWCL are plotted versus time in Fig. 40.

The simulation is conducted at a frame rate of 3 kHz with a Greenwood frequency of 25.5 Hz, and the loop is not closed until $tf_G = 0.5$. The results indicate that the optimal value of h fluctuates over a wide range while the loop is open. Once closed, the optimal phase shift stays close to zero. Figure 41 provides examples of the h parameter space for four non-sequential frames. In closed-loop, the plots show a shape that is mostly concave-up, centered near $h = 0$. To further analyze this behavior, ten LSPV+200 closed-loop simulations of 500 frames each are executed ranging in turbulence strengths from a Rytov number of 0.2 to 1.0 and for frame rates of 1, 3, 5, 8, and 10 kHz. The resulting parameter space is shown in Fig. 42 and consist of 25 distinct conditions. The most challenging conditions are with slow frame rates and strong turbulence, as indicated by the arrow.

Figure 43 shows an example of the PDF's for the optimal value of h at a frame rate of 5 kHz, broken out by turbulence strength. For Rytov numbers of 0.2 and 0.4, the PDF's are weighted towards a negative h value. At these turbulence strengths, branch points just begin to appear, causing the PDF to shift to the left. When branch points are not present, the rotational component is nonexistent. Accordingly,

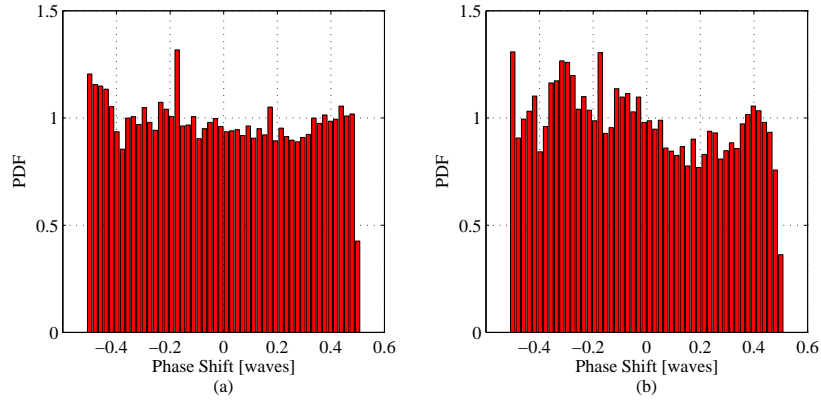


Figure 39. PDF for the optimal value of h between -0.5 and 0.5 wave for open-loop simulations at (a) 3 kHz, and (b) 10 kHz.

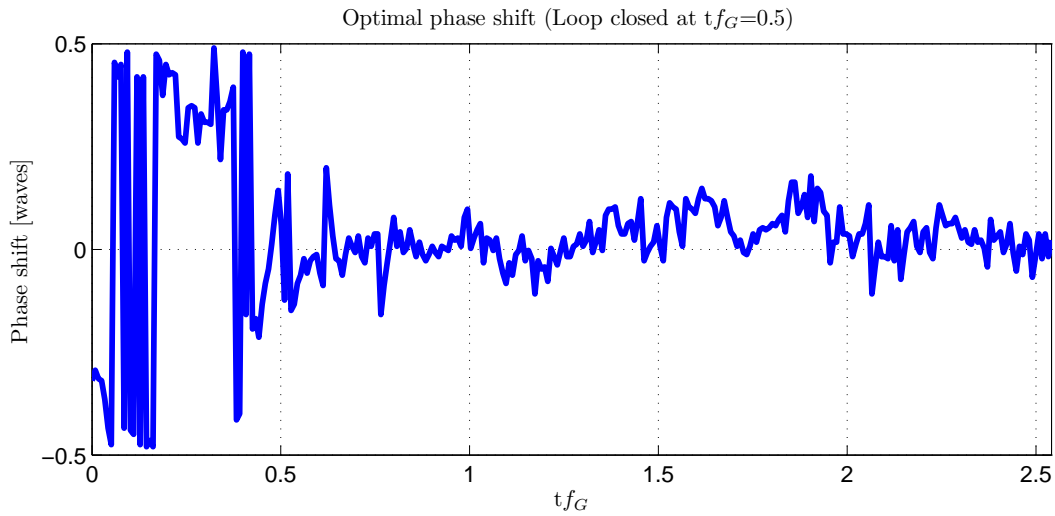


Figure 40. Phase shift h chosen by LSPV+200 in a closed-loop 100 ms simulation. The loop is not closed until $tf_G = 0.5$.

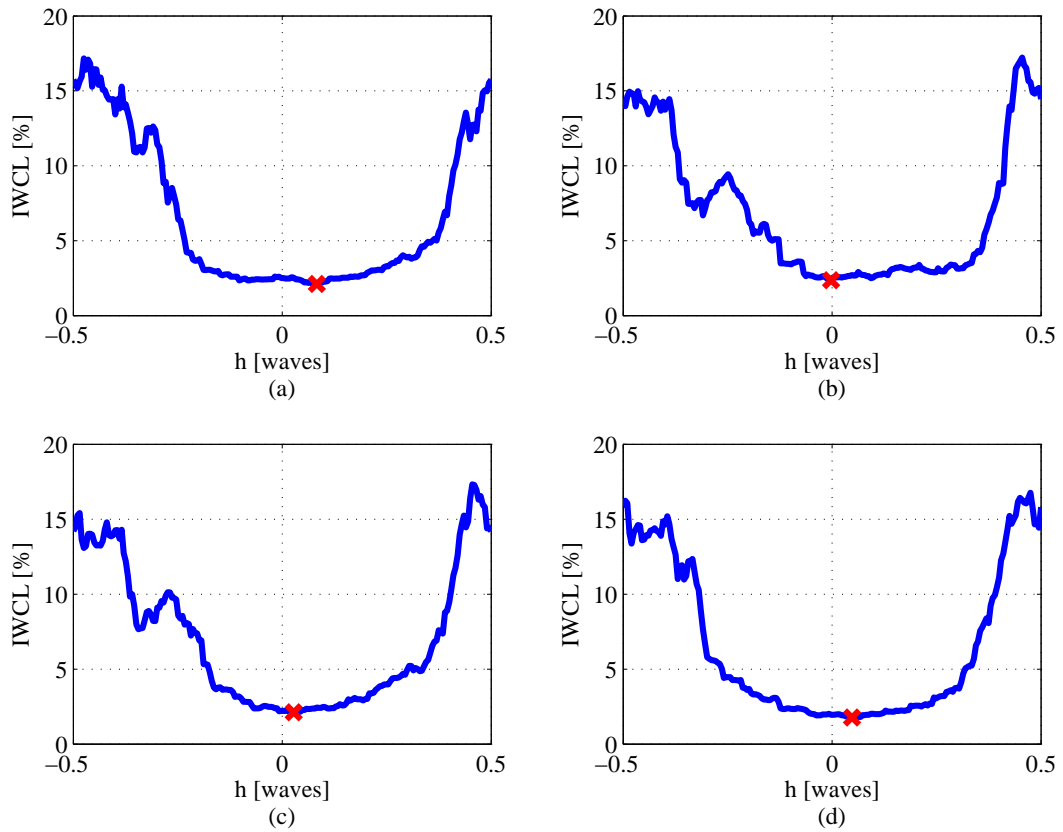


Figure 41. Examples of IWCL vs. h for four different non-sequential frames of a closed-loop simulation with a Rytov number of 0.8 and a 5 kHz frame rate. The red x marks the point of lowest IWCL found by the LSPV+200 unwrapping algorithm.

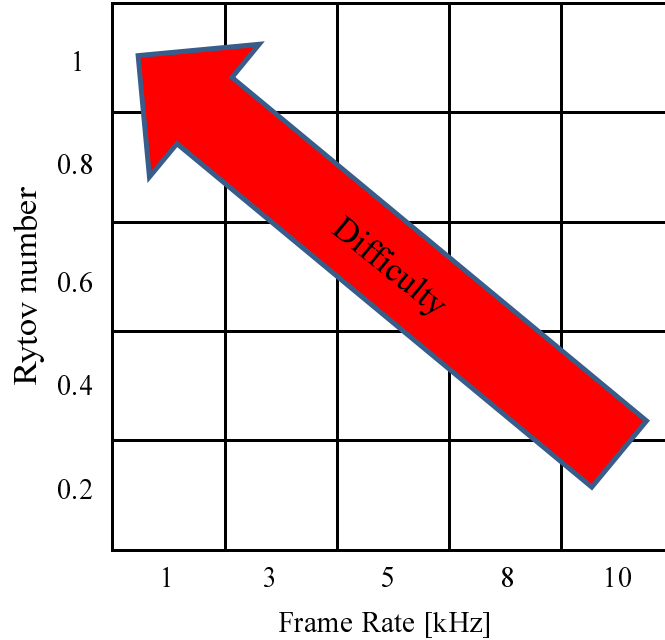


Figure 42. Simulation parameter space broken out by frame rate and Rytov number. The most challenging conditions are with slow frame rates and strong turbulence, as indicated by the arrow.

the IWCL is zero for all values of h . The LSPV+200 algorithm calculates IWCL at 200 points, stores that information, and chooses the minimum value. When all values of an array are zero, MATLAB[®] chooses the first component as the minimum. In this case, the first component corresponds to $h = -0.5$. As shown in Fig. 44, when there are only a few branch points present, the IWCL is approximately flat for a range of values surrounding $h = 0$, and rises sharply as $|h|$ increases. Although several optimal values of h are centered around zero in these cases, MATLAB[®] chooses the first instance as the minimum. This is the reason that the lower turbulence cases shown in Fig. 43 are weighted toward $h = -0.5$.

Since this behavior is a result of MATLAB[®] and not a physical phenomenon, the two lower turbulence simulations are excluded for the sake of computing an overall PDF. Figure 45 shows the closed-loop PDF's as a function of frame rate when only Rytov numbers of 0.6, 0.8, and 1.0 are considered. The PDF's are very similar for

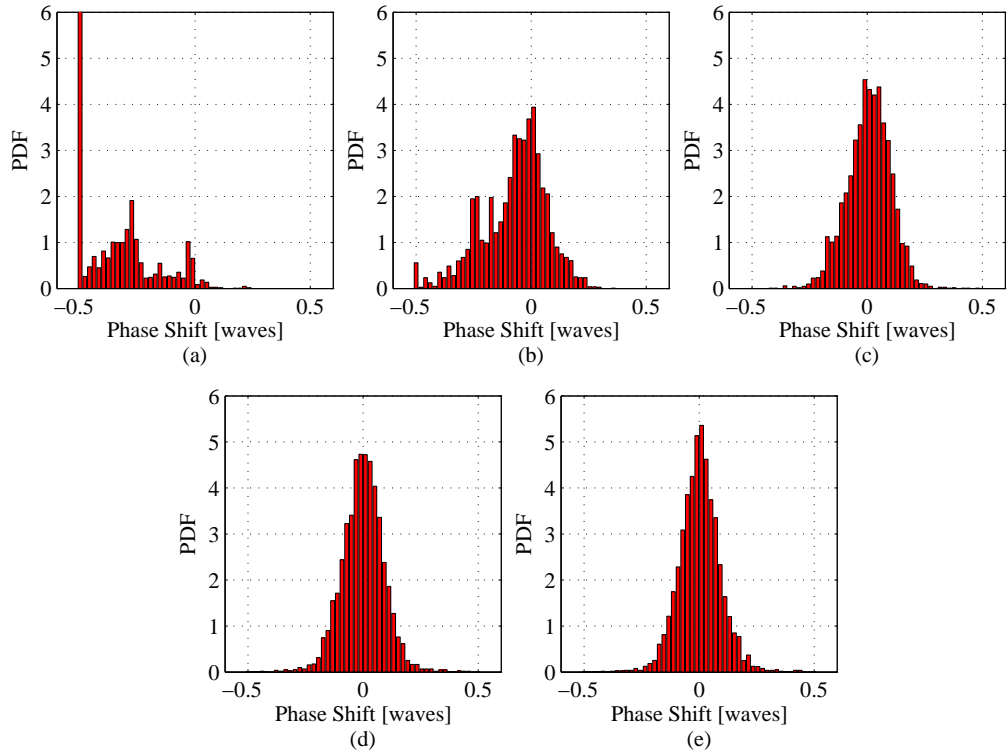


Figure 43. PDF's for the optimal value of h between -0.5 and 0.5 wave for closed-loop simulations (frame rate of 5 kHz) at Rytov numbers of (a) 0.2, (b) 0.4, (c) 0.6, (d) 0.8, (e) 1.0.

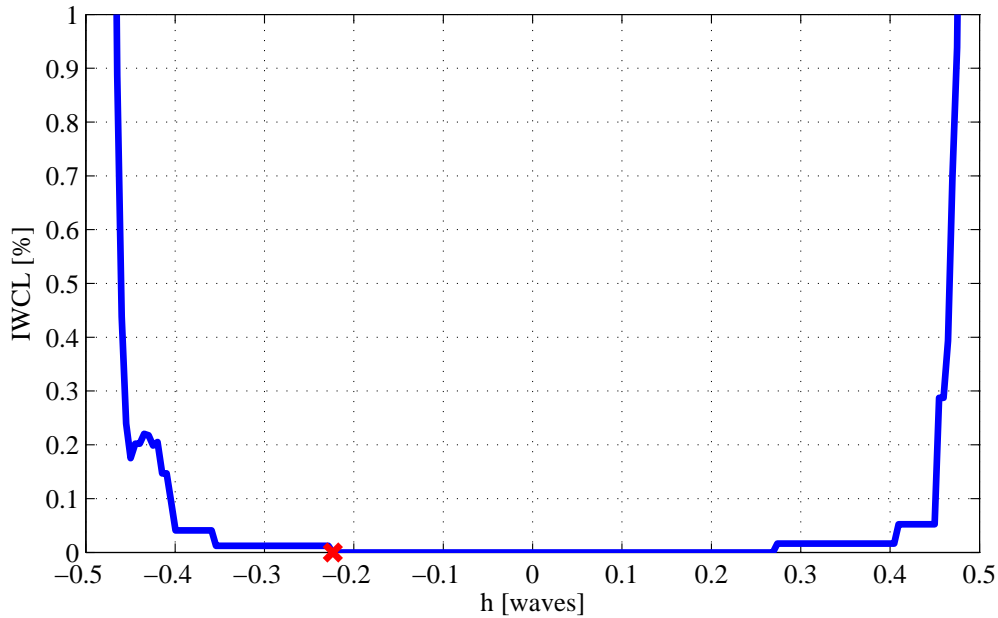


Figure 44. IWCL as a function of h for weak turbulence (Rytov number of 0.2, 5 kHz frame rate). The red x marks the lowest IWCL found by the LSPV+200 phase-unwrapping algorithm.

each frame rate, with the exception of Fig. 45 (a), which is slightly more spread out. This allows all the data to be combined into one general PDF that closely represents the optimal value of h in any of the simulations.

Figure 46 shows the general PDF, along with two distributions considered for an analytical model. The Gaussian PDF matches well near $h = 0$, but approaches zero quickly, leaving little probability of h falling near the edges. The Cauchy-Lorentzian shown has been modified by subtracting a constant value so that it crosses the x -axis at $h = -0.5$ and $h = 0.5$, and by a scaling term to ensure the total probability under the curve equals one. It more closely follows the gradual taper of the simulated data and is given by

$$P[h] = \begin{cases} \frac{1.25}{\pi\gamma\left[1+\left(\frac{h}{\gamma}\right)^2\right]} - 0.1131, & |h| \leq 0.5, \\ 0, & |h| > 0.5, \end{cases} \quad (72)$$

where $P[h]$ is the zero-mean PDF evaluated at h , and γ is the scale parameter equal

to 0.0799. From this PDF, it is clear that closing the loop shifts the optimal h value towards zero. This means that an optimized PCO algorithm could narrow its search space after the loop has had sufficient time to reach steady-state. In addition to the PDF, it is useful to determine the cumulative distribution function (CDF). Integrating Eq. (72) to find the probability of h being less than some number \hat{h} gives

$$P[h \leq \hat{h}] = \begin{cases} \left[-0.1131\hat{h} + 0.398 \operatorname{Tan}^{-1}(\hat{h}, \gamma) \right] + 0.5, & |\hat{h}| \leq 0.5, \\ 0, & |\hat{h}| > 0.5. \end{cases} \quad (73)$$

Figure 47 shows Eq. (73) plotted along with the experimental CDF.

Large sudden changes in h can significantly alter the location of branch cuts. With an integral controller, closed-loop DM commands consist of a weighted sum of previous commands and commands computed from the residual error in the current frame. This is satisfactory for a slowly evolving phase, but when branch cuts move significantly from frame-to-frame, 2π discontinuities suddenly accumulate in more areas across the DM. This can lead to sudden drops in Strehl ratio. Figure 48 plots Strehl ratio and the optimal h chosen by a LSPV+200 and LSPV+1 unwrapper versus time. The large sudden drops in Strehl ratio of the LSPV+200 unwrapper at approximately $tf_G = 3$ correspond to the relatively large changes in h . The value of h never changes in the LSPV+1 unwrapper, so it avoids the Strehl ratio drops. Since the LSPV+200 can be considered to have found the lowest IWCL for each frame, Fig. 48 shows that minimizing IWCL does not always increase closed-loop AO system performance. Rather, it is a combination of both minimizing the irradiance around branch cuts and avoiding sudden changes in branch cut location from frame-to-frame when using an integral controller.

Another interesting effect of the parameter h is how it changes the distribution of wrapped phase values in the non-LS component. Figure 49 shows how increasing h

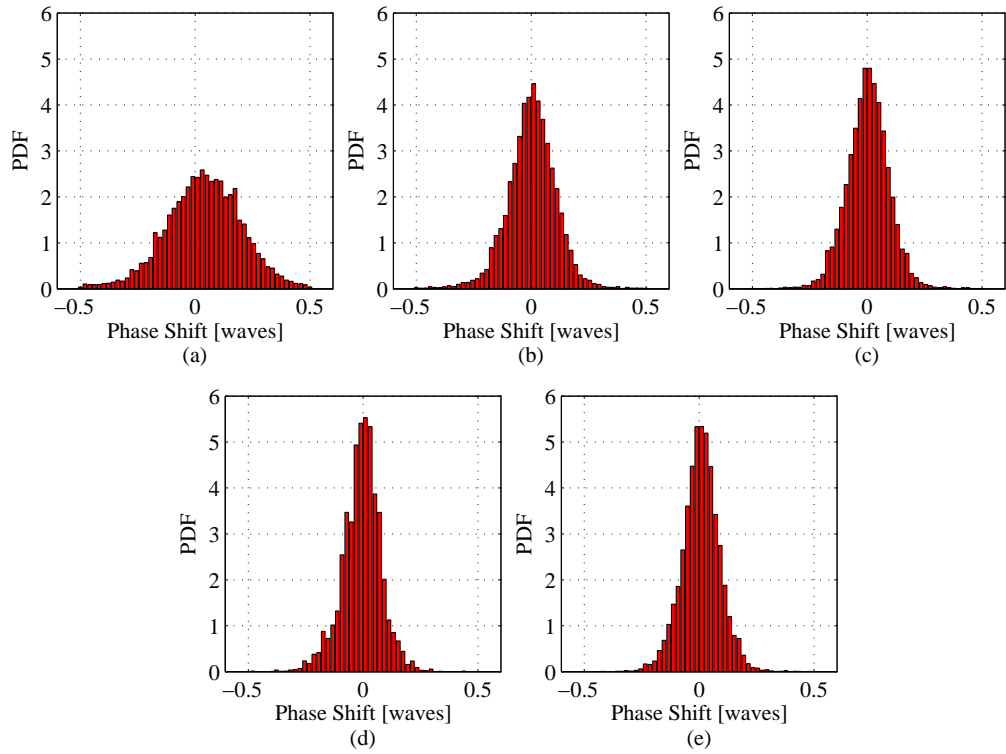


Figure 45. PDF's for the optimal value of h between -0.5 and 0.5 wave for closed-loop simulations with Rytov numbers of 0.6, 0.8, and 1.0 at frame rates of (a) 1 kHz, (b) 3 kHz, (c) 5 kHz, (d) 8 kHz, (e) 10 kHz.

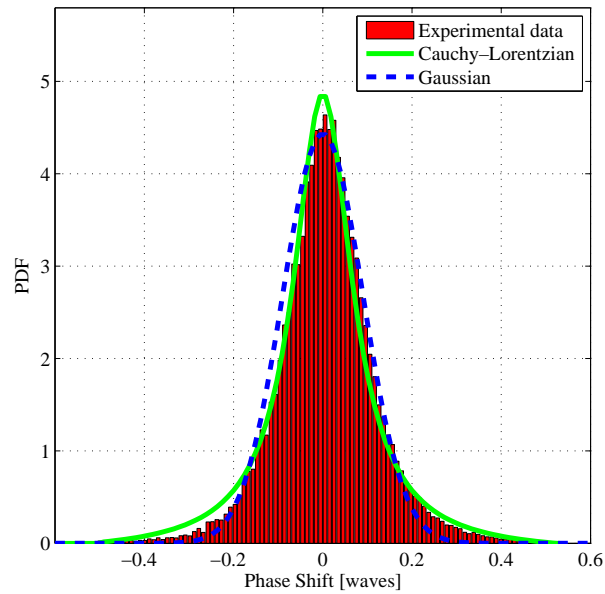


Figure 46. PDF for the optimal value of h between -0.5 and 0.5 wave for all data from closed-loop simulations with Rytov numbers of 0.6 , 0.8 , and 1.0 and all frame rates. Both Cauchy-Lorentzian and a Gaussian distribution have been fitted to the data.

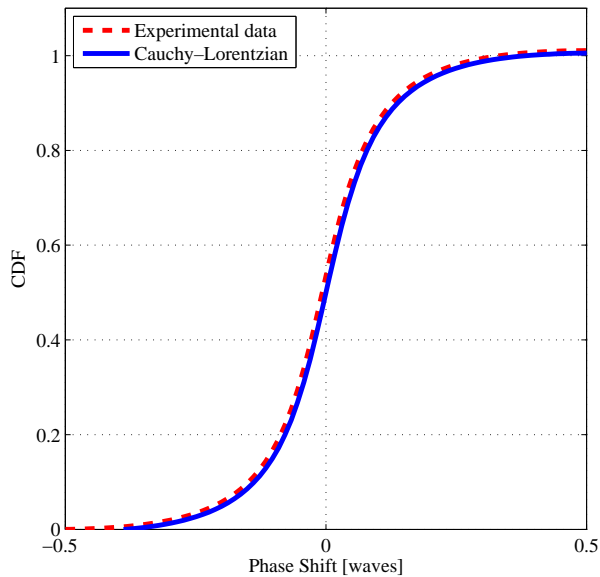


Figure 47. CDF for the optimal value of h between -0.5 and 0.5 wave for all data from closed-loop simulations with Rytov numbers of 0.6 , 0.8 , and 1.0 and all frame rates.

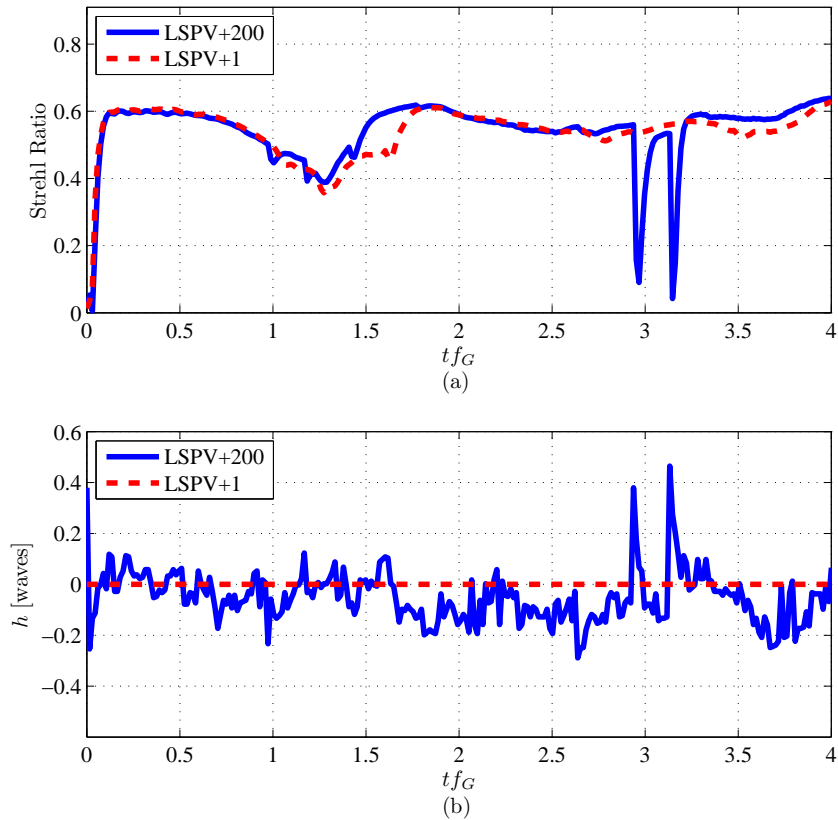


Figure 48. Strehl ratio (a) and optimal h chosen (b) versus time for both LSPV+1 and LSPV+200. The large drops in Strehl ratio for the LSPV+200 unwrapper occur when there is a large jump in the value of h at approximately $tf_G = 3$. The LSPV+1 never varies h and correspondingly avoids the sudden drop in Strehl ratio.

causes the distribution to shift to the left. This is due to the fact that increasing h means subtracting the same value from all samples. The important thing to note is that when there are many samples in the phase near the wrapping boundaries, as seen in plots (a) and (b), there is more opportunity to experience a one-wave discontinuity as seen in the corresponding high IWCL. An algorithm may use this information to choose h such that the phase at the boundaries is minimized.

3.2 Algorithm Design

Once attributes of the parameter space are known, the information is used in an attempt to make more effective phase unwrappers. This section describes the various algorithms designed based on that information. It also describes the other unwrappers that are used for a comparison.

3.2.1 Optimization Attempts.

Over the course of this research, thirteen PCO optimizing algorithms are developed and tested. Two of these algorithms are not practical because they require too many computations, and are used only as tools for exploring objective functions. Tables 2 and 3 list each algorithm along with their attributes. The second and third columns list the number and type of objective-function evaluations for each method. In cases where two objective function types are listed, the primary is listed first. The next column displays information being used by the unwrapper to help make its decision. Entries include histograms of the ϕ_{non-LS} values (in one case, weighted by the irradiance), the h value from the previous frame, and the CDF. The *search type* column refers to how each algorithm narrowed in on a value of h . Additional detail on the various techniques can be found in Sec. 3.2.2. The *starting point* column indicates the point at which each search begins. Algorithms with $h_{prev}/2$ start halfway

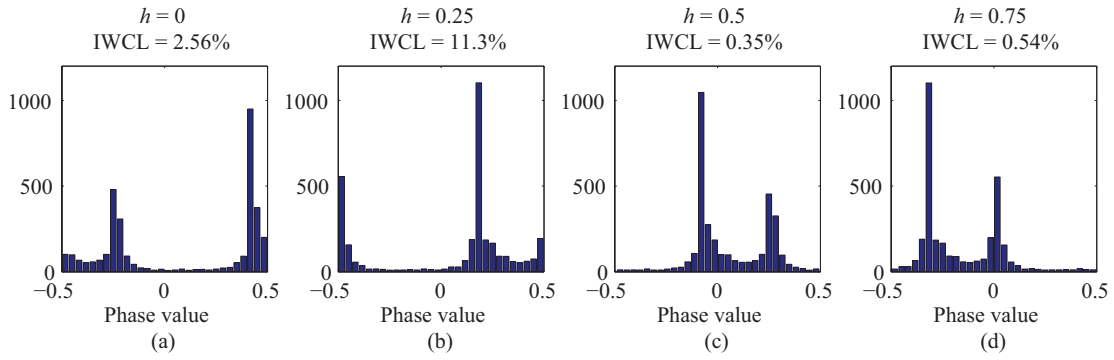


Figure 49. Histogram showing the distribution of wrapped phase values in the non-LS component for one open-loop realization evaluated at (a) $h = 0$, (b) $h = 0.25$, (c) $h = 0.5$, (d) $h = 0.75$.

between the previous frame's h value and zero to avoid large phase shifts in ϕ_{non-LS} . The second-to-last column notes the range of each algorithm. Iterative techniques have two rows, one for each iteration. The use of \pm indicates that *seeds* are distributed around the starting point at the values listed. For the sake of continuity and reproducibility, all MATLAB[®] code for the algorithms presented in Tables 2 and 3 can be found in App. A.

Table 2. Descriptions of optimized algorithms.

Alg.	Evals. ϕ_{non-LS}	Obj. Fun.	Info.	Search type	Start pt.	Range [waves]	Comment
Opt1	10	IWCL		Iterative probing	0	$\pm 0.300, \pm 0.100$ $\pm 0.070, \pm 0.050$	Basic search using non-uniformly spaced seeds.
Opt2	8	IWCL	Hist.	Comparison/ probing	0	0 ± 0.030 $h_{hist} \pm 0.100$ $h_{hist} \pm 0.060$	Minimizes ϕ_{non-LS} at wrapping boundaries to estimate h . It then searches that area and compares results to three points centered at zero.
Opt3	10	IWCL, $\text{Var}(h_{diff})$	Prev. h	Iterative probing	$\frac{h_{prev}}{2}$	$\pm 0.150, \pm 0.050$ $\pm 0.075, \pm 0.025$	Modified version of Opt1 with half the range. Similarly to Opt6, Opt3 starts in between previous h and zero.
Opt4	1		Irr. weighted hist.			1	$h = \frac{\int \int I(x,y) \phi_{non-LS} dx dy}{\int \int I(x,y) dx dy}$ where I is the irradiance over the aperture and ϕ_{non-LS} is the non-LS phase.
Opt5	5		Hist			1	Uses the median value of the ϕ_{non-LS} for single h
Opt6	10	IWCL, $\text{Var}(h_{diff})$	Prev. h	Iterative probing	$\frac{h_{prev}}{2}$	$\pm 0.300, \pm 0.100$ $\pm 0.070, \pm 0.050$	Same as Opt1 but starts between previous h and zero.
Opt7	200	IWCL+ Corr.	Previous ϕ_{non-LS}	Brute force		1	Not a practical unwrapper. Opt7 is being used to test the inclusion of correlation data between ϕ_{non-LS} from frame-to-frame. Opt7 maximizes $\frac{\rho}{I_{WCL}}$, where ρ is the correlation coefficient of the current ϕ_{non-LS} to the previous ϕ_{non-LS} .

Table 3. Descriptions of optimized algorithms (cont.).

Alg.	Evals.	Obj. Fun.	Info. used	Search type	Starting pt.	Range [waves]	Comment
Opt8	200	Corr.	Previous ϕ_{non-LS}	Brute force		1	Not a practical unwrapper. Opt8 is similar to Opt7 except that it does not consider IWCL. It maximizes ρ only.
Opt9	8	IWCL, $\text{Var}(h_{diff})$	CDF+ prev. h	Probing	Previous h	0.4	Creates bins of equal probability across the range centered at the starting point. It then searches the center of each bin for the lowest IWCL.
Opt10	8	IWCL		Golden ratio search (GRS)/parabolic interp.		1	Uses MATLAB [®] 's <code>fminbnd</code> function with <code>MaxFunEvals=4</code> , and <code>MaxIter=2</code> to minimize IWCL.
Opt11	8	IWCL	CDF	Iterative probing	0	0.4	Same as Opt9 except it always starts at zero.
Opt12	9	IWCL		GRS/parabolic interp.		0.6	My version of MATLAB [®] 's <code>fminbnd</code> function.
Opt13	9	IWCL	CDF	GRS/parabolic interp.		0.8	Uses a Golden section search weighted by the CDF. Instead of searching the parameter space in terms of ratios of length, Opt13 uses ratios of probability.

3.2.2 Key Algorithms.

Venema and Schmidt’s LSPV+4 utilizes one of the simplest search techniques, which is to evaluate or probe a function at several locations, or *seeds*. A logical extension of this technique is to pick the seed locations based on information obtained during the parameter exploration, such as the PDF. A second iteration of this search technique focusing on the local area around the seed with the lowest IWCL could find a more optimal value of h . Optimizations one (Opt1), three (Opt3), and six (Opt6) use these ideas, and are essentially LSPV+10 algorithms. The distribution of seeds and iterative process of Opt6 is illustrated in Fig. 50. The wide search in the first iteration improves the ability to find a low IWCL when the AO system has not yet reached steady-state. To improve the performance in closed-loop, the algorithm always begins the search at $h_{prev}/2$, the halfway point between the previous value of h and zero. This attribute alone distinguishes Opt6 from Opt1. It gives preferential weight to $h = 0$, while avoiding large frame-to-frame jumps in h , which can happen with Opt1. Opt6 evaluates IWCL at two points (± 0.1 and ± 0.3 waves) on either side of $h_{prev}/2$ in the first iteration. The second iteration is identical to the first, except that it begins with the value of h having the lowest IWCL found in the first step, and the search space narrows (± 0.05 and ± 0.07 waves). The only difference between Opt3 and Opt6 is that the range in which the seeds are distributed for Opt3 is cut in half. Reducing the range limits large changes in h between frames.

Optimization Two (Opt2) uses a similar approach to find the optimal value of h . With eight total evaluations of ϕ_{non-LS} , it consists of two separate processes, a LSPV+5 search and a LSPV+3 search. Neither process is iterative, and the algorithm chooses the best result from either search. The key difference between Opt2 and the algorithms in the previous paragraph is that Opt2 takes advantage of physical information regarding the non-LS phase to estimate a starting location for the LSPV+5

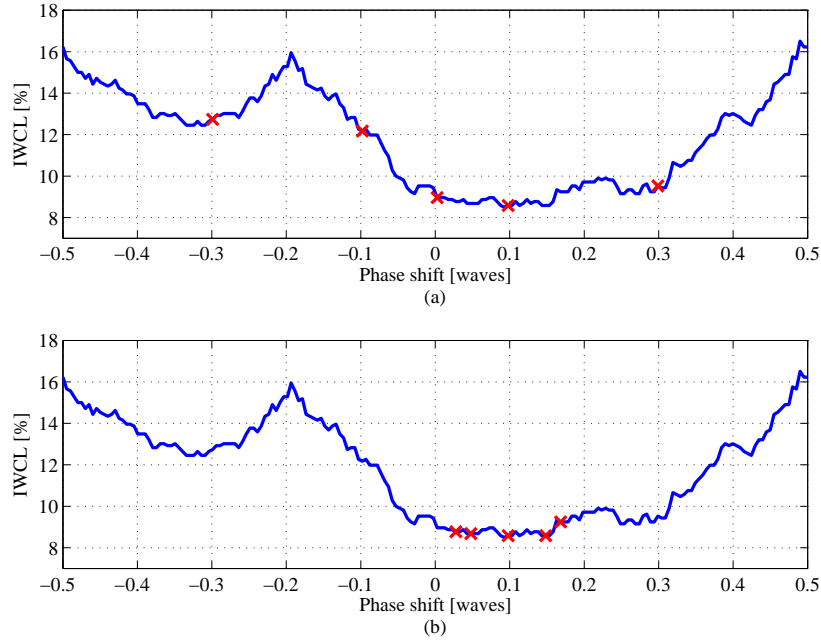


Figure 50. Distribution of *seeds* for Opt6. In the first iteration (a), the *seeds* are distributed symmetrically around $h_{prev}/2$ (zero in this case). The seed with the lowest IWCL becomes the center for the second, more narrow iteration (b).

search. The LSPV+5 process can be considered a global search, and is able to select any value of h in the one-wave range. This improves performance when the loop has not reached steady-state. It utilizes the concept illustrated in Fig. 49 to minimize the phase at the wrapping boundary for ϕ_{non-LS} . The wrapped phase values are sorted into 30 bins to create a distribution as shown in Fig. 49. The distance of the smallest bin from the closest edge is calculated. This represents the value h_{hist} which minimizes phase at the wrapping boundary.

In the second part of this process, a local search similar to a single iteration in Opt6 is executed with h_{hist} as the starting point. The *seeds* for this search are distributed at $h_{hist} \pm 0.1$, ± 0.06 . Evaluation of IWCL at these seeds concludes the LSPV+5 portion of Opt2. The LSPV+3 process focuses on optimizing performance after the loop has reached steady-state. It consists of searching three points near zero, $h = 0 \pm 0.05$. The results of the two processes are compared, and the process

with the lower IWCL is chosen.

Both Optimizations Seven (Opt7) and Eight (Opt8) are LSPV+200 probing algorithms with uniformly distributed seeds. Although the large number of ϕ_{non-LS} evaluations make both algorithms impractical for real-time implementation, they are designed to test the concept of minimizing changes in ϕ_{non-LS} from frame-to-frame. Instead of only minimizing IWCL, both algorithms compute the correlation coefficient ρ between ϕ_{non-LS} of the previous frame and the current frame for each shift in h . Opt8 chooses the value of h corresponding to the highest ρ . Opt7 incorporates ρ with IWCL and attempts to maximize $\rho/IWCL$.

Opt6 and Opt3 use the mean of the PDF to weight their searches towards $h = 0$. Another logical extension to this concept is to incorporate more of the distribution information into the probing process. Optimizations Nine (Opt9) and Eleven (Opt11) are LSPV+8 algorithms which utilize the CDF to more efficiently probe the parameter space for values of h . Both algorithms separate a 0.4-wave range into bins of equal probability and place the *seeds* at the center of each bin. The 0.4-wave range is chosen based on the width of the PDF shown in Fig. 46, and based on trial simulations. Opt11 centers the window around $h = 0$ each time, and therefore can never choose values of $|h|$ greater than 0.2. Opt9 on the other hand, centers the window around the previous value of h in an attempt to minimize large changes from frame-to-frame. This algorithm is able to reach values of $|h|$ greater than 0.2, as opposed to Opt11.

Finally, Optimizations Ten (Opt10), Twelve (Opt12), and Thirteen (Opt13) abandon the basic probing technique for a more sophisticated search method. The three are LSPV+8/9 algorithms, each utilizing a **golden ratio search** (GRS) [29] and parabolic interpolation to find h . As seen in Fig. 41, the IWCL parameter space is often concave-up. This allows the use of parabolic interpolation to estimate the optimal value of h with only three points known. Although Fig. 41 and the PDF

indicate that a simple parabolic interpolation by itself may be used to find a minimum near $h = 0$, Fig. 51 shows that at times, the parabola can significantly change shape and location. This change foils a simple parabolic interpolation. Such behavior requires general knowledge of the minimum location prior to applying a parabolic interpolation. This general knowledge is gained by using a GRS.

GRS is one of the most efficient techniques used to find local minima of a one-variable function $f(x)$ by evaluating it a minimal number of times [29]. The method is named for the ratio being used in the division of the parameter space. Figure 52 shows an example of how GRS decomposes a search space into iteratively smaller segments containing local minima. The function $f(x)$ is evaluated at the edges of the range (x_1 and x_3), and near the middle point x_2 . A fourth point x_4 is chosen in the larger of the two new segments d_{12} and d_{23} . If $f(x_4)$ is less than $f(x_2)$, then the algorithm assumes that the minimum is in the range d_{23} . This then becomes the new search space. Conversely, if $f(x_4)$ is greater than $f(x_2)$, the new search space is $d_{12} + d_{24}$. Because GRS's assume the worst-case scenario of a uniformly distributed minimum, both possible new search spaces are equal in length to maximize the likelihood of finding the minimum [35]. Each new iteration reuses three of the four points required to repeat the process on the new segment. The fourth point is always evaluated in the larger of the two sections separated by the three points. Applying the Golden Ratio allows reuse of points. By requiring that

$$\frac{d_{23}}{d_{12}} = \frac{d_{43}}{d_{24}} = \frac{d_{14}}{d_{24}} = \frac{1 + \sqrt{5}}{2} = 1.618033988\dots, \quad (74)$$

the GRS obtains maximum reuse of points, thereby reducing the computational burden of the minimization. At each step, the three points are used in a parabolic

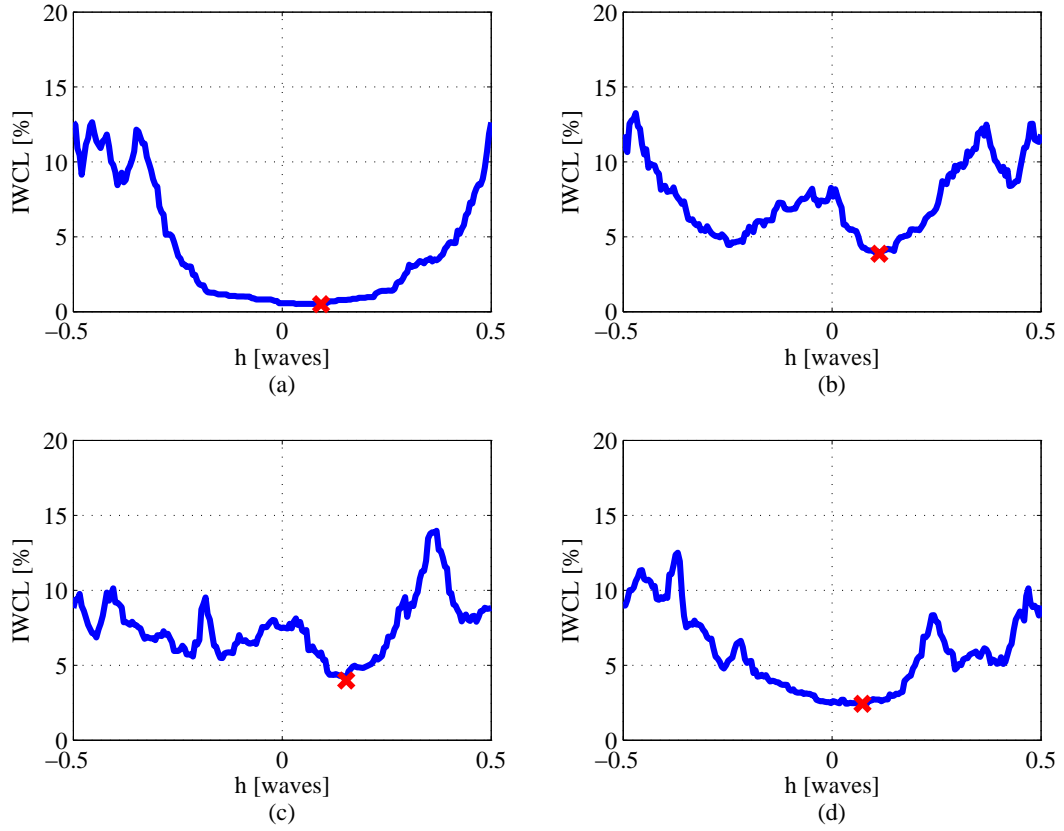


Figure 51. Additional examples of IWCL vs. h for four different non-sequential frames of a closed-loop simulation with a Rytov number of 0.8 and a 5kHz frame rate. The red x marks the point of lowest IWCL found by the LSPV+200 unwrapping algorithm. Unlike Fig. 41, plots (a) through (d) show that at times, the parabolic minimum can significantly change shape and location.

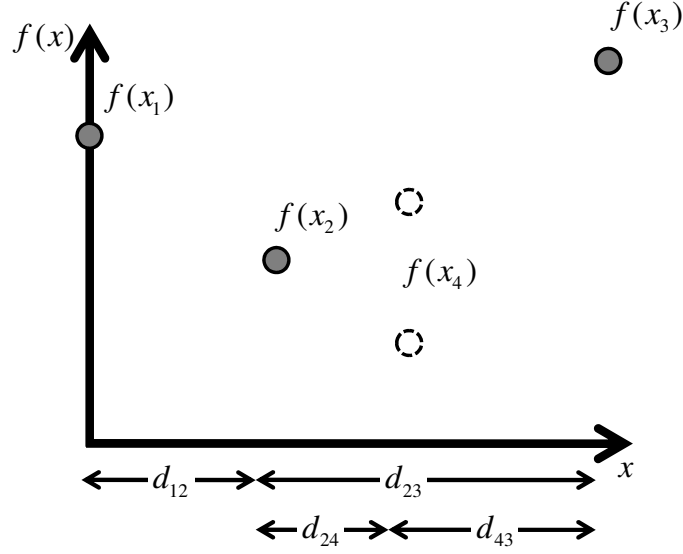


Figure 52. The golden ratio search (GRS) divides the parameter space into sections corresponding to the golden ratio. The algorithm then decides which section contains the minimum based on the relative values of $f(x_2)$ and $f(x_4)$. If $f(x_4)$ is less than $f(x_2)$, the new search space will be d_{23} , otherwise it will be $(d_{12} + d_{24})$.

interpolation given by

$$x_{min} = x_2 + \frac{1}{2} \frac{(x_2 - x_1)^2 [f(x_2) - f(x_3)] - (x_2 - x_3)^2 [f(x_2) - f(x_1)]}{(x_2 - x_1) [f(x_2) - f(x_3)] - (x_2 - x_3) [f(x_2) - f(x_1)]}, \quad (75)$$

where x_{min} is the location of the minimum for the parabola running through the three points [35].

Opt10 uses the GRS/parabolic interpolation built into MATLAB[®], while Opt12 is developed specifically for this research to avoid any unknown processes that may take place in the MATLAB[®] function. Opt13 uses GRS/parabolic interpolation as well, but unlike Opt10 and Opt12, it incorporates information from the CDF. Normally, the GRS decomposes the search space into segment lengths proportional to the golden ratio so that each possible new search space is of the same length. In the case of a uniform distribution, this ensures equal chances of finding the minima, regardless of the segment chosen [35]. Opt13 decomposes the search space into intervals of probability proportional to the golden ratio. Referring back to Fig. 52, Opt13 chooses

points so that

$$\frac{\int_{i_{23}} P[h]dh}{\int_{i_{12}} P[h]dh} = \frac{\int_{i_{43}} P[h]dh}{\int_{i_{24}} P[h]dh} = \frac{\int_{i_{14}} P[h]dh}{\int_{i_{24}} P[h]dh} = \frac{1 + \sqrt{5}}{2} = 1.618033988\dots, \quad (76)$$

where i_{xy} is the interval between points x and y and $P[h]$ is the PDF of h given by Eq. (72). This ensures that any possible new search space has equal probability (not equal length), and it also allows for point reuse, making the modified GRS computationally efficient. This augmentation of the basic GRS transforms the deterministic search technique into one that can be used in the stochastic realm, and is unique to this research.

3.2.3 Algorithm Comparison.

Venema and Schmidt conducted a comparison of LSPV+4 to various unwrapping algorithms designed to deal with branch points [48]. Their results show that the algorithm outperforms LSPV+1, Goldstein’s algorithm, and Fried’s SmoothPhase. Using a weighted LS unwrapper for the irrotational component is shown to yield a lower normalized cut length, but at an unacceptable computational increase. WaveProp’s Xphase algorithm outperforms LSPV+4, but also at a high computational cost. It is important to note that these results are obtained using an exponential control law prior to unwrapping. This research uses a more preferable linear control law after unwrapping, resulting in a smooth response to disturbances [5].

To ensure the optimized algorithms are evaluated against comparable unwrappers, closed-loop simulations are conducted using the setup described in Sec. 3.1. The comparison began by considering Goldstein’s [17], Fried’s SmoothPhase [13], WaveProp’s Sphase and Xphase, LS (no addition of the ϕ_{non-LS} component), LSPV+1, LSPV+4, and a Quality-Guided (QG) unwrapping algorithm [17]. In the initial evaluation, Goldstein’s, WaveProp’s Sphase, and the QG algorithms have an excessive computa-

tional burden unacceptable for closed-loop AO and thus are dismissed. After further evaluation, the performance of Fried’s SmoothPhase and the LS algorithm are observed to be much poorer than the remaining unwrappers, so they are also dismissed. The results of this research differ from Venema and Schmidt in that Xphase is observed to be faster than LSPV+4. This may be due to the fact that Venema and Schmidt used Sphase, a more robust version of Xphase which requires more time to execute. Their results claim that Xphase on average, lowered the IWCL the most, so it is used in this research to compare against all algorithms developed. It is the only non-PCO algorithm thoroughly being tested. The non-optimized but practical PCO algorithms for compare new unwrappers against include LSPV+1 and LSPV+4. Two LSPV+200 unwrappers are also simulated to examine whether significantly increasing the number of h evaluations can increase system performance. LSPV+200(w) evaluates 200 points across a one-wave period of h , while LSPV+200(n) evaluates the same number of point across a much narrower region (0.05 waves) centered at the previous value of h .

The closed-loop simulations are conducted across a range of conditions given in Table 1. The following list summarizes the objective functions and metrics for use during simulations. Objective functions refer to the function which a particular algorithm is designed to maximize or minimize. A metric refers to attribute being used to compare performances.

- *Strehl Ratio (Metric)*. The metric Strehl ratio is the most important measure of AO performance, and its improvement is the sole purpose of optimizing the PCO. One method of comparing algorithms is to examine the Strehl ratio CDF’s. These easily show the probability of Strehl ratios falling below a given threshold. This method would be a useful comparison for systems which require performance above a threshold, such as AO for laser communication. The dif-

ferences in CDF's can be subtle and vary over the range of Strehl ratios, making it difficult to compare overall performance. A single-valued statistic of Strehl ratio representing performance over time would be ideal for easy comparisons. However, simply taking the time-averaged Strehl ratio for each unwrapper does not reflect sudden, undesirable drops in performance as shown in Fig. 48. Since it is desired to have a high and steady Strehl ratio, normalized variance is being used to compare algorithms overall performances. Normalized variance σ_{norm}^2 is given by

$$\sigma_{norm}^2 = \frac{\sigma_S^2}{\langle S \rangle^2}, \quad (77)$$

where σ_S^2 is the variance of the Strehl ratios S for each set of turbulence conditions and frame rate given in Table 1 (25 total). This statistic of the Strehl ratio penalizes algorithms that result in large Strehl-ratio fluctuations. As discussed in Sec. 2.3.3.5, Strehl is not available during the unwrapping process to help choose the optimal value of h . It can only be used afterwards to measure performance and is therefore considered a metric.

- *IWCL (Obj. Fun. and Metric)*. IWCL is shown to have a high anti-correlation to Strehl ratio. Since it can be computed during the unwrapping process and requires relatively few floating point operations, it is an ideal objective function to be used for optimizing the PCO. Since most of the unwrappers are designed solely to reduce IWCL, it can also be used as a performance metric to observe the effectiveness of various minimization techniques. For this reason it is measured across the range of conditions for each algorithm after unwrapping, but prior to down-sampling.
- *Correlation ρ (Obj. Fun.)*. Correlation is being used only by Opt7 and Opt8 as an objective function to minimize changes in ϕ_{non-LS} between frames in

an attempt to avoid a build up of branch cuts on the DM as discussed in Sec. 3.1.3. This objective function can be computed during the unwrapping process, however it requires excessive floating point operations to be considered for practical applications. Its use in this research is solely to observe the effect of reducing changes in branch cuts between frames. Note that the relationship between ρ and Strehl ratio is not fully explored in this research.

- $Var(h_{diff})$ (*Secondary Obj. Fun. and Metric*). The motivation for minimizing differences in h between frames (h_{diff}) is to also minimize changes in branch cuts. As shown in Fig. 48, large changes in h can lead to sudden drops in Strehl ratio. This objective function is available during the unwrapping process and little computation is needed to compare the current value of h with the previous one. Although no algorithm uses this as its primary objective function, some did treat it as a secondary objective function. Variances of h_{diff} are analyzed for each algorithm, so that it may also be used as a performance metric to more fully characterize their behavior.

When comparing algorithms it is helpful to distinguish those algorithms which are practical for closed-loop AO, and those which are meant as a proof-of-concept or focus on a fundamental attribute discussed in Sec. 3.1.3. The purpose of the first grouping is to compare unwrapping algorithms based on the various metrics which may be realistically used for real-time, closed-loop compensation. This group includes Xphase, LSPV+1, LSPV+4, Opt1-Opt6, and Opt9-Opt13. In contrast, the second group is being used to compare different attributes of strong-turbulence phase unwrappers, and not the algorithms themselves. The attributes include: low h_{diff} variance [LSPV+200(n)], low IWCL [LSPV+200(w)], and high correlation in ϕ_{non-LS} between frames (Opt7 and Opt8).

3.3 Chapter Summary

In summary, an exploration of the parameter space leads to information which can be used to optimize the PCO. The PDF for optimal values of h shows that in closed-loop AO, the distribution of IWCL minima is given by Eq. (72). Also, the value of h which minimizes the phase near the wrapping boundaries can be used as an estimate to begin a local search. In addition to minimizing IWCL, consideration must be given to minimizing the negative effects of the control law. Several algorithms are developed and their performances compared in closed-loop AO simulations. Chapter IV presents the results of that comparison.

IV. Results and Analysis

This chapter presents the results and analysis for the comparison of phase-unwrapping algorithms developed in Ch. III. The results are organized into three sections. The first considers how well each algorithm minimized IWCL, as this is the primary objective function in most cases. Next, the variances of h_{diff} are compared since this is a secondary objective function for some algorithms. This section also discusses the similarities between minimizing changes in h_{diff} and maximizing ρ . Finally, Strehl ratio results are presented and analyzed as functions of frame rate and turbulence strength.

4.1 IWCL

Before comparing the average IWCL for the practical algorithms, it is helpful to compare the differences between a PCO unwrapper with no optimization (LSPV+1) and one with nearly perfect optimization (LSPV+200). Figure 53 shows the average difference in IWCL between LSPV+1 and LSPV+200 broken out by frame rate and Rytov number. Since the IWCL is on average higher for LSPV+1, all the values shown in Fig. 53 are positive. The key point from this plot is that at small Rytov numbers and fast frame rates, all PCO algorithms essentially have the same IWCL, but as the conditions become more challenging, a more thorough search algorithm finds lower values of IWCL. Figure 53 gives cause for optimism for the success of PCO optimizations in challenging conditions. In more benign conditions, there are few branch points and very little lag in the compensation, and therefore little room for improvement. Figure 44 showed that in weak turbulence, there is a wide range of h values which can result in the same IWCL. As the frame rate decreased, Fig. 53 shows that LSPV+200 is able to find values of h with lower IWCL than LSPV+1.

This is consistent with the information shown in Fig. 45. As the frame rate is reduced, the compensation lagged too far behind the turbulence. This caused the variance of the PDF to increase as seen in Fig. 45 (a). At slower frame rates, there is less chance of finding the minimum at $h = 0$, and therefore LSPV+200 should have performed better than LSPV+1, which it did.

The data presented in Table 4 compares the practical algorithms across all Rytov numbers and frame rates. Although not practical, LSPV+200(w) is included for comparison purposes. The mean IWCL for each algorithm is normalized by the mean IWCL for the LSPV+1 unwrapper. To be more specific, the table shows

$$\text{IWCL Reduction [\%]} = 100 \times \left[\frac{\langle \text{IWCL}_{LSPV+1} \rangle - \langle \text{IWCL}_{OptX} \rangle}{\langle \text{IWCL}_{LSPV+1} \rangle} \right], \quad (78)$$

where $\langle \cdot \rangle$ indicates an ensemble average, and OptX indicates the various optimizations. Observing the percent reduction compared to LSPV+1 allows easy comparison of each algorithm against an PCO with no optimization. All numbers are positive, which indicates a reduction in IWCL (better performance).

Xphase clearly has the lowest IWCL when averaged over all conditions, reducing IWCL 10% more than LSPV+200. This is surprising since LSPV+200 always finds the value of h with the lowest IWCL down to a 0.005 wave resolution (1 wave/200). Testing the inputs and outputs of Xphase confirms that they are congruent. This means that the Xphase output is not missing information (branch cuts). It legitimately has lower IWCL on average than LSPV+200, a nearly perfectly optimized PCO algorithm. An explanation for this result may be that the CER Xphase has access to sets of branch cuts that cannot be realized by a PCO unwrapper. There are many branch cut paths possible on a 44×44 grid. Adjusting h during the PCO can only access a subset of the possible realizations. Since Xphase is a CER, it is not constrained in this way and may be finding branch cut arrangements that produce lower

IWCL than any of the possible PCO realizations. Although, for reasons presented in Sec. 4.3, Xphase is not the best possible unwrapping algorithm.

Comparing just the practical PCO algorithms, Opt13 (CDF-weighted GRS / parabolic interpolation algorithm), reduced the IWCL by 40% when compared to LSPV+1. When compared to LSPV+4 it reduced the IWCL by over 10%. Opt1 and Opt6 perform practically as well as Opt13 in reducing IWCL. This is surprising because Opt1 and Opt6 use a basic search technique, whereas Opt13 uses a more advanced algorithm, as well as *a priori* information derived from the CDF. LSPV+200 represents the approximate minimum IWCL achievable by a PCO, and Opt13, Opt1, and Opt6 close to this upper performance limit, accounting for the lack of differential in the results.

4.2 Frame-to-Frame variations in optimal values of h

As mentioned in Sec. 3.1.3, large changes in h from frame to frame can be undesirable due to the effects of the integral controller. It is important to note the difference between the variances of h and h_{diff} . Minimizing the variance in h means that the values chosen over a simulation never stray far from $h = 0$. This is not found to be an important attribute and therefore is not discussed in this section. Instead, minimizing the variance of h_{diff} is shown in Fig. 48 to avoid large and sudden drops in Strehl ratio. Several of the unwrapping algorithms developed in Ch. III are designed to minimize these changes. This section compares the average variances in h_{diff} for all unwrappers, including those which only minimize IWCL. Xphase has been omitted since it is a CER, so h_{diff} has no meaning. Table 5 presents the algorithms in order, starting with LSPV+1, which has zero change in h from frame to frame. Opt8 has the lowest variance aside from LSPV+1. This suggests that minimizing changes in ϕ_{non-LS} between frames, on average results in small h_{diff} variances. Although

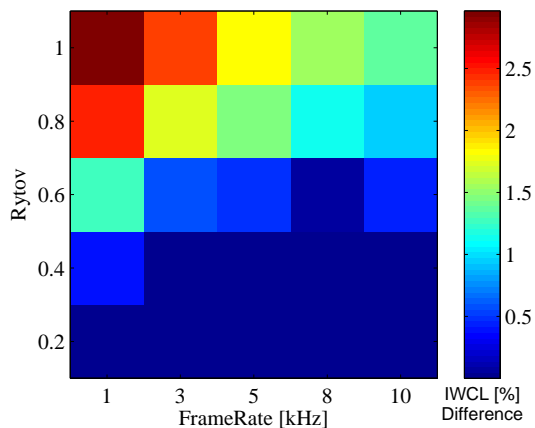


Figure 53. The difference in average IWCL between a PCO unwrapper without optimization (LSPV+1) and one with nearly perfect optimization (LSPV+200) as a function of frame rate and Rytov number.

Table 4. Reduction of mean IWCL for practical phase-unwrapping algorithms when compared to LSPV+1. The LSPV+200(w) unwrapper is included for comparison purposes.

Algorithm	IWCL Reduction [%]
Xphase	52.2
LSPV+200(w)	42.8
Opt13	40.0
Opt1	39.9
Opt6	39.3
Opt12	35.1
Opt3	32.8
Opt10	32.3
Opt2	29.9
LSPV+4	29.3
Opt9	23.9
Opt11	20.5
Opt5	1.5
Opt4	0.1
LSPV+1	0

not directly proven, the reverse may also be true, minimizing changes in h minimize changes in ϕ_{non-LS} . This supports the theory that large changes in h can drastically change the branch cuts in the ϕ_{non-LS} , and build up discontinuities on the DM via the integral controller. After Opt8 is LSPV+200(n) with a range limited to 0.05 waves. This is the largest jump possible between frames resulting in low variances. Opt4 and Opt5 also have relatively low variances. They use median values of the phase and intensity-weighted phase, respectively, and do not search for lower IWCL. Opt9 and Opt11 use the CDF to distribute *seeds* in bins of equal probability. Most of the area under the PDF curve lies close to zero, so the majority of the *seeds* are placed here. This means that large changes in h are rare with these algorithms.

4.3 Strehl Ratio

As previously mentioned, the sole purpose of developing optimizations is to maximize system performance, as measured by Strehl ratio. This section presents the results of the practical algorithm comparison based on Strehl ratio. Table 6 shows the percent reduction in σ_{norm}^2 compared to LSPV+1 for the practical algorithms, sorted in order of performance, high to low. A positive number indicates a lower σ_{norm}^2 , and therefore better performance. Negative numbers indicate a degradation in performance when compared to LSPV+1. The AO system which uses Opt2 has on average, the best performance. It reduces σ_{norm}^2 by 17.3% when compared to the non-optimized PCO algorithm LSPV+1. It also outperforms LSPV+4, a slightly more robust PCO algorithm by 10.4%. Opt6 also performs well compared to LSPV+1 and LSPV+4. In general, the PCO algorithms which reduce IWCL by at least 20% when compared to LSPV+1, also reduce the Strehl ratio σ_{norm}^2 . The exceptions are Opt10 and Xphase which significantly increase σ_{norm}^2 .

The poor performance of Xphase highlights the effects of the integral-control law.

Table 5. Average variances of h_{diff} for all unwrapping algorithms (except Xphase). Values are based on h given in waves and have not been normalized.

Algorithm	$\text{Var}(h_{diff}) \times 10^{-3}$
LSPV+1	0
Opt8	0.7
LSPV+200(n)	0.8
Opt4	1.1
Opt7	1.6
Opt5	2.0
Opt9	2.3
Opt11	2.7
LSPV+200(w)	7.4
Opt3	7.8
Opt10	10.0
Opt12	10.4
Opt6	13.0
LSPV+4	13.6
Opt13	20.8
Opt1	27.0
Opt2	32.4

Although Xphase has the lowest IWCL of the tested algorithms, it results in an unsteady performance. Figure 54 shows the results of a typical closed-loop simulation for Xphase and Opt2 at a Rytov number of 0.6 and frame rate of 5 kHz. The Strehl ratio for Xphase has numerous sudden drops, similar to what is observable in Fig. 48. If Xphase can produce sets of branch cuts not available to PCO algorithms, it has a larger variation of cuts possible. Larger variations in branch cuts mean that the integral control law has more of an effect on Strehl ratio. This is a possible explanation for the high variance in Strehl ratio for Xphase. When using a PCO, minimizing the variance of (h_{diff}) can help mitigate the effects of the control law. Unfortunately, Xphase has no such adjustable parameter, so nothing can be done avoid these effects.

The results shown in Table 6 generalize the performances over a wide range of Rytov numbers and frame rates. To gain a better understanding of each algorithm's behavior, it is useful to divide the parameter space up and analyze one section at a time.

4.3.1 Frame Rate.

The first division is by frame rate. At slow frame rates of 1 and 3 kHz (for all five Rytov numbers), AO compensation is difficult because of the lag in system bandwidth compared to the changing turbulence. As discussed in Sec. 4.1, the optimal values of h are more uniformly distributed at slower frame rates, making optimizing the PCO more difficult. In these conditions, Opt2 has the highest mean and lowest variance in Strehl ratio resulting in the lowest σ_{norm}^2 of any unwrapper, including the proof-of-concept algorithms. LSPV+4 is 11.8% higher in σ_{norm}^2 when compared to Opt2, and LSPV+1 is 25.4% higher. At fast frame rates of 8 and 10 kHz, Opt2 has the highest mean and lowest variance of the practical unwrappers. Although the mean Strehl ratios are similar at fast frame rates, the variances are not. The small variance

Table 6. Normalized Strehl Ratio Variance σ_{norm}^2 for practical algorithms. This table shows how well each algorithm reduced σ_{norm}^2 when compared to LSPV+1.

Algorithm	% Reduction in σ_{norm}^2 compared to LSPV+1
Opt2	17.3
Opt6	14.2
Opt1	12.2
Opt13	10.1
Opt12	7.6
Opt9	7.2
LSPV+4	6.9
Opt3	4.9
LSPV+1	0
Opt5	-3.7
Opt11	-4.2
Opt4	-9.6
Opt10	-24.8
Xphase	-86.3

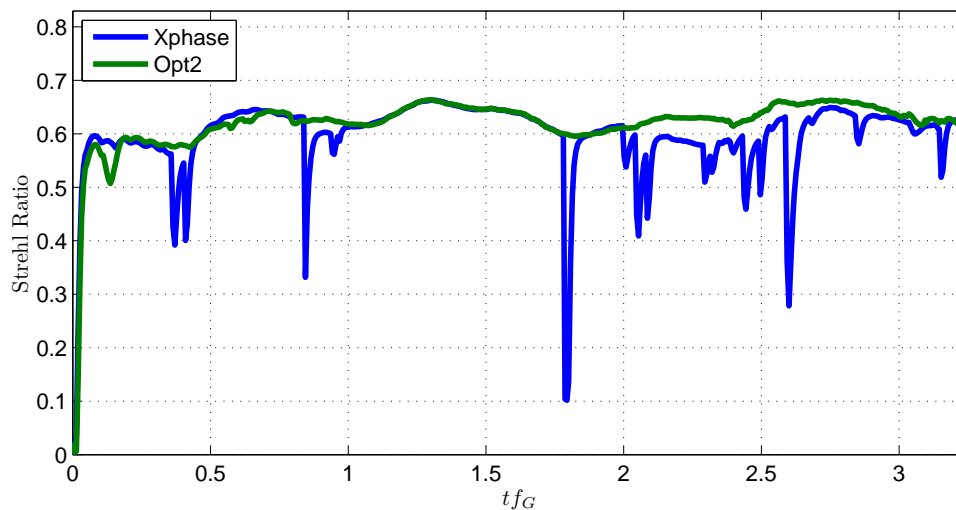


Figure 54. Comparison of Strehl ratio for Xphase and Opt2 at a Rytov number of 0.6 and frame rate of 5 kHz. Using Xphase in closed-loop AO results in unsteady performance.

of Opt2 distinguishes it from the other unwrappers. With respect to σ_{norm}^2 , LSPV+1 performs 9.0% worse while LSPV+4 is 19.8% worse. Clearly, Opt2 is an effective optimization of the PCO in both slow and fast frame rates. This makes it a wise selection for both slow and fast AO systems.

Analyzing the behavior of the proof-of-concept algorithms provides insight into how the different attributes affect system performance as a function of frame rate. As the frame rate increases, wide searches of h to minimize IWCL became less important. Figure 55 shows how the percent difference in σ_{norm}^2 (compared to Opt2) between LSPV+1 and LSPV+200(w) becomes less pronounced as the frame rate increases. At slower frame rates, LSPV+1 and LSPV+200(w) result in very different IWCL values. This leads to a significant difference in normalized Strehl ratio variance. At fast frame rates, the two algorithms converge towards a zero percent difference from Opt2. Fast frame rates are ideal, so it makes sense that all of the algorithms perform similarly well at 10 kHz. The important lesson learned from these results is that slower AO systems must minimize IWCL more than a fast system would need to do when executing the PCO.

The next interesting result as a function of frame rate, is the effectiveness of minimizing changes in ϕ_{non-LS} . Figure 56 shows the percent difference in σ_{norm}^2 (compared to Opt2) for Opt7 and Opt8 at a Rytov number of 0.4. At slow frame rates, their normalized variances are significantly higher than Opt2. As the frame rate increases, the difference reduces. This is due to an increase in correlation of ϕ_{non-LS} from frame to frame as the time between frames decreases (increase in frame rate). Less time between frames means that less has changed, assuming the Greenwood frequency has remained constant. When the frame rate is slow, Opt7 and Opt8 produce a ϕ_{non-LS} based on the previous frame, and contain a set of branch cuts which may not be the best fit for the current intensity profile. PCO algorithms

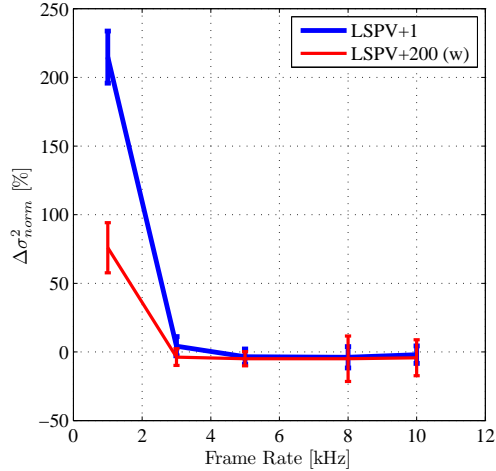


Figure 55. The percent increase in σ_{norm}^2 over Opt2 as a function of frame rate for LSPV+1 and LSPV+200(w) at a Rytov number of 0.4. The difference between the two algorithms is greatest at slow frame rates.

should not attempt to minimize changes in branch cuts for slow frame rates.

4.3.2 Turbulence Strength.

The next division that provides insight into the results of Table 6 is that of turbulence strength. At low Rytov numbers of 0.2 and 0.4 (all frame rates), there is little difference in the mean Strehl ratio of the different algorithms. Conversely, there is a large difference in the variances. Of all algorithms, Opt6 and Opt2 (approximately equal) have the smallest σ_{norm}^2 in weak turbulence, due to their low variances. LSPV+1 has a normalized variance 93% higher than Opt6 and Opt2, although its mean is virtually the same. At large Rytov numbers of 0.8 and 1.0, Opt2 has the lowest normalized variance and the highest mean. LSPV+4 performs 13.2% worse and LSPV+1 is 16.8% worse. Opt2 performs well at both weak and strong turbulence strengths, and is a wise choice regardless of the Rytov number.

Although its Strehl ratio variance is 24% higher than Opt2 in weak turbulence, Opt7 has the lowest variance at the highest turbulence strengths. In these stronger cases, there is a higher density of branch cuts in ϕ_{non-LS} , and therefore a greater

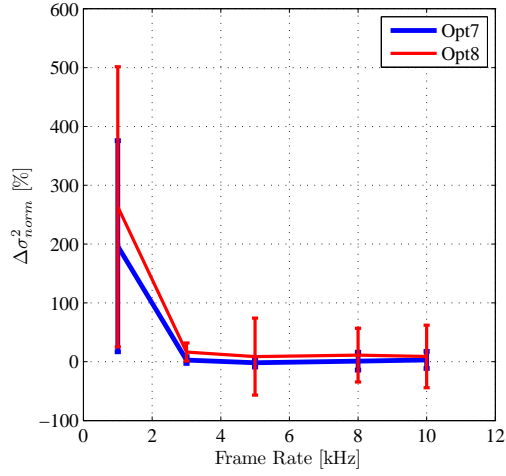


Figure 56. The percent increase in σ_{norm}^2 over Opt2 as a function of frame rate for Opt7 and Opt8 at a Rytov number of 0.4. Minimizing changes in ϕ_{non-LS} is less effective at slow frame rates.

probability of build up on the DM. This suggests that in strong turbulence, reducing changes in ϕ_{non-LS} in addition to minimizing IWCL, can help reduce the variance of the Strehl ratio. Opt7 applies equal weighting to the metrics which does reduce the variance, but also slightly reduces the mean. For this reason, it has a normalized variance 9.7% higher than Opt2. Applying more weight towards minimizing IWCL and less towards maximizing ρ may lead to a higher mean, and therefore a higher normalized variance. Based on the results from this section, minimizing both IWCL and changes in ϕ_{non-LS} becomes more effective in strong turbulence, but not with equal weighting. A careful balance must be obtained to optimize closed-loop AO performance.

4.3.3 Benign and Challenging Conditions.

The final analysis for comparing the PCO optimizations is to consider their behavior in benign operating conditions (weak turbulence and fast frame rate) and also challenging conditions (strong turbulence and slow frame rate). In the benign conditions, all unwrappers including the proof-of-concept algorithms perform essen-

tially equal in Strehl ratio mean, variance, and normalized variance. The exception is LSPV+200(n), which generally performs poorly in all conditions. Other than that algorithm, there is little differentiation between optimizations. This indicates that AO systems which operate in weak turbulence and with high bandwidth, can perform adequately with any PCO algorithm. In challenging conditions, there is a larger spread in performance. Just as it does when only considering strong turbulence, Opt7 has the lowest Strehl ratio variance, but also with a low mean. Opt2 has the lowest normalized variance and highest Strehl ratio mean of all unwrappers. LSPV+4 has a 12.2% higher σ_{norm}^2 , and LSPV+1 has a 19.5% higher σ_{norm}^2 as well as a 6% lower Strehl ratio mean. The change in relative performances from the benign case clearly demonstrates that AO systems which operate in strong turbulence and with low bandwidth should choose a phase unwrapper which best optimizes the PCO. The objective function of choice depends on the application. Minimizing changes in ϕ_{non-LS} provides the smallest Strehl ratio variance, while minimizing IWCL can provide a higher mean. Opt2 strikes the best balance of all the algorithms considered.

4.3.4 Strehl Ratio CDF.

As previously mentioned, another way of comparing results is to observe the Strehl ratio CDF's. Figure 57 shows the CDF's for the top two PCO optimizations Opt2 and Opt6, compared to the CDF for Xphase. The axis limits are chosen to highlight differences in low Strehl ratios. Clearly, the PCO optimizations reduce the occurrences of low values. As an example, the probability of $S \leq 0.4$ is 18.9% for Xphase, and approximately 14.4% for the PCO algorithms. Four and a half percent may not appear significant at first, but it represents nearly a one-quarter drop in the chance that a system performs below the $S = 0.4$ threshold (arbitrarily chosen concrete example).

Figure 58 shows the CDF's for the top two PCO optimizations Opt2 and Opt6,

as well as the most basic PCO algorithms LSPV+1 and LSPV+4. It shows that Strehl ratio CDF's do not clearly highlight differences in the better performing PCO algorithms. For this reason, CDF's of the other optimizations are not shown. Little insight is gained from their comparisons. There is a clear difference between the most basic PCO LSPV+1 and algorithms which attempt to minimize IWCL. For example, if an AO application requires Strehl ratios of $S \geq 0.35$ at a minimum of 90% of the time, LSPV+1 is unacceptable, and an optimization is needed.

4.4 Chapter Summary

The results presented in this chapter highlight the differences between the various optimization designs. The sophisticated search technique Opt13 is shown on average to minimize IWCL 40% better than LSPV+1. This performance is comparable to that of LSPV+200. However, Opt13 has a σ_{norm}^2 , only 10% lower than LSPV+1. Opt2 has the lowest σ_{norm}^2 when averaged over all conditions. It performs especially well in the most challenging conditions. In addition, it is shown that optimization of the PCO is most effective at slow frame rates and/or strong turbulence. This is observable in both the IWCL and Strehl-ratio results. Systems operating in these conditions should use Opt2.

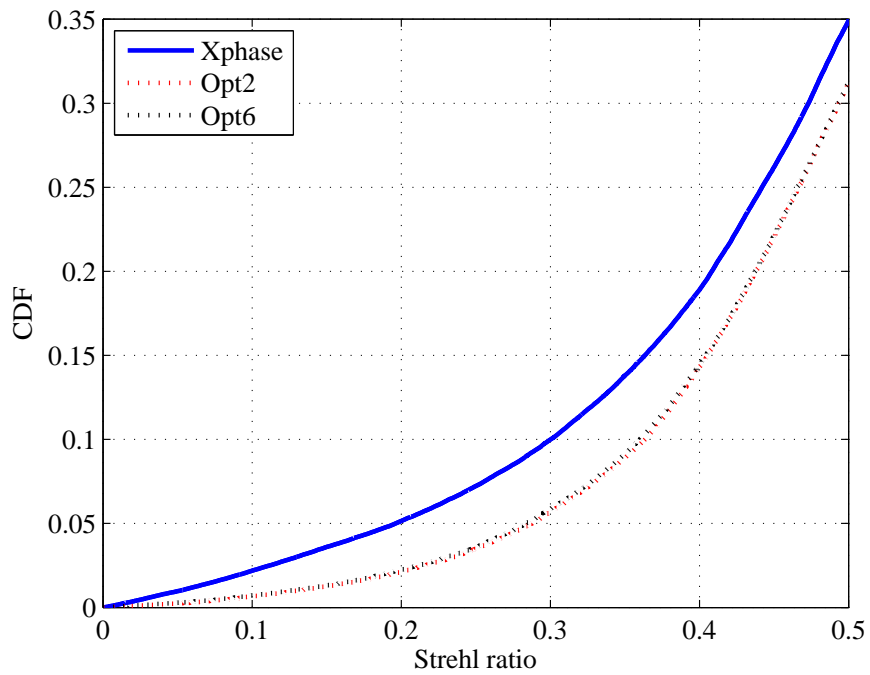


Figure 57. CDF's for Xphase, Opt2, and Opt6 for Strehl ratios below $S = 0.5$.

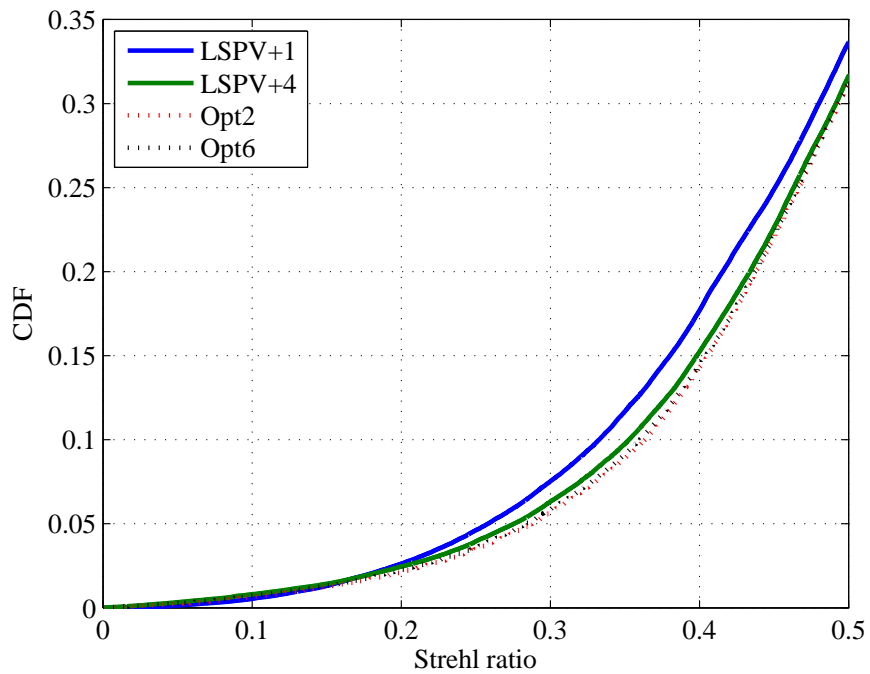


Figure 58. CDF's for Opt2 and Opt6 compared to the basic PCO algorithms of LSPV+1 and LSPV+4 for Strehl ratios below $S = 0.5$.

V. Conclusions and Recommendations

This chapter summarizes the activities of this research and the results obtained. Challenges overcome and significant contributions are highlighted. These key results are the basis for recommendations regarding future efforts to optimize PCO algorithms. The recommendations are intended to help further the research of phase-unwrapping in strong turbulence.

5.1 Summary

As discussed in Ch. I, the objectives of this research have been to validate the correlation between Strehl ratio and IWCL, determine the statistical relationship between the PCO parameters chosen and IWCL, determine the effects of the integral-control law, develop an unwrapping algorithm, and compare it against conventional methods. To meet these objectives, a simulation environment is developed in Ch. III to explore the parameter space. Atmospheric propagation of an ideal beacon through weak and strong turbulence is modeled to induce branch points in the observed field. A complete AO system is also simulated for both open- and closed-loop compensation. Operations are conducted over a wide range of turbulence conditions and system frame rates to fully explore PCO behavior. Probability density and cumulative distribution functions are developed to help locate the value of h in the PCO corresponding to minimum values of IWCL. Based on this information, 11 new and practical PCO algorithms are developed and tested. In addition, four proof-of-concept algorithms are tested, providing insight into how specific attributes of the PCO affect AO performance over varying conditions. In Ch. IV, results of the simulations and comparisons are presented and analyzed based on key metrics developed throughout the research. Algorithm performance based on Strehl ratio is analyzed as a function

of AO system frame rate, turbulence strength, and benign and challenging combinations of the two. Finally, Opt2 is shown to be the most robust phase-unwrapping algorithm for practical use in strong turbulence. Across all conditions, it outperforms LSPV+1 and LSPV+4 decreasing normalized Strehl ratio variance by 17.3% when compared to LSPV+1 and 10.4% when compared to LSPV+4. Under the most challenging conditions, Opt2 outperforms LSPV+1 and LSPV+4 by 19.5% and 12.2%, respectively.

5.2 Challenges Overcome

The challenges overcome during the course of this research are:

- *Large amount of phase-unwrapping research published.* Sorting through the research published on phase-unwrapping is a daunting task. As discussed in Sec. 2.3, phase unwrapping is found in many applications and hundreds of papers are written on the subject. In addition, the wide range of applications has led to numerous sets of terminologies, making it difficult to compare similar concepts. A clear organization of unwrapping methodologies helps sort the proposed algorithms into like groups. Comparing similar algorithms allows the dismissal of redundant techniques. Also, applying the requirements of real-time closed-loop AO allows entire groups to be dismissed. This significantly reduces the number of algorithms to be modeled during simulations.
- *Realistic Models.* Using modeling and simulation requires careful analysis to determine the validity of results. During this research, there are cases in which the models exhibit non-physical behavior. These include edge effects from Wave-Prop's interpolation of the phase and the behavior of MATLAB[®] when choosing minimum and maximum values in an array of similar values. In each case, the effects make the results difficult to interpret. Careful examination of the code

is required to diagnose the problems. In some cases, the simulations have to be stripped down to a minimum and built back up to isolate the issues.

- *Development of AO system using an SRI.* Since there are no robust models available of AO systems using an SRI, a significant amount of time is spent developing one. An observed degradation in Strehl ratio over time required a redesign of the AO system model. An analysis found that this behavior is most likely a physical phenomenon caused by a build up of unsensed branch cuts on the DM and not an error in the code. To remedy the problem, the SRI resolution is doubled to sense these discontinuities. The use of a SRI and DM with two different resolutions requires careful alignment of system components. In addition, a method for filtering and down-sampling the SRI phase to create DM commands has to be implemented.
- *Parameter space complexity.* The most difficult challenge faced is the complexity of the parameter space. There are many variables and parameters which interact to affect the results. These include Rytov number, frame rate, Greenwood frequency, grid spacing, DM actuator coupling and stroke limit, whether the phase is unwrapped in SRI or DM resolution, threshold being used for discontinuity measurements (IWCL), open- versus closed-loop, and control-law coefficients. Choosing these variables to isolate the effects of unwrapping proves difficult. Variables have to be studied one at a time to determine their affect on an AO systems using the PCO and to isolate interesting behavior. This trial-and-error process is very time consuming. In addition, determining the interactions of IWCL, $\text{Var}(h)$, $\text{Var}(h_{diff})$, and ρ is difficult. Only a few simple relationships are determined.
- *Excessive amounts of data.* Finally, studying such a complex parameter space

leads to large amounts of data. An effective methodology for the analysis of this data has to be developed to tie it all together and provide simple yet meaningful conclusions. In the end, the parameter space is analyzed by Rytov number and frame rate. The use of normalized variance allows for easy comparisons of an algorithm's affect on Strehl ratio mean and variance through a single metric.

5.3 Key Results

The key results obtained during this research are:

- *Well defined IWCL parameter space.* The parameter h chosen during the PCO can take on any value. However, the IWCL that results is periodic over a one-wave range of h . The most ideal range is $-0.5 \leq h < 0.5$, which results in small variations of the parameter centered at $h = 0$. As h varies over this range, the resulting IWCL is somewhat concave with a well defined minimum value.
- *Distribution of optimal h values.* The location of the minimum IWCL with respect to h , is uniformly distributed over the one-wave period when using a PCO phase-unwrapping algorithm in open-loop. When unwrapping the phase of the compensated field in closed-loop, the distribution of IWCL minima takes on an approximately Cauchy-Lorentzian distribution centered at zero and given by Eq. (72). The resulting CDF is easily used to minimize IWCL.
- *Relationship between IWCL and frame rate.* The frame rate of an AO system can affect the variance in the optimal values of h , as shown in Fig. 45. At very slow frame rates, the compensation lags behind the turbulence and the phase being unwrapped resembles uncompensated cases found in open-loop. This causes the Cauchy-Lorentzian distribution to spread out, becoming more uniform. PCO algorithms operating at slow frame rates must be able to search

a wider portion of the one-wave range to find the lowest IWCL.

- *Augmentation of formal search techniques with statistical data to minimize IWCL.* A traditional, deterministic GRS is augmented with statistical data for application in the stochastic realm for the first time. This modification, while devised for this application, may have broader applicability. In this work, when used to minimize IWCL as part of Opt13, it is shown to enhance an algorithm's ability to obtain low values. Opt13 has the lowest average IWCL of any PCO algorithm. Its average IWCL is five percent lower than the deterministic GRS Opt12, which highlights the improvement gained by weighting the search by the CDF. In addition, Opt13 obtains IWCL values 10.7% lower than LSPV+4 and 40.0% lower than LSPV+1. When considering AO performance, Opt13 reduces the normalized variance of the Strehl ratio by 10.1% when compared to LSPV+1.
- *Effect of integral-control law on branch-point-tolerant unwrappers* Minimizing IWCL alone is not sufficient for optimum AO performance in closed-loop. It is shown that large changes in h from frame to frame are undesirable, as they can lead to sudden drops and high variances in Strehl ratio. This results from significant changes in the branch cuts between frames, which can cause a build up of 2π discontinuities on the DM because of the integral controller.
- *Effectiveness of temporal correlation.* Due to the significant lag in compensation at slow frame rates, PCO algorithms should not attempt to minimize changes in h or ϕ_{non-LS} in these conditions. Algorithms that focus on these attributes perform poorly at slow frame rates.
- *PCO optimization most important in challenging conditions.* In benign operating conditions, there is little room for improvement so most unwrappers produce

similar IWCL and Strehl ratio values. AO systems operating in these conditions should choose the simplest PCO algorithm to implement, LSPV+1. Once the turbulence increases in strength or the bandwidth drops (or both), optimizing the PCO becomes necessary. The results in Ch. IV show that as the turbulence strength increases, algorithms with lower IWCL perform better. Decreases in frame rate bring about similar results. Minimizing IWCL is more important at slower frame rates than it is at faster frame rates.

- *Novel technique for determining starting point in IWCL minimization.* Minimizing the number of samples in ϕ_{non-LS} near the wrapping boundaries also minimizes the total length of branch cuts. By constructing a histogram of the ϕ_{non-LS} values, an estimate of an optimal h can be calculated by determining the value which minimizes phase at the wrapping boundaries. The estimate proved an effective starting point for the IWCL minimum search. This technique of applying the PCO results in higher, steadier Strehl ratio values with lower chances of sudden drops in performance when compared to other PCO algorithms.

5.4 Recommendations

The following are recommendations for the improvement of processes and extension of technical objectives intended to further the research of phase-unwrapping in strong turbulence:

- *Opt2 Improvement.* Although Opt2 is shown to be practical for closed-loop AO, it is truly a proof-of-concept algorithm at this point and needs further development before it is capable of real-time implementation. To begin with, it doubles the number of ϕ_{non-LS} evaluations necessary when compared to LSPV+4. Since

this number is somewhat arbitrarily chosen, it may be possible to reduce it without a significant degradation in performance. Computing h_{hist} also increases the computational burden, but far less than the ϕ_{non-LS} evaluations. Would using the h_{hist} estimate be sufficient for a LSPV+2 algorithm (one ϕ_{non-LS} evaluation for the histogram and another based on h_{hist})? Is the second process which checks the IWCL at $h = 0$ necessary? These are questions which, once answered, may help further reduce the computational burden of the algorithm. A faster version of Opt2 would be more suitable for implementation with real hardware.

- *Analysis of the PCO in the Presence of Noise.* Prior to implementing a PCO unwrapping algorithm with real hardware, it is important to understand how noise affects the operation. As discussed in Sec. 2.3.3.5, research suggests that when noise is present, the PCO requires h be applied both inside and outside the wrapping operator. This research focused on cases without noise, and thus the results may differ when it is included.
- *Laboratory Implementation.* Once the effects of noise are understood, a comparison using real hardware in the laboratory can be conducted. Although a significant amount of time was put into the development of the wave-optics simulations for this research, it can not fully account for all attributes of a physical system. In addition, certain behavior observable in simulations such as edge effects from modeling the DM, are the result of the simulations themselves and are not a physical phenomenon. A comparison using hardware can better determine the affect of PCO optimization on system performance.
- *Development of Alternative Control Schemes.* One of the most complicating factors when optimizing the PCO is the integral-control law's effect on shift-

ing branch cuts. An alternative control scheme could be implemented which separates commands being applied to the DM into LS and non-LS components. The commands developed from ϕ_{LS} would be applied in closed-loop, while those developed from ϕ_{non-LS} could be applied in open-loop or with different control coefficients. This would possibly provide the effectiveness of a closed-loop system, while preventing a build up of branch cuts on the DM. If such a decomposition is not possible to command one DM, two DM's could be used [7]. Perhaps closed-loop commands from ϕ_{LS} would control one mirror while the open-loop commands from ϕ_{non-LS} would control the other.

5.5 Chapter Summary

Accomplishing the objectives presented in Ch. I support the hypothesis that a LS unwrapper with a PCO can improve system performance in strong turbulence if the PCO is optimized in real time to minimize IWCL as well as mitigate the effects of the integral-control law. This improvement becomes more significant at slower frame rates and stronger turbulence strengths. Finally, a hybrid algorithm (Opt2) is developed which results in higher and more steady Strehl ratios when compared to other unwrapping algorithms.

Appendix A. Appendix A - Matlab® Code

Listing A.1. LSPV_Plus_Four.m

```
1 % LSPV+4 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10 % Outputs:
11     % UnwrappedPhase - unwrapped phase [waves]
12     % h - phase shift chosen for PCO with lowest IWCL [waves]
13 % Required Functions:
14     % unwrap_ls.m
15     % wrap-wave.m
16     % Fast_IWCL.m
17
18 function [UnwrappedPhase h]= LSPV_Plus_Four(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask);
19
20 NN=4; % number of non-LS evaluations
21
22 [n m]=size(w_phase);
23 LS_phase=unwrap_ls(mask,w_phase)/(2*pi); % unwrapped LS phase [waves]
24
25 % initialize arrays
26 iwcl=zeros(1,NN);
27 NonLS_phase=zeros(n,m,NN);
28
29 % Calculate NN rotational phases
30 hh=linspace(-0.5,0.25,NN);
31 for index=1:NN
32     phaseshift=hh(index);
33     NonLS_phase(:,:,index) = wrap-wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
34     [iwcl(index)]=Fast_IWCL(intensity , NonLS_phase(:,:,index) , masksize , xSRI , ySRI , deltaSRI);
35 end
36
37 % Pick the best one
38 [cc, IIndex] = min(iwcl);
39 UnwrappedPhase=LS_phase+NonLS_phase(:,:, IIndex);
40
41 % Get phase shift from lowest IWCL
42 hh=linspace(-0.5,0.25,NN);
43 h=hh(IIndex);
```

Listing A.2. Fast_IWCL.m

```

1  % Function for calculating IWCL
2
3  % Inputs:
4      % Intensity - intensity across the phase
5      % phase - phase [waves]
6      % masksize - physical size of mask [m]
7      % xn, yn - coordinates in phase space
8      % deltan - grid spacing in phase space
9  % Outputs:
10     % IntensityWeightedCutLength - IWCL [fraction]
11
12  function [IntensityWeightedCutLength]=Fast_IWCL(intensity, phase, masksize, xn, yn, deltan);
13
14  mask2 = circ(xn, yn, masksize-4*deltan); % create a circular ap for IWCL that doesnt include the edges
15  [rows, cols]=size(phase);
16
17  ydiff=(diff(phase,1,1)); % get differences for both x and y directions
18  xdiff=(diff(phase,1,2));
19
20  s=0.6; % set discontinuity sensitivity
21
22  sumxdiff=xdiff<=-s | xdiff>=s; % determine where the differences are
23  sumydiff=ydiff<=-s | ydiff>=s; % equal to plus or minus 2pi and put a 1
24
25
26  % Develop a matrix the size of the original input phase with 1's on both
27  % sides of any point where the difference is plus/minus 2pi. This marks
28  % the areas that the cuts go through for intensity weighting purposes
29  Cutmap=zeros(rows, cols);
30  Cutmap(:,1:cols-1)=sumxdiff;
31  Cutmap(:,2:cols)=Cutmap(:,2:cols)+sumxdiff;
32  Cutmap(1:rows-1,:)=Cutmap(1:rows-1,:)+sumydiff;
33  Cutmap(2:rows,:)=(Cutmap(2:rows,:)+sumydiff)>=1;
34
35  CutIntensityMap=Cutmap.*intensity.*mask2; %Map the intensity to the matrix of 1's
36  % and 0's to find the intensity around the cuts
37  % [a b IntensityinMask]=find(intensity.*mask2);
38  TotalIntensity=sum(sum(intensity.*mask2)); % Determine total intensity
39  % TotalIntensity=mean(IntensityinMask(:));
40
41  IntensityAroundCuts=sum(sum(CutIntensityMap)); % Determine the total intensity of
42  % pixels with a cut running next to them
43
44  IntensityWeightedCutLength=IntensityAroundCuts/TotalIntensity; % Calculate
45  % the IWCL

```

Listing A.3. LSPV_Plus_N.m

```

1 % LSPV+N (LSPV+200) unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10 % Outputs:
11     % UnwrappedPhase - unwrapped phase [waves]
12     % h - phase shift chosen for PCO with lowest IWCL [waves]
13     % iwcl - the N IWCL values (for plotting parameter space)
14     % pp - the rotational phase of for lowest IWCL (FYI)
15 % Required Functions:
16     % unwrap_ls.m
17     % wrap_wave.m
18     % Fast_IWCL.m
19
20 function [UnwrappedPhase h iwcl pp]=...
21     LSPV_Plus_N(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask , NN);
22
23 [n m]=size(w_phase);
24 LS_phase=unwrap_ls(mask , w_phase)/(2*pi); % unwrapped LS phase [waves]
25
26 % initialize arrays
27 iwcl=zeros(1,NN);
28 NonLS_phase=zeros(n,m,NN);
29
30 % Calculate NN rotational phases
31 hh=linspace(-.5,0.5,NN);
32 for index=1:NN
33     phaseshift=hh(index);
34     NonLS_phase(:, :, index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
35     [iwcl(index)]=Fast_IWCL(intensity , NonLS_phase(:, :, index) , masksize , xSRI , ySRI , deltaSRI);
36 end
37
38 % Pick the best one
39 [cc , IIndex] = min(iwcl);
40 hh=linspace(-.5,0.5,NN);
41 h=hh(IIndex); % Get corresponding value of h
42 UnwrappedPhase=LS_phase+NonLS_phase(:, :, IIndex)+h; % Calculate total phase
43 pp=NonLS_phase(:, :, IIndex);

```

Listing A.4. LSPV_Plus_N_Optimized.m

```

1 % Opt1 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10 % Outputs:
11     % UnwrappedPhase - unwrapped phase [waves]
12     % h - phase shift chosen for PCO with lowest IWCL [waves]
13 % Required Functions:
14     % unwrap_ls.m
15     % wrap_wave.m
16     % Fast_IWCL.m
17
18 function [UnwrappedPhase h]= ...
19 LSPV_Plus_N_Optimized(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask);
20
21 [n m]=size(w_phase); % get size of grid
22 LS_phase=unwrap_ls(mask , w_phase)/(2*pi); % unwrapped waves
23
24 % initialize arrays
25 iwcl1=zeros(1,5);
26 iwcl2=zeros(1,5);
27 NonLS_phase1=zeros(n,m,5);
28 NonLS_phase2=zeros(n,m,6);
29
30 % set location of seeds
31 h1=0;
32 delta1=0.1;
33 delta2=0.3;
34 delta3=0.05;
35 delta4=0.15;
36
37 hh=[(h1-delta2) (h1-delta1) h1 (h1+delta1) (h1+delta2)];
38
39 for index=1:5
40     phaseshift=hh(index);
41     NonLS_phase1(:, :, index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
42     [iwcl1(index)]=Fast_IWCL(intensity , NonLS_phase1(:, :, index) , masksize , xSRI , ySRI , deltaSRI);
43 end
44
45 % Pick the best one
46 [cc , IIndex] = min(iwcl1);
47
48 % Use the best one for the starting pt of second iteration
49 h2=hh(IIndex);
50 hhh=[(h2-delta4) (h2-delta3) h2 (h2+delta3) (h2+delta4)];
51

```

```

52 for index=1:5
53     phaseshift=hhh(index);
54     NonLS_phase2(:, :, index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
55     [iwcl2(index)]=Fast_IWCL(intensity , NonLS_phase2(:, :, index) , masksize , xSRI , ySRI , deltaSRI);
56 end
57
58 % Pick the best one
59 [cc , IIndex] = min(iwcl2);
60 UnwrappedPhase=LS_phase+NonLS_phase2(:, :, IIndex);
61
62 h=hhh(IIndex); % Get phase shift from lowest IWCL

```

Listing A.5. LSPV_Plus_Opt.m

```

1 % Opt2 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10 % Outputs:
11     % UnwrappedPhase - unwrapped phase [waves]
12     % h - phase shift chosen for PCO with lowest IWCL [waves]
13 % Required Functions:
14     % unwrap_ls.m
15     % wrap_wave.m
16     % Fast_IWCL.m
17
18 function [UnwrappedPhase h]= LSPV_Plus_Opt(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask);
19
20 B=30; % number of bins in histogram
21 [n m]=size(w_phase); % get size of array
22 iwcl1=zeros(1,5); % initialize array for first process
23 iwcl2=zeros(1,3); % initialize array for second process
24 NonLS_phase1=zeros(n,m,5); % initialize storage of non-LS phase
25 NonLS_phase2=zeros(n,m,3); % initialize storage of non-LS phase
26
27 LS_phase=unwrap_ls(mask , w_phase)/(2*pi); % unwrapped LS phase [waves]
28 NonLS_phase1(:, :, 1) = wrap_wave(w_phase/(2*pi)-LS_phase); % wrapped non-LS phase [waves]
29
30 % Create Histogram
31 [rows , columns , PhaseValues]=find (mask.*NonLS_phase1(:, :, 1));
32 [Nums Bins]=hist (PhaseValues , B);
33 Norm=max(Nums); %for normalized

```

```

34 BinSpacing=abs(Bins(3)-Bins(2));
35
36 [a b]=find(Nums==min(Nums)); % Find the minimum bin
37 h1=b*BinSpacing; % Estimate h
38 % Wrap the value of h to the range being used
39 if h1>0.5
40     h2=h1-1;
41 elseif h1<-0.5
42     h2=h1+1;
43 else
44     h2=h1;
45 end
46
47 % Plant seeds around h estimate
48 hh=[h2-0.10 h2-0.06 h2 h2+0.06 h2+0.10];
49
50 % Calculate non-LS phase and IWCL
51 for index=1:5
52     phaseshift=hh(index);
53     NonLS_phase1(:, :, index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
54     [iwcl1(index)]=Fast_IWCL(intensity, NonLS_phase1(:, :, index), masksize, xSRI, ySRI, deltaSRI);
55 end
56
57 [cc1, IIndex1] = min(iwcl1); % Get minimum IWCL and corresponding h
58 % Wrap value of h to range being used
59 if hh(IIndex1)>0.5
60     h3=-0.5+(hh(IIndex1)-0.5);
61 elseif hh(IIndex1)<-0.5
62     h3=0.5+(hh(IIndex1)+0.5);
63 else
64     h3=hh(IIndex1);
65 end
66
67 %%%%% Second process
68 delta=.05; % set seed spacing around zero
69 hhh=[(-delta) 0 (delta)];
70
71 % Calculate non-LS phase and IWCL
72 for index=1:3
73     phaseshift=hhh(index);
74     NonLS_phase2(:, :, index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
75     [iwcl2(index)]=Fast_IWCL(intensity, NonLS_phase2(:, :, index), masksize, xSRI, ySRI, deltaSRI);
76 end
77 [cc2, IIndex2] = min(iwcl2); % Get minimum IWCL and corresponding h
78 h4=hhh(IIndex2);
79
80 % Choose the lowest IWCL and h from the two processes
81 if cc1<cc2
82     UnwrappedPhase=LS_phase+NonLS_phase1(:, :, IIndex1);
83     h=h3;
84 else
85     UnwrappedPhase=LS_phase+NonLS_phase2(:, :, IIndex2);
86     h=h4;
87 end

```


Listing A.6. LSPV_Plus_Opt3.m

```
1 % Opt3 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10    % h - previous value of h
11 % Outputs:
12    % UnwrappedPhase - unwrapped phase [waves]
13    % l1 - phase shift (new h) chosen for PCO with lowest IWCL [waves]
14 % Required Functions:
15    % unwrap_ls.m
16    % wrap_wave.m
17    % Fast_IWCL.m
18
19 function [UnwrappedPhase l1]= LSPV_Plus_Opt3(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask , h);
20
21 [n m]=size(w_phase); % get size of grid
22 LS_phase=unwrap_ls(mask , w_phase)/(2*pi); % unwrapped waves
23
24
25 % initialize arrays
26 iwcl1=zeros(1,5);
27 iwcl2=zeros(1,5);
28 NonLS_phase1=zeros(n,m,5);
29 NonLS_phase2=zeros(n,m,6);
30
31 % set location of seeds
32 h1=h/2;
33 delta1=0.05;
34 delta2=0.15;
35 delta3=0.025;
36 delta4=0.075;
37
38 hh=[(h1-delta2) (h1-delta1) h1 (h1+delta1) (h1+delta2)];
39
40 for index=1:5
41     phaseshift=hh(index);
42     NonLS_phase1(:, :, index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
43     [iwcl1(index)]=Fast_IWCL(intensity , NonLS_phase1(:, :, index) , masksize , xSRI , ySRI , deltaSRI);
44 end
```

```

45
46 % Pick the best one
47 [cc, IIndex] = min(iwcl1);
48
49 % Use the best one for the starting pt of second iteration
50 h2=hh(IIndex);
51 hhh=[(h2-delta4) (h2-delta3) h2 (h2+delta3) (h2+delta4)];
52
53 for index=1:5
54     phaseshift=hhh(index);
55     NonLS_phase2(:, :, index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
56     [iwcl2(index)]=Fast_IWCL(intensity, NonLS_phase2(:, :, index), masksize, xSRI, ySRI, deltaSRI);
57 end
58
59 % Pick the best one
60 [cc, IIndex] = min(iwcl2);
61 UnwrappedPhase=LS_phase+NonLS_phase2(:, :, IIndex);
62
63 ll=hhh(IIndex);

```

Listing A.7. LSPV.Plus_Opt4

```

1 % Opt4 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % mask - the mask in phase space
7 % Outputs:
8     % UnwrappedPhase - unwrapped phase [waves]
9     % h - phase shift chosen [waves]
10 % Required Functions:
11     % unwrap_ls.m
12     % wrap_wave.m
13     % Fast_IWCL.m
14
15 function [UnwrappedPhase h]= LSPV.Plus_Opt4(intensity, w_phase, mask);
16
17 LS_phase=unwrap_ls(mask, w_phase)/(2*pi); % Unwrapped LS phase [waves]
18 NonLS_phase1 = wrap_wave(w_phase/(2*pi)-LS_phase); % Wrapped non-LS phase [waves]
19
20 h=sum(sum(intensity.*NonLS_phase1))/(sum(sum(intensity))); % Find h
21
22 NonLS_phase2 = wrap_wave(w_phase/(2*pi)-LS_phase-h); % Get new non-ls phase
23 UnwrappedPhase=LS_phase+NonLS_phase2; % Calculate total phase

```

Listing A.8. LSPV_Plus_Opt5

```
1 % Opt5 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % mask - the mask in phase space
7 % Outputs:
8     % UnwrappedPhase - unwrapped phase [waves]
9     % h - phase shift chosen [waves]
10 % Required Functions:
11     % unwrap_ls.m
12     % wrap_wave.m
13
14 function [UnwrappedPhase h]= LSPV_Plus_Opt5(intensity , w_phase , mask);
15
16 LS_phase=unwrap_ls(mask, w_phase)/(2*pi); % Unwrapped LS phase [waves]
17 NonLS_phase1 = wrap_wave(w_phase/(2*pi)-LS_phase); % Wrapped non-LS phase [waves]
18
19 % Get non-zero values
20 [rows , columns , PhaseValues]=find (mask.*NonLS_phase1);
21 % Sort them to put in order
22 SortedPhasevalues=sort (PhaseValues (:)' );
23 % Determine the size so we can find the median
24 s=size (SortedPhasevalues);
25 % Set h= to the median (middle value)
26 h=SortedPhasevalues (1,round (s (2)/2));
27
28 NonLS_phase2 = wrap_wave(w_phase/(2*pi)-LS_phase-h); % Get new non-ls phase
29 UnwrappedPhase=LS_phase+NonLS_phase2; % Calculate total phase
```

Listing A.9. LSPV_Plus_Opt6.m

```
1 % Opt6 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10    % ph - previous h value
11 % Outputs:
12    % UnwrappedPhase - unwrapped phase [waves]
```

```

13     % h - phase shift chosen for PCO with lowest IWCL [waves]
14 % Required Functions:
15     % unwrap_ls.m
16     % wrap_wave.m
17     % Fast_IWCL.m
18
19 function [UnwrappedPhase h]= LSPV.Plus_Opt6(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask , ph);
20
21 [n m]=size(w_phase); % get size of grid
22 LS_phase=unwrap_ls(mask , w_phase)/(2*pi); % unwrapped waves
23
24 % initialize arrays
25 iwcl1=zeros(1,5);
26 iwcl2=zeros(1,5);
27 NonLS_phase1=zeros(n,m,5);
28 NonLS_phase2=zeros(n,m,6);
29
30 % set location of seeds
31 h1=ph/2;
32 delta1=0.1;
33 delta2=0.3;
34 delta3=0.05;
35 delta4=0.15;
36
37 hh=[(h1-delta2) (h1-delta1) h1 (h1+delta1) (h1+delta2)];
38
39 for index=1:5
40     phaseshift=hh(index);
41     NonLS_phase1(:,:,index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
42     [iwcl1(index)]=Fast_IWCL(intensity , NonLS_phase1(:,:,index) , masksize , xSRI , ySRI , deltaSRI);
43 end
44
45 % Pick the best one
46 [cc, IIndex] = min(iwcl1);
47
48 % Use the best one for the starting pt of second iteration
49 h2=hh(IIndex);
50 hhh=[(h2-delta4) (h2-delta3) h2 (h2+delta3) (h2+delta4)];
51
52 for index=1:5
53     phaseshift=hhh(index);
54     NonLS_phase2(:,:,index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
55     [iwcl2(index)]=Fast_IWCL(intensity , NonLS_phase2(:,:,index) , masksize , xSRI , ySRI , deltaSRI);
56 end
57
58 % Pick the best one
59 [cc, IIndex] = min(iwcl2);
60 UnwrappedPhase=LS_phase+NonLS_phase2(:,:,IIndex);
61
62 h=hhh(IIndex); % Get phase shift from lowest IWCL

```

Listing A.10. LSPV_Plus_Opt7.m

```

1 % Opt7 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10    % NN - number of h evaluations desired
11    % RCold - Old rotational component
12 % Outputs:
13    % UnwrappedPhase - unwrapped phase [waves]
14    % h - phase shift chosen for PCO with lowest IWCL [waves]
15    % RCnew - new rotational component
16 % Required Functions:
17    % unwrap_ls.m
18    % wrap_wave.m
19    % Fast_IWCL.m
20
21 function [UnwrappedPhase h RCnew]= ...
22 LSPV_Plus_Opt7(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask , NN , RCold);
23
24 [n m]=size(w_phase);
25 LS_phase=unwrap_ls(mask,w_phase)/(2*pi); % LS phase [waves]
26
27 % initialize arrays
28 iwcl=zeros(1,NN);
29 Rhos=zeros(1,NN);
30 NonLS_phase=zeros(n,m,NN);
31
32 % Calculate NN rotational phases
33 hh=linspace(-.5,0.5,NN);
34 for index=1:NN
35     phaseshift=hh(index);
36     NonLS_phase(:,:,index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
37     [iwcl(index)]=Fast_IWCL(intensity , NonLS_phase(:,:,index) , masksize , xSRI , ySRI , deltaSRI);
38     NLS=NonLS_phase(:,:,index);
39     p=corrcoef(RCold(:),NLS(:)); % Calculate corr. coefficient with old non-LS phase
40     Rhos(index)=p(1,2); % Get scalar value for rho from array
41 end
42
43 % Invert and normalize IWCL
44 iwcl=(1./iwcl);
45 iwcl=iwcl/max(iwcl);
46 % Normalize corr. coeff.
47 Rhos=Rhos/max(Rhos);
48
49 weights=iwcl.*Rhos; % Create weights based on rho and 1/IWCL
50 [cc,IIndex] = max(weights); % Choose highest value

```

Listing A.11. LSPV_Plus_Opt8.m

```

1  % Opt8 unwrapper
2
3  % Inputs:
4      % intensity - intensity across the phase
5      % w_phase - wrapped phase [rad]
6      % masksize - physical size of mask [m]
7      % xSRI, ySRI - coordinates in phase space
8      % deltaSRI - grid spacing in phase space
9      % mask - the mask in phase space
10     % NN - number of h evaluations desired
11     % RCold - Old rotational component
12 % Outputs:
13     % UnwrappedPhase - unwrapped phase [waves]
14     % h - phase shift chosen for PCO with lowest IWCL [waves]
15     % RCnew - new rotational component
16 % Required Functions:
17     % unwrap_ls.m
18     % wrap_wave.m
19
20 function [UnwrappedPhase h RCnew]= ...
21 LSPV_Plus_Opt8(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask , NN , RCold);
22
23 [n m]=size(w_phase);
24 LS_phase=unwrap_ls(mask , w_phase)/(2*pi); % Unwrapped LS phase [waves]
25
26 % Initialize some arrays
27 Rhos=zeros(1,NN);
28 NonLS_phase=zeros(n,m,NN);
29
30 % Calculate NN rotational phases
31 hh=linspace(-.5 , 0.5 , NN);
32
33 for index=1:NN
34     phaseshift=hh(index);
35     NonLS_phase(:,:,index) = wrap_wave(w_phase/(2*pi)-LS_phase-phaseshift); % wrapped waves
36     NLS=NonLS_phase(:,:,index);
37     p=corrcoef(RCold(:),NLS(:)); % Calculate corr. coefficient with old non-LS phase
38     Rhos(index)=p(1,2); % Get scalar value for rho from array
39 end
40
41 [cc, IIndex] = max(Rhos); % Find max correlation
42
43 UnwrappedPhase=LS_phase+NonLS_phase(:,:,IIndex); % Compute total unwrapped phase
44 RCnew=NonLS_phase(:,:,IIndex); % Get new rotational component
45
46 hh=linspace(-.5 , 0.5 , NN);
47 h=hh(IIndex); % Get corresponding value of h

```

Listing A.12. PDF_Unwrapper.m

```

1 % Opt9 unwrapper (when h0 is set to previous h value)
2 % Opt11 unwrapper (when h0 is set to zero)
3
4 % Inputs:
5     % intensity - intensity across the phase
6     % w_phase - wrapped phase [rad]
7     % masksize - physical size of mask [m]
8     % xSRI, ySRI - coordinates in phase space
9     % deltaSRI - grid spacing in phase space
10    % mask - the mask in phase space
11    % ho - previous value of h [waves]
12    % window - width of probability bins surrounding h0
13    % bins - number of bins of equal prob. to divide window into
14 % Outputs:
15    % UnwrappedPhase - unwrapped phase [waves]
16    % h - phase shift chosen for PCO with lowest IWCL [waves]
17 % Required Functions:
18    % CL_Density_Fun.m
19    % CL_Density_FunB.m
20    % unwrap_ls.m
21    % wrap_wave.m
22    % Fast_IWCL.m
23
24 function [UnwrappedPhase h] = ...
25 PDF_Unwrapper(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask , h0 , window , bins );
26
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Estimate h values %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28
29 v=zeros(1,bins); % v stores the values of the CDF
30 Phaseshift=zeros(1,bins); % stores values of h to evaluate
31
32 % PT is the total amount of probability contained in the window centered
33 % around h0 and Vs is the starting value of PT
34 if (h0-window/2)<-0.5
35     Vs = CL_Density_Fun(-0.5);
36     PT = CL_Density_Fun(-0.5+window) - Vs;
37 elseif (h0+window/2)>0.5
38     Vs = CL_Density_Fun(0.5-window);
39     PT = CL_Density_Fun(0.5) - Vs;
40 else
41     PT = CL_Density_Fun(h0+window/2) - CL_Density_Fun(h0-window/2);
42     Vs = CL_Density_Fun(h0-window/2);
43 end
44
45 v(1)=Vs+0.5*(1/bins)*PT;
46 Phaseshift(1)=CL_Density_FunB(v(1));
47
48 for ind=2:bins
49     v(ind)=v(ind-1)+(1/bins)*PT;
50     Phaseshift(ind)=CL_Density_FunB(v(ind));
51 end

```

```

52
53 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LSPV+bins part %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
54
55 [n m]=size(w_phase);
56 LS_phase=unwrap_ls(mask,w_phase)/(2*pi); % Unwrapped LS phase [waves]
57
58 iwcl=zeros(1,bins);
59 NonLS_phase=zeros(n,m,bins);
60
61 % Calculate rotational phases for each bin
62 for index=1:bins
63     NonLS_phase(:, :, index) = wrap.wave(w_phase/(2*pi)-LS_phase-Phaseshift(index)); % wrapped waves
64     [iwcl(index)]=Fast_IWCL(intensity, NonLS_phase(:, :, index), masksize, xSRI, ySRI, deltaSRI);
65 end
66
67 % Pick the best one
68 [cc, IIndex] = min(iwcl);
69 UnwrappedPhase=LS_phase+NonLS_phase(:, :, IIndex);
70
71 h=Phaseshift(IIndex); % Find the corresponding h value

```

Listing A.13. CL_Density_Fun.m

```

1 % This subfunction gives the value of the CDF [prob.] given a value of h
2
3 % Input:
4     % h - phase shift of PCO
5 % Output:
6     % LorCDF - Probability of optimal IWCL given h
7 % Required Functions:
8     % none
9
10 function [LorCDF] = CL_Density_Fun(h);
11 h=round(h*1000)/1000;
12 u=inline('h>=0','h');
13 gama=0.072673*1.1;
14 LorPDF=(1.25./(pi*gama*(1+(h/gama).^2))-0.1131).*(u(h+0.5)-u(h-5));
15 LorCDF=(-0.1131*h+0.397887*atan(h/gama)+0.50).*(u(h+0.5)-u(h-5));

```


Listing A.14. CL_Density_FunB.m

```

1 % This subfunction gives the value of h corresponding to a CDF value v
2
3 % Input:
4     % v - value of CDF (probability)
5 % Output:
6     % H - corresponding phase shift from CDF
7 % Required Functions:
8     % none
9
10 function [H]= CL_Density_FunB(v);
11 h=-0.5:0.001:0.5;
12 u=inline('h>=0','h');
13 gama=0.072673*1.1;
14 LorPDF=(1.25./(pi*gama*(1+(h/gama).^2))-0.1131).*(u(h+0.5)-u(h-5));
15 LorCDF=(-0.1131*h+0.397887*atan(h/gama)+0.50).*(u(h+0.5)-u(h-5));
16 [C I]=min(abs(LorCDF-v));
17 H=h(I);

```

Listing A.15. GD.m

```

1 % Opt10 unwrapper
2
3 % Inputs:
4     % intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10 % Outputs:
11     % UnwrappedPhase - unwrapped phase [waves]
12     % h - phase shift chosen for PCO with lowest IWCL [waves]
13 % Required Functions:
14     % unwrap_ls.m
15     % h-vs-IWCL.m
16     % wrap-wave.m
17
18 function [UnwrappedPhase h]= GD(intensity, w_phase, masksize, xSRI, ySRI, deltaSRI, mask);
19
20 LS_phase=unwrap_ls(mask, w_phase)/(2*pi); % unwrapped waves
21 CutLength = @(h)h-vs-IWCL(intensity, w_phase, LS_phase, masksize, xSRI, ySRI, deltaSRI, h);
22 options = optimset('Display','off','MaxFunEvals',4,'TolX',.01);
23 h=fminbnd(CutLength,-0.5,0.5,options); % Use matlab function to minimize
24 NonLS_phase = wrap_wave(w_phase/(2*pi)-LS_phase-h); % [wrapped waves]

```

```
25 UnwrappedPhase=LS_phase+NonLS_phase; % Compute total phase
```

Listing A.16. h_vs_IWCL.m

```
1 % Subfunction for computing IWCL given inputs (mainly h)
2
3 % Inputs:
4     % intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % LS_phase - LS phase [rad]
7     % masksize - physical size of mask [m]
8     % xSRI, ySRI - coordinates in phase space
9     % h - value of h [waves]
10 % Outputs:
11     % iwcl - IWCL
12 % Required Functions:
13     % wrap_wave.m
14     % Fast_IWCL.m
15
16 function [iwcl]= h_vs_IWCL(intensity , w_phase , LS_phase , masksize , xSRI , ySRI , deltaSRI , h);
17
18 NonLS_phase = wrap_wave(w_phase/(2*pi)-LS_phase-h); % wrapped waves
19 [iwcl]=Fast_IWCL(intensity , NonLS_phase+LS_phase , masksize , xSRI , ySRI , deltaSRI );
```

Listing A.17. BretsMethod.m

```
1 % Opt12 unwrapper
2
3 % Inputs:
4     % Intensity - intensity across the phase
5     % w_phase - wrapped phase [rad]
6     % masksize - physical size of mask [m]
7     % xSRI, ySRI - coordinates in phase space
8     % deltaSRI - grid spacing in phase space
9     % mask - the mask in phase space
10    % W - window size/range of first set of points
11 % Outputs:
12    % UnwrappedPhase - unwrapped phase [waves]
13    % h - phase shift chosen for PCO with lowest IWCL [waves]
14    % RCnew - new rotational component
15 % Required Functions:
```

```

16     % unwrap_ls.m
17     % h_vs_IWCL.m
18     % Gold_Ratio.m
19     % wrap_wave.m
20
21     function [UnwrappedPhase h]= BretsMethod(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask , W);
22
23     LS_phase=unwrap_ls(mask , w_phase)/(2*pi) ; % Unwrapped LS phase [waves]
24     % Define function to calc. IWCL
25     CutLength = @(h)h_vs_IWCL(intensity , w_phase , LS_phase , masksize , xSRI , ySRI , deltaSRI , h);
26
27     iterations=2; % Number of search iterations
28
29     % Initialize storage for the various points calculated
30     a=zeros(1,iterations+1);
31     b=zeros(1,iterations+1);
32     c=zeros(1,iterations+1);
33     d=zeros(1,iterations+1);
34     m=zeros(1,iterations+1);
35     iwcl=zeros(iterations+1,5);
36
37     % Set starting points on edge of window
38     a(1)=0-W/2;
39     b(1)=0+W/2;
40
41     c(1)=Gold_Ratio([a(1) b(1)]);
42     d(1)=Gold_Ratio([a(1) c(1)]);
43     iwcl(1,:)=[CutLength(a(1)) CutLength(b(1)) CutLength(c(1)) CutLength(d(1)) 0];
44     A=a(1);AI=iwcl(1,1);
45     B=c(1);BI=iwcl(1,3);
46     C=b(1);CI=iwcl(1,2);
47     BA=B-A;
48     BC=B-C;
49     BICI=BI-CI;
50     BIAI=BI-AI;
51     % Approximate the min/max for each triplet of h values
52     m(1)=B-0.5*((BA^2)*BICI-(BC^2)*BIAI)/(BA*BICI-BC*BIAI);
53     if sum(isnan(m))>=1
54         t=isnan(m);
55         m(t)=m([t(end) t(1:end-1)]);
56     elseif abs(m(1))>0.5
57         m(1)=angle(exp(i*m(1)*2*pi))/(2*pi);
58     else
59     end
60     iwcl(1,5)=CutLength(m(1));
61
62     D=zeros(iterations+1,5);
63     D(1,:)=[a(1) b(1) c(1) d(1) m(1)];
64
65     for ind=2:iterations+1
66         if d(ind-1) <= c(ind-1)
67             if iwcl(ind-1,4) <= iwcl(ind-1,3)
68                 a(ind) = a(ind-1); iwcl(ind,1) = iwcl(ind-1,1);
69                 b(ind) = c(ind-1); iwcl(ind,2) = iwcl(ind-1,3);

```

```

70         c(ind) = d(ind-1); iwcl(ind,3) = iwcl(ind-1,4);
71         d(ind) = Gold_Ratio([a(ind) c(ind)]); iwcl(ind,4) = CutLength(d(ind));
72     elseif iwcl(ind-1,4) > iwcl(ind-1,3)
73         a(ind) = d(ind-1); iwcl(ind,1) = iwcl(ind-1,4);
74         b(ind) = b(ind-1); iwcl(ind,2) = iwcl(ind-1,2);
75         c(ind) = c(ind-1); iwcl(ind,3) = iwcl(ind-1,3);
76         d(ind) = Gold_Ratio([a(ind) c(ind)]); iwcl(ind,4) = CutLength(d(ind));
77     else
78     end
79 elseif d(ind-1) > c(ind-1)
80     if iwcl(ind,4) ≤ iwcl(ind,3)
81         a(ind) = c(ind-1); iwcl(ind,1) = iwcl(ind-1,3);
82         b(ind) = b(ind-1); iwcl(ind,2) = iwcl(ind-1,2);
83         c(ind) = d(ind-1); iwcl(ind,3) = iwcl(ind-1,4);
84         d(ind) = Gold_Ratio([a(ind) c(ind)]); iwcl(ind,4) = CutLength(d(ind));
85     elseif iwcl(ind,4) > iwcl(ind,3)
86         a(ind) = a(ind-1); iwcl(ind,1) = iwcl(ind-1,1);
87         b(ind) = d(ind-1); iwcl(ind,2) = iwcl(ind-1,4);
88         c(ind) = c(ind-1); iwcl(ind,3) = iwcl(ind-1,3);
89         d(ind) = Gold_Ratio([a(ind) c(ind)]); iwcl(ind,4) = CutLength(d(ind));
90     else
91     end
92 else
93 end
94
95 % Create a-d for a smaller parabolic approx equation
96 A=a(ind); AI=iwcl(ind,1);
97 B=c(ind); BI=iwcl(ind,3);
98 C=b(ind); CI=iwcl(ind,2);
99 BA=B-A;
100 BC=B-C;
101 BICI=BI-CI;
102 BIAI=BI-AI;
103 % Approximate the min/max for each triplet of h values
104 m(ind)=B-0.5*((BA^2)*BICI-(BC^2)*BIAI)/(BA*BICI-BC*BIAI);
105 if sum(isnan(m))≥1
106     t=isnan(m);
107     m(t)=m([t(end) t(1:end-1)]);
108 elseif abs(m(ind))>0.5
109     m(ind)=angle(exp(i*m(ind)*2*pi))/(2*pi);
110 else
111 end
112 iwcl(ind,5)=CutLength(m(ind));
113 D(ind,:)= [a(ind) b(ind) c(ind) d(ind) m(ind)];
114 end
115
116 CL=iwcl(:)';
117 D=D(:)';
118 [cc, IIndex] = min(CL);
119 h=D(IIndex);
120 NonLS_phase = wrap_wave(w_phase/(2*pi)-LS_phase-h); % wrapped waves
121 UnwrappedPhase=LS_phase+NonLS_phase; % Compute total phase

```

Listing A.18. Gold_Ratio.m

```
1 % This subfunction computes the third point based on the golden ratio of a
2 % given range
3
4 % Input:
5 % In - [a b] vector containing two end points of a bracket
6 % Output:
7 % Out - point c in between a and b
8 % Required Functions:
9 % None
10 function [Out]= Gold_Ratio(In);
11 if size(In,2)==3
12     a=abs(In(2)-In(1));
13     b=abs(In(3)-(In(2)));
14     if a>b
15         c=a-b;
16         Out=In(2)-c;
17     else
18         c=b-a;
19         Out=In(2)+c;
20     end
21 elseif size(In,2)==2
22     a=In(2)-In(1);
23     Out=0.6180*a+In(1);
24 else
25     end
```

Listing A.19. Golden_Hybrid.m

```
1 % Opt13 unwrapper
2
3 % Inputs:
4 % Intensity - intensity across the phase
5 % w_phase - wrapped phase [rad]
6 % masksize - physical size of mask [m]
7 % xSRI, ySRI - coordinates in phase space
8 % deltaSRI - grid spacing in phase space
9 % mask - the mask in phase space
10 % W - Window/range of first search centered at zero [waves]
11 % Outputs:
12 % UnwrappedPhase - unwrapped phase [waves]
13 % h - phase shift chosen for PCO with lowest IWCL [waves]
14 % Required Functions:
15 % unwrap_ls.m
16 % h-vs-IWCL.m
```

```

17     % Gold_Probability.m
18     % wrap_wave.m
19
20 function [UnwrappedPhase h]= Golden_Hybrid(intensity , w_phase , masksize , xSRI , ySRI , deltaSRI , mask , W);
21
22 LS_phase=unwrap_ls(mask , w_phase)/(2*pi); % unwrapped waves
23 % Define function to calc. IWCL
24 CutLength = @(h)h_vs_IWCL(intensity , w_phase , LS_phase , masksize , xSRI , ySRI , deltaSRI , h);
25
26 iterations=2; % Number of searches to execute
27
28 % Initialize storage for the various points calculated
29 a=zeros(1,iterations+1);
30 b=zeros(1,iterations+1);
31 c=zeros(1,iterations+1);
32 d=zeros(1,iterations+1);
33 m=zeros(1,iterations+1);
34 iwcl=zeros(iterations+1,5);
35
36 % Set starting points on edge of window
37 a(1)=0-W/2;
38 b(1)=0+W/2;
39
40 c(1)=Gold_Probability([a(1) b(1)]); % Get new point for GS search
41 d(1)=Gold_Probability([a(1) c(1)]); % Get new point for GS search
42 iwcl(1,:)=[CutLength(a(1)) CutLength(b(1)) CutLength(c(1)) CutLength(d(1)) 0];
43 A=a(1);AI=iwcl(1,1);
44 B=c(1);BI=iwcl(1,3);
45 C=b(1);CI=iwcl(1,2);
46 BA=B-A;
47 BC=B-C;
48 BICI=BI-CI;
49 BIAI=BI-AI;
50
51 % Approximate the min/max for each triplet of h values
52 m(1)=B-0.5*((BA^2)*BICI-(BC^2)*BIAI)/(BA*BICI-BC*BIAI);
53 if sum(isnan(m))≥1
54     t=isnan(m);
55     m(t)=m([t(end) t(1:end-1)]);
56 elseif abs(m(1))>0.5
57     m(1)=angle(exp(i*m(1)*2*pi))/(2*pi);
58 else
59 end
60 iwcl(1,5)=CutLength(m(1));
61
62 D=zeros(iterations+1,5);
63 D(1,:)=[a(1) b(1) c(1) d(1) m(1)]
64
65 for ind=2:iterations+1
66
67     if d(ind-1) ≤ c(ind-1)
68         if iwcl(ind-1,4) ≤ iwcl(ind-1,3)
69             a(ind) = a(ind-1); iwcl(ind,1) = iwcl(ind-1,1);
70             b(ind) = c(ind-1); iwcl(ind,2) = iwcl(ind-1,3);

```

```

71         c(ind) = d(ind-1); iwcl(ind,3) = iwcl(ind-1,4);
72         d(ind) = Gold_Probability([a(ind) c(ind)]); iwcl(ind,4) = CutLength(d(ind));
73     elseif iwcl(ind-1,4) > iwcl(ind-1,3)
74         a(ind) = d(ind-1); iwcl(ind,1) = iwcl(ind-1,4);
75         b(ind) = b(ind-1); iwcl(ind,2) = iwcl(ind-1,2);
76         c(ind) = c(ind-1); iwcl(ind,3) = iwcl(ind-1,3);
77         d(ind) = Gold_Probability([a(ind) c(ind)]); iwcl(ind,4) = CutLength(d(ind));
78     else
79     end
80 elseif d(ind-1) > c(ind-1)
81     if iwcl(ind,4) ≤ iwcl(ind,3)
82         a(ind) = c(ind-1); iwcl(ind,1) = iwcl(ind-1,3);
83         b(ind) = b(ind-1); iwcl(ind,2) = iwcl(ind-1,2);
84         c(ind) = d(ind-1); iwcl(ind,3) = iwcl(ind-1,4);
85         d(ind) = Gold_Probability([a(ind) c(ind)]); iwcl(ind,4) = CutLength(d(ind));
86     elseif iwcl(ind,4) > iwcl(ind,3)
87         a(ind) = a(ind-1); iwcl(ind,1) = iwcl(ind-1,1);
88         b(ind) = d(ind-1); iwcl(ind,2) = iwcl(ind-1,4);
89         c(ind) = c(ind-1); iwcl(ind,3) = iwcl(ind-1,3);
90         d(ind) = Gold_Probability([a(ind) c(ind)]); iwcl(ind,4) = CutLength(d(ind));
91     else
92     end
93 else
94 end
95
96 % Create a-d for a smaller parabolic approx equation
97 A=a(ind); AI=iwcl(ind,1);
98 B=c(ind); BI=iwcl(ind,3);
99 C=b(ind); CI=iwcl(ind,2);
100 BA=B-A;
101 BC=B-C;
102 BICI=BI-CI;
103 BIAI=BI-AI;
104 % Approximate the min/max for each triplet of h values
105 m(ind)=B-0.5*((BA^2)*BICI-(BC^2)*BIAI)/(BA*BICI-BC*BIAI);
106 if sum(isnan(m))≥1
107     t=isnan(m);
108     m(t)=m([t(end) t(1:end-1)]);
109 elseif abs(m(ind))>0.5
110     m(ind)=angle(exp(i*m(ind)*2*pi))/(2*pi);
111 else
112 end
113 iwcl(ind,5)=CutLength(m(ind));
114 D(ind,:)= [a(ind) b(ind) c(ind) d(ind) m(ind)];
115 end
116
117 CL=iwcl(:)';
118 D=D(:)';
119 [cc, IIndex] = min(CL);
120 h=D(IIndex);
121 NonLS_phase = wrap_wave(w_phase/(2*pi)-LS_phase-h); % wrapped waves
122 UnwrappedPhase=LS_phase+NonLS_phase;

```

Listing A.20. Gold_Probability.m

```
1 % This subfunction computes the third point based on the golden ratio of a
2 % given range, in terms of probabilities
3
4 % Input:
5     % In - [a b] vector containing two end points of a bracket
6 % Output:
7     % Out - point c in between a and b
8 % Required Functions:
9     % CL_Density_Fun
10    % CL_Density_FunB
11
12 function [C]= Gold_Probability(In);
13 w=CL_Density_Fun(In(2))-CL_Density_Fun(In(1));
14 if abs(In(1)) <= abs(In(2))
15     Pc=CL_Density_Fun(In(1))+ 0.6180*w;
16     C=CL_Density_FunB(Pc);
17 else
18     Pc=CL_Density_Fun(In(2))- 0.6180*w;
19     C=CL_Density_FunB(Pc);
20 end
```


Bibliography

- [1] *Joint Vision 2020*. Washington DC, 2000. Director for Strategic Plans and Policy, J5; Strategy Division.
- [2] Andrews, L. and R. Phillips. *Laser Beam Propagation through Random Media*. SPIE, Bellingham WA, second edition, 2005.
- [3] Arrasmith, W.W. “Branch-point-tolerant least-squares phase reconstructor”. *Appl. Opt.*, 16(7):1864–1872, 1999.
- [4] Balmer, R., G.W. Davidson, and N. Adam. “On the Nature of Noise in 2D phase-unwrapping”. *SPIE*, 2958:216–225, 1996.
- [5] Barchers, Jeffrey D. *Control law for a high resolution self-referencing interferometer wave-front sensor used with a low resolution deformable mirror AR-17*. Technical report, Longmont CO, December 2002.
- [6] Born, M. and E. Wolf. *Principles of optics, Electromagnetic theory of propagation, interference and diffraction of light*. Cambridge University Press, New York NY, 7th edition, 2006.
- [7] Brennan, T. J. and T. A. Rhoadarmer. “Performance of a woofer-tweeter deformable mirror control architecture for high-bandwidth high-spatial resolution adaptive optics”. *SPIE*, 6306(9):63060B–1–63060B–12, 2006.
- [8] Brennan, T.J. and P.H. Roberts. *The Adaptive Optics Toolbox, For Use with MATLAB*. Anaheim CA, 2008. the Optical Sciences Company.
- [9] Brennan, T.J., P.H. Roberts, and D.C. Zimmerman. *A Wave Optics Simulation System, For Use with MATLAB*. Anaheim CA, 2008. the Optical Sciences Company.
- [10] Cusack, R., J.M. Huntley, and H.T. Goldrein. “Improved noise-immune phase-unwrapping algorithm”. *Appl. Opt.*, 34(5):781–789, 1995.
- [11] Dekany, R. “Wavefront sensing for adaptive optics”. Caltech Optical Observatories, Pasadena CA, August 2009.
- [12] Ellerbroek, B. L. “Efficient computation of minimum-variance wave-front reconstructors with sparse matrix techniques”. *J. Opt. Soc. Am. A*, 19(9):1803–1816, 2002.
- [13] Fried, D. L. “Adaptive optics wave function reconstruction and phase-unwrapping when branch points are present”. *Opt. Commun.*, 200:43–72, 1992.
- [14] Fried, D. L. “Branch Cuts in the Phase Function”. *J. Opt. Soc. Am. A*, 31(15):2865–2881, 1992.

- [15] Fried, D. L. “Branch Point Problem in Adaptive Optics”. *J. Opt. Soc. Am. A*, 15(10):2759–2768, 1998.
- [16] Geary, J.M. *Introduction to Wavefront Sensors*. SPIE, Bellingham WA, 1995.
- [17] Ghiglia, D.C. and M.D. Pritt. *Two-Dimensional phase-unwrapping Theory, Algorithms, and Software*. John Wiley and Sons, Inc., New York NY, 1998.
- [18] Goodman, J.W. *Statistical Optics*. John Wiley and Sons, Inc., New York NY, 1985.
- [19] Goodman, J.W. *Introduction to Fourier Optics*. Roberts and Company Publishers, Greenwood Village CO, 3rd edition, 2005.
- [20] Gutmann, B. and H. Weber. “Phase-unwrapping with the branch-cut method: role of phase-field direction”. *Appl. Opt.*, 39(26):4802–4816, 2000.
- [21] Hardy, J. W. *Adaptive optics for astronomical telescopes*. Oxford University Press, New York NY, 1998.
- [22] Herraiez, M.A., D.R. Burton, M.J. Lalor, and D.B. Clegg. “Robust, simple, and fast algorithm for phase-unwrapping”. *Appl. Opt.*, 35(29):5847–5852, 1996.
- [23] Huntley, J.M. “Three-dimensional noise-immune phase-unwrapping algorithm”. *Appl. Opt.*, 40(23):3901–3908, 2001.
- [24] Huntley, J.M. and J.R. Buckland. “Characterization of sources of 2π phase discontinuity in speckle interferograms”. *J. Opt. Soc. Am. A*, 12(9):1990–1996, 1995.
- [25] Jablon, R. “Test Success: Laser weapon knocks out missile”. *Associated Press*, 12 Feb, 2010.
- [26] Kreyszig, E. *Advanced Engineering Mathematics*. John Wiley and Sons, Inc., New York NY, 1999.
- [27] Le Bigot, Eric O. and W. Wild. “Theory of branch-point detection and its implementation”. *Appl. Opt.*, 16(7):1724–1729, 1999.
- [28] M.A. Herraiez, *et al.* “Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path”. *Appl. Opt.*, 41(35):7434–7444, 2002.
- [29] Mathews, J. H. *Numerical Methods for Mathematics, Science, and Engineering*. Prentice Hall, Englewood Cliffs NJ, 2nd edition, 1992.
- [30] Missile Defense Agency, U.S. Department of Defense. “Missile Defense Test Conducted”. <http://www.mda.mil/news/10news0001.html>.

- [31] Olson, D.F., S.M. Long, and L.J. Ulibarri. “Comparison of Complex Exponential and Least Squares Wavefront Reconstructors for Regular Square and Non-Square grid Arrays”. 4091, 323–332. *Proc. SPIE*, July 2000.
- [32] Perkins, M.P. and R.J. Vernon. “Two-dimensional phase-unwrapping to help characterize an electromagnetic beam for quasi-optical mode converter design”. *Appl. Opt.*, 47(35):6606–6614, 2008.
- [33] Porter, J. *Adaptive Optics for Vision Science Principals, Practices, Design, and Applications*. John Wiley and Sons, Inc., New York NY, 2006.
- [34] Poyneer, L.A. “Wavefront Reconstruction”. Lawrence Livermore National Laboratory, Livermore CA, AO Summer School, August 2009.
- [35] Press, W., B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press, New York NY, 1986.
- [36] Pritt, M.D. “Congruence in least-squares phase-unwrapping”. *Geoscience and Remote Sensing, 1997. IGARSS '97. Remote Sensing - A Scientific Vision for Sustainable Development., 1997 IEEE International*, volume 2, 875–877. Aug 1997.
- [37] Pritt, M.D. “Comparison of path-following and least-squares phase-unwrapping algorithms”. *Geoscience and Remote Sensing, 1997. IGARSS '97. Remote Sensing - A Scientific Vision for Sustainable Development., 1997 IEEE International*, 2, Aug 1997.
- [38] Rhoadarmer, T.A. “Wave front reconstruction using a second-order model for Shack-Hartmann wave front sensor measurements”. 4724, 17–29. *Proc. SPIE*, July 2002.
- [39] Rhoadarmer, T.A. “Development of a self-referencing interferometer wave-front sensor”. volume 5553, 112-126. SPIE, 2004.
- [40] Rhoadarmer, T.A. and J.D. Barchers. “Noise analysis for complex field estimation using a self-referencing interferometer wave front sensor”. 4825, 215–227. *Proc. SPIE*, July 2002.
- [41] Roggemann, M. C. and B. Welsh. *Imaging Through Turbulence*. CRC Press Inc., Boca Raton FL, 1996.
- [42] Roggemann, M.C. and A.C. Koivunen. “Branch-point reconstruction in laser beam projection through turbulence with finite-degree-of-freedom phase-only wave-front correction”. *J. Opt. Soc. Am. A*, 17(1):53–62, 2000.

- [43] S.A. Karout, D.R. Burton, M. A. Gdeisat and M. J. Lalor. “Two-dimensional phase-unwrapping using a hybrid genetic algorithm”. *Appl. Opt.*, 46(5):730–743, 2007.
- [44] Sasiela, R. J. *Electromagnetic Wave Propagation in Turbulence*. SPIE, Bellingham WA, 2nd edition, 2007.
- [45] Schmidt, J. *Numerical Simulation of Optical Wave Propagation, With Examples in MATLAB*. in press, Wright-Patterson AFB, October edition, 2009.
- [46] Scott, D. and D. Robie. “Directed Energy, A Look to the Future”. *Air and Space Power*, XXIII(4):6–12, 2009.
- [47] Tyson, R.K. *Introduction to Adaptive Optics*. SPIE, Bellingham Washington, 2000.
- [48] Venema, T. M. and J. D. Schmidt. “Optical phase-unwrapping in the presence of branch points”. *J. Opt. Soc. Am. A*, 16(10):6985–6998, 2008.
- [49] Venema, Todd M. *Closed-Loop Adaptive Optics Control in Strong Atmospheric Turbulence*. Ph.D. dissertation, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, August 2008. AFIT/DS/ENG/08-21.
- [50] Voitsekhovich, V.V., D. Kouznetsov, and D.K. Morozov. “Density of turbulence-induced phase dislocations”. *Appl. Opt.*, 37(21):4525–4535, 1998.
- [51] Wild, W. and Eric O. Le Bigot. “Rapid and robust detection of branch points from wave-front gradients”. *Opt. Lett.*, 24(4):1724–1729, 1999.

Vita

Captain Casey Pellizzari received a BSEE from California State University, Fullerton in May 2005 and was commissioned in August of the same year. His first assignment was to the Electronic Systems Center at Hanscom AFB, MA. There he served as a force protection research and development engineer for the 642 Electronic System Squadron. In August 2008, he was assigned to the Air Force Institute of Technology, Wright-Patterson AFB, OH, to earn a MSEE in Electro-Optics. His follow-on assignment is to AFRL Detachment 15, Maui Island, HI. Capt Pellizzari is a member of Tau Beta Pi Engineering Honor Society, Eta Kappa Nu Electrical and Computer Engineering Honor Society, and the International Society for Optical Engineering.

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) 25-03-2010		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) Aug 2008 — Mar 2010	
4. TITLE AND SUBTITLE PHASE UNWRAPPING IN THE PRESENCE OF STRONG TURBULENCE				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER F1ATA0835J002	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Casey J. Pellizzari, Capt, USAF				5d. PROJECT NUMBER ENGJON229	
				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 (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/10-23	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research Kent Miller 3875 Randolph St., Suite 3112 Arlington, VA 22203 (703) 696-8573, kent.miller@afosr.af.mil				10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR/NE	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Phase unwrapping in the presence of branch points using a least-squares wave-front reconstructor requires the use of a Postprocessing Congruence Operation (PCO). Branch cuts in the unwrapped phase are altered by the addition of a constant parameter h to the rotational component when applying the PCO. Past research has shown that selecting a value of h which minimizes the proportion of irradiance in the pupil plane adjacent to branch cuts helps to maximize performance of adaptive-optics (AO) systems in strong turbulence. In continuation of this objective, this research focuses on optimizing the PCO while accounting for the cumulative effects of the integral control law. Several optimizations are developed and compared using wave-optics simulations. The most successful optimization is shown to reduce the normalized variance of the Strehl ratio across a wide range turbulence strengths and frame rates, including decreases of up to 25 percent when compared to a non-optimized PCO algorithm. AO systems which depend on high, steady Strehl ratio values serve to benefit from these algorithms.					
15. SUBJECT TERMS Branch Points, Branch Cuts, Strong Turbulence, Phase-Unwrapping, Post-Processing Congruence Operation, Rotational Phase					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj Jason D. Schmidt
U	U	U	U	154	19b. TELEPHONE NUMBER (include area code) (937) 255-3636, x7224; jason.schmidt@afit.edu