

A TRIDENT SCHOLAR PROJECT REPORT

NO. 433

A Modeling and Data Analysis of Laser Beam Propagation in the Maritime Domain

by

Midshipman 1/C Benjamin C. Etringer, USN



UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

This document has been approved for public
release and sale; its distribution is limited.

U.S.N.A. --- Trident Scholar project report; no. 433 (2015)

**A MODELING AND DATA ANALYSIS OF LASER BEAM PROPAGATION
IN THE MARITIME DOMAIN**

by

Midshipman 1/C Benjamin C. Etringer
United States Naval Academy
Annapolis, Maryland

Certification of Adviser Approval

Professor Reza Malek-Madani
Mathematics Department

(signature)

(date)

Acceptance for the Trident Scholar Committee

Professor Maria J. Schroeder
Associate Director of Midshipman Research

(signature)

(date)

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
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 this 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) 05-18-2015		2. REPORT TYPE		3. DATES COVERED (From - To)
4. TITLE AND SUBTITLE A Modeling and Data Analysis of Laser Beam Propagation in the Maritime Domain		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Etringer, Benjamin Charles		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Naval Academy Annapolis, MD 21402		10. SPONSOR/MONITOR'S ACRONYM(S)		
		11. SPONSOR/MONITOR'S REPORT NUMBER(S) Trident Scholar Report no. 433 (2015)		
12. DISTRIBUTION / AVAILABILITY STATEMENT This document has been approved for public release; its distribution is UNLIMITED.				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT In this project we investigate the impact of the maritime environment on the propagation of laser beams. This study primarily uses data collected at the Naval Academy with the goal of quantifying the correlation between the statistics of the environmental parameters and the statistics of laser beam intensity at the target. The project has two parts to it: 1) we present a computational analysis of different probability density function approximation techniques; and 2) we introduce preliminary steps towards developing a stochastic model for the maritime laser beam propagation. In the first part of this work we apply three mathematical methods to construct the probability density function of the data: i) the Kernel Density Estimator (KDE) method, ii) the Barakat Method using lower-order moments, and iii) the Bayesian Mixture Model. We compare and contrast the features of the three approximation techniques, first in the context of a synthetic data whose true pdf is known, and next in the context of the laser data. In the second task, we analyze how a complex medium causes the photons of the laser light to behave differently than if they were acting in freespace, by focusing on the stochastic behavior that our data exhibits. We develop a stochastic paraxial wave equation in order to have a mathematical model capable of accepting statistical parameters from the atmosphere as input to allow us to investigate the statistical properties of light intensity at a specified target.				
15. SUBJECT TERMS laser propagation, data analysis, Bayesian statistics, probability density function				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 59
a. REPORT	b. ABSTRACT	c. THIS PAGE		
				19a. NAME OF RESPONSIBLE PERSON
				19b. TELEPHONE NUMBER (include area code)

Abstract

In this project we investigate the impact of the maritime environment on the propagation of laser beams. This study primarily uses data collected at the Naval Academy with the goal of quantifying the correlation between the statistics of the environmental parameters and the statistics of laser beam intensity at the target. The project has two parts to it: 1) we present a computational analysis of different probability density function approximation techniques; and 2) we introduce preliminary steps towards developing a stochastic model for the maritime laser beam propagation. In the first part of this work we apply three mathematical methods to construct the probability density function of the data: i) the Kernel Density Estimator (KDE) method, ii) the Barakat Method using lower-order moments, and iii) the Bayesian Mixture Model. We compare and contrast the features of the three approximation techniques, first in the context of a synthetic data whose true pdf is known, and next in the context of the laser data. In the second task, we analyze how a complex medium causes the photons of the laser light to behave differently than if they were acting in freespace, by focusing on the stochastic behavior that our data exhibits. We develop a stochastic paraxial wave equation in order to have a mathematical model capable of accepting statistical parameters from the atmosphere as input to allow us to investigate the statistical properties of light intensity at a specified target.

Keywords: laser propagation, data analysis, Bayesian statistics, probability density function.

Acknowledgments

I take this opportunity to express sincere gratitude to all the faculty members who have guided me throughout my time at the Academy. In particular, this project required a great deal of work and would not have been possible without the support and patience from my adviser, Professor Reza Malek-Madani. Thanks is also given to Professor Maria Schroeder and the Trident Committee for their support. Finally, I am deeply grateful for the support from my family and friends.

Contents

Abstract	1
Acknowledgments	2
List of Figures	5
1 Introduction and Motivation	7
2 Laser Beam Propagation in Free Space	9
2.1 Exact Solution of PWE in half-space	10
3 Review of Histogram and PDF Concepts	13
3.1 Histograms	13
3.2 Normal "Gaussian" Probability Density Function	16
4 Laser Data Conveyed Through Histograms	18
4.1 Analysis of Nighttime Data Using Histograms	18
4.2 Daytime Laser Data Analysis Using Histograms	20
5 Barakat Method for Computing PDFs Using Lower-Order Moments	22
6 Kernel Density Method for Computing PDFs	27
6.1 Kernel Method Implemented on Synthetic Data	28
7 Gaussian Mixture Method	30
8 Description of Comparison Methods and Implementation of PDF Approximation Techniques on Synthetic Data	32
8.1 Kolmogorov-Smirnov (K-S) Test	32
8.2 Least Squares Test	33
8.3 Hellinger Distance	33
8.4 Application of K-S Test and Least Squares Test on Kernel Method with Synthetic Data from Gaussian Distribution	34
8.5 Hellinger Distance on Synthetic Data	37

9	Implementation and Comparison of PDF Approximation Techniques on Laser Data	40
9.1	Nighttime Laser Data	40
9.2	Daytime Laser Data	43
10	Solution to Paraxial Wave Equation with Stochastic Refraction Coefficient	46
11	Conclusion	48
11.1	Barakat	48
11.2	Kernel	48
11.3	Gaussian Mixture Method	48
11.4	Summary	49
	Appendix	51
	Barakat Function	51
	ECDF Computation Function	52
	Final PDF Comparison	53
	Gaussian Mixture Method Function	54
	Hellinger Distance	56
	Kernel Method Function	57
	PDF to CDF Converting Function	58

List of Figures

1.1	Time Series Plot of Laser Data.	8
2.1	Contours of the intensity, i.e., $V_1^2 + V_2^2$ at a fixed distance ($z = 1200$ meters) from the aperture. This figure is obtained with parameter values $\lambda = 633$ nanometers, $F_0 = 500$ meters, $W_0 = 0.03$ meters.	12
3.1	Example of a histogram.	13
3.2	Histogram of Ages with 10 Bins.	14
3.3	Histogram of Ages with 2 Bins.	15
3.4	Histogram of Ages with 50 Bins.	15
3.5	Gaussian PDF with $\mu = 0$ and $\sigma = 1$	16
3.6	Comparison of Different Values for μ	17
3.7	Comparison of Different Values for σ	17
4.1	Histogram of Black Data with 10 Bins.	18
4.2	Histogram of Black Data with 100 Bins.	19
4.3	Histogram of Black Data with 1,000 Bins.	19
4.4	Histogram of Black Data with 10,000 Bins.	19
4.5	Histogram of Daytime Laser Data with 10 Bins.	20
4.6	Histogram of Black Data with 100 Bins.	20
4.7	Histogram of Daytime Data with 1,000 Bins.	21
4.8	Histogram of Daytime Data with 10,000 Bins.	21
5.1	Barakat Method Implemented on Synthetic Data.	26
6.1	Pictorial Representation of the Kernel Method.	28
6.2	Kernel Method Implemented on Synthetic Data.	29
7.1	Gaussian Mixture Method Implemented on Synthetic Data with 13 Clusters.	31
8.1	K-S Test Implemented on 50 Synthetic Data Points from Gaussian Distribution.	33
8.2	Comparison of Kernel CDF, ECDF, and Gaussian CDF.	34
8.3	Zoomed in Comparison of Kernel CDF, ECDF, and Gaussian CDF.	35
8.4	Barakat CDF Implemented on Synthetic Data.	36
8.5	Zoomed in Comparison of Barakat CDF, ECDF, and Gaussian CDF.	37
8.6	Table of Average Hellinger Distance after 50 Simulations.	37
8.7	Table of Full Results of Hellinger Distances after 50 Simulations.	38

8.8	Visual Comparison of the Three Approximation Techniques Implemented on Synthetic Data.	39
9.1	Time Series Function of Laser Intensity During Nighttime 1.	41
9.2	Visual Comparison of the Three Approximation Techniques Implemented on Nighttime Data 1.	41
9.3	Time Series Function of Laser Intensity During Nighttime 2.	41
9.4	Visual Comparison of the Three Approximation Techniques Implemented on Nighttime Data 2.	42
9.5	Visual Representation for Convergence of Barakat Method for Nighttime Laser Data.	42
9.6	Time Series Function of Laser Intensity During Daytime 1.	43
9.7	Visual Comparison of the Three Approximation Techniques Implemented on Daytime Data 1.	44
9.8	Time Series Function of Laser Intensity During Daytime 2.	44
9.9	Visual Comparison of the Three Approximation Techniques Implemented on Daytime Data 2.	44
9.10	Visual Representation for Convergence of Barakat Method for Daytime Laser Data. 45	
10.1	Simulated Time Series Laser Data.	47
10.2	PDF of Simulated Time Series Laser Data.	47

Chapter 1

Introduction and Motivation

The intensity of laser light plays an integral role in many naval applications. From laser-based weapon systems to determining distances with Laser Range Finders, the Navy has much to gain from increased knowledge in this field. Philip Nielsen, in *Effects of Directed Energy Weapons*, [8], emphasizes the uses for lasers in naval weaponry. Andrews and Phillips, *Laser Beam Propagation Through Random Media*, [1], develop the mathematical framework associated with laser beams, and we intend to use these principles in an effort to further investigate the mathematical properties of laser beams in the maritime domain. Our goal by the end of the project is to better understand and quantify the properties of laser light by analyzing the intensities of laser light after the light has traveled through complex media. It is well-known that fluctuations in air temperature, as well as the presence of aerosols and other scatterers, impacts the direction of propagation of a beam as well as the amount of power it may deposit on a target (see Chapter 2 of [1]).

This project primarily deals with the mathematical and computational analysis of data collected from a Helium-Neon laser. The data we analyze has already been collected by a group supervised by Prof. S. Avramov-Zamurovic of the Weapons and Systems Engineering Department. The data is of light intensity and is collected from a field experiment, in which we have set up a laser and sensor in stationary locations on Sherman Field, located on the premises of the Naval Academy. Figure 2 shows a typical stochastic behavior of the signals in the data and our goal is to understand the extent of the contribution of the atmospheric conditions to the stochastic behavior of light intensity in signals such as the one in Figure 2. To that end, we will use the recorded atmospheric parameters, including humidity, precipitation, temperature, wind, and cloud cover, with the goal of correlating the propagation properties with the effects of the turbulence in the medium. Most of the laser beam data we have is from a target that is stationed 375 meters from the propagation source (see [6] for a detailed description of the data collection). We will analyze the environmental diversity in data, ranging from the ambient light (day or night) to the aerosol conditions (land vs. water); we will compute the statistical parameters of the data in order to construct its probability density function, with the goal of identifying the signature of the atmospheric turbulence in our data.

The sensor that detects photons of light is capable of recording intensities of the laser beam at a rate of 10000 Hz. That rate is equivalent to taking a data point every 150 microseconds. Data is collected for approximately three minutes, resulting in over a million data points to work with for each data set. The data is stored electronically, and then converted into a vector that can be interpreted by MATLAB. Currently over 100 such data sets are available on file for a variety of

different atmospheric scenarios, equating to more than 100 million data points for the intensity of laser light. See Figure 1.1 for an example of a time series plot of laser data. This graph shows the intensity of a recorded laser beam at a single pixel on a sensor located 375 meters from the source. The duration of the time series is 180 seconds, at a time interval of $150 \mu s$.

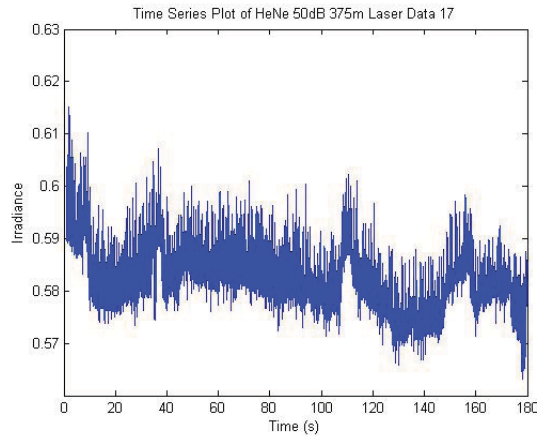


Figure 1.1: Time Series Plot of Laser Data.

The structure of this thesis is as follows: In Chapter 2 we review the structure of a laser beam in free space. In Chapter 3 we review concepts of histograms and probability density functions, and we will create histograms for our laser data in Chapter 4. In Chapter 5, the method of Barakat will be introduced. Chapter 6 is dedicated to the Kernel Density Method. Chapter 7 consists of the computation related to the Gaussian Mixture Methods. Chapter 8 provides a comparison and contrast of the three methods for computing a pdf using synthetic data. Chapter 9 implements these techniques on real data collected in the maritime domain. Chapter 10 provides an introduction into future research by computing a solution to the paraxial wave equation that with a stochastic refraction coefficient. Chapter 11 is the conclusion section.

Chapter 2

Laser Beam Propagation in Free Space

Light propagation is governed by Maxwell's equations. Let $\{\mathbf{E}, \mathbf{B}\}$ be the electric and magnetic field pair that defines light. Then in a region with no charges or currents, Maxwell's equations are

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad \nabla \times \mathbf{B} = \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t}, \quad (2.1)$$

$$\nabla \cdot \mathbf{E} = 0, \quad \nabla \cdot \mathbf{B} = 0. \quad (2.2)$$

Here c is the speed of light and is related to the medium's permittivity ϵ_0 and permeability μ_0 by

$$c^2 = \frac{1}{\epsilon_0 \mu_0}.$$

In what follows, where we closely follow the development in Chapter 4 of [1], we will show that each component of $\mathbf{E}$ and $\mathbf{B}$ satisfies the wave equation. To see this first take the curl of the second equation in (2.1) and use the identity $\nabla \times (\nabla \times \mathbf{E}) = \nabla(\nabla \cdot \mathbf{E}) - \Delta \mathbf{E}$, and the first equation in (2.2), $\nabla \cdot \mathbf{E} = 0$, to get

$$\Delta \mathbf{E} = \frac{\partial}{\partial t} (\nabla \times \mathbf{B})$$

Next, use the first equation in (2.1), $\nabla \times \mathbf{B} = \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t}$, to eliminate $\mathbf{B}$ and get a single equation for $\mathbf{E}$:

$$\Delta \mathbf{E} = \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2}.$$

The above equation is the well-known and standard wave equation.

A similar development shows that $\mathbf{B}$ satisfies the same wave equation. Since the equations are linear, we conclude that each component of either $\mathbf{E}$ or $\mathbf{B}$ satisfies the *scalar* wave equation

$$\Delta u = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2}. \quad (2.3)$$

Next we look for special solutions of the wave equation (2.3), namely, monochromatic solutions of the form

$$u(\mathbf{r}, t) = v(\mathbf{r})e^{i\omega t}, \quad \mathbf{r} = \langle x, y, z \rangle$$

After substituting the above template into (2.3), we see that v satisfies Helmholtz's equation

$$\Delta v + k^2 v = 0, \quad k^2 = \frac{\omega^2}{c^2}. \quad (2.4)$$

Finally, we look for beam-like solutions, i.e.,

$$v(\mathbf{r}) = V(\mathbf{r})e^{ikz}. \quad (2.5)$$

Note that the function V in (2.5) is complex-valued, i.e., $V = V_1 + iV_2$. We substitute the template (2.5) into the Helmholtz equation (2.4) and get

$$\Delta_{\perp} V + \frac{\partial^2 V}{\partial z^2} + 2ik \frac{\partial V}{\partial z} = 0,$$

where $\Delta_{\perp} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.

One of the main assumptions of laser beam propagation theory (again, see Chapter 4, [1]) is the assumption that the term $|\frac{\partial^2 V}{\partial z^2}|$ is much smaller than $|2ik \frac{\partial V}{\partial z}|$, and is therefore ignored in the above equation, resulting in the partial differential equation

$$\Delta_{\perp} V + 2ik \frac{\partial V}{\partial z} = 0. \quad (2.6)$$

The above equation is called the Paraxial Wave Equation (PWE). We emphasize again that this equation is complex-valued.

2.1 Exact Solution of PWE in half-space

Continuing to follow the material in Chapter 4 of [1], we now present the exact solution of (2.6) when V is known at the aperture $z = 0$. It turns out that if $V(x, y, 0)$ is Gaussian, a term that we will define precisely in the next section, then $V(x, y, z)$ remain Gaussian for all z : Let

$$V(x, y, 0) = A \exp\left(-\frac{r^2}{W_0^2} + \frac{ikr^2}{2F_0}\right), \quad (2.7)$$

where W_0 and F_0 are called the spot-size radius and the phase radius of curvature of the beam at $z = 0$, then (2.6) has the exact solution

$$V(x, y, z) = \frac{A}{\sqrt{\Lambda_0^2 + \Theta_0^2}} \exp\left(-\frac{r^2}{W(z)^2} + i\left(\phi + \frac{kr^2}{2F(z)}\right)\right), \quad (2.8)$$

where Λ_0 , called the refraction parameter, and Θ_0 , the diffraction parameter, are

$$\Lambda_0 = 1 - \frac{z}{F_0}, \quad \Theta_0 = \frac{2z}{kW_0^2} \quad (2.9)$$

and W and F , which are the spot-size radius of the beam and its phase radius of curvature at any z , are

$$W(z) = W_0 \sqrt{\Lambda_0^2 + \Theta_0^2}, \quad \phi = \tan^{-1} \frac{\Lambda_0}{\Theta_0} \quad (2.10)$$

$$F(z) = F_0 \frac{(\Theta_0^2 + \Lambda_0^2)(\theta_0 - 1)}{\Theta_0^2 + \Lambda_0^2 - \Theta_0} \quad (2.11)$$

Figure 2.6 shows a typical instance of the solution of PWE. In this figure the contours of intensity (irradiance) of a laser beam are plotted at a fixed distance z for the aperture – the intensity of light is defined as $|V|^2$. As expected, the beam has kept its gaussian shape far from the aperture. This feature, that a beam starting out as a gaussian, and remaining gaussian for all values of z is a special feature of beams propagating in vacuum or free space. In the problems that we take up in this paper, the laser beams that propagate in the maritime domain must interact with the atmosphere and consequently the gaussian nature of the beam is destroyed. One of the main goals of this paper is to attempt to quantify the impact of the atmosphere on a laser beam based on the data we collect from various experiments. The next chapter introduces some of the mathematical tools we need for this quantification.

Figure 2.6 is obtained by executing the following program in MATLAB:

```
lambda=633*10^-9; k=2*pi/lambda;F0=500;W0=0.03;
Theta0=inline('1-z/F0','z','F0');
Lambda0=inline('2*z/k/W0^2','z','k','W0');
z=0:1200;
TH=Theta0(z,F0); LA=Lambda0(z,k,W0);
W=W0*sqrt(TH.^2+LA.^2);
figure(1)
plot(z,W)
title('Graph of W')
F=F0*(TH.^2+LA.^2).*(TH-1)./(TH.^2+LA.^2-TH);
figure(2)
plot(z,F)
title('Graph of F')
[X,Y]=meshgrid(-0.05:0.001:0.05,-0.05:0.001:0.05);
z=1200;
TH=Theta0(z,F0); LA=Lambda0(z,k,W0);
W=W0*sqrt(TH.^2+LA.^2);
Irradiance=1./(TH.^2+LA.^2).*exp(-2*(X.^2+Y.^2)./W.^2);
figure(3)
[c, h]=contour(X,Y,Irradiance)
clabel(c,h)
colorbar
title('Contours of Irradiance at z =1200')
```

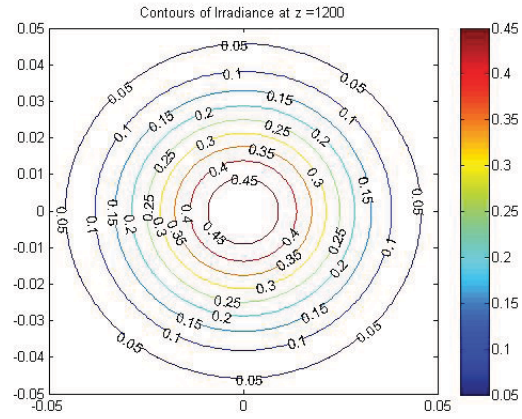


Figure 2.1: Contours of the intensity, i.e., $V_1^2 + V_2^2$ at a fixed distance ($z = 1200$ meters) from the aperture. This figure is obtained with parameter values $\lambda = 633$ nanometers, $F_0 = 500$ meters, $W_0 = 0.03$ meters.

Chapter 3

Review of Histogram and PDF Concepts

3.1 Histograms

We begin by analyzing the data using histograms. Histograms, arguably the most elementary method of analyzing data points, display the frequency of occurrence of a random variable in an interval or bin. See Figure 3.1 for an example of a histogram, in which we take a sample of 400 points from a normal distribution with mean 0 and standard deviation 1. We create ten equally spaced bins and place the points in the corresponding bins to create our histogram. The MATLAB code used to create the histogram in Figure 3.1 is stated below:

```
>> data=randn(400,1);  
>> hist(data,10)  
>> title('Normally Distributed Random Numbers')
```

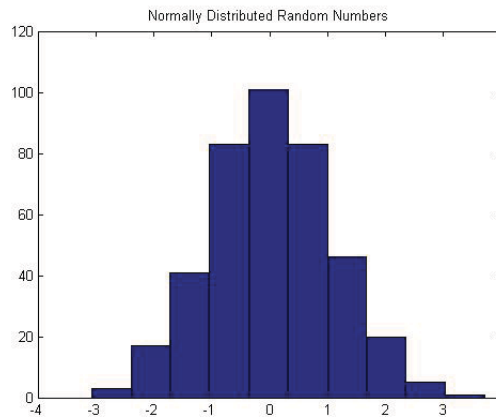


Figure 3.1: Example of a histogram.

In addition to being easy to create, histograms are not computationally strenuous. For example, we will see with later methods that it is computationally impossible to use all the data points of a data set, while, with histograms, we can use all the data points. Future methods will require us to create functions for each data point and sum the functions together. The summation can be com-

putationally difficult if the sample size of our data is significantly high. However, with histograms, the points are simply placed into pre-allocated bins, which is not computationally difficult. By using all the data points in our data set, we ensure that we are not ignoring any crucial data points.

The bin size corresponds to the width of each bar in the histogram. Perhaps the easiest way to understand bins is with an example. Consider the ages of 100 random individuals who are walking into a Wal-Mart, given below—synthetic Wal-Mart data was generated from a uniform distribution of integers from 0-99.

58	32	27	89	2	41	39	23	78	77
64	10	35	23	88	31	15	0	23	76
60	56	97	65	97	36	24	8	47	78
20	86	63	93	72	12	1	81	61	97
16	9	60	7	40	19	2	52	7	89
22	38	4	70	84	45	38	4	86	27
54	72	81	9	36	76	41	57	83	79
73	21	5	94	80	93	91	65	37	87
96	48	87	13	0	0	7	64	92	95
53	38	19	3	59	3	36	24	32	23

A bin size of 10 would allow us to focus on the variation of customers whose ages range within a ten year period. For example, the first bin takes into account all customers between the ages of 0 and 9, the second bin those whose ages range between 10 and 19, and so forth. Figure 3.2 shows the histogram of the data:

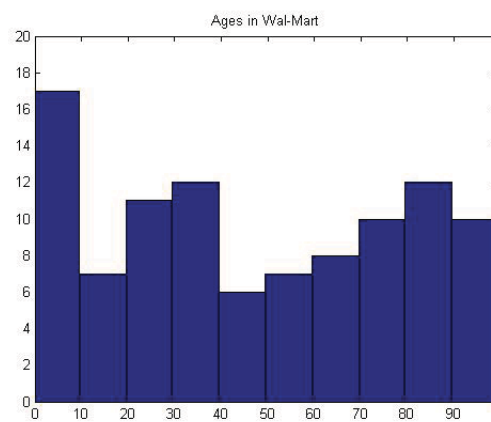


Figure 3.2: Histogram of Ages with 10 Bins.

It is easy to see from the histogram that we had 17 people in the age range of 0-9 years old, and 6 people in the age range of 40-49 years old. If we change the number of bins to fewer than ten, we lose some of the details of the variability that the data carries. Figure 3.3 shows the histogram when we use only two bins:

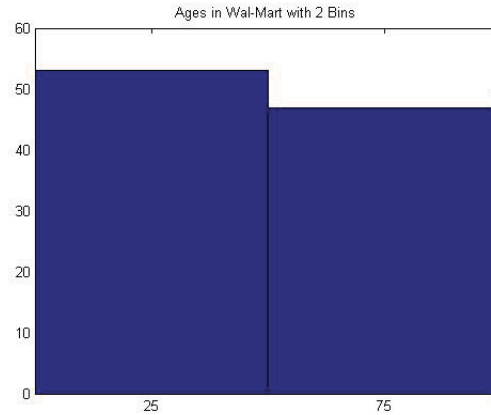


Figure 3.3: Histogram of Ages with 2 Bins.

Similarly, if we have too many bins, we can lose meaningful information contained in the data. In Figure 3.4, for example, there are holes in the histogram that depict a 0% probability of an individual walking into Wal-Mart between the ages of 16 and 18. However, we know that the data was pulled from a uniform distribution. Therefore, there should be just as much chance as seeing an individual walk into Wal-Mart between the ages of 16-18 and 18-20. Hence, we need to decrease the bin size in order to plot a more accurate histogram for the synthetic, uniformly distributed, Wal-Mart data.

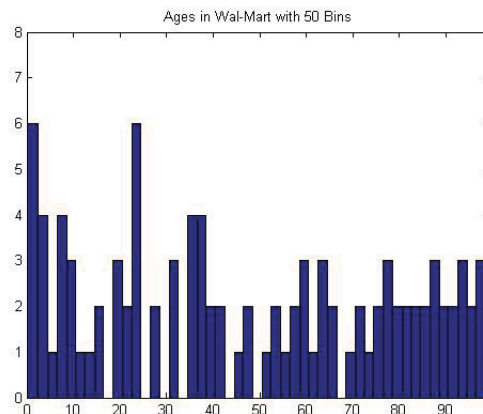


Figure 3.4: Histogram of Ages with 50 Bins.

What we should see from the Wal-Mart example is that we need a sufficiently large number of bins with our histograms in order to create a histogram that is well represented by the data, but not make the number so large that we lose the overall shape that the histogram should display. We will do the same process with our Laser Data later in Chapter 3.

3.2 Normal "Gaussian" Probability Density Function

A probability density function (pdf) is a function used for determining the probability of a continuous random variable; particularly the probability that a certain data point will fall between two values (a and b). In order to determine the probability, we compute the area under the curve of the pdf, or the integral from a to b . That is, a probability density function $f(x)$ is defined as,

$$P(a < x < b) = \int_a^b f(x)dx.$$

By definition, the area under any pdf must equal 1. Which means

$$\int_{-\infty}^{\infty} P(x)dx = 1 \quad (3.1)$$

The cumulative density function (cdf) is used when comparing different pdfs using the Kolmogorov-Smirnov (K-S) Test or Least Squares Test (described in Chapter 7). The cdf, $F(x)$ is defined as,

$$F(x) = \int_{-\infty}^x f(t)dt. \quad (3.2)$$

Note that $\lim_{x \rightarrow -\infty} F(x) = 0$ and $\lim_{x \rightarrow \infty} F(x) = 1$.

The Gaussian pdf is perhaps the most commonly used pdf in statistics and physical applications. The Gaussian pdf is sometimes colloquially referred to as the "bell curve" based on its shape. The Gaussian pdf has two arguments: the mean (μ) and standard deviation (σ), and is given by the following equation:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (3.3)$$

The standard Gaussian pdf has $\mu = 0$ and $\sigma = 1$. See Figure 3.5.

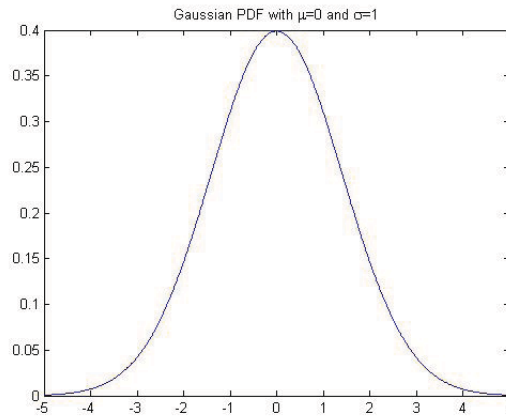


Figure 3.5: Gaussian PDF with $\mu = 0$ and $\sigma = 1$.

Let us note that $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$. We should also note that a change in μ corresponds to a horizontal shift, or translation, of the pdf. In addition, a change in σ corresponds to a horizontal stretch of the pdf. See Figures 3.6 and 3.7 as a reference:

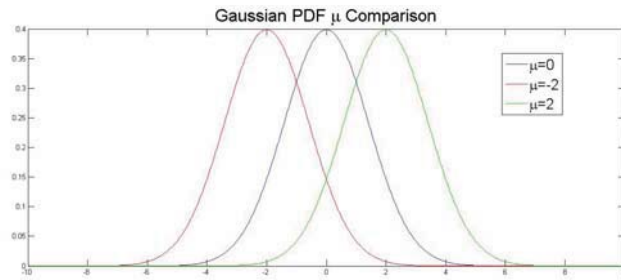


Figure 3.6: Comparison of Different Values for μ .

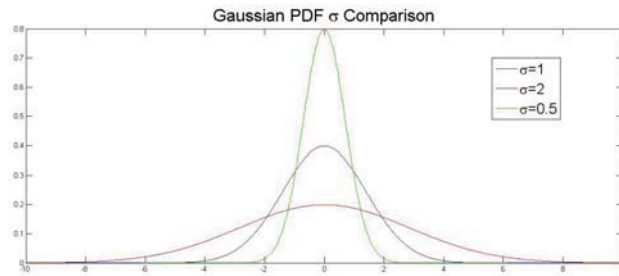


Figure 3.7: Comparison of Different Values for σ .

We will use the Gaussian pdf extensively throughout the remainder of the paper in various ways to compute approximate data-driven pdfs. Both the Kernel Method and Gaussian Mixture Method use the Gaussian pdf as a basis for the approximate pdf.

Chapter 4

Laser Data Conveyed Through Histograms

For this paper we will rely on two data sets, one collected during the daytime and the other at nighttime. The first data set that we will consider will be nighttime data. The data was taken in the middle of the night, when there was little to no light interfering with the laser beam. We will use this as our control for other sets of data. The other data set we will use is daytime data, which was collected while the sun was actively impacting the laser beam. The data was collected from a stationary sensor, located 375 meters away from the laser aperture, that recorded scalar intensity values at a rate of 10 kHz for three minutes. The wavelength of the laser beam is 633 nm.

4.1 Analysis of Nighttime Data Using Histograms

We will begin by looking at a histogram with only 10 bins. See Figure 4.1. We can see that the histogram above does not reveal much information about the data set other than giving us the upper and lower bounds. Therefore, we will increase the number of bins by an order of magnitude. The resulting histogram, shown in Figure 4.2, reveals more information about the data, but still does not depict the features that some of the histograms with smaller bin sizes are able to depict.

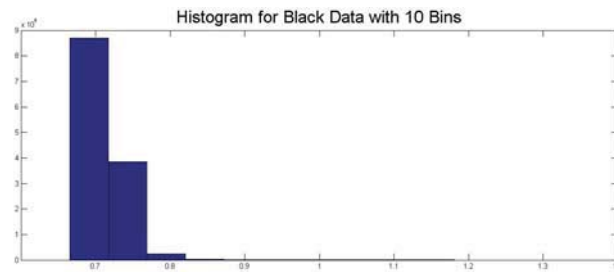


Figure 4.1: Histogram of Black Data with 10 Bins.

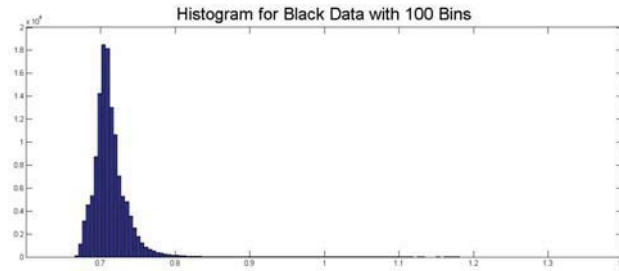


Figure 4.2: Histogram of Black Data with 100 Bins.

Figure 4.2 depicts the overall skewed nature of the pdf, but with over a million points of data, we can increase the number of bins by another order of magnitude in order to improve the shape of our histogram. The resulting histogram, as seen in Figure 4.3.

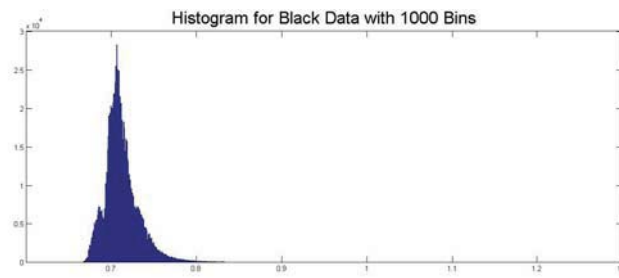


Figure 4.3: Histogram of Black Data with 1,000 Bins.

The bins are so narrow now, that we can no longer distinguish well the lines that separate bins. We should note that the current histogram exhibits significantly more features than the histogram with 10 bins. For example, we can now see two local maximums in our histogram plot with 1,000 bins that we could not see with only 10 bins. To ensure that no other local maximums occur in the histogram plot of the data, we should look at another histogram with 10,000 bins. See Figure 4.4 for histogram:

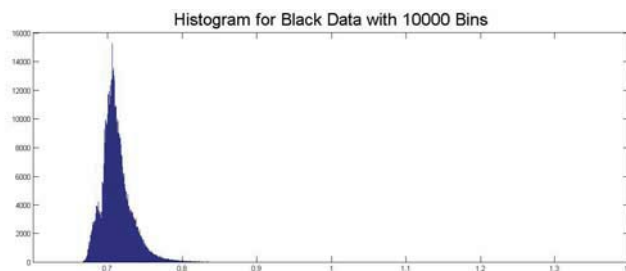


Figure 4.4: Histogram of Black Data with 10,000 Bins.

Note that there is not much of a difference between the histogram with 1,000 bins vs. the histogram with 10,000 bins. Therefore, we will use the histogram with 1,000 bins for our analysis later in the paper.

The following is the MATLAB code used to create the above histograms (example for 100 bins):

```
>> laser=ConvertedData.Data.MeasuredData(1, 4).Data;
>> hist(laser,100)
>> title('Histogram for Black Data with 100 Bins')
```

4.2 Daytime Laser Data Analysis Using Histograms

The Daytime Laser Data was taken during the daytime with the same laser and sensor. The distances are the same (375 meters). We will now do the same process for finding the best histogram that we did for the Black Data. First, let's look at the data under 10 bins. See Figure 4.5 for histogram:

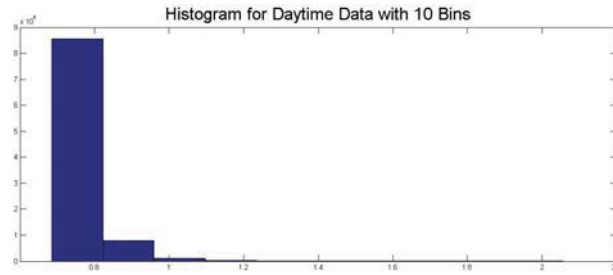


Figure 4.5: Histogram of Daytime Laser Data with 10 Bins.

Again, we can see that with only 10 bins, we cannot conclude much about the shape of the probability density function we would like to find. Outside of determining upper and lower bounds, we cannot obtain much from this histogram. Hence, we should increase the number of bins by an order of magnitude. See Figure 4.6 for histogram:

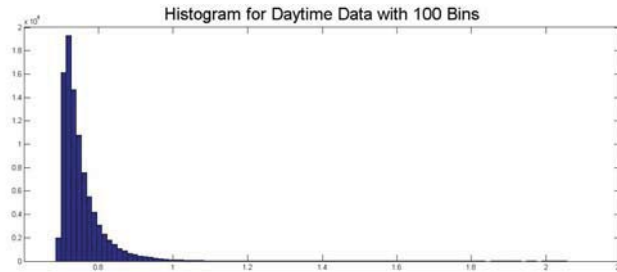


Figure 4.6: Histogram of Black Data with 100 Bins.

We can see a much better trend with 100 bins, but much of the variance in the data is still hidden with the smaller number of bins. Therefore we will continue to increase the number of bins by another order of magnitude. See Figure 4.7 for histogram:

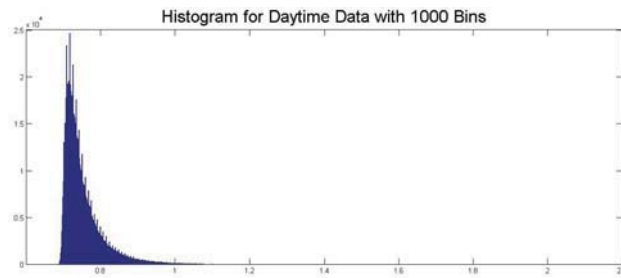


Figure 4.7: Histogram of Daytime Data with 1,000 Bins.

Note, with 1,000 bins we can see spikes on the right-hand side of our histogram. These spikes may make it more difficult to approximate the curve for a probability density function in the future. Therefore, we will use a different histogram to base our analysis off of. By increasing the order of magnitude again, we see what our histogram is still well below the number of bins that it would take to lose the trends of the data set. See Figure 4.8 for histogram:

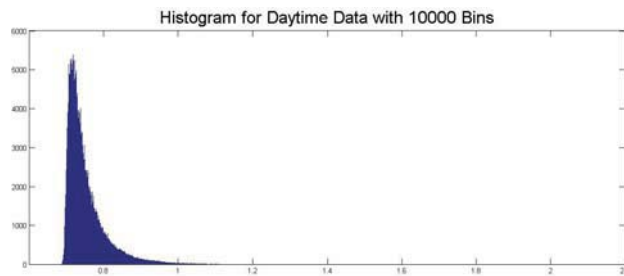


Figure 4.8: Histogram of Daytime Data with 10,000 Bins.

We can see that in this histogram, the spikes are no longer as prevalent in the right-hand side of the histogram. We can see a substantial amount of detail in the data, and therefore, we will use this histogram as a basis for our future probability density function.

Chapter 5

Barakat Method for Computing PDFs Using Lower-Order Moments

The concept of computing probability density functions for laser light intensities has attracted the attention of numerous researchers. One of these researchers, Richard Barakat, proposes a method that incorporates Gamma functions and the Associated Laguerre Polynomials to create a unimodal probability density function for a laser data set [2]. Barakat's method for computing pdfs using lower-order moments can be described, below, by $W(h)$, where W_N are the mixing weights, L_N are the Associated Laguerre Polynomials, and $W_g(h)$ is the Gamma pdf:

$$W(h) = W_g(h) \sum_{N=0}^{\infty} W_N L_N^{\beta-1} \left(\frac{\beta h}{\mu} \right) \quad (5.1)$$

where,

$$W_g(h) = \frac{1}{\Gamma(\beta)} \left(\frac{\beta}{h} \right)^{\beta} h^{\beta-1} \exp(-\beta h/\mu), \quad (5.2)$$

and,

$$\beta = \frac{\langle h \rangle^2}{\langle h^2 \rangle - \langle h \rangle^2}, \quad (5.3)$$

and

$$L_N^{\beta-1}(x) = \sum_{n=0}^N \binom{N+\beta-1}{N-n} \frac{(-x)^n}{n!}, \quad (5.4)$$

and

$$W_N = N! \Gamma(\beta) \sum_{n=0}^N \frac{(-\beta/\mu)^n \langle h^n \rangle}{n! (N-n)! \Gamma(\beta+n)}. \quad (5.5)$$

Barakat argues that the first five terms of h are sufficient to approximate the PDF of h reasonably well [2]. In addition, Barakat asserts that the first three terms are independent of whatever data set we look at. Particularly, $W_0 = 1, W_1 = W_2 = 0$. We will prove his assertion of the first three terms, and attempt to evaluate his assumption for the first five terms for laser data that exhibits significant noise once propagated in the maritime domain.

Definition: $\langle h \rangle$ is the arithmetic mean of the set h . That is,

$$\langle h \rangle = \frac{h_1 + h_2 + h_3 + \dots + h_n}{n}. \quad (5.6)$$

Theorem: $W_0 = 1, W_1 = W_2 = 0$.

Proof. Show $W_0 = 1, W_1 = W_2 = 0$.

We start with the equation,

$$W_N = N! \Gamma(\beta) \sum_{n=0}^N \frac{(-\beta/\mu)^n \langle h^n \rangle}{n!(N-n)! \Gamma(\beta+n)}$$

Let $N = 0$.

$$W_0 = 0! \Gamma(\beta) \sum_{n=0}^0 \frac{(-\beta/\mu)^n \langle h^n \rangle}{n!(0-n)! \Gamma(\beta+n)}$$

$$W_0 = 0! \Gamma(\beta) \frac{(-\beta/\mu)^0 \langle h^0 \rangle}{0!(0-0)! \Gamma(\beta+0)}$$

Since $\langle h^0 \rangle = \langle 1 \rangle = 1$,

$$W_0 = 1 \Gamma(\beta) \frac{(1)(1)}{(1)(1) \Gamma(\beta)}$$

$$W_0 = 1 \frac{\Gamma(\beta)}{\Gamma(\beta)} = 1.$$

Let $N = 1$.

$$W_1 = 1! \Gamma(\beta) \sum_{n=0}^1 \frac{(-\beta/\mu)^n \langle h^n \rangle}{n!(1-n)! \Gamma(\beta+n)}$$

$$W_1 = 1! \Gamma(\beta) \left(\frac{(-\beta/\mu)^0 \langle h^0 \rangle}{0!(1-0)! \Gamma(\beta+0)} + \frac{(-\beta/\mu)^1 \langle h^1 \rangle}{1!(1-1)! \Gamma(\beta+1)} \right)$$

$$W_1 = \Gamma(\beta) \left(\frac{(1) \langle 1 \rangle}{(1)(1)! \Gamma(\beta)} + \frac{(-\beta/\mu) \langle h \rangle}{1!(0)! \Gamma(\beta+1)} \right)$$

Since $\langle h \rangle = \mu$,

$$W_1 = \Gamma(\beta) \left(\frac{1}{\Gamma(\beta)} + \frac{(-\beta/\mu) \mu}{\Gamma(\beta+1)} \right)$$

$$W_1 = \frac{\Gamma(\beta)}{\Gamma(\beta)} - \frac{\beta \Gamma(\beta)}{\Gamma(\beta+1)}$$

Since $\beta\Gamma(\beta) = \Gamma(\beta + 1)$,

$$W_1 = 1 - \frac{\Gamma(\beta + 1)}{\Gamma(\beta + 1)}$$

$$W_1 = 1 - 1 = 0$$

Let $N = 2$.

$$W_2 = 2!\Gamma(\beta) \sum_{n=0}^2 \frac{(-\beta/\mu)^n \langle h^n \rangle}{n!(2-n)!\Gamma(\beta+n)}$$

After expanding the summation,

$$W_2 = 2!\Gamma(\beta) \left(\frac{(-\beta/\mu)^0 \langle h^0 \rangle}{0!(2-0)!\Gamma(\beta+0)} + \frac{(-\beta/\mu)^1 \langle h^1 \rangle}{1!(2-1)!\Gamma(\beta+1)} + \frac{(-\beta/\mu)^2 \langle h^2 \rangle}{2!(2-2)!\Gamma(\beta+2)} \right)$$

$$W_2 = 2\Gamma(\beta) \left(\frac{(1) \langle 1 \rangle}{(1)(2)\Gamma(\beta)} + \frac{(-\beta/\mu) \langle h \rangle}{(1)1!\Gamma(\beta+1)} + \frac{(\beta/\mu)^2 \langle h^2 \rangle}{(2)0!\Gamma(\beta+2)} \right)$$

Using identities from previous derivations and simplifying,

$$W_2 = 2\Gamma(\beta) \left(\frac{1}{2\Gamma(\beta)} - \frac{(\beta/\mu)\mu}{\Gamma(\beta+1)} + \frac{(\beta/\mu)^2 \langle h^2 \rangle}{2\Gamma(\beta+2)} \right)$$

Distributing $2\Gamma(\beta)$ through,

$$W_2 = \frac{2\Gamma(\beta)}{2\Gamma(\beta)} - \frac{2\beta\Gamma(\beta)}{\Gamma(\beta+1)} + \frac{(\beta/\mu)^2 \langle h^2 \rangle 2\Gamma(\beta)}{2\Gamma(\beta+2)}$$

Since $\Gamma(\beta+2) = (\beta)(\beta+1)\Gamma(\beta)$,

$$W_2 = 1 - \frac{2\Gamma(\beta+1)}{\Gamma(\beta+1)} + \frac{2(\beta/\mu)^2 \langle h^2 \rangle}{2\beta(\beta+1)}$$

$$W_2 = 1 - 2 + \frac{\beta^2 \langle h^2 \rangle}{\mu^2 \beta(\beta+1)}$$

See claim below for further proof:

$$W_2 = -1 + \frac{\beta \langle h^2 \rangle}{\mu^2(\beta+1)} = -1 + 1 = 0.$$

Claim :

$$\frac{\beta \langle h^2 \rangle}{\mu^2(\beta+1)} = 1$$

Numerator :

$$\beta \langle h^2 \rangle = \left(\frac{\mu^2}{\langle h^2 \rangle - \mu^2} \right) \langle h^2 \rangle$$

$$\beta < h^2 > = \frac{\mu^2 < h^2 >}{< h^2 > - \mu^2}$$

Denominator :

$$\begin{aligned}\mu^2(\beta + 1) &= \mu^2 \left(\left(\frac{\mu^2}{< h^2 > - \mu^2} \right) + 1 \right) \\ \mu^2(\beta + 1) &= \mu^2 \left(\frac{\mu^2}{< h^2 > - \mu^2} + \frac{< h^2 > - \mu^2}{< h^2 > - \mu^2} \right) \\ \mu^2(\beta + 1) &= \mu^2 \left(\frac{< h^2 >}{< h^2 > - \mu^2} \right) \\ \mu^2(\beta + 1) &= \frac{\mu^2 < h^2 >}{< h^2 > - \mu^2}\end{aligned}$$

□

For our MATLAB code, we will let $I = \frac{h}{< h >}$. Then our functions for the lower-order pdf described by Barakat become:

$$W(I) = W_g(I) \left[1 + \sum_{N=3}^{\infty} W_N L_N^{\beta-1}(\beta I) \right]$$

Where,

$$W_g(I) = \frac{1}{\Gamma(\beta)} \beta^\beta I^{\beta-1} \exp(-\beta I)$$

And

$$W_N = N! \Gamma(\beta) \sum_{n=0}^N \frac{(-\beta)^n \langle I^n \rangle}{n! (N-n)! \Gamma(\beta+n)}$$

We define W_g in MATLAB with the following code:

```
function [ W_g2 ] = Barakat_W_g2( beta, I )
    W_g2=1/gamma(beta)*(beta)^beta*I.^(beta-1).*exp(-beta*I);
end
```

We define W_N in MATLAB with the following code:

```
function [ W ] = Barakat_W_n2( beta, I, N)
    ww=0;
    for n=0:N
        term=(-beta)^n*mean(I.^n)./(factorial(n)*...
            factorial(N-n)*gamma(beta+n));
        ww=ww+term;
    end
    W=factorial(N)*gamma(beta)*ww;
```

end

And we define $L_N^{\beta-1}(x)$ in MATLAB with the following code:

```
function [ L ] = Laguerre_Beta( beta, x, N )
    L=0;
    for n=0:N
        L_term=gamma_nchoosek(N+beta-1,N-n)*(-x).^n/factorial(n);
        L=L+L_term;
    end
end
```

Using the functions defined above, we can write a script to execute $W(I)$ to generate an approximation to probability density function for our data. $W(I)$ can be executed with the following code (see Figure 5.1 for execution of code)

```
.
x=%imported_data;
x=sort(x); I=x/mean(x);
beta=mean(x)^2/(mean(x.^2)-mean(x)^2);
NN=5; WW=0;
for N=3:NN
    term=Barakat_W_n2(beta, I, N).*Laguerre_Beta2(beta, beta*I, N);
    WW=WW+term;
end
W=Barakat_W_g2(beta, I).*(1+WW);
plot(x,abs(W)/trapz(x,abs(W)))
```

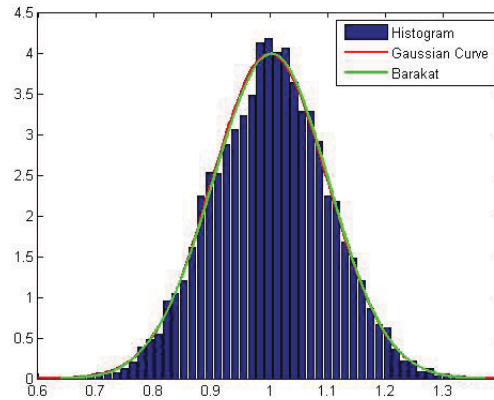


Figure 5.1: Barakat Method Implemented on Synthetic Data.

Again, by visual inspection, the Barakat Method appears to represent the data well. Once we convert the pdf into a cdf and perform the K-S Test and Least Squares test, we can see that the Barakat Method also represents the data better than the Gaussian CDF (See Figures 8.4 & 8.5).

Chapter 6

Kernel Density Method for Computing PDFs

A second approach to computing pdfs is the Kernel Density Method (Reference [9] has an introduction to the method), which we will apply to compute the pdf of our laser data.

Given a time series $x = x_i$ and a kernel function $K(x)$, we define an approximation $P_K(x)$ of the pdf by:

$$p_K(x) = \frac{1}{N} \sum_{i=1}^N G_n, \quad (6.1)$$

where G_n is a pdf of a distribution centered at the n -th term of the time series. In our work, we choose

$$K(x) = e^{-x^2}, \quad (6.2)$$

and each G_n will be a Gaussian pdf centered at x_i , with mean $\mu_i = x_i$ and standard deviation

$$\sigma = \frac{2\Delta_x}{\sqrt{N}}, \quad (6.3)$$

where $\Delta_x = X_{max} - X_{min}$ and N is the number of elements in the time series. Hence,

$$G_n(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x_n)^2}{2\sigma^2}}, \quad (6.4)$$

becomes the basis for our Kernel Method.

Note, that in the general Kernel Method, the standard deviation is a parameter to be determined by the user to obtain the best estimate of the pdf. From our experience with laser data, we have found that 6.3 gives us an approximation of the pdf that is comparable to the pdfs we obtain through different methods of the same data.

For all $n \in N$ we will plot each G_n and then sum the plots together to create a curve that has area N . In order to turn this curve into a probability density function, we will scale the curve down by factor N . This will ensure that we have area equal to 1 under our curve.

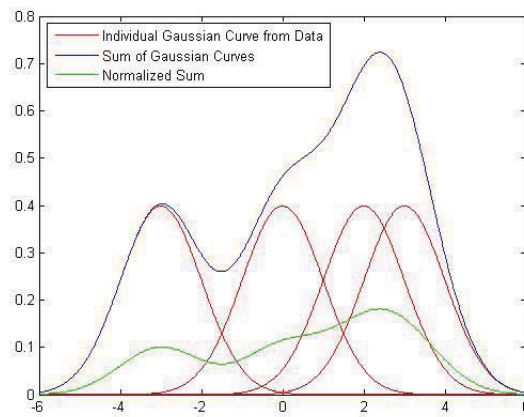


Figure 6.1: Pictorial Representation of the Kernel Method.

```

y=0;
data=[-3 0 2 3];

for n=1:4
    mu=data(n);
    x=linspace(-6,6,1000);
    f=1/sqrt(2*pi)*exp(-(x-mu).^2./2);
    h1=plot(x,f,'r');
    hold on
    y=f+y;
end

h2=plot(x,y,'b');
h3=plot(x,y/4,'g');

legend([h1,h2,h3],'Individual Gaussian Curve from Data',...
        'Sum of Gaussian Curves','Normalized Sum','Location','NorthWest')

```

Figure 6.1 is a visual representation of the Kernel Method, which is the output of the MATLAB code displayed above.

6.1 Kernel Method Implemented on Synthetic Data

Before we implement the Kernel Method on our laser data, we will first implement the method on a synthetic data set in which we know the true probability density function. For our synthetic data, we will use 10,000 points selected from a normal distribution with $\mu = 1$ and $\sigma = 0.1$. We will use the following MATLAB code to run the Kernel Method on our synthetic data. See Figure 6.2 for the output of the following code

```

mu=1; sigma=.1; x=sigma*randn(10000,1)+mu;
ksigma=2*(max(x)-min(x))/(size(x,1)^(1/2));

```

```

x_axis=linspace(min(x)-3*ksigma,max(x)+3*ksigma,20000);
kernel=zeros(size(x_axis));
for n=1:length(x)
    kmu=x(n);
    f=1/ksigma/sqrt(2*pi)*exp(-(x_axis-kmu).^2./2/ksigma^2);
    kernel=f+kernel;
end
plot(x_axis,kernel/length(x),'g');

```

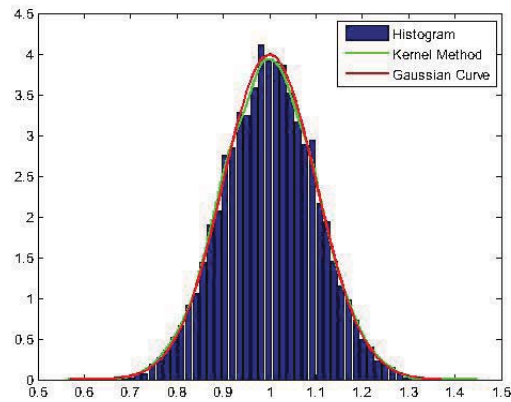


Figure 6.2: Kernel Method Implemented on Synthetic Data.

From a strictly visual analysis, we can see that the Kernel Method approximates the curve remarkably well. In order to obtain a more objective comparison between the two pdfs, we will apply the Kolmogorov-Smirnov (K-S) Test, the Least Squares Test, and compute the Hellinger Distance in Chapter 8.

Chapter 7

Gaussian Mixture Method

The Kernel Method makes intuitive sense when attempting to approximate a pdf. However, we run into computational problems when we deal with data sets that have substantially large numbers of points. For example, creating a probability density function for a laser data set results in the summation of over a million Gaussian curves. In order to decrease the computational issues, we will implement the Gaussian Mixture Method (GMM) (see [11] for an introduction to the method).

The Kernel Method is a specialized case of GMM, in which the number of mixture components is equal to the number of data points in the data set, and the mixture weights are uniform. The generalized GMM can be described as:

$$p_{GMM}(x) = \sum_{n=1}^K \Phi_n p(\mu_n, \sigma_n), \quad (7.1)$$

where K is the number of mixture components, or clusters, Φ_n is the n -th mixing weight, and $p(\mu_n, \sigma_n)$ is the n -th associated Gaussian Curve.

We begin by selecting how many clusters we would like to use in our model. We then calculate μ_n by randomly placing K points along the domain of the data points. We invoke a machine learning technique that iteratively reassigns the positions of μ_n according to the formula:

$$\mu_{n_i} = \sum_{j=1}^N \frac{x_j}{N}, \quad (7.2)$$

where x_j are the data points in which $|x_j - \mu_{n_{i-1}}|$ is a minimum $\forall n \in K$. While the iterative sequence will always stabilize to a solution, the solution is not unique. The solution depends upon the initial distribution of the μ_n . Once we have the "centers" for the Gaussian components, we assign a standard deviation to each cluster by:

$$\sigma_k = \frac{\max\{x_j : j \in N\} - \min\{x_j : j \in N\}}{2}. \quad (7.3)$$

The mixing weights, Φ_n , is the ratio of the number of data points closest to μ_n to the total number of data points in the data set. The weights are chosen so that $\sum_{n=1}^K \Phi_n = 1$ for all possible Φ_n . This ensures that $\int_{-\infty}^{\infty} p_{GMM}(x) dx = 1$.

We then implement the GMM on the synthetic data pulled from a normal distribution (see Figure 7.1). See Appendix A for MATLAB code that was used to generate the figure.

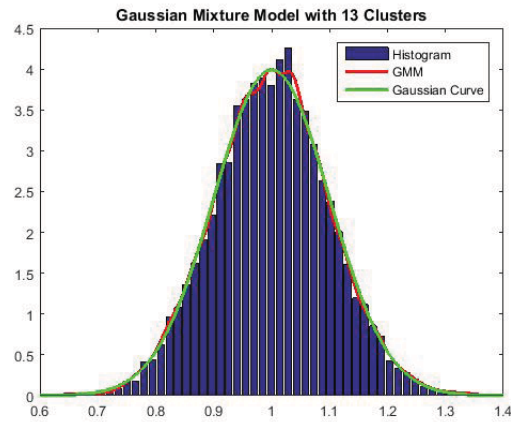


Figure 7.1: Gaussian Mixture Method Implemented on Synthetic Data with 13 Clusters.

Again, from a strictly visual analysis, we can see that GMM approximates the histogram curve remarkably well. In order to obtain a quantitative comparison between the two pdfs, we will apply the Kolmogorov-Smirnov (K-S) Test, the Least Squares Test, and compute the Hellinger Distance as described in Chapter 8.

Chapter 8

Description of Comparison Methods and Implementation of PDF Approximation Techniques on Synthetic Data

In this section we will describe and implement the three pdf approximation techniques on synthetic data and compare the results using the K-S Test, Least Squares Test, and by computing the Hellinger distance.

8.1 Kolmogorov-Smirnov (K-S) Test

The Kolmogorov-Smirnov Test yields a scalar value known as the Kolmogorov-Smirnov Statistic. For a given data set, we compute the Empirical Cumulative Distribution Function (ECDF) and test it against a proposed cumulative distribution function (cdf). The lower a K-S Statistic, the better the fit for the proposed cdf. The K-S Statistic, D_n can be computed by the following equation:

$$D_n = \sup_x |F_n(x) - F(x)|, \quad (8.1)$$

and

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I(x_i \leq x), \quad (8.2)$$

where $I(x_i \leq x)$ is the Indicator function, and $F(x)$ is the proposed cdf. The function, $F_n(x)$ is the ECDF from our data. Since the ECDF is a continuous function on a closed and bounded interval,

$$D_n = \max_x |F_n(x) - F(x)|. \quad (8.3)$$

The K-S Statistic is solely dependent upon the "worst" point in the ECDF. See Figure 8.1 for visual representation of K-S Test.

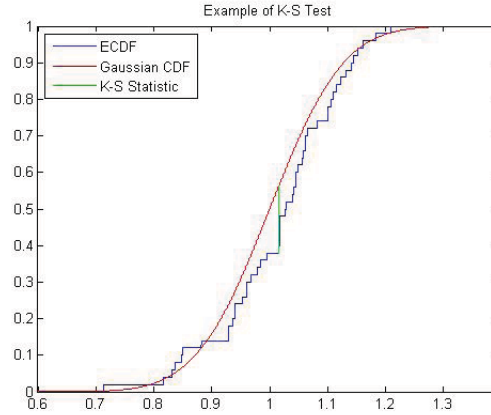


Figure 8.1: K-S Test Implemented on 50 Synthetic Data Points from Gaussian Distribution.

8.2 Least Squares Test

The Least Squares Test yields a scalar value, L_n that can compare the quality of the fit for a proposed cdf:

$$L_n = \sum_{i=1}^n |F_n(x) - F(x)|^2, \quad (8.4)$$

where $F_n(x)$ and $F(x)$ were defined in the previous section. Note that the Least Squares Test takes into account all points in the cdf.

8.3 Hellinger Distance

The Hellinger Distance yields a scalar value, H that can quantify the distance between two probability density functions. For two pdfs, $p_1(x)$ and $p_2(x)$, H is defined as:

$$H = 1 - \int_{-\infty}^{\infty} \sqrt{p_1(x)p_2(x)} dx. \quad (8.5)$$

Note that $0 \leq H \leq 1$ where $H = 0 \Leftrightarrow p_1(x)$ and $p_2(x)$ are equal, and $H = 1 \Leftrightarrow p_1(x)$ and $p_2(x)$ are disjoint. The Hellinger Distance takes into account is computationally inexpensive, as it does not require converting the pdf into a cdf. In addition, the Hellinger Distance accounts for the entire distribution, as opposed to comparing via a single-point supremum.

8.4 Application of K-S Test and Least Squares Test on Kernel Method with Synthetic Data from Gaussian Distribution

We will begin by defining ECDF within MATLAB. We will use the following function in MATLAB to generate the ECDF.

```
function [ y ] = ecdf_mod( x_axis, data )
y=ones(size(x_axis));
for n=1:size(x_axis,2);
    y(n)=size(find(data<=x_axis(n)),1);
end
y=y/size(data,1);
end
```

If we look at the visual comparison between the ECDF, the Kernel CDF, and the Gaussian CDF, there is almost negligible difference between them (Figure 8.2). If we zoom in, we can see the differences more clearly (Figure 8.3).

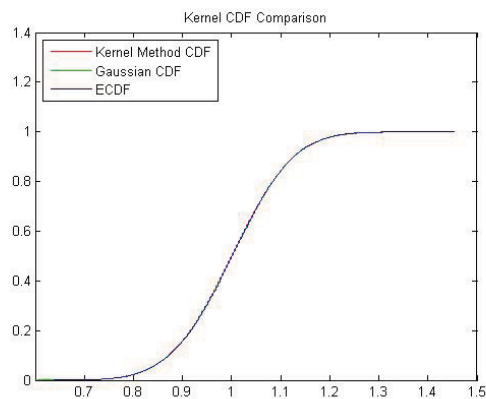


Figure 8.2: Comparison of Kernel CDF, ECDF, and Gaussian CDF.

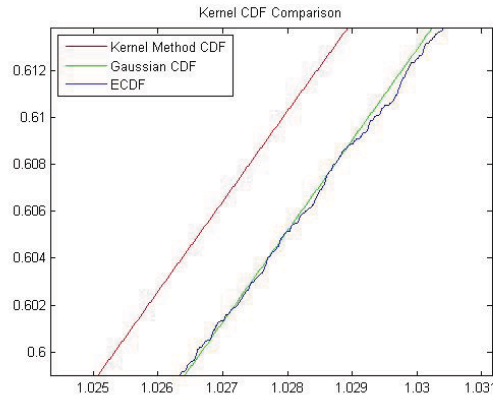


Figure 8.3: Zoomed in Comparison of Kernel CDF, ECDF, and Gaussian CDF.

We will run the following code to create a table to compute K-S and Least Squares statistics in order to compare the Kernel Method to the Gaussian CDF:

```
sigma=.1; mu=1; x=sigma*randn(20000,1)+mu; x=sort(x);
ksigma=2*(max(x)-min(x))/(size(x,1)^(1/2));
x_axis=linspace(min(x)-3*ksigma,max(x)+3*ksigma,50000);
kernel=zeros(size(x_axis));
for n=1:length(x)
    muk=x(n);
    f=1/ksigma/sqrt(2*pi)*exp(-(x_axis-muk).^2./2/ksigma^2);
    kernel=f+kernel;
end
yy=1/(sigma*sqrt(2*pi))*exp(-.5*((x_axis-mu)/sigma).^2);
kernel_cdf=cumsum(kernel/sum(kernel));
gauss_cdf=cumsum(yy/sum(yy));
y=ecdf_mod(x_axis,x);
ls_stat_kernel=sum((y-kernel1).^2);
ls_stat_gauss=sum((y-yy1).^2);
ks_stat_kernel=max(abs(y-kernel1));
ks_stat_gauss=max(abs(y-yy1));
```

Trial	Gaussian K-S Test	Kernel K-S Test	Gaussian L.S. Test	Kernel L.S. Test
1	0.008730316	0.003514486	0.270325412	0.043871312
2	0.004772498	0.003290708	0.082172717	0.038146703
3	0.004390352	0.003361794	0.064362886	0.03988743
4	0.006107503	0.002899473	0.206844817	0.033283234
5	0.007465381	0.003528832	0.320147363	0.04676351
6	0.005121885	0.004373087	0.115170686	0.043326217
7	0.005520343	0.003548023	0.110128309	0.038915194
8	0.004898304	0.003086733	0.045160592	0.044495731
9	0.009677349	0.003321411	0.498252486	0.044058385
10	0.006414521	0.003207911	0.127893274	0.043694205
11	0.005622353	0.003422341	0.133494516	0.045780496
12	0.005544909	0.003300643	0.066306315	0.047292619
13	0.005325703	0.003664077	0.125827956	0.042060459
14	0.004441213	0.003270251	0.082478346	0.039470854
15	0.004667689	0.003082527	0.077155772	0.039204603
16	0.005212049	0.00298129	0.108168645	0.040054665
17	0.004781662	0.002712938	0.147263091	0.034870995
18	0.004918177	0.003511495	0.075119744	0.042089731
19	0.005429505	0.002915812	0.125410971	0.03479343
20	0.004425386	0.003148757	0.075171545	0.042178424
Average:	0.005673355	0.003307129	0.142842772	0.04121191

From the table, we can see that the Kernel Method Cumulative Distribution Function represents the solely data-driven ECDF better than the Gaussian CDF. The K-S Test and Least Squares Test both confirm that the Kernel Method is a better representation of the empirical cumulative distribution function for the data than the distribution from which the data was selected.

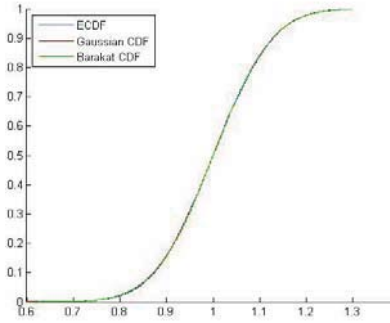


Figure 8.4: Barakat CDF Implemented on Synthetic Data.

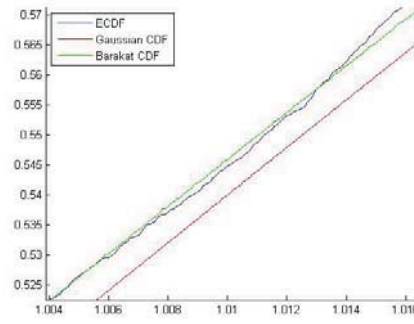


Figure 8.5: Zoomed in Comparison of Barakat CDF, ECDF, and Gaussian CDF.

Trial	Gaussian K-S Test	Barakat K-S Test	Gaussian L.S. Test	Barakat L.S. Test
1	0.003741717	0.004782784	0.010816655	0.013854076
2	0.009100771	0.004365288	0.074162032	0.017477223
3	0.00587217	0.006620209	0.029062625	0.026011631
4	0.005310268	0.004604807	0.023837694	0.013546505
5	0.009869642	0.006506674	0.070711684	0.020875552
6	0.012072861	0.007640935	0.140883954	0.033825081
7	0.00988999	0.005497651	0.097202076	0.028887014
8	0.014180303	0.009703637	0.227673835	0.081647998
9	0.009411214	0.005504155	0.039722674	0.020847152
10	0.008335981	0.00445205	0.060843847	0.016406816
11	0.008769523	0.005991756	0.082320883	0.021333036
12	0.0084606	0.007036387	0.048399194	0.029834825
13	0.006689778	0.005098268	0.047057498	0.019582776
14	0.007631687	0.004748362	0.060000535	0.017543446
15	0.009348316	0.003951488	0.114133994	0.014394307
16	0.004386149	0.004872546	0.012045822	0.017690785
17	0.006988732	0.004788965	0.031973471	0.017049715
18	0.007004244	0.005483672	0.055884698	0.01862186
19	0.010853364	0.004452651	0.117317925	0.016806959
20	0.008025353	0.007026941	0.040481445	0.040422328
Average:	0.008297133	0.005656461	0.069226627	0.024332954

8.5 Hellinger Distance on Synthetic Data

In this section we will implement all three approximation techniques and compare the similarities between the pdfs using the Hellinger Distance. The synthetic data simulation creates 50 data sets of synthetic data pulled from the same distribution. We implement the three approximation techniques and compute the Hellinger Distance using the Gaussian distribution from which the data was pulled as the baseline pdf. The results are shown below in depth: (See Appendix for supporting MATLAB code)

Barakat	GMM	Kernel
1.2463358E-04	4.4506809E-04	3.4055791E-03

Figure 8.6: Table of Average Hellinger Distance after 50 Simulations.

Barakat	GMM	Kernel
0.000317342	0.000864476	0.003930431
0.000159443	0.00046888	0.003617357
0.000108236	0.000413539	0.003422314
4.25E-05	0.000256833	0.003230477
-0.000120685	3.07E-05	0.003145302
-0.000207601	-0.000121275	0.003218858
2.83E-05	0.000255226	0.003481929
0.000265932	0.000382234	0.003052538
6.90E-05	0.000372775	0.003276085
6.53E-05	0.000209794	0.00368244
0.000167666	0.000475029	0.003682658
4.06E-05	0.000408525	0.003621717
0.000145774	0.000374245	0.00335986
0.000192333	0.000578214	0.003858138
9.22E-05	0.000367495	0.003547981
0.000191431	0.000491128	0.00372318
0.000125497	0.000523188	0.003593297
0.000135682	0.000507496	0.003644294
0.000133122	0.000313051	0.003607424
0.000176125	0.000598791	0.003342113
-5.11E-06	0.000344593	0.003050142
0.000148496	0.000299137	0.003502376
9.00E-05	0.000408509	0.003485367
0.000363634	0.000586945	0.00380843
0.000201672	0.000730999	0.003403014
0.000240903	0.000504313	0.003723927
0.000125479	0.000565985	0.003338416
0.000148184	0.000615012	0.002900943
1.07E-05	0.000416201	0.003445834
0.000334906	0.000623258	0.003617947
0.000256256	0.000751865	0.003563757
9.26E-05	0.000450765	0.003748495
-7.90E-05	0.000135497	0.00282983
0.000121118	0.000478484	0.003135431
8.91E-05	0.000194767	0.003469072
0.00013715	0.000568589	0.003253515
2.27E-05	0.0002712	0.002931291
5.64E-05	0.000483626	0.002998006
7.51E-05	0.000410133	0.003232675
0.000258302	0.000738333	0.00370976
0.00030403	0.00085435	0.003713396
0.000142529	0.000551272	0.003317579
0.00021561	0.000594996	0.004030471
0.000128206	0.000512315	0.003055812
-6.40E-06	0.000342636	0.002816854
0.000154352	0.000429945	0.003961581
0.000158084	0.000500878	0.003103454
0.000165815	0.000434945	0.003549039
2.26E-05	0.000317952	0.002799158
0.000130099	0.000365608	0.002744995

Figure 8.7: Table of Full Results of Hellinger Distances after 50 Simulations.

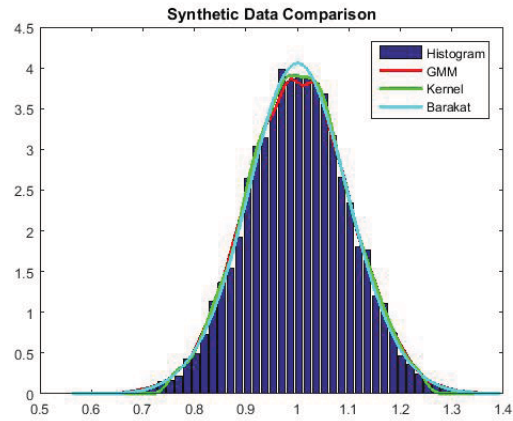


Figure 8.8: Visual Comparison of the Three Approximation Techniques Implemented on Synthetic Data.

We can see from the Hellinger Distance data in the table above that the Barakat Method was best at approximating the Gaussian Curve. Although the Kernel Method visually represented the data better, and represented the ECDF better, the Barakat Method was optimal for producing a pdf most similar to the pdf of the distribution from which the data was pulled.

Chapter 9

Implementation and Comparison of PDF Approximation Techniques on Laser Data

In this chapter we will use the methods described in the previous section to evaluate laser data sets.

9.1 Nighttime Laser Data

The first laser data set we will be using is the HeNe laser data set taken in a nighttime, outdoor environment. The impact of the atmosphere is less during the nighttime because the light from the sun is not interacting with the photons from the laser light. The sensor is stationary, 375 meters away, and recording intensities at a rate of 10 kHz. The wavelength of the laser light is 633 nm.

We will implement the three approximation techniques on the nighttime data sets, and use the Hellinger Distance to determine which method is best for approximating the pdf of the laser data. In addition, we will implement a visual test for convergence on the Barakat Method for real laser data. In regards to the Barakat Method, the β value is often too large to implement the method directly. Therefore, we will remove the mean from the data, and implement the method to decrease the β value. Once the method has completed, we will insert the mean back into the domain.

We will begin by analyzing two data sets taken in the nighttime environment. Figures 9.1 and 9.3, are the time series plot of the two data sets we are analyzing. Figures 9.2 and 9.4 depict comparisons of the three approximation techniques discussed in the previous chapters. We can see that the GMM and Kernel Method approximate the curve of the histogram remarkably well. However, the Barakat method does not converge to an accurate approximation to the pdf for the data. In Figure 9.5 we can see the progression of the Barakat pdf through increased terms (terms 5-15 are displayed). Note that there is not much visual difference between the pdf with 5 terms compared to the pdf with 15 terms.

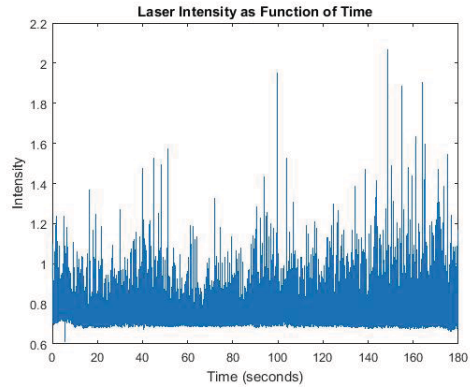


Figure 9.1: Time Series Function of Laser Intensity During Nighttime 1.

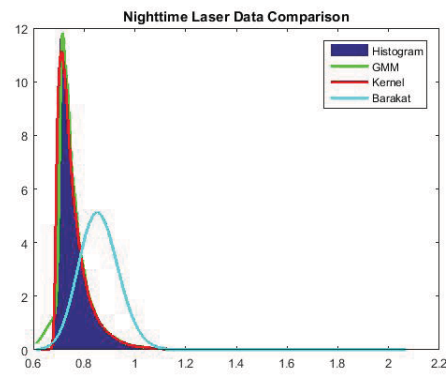


Figure 9.2: Visual Comparison of the Three Approximation Techniques Implemented on Nighttime Data 1.

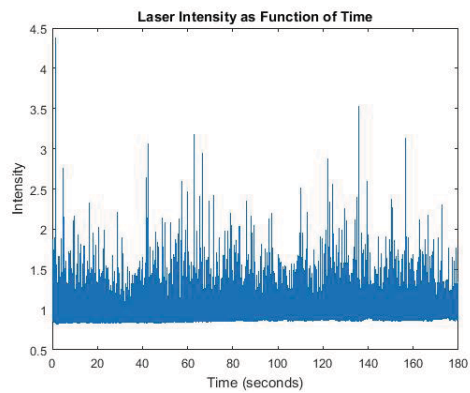


Figure 9.3: Time Series Function of Laser Intensity During Nighttime 2.

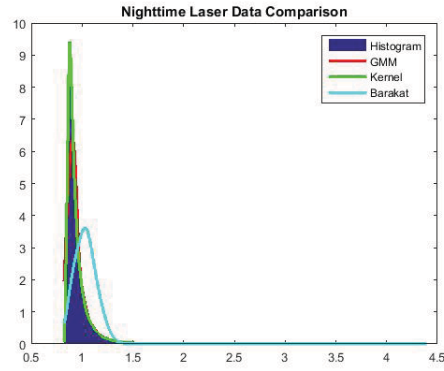


Figure 9.4: Visual Comparison of the Three Approximation Techniques Implemented on Nighttime Data 2.

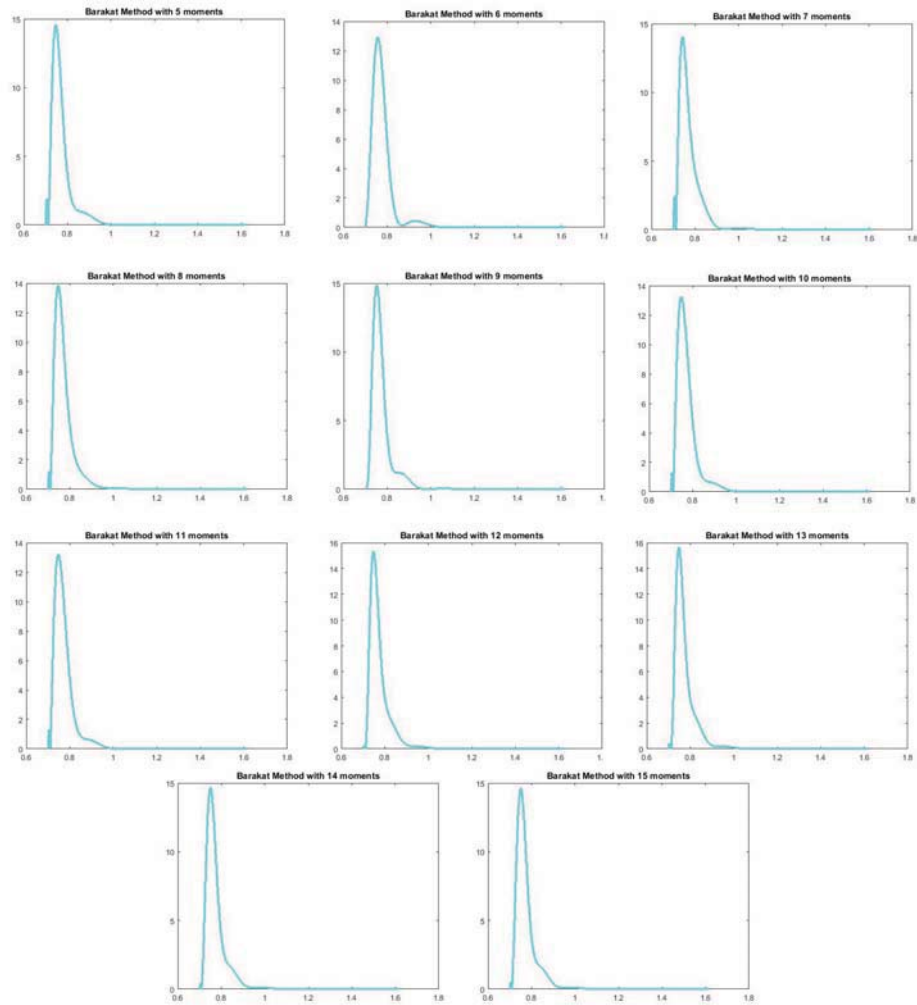


Figure 9.5: Visual Representation for Convergence of Barakat Method for Nighttime Laser Data.

9.2 Daytime Laser Data

The final laser data sets we will be using is the HeNe laser data set taken in a daytime, outdoor environment. The impact of the atmosphere is greater during the daytime because the light from the sun is now interacting with the photons from the laser light. The light waves from the sun dominate the light waves of the laser light, creating less fluctuations in intensity at the target. The sensor is stationary, 375 meters away, and recording intensities at a rate of 10 kHz. The wavelength of the laser light is 633 nm. Figure 9.7 and Figure 9.9 represent visual comparisons of the three pdf approximation techniques executed on daytime laser data.

Paralleled with the previous section, we will now analyze two data sets taken in the daytime environment. Figures 9.6 and 9.8, are the time series plot of the two data sets we are analyzing. Figures 9.7 and 9.9 depict comparisons of the three approximation techniques discussed in the previous chapters. Again, we can see that the GMM and Kernel Method approximate the curve of the histogram remarkably well in both figures. The Barakat method also converges to an accurate pdf approximation in Figure 9.7. However, the Barakat method, in Figure 9.9 does not converge to any pdf (19 terms were used). In Figure 9.10 we can see the progression of the Barakat pdf through increased terms (terms 5-15 are displayed). Note that the pdf does not appear to converge through all 15 terms.

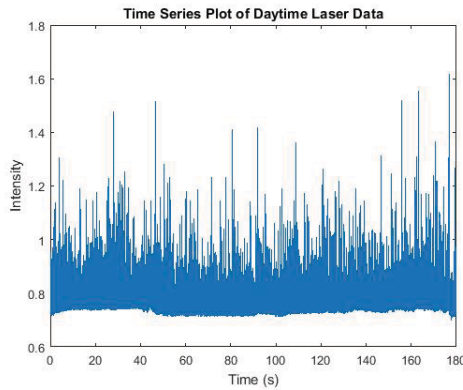


Figure 9.6: Time Series Function of Laser Intensity During Daytime 1.

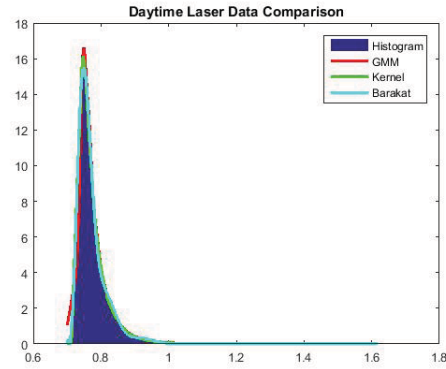


Figure 9.7: Visual Comparison of the Three Approximation Techniques Implemented on Daytime Data 1.

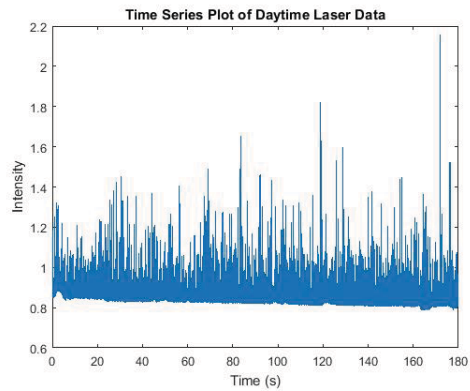


Figure 9.8: Time Series Function of Laser Intensity During Daytime 2.

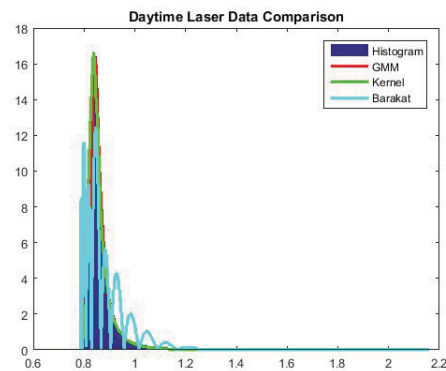


Figure 9.9: Visual Comparison of the Three Approximation Techniques Implemented on Daytime Data 2.

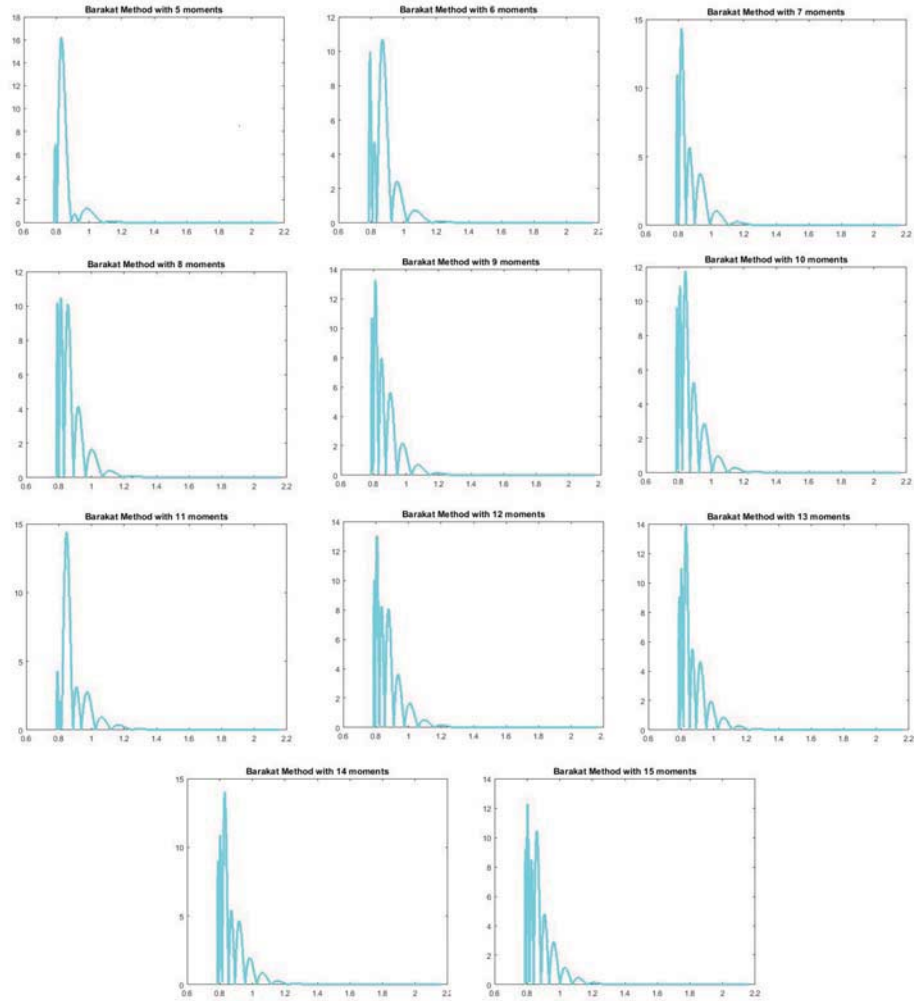


Figure 9.10: Visual Representation for Convergence of Barakat Method for Daytime Laser Data.

Chapter 10

Solution to Paraxial Wave Equation with Stochastic Refraction Coefficient

Computing a solution to the paraxial wave equation, defined in Chapter 2, is non-trivial and of great importance when attempting to model the behavior of the laser beam at a target once propagated through a turbulent medium. From Chapter 2,

$$V(x, y, z) = \frac{A}{\sqrt{\Lambda_0^2 + \Theta_0^2}} \exp\left(-\frac{r^2}{W(z)^2} + i\left(\phi + \frac{kr^2}{2F(z)}\right)\right), \quad (10.1)$$

where Λ_0 , called the refraction parameter, and Θ_0 , the diffraction parameter, are

$$\Lambda_0 = 1 - \frac{z}{F_0}, \quad \Lambda_0 = \frac{2z}{kW_0^2} \quad (10.2)$$

and W and F , which are the spot-size radius of the beam and its phase radius of curvature at any z , are

$$W(z) = W_0 \sqrt{\Lambda_0^2 + \Theta_0^2}, \quad \phi = \tan^{-1} \frac{\Lambda_0}{\Theta_0} \quad (10.3)$$

$$F(z) = F_0 \frac{(\Theta_0^2 + \Lambda_0^2)(\theta_0 - 1)}{\Theta_0^2 + \Lambda_0^2 - \Theta_0} \quad (10.4)$$

Note that 10.1 results in a plot of contours of the intensity of the laser beam. Since our data is obtained from a sensor that records the time series average intensities from a specified area, we will average the values of V to obtain a single time-series data point. We will vary our refraction coefficient k as such:

$$k = k_0 + \gamma(2 + \eta_\alpha, 2 + \eta_\beta), \quad (10.5)$$

where k_0 is fixed, $\gamma(2 + \eta_\alpha, 2 + \eta_\beta)$ is a random variable pulled from a gamma distribution with parameters $\alpha = 2 + \eta_\alpha$ and $\beta = 2 + \eta_\beta$. From experimenting with probability density functions of laser light intensities in the maritime domain, an underlying gamma distribution was selected for the stochastic parameter k . η_α and η_β are uniform random variables to increase the variance in the distribution, which accounts for the extra noise we see in the data, and provides an addition parameter to vary when accounting for atmospheric conditions. We will run a time series simulation that uses 10.1 and 10.5 to generate simulated time series data (See Figure 10.1). We will then

implement the three pdf approximation techniques to generate a pdf of the simulated data (See Figure 10.2).

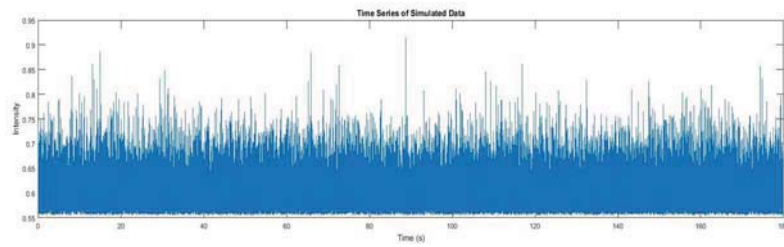


Figure 10.1: Simulated Time Series Laser Data.

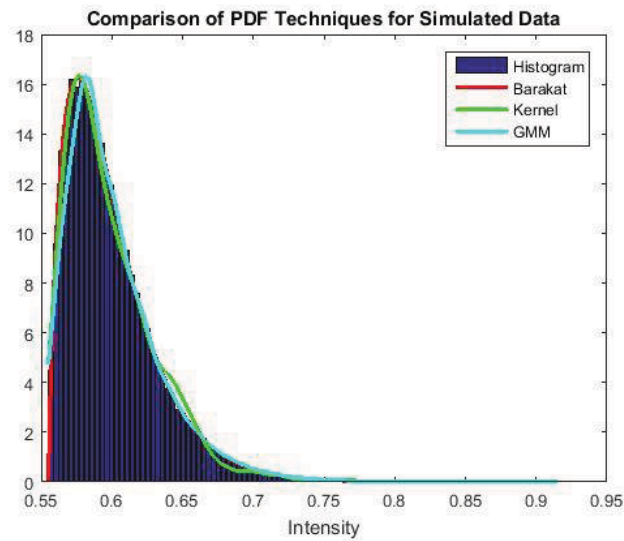


Figure 10.2: PDF of Simulated Time Series Laser Data.

Chapter 11

Conclusion

11.1 Barakat

In conclusion, we can see that while the Barakat Method is capable of modeling the synthetic data remarkably well, it cannot always account for the noise in the real data. In addition, the beta values from the Barakat Method that result from the real data sets are often too large to support numerical approximations computationally. However, this can be offset by removing the mean from the data, and adding it back once the method has been implemented. Regardless, we have seen instances in which the Barakat Method still does not converge to a smooth pdf.

11.2 Kernel

The Kernel Method is capable of representing the empirical data well, but does not yield as much information about the underlying distribution from which the data is pulled (as discovered from the synthetic data simulation). In addition, the Kernel Method is computationally very strenuous. The computational power can be minimized by not using every data point in the data set, but rather generated from a user-defined number of data points randomly chosen from the data set. In addition, the Kernel Method is also subjected to the user's input for standard deviation. While we have found that 6.3 is a good rule of thumb for the data sets, σ can be altered by the user to make minor adjustments to fit the approximate pdf to the histogram curve.

11.3 Gaussian Mixture Method

Ultimately, the Gaussian Mixture Method represented the data set well, and was not as computationally strenuous. However, the resulting pdf is not a unique solution, because it depends on the initial location of the center points of the clusters. In addition, it requires the user to input the number of clusters beforehand. Furthermore, we can see oscillations at the peak of the pdf we know from the histograms that these oscillations should not be present. Ultimately, there are numerous variations to computing a pdf via GMM (see reference [11] for examples of different techniques of GMM).

11.4 Summary

In essence, if the data set is unimodal, and the beta value is sufficiently small, the Barakat Method approximates the underlying distribution function of the data set the best. The method is efficient and most similar to the distribution from which the data was drawn. However, if the data is complex, we have seen that the Barakat method can sometimes break down and not converge. Even if the Barakat method does converge, it could still not represent the data well in some instances. In these cases, the Gaussian Mixture Model is the best solution to approximating the pdf, when evaluating objectively through the K-S, Least Squares, and Hellinger Distance tests. It is less computationally strenuous than the Kernel Method, and represents the underlying distribution function well, by the Hellinger Distance. The Kernel Method exhibits less oscillation, and results in a smoother pdf, but does not represent the underlying distribution as well as the Barakat and GMM techniques.

Bibliography

- [1] Andrews, Larry C., and Ronald L. Phillips. *Laser Beam Propagation through Random Media*. Bellingham, WA: SPIE Optical Engineering, 1998.
- [2] Barakat, Richard. *First-Order Intensity and Log-Intensity Probability Density Functions of Light Scattered By the Turbulent Atmosphere in Terms of Lower-Order Moments*. Aiken Computation Lab, Cambridge, Massachusetts, 1998.
- [3] Barakat, Richard. *Second-Order Statistics of Integrated Intensities and Detected Photons, The Exact Analysis*. Journal of Modern Optics, 43.6. 1237-1252. 1996.
- [4] Grayshan, K.J., *Waves in Complex Media: A Marine Atmospheric Spectrum for Laser Propagation*. Waves in Random and Complex Media. Volume 18, Issue 1, 2008.
- [5] Higham, Desmond J. "An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations." SIAM Review 43.3 (2001): 525.
- [6] Korotkova, Olga, Svetlana Avramov-Zamurovic, Reza Malek-Madani, and Charles Nelson. *Probability Density Function of the Intensity of a Laser Beam Propagating in the Maritime Environment*. Optics Express, Vol. 19, Issue 21, pp. 20322-31, 2011.
- [7] Nelson, Charles, Olga Korotkova, Svetlana Avramov-Zamurovic, Raymond Sova, Frederic Davidson, and Reza Malek-Madani. *Probability Density Function of Power-in-Bucket and Power-in-Fiber for an Infrared Laser Beam Propagating in the Maritime Environment*. Applied Optics, Vol. 52, Issue 31, pp. 7449-61, 2013.
- [8] Nielson, Philip E. *Effects of Directed Energy Weapons*. Directed Energy Professional Society: Albuquerque, NM, 2009.
- [9] Silverman, B. W., *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.
- [10] Toselli, Italo, Brij Agrawal, and Sergio Restaino. *Gaussian Beam Propagation in Maritime Atmospheric Turbulence: Long Term Beam Spread and Beam Wander Analysis*. SPIE Digital Library. 12 August 2010.
- [11] Wang, Eric, Svetlana Avramov-Zamurovic, RJ Watkins, and Reza Malek-Madani. *PDF Estimation of Laser Light Scintillation via Bayesian Mixtures*. Journal of the Optical Society of America, 2014.

Appendix

Barakat Function

```

1 function [ W, barakat_pdf, graph ] = Barakat_Function( laser ,
    moments, logic )
2
3 laser=sort(laser);
4 min_laser=min(laser);
5 laser=laser-min_laser;
6 I=laser/mean(laser);
7
8 beta=mean(laser)^2/(mean(laser.^2)-mean(laser)^2);
9 NN=moments;
10
11 WW=0;
12
13 for N=3:NN
14     term=Barakat_W_n2(beta , I , N).*Laguerre_Beta(beta , beta*I , N)
15     ;
16     WW=WW+term;
17 end
18 W=Barakat_W_g2(beta , I ).*(1+WW);
19
20 barakat_pdf=abs(W)/trapz(laser ,abs(W));
21 laser=laser+min_laser;
22 if logic ==1
23     graph=plot(laser ,barakat_pdf , 'c' , 'LineWidth' ,2);
24     title([ 'Barakat Method with ' ,num2str(NN) , ' moments' ])
25 else
26     graph=NaN;
27 end
28 end

```

ECDF Computation Function

```

1 function [ y ] = ecdf_mod( x_axis , data )
2 %UNTITLED6 Summary of this function goes here
3 %   Detailed explanation goes here
4
5 y=ones( size( x_axis ) );
6
7 for n=1:size( x_axis ,2);
8     y(n)=size( find( data <= x_axis(n) ) ,1);
9 end
10
11 y=y / size( data ,1);
12
13 end

1 function [ y ] = ECDF_data( laser )
2 %UNTITLED11 Summary of this function goes here
3 %   Detailed explanation goes here
4
5 y=1:size( laser ,1);
6 y=y / size( laser ,1);
7
8
9 end

```

Final PDF Comparison

```

1  for n=1:1
2      clear all
3      clc
4      data=.1*randn(10000,1)+1;
5      data=sort(data);
6      gauss=1/sqrt(2*pi)/.1*exp(-(data-1).^2/(2*.1^2));
7      %Compute GMM
8      [ xg, yg, graph ] = Gaussian_Mixture( data, 13, 10 );
9      hold on
10     %Compute Kernel
11     [xk, yk, graph]=Kernel_Method_function(data);
12
13     %Compute Barakat
14     [ W, barakat_pdf, graph ] = Barakat_Function( data, 5 );
15     title('Synthetic Data Comparison')
16     legend('Histogram','GMM','Kernel','Barakat')
17
18
19
20 end

```

Gaussian Mixture Method Function

```

1 function [ x, y, graph ] = Gaussian_Mixture( laser , k, N, logic )
2     %laser is the data
3     %k is number of clusters
4     %N is number of iterations
5
6     %load( ' LaserData\HeNe_50dB_gain_375meters_1024_USNA_4 ' )
7     %laser=ConvertedData.Data.MeasuredData(1,4).Data;
8
9     % mu=1; sigma=.1;
10    % laser=sigma*randn(100000,1)+mu;
11    laser=sort(laser);
12
13    if logic==1
14        [ hist_y , hist_x ]=hist(laser,200);
15        hist_y=hist_y / trapz(hist_x, hist_y);
16        bar(hist_x, hist_y)
17        hold on
18    end
19
20
21    deltax=2*min([max(laser)-mean(laser), mean(laser)-min(laser)]);
22
23    centers=deltax*rand(k,1)+mean(laser)-deltax;
24    centers=linspace(min(laser), max(laser), k)';
25    centers=sort(centers);
26
27    m=zeros(k,N);
28    for iter=1:N
29        a=1;
30        ind=zeros(k,1);
31        sigma=zeros(k,1);
32        m(:,iter)=centers;
33        for n=1:k-1
34            upp=(centers(n+1)+centers(n))/2;
35            b=find(laser>upp,1);
36            centers(n)=mean(laser(a:b));
37            ind(n)=b-a;
38            sigma(n)=(laser(b)-laser(a))/2;
39            a=b;
40        end
41
42        centers(k)=mean(laser(b:end));

```

```

43     sigma(k)=(max(laser)-laser(b))/2;
44     ind(k)=size(laser,1)-b;
45     centers=centers(~isnan(centers));
46     k=size(centers,1);
47
48 end
49
50 weights=ind/(size(laser,1)-1);
51
52 for n=1:k
53     %plot steady state center points of clusters
54     %plot(m(n,:))
55     %hold on
56 end
57
58 %x=linspace(min(laser)-4*sigma(1),max(laser)+4*sigma(k),10000);
59 x=sort(laser);
60
61 y=zeros(size(x));
62 for n=1:k
63     if sigma(n)~=0
64         y=y+weights(n)/sigma(n)/sqrt(2*pi)*exp(-((x-centers(n))
65             .^2)/(2*sigma(n)^2));
66     end
67 end
68 %figure
69 if logic == 1
70     graph=plot(x,y,'r','LineWidth',2);
71     title(['Gaussian Mixture Model with ',num2str(k),' Clusters'])
72 else
73     graph=NaN;
74 end
75
76
77 end

```

Hellinger Distance

```
1 function [ h ] = Hellinger( x, pdf1, pdf2 )
2 %UNTITLED Summary of this function goes here
3 %   Detailed explanation goes here
4
5 fg=sqrt(pdf1.*pdf2);
6
7 h=1-trapz(x,fg);
8
9 end
```

Kernel Method Function

```

1 function [ x, y, graph ] = Kernel_Method_function( laser , logic )
2 % laser = data
3
4     sigma=2*(max(laser)-min(laser))/sqrt(length(laser));
5     %x=linspace(min(laser)-3*sigma,max(laser)+3*sigma,10000);
6     x=sort(laser);
7     y=zeros(size(x));
8     h=waitbar(0,'Starting Kernel Method');
9     for n=1:length(laser)/100-1
10         waitbar(n/(length(laser)/100-1),h,'Loading Kernel Method'
11             )
12         mu=laser(100*n);
13         f=1/sigma/sqrt(2*pi)*exp(-(x-mu).^2./2/sigma^2);
14         y=f+y;
15     end
16     close(h)
17
18     y=y/(length(laser)/100-1);
19     if logic ==1
20         graph=plot(x,y,'g','LineWidth',2);
21         title('Kernel Method')
22     else
23         graph=NaN;
24     end
25
26 end

```

PDF to CDF Converting Function

```
1 function [ cdf ] = pdf_to_cdf( x, pdf )
2 %UNTITLED13 Summary of this function goes here
3 % Detailed explanation goes here
4 cdf=zeros(size(pdf));
5 h=waitbar(0,'Converting PDF to CDF');
6 for n=2:size(x,1)
7     waitbar(n/(size(x,1)),h,'Converting PDF to CDF')
8     cdf(n)=trapz(x(1:n),pdf(1:n));
9 end
10 close(h)
11
12 cdf(end)=1;
```