K-distribution Fading Models for Bayesian Estimation of an Underwater Acoustic Channel

by

Alison Beth Laferriere

B.S., B.A., University of Rhode Island (2008)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
and the
WOODS HOLE OCEANOGRAPHIC INSTITUTION

February 2011

Author ....
Department of Electrical Engineering and Computer Science
January 28, 2011

Certified and accepted by ....
James C. Preisig
Associate Scientist, Woods Hole Oceanographic Institution
Chairman, Joint Committee for Applied Ocean Science and Engineering
Thesis Supervisor

Accepted by ....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

K-distribution Fading Models for Bayesian Estimation of an Underwater Acoustic Channel

by

Alison Beth Laferriere

Submitted to the Department of Electrical Engineering and Computer Science
on January 28, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Current underwater acoustic channel estimation techniques generally apply linear MMSE estimation. This approach is optimal in a mean square error sense under the assumption that the impulse response fluctuations are well characterized by Gaussian statistics, leading to a Rayleigh distributed envelope. However, the envelope statistics of the underwater acoustic communication channel are often better modeled by the K-distribution. In this thesis, by presenting and analyzing field data to support this claim, I demonstrate the need to investigate channel estimation algorithms that exploit K-distributed fading statistics. The impact that environmental conditions and system parameters have on the resulting distribution are analyzed. In doing so, the shape parameter of the K-distribution is found to be correlated with the source-to-receiver distance, bandwidth, and wave height. Next, simulations of the scattering behavior are carried out in order to gain insight into the physical mechanism that cause these statistics to arise. Finally, MAP and MMSE based algorithms are derived assuming K-distributed fading models. The implementation of these estimation algorithms on simulated data demonstrates an improvement in performance over linear MMSE estimation.

Thesis Supervisor: James C. Preisig
Title: Associate Scientist, Woods Hole Oceanographic Institution
Chairman, Joint Committee for Applied Ocean Science and Engineering

Acknowledgments

First and foremost, I would like to thank my thesis advisor, Dr. James C. Preisig from the Woods Hole Oceanographic Institution for his guidance throughout this thesis project. Jim not only provided me with direction and leadership, but was always enthusiastic, encouraging, and incredibly understanding of the hurdles one encounters during graduate studies at MIT. Also of great help were the past and present members of our research group, whose ideas and discussions during group meetings helped drive parts of this thesis.

I would also like to thank my classmates for their assistance and companionship throughout all of the coursework I have completed, and my TAs, who were essential to learning the necessary material to complete this thesis. Last but not least, I owe a great deal of gratitude to my friends in the MIT/WHOI Joint Program, all of my friends outside of MIT, and my family. Without their love and support, I am not sure I could have made it this far.

Contents

List of Figures

List of Tables

Nomenclature

α	Scale parameter in the K-distribution
$\Gamma(\cdot)$	Gamma function
Γ_z	Diagonal matrix containing the elements of $\mathbf{z}$
Λ_w	Half the covariance matrix of the noise $\mathbf{w}$
$\mathbf{D}$	Matrix of transmitted data values
$\mathbf{d}_i$	i^{th} row vector of D matrix
$\mathbf{h}, h$	Channel taps in the linear channel model
$\mathbf{N}$	Diagonal matrix containing the elements of $\mathbf{n}$
$\mathbf{n}, n$	Complex Gaussian random variable in component representation of K-distributed fading
$\mathbf{R}$	Diagonal matrix containing the elements of $\mathbf{r}$
$\mathbf{r}, r$	$\sqrt{\mathbf{z}}, \sqrt{z}$
$\mathbf{w}, w$	Complex Gaussian noise in the linear channel model
$\mathbf{y}, y$	Observed data in the linear channel model
μ	Parameter in the Rician distribution
∇, ∇_n	Gradient, gradient with respect to n
∇^2, ∇_n^2	Hessian, Hessian with respect to n
ν	Shape parameter in the K-distribution

σ	Parameter in the Rician and Rayleigh distributions
σ_w^2	Half the variance of the complex-valued noise w, variance of the real/imaginary parts
θ	Scale parameter in the Gamma distribution
$\tilde{\mathbf{D}}_l$	Transmitted data matrix with l^{th} column removed
$\tilde{\mathbf{h}}_l$	Channel tap vector with l^{th} element removed
$\underline{\mathbf{d}}_i$	i^{th} column vector of D matrix
$\zeta(\beta, \eta; k, \theta)$	Integral defined in the derivation of the K-distribution MMSE estimate
B	Low-pass filter bandwidth
$E[\cdot]$	Expected value
f_c	Carrier frequency
f_s	Sampling frequency
$J(\cdot)$	The objective function in a maximization problem
k	Shape parameter in the Gamma distribution
$K_p(\cdot)$	p^{th} order modified Bessel function of the second kind
L	Length of the low-pass filter
m_k	k^{th} sample moment
$p(x;\theta)$	The probability density function of x, parameterized by parameter θ
$p(x\|z)$	The probability density function of x, conditioned on the random variable z
$z, \mathbf{z}$	Gamma distributed variable in component representation of K-distributed fading, half the variance of complex-valued $h, \mathbf{h}$

Chapter 1

Introduction

Since electromagnetic waves experience high levels of attenuation in the ocean, electromagnetic communication systems have had very limited applicability to the underwater environment. Instead, acoustic systems have become the dominant means by which wireless signals are transmitted in the ocean. However, designing efficient underwater acoustic communication systems is a challenging task due to the inherent nature of sound wave propagation and the unforgiving ocean environment.

There are many aspects of the ocean that hinder the development of efficient underwater acoustic communication systems. Scattering from the sea surface results in multi-path and, along with the motion of the source and receiver, causes the channel to have a large Doppler spread. Furthermore, the rapidly varying dynamics of the ocean surface decrease the channel's coherence time. Another fundamental property that differentiates the underwater acoustic channel from typical electromagnetic communication channels is its propagation speed. Since sound propagates in the ocean at a speed of approximately 1500 m/s (compared to the 300,000,000 m/s speed of propagation for electromagnetic waves), the channel's state may fluctuate faster than it can be updated and recognized by the receiver. The performance of an underwater acoustic communication system relies on the ability of the receiver to estimate the channel's time-varying impulse response. For this reason, the effect of scattering from the ocean surface presents one of the biggest challenges to underwater acoustic communication [11].

1.1 Motivation

The underwater acoustic channel impulse response is often assumed to exhibit Rayleigh or Rician fading behavior. This corresponds to the complex valued channel taps following a complex normal distribution. A common justification for such an assumption is that each resolvable "arrival" in the impulse response is comprised of a sufficient number of independently scattered signals for the central limit theorem to hold. However, it has been shown in the case of sonar, that non-Rayleigh reverberation can occur when the central limit theorem is violated. This typically results in probability distributions for the envelope of the channel impulse response that are characterized by heavier tails. One such distribution, the K- distribution, is widely used to model radar clutter and has also proven accurate in analysis of sonar reverberation [1].

It can be shown that the K-distribution provides a more accurate statistical description of fading in certain underwater channels. The goal of this thesis is to explore the benefits of utilizing this description in underwater acoustic communication applications and to discover connections it may have with the environmental conditions.

1.2 Review of the Literature

The K-distribution has probability density function

$$f(x;\nu,\alpha) = \frac{2}{\alpha\Gamma(\nu+1)}\left(\frac{x}{2\alpha}\right)^{\nu+1} K_\nu\left(\frac{x}{\alpha}\right), \tag{1.1}$$

which has shape parameter, ν, and scale parameter, α. This distribution first garnered attention in the 1970's as a statistical model for scattering in radar applications after the advent of high resolution radar [8]. After widespread success in this field, it was also adopted as a statistical scattering model by the sonar community [1]. More recently, interest in the K-distribution has resurfaced in the context of mobile and underwater acoustic communication systems.

Jakeman and Pusey [8] first introduced the K-distribution as a computationally convenient representation of amplitude statistics in their model of the scattered field:

$$E(\mathbf{r},t) = e^{j\omega t}\sum_{i=1}^{N} a_i(\mathbf{r},t)e^{j\phi_i(\mathbf{r},t)} = A(\mathbf{r},t)e^{j(\Phi(\mathbf{r},t)+\omega t)} \tag{1.2}$$

In this model, the random variables $a_i(\mathbf{r},t)$ and $\phi_i(\mathbf{r},t)$ are the amplitude and phase of the radiation from the i^{th} scatterer at time t and position $\mathbf{r}$. Each element of $\{a_i\}$ and $\{\phi_i\}$ is assumed independent of all other elements contained in both sets. The $\{\phi_i\}$ are assumed uniformly distributed between 0 and 2π, which inherently assumes that the position of the scatterers are such that the induced path difference is larger than the incident radiation wavelength.

The authors noted that if the amplitudes were distributed in such a way that only a small portion of the total number of scatterers contributed significantly to the field at any given point, the resulting envelope could be non-Rayleigh, even for large N. The "effective" number of scatterers were identified as

$$N_{eff} = N\frac{E[a^2(\mathbf{r})]^2}{E[a^4(\mathbf{r})]}. \tag{1.3}$$

Jakeman and Pusey suggested modeling the $\{a_i\}$ as K-distributed random variables, since this led to analytically feasible expressions for the distribution of the squared envelope and its corresponding moments. While they did not offer a physical justification for this choice, they eluded to work done by Valenzuela and Laing [14], claiming that non-Rayleigh sea clutter could be explained by the composite scattering model. In their model, the return from a single "patch" on the sea surface could be represented as the product of two variables: one for small scale roughness, the other for large-scale roughness. Jakeman and Pusey suggested that the K-distribution could fit a model of this form if the energy spectral density of the small-scale roughness were exponential (corresponding to a Gaussian amplitude) while that of the large-scale roughness followed the Chi-square distribution.

Equation 1.2 can be interpreted as a two dimensional random walk at its N^{th} step. That is, the step sizes are represented by independent complex random variables. Jakeman [8] was able to relate the overall amplitude of this random walk to the K-distribution. He noted that for constant N, A will be Rayleigh distributed in the limit of large N as a consequence of the central limit theorem. If the $\{a_i\}$ are each Rayleigh distributed, the resulting amplitude will also be Rayleigh distributed for any N. However, if N is a random variable governed by the negative binomial distribution, the distribution of A will approach the K-distribution as the mean of N gets very large. Jakeman also states that if the step sizes $\{a_i\}$ are each K-distributed, A will also be K-distributed for any fixed N. As the number of step sizes is increased, the shape parameter increases linearly with N [7]. Consequently, as the number of steps is taken to infinity, the K-distributed amplitude becomes Rayleigh distributed as the central limit theorem requires.

Ward [16] expanded on the compound scattering theory in his representation of K-distributed clutter. He claimed that the amplitude of K-distributed clutter could be modeled as the product of two components with different correlation times. Through the use of frequency agility, he was able to de-correlate the returns, isolating the component with a slow correlation time. Experimental data suggested that the slowly varying component was well fit by a chi distribution [16], and then more generally, its square root by the gamma distribution [17]. The second component was assumed to be a result of the changing interference pattern, and it was assumed to be Rayleigh distributed. Ward et. al. [17] related the gamma distributed component to the local power, which depends on the current sea state.

Abraham [1] later examined the K-distribution in the context of match-filtered Sonar clutter. The envelope of the match-filter output was approximated by a two dimensional random walk of the same form as (1.2). He was then able to show that exponentially distributed amplitudes in the random walk also result in K-distributed envelopes. The use of the exponential distribution was justified, as it is often used to describe the size distribution of natural objects and could therefore be considered a valid model for the scatterer size. It was also noted that other amplitude distributions could yield a K-distributed envelope, such as the Gamma distribution. His model suggested that the scale parameter of the K-distribution would be proportional to the number of scatterers, and consequently, the beam-width of the the Sonar's array. This prediction was verified experimentally.

In communications literature, the time-varying nature of the amplitude and phase of a transmitted signal are referred to as the "fading statistics", and have been of great interest since the introduction of wireless communication systems. Fading is generally characterized as either slow or fast. Fast fading quickly de-correlates from pulse to pulse, and is generally associated with multi-path. Slow fading has a longer correlation time, and is usually attributed to shadowing effects. The K-distribution also has connections to previous and current research in this field, as it is one example of a spherically invariant random process (SIRP). It may also be considered a good approximation to the Rayleigh/lognormal fading model, which is a composite model consisting of lognormal shadowing and Rayleigh multi-path [13].

1.3 Scope of the Thesis

A discrete-time received signal y(n) can be expressed as the convolution of the channel taps with the transmitted signal, plus a noise term.

$$y(n) = \sum_{i=0}^{M-1} h_k(n)d(n-k) + w(n) \tag{1.4}$$

The channel taps are denoted $h_k(n)$, where k denotes the k^{th} tap and n represents the time dependence of the channel impulse response. If $h_k(n) = h_k$ is time-invariant, the channels taps can be treated as parameters and estimated accordingly. However, time-varying channel taps (as in the underwater channel) are generally modeled as random processes which must be tracked by the receiver. Further complication is introduced to the system when the parameters of the distributions used to model the time-varying channel taps also fluctuate in time. A receiver must then jointly estimate the channel coefficients and the parameters of its fading statistics.

When the complex valued channel taps are modeled as Gaussian random processes, the estimators take on particularly simple, linear forms that depend solely on the first and second order statistics. Adaptive algorithms can then be employed to track the channel fluctuations.

Chapter 2 of this thesis will investigate the fading statistics of a shallow water acoustic communication channel with source to receiver distances of 80, 250, 500, and 1000 meters. It will demonstrate that the underwater acoustic channel's fading behavior is not always Gaussian, and can often be better characterized by the K-distribution. The fading parameters will be tested for dependence on environmental conditions (i.e. wind speed, wave height) and the bandwidth of the system. In Chapter 3, an empirical study of the scattering model will be done, in an attempt to better understand the physical mechanisms which cause the channel to exhibit K-distributed fading.

Finally, in Chapter 4, Bayesian estimation will be applied to simple channel models in the form of MAP and MMSE estimators. While this work will unrealistically assume that the channel taps are i.i.d. and of a relatively low order, it will provide some insight into the benefit of incorporating K-distributed fading models in more advanced channel estimation algorithms.

Chapter 2

Statistical Analysis of Experimental Data

2.1 Introduction

Before pursuing estimation procedures utilizing K-distributed fading statistics, the need for such estimators must be demonstrated. The first goal of this chapter is to present an underwater acoustic communication channel which exhibits K-distributed fading. The second, is to relate the parameters of the fading statistics to the environmental conditions and the physical parameters of the system. This will illustrate the need to analyze algorithms that can exploit this additional knowledge of the channel statistics.

2.2 Parameter Estimation

2.2.1 Maximum Likelihood

The maximum likelihood (ML) estimate of a distribution's parameter $\boldsymbol{\theta}$ is the estimate which maximizes the likelihood of the observed data. Given the vector $\mathbf{x}$ of independent realizations of the variable x, the Likelihood function is given by

$$L(\boldsymbol{\theta}|\mathbf{x}) = p(\mathbf{x};\boldsymbol{\theta}), \tag{2.1}$$

The ML estimate of $\boldsymbol{\theta}$ is then

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} L(\boldsymbol{\theta}|\mathbf{x}). \tag{2.2}$$

Often, the function $\ln(L(\boldsymbol{\theta}|\mathbf{x}))$ is maximized in place of (2.2). Since the natural logarithm is a monotonically increasing function, this is equivalent to maximizing $L(\boldsymbol{\theta}|\mathbf{x})$, and often leads to computationally simpler maximization problems.

2.2.2 Method of Moments

The method of moments is a parameter estimation technique based on sample moments from a set of observations. The sample moments are equated to the theoretical expressions for the corresponding moments, producing a set of equations that can be solved for the distribution parameters. Although this technique has no optimality properties, it is easy to implement and, given a large enough data set, tends to be fairly consistent [10]. The k^{th} sample moment of the observation set $\{x_1, x_2, ...x_N\}$ is given by

$$m_k = \frac{1}{N}\sum_{i=1}^{N} x_i^k. \tag{2.3}$$

The sample moments are equated to the calculated moments,

$$m_k = E[x^k], \tag{2.4}$$

where the expectation is taken with respect to $p(x;\boldsymbol{\theta})$. This results in the necessary equations. For the purpose of this thesis, only the necessary moments required to obtain a closed-form solution are used.

2.2.3 Rayleigh Distribution

If many scatterers contribute to the intensity of the arrival at a given delay, the central limit theorem can be applied. In this case, a single tap of the channel impulse response, denoted by h, is modeled as a zero-mean complex Gaussian random variable. The resulting envelope, $x = |h|$, is Rayleigh distributed with the probability density function

$$p(x_i;\sigma) = \frac{x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}}. \tag{2.5}$$

The parameter of the Rayleigh distribution, σ^2, is half the variance of the complex Gaussian random variable.

Given a vector $\mathbf{x}$ of independent observations, the estimate of σ can be made by use of the maximum likelihood method.

$$\hat{\sigma} = \arg\max_{\sigma} p(\mathbf{x};\sigma) = \arg\max_{\sigma} \ln(p(\mathbf{x};\sigma)) \tag{2.6}$$

$$= \arg\max_{\sigma} \left\{ \ln \prod_{i=1}^{N} \frac{x e^{\frac{-x^2}{2\sigma^2}}}{\sigma^2} \right\} \tag{2.7}$$

$$= \arg\max_{\sigma} \left\{ -2N \ln \sigma + \sum_{i=1}^{N} \ln x - \frac{1}{2\sigma^2} \sum_{i=1}^{N} x^2 \right\} \tag{2.8}$$

Taking the derivative with respect to σ and setting equal to zero yields

$$\frac{d}{d\sigma}(\ln p(\mathbf{x};\sigma)) = \frac{-2N}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{N} x^2 = 0 \tag{2.9}$$

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{N} x^2}{2N}} \tag{2.10}$$

2.2.4 Rician Distribution

When the arrival process has a nonzero mean, the Rayleigh distribution can be generalized to the Rician distribution. This distribution is commonly observed in fading channels that consist of a direct line of sight component that is combined with many weaker signals. The probability distribution function is

$$p(x;\mu,\sigma) = \frac{x}{\sigma^2} e^{\frac{-(x^2+\mu^2)}{2\sigma^2}} I_0\left(\frac{x\mu}{\sigma^2}\right) \tag{2.11}$$

where I_0 is the zeroth order modified Bessel function of the first kind, σ^2 is again half the variance of the complex Gaussian random variable, and μ is the mean of the complex random variable. Since there is no closed form solution to the maximum likelihood estimate of the parameters μ and σ, the method of moments was applied. The k^{th} moment of the Rician random variable x can be expressed as [12],

$$E[x^k] = \int_0^\infty \frac{x^{k+1}}{\sigma^2} e^{\frac{-(x^2+\mu^2)}{2\sigma^2}} I_0\left(\frac{x\mu}{\sigma^2}\right) \tag{2.12}$$

$$= \left(2\sigma^2\right)^{k/2} \Gamma\left(1 + \frac{k}{2}\right) {}_1F_1\left[\frac{k}{2}; 1; \frac{-\mu^2}{2\sigma^2}\right] \tag{2.13}$$

where ${}_1F_1$ is the confluent hypergeometric function. From (2.13) the second moment is [12]

$$E[x^2] = \mu^2 + 2\sigma^2. \tag{2.14}$$

Moment estimates can be made using the k^{th} sample moment as in (2.3), Estimates of σ can be made by using the 4^{th} moment in conjunction with (2.14)

$$E[x^4] = \mu^4 + 8\sigma^2\mu^2 + 8\sigma^4. \tag{2.15}$$

However, if observations of the real and imaginary parts of the signal are available, the method of moments can be applied to them instead. Given real $R \sim N(\mu \cos\theta, \sigma^2)$ and imaginary $Z \sim N(\mu \sin\theta, \sigma^2)$, the magnitude $X = \sqrt{R^2 + Z^2}$ will be Rician with variance σ^2. An unbiased estimate of σ^2 can be made using both the real and imaginary parts of the signal.

$$\widehat{\sigma_R^2} = \frac{1}{N-1} \sum_{i=1}^{N} (r_i - m_R)^2 \tag{2.16}$$

$$\widehat{\sigma_Z^2} = \frac{1}{N-1} \sum_{i=1}^{N} (z_i - m_Z)^2 \tag{2.17}$$

Here m_R and m_Z are sample means of R and Z, respectively. Equations (2.16) and (2.17) use $N - 1$ instead of N. Although these values are very similar for large N, the former results in an unbiased estimate of the variance. The estimates obtained by Z and R are then averaged together to form the final estimate.

$$\widehat{\sigma^2} = \frac{\widehat{\sigma_R^2} + \widehat{\sigma_Z^2}}{2} \tag{2.18}$$

The value of μ is then estimated using the second sample moment of x, via (2.14).

$$\hat{\mu} = \sqrt{m_2 - 2\widehat{\sigma^2}} \tag{2.19}$$

Due to the square root operation, the estimator in (2.19) can produce an imaginary result that falls outside the range of admissible values for μ. In this case, the absolute value of the estimate will be taken. Should a more accurate estimate be needed, a method for numerically finding the maximum likelihood estimator is discussed in [12].

2.2.5 K-Distribution

Repeated from (1.1), the K-distribution is

$$p(x; \nu, \alpha) = \frac{2}{\alpha\Gamma(\nu+1)} \left(\frac{x}{2\alpha}\right)^{\nu+1} K_\nu\left(\frac{x}{\alpha}\right). \tag{2.20}$$

No closed form solution exists for the maximum likelihood estimate of ν and α, thus the method of moments can be applied once again. The k^{th} moment of a K-distributed random variable is

$$m_k = \frac{\Gamma\left(\frac{k}{2}+1\right)\Gamma\left(\nu+1+\frac{k}{2}\right)(2\alpha)^k}{\Gamma(\nu+1)} \tag{2.21}$$

The simplest parameter estimates can be obtained from the second and fourth moments, which lead to closed form expressions.

$$\frac{m_4}{(m_2)^2} = \frac{2(\nu+2)}{\nu+1} = m \tag{2.22}$$

$$\hat{\nu} = \frac{m-4}{2-m} \tag{2.23}$$

$$\hat{\alpha} = \sqrt{\frac{m_2}{4(1+\nu)}} \tag{2.24}$$

Alternatively, as higher order moments tend to have a higher variance, a better estimate might be expected using the first and second moments. However, there is no closed form solution in this case. Even better performance could be obtained by numerically evaluating the maximum-likelihood solution [9].

2.3 Experimental Procedure

2.3.1 Experimental Setup

The data analyzed in this thesis were collected in 2002 as part of the SPACE02 (Surface Processes and Acoustic Communication Experiment 2002) conducted by the Woods Hole Oceanographic Institution. During this experiment, an acoustic transmitter was placed off the coast of Edgartown, Massachusetts in the vicinity of the Marthas Vineyard Coastal Observatory (MVCO). Vertical receiver arrays were placed southwest of the transmitter at distances of 80, 250, 500, and 1000 meters. Another receiver comprised of both a vertical and horizontal array was placed 80 meters southeast of the transmitter.

The experiment was conducted in a shallow water environment (16 meters), with the source located 6.25 meters above the sea floor. The receivers were located at distances of 80, 250, 500, and 1000 meters from the source, positioned 3.3 meters above the sea floor. Assuming perfect reflectance from both the ocean bottom and sea surface, the arrival times of the scattered signals can be estimated using the "method of images". In this method, the sound reflected from the boundary is assumed to originate in a location that corresponds to the mirror image of the source. Trigonometry gives us the distance of each propagation

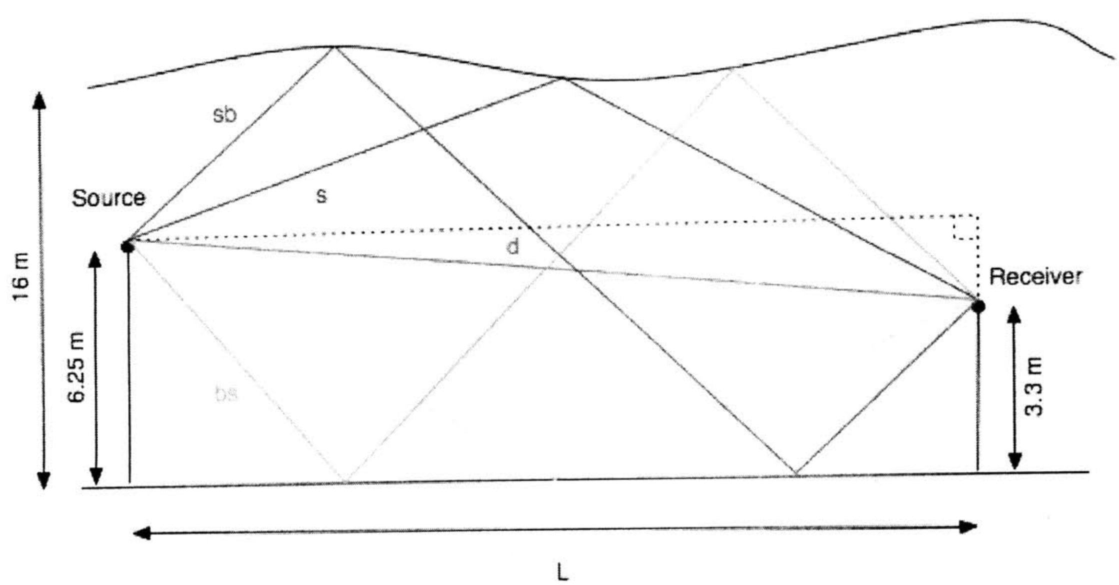


Figure 2-1: Channel geometry: the method of images

path, and the arrival time is determined by dividing this by the sound speed, c=1450 m/s.

$$\tau_d = \frac{\sqrt{L^2 + (6.25 - 3.3)^2}}{c} \tag{2.25}$$

$$\tau_s = \frac{\sqrt{L^2 + (16 - 6.25 + 16 - 3.3)^2}}{c} \tag{2.26}$$

$$\tau_b = \frac{\sqrt{L^2 + (6.25 + 3.3)^2)}}{c} \tag{2.27}$$

$$\tau_{bs} = \frac{\sqrt{L^2 + (16 - 3.3 + 16 + 6.25)^2}}{c} \tag{2.28}$$

$$\tau_{sb} = \frac{\sqrt{L^2 + (16 + 3.3 + 16 - 6.25)^2}}{c} \tag{2.29}$$

Table 2.1 summarizes the relative arrival times. That is, the direct arrival (τ_d) has been subtracted from each arrival as calculated above.

Table 2.1: Relative arrival times

L (m)	$\tilde{\tau}_b$ (ms)	$\tilde{\tau}_s$ (ms)	$\tilde{\tau}_{sb}$ (ms)	$\tilde{\tau}_{bs}$ (ms)
80	0.354	2.09	3.49	4.99
250	0.114	0.682	1.15	1.66
500	0.0569	0.341	0.576	0.835
1000	0.0284	0.171	0.288	0.418

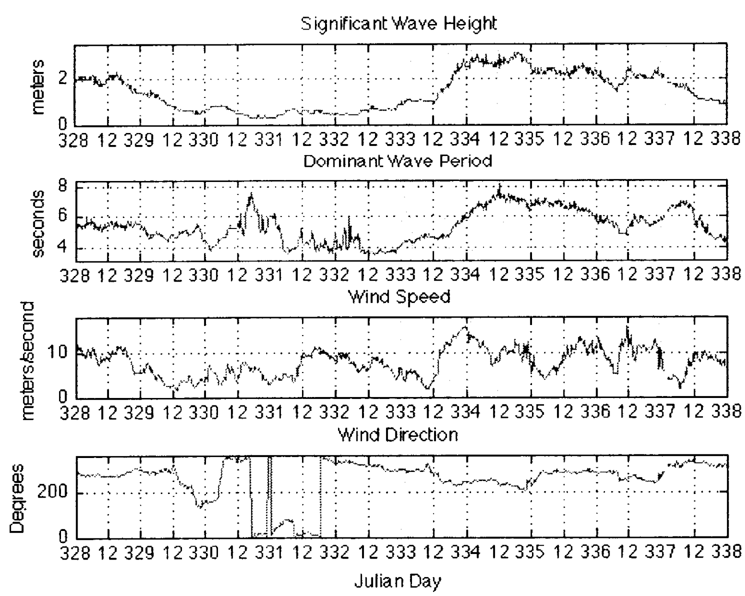


Figure 2-2: Environmental conditions: wind and wave data

Environmental conditions were monitored by the MVCO. The significant wave height, dominant wave period, wind speed, and wind direction were recorded throughout the course of the experiment. The experiment spanned the length of a passing storm, creating a broad range of environmental conditions. The wind and wave data are summarized in Figure 2-2, where vertical green lines mark the epochs chosen for the analysis in this thesis. Epoch will be labeled by its Julian Day number.

2.3.2 Data Processing

At the start of each epoch, pulses were sent out over a carrier frequency of $f_c = 14kHz$ in seven segments. In each segment, there were 1307 pulses, with a pulse transmitted every 0.04 seconds. The received signals were sampled at a rate of $f_s = 5 \times 10^6/112$ kHz. The discrete-time received signals, $r_i(n)$, were shifted to baseband and then passed through discrete moving-average filters with filter lengths of 4, 8, 16, 32, 64, 128, and 256 samples. The low pass filter of length L is defined as

$$g(n) = \begin{cases} \frac{1}{L} & 0 \leq n \leq L-1 \\ 0 & otherwise \end{cases} \tag{2.30}$$

and has discrete-time Fourier transform

$$G(j\omega) = \frac{sin(\omega L/2)}{sin(\omega/2)} e^{-j\omega(L-1)/2}. \tag{2.31}$$

The 3dB bandwidth of the filter in Hz is approximately

$$B \approx \frac{f_s}{L} \tag{2.32}$$

The baseband, low-pass filtered, complex-valued received signal for the i^{th} pulse is denoted $\tilde{r}_i(n)$ and is given by the equation

$$\tilde{r}_i(n) = g(n) * (r_i(n) e^{j2\pi n f_c/f_s}), \tag{2.33}$$

and its real-valued envelope, $x_i(n)$, is defined as

$$x_i(n) = |\tilde{r}_i(n)|. \tag{2.34}$$

High intensity spikes spanning the length of the pulse were present in the majority of the data. These anomalies were most likely due to noise associated with equipment motion, such as the rattling of chains. To analyze the channel statistics accurately, the majority of these spikes needed to be culled from the data. For each segment, the normalized difference in magnitude for the i^{th} pulse was calculated as

$$\Delta x_i(n) = \frac{x_{i+1}(n) - x_i(n)}{\max_{j,k}(x_j(k))} \tag{2.35}$$

The i^{th} pulse was then flagged as abnormal and removed from the data set if

$$\Delta x_i(n) > 0.3 \quad \text{and} \quad \Delta x_{i+1}(n) < -0.3. \tag{2.36}$$

To determine the characteristics of the channel statistics, histograms and moments were calculated at each time in delay, using the retained "good" pulses. Figure 2-3 displays a plot of the average and maximum intensity values at each time in delay. The histograms and statistics were averaged over a 0.2016 millisecond window in delay about the designated single surface bounce arrival time (the local peak in average intensity). This window is marked by vertical green lines in the figure.

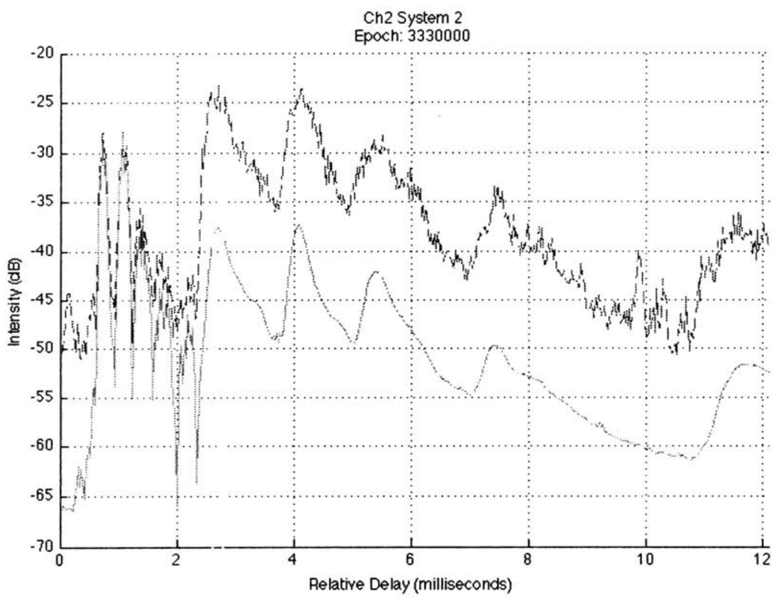


Figure 2-3: Maximum (blue dotted line) and average (red solid line) intensity, epoch 3330000

From this, parameter estimates were made either using maximum likelihood estimation or the method of moments. Finally, a Matlab algorithm was run to find the "best fit" distribution. This algorithm was initialized with the parameter estimates and then refined them by conducting a local 11x11 grid search to minimize total absolute error between the histogram and the estimated distribution. When trying to match a given histogram, $hist(x)$, to a distribution parametrized by s and g, the algorithm is as follows:

Grid Search Algorithm

- Initialize $\Delta s = 0.1$
- While $\Delta s > 0.001$ Do:

  ◇ $s(k) = s_{opt} \pm k s_{opt} \Delta s$, for $k = \pm 1, 2, ...5$
  $g(i) = g_{opt} \pm i g_{opt} \Delta s$, $i = \pm 1, 2, ...5$

  ◇ $(k_{opt}, i_{opt}) = \arg\min_{k,i} \sum_x |hist(x) - p(x; s_k, g_i)|$

  ◇ if 5-$|k_{opt}| \leq 2$ and 5-$|i_{opt}| \leq 2$ then $\Delta s := \frac{\Delta s}{2}$

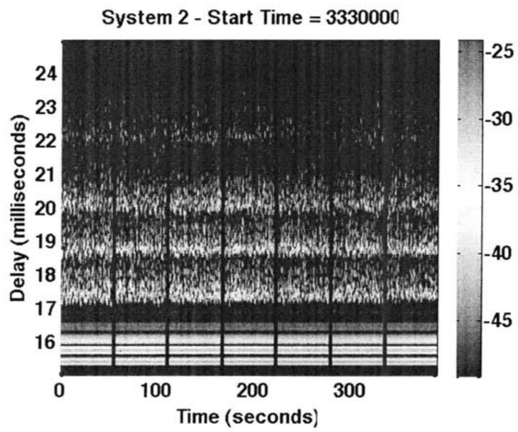
  ◇ $s_{opt} := s(k_{opt})$, $g_{opt} = g(i_{opt})$

The Matlab code implementing the procedure outlined in this section can be found in Appendix A.

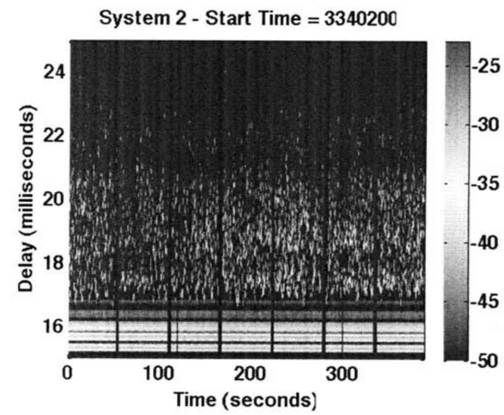
2.4 Results

2.4.1 80 Meter Data Analysis

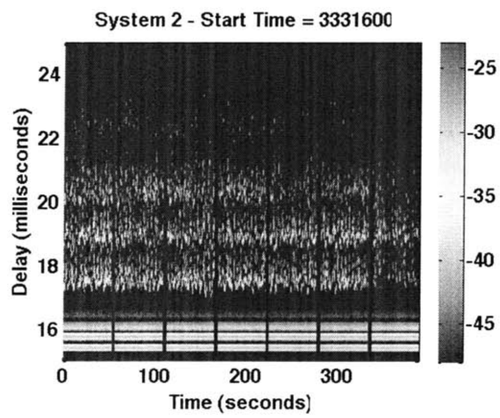
The data obtained from the receiver positioned 80 meters from the source are particularly interesting because the reflected signal paths portrayed in Figure 2-1 can each be individually identified. The channel impulse response from epoch 3310000, 3331600, 3340200, and 3341200 are shown in Figure 2-4. Epoch 3310000 corresponds to a particularly calm day, while epoch 3340200 represents a day with both high waves and strong wind activity. Epochs 3331600 and 3341200 correspond to days of moderate wind with high waves and high wind with moderate waves, respectively. A change in environmental conditions is evident in the nature of the scattering behavior. Higher wind and wave activity has a "smearing" effect on the delay spread, causing scattered arrivals to be less distinct and spread further in delay. It is common in acoustic modeling to express the acoustic surface loss as a function of wave height. A rougher sea state will increase the scattering effect that the surface has on the incident wave, thereby causing the coherent received signal to have a smaller amplitude [4]. An arrival at a specific time in delay will, on average, experience a reduction in amplitude during high wind and wave activity. However, large peaks in amplitude may be present, due to wave focusing.



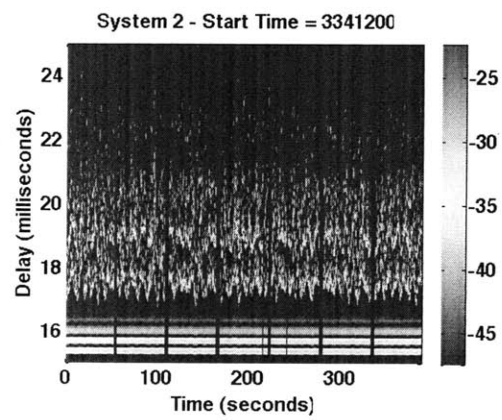
(a) Low wind and wave activity



(b) High wind and wave activity



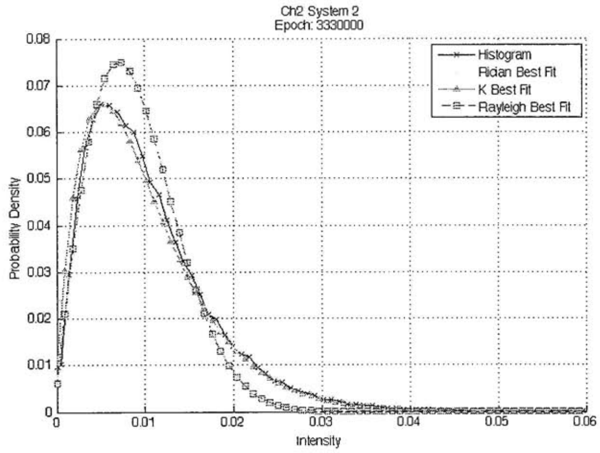
(c) High waves and moderate wind activity



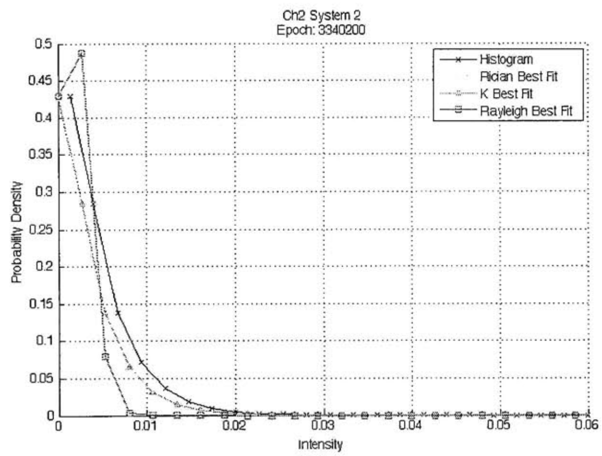
(d) High wind and moderate wave activity

Figure 2-4: Channel impulse response: 80 meters

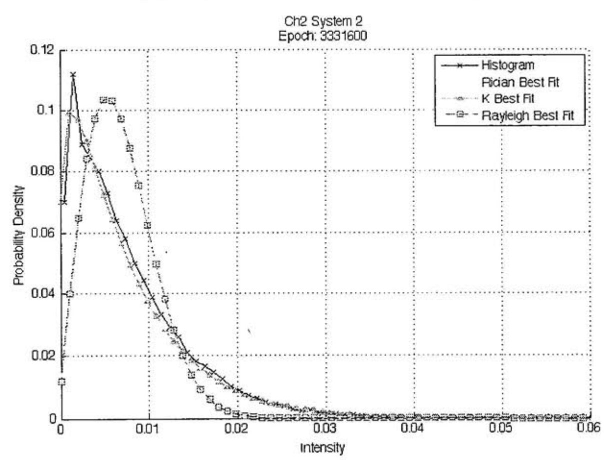
Envelope probability distribution functions for the arrivals with a single surface interaction are plotted in Figure 2-5. The parameters of the envelope distributions for the 12 epochs are summarized in Tables 2.4(a) and 2.5. For all filter lengths, the largest value of ν was consistently observed at epoch 3310000, under the calmest weather conditions. Conversely, the minimum values were obtained at either epoch 3340200, 3331600, or 3341400, all epochs of high wind and/or wave activity.Qualitatively, this is expected because of the "smearing" effect discussed earlier. These results suggest that the size of the shape parameter is correlated with the surface conditions.



(a) Low wind and wave activity



(b) High wind and wave activity



(c) High waves and moderate wind activity

Figure 2-5: Envelope distributions: 80 meters

Further evidence for this correlation can be obtained by observing the sample partial correlation coefficients between ν and the significant wave height. Partial correlation coefficients were evaluated in place of ordinary correlation coefficients in order to isolate the dependence on wind speed, wave height, and wave period. Note that the partial correlation falls between -0.8 and -0.9 for small filter lengths in the first column of Table 2.6(a). This implies that the shape parameter decreases as wave activity increases. Abraham showed that the shape parameter was directly proportional to the number of scatterers. [1]. This observation is therefore in agreement with the physical description of the channel.

Table 2.2: Envelope distribution parameters for varying filter length: 80 meters

(a) K-distribution paramters

Filter Length	Maximum		Minimum		Average	
	ν	α	ν	α	ν	α
4	6.87	0.00726	-0.427	0.00345	0.665	0.00547
8	137	0.00579	-0.107	0.000715	11.1	0.00401
16	123	0.00229	1.98×10^{-13}	0.000539	11.6	0.00229
32	123	0.00147	0.924	0.00028	18.2	0.000869
64	133	0.000715	1.06	0.000132	20.1	0.000373
128	136	0.000246	5.09	6.21×10^{-5}	43.3	0.000145
256	131	0.000131	4.11	1.96×10^{-5}	47.2	7.42×10^{-5}

(b) Rician parameters

Filter Length	Maximum		Minimum		Average	
	μ	σ	μ	σ	μ	σ
4	0.00303	0.0136	2.33×10^{-11}	0.00217	0.000826	0.00743
8	0.0103	0.0092	1.44×10^{-11}	0.0028	0.000984	0.00667
16	0.00803	0.00651	1.14×10^{-11}	0.00219	0.000677	0.00507
32	0.00391	0.00388	3.71×10^{-12}	0.0017	0.000301	0.0031
64	0.00199	0.00199	3.28×10^{-12}	0.000941	0.000153	0.0016
128	0.00125	0.00102	1.87×10^{-12}	0.000551	0.00019	0.000852
256	0.000601	0.000514	5.13×10^{-13}	0.000292	8.35×10^{-5}	0.00043

(c) Rayleigh parameters

Filter Length	Maximum	Minimum	Average
	σ	σ	σ
4	0.0136	0.00226	0.0075
8	0.0119	0.0028	0.00686
16	0.00845	0.00245	0.00507
32	0.00439	0.0017	0.0032
64	0.00216	0.000943	0.00164
128	0.00132	0.000551	0.000905
256	0.000643	0.000292	0.000453

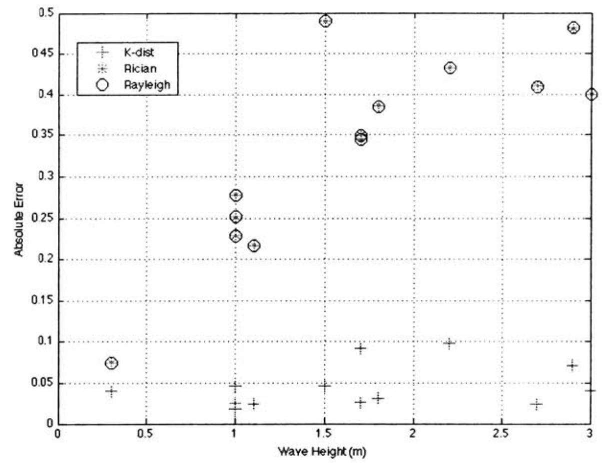
The dependence on wave height seems to diminish as the filter length increases. The value of the shape parameter (and consequently the number of scatterers) also increases on average with increasing filter length. There is a similar correlation between wave height and the value of σ in the Rician/Rayleigh distributions. Additionally, the error caused by assuming Rayleigh/Rician fading increases with wave height, while the error obtained using a K-distribution remains relatively constant. This is shown in Figure 2-6(a), along with the associated sample partial correlation coefficients. Conversely, the average distribution errors associated with Rayleigh and Rician assumptions decrease with increasing filter length as seen in Figure 2-6(b). All of these observations indicate that the channel is becoming more Rayleigh-like as the filter length increases, and therefore the bandwidth decreases. Essentially, this corresponds to a larger "patch" on the surface, and consequently more scatterers, contributing to each resolvable arrival.

2.4.2 250, 500, and 1000 Meter Data Analysis

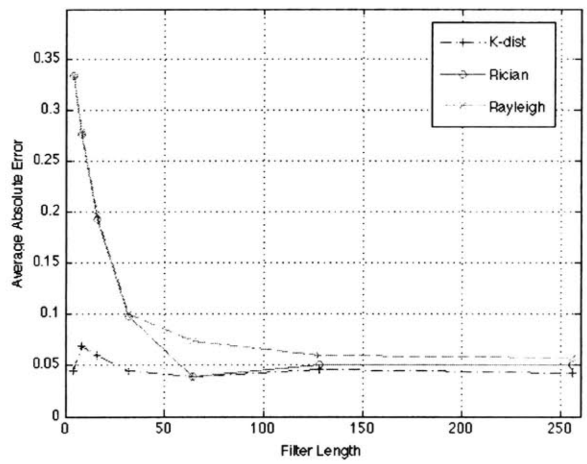
Receivers were also placed at source-to-receiver distances of 250, 500, and 1000 meters. The channel impulse response for each of these scenarios during epoch 3310000 and 3340200 is displayed in Figure 2-7 and Figure 2-8, respectively. As the receiver is moved further from the transmitter, the scattered arrivals experience higher attenuation levels. The difference between scattered arrivals also becomes less significant, and at 1000 meters it is no longer possible to individually distinguish each combination of scatterers.

Interestingly, the first scattered arrival begins to exhibit isolated episodes of Rician fading behavior at a distance of 250 meters, and then predominantly Rician fading at 500 meters. At 1000 meters, the channel is well characterized by Rayleigh fading, and consequently, also by the K-distribution with a large shape parameter. However, it should be noted that the "first" scattered arrival at 1000 meters is actually due to the surface-bottom reflection, whereas in both the 250 and 500 meter cases it is due to the first single surface reflection. The surface-bottom reflection in the 1000 meter case experiences more attenuation and is subject to more scattering from the second boundary interaction. By observing the absolute error in Figure 2-9, one can see that these Rician fading characteristics do not seem to be related to wave activity. However, they do occur more frequently and cause more severe differences in error at low wind speeds as seen in Figure 2-10.

As the K-distribution of a signal's envelope does not account for a process with non-zero



(a) Absolute error vs. wave height



(b) Average absolute error vs. filter length

Figure 2-6: Absolute distribution error trends: 80 meters

mean, the Rician distribution provides a better fit in the presence of a dominant arrival. This discrepancy could potentially be circumvented by using a generalized version of the K-distribution that relies on a Rician distribution in place of the Rayleigh distribution in the compound representation. Also, the K-distribution parameter estimate fails (i.e. returns an imaginary number) in the presence of a strong mean value. In this case, the estimate of ν is set to 100, where the distribution essentially becomes Rayleigh. Better initial parameter estimates could be obtained by subtracting out the mean prior to forming the envelope and making the estimate of the shape parameter.

Table 2.3: Partial correlation coefficients between environmental conditions and distribution parameters: 80 meters

(a) K-distribution

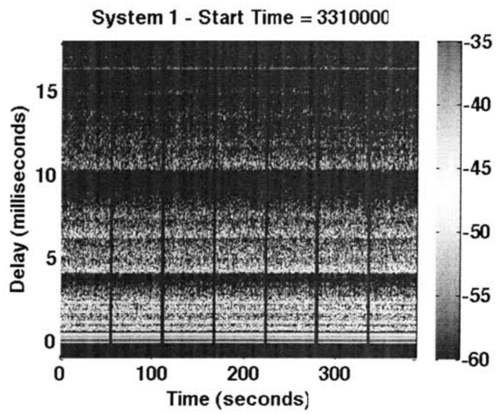
Filter Length	Wave Height (m)		Wave Period (m)		Wind Speed (m/s)	
	ν	α	ν	α	ν	α
4	-0.880	0.200	0.795	-0.252	-0.464	0.662
8	-0.824	0.595	0.776	-0.533	-0.537	0.726
16	-0.842	0.712	0.785	-0.335	-0.528	0.392
32	-0.843	0.700	0.726	-0.267	-0.332	-0.0507
64	-0.862	0.539	0.809	-0.411	-0.453	-0.0572
128	-0.571	0.389	-0.0535	-0.00508	0.0073	0.315
256	-0.215	0.146	0.0467	0.0696	-0.474	0.504

(b) Rician distribution

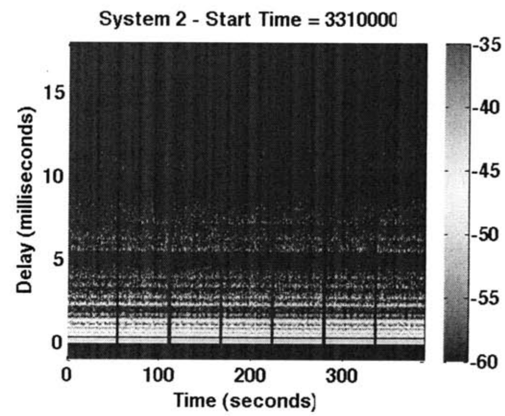
Filter Length	Wave Height (m)		Wave Period (m)		Wind Speed (m/s)	
	μ	σ	μ	σ	μ	σ
4	0.267	-0.958	-0.53	0.854	0.263	0.702
8	-0.792	-0.961	0.717	0.783	-0.509	0.910
16	-0.798	-0.907	0.767	0.455	-0.589	0.895
32	-0.818	-0.810	0.774	0.327	-0.538	0.874
64	-0.818	-0.749	0.774	0.217	-0.538	0.896
128	-0.499	-0.660	0.222	-0.197	0.0731	0.895
256	-0.516	-0.718	-0.183	0.0882	0.0397	0.861

(c) Rayleigh distribution

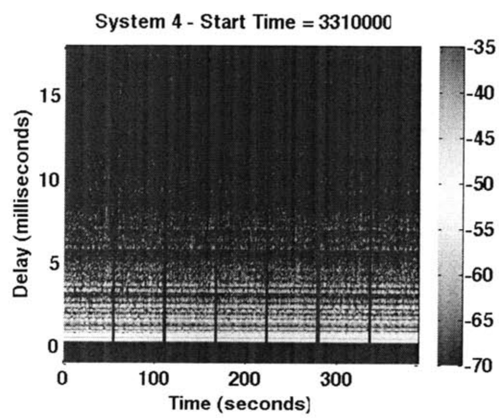
Filter Length	Wave Height (m)	Wave Period (m)	Wind Speed (m/s)
	σ	σ	σ
4	-0.963	0.860	0.736
8	-0.966	0.879	0.767
16	-0.954	0.841	0.762
32	-0.928	0.790	0.826
64	-0.919	0.772	0.85
128	-0.724	0.0839	0.727
256	-0.736	0.189	0.68



(a) 250 meters, low wind/wave activity

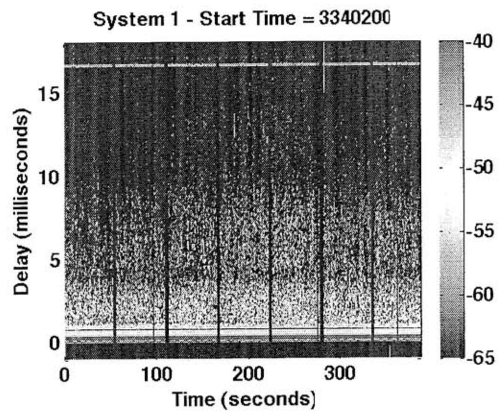


(b) 500 meters, low wind/wave activity

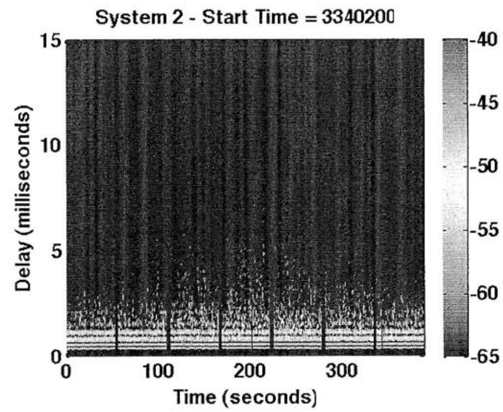


(c) 1000 meters, low wind/wave activity

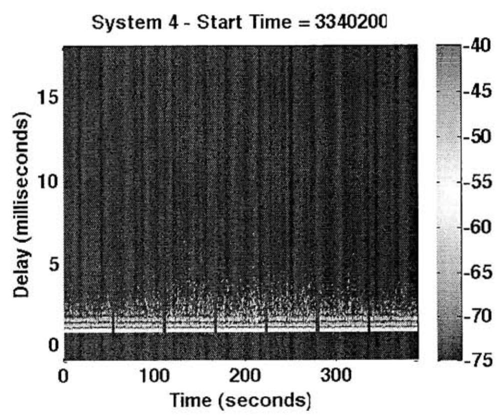
Figure 2-7: Channel impulse response during low wind and wave activity: 250, 500, and 1000 meters



(a) 250 meters, high wind/wave activity



(b) 500 meters, high wind/wave activity



(c) 1000 meters, high wind/wave activity

Figure 2-8: Channel impulse response during high wind and wave activity: 250, 500, and 1000 meters

Table 2.4: Envelope distribution parameters: 250, 500, and 1000 meters

(a) 250 meter

	K-dist		Rician		Rayleigh
	ν	α	μ	σ	σ
Maximum	127	0.00191	0.00307	0.00452	0.00503
Minimum	2.21	1.41×10^{-4}	3.19×10^{-11}	0.00112	0.00224
Average	41.5	8.51×10^{-4}	0.00112	0.003	0.0033

(b) 500 meter

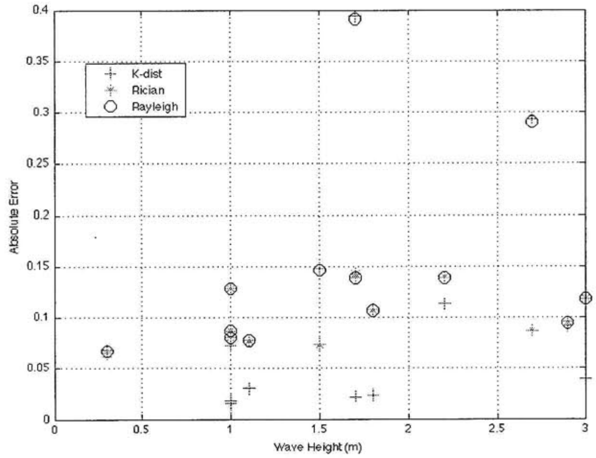
	K-dist		Rayeigh		Rician
	ν	α	μ	σ	σ
Maximum	170	0.000394	0.00384	0.0022	0.00364
Minimum	2.04	6.29×10^{-5}	1.32×10^{-12}	0.000507	0.00088
Average	114	0.000128	0.0013	0.00103	0.00155

(c) 1000 meter

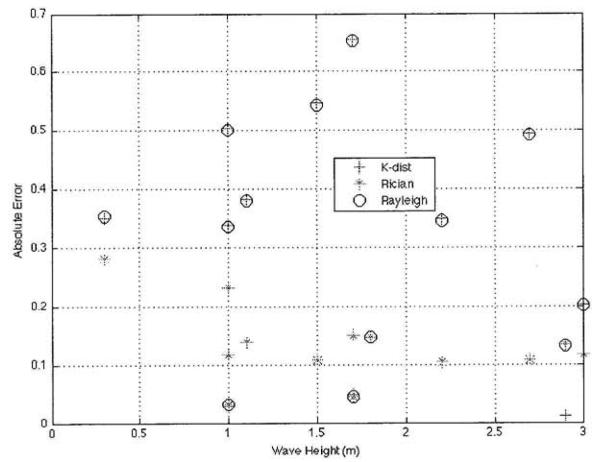
	K-dist		Rayeigh		Rician
	ν	α	μ	σ	σ
Maximum	165	0.0019	0.00268	0.00124	0.00236
Minimum	-0.0879	1.4×10^{-4}	8.02×10^{-13}	0.000371	0.000371
Average	14.7	8.95×10^{-4}	0.000302	0.000554	0.00065

Table 2.5: Total absolute distribution errors: 250, 500, and 1000 meters

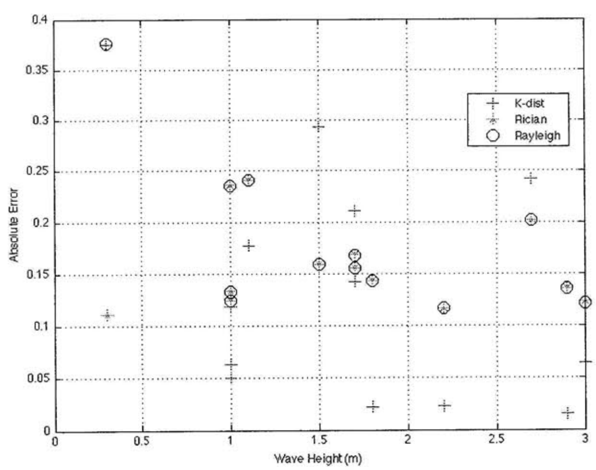
Receiver	K-distribution			Rician			Rayleigh		
Distance	Max	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.
250	0.394	0.0159	0.102	0.142	0.065	0.103	0.391	0.0668	0.143
500	0.655	0.0117	0.312	0.283	0.0320	0.132	0.653	0.0320	0.32
1000	0.375	0.0154	0.138	0.241	0.111	0.158	0.376	0.117	0.178



(a) 250 meters

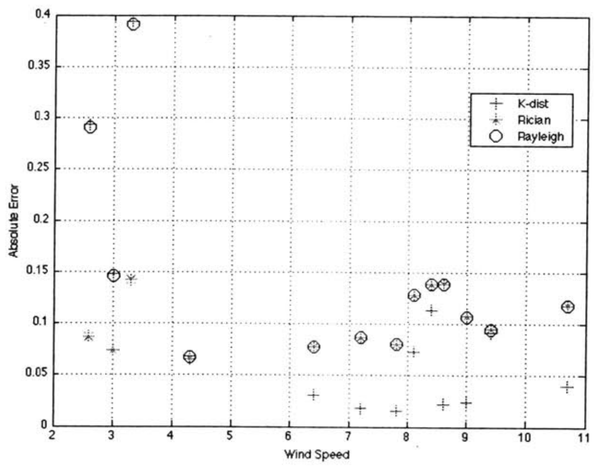


(b) 500 meters

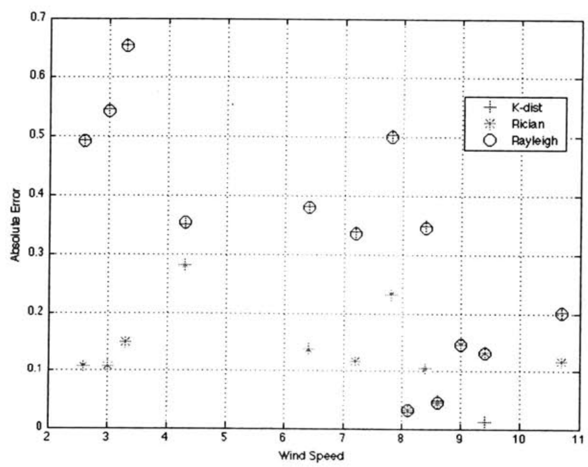


(c) 1000 meters

Figure 2-9: Error vs. wave height: 250, 500, and 1000 meters



(a) 250 meters



(b) 500 meters

Figure 2-10: Error vs. wind speed: 250 and 500 meters

Table 2.6: Sample partial correlation coefficients between environmental conditions and distribution parameters: 250, 500, and 1000 meters

(a) K-distribution

Receiver Distance (m)	Wave Height (m)		Wave Period (m)		Wind Speed (m/s)	
	ν	α	ν	α	ν	α
250	0.0782	-0.396	0.497	0.0199	-0.942	0.803
500	-0.42	-0.0967	0.315	0.314	-0.457	0.134
1000	-0.806	-0.0299	0.752	0.346	-0.51	-0.14

(b) Rician distribution

Receiver Distance (m)	Wave Height (m)		Wave Period (m)		Wind Speed (m/s)	
	μ	σ	μ	σ	μ	σ
250	-0.177	-0.914	0.563	0.776	-0.879	0.869
500	0.554	-0.76	0.548	0.722	-0.655	0.172
1000	-0.848	-0.863	0.816	0.856	-0.479	-0.374

(c) Rayleigh distribution

Receiver Distance (m)	Wave Height (m)	Wave Period (m)	Wind Speed (m/s)
	σ	σ	σ
250	-0.897	0.786	0.58
500	-0.893	0.874	-0.6
1000	-0.852	0.829	-0.487

Chapter 3

An Empirical Study of the Scattering Model

3.1 Introduction

Both the amplitude and arrival time of the surface scattered signals vary in time. It is this fluctuation that leads to the delay spread of the channel and the overall statistics of the envelope. By examining these fluctuations more closely, perhaps more insight can be gained into the phenomena that govern the channel statistics.

3.2 Modeling the Channel Statistics

The received signa is comprised of three main components: the part of the signal that varies due to the physical path taken, the transmitted waveform that has been processed, and the phase of the signal. The received signal due to a single scatterer can be expressed as

$$r_i(t) = \underbrace{A_i(t)}_{path}\,\underbrace{s(t-\tau_i)}_{waveform}\,\underbrace{e^{-j(\omega\tau_i+\theta_i(t))}}_{phase} \tag{3.1}$$

With a model for the amplitude and arrival time statistics, a crude simulation of the scattering process can be constructed. For the purpose of the simulation, the waveform is modeled as a Gaussian pulse with a pulse-width $(1/\sqrt{\alpha})$ of 0.1 milliseconds. If the signal's surface scattered component were due to just one scatterer, the envelope would be given by

$$|r(t)| = Ae^{-\alpha(t-\tau)^2} \tag{3.2}$$

where τ and A are random variables denoting the fluctuating amplitude and arrival time.

It turns out that this is not a very realistic model. Upon closer examination of the received signal, it becomes clear that several scatterers contribute to the received signal at each instance in time. This is demonstrated in Figure 3.2. The resulting received signal can be expressed as the sum of N independent, complex-valued random variables reminiscent of the "discrete scatterer model" explained by Jakeman and Pusey [8]. As in [8] and [1], the phase of each scattered component is assumed uniformly distributed between 0 and 2π.

$$r(t) = \sum_{i=1}^{N} A_i e^{-\alpha(t-\tau_i)^2} e^{j\phi_i} \tag{3.3}$$

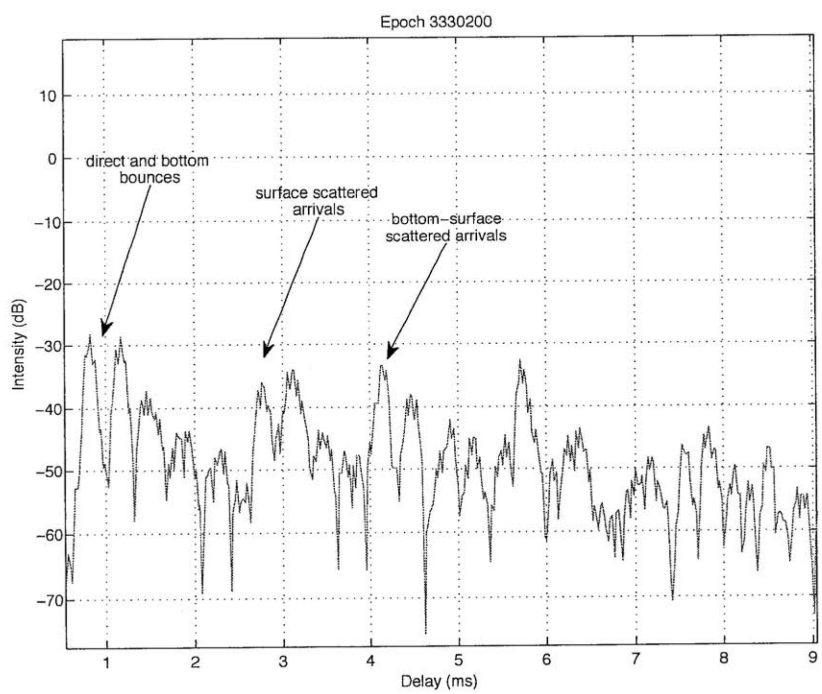
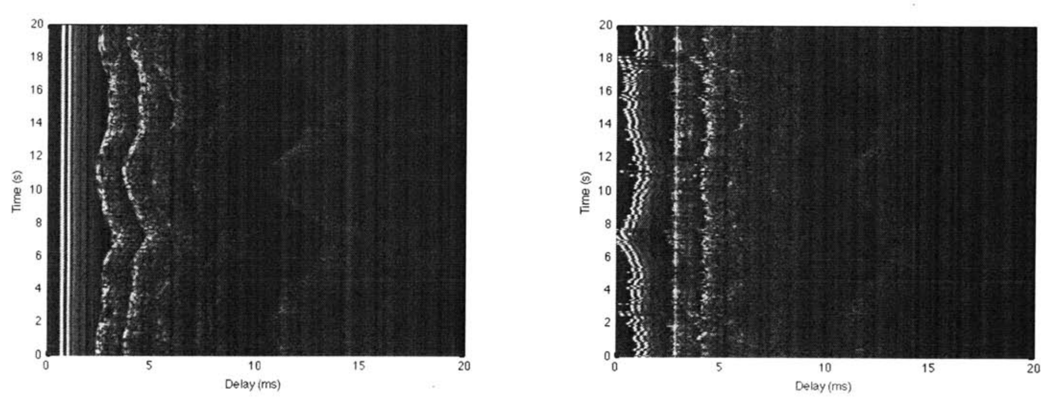


Figure 3-1: Detailed view of a received signal: epoch 3330200

3.2.1 Peak Amplitude Fluctuations

The peak amplitude and arrival time fluctuations were analyzed using the data from SPACE02, described in Section 2.3. For now, only the arrivals whose nominal propagation path includes one surface interaction and no bottom interactions are considered. The dominant single surface scattered arrival for each pulse was identified by searching for the maximum value of intensity over the extent of the delay spread for the scattered return. Figure 3-2(b) shows the channel impulse response with each pulse shifted, such that the peak single surface scattered arrivals are aligned in delay. The "straightened" impulse response was processed as discussed in Section 2.3.2. However, histogram and statistics for the envelope were averaged over a smaller window of 0.112 ms in delay. A plot of the average and maximum intensity values at each time in delay for the straightened channel impulse response of epoch 3330000 can be found in Figure 3-3. Again, vertical green lines mark the window over which the statistics are averaged.



(a) Impulse response (b) "Straightened" around first surface reflection

Figure 3-2: Channel fluctuations: 80 meters, epoch 3341200

With the effect of the changing arrival time removed, the amplitude statistics of the scattered signal can be evaluated. Figure 3-4 contains the histograms of the peak amplitudes from epochs 3330000 and 3340200. Several distributions were tested as fits, the best of wich were log-normal and gamma distributions. The log-normal distribution parameters were obtained through maximum likelihood, as discussed in Chapter 2. However, since no closed-form solution exists for both parameters of the gamma distribution, these parameters were found by the method of moments, which is described in Section 2.2.2

The log-normal distribution results when the natural logarithm of the variable is nor-

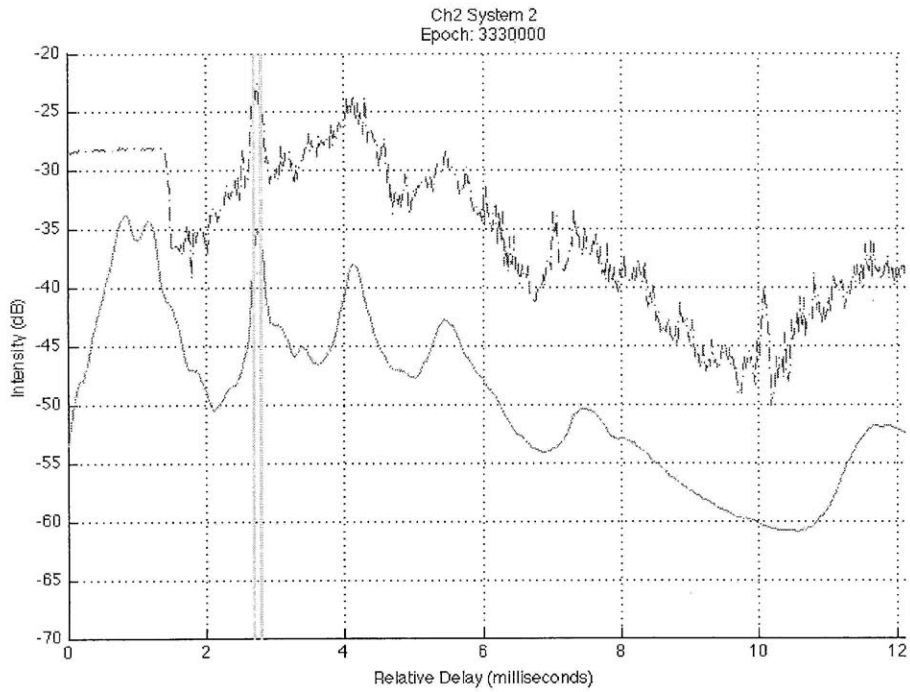


Figure 3-3: Maximum (blue dotted line) and average (red solid line) intensity: 80 meters, epoch 3330000 "straightened"

mally distributed. It has probability density function

$$p(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, \; x > 0, \tag{3.4}$$

where μ and σ are the mean and standard deviation of $\ln x$. The parameters of the lognormal distribution are estimated as follows:

$$\begin{aligned} \hat{\mu} &= \arg\max_{\mu} f(\mathbf{x}; \mu, \sigma) = \arg\max_{\mu} \ln(f(\mathbf{x}; \mu, \sigma)) & (3.5) \\ &= \arg\max_{\mu} \left\{ \prod_{i=1}^{N} \frac{1}{x_i\sqrt{2\pi\sigma^2}} e^{-\frac{(\ln x_i - \mu)^2}{2\sigma^2}} \right\} & (3.6) \\ &= \arg\max_{\mu} \left\{ -\frac{N}{2}\ln(2\pi\sigma^2) - \sum_{i=1}^{N} \frac{(\ln x_i - \mu)^2}{2\sigma^2} - \sum_{i=1}^{N} \ln x_i \right\}. & (3.7) \end{aligned}$$

Taking the derivative with respect to μ and setting equal to zero gives

$$\frac{\partial}{\partial \mu} \ln(f(\mathbf{x}; \mu, \sigma)) = \sum_{i=1}^{N} \frac{\ln x_i - N\mu}{\sigma^2} = 0 \tag{3.8}$$

$$\hat{\mu} = \sum_{i=1}^{N} \frac{\ln x_i}{N}. \tag{3.9}$$

Repeating this procedure for σ yields

$$\frac{\partial}{\partial \sigma} \ln(f(\mathbf{x}; \mu, \sigma)) = \sum_{i=1}^{N} \frac{(\ln x_i - \mu)^2}{2\sigma^4} - \frac{N}{2\sigma^2} = 0 \tag{3.10}$$

$$\hat{\sigma^2} = \frac{\sum_{i=1}^{N} (\ln x_i - \mu)^2}{N}. \tag{3.11}$$

The gamma probability density function is given by

$$p(z) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\theta^k \Gamma(k)}, \tag{3.12}$$

where k is the shape parameter and θ is the scale parameter. The parameters of the gamma distribution are found by equating the sample moments to the moments of the distribution.

$$m_1 = \sum_{i=1}^{N} x_i = E[x] = k\theta \tag{3.13}$$

$$m_2 = \sum_{i=1}^{N} x_i^2 = E[x^2] = k\theta^2(1+k) \tag{3.14}$$

Solving these equations yields the estimates:

$$\hat{k} = \frac{m_1^2}{m_2 - m_1^2} \tag{3.15}$$

$$\hat{\theta} = \frac{m_2 - m_1^2}{m_1} \tag{3.16}$$

As demonstrated in Figure 3-4, the data are best characterized by the gamma distribution. The corresponding shape and scale parameters for each epoch are given in Table 3.1.

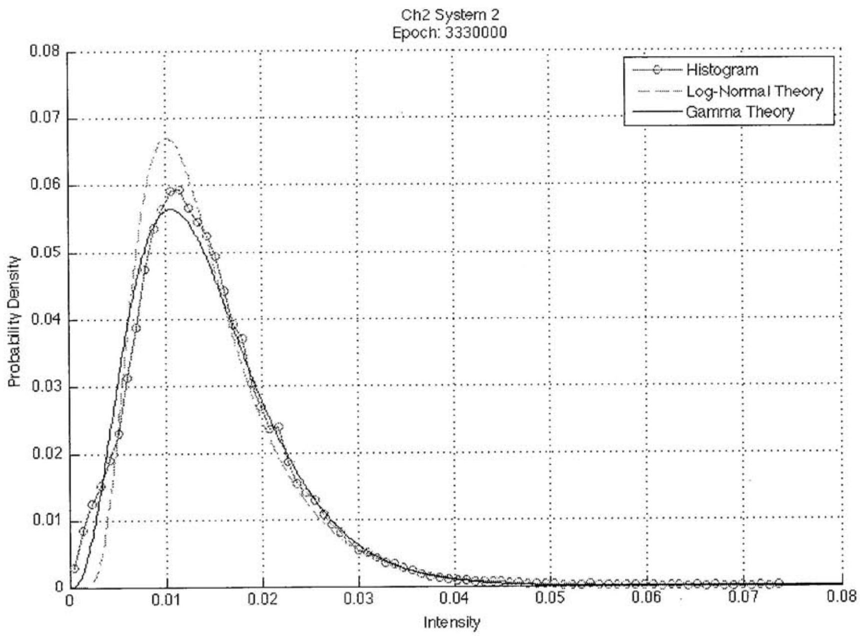


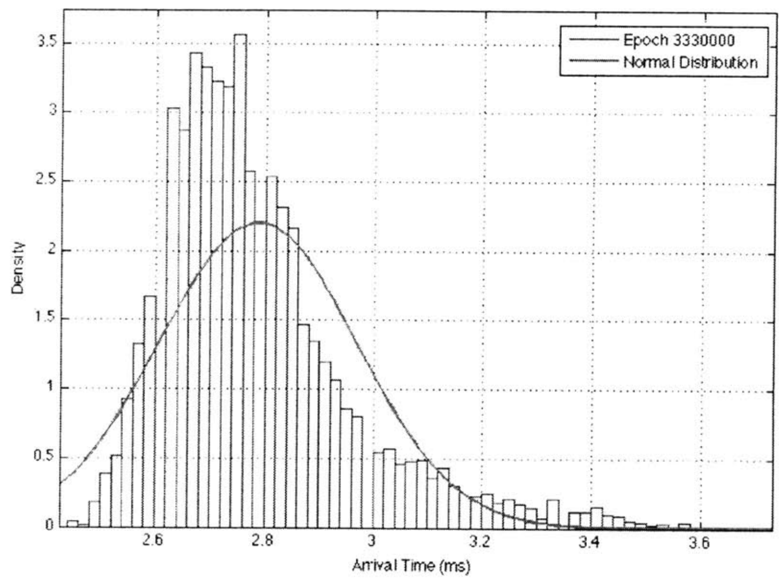
Figure 3-4: "Peak amplitude" histogram: 80 meters, epoch 3330000

Table 3.1: Gamma "peak" amplitude parameters: 80 meters

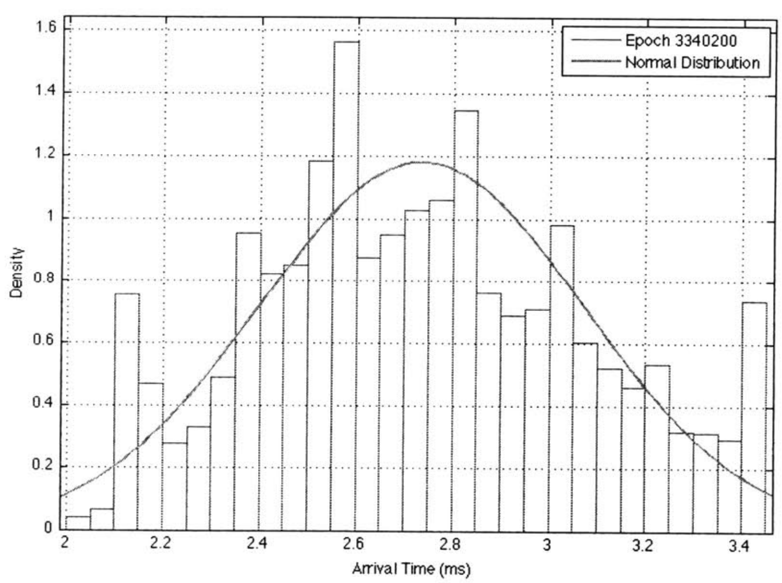
Epoch	k (shape)	θ (scale)
3310000	6.3841	0.0033
3330000	3.7900	0.0038
3330200	5.0211	0.0033
3330400	5.0992	0.0035
3331600	2.8226	0.0044
3331800	3.8021	0.0032
3340200	3.6120	0.0026
3341200	4.8721	0.0037
3341400	3.3334	0.0040
3370000	5.1042	0.0034
3370200	4.7741	0.0039
3370800	5.0716	0.0039
3371400	4.6702	0.0033

3.2.2 Arrival Time Fluctuations

The time in delay at which the peak amplitude arrives is fluctuating due to the motion of the sea surface. This fluctuation is highly correlated and even appears somewhat sinusoidal. However, for simplicity the arrival times were analyzed as independent realizations of a random variable. Figure 3-5 contains histograms of the arrival times for epoch 3330000 along with normal distributions fit to the data using maximum likelihood estimates for the mean and variance. This is not a particularly good fit, however it will suffice for the purpose of the simulation. The corresponding parameters are given in Table 3.2.



(a)



(b)

Figure 3-5: Arrival time histograms: 80 meters

Table 3.2: Arrival time parameters: 80 meters

Epoch	$\mu_\mathcal{T}$	$\sigma_\mathcal{T}$
3330000	2.78746	0.18113
3331600	2.93277	0.235705
3340200	2.73522	0.00354509

3.3 Simulation Results

The scattering process was simulated using the model in (3.3), with the Matlab code found in Appendix B. The simulations presented in the following section were run with statistics chosen to represent epoch 3330000, such that the A_i were gamma distributed with $k = 3.79$ and $\theta = 0.0038$. A histogram was created for the simulated data using 9149 simulated pulses and averaging the statistics over 0.1120 ms in delay (5 samples). For the simulation in 3.3, the number of scatterers is set to one. This simulation clearly does not represent the behavior of the actual system. However, the resulting simulation histogram bears some resemblance to a gamma distribution. When used as the step-size statistic in the random walk model discussed earlier, the gamma distribution is known to produce a K-distributed envelope [1]. This suggests that the summation of these variables might produce a histogram which resembles the K-distribution.

Figure 3.3 contains the histogram from a simulation with N=6. The arrival times were formed such that τ_1 was normally distributed with $\mu_\mathcal{T} = 2.787$ and $\sigma_\mathcal{T} = 0.18113$. For all $i > 1$, the arrival times were given by an independent interval processes such that $\tau_i = \tau_{i-1} + \Delta\tau_i$, where the $\Delta\tau_i$ were uniformly distributed. For this simulation, $\Delta\tau_i$ was uniformly distributed between 0 and 0.3136 ms. The simulated histogram fits very closely with the actual histogram. This suggests that the dynamics assumed in the simulation model result in statistics which mimic that of the experimental data.

The model was analyzed for its sensitivity to changes in the parameters. For this test, each parameter was varied while the others were held constant in order to observe the change in mean square error associated with a fractional change in the parameter. The results are displayed in Figure 3-8. The parameters with the largest influence on error are the interval range of the independent interval process and the number of scatterers. The shape parameter of the amplitude has a relatively significant impact on the resulting mean

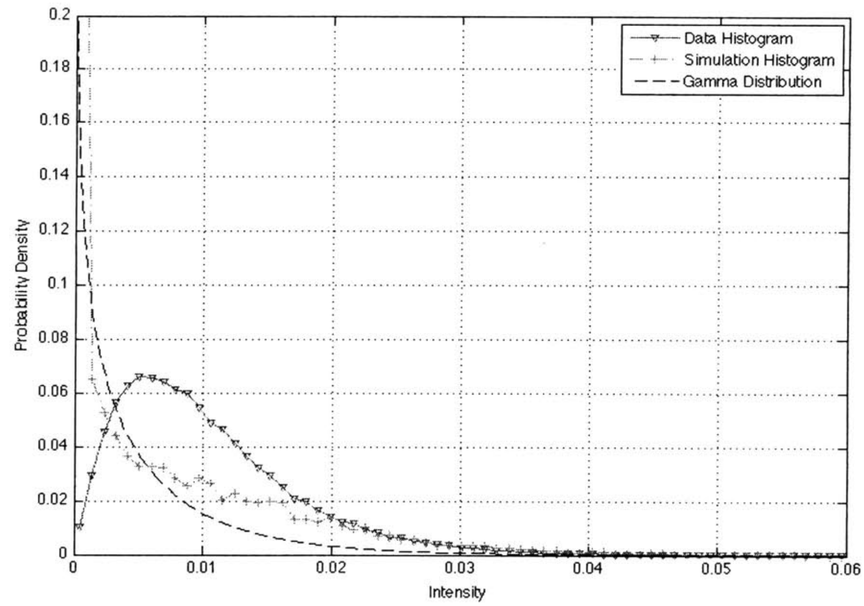


Figure 3-6: Scatterer simulation histogram with N=1 scatterer

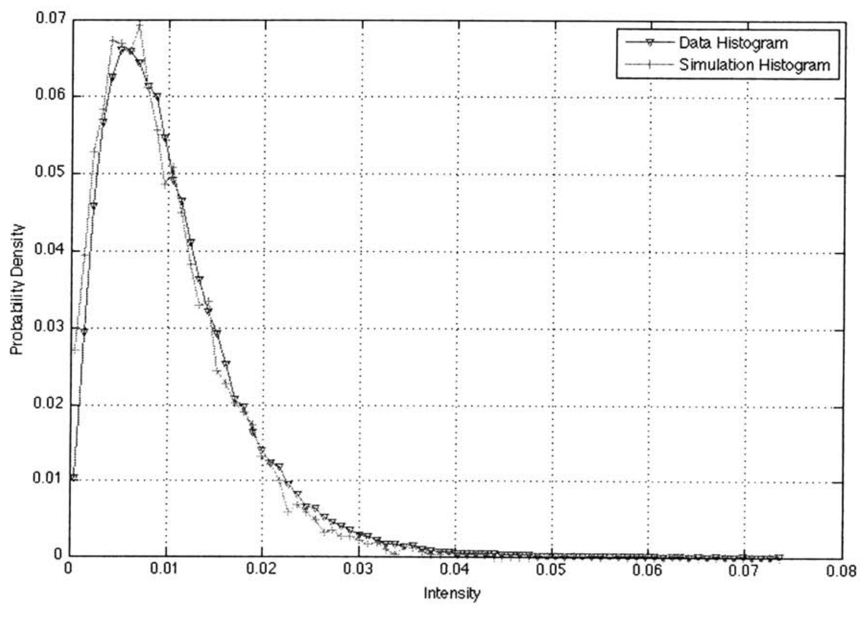


Figure 3-7: Scatterer simulation histogram with N=6 scatterers

square error, however the scale parameter does not. The standard deviation of the initial arrival time also has an impact on the error, while its mean has no apparent influence. The latter is expected, since a change in mean simply changes the location of the arrivals.

These results imply that this model could provide a reasonable representation of the

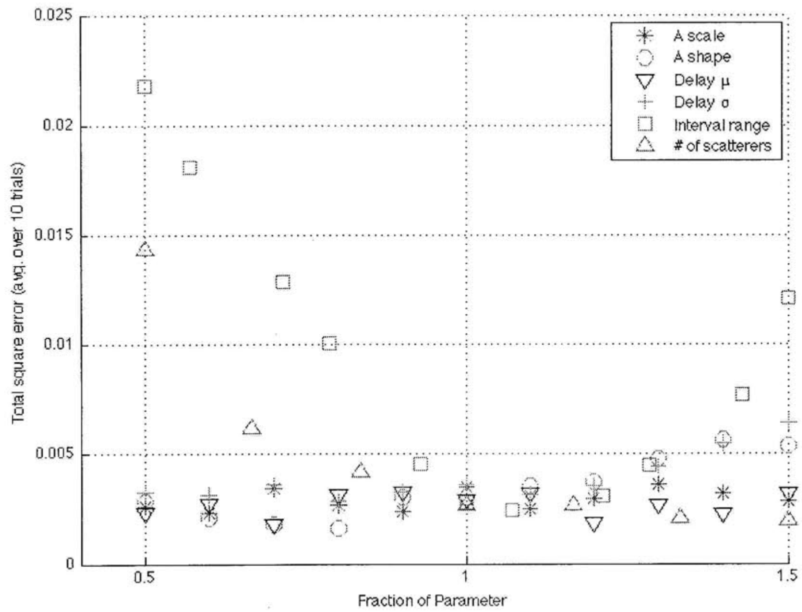


Figure 3-8: Scatterer model sensitivity to change in parameters

scattering statistics. However, it is not a realistic model for several reasons. Perhaps most importantly, it does not take into account the correlation structure of the arrival times. Furthermore, the independent interval process and the number of scatterers were chosen by trial and error and consequently have no physical justification. The parameters that ultimately provided the matching histogram of Figure 3.3 were not unique, and by tuning both parameters simultaneously other acceptable matches could be found. Additional matches could be produced by using other distributions to represent the independent interval process.

Chapter 4

Bayesian Estimation in K-Distributed Fading Models

4.1 Introduction

The dynamics of the ocean surface cause the underwater acoustic communication channel to fluctuate rapidly and suffer from a significant delay spread. For effective communication, the time-varying channel impulse response must be both estimated and tracked. Channel estimation errors will degrade the performance of any communication system, and it is therefore our goal to reduce these errors whenever possible.

Estimation problems are typically approached from either a classical or Bayesian viewpoint. While classical estimation assumes that the unknown parameters are deterministic but unknown, Bayesian estimation seeks to estimate a particular realization of a random variable. The Bayesian approach uses a-priori statistics of the parameters to formulate estimates optimized for a given cost criterion. If these statistical models are accurate, Bayesian estimation offers an improvement over classical methods, which do not consider any a-priori statistics.

Most channel estimation techniques in use today assume that the underwater acoustic channel exhibits Rayleigh or Rician fading behavior. However, as was demonstrated in Chapter 2, this is not always accurate. The question remains: what, if any, performance gains can be achieved by incorporating K-distribution fading models into channel estimation algorithms?

4.1.1 The Complex Bayesian Linear Channel Model

The complex Bayesian linear channel model will be used throughout this work for the analysis of estimator performance. The N x 1 output vector $\mathbf{y}$ is given by

$$\mathbf{y} = \mathbf{D}\mathbf{h} + \mathbf{w}, \tag{4.1}$$

where $\mathbf{D}$ is a known N x M matrix of transmitted data values, $\mathbf{h}$ is the channel impulse response, and $\mathbf{w}$ is a zero-mean, complex Guassian noise vector, which is independent of $\mathbf{h}$.

In the standard model, $\mathbf{h}$ is generally assumed to be a complex Guassian random process with mean $\boldsymbol{\mu}_{\boldsymbol{h}}$ and covariance matrix $\mathbf{R}_{\mathbf{h}}$. With this assumption, the envelope of the channel response is Rician distributed (or Rayleigh distributed for a zero-mean process). This work will explore the implications of assuming that the channel envelope is K-distributed.

4.1.2 Bayesian Estimators

Minimum Mean Square Error Estimation

It is often desirable to minimize the mean square error (MSE) of the estimate.

$$mse(\hat{\mathbf{h}}) = E[|\mathbf{h} - \hat{\mathbf{h}}|^2] \tag{4.2}$$

The estimate that minimizes this cost function is the mean of the posterior distribution and is known as the Minimum Mean Square Error Estimate (MMSE).

$$\hat{\mathbf{h}}_{\mathbf{MMSE}} = E[\mathbf{h}|\mathbf{y}] \tag{4.3}$$

Consequently, the minimum MSE obtained by this estimate is given by the conditional variance.

$$mse(\hat{\mathbf{h}}) = Var(\mathbf{h}|\mathbf{y}) \tag{4.4}$$

Maximum A Posteriori Estimation

A popular alternative to the MMSE estimate is the Maximum A Posteriori (MAP) estimate. Although this estimate may lead to a larger MSE than the former, it is often simpler to compute. The MAP estimate is the mode of the posterior distribution.

$$\hat{\mathbf{h}}_{MAP}(\mathbf{y}) = \arg\max_{\mathbf{h}} p_{\mathbf{h}|\mathbf{y}}(\mathbf{h}|\mathbf{y}) \tag{4.5}$$

This is equivalent to maximizing the logarithm of the posterior distribution, which is often easier to compute.

$$\hat{\mathbf{h}}_{MAP}(\mathbf{y}) = \arg\max_{\mathbf{h}} \left\{ \frac{p(\mathbf{y}|\mathbf{h})p(\mathbf{h})}{p(\mathbf{y})} \right\} \tag{4.6}$$

$$= \arg\max_{\mathbf{h}} \{\ln p(\mathbf{y}|\mathbf{h}) + \ln p(\mathbf{h})\} \tag{4.7}$$

$$= \arg\max_{\mathbf{h}} \{J(\mathbf{y}, \mathbf{h})\} \tag{4.8}$$

The problem reduces to maximizing the objective function, $J(\mathbf{y}, \mathbf{h})$.

4.1.3 Optimization Methods

The maximizations in (4.5) and (4.8) sometimes lead to expressions that are difficult, or impossible, to compute in closed form. In this case, standard numerical optimization methods can be applied. Although there are many methods available, this work will consider only Coordinate Descent and Newton's Method.

Coordinate Descent

Coordinate descent is a simple descent algorithm for finding extrema of a convex objective function. The function is iteratively optimized along each coordinate axis. For example, a two dimensional function of x and y could be iteratively minimized by the following algorithm:

- Initialize $x^k = x_0$, $y^k = y_0$
- $x^{k+1} = \arg\min_x f(x, y^k)$
- $y^{k+1} = \arg\min_y f(x^{k+1}, y)$

This is naturally extended to convex functions of higher dimension. Coordinate descent converges linearly to the optimal solution. A similar approach can be applied to vectors variables $\mathbf{x}$ and $\mathbf{y}$, and will be termed the Grouped Coordinate Descent. The Grouped Coordinate Descent also converges linearly in both variables[2].

Newton's Method

Newton's Method is a hugely successful iterative algorithm for minimization and maximization. It is well known for its very fast convergence rate, which is locally quadratic. At each iteration, the estimate is advanced in a direction given by the *Newton Step*, $\Delta\mathbf{h}$.

$$\boldsymbol{\Delta}\mathbf{h} \triangleq -\boldsymbol{\nabla}^2 f(\mathbf{h})^{-1}\boldsymbol{\nabla} f(\mathbf{h}) \tag{4.9}$$

A line search (either exact or inexact) is conducted to find the optimal step length. The *Newton Decrement*, $\lambda(\mathbf{h})$, provides a measure of the estimate's proximity to the optimal solution and is used as a stopping criterion in the algorithm.

$$\lambda(\mathbf{h}) \triangleq (\boldsymbol{\nabla} f(\mathbf{h})^T \boldsymbol{\nabla}^2 f(\mathbf{h})^{-1}\boldsymbol{\nabla} f(\mathbf{h}))^{1/2} \tag{4.10}$$

The method can be summarized as follows:

- Initialize $\mathbf{h} = \mathbf{h_0} \in \mathrm{dom} f(\mathbf{h})$, choose tolerance $\epsilon > 0$
- Compute $\boldsymbol{\Delta}\mathbf{h}$ and λ^2
- Quit if $\lambda^2 \leq \epsilon$
- Line search for t
- Update $\mathbf{h} := \mathbf{h} + t\boldsymbol{\Delta}\mathbf{h}$

Complex Gradient and Hessian of a Real Function

In order to compute the optimal solution, the gradient and hessian of a real function, $J(\mathbf{n})$, must be computed with respect to a complex vector, $\mathbf{n}$. Although such a function is not differentiable in the traditional sense, a complex gradient operator can be used to achieve the same optimality criteria [3].

$$\nabla_n J(\mathbf{n},\mathbf{r}) \triangleq \begin{bmatrix} \frac{dJ}{dn_1^*} \\ \frac{dJ}{dn_2^*} \\ \vdots \\ \frac{dJ}{dn_M^*} \end{bmatrix} \tag{4.11}$$

Similarly, the Hessian is defined as [15]:

$$\nabla_n^2 J(\mathbf{n},\mathbf{r}) \triangleq \begin{bmatrix} \frac{dJ}{dn_1^* dn_1} & \cdots & \frac{dJ}{dn_M^* dn_1} \\ \vdots & \ddots & \vdots \\ \frac{dJ}{dn_1^* dn_M} & \cdots & \frac{dJ}{dn_M^* dn_M} \end{bmatrix}. \tag{4.12}$$

4.2 Scalar Estimation in K-distributed Fading

Suppose h is corrupted by complex Gaussian noise such that

$$y = dh + w, \tag{4.13}$$

where d is a known (possibly complex) constant, w is a complex normal random variable with variance $2\sigma_w^2$, and h is a complex random variable with a K-distributed envelope. The channel is said to exhibit K-distributed fading when the envelope of its impulse response follows a K-distribution. In channel estimation, the *complex-valued* channel tap is the parameter to be estimated, and consequently its probability density function (PDF) must be derived.

4.2.1 The PDF of a Variable with K-distributed Envelope

Let h be the complex-valued variable with an envelope, $x = |h|$, which is K-distributed. This variable can be represented in component form by the product of a zero-mean, unit variance complex Gaussian random variable and the square root of a gamma distributed variable. Equivalently, this can be interpreted as a zero-mean, complex Gaussian random variable with a random variance that follows a gamma distribution [17].

$$h = \sqrt{z}(n_R + jn_I) = rn \tag{4.14}$$

The Gamma distribution is given by

$$p(z) = \frac{z^{k-1}e^{-\frac{z}{\theta}}}{\theta^k \Gamma(k)}, \tag{4.15}$$

where $\Gamma(\cdot)$ is the Gamma function, k the shape parameter, and θ the scale parameter. The Gamma distribution has mean,

$$E[z] = k\theta, \tag{4.16}$$

and variance,

$$Var(z) = k\theta^2. \tag{4.17}$$

Both n_R and n_I follow a unit-variance, zero mean Gaussian distribution,

$$p(n_R) = \frac{1}{\sqrt{2\pi}} e^{-\frac{n_R^2}{2}}, \tag{4.18}$$

such that $p(h_R|z)$ and $p(h_I|z)$ are i.i.d Gaussian distributions with variance z.

$$p(h_R|z) = \frac{1}{\sqrt{2\pi z}} e^{-\frac{h_R^2}{2z}} \tag{4.19}$$

Thus,

$$p(h|z) = \frac{1}{2\pi z} e^{-\frac{|h|^2}{2z}}. \tag{4.20}$$

The component form in (4.14) can be shown to produce the desired K-distributed envelope by first noting that the amplitude of h is simply Rayleigh distributed when conditioned on z.

$$x = |h| \tag{4.21}$$

$$p_{x|z}(x|z) = \frac{x}{z} e^{\frac{-x^2}{2z}} \tag{4.22}$$

Using this conditional distribution to calculate the unconditional probability density function of x with $\theta = 2\alpha^2$ and $k = \nu + 1$ yields the familiar form of the K-distributed random variable.

$$p_x(x) = \int_0^\infty p_{x|z}(x|z)p_z(z)dz \tag{4.23}$$

$$= \int_0^\infty \frac{xz^{\nu-1}}{(2\alpha^2)^{\nu+1}\Gamma(\nu+1)} e^{-(\frac{z}{2\alpha^2}+\frac{x^2}{2z})}dz \tag{4.24}$$

$$= \frac{2}{\alpha\Gamma(\nu+1)}\left(\frac{x}{2\alpha}\right)^{\nu+1} K_\nu\left(\frac{x}{\alpha}\right) \tag{4.25}$$

The pdf of $r = \sqrt{z}$, which will be of use later, can be found by derived distributions. Since $\sqrt{z}$ is a monotonically increasing function, this can be done by applying the following formula:

$$p_R(r) = p_Z(r^2)\left|\frac{d}{dr}(r^2)\right| \tag{4.26}$$

$$= \frac{2r^{2k-1}e^{-\frac{r^2}{\theta}}}{\theta^k\Gamma(k)}. \tag{4.27}$$

The PDF of the complex-valued h is defined as the joint distribution of its real and imaginary parts. Defining the random variables v and u as

$$v = \sqrt{z}n_R \quad \text{and} \quad u = \sqrt{z}n_I, \tag{4.28}$$

which are independent and identically distributed when conditioned on z, the distribution of h becomes

$$p_h(h) \triangleq p_{u,v}(u, v) \tag{4.29}$$

$$= \int_0^\infty p(u|z)p(v|z)p(z)dz \tag{4.30}$$

$$= \int_0^\infty \frac{1}{2\pi z} e^{\frac{-u^2-v^2}{2z}} \frac{z^\nu e^{-\frac{z}{2\alpha^2}}}{\theta^{\nu+1}\Gamma(\nu+1)}dz \tag{4.31}$$

$$= \frac{1}{2\pi(2\alpha^2)^{\nu+1}\Gamma(\nu+1)} \int_0^\infty z^{\nu-1}e^{-\frac{|h|^2}{2z}-\frac{z}{2\alpha^2}}dz \tag{4.32}$$

$$= \frac{1}{2\pi\alpha^2\Gamma(\nu+1)}\left(\frac{|h|}{2\alpha^2}\right)^\nu K_\nu\left(\frac{|h|}{\alpha}\right) \tag{4.33}$$

4.2.2 MAP Estimation of a Scalar

Direct implementation of the MAP estimator in (4.5) requires the maximization of the function $p(y|h)p(h)$, or equivalently the maximization of

$$J(y,h) = -\frac{|y-dh|^2}{2\sigma_w^2} + \nu \ln|h| + \ln K_\nu\left(\frac{|h|}{\alpha}\right) + c, \tag{4.34}$$

where c is a constant that does not depend on y or h.

While there may exist techniques to maximize such a function, the MAP estimate can be simplified by considering the component representation in (4.14) and instead estimating the parameter γ.

$$\gamma = \begin{bmatrix} r \\ n \end{bmatrix} \tag{4.35}$$

The posterior distribution is again a complex normal distribution with mean $h = drn$.

$$p(y|\gamma) = \frac{1}{2\pi\sigma_w^2} e^{-\frac{|y-drn|^2}{2\sigma_w^2}} \tag{4.36}$$

$$= \frac{1}{2\pi\sigma_w^2} e^{-\frac{|y|^2 - rd^*n^*y - rdny^* + r^2|dn|^2}{2\sigma_w^2}} \tag{4.37}$$

Since n and r are independent, the log-posterior function becomes

$$J(\gamma, y) = \ln p(y|\gamma) + \ln p(r) + \ln p(n) \tag{4.38}$$

$$= -\frac{|y|^2 - rd^*n^*y - rdny^* + r^2|d|^2|n|^2}{2\sigma_w^2} - \frac{|n|^2}{2} + (2k-1)\ln r - \frac{r^2}{\theta} + c \tag{4.39}$$

The Jacobian of the log-posterior function is

$$\nabla J = \left(\frac{2k-1}{r} - \frac{2r}{\theta} - \frac{2r|d|^2|n|^2 - d^*n^*y - dny^*}{2\sigma_w^2}, \quad \frac{d^*ry - r^2|d|^2 n}{2\sigma_w^2} - \frac{n}{2} \right), \tag{4.40}$$

and the Hessian is

$$\nabla^2 J = \begin{pmatrix} -\frac{|d|^2|n|^2}{\sigma_w^2} - \frac{2k-1}{r^2} - \frac{2}{\theta}, & \frac{d^*y - 2r|d|^2 n}{2\sigma_w^2} \\ \frac{d^*y - 2r|d|^2 n}{2\sigma_w^2} & -\frac{r^2|d|^2}{2\sigma_w^2} - \frac{1}{2} \end{pmatrix}. \tag{4.41}$$

It follows from (4.40) that the optimal n and r satisfy the following criteria:

$$\hat{n} = \frac{d^* r y}{\sigma_w^2 + |d|^2 \hat{r}^2} \tag{4.42}$$

$$r^2 - \frac{d^* n^* y + d n y^*}{4(2\sigma_w^2 + \theta |d|^2 |n|^2)} r + \frac{2k-1}{2} = 0. \tag{4.43}$$

The estimate of n, assuming r is known, is simply the MAP estimate (also the MMSE estimate) of a complex Gaussian random process, as expected. The estimate $\hat{\gamma}$ could be found by substituting (4.42) into (4.43) and solving for r. However, as a simpler approach, the solution is found iteratively using coordinate descent (described in Section 4.1.3).

4.2.3 MMSE Estimation of a Scalar

The complicated form of the K-distributed variable leads to computationally cumbersome expressions. However, the calculation of the MMSE estimate for h can be simplified via the law of iterated expectations.

$$\hat{h}_{MMSE} = E[h|y] \tag{4.44}$$

$$= E_{z|y}\left[E[h|y,z]\right] \tag{4.45}$$

where the expectation, $E_{z|y}[\cdot]$, in (4.45) is with respect to $p(z|y)$.

The posterior distribution, $p(h|y,z)$ can be calculated using Baye's rule

$$p(h|y,z) = \frac{p(y|h,z)p(h|z)}{p(y|z)}, \tag{4.46}$$

where

$$p(y|h,z) = p(y|h) \tag{4.47}$$

$$= \frac{1}{2\pi\sigma_w^2} e^{-\frac{|y-dh|^2}{2\sigma_w^2}}. \tag{4.48}$$

The mean of the conditional posterior in (4.45) can be found without explicitly calculating the distribution by recognizing that its form is that of another complex Gaussian.

$$p(h|y,z) \sim \exp\left(-\frac{|y-dh|^2}{2\sigma_w^2}\right)\exp\left(-\frac{|h|^2}{2z}\right) \tag{4.49}$$

$$= \exp\left(-\frac{z|d|^2+\sigma_w^2}{2z\sigma_w^2}\left(|h|^2+\frac{z}{z|d|^2+\sigma_w^2}(d^*h^*y+dhy^*)-g(y,z)\right)\right) \tag{4.50}$$

$$\sim \exp\left(-\frac{z|d|^2+\sigma_w^2}{2z\sigma_w^2}\left|h-\frac{d^*z}{z|d|^2+\sigma_w^2}y\right|^2\right) \tag{4.51}$$

The conditional mean is therefore

$$E[h|y,z] = \frac{d^*zy}{z|d|^2+\sigma_w^2}. \tag{4.52}$$

This is exactly as one would anticipate, since given z, h is a complex Gaussian process. The expression derived in (4.52) is merely the Wiener filter.

Applying Baye's Rule once more yields

$$p(z|y) = \frac{p(y|z)p(z)}{p(y)} \tag{4.53}$$

$$= \frac{\frac{1}{2\pi(\sigma_w^2+|d|^2z)}\exp\left(-\frac{|y|^2}{2(\sigma_w^2+|d|^2z)}\right)\frac{z^{k-1}e^{-\frac{z}{\theta}}}{\theta^k\Gamma(k)}}{\int_0^\infty \frac{1}{2\pi(\sigma_w^2+|d|^2z)}\exp\left(-\frac{|y|^2}{2(\sigma_w^2+|d|^2z)}\right)\frac{z^{k-1}e^{-\frac{z}{\theta}}}{\theta^k\Gamma(k)}dz} \tag{4.54}$$

$$= \frac{\frac{z^{k-1}}{z+\eta}\exp\left(-\frac{\beta}{z+\eta}-\frac{z}{\theta}\right)}{\int_0^\infty \frac{z^{k-1}}{z+\eta}\exp\left(-\frac{\beta}{z+\eta}-\frac{z}{\theta}\right)dz}, \tag{4.55}$$

where $\eta = \sigma_w^2/|d|^2$ and $\beta = |y|^2/2|d|^2$. The estimate becomes

$$\hat{h}_{MMSE} = \frac{y}{d}E_{z|y}\left[\frac{z}{z+\eta}\right] \tag{4.56}$$

$$= \frac{y}{d}\frac{\int_0^\infty \frac{z^k}{(z+\eta)^2}\exp\left(-\frac{\beta}{z+\eta}-\frac{z}{\theta}\right)dz}{\int_0^\infty \frac{z^{k-1}}{z+\eta}\exp\left(-\frac{\beta}{z+\eta}-\frac{z}{\theta}\right)dz}. \tag{4.57}$$

The value of η can be interpreted as inverse SNR, scaled by the variance of h. It represents the ratio of energy in the noise to energy in the transmitted signal. The value of β is the ratio of total instantaneous energy in the received signal to energy in the transmitted signal. Defining the integral in the numerator of (4.57) as $\zeta(\beta,\eta;k,\theta)$, we obtain

$$\zeta(\beta,\eta;k,\theta) \triangleq \int_0^\infty \frac{z^k}{(z+\eta)^2} e^{-\frac{\beta}{z+\eta}} e^{-\frac{z}{\theta}} dz \tag{4.58}$$

$$= -\frac{1}{\beta}\int_0^\infty e^{-z/\theta} e^{-\frac{\beta}{z+\eta}} \left[kz^{k-1} - \frac{1}{\theta}z^k\right] dz. \tag{4.59}$$

The expression in (4.59) is found by applying integration by parts:

$$\int_0^\infty g'(z)f(z)dz = \lim_{c\to\infty}\Big[f(z)g(z)\Big]_0^c - \int_0^\infty f'(z)g(z)dz, \tag{4.60}$$

with

$$g'(z) = \frac{1}{(z+\eta)^2} e^{-\frac{\beta}{z+\eta}} \tag{4.61}$$

$$f(z) = z^k e^{-\frac{z}{\theta}}. \tag{4.62}$$

Assuming the energy of the transmitted signal is much greater than the energy of the noise, a 2^{nd} order Taylor series expansion about $\eta = 0$ is a reasonable approximation.

$$e^{-\frac{\beta}{z+\eta}} \approx e^{\frac{-\beta}{z}}\left[1 + \frac{\beta\eta}{z^2}\right] \tag{4.63}$$

Substituting this into (4.59) yields

$$\zeta(\beta,\eta;k,\theta) \approx \frac{1}{\beta}\int_0^\infty e^{-z/\theta-\beta/z}\left[\frac{1}{\theta}z^k - kz^{k-1} + \frac{\beta\eta}{\theta}z^{k-2} - k\beta\eta z^{k-3}\right] dz. \tag{4.64}$$

Since β and θ are both guaranteed to be greater than zero, this integral can be computed as [5]:

$$\begin{aligned}\zeta(\beta,\eta;k,\theta) \approx 2(\beta\theta)^{k/2}\Bigg[&\frac{\sqrt{\beta\theta}}{\beta\theta}K_{k+1}\left(2\sqrt{\beta/\theta}\right) - \frac{k}{\beta}K_k\left(2\sqrt{\beta/\theta}\right) \\ &+\frac{\eta}{\theta\sqrt{\beta\theta}}K_{k-1}\left(2\sqrt{\beta/\theta}\right) - \frac{k\eta}{\beta\theta}K_{k-2}\left(2\sqrt{\beta/\theta}\right)\Bigg].\end{aligned} \tag{4.65}$$

The function, $K_p(\cdot)$, is the p^{th} order modified Bessel function of the 2^{nd} kind. The integral in the denominator of (4.57) can also be written in terms of the function $\zeta(\beta, \eta; k, \theta)$:

$$\int_0^\infty \frac{z^{k-1}}{z+\eta} e^{-\frac{\beta(y)}{z+\eta}} e^{-\frac{z}{\theta}} dz \tag{4.66}$$

$$= \int_0^\infty \frac{z^k}{(z+\eta)^2} e^{-\frac{\beta(y)}{z+\eta}} e^{-\frac{z}{\theta}} dz + \eta \int_0^\infty \frac{z^{k-1}}{(z+\eta)^2} e^{-\frac{\beta(y)}{z+\eta}} e^{-\frac{z}{\theta}} dz \tag{4.67}$$

$$= \zeta(\beta, \eta; k, \theta) + \eta\zeta(\beta, \eta; k-1, \theta). \tag{4.68}$$

Finally, the MMSE estimate of h is

$$\hat{h}_{MMSE} = \frac{y}{d} \frac{\zeta(\beta, \eta; k, \theta)}{\zeta(\beta, \eta; k, \theta) + \eta\zeta(\beta, \eta; k-1, \theta)}. \tag{4.69}$$

4.2.4 Results

The MAP and MMSE estimators derived above were evaluated in Matlab with the simulation in Appendix C. Realizations of the random variable h were generated using the component form in (4.14). The variance of h was held constant throughout each simulation, while the variance of the noise was adjusted to evaluate the performance of each estimator as a function of the Signal to Noise Ratio (SNR). For the purpose of these simulations, the value of the input parameter, d, was set to one. The MSE of each estimator was approximated by a sample average over 2000 trials.

The K-distribution approaches a Rayleigh distribution as $\nu \to \infty$. Likewise, the largest discrepancies between the two distributions occur as $\nu \to -1$. The smallest value of the shape parameter observed in Table 2.4(a) was $\nu = -0.3447$, accompanied by a scale parameter of $\alpha = 0.0062$. Since the Rayleigh parameter estimates are based on second order statistics, the value of the Rayleigh σ parameter was chosen such that the variance of the two distributions were equivalent. The corresponding distributions that were used in the channel simulation are shown in Figure 4-1.

The K-distribution MAP estimate was made using the coordinate descent outlined in Section 4.1.3. The MMSE estimate was implemented with both a first and second order Taylor approximation. The Raleigh MMSE estimate is also its MAP estimate, as well as the Linear Minimum Mean Square Error (LMMSE) estimate of the K-distribution. These are compared with the error obtained by taking the output y as the estimate of h. The resulting sample MSEs are plotted in Figure 4-4.

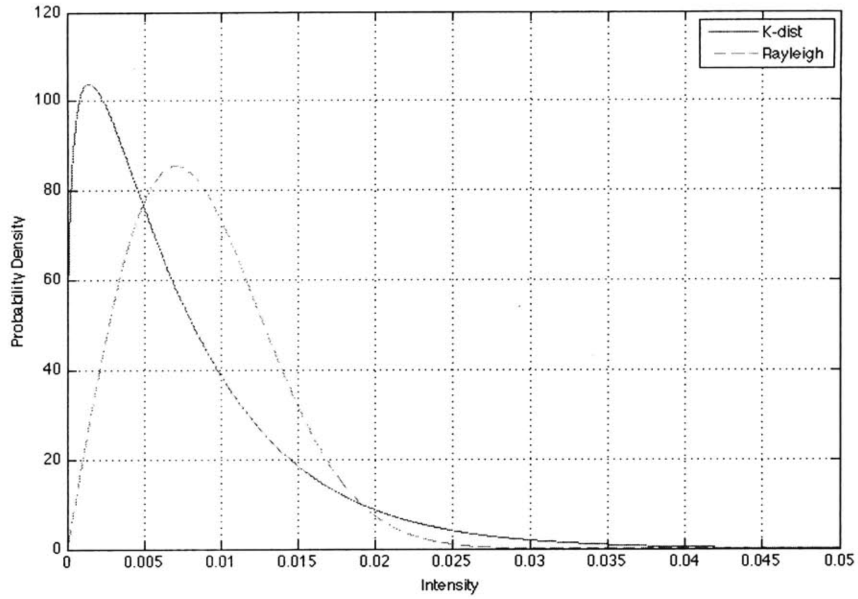


Figure 4-1: Envelope PDFs used in channel simulations for $\nu = -0.3447$, $\alpha = 0.0062$

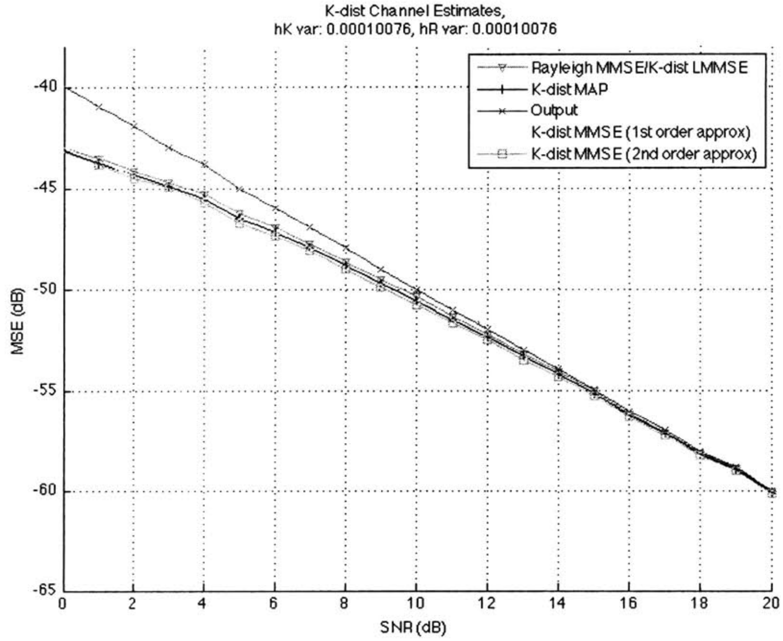


Figure 4-2: K-distribution channel estimate MSE: $\nu = -0.3447$, $\alpha = 0.0062$

The results are displayed in Figure 4-4. The 2nd order K-distribution MMSE estimate outperforms all other estimates, while the 1st order approximation performs comparably, except at very low SNR. A maximum improvement in MSE of 0.5 dB is observed using the 2nd order K-distribution MMSE in place of the Rayleigh estimate. The MAP estimate offers an improvement of 0.26 dB over the Rayleigh estimate.

Although there is some advantage in utilizing the K-distribution for the estimation, it is not incredibly significant. In order to illustrate a maximal performance gain obtainable by using the K-distribution in place of the Rayleigh model, we will consider extreme parameters that produce a larger disparity between the two distributions. The distributions in Figure 4-3 are obtained with $\nu = -0.95$, and α chosen such that the variance is the same as those in Figure 4-1.

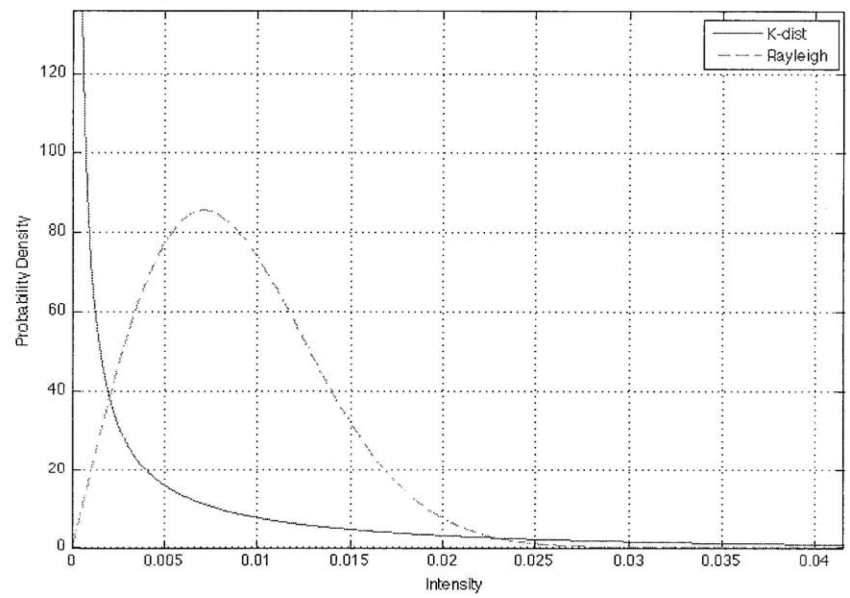


Figure 4-3: Envelope PDFs used in channel simulations for $\nu = -0.95$, $\alpha = 0.0224$

The results for this model are shown in Figure 4-3. This time, the K-distribution estimates perform significantly better than the Rayleigh estimate. The 2nd order K-distribution MMSE performs substantially better than the others, while the 1st order MMSE and the MAP estimates have similar performance. A maximum improvement in MSE of 3.64 dB is obtained by the 2nd order MMSE and 2.30 dB by the MAP estimate.

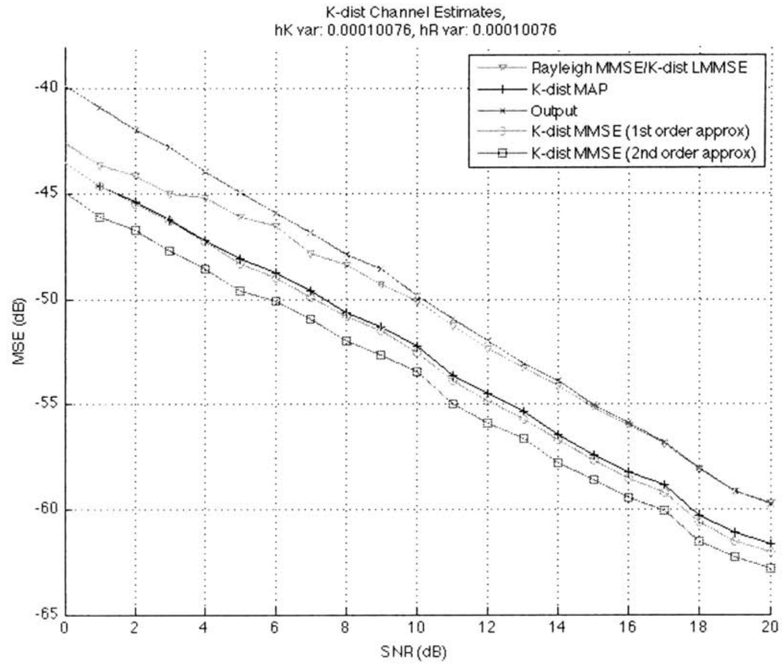


Figure 4-4: K-distribution channel estimate MSE: $\nu = -0.95$, $\alpha = 0.0224$

4.3 Vector Estimation in K-distributed Fading

Now suppose the variable to be estimated is a vector of independent complex random variables with K-distributed envelopes. The received signal is given by

$$\mathbf{y} = \mathbf{D}\mathbf{h} + \mathbf{w}. \tag{4.70}$$

In this representation $\mathbf{D}$ is a known N x M matrix of transmitted data values, $\mathbf{w}$ is a N x 1 complex normal random vector, and each element of the M x 1 vector $\mathbf{h}$ is a complex random variable with a K-distributed envelope. In component form, this can be represented as

$$\mathbf{y} = \mathbf{D}\mathbf{R}\mathbf{n} + \mathbf{w} \tag{4.71}$$

$$= \mathbf{D}\mathbf{N}\mathbf{r} + \mathbf{w}, \tag{4.72}$$

where

$$\mathbf{R} = \begin{bmatrix} r_1 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_M \end{bmatrix} \quad \text{and} \quad \mathbf{N} = \begin{bmatrix} n_1 & 0 & 0 & 0 \\ 0 & n_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n_M \end{bmatrix}. \tag{4.73}$$

For the vector $\mathbf{h}$ with independent components the joint distribution becomes

$$p_{\mathbf{h}}(\mathbf{h}) = \prod_{i=1}^{M} \frac{1}{2\pi\alpha_i^2\Gamma(\nu_i+1)} \left(\frac{|\tilde{h}_i|}{2\alpha_i^2}\right)^{\nu_i} K_{\nu_i}\left(\frac{|\tilde{h}_i|}{\alpha_i}\right). \tag{4.74}$$

4.3.1 MAP Estimation of a Vector

The new parameter, γ, contains the parameterization from (4.14) for each element of $\tilde{\mathbf{h}}$.

$$\gamma = \begin{bmatrix} \mathbf{r} \\ \tilde{\mathbf{n}} \end{bmatrix} \tag{4.75}$$

Its MAP estimate can be found by maximizing the log-posterior function.

$$\hat{\gamma} = \arg\max_{\gamma}\{\ln p(\mathbf{y}|\tilde{\mathbf{h}}(\gamma)) + \ln p(\mathbf{r}) + \ln p(\tilde{\mathbf{n}})\} \tag{4.76}$$

Defining the moment matrix of the noise vector as half the covariance matrix,

$$\Lambda_w = \frac{1}{2}E[\mathbf{w}\mathbf{w}^H], \tag{4.77}$$

the distributions in the vector case become:

$$p(\mathbf{w}) = \frac{\exp(-\frac{1}{2}\mathbf{w}^H \mathbf{\Lambda_w}^{-1}\mathbf{w})}{(2\pi)^M \det(\mathbf{\Lambda_w})} \tag{4.78}$$

$$p(\mathbf{n}) = \frac{\exp(-\frac{1}{2}\mathbf{n}^H \mathbf{n})}{(2\pi)^M} \tag{4.79}$$

$$p(\mathbf{r}) = \prod_{i=1}^{M} \frac{2r_i^{2k-1} e^{-\frac{r_i^2}{\theta}}}{\theta^k \Gamma(k)} \tag{4.80}$$

$$= \left(\frac{2}{\theta^k \Gamma(k)}\right)^M \exp\left(-\frac{1}{\theta}\mathbf{r}^T\mathbf{r}\right) \prod_{i=1}^{M} r_i^{2k-1} \tag{4.81}$$

$$p(\mathbf{y}|\mathbf{h}(\gamma)) = \frac{\exp(-\frac{1}{2}(\mathbf{y}-\mathbf{Dh})^H \mathbf{\Lambda_w}^{-1}(\mathbf{y}-\mathbf{Dh}))}{(2\pi)^M \det(\mathbf{\Lambda_w})} \tag{4.82}$$

Subsituting these into (4.76) yields the objective function for the vector case,

$$\hat{\gamma} = \arg\max_{\gamma} \left\{ -\frac{1}{2}(\mathbf{y}-\mathbf{DRn})^H \mathbf{\Lambda_w}^{-1}(\mathbf{y}-\mathbf{DRn}) - \frac{1}{\theta}\mathbf{r}^T\mathbf{r} + (2k-1)\sum_{i=1}^{M} \ln r_i - \frac{1}{2}\mathbf{n}^H\mathbf{n} \right\} \tag{4.83}$$

$$= \arg\max_{\gamma} \{J(\mathbf{r},\mathbf{n})\} \tag{4.84}$$

In order to compute the optimal solution, the gradient is computed with respect to $\mathbf{r}$ and $\mathbf{n}$. Although $\mathbf{r}$ is a real variable, $J(\mathbf{r},\mathbf{n})$ is also a real function of the complex-valued variable $\mathbf{n}$. Therefore, the complex gradient operator, described in Section 4.1.3, must be applied. Using this definition of the gradient,

$$\nabla_{\mathbf{n}} J(\mathbf{n},\mathbf{r}) = -\frac{1}{2}\mathbf{R}^H\mathbf{D}^H\mathbf{\Lambda_w}^{-1}(\mathbf{DRn}-\mathbf{y}) - \frac{1}{2}\mathbf{n}. \tag{4.85}$$

Setting (4.85) to zero, the optimal $\mathbf{n}$ will satisfy

$$\hat{\mathbf{n}} = \left(\mathbf{R}^H\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{DR} + \mathbf{I}\right)^{-1}\mathbf{R}^H\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{y}. \tag{4.86}$$

The gradient with respect to $\mathbf{r}$ yields

$$\nabla_{\mathbf{r}} J(\mathbf{n},\mathbf{r}) = Re\{\mathbf{N}^H\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{y}\} - \left(Re\{\mathbf{N}^H\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{DN}\} + \frac{2}{\theta}\mathbf{I}\right)\mathbf{r} - (2k-1)\mathbf{r_I}, \tag{4.87}$$

where $\mathbf{r_I}$ is defined as

$$\mathbf{r_I} = \begin{bmatrix} 1/r_1 \\ 1/r_2 \\ \vdots \\ 1/r_M \end{bmatrix}. \tag{4.88}$$

The set of multivariate polynomial equations resulting from (4.87) is non-trivial to solve. Consequently, the coordinate descent approach as applied in the scalar case will not suffice. Instead, an inexact maximization via Newton's Method is applied to optimize $\mathbf{r}$ in the coordinate descent. Even with this additional step, local linear convergence is preserved. In fact, only a small number of Newton Method iterations are necessary [6]. To apply Newton's Method, the Hessian with respect to $\mathbf{r}$ is required.

$$\nabla^2_{\mathbf{r}} J(\mathbf{n}, \mathbf{r}) = Re\{\mathbf{N}^H \mathbf{D}^H \mathbf{\Lambda_w}^{-1} \mathbf{D} \mathbf{N}\} - \frac{2}{\theta}\mathbf{I} + (1-2k)\mathbf{R}^{-1}\mathbf{R}^{-1} \tag{4.89}$$

Combining coordinate descent with Newton's method for $\mathbf{r}$ results in the "Grouped Coordinate Descent with Newton's Method" described bellow. This method was implemented by Bezdek [6] and shown to preserve the linear convergence of the traditional coordinate descent with only one or two Newton iterations (P=1 or 2).

Grouped Coordinate Descent with Newton's Method

- Initialize $\mathbf{r}_k = \mathbf{r}_0$, $\mathbf{r} > 0$, choose tolerance $\epsilon > 0$
- $\mathbf{n}_{k+1} = \left(\mathbf{R}_k^H \mathbf{D}^H \mathbf{\Lambda_w}^{-1} \mathbf{D} \mathbf{R}_k + \mathbf{I}\right)^{-1} \mathbf{R}_k^H \mathbf{D}^H \mathbf{\Lambda_w}^{-1} \mathbf{y}$
- Set $\mathbf{r}_p = \mathbf{r}_k$. While $\lambda^2 \leq \epsilon$ or $p \leq P$, Do:
  - ◇ Compute $\mathbf{\Delta r} = -\left(\nabla^2 J(\mathbf{n}_{k+1}, \mathbf{r}_p)\right)^{-1} \left(\nabla J(\mathbf{n}_{k+1}, \mathbf{r}_p)\right)$
  - ◇ Compute $\lambda^2 = \left(\nabla J(\mathbf{n}_{k+1}, \mathbf{r}_p)\right)^T \left(\nabla^2 J(\mathbf{n}_{k+1}, \mathbf{r}_p)\right)^{-1} \left(\nabla J(\mathbf{n}_{k+1}, \mathbf{r}_p)\right)$
  - ◇ Line search for t
  - ◇ Update $\mathbf{r}_{p+1} := \mathbf{r}_p + t\mathbf{\Delta r}$
- Update $\mathbf{r}_{k+1} = \mathbf{r}_P$

4.3.2 MMSE Estimation of a Vector

Following the same method used in the scalar estimation problem, the MMSE estimate of the vector $\mathbf{h}$ is

$$\mathbf{h}_{\mathbf{MMSE}} = E[\mathbf{h}|\mathbf{y}] \tag{4.90}$$

$$= E_{\mathbf{z}|\mathbf{y}}\left[E[\mathbf{h}|\mathbf{y},\mathbf{z}]\right], \tag{4.91}$$

where the conditional posterior distribution is now:

$$p(\mathbf{h}|\mathbf{y},\mathbf{z}) = \frac{p(\mathbf{y}|\mathbf{h},\mathbf{z})p(\mathbf{h}|\mathbf{z})}{p(\mathbf{y}|\mathbf{z})} \tag{4.92}$$

$$= \frac{\exp(-\frac{1}{2}(\mathbf{y}-\mathbf{Dh})^H\mathbf{\Lambda_w}^{-1}(\mathbf{y}-\mathbf{Dh}) - \frac{1}{2}\mathbf{h}^H\mathbf{\Gamma_z}^{-1}\mathbf{h})}{(2\pi)^{2M}\det(\mathbf{\Lambda_w})\det(\mathbf{\Gamma_z})p(\mathbf{y}|\mathbf{z})} \tag{4.93}$$

$$= \frac{\exp(\frac{1}{2}(-\mathbf{y}^H\mathbf{\Lambda_w}^{-1}\mathbf{y} + \mathbf{y}^H\mathbf{\Lambda_w}^{-1}\mathbf{Dh} + \mathbf{h}^H\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{y} - \mathbf{h}^H(\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{D} + \mathbf{\Gamma_z}^{-1})\mathbf{h}))}{(2\pi)^{2M}\det(\mathbf{\Lambda_w})\det(\mathbf{\Gamma_z})p(\mathbf{y}|\mathbf{z})}. \tag{4.94}$$

Again, this is recognized as a complex normal distribution,

$$p(\mathbf{h}|\mathbf{y},\mathbf{z}) \sim \mathcal{CN}((\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{D} + \mathbf{\Gamma_z}^{-1})^{-1}\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{y}, 2(\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{D} + \mathbf{\Gamma_z}^{-1})^{-1}), \tag{4.95}$$

from which the MMSE estimate becomes

$$\hat{\mathbf{h}}_{MMSE} = E_{\mathbf{z}|\mathbf{y}}\left[(\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{D} + \mathbf{\Gamma_z}^{-1})^{-1}\mathbf{D}^H\mathbf{\Lambda_w}^{-1}\mathbf{y}\right] \tag{4.96}$$

$$= E_{\mathbf{z}|\mathbf{y}}\left[\mathbf{\Gamma_z}\mathbf{D}^H(\mathbf{D}\mathbf{\Gamma_z}\mathbf{D}^H + \mathbf{\Lambda_w})^{-1}\right]\mathbf{y}. \tag{4.97}$$

The estimate suggested by (4.97) requires a multidimensional integral of a complicated function. To avoid this, an iterative method for estimating the vector $\mathbf{h}$ will be employed. Each element, h_l, can be estimated separately, assuming the set of all other elements, $\{h_{i\neq l}\}$, is known. Once each channel tap has been estimated, the process is repeated, until some convergence criteria is met. This is an implementation of coordinate descent, where the objective function being minimized is the Bayesian MSE cost function.

For convenience, the elements of the matrix $\mathbf{D}$ will be denoted as

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1M} \\ d_{21} & d_{22} & \dots & d_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \dots & d_{NM} \end{bmatrix} = \begin{bmatrix} \mathbf{d_1}^T \\ \mathbf{d_2}^T \\ \vdots \\ \mathbf{d_N}^T \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{d}}_\mathbf{1} & \underline{\mathbf{d}}_\mathbf{2} & \dots & \underline{\mathbf{d}}_\mathbf{M} \end{bmatrix}, \tag{4.98}$$

where d_{ij} is the element of $\mathbf{D}$ in the i^{th} row and j^{th} column, $\mathbf{d}_i$ is the i^{th} row vector, and $\underline{\mathbf{d}}_i$ is the i^{th} column vector.

The MMSE estimate of h_l, assuming all other tap values are known, is

$$\hat{h}_l = E[h_l|\mathbf{y}, \{h_{i\neq l}\}] \tag{4.99}$$

$$= E_{z_l|\mathbf{y}}\left[E\left[h_l|\mathbf{y}, z_l, \{h_{i\neq l}\}\right]\right]. \tag{4.100}$$

The conditional posterior distribution is then:

$$p(h_l|\mathbf{y}, z_l, , \{h_{i\neq l}\}) = \frac{p(\mathbf{y}|h_l, z_l, \{h_{i\neq l}\})p(h_l|z_l, , \{h_{i\neq l}\})}{p(\mathbf{y}|z_l, \{h_{i\neq l}\})} \tag{4.101}$$

$$= \frac{p(\mathbf{y}|\mathbf{h})p(h_l|z_l)}{p(\mathbf{y}|z_l, \{h_{i\neq l}\})} \tag{4.102}$$

$$\sim \exp\left(-\frac{|\mathbf{y} - \mathbf{D}\mathbf{h}|^2}{2\sigma_w^2} - \frac{|h_l|^2}{2z_l}\right) \tag{4.103}$$

$$= \exp\left(-\frac{\sum_{i=1}^N |y_i - \mathbf{d_i}^T\mathbf{h}|^2}{2\sigma_w^2} - \frac{h_l}{2z_l}\right) \tag{4.104}$$

$$= \exp\left(-\frac{\sum_{i=1}^N |y_i - \sum_{j\neq l}^M d_{ij}h_j - d_{il}h_l|^2}{2\sigma_w^2} - \frac{|h_l|^2}{2z_l}\right) \tag{4.105}$$

$$\sim \exp\left(\frac{\sigma_w^2 + \sum_{i=1}^N |d_{il}|^2}{2z_l\sigma_w^2}\left|h_l - \frac{z_l\left(\sum_{i=1}^N d_{il}^* y_i - \sum_{i=1}^N d_{il}^* \sum_{j\neq l} d_{il}h_l\right)}{\sigma_w^2 + z_l\sum_{i=1}^N |d_{il}|^2}\right|^2\right). \tag{4.106}$$

Defining $\tilde{\mathbf{D}}_l$ as the matrix $\mathbf{D}$ with the l^{th} column removed and $\tilde{\mathbf{h}}_l$ as the vector $\mathbf{h}$ with the

l^{th} element removed, we have

$$E\left[h_l|\mathbf{y}, z_l, \{h_{i\neq l}\}\right] = \frac{z_l\left(\sum_{i=1}^{N} d_{il}^* y_i - \sum_{i=1}^{N} d_{il}^* \sum_{j\neq l} d_{il} h_l\right)}{\sigma_w^2 + z_l \sum_{i=1}^{N} |d_{il}|^2} \tag{4.107}$$

$$= \frac{\underline{\mathbf{d}}_l^H(\mathbf{y} - \tilde{\mathbf{D}}_l\tilde{\mathbf{h}}_l)}{|\underline{\mathbf{d}}_l|^2} \frac{z_l}{z_l + \sigma_w^2/|\underline{\mathbf{d}}_l|^2}. \tag{4.108}$$

When estimating a scalar h, $\tilde{\mathbf{D}}_l$ and $\tilde{\mathbf{h}}_l$ are empty, reducing the expression in (4.108) to the one in (4.52).

$$p(z_l|\mathbf{y}, \{h_{i\neq l}\}) = \frac{p(\mathbf{y}|z_l, \{h_{i\neq l}\})p(z_l|\{h_{i\neq l}\})}{p(\mathbf{y}|\{h_{i\neq l}\})}. \tag{4.109}$$

Making the observation that

$$y_i = d_{il}\sqrt{z_l}n_l + \sum_{j\neq l}^{M} d_{ij}h_j + w_i \tag{4.110}$$

$$\mathbf{y} = \sqrt{z_l}n_l\underline{\mathbf{d}}_l + \tilde{\mathbf{D}}_l\tilde{\mathbf{h}}_l + \mathbf{w}, \tag{4.111}$$

where n_l is a complex normal random variable with $\sigma_n^2 = 2$, the conditional posterior of z_l can be computed as

$$p(z_l|\mathbf{y}, \{h_{j\neq l}\}) = \frac{\exp\left(-(\mathbf{y} - \tilde{\mathbf{D}}_l\tilde{\mathbf{y}}_l)^H(2z_l\underline{\mathbf{d}}_l\underline{\mathbf{d}}_l^H + 2\sigma_w^2\mathbf{I}_N)^{-1}(\mathbf{y} - \tilde{\mathbf{D}}_l\tilde{\mathbf{y}}_l)\right) z_l^{k-1}\exp(-z_l/\theta)}{p(\mathbf{y}|\{h_{j\neq l}\})\pi^N \det(2z_l\underline{\mathbf{d}}_l\underline{\mathbf{d}}_l^H + 2\sigma_w^2\mathbf{I}_N)\theta^k\Gamma(k)}. \tag{4.112}$$

Next, noting that

$$\det\left(2z_l\underline{\mathbf{d}}_l\underline{\mathbf{d}}_l^H + 2\sigma_w^2\mathbf{I}_N\right) = (2\sigma_w^2)^N \det\left(\frac{z_l\underline{\mathbf{d}}_l\underline{\mathbf{d}}_l^H}{\sigma_w^2} + \mathbf{I}_N\right) \tag{4.113}$$

$$= (2\sigma_w^2)^N \frac{|\underline{\mathbf{d}}_l|^2}{\sigma_w^2}\left(z_l + \frac{\sigma_w^2}{|\underline{\mathbf{d}}_l|^2}\right), \tag{4.114}$$

and applying the well-known Woodbury matrix identity,

$$(A + UCV) = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}, \tag{4.115}$$

with

$$A = \sigma_w^2 \mathbf{I}_N, \quad U = \underline{\mathbf{d}}_l, \quad V = \underline{\mathbf{d}}_l^H, \quad C = z_l \mathbf{I}_N \tag{4.116}$$

to obtain

$$\left(2z_l \underline{\mathbf{d}}_l \underline{\mathbf{d}}_l^H + 2\sigma_w^2 \mathbf{I}_N\right)^{-1} = \frac{1}{2\sigma_w^2}\left(\mathbf{I}_N - \frac{z_l}{z_i + \sigma_w^2/|\underline{\mathbf{d}}_l|^2} \frac{\underline{\mathbf{d}}_l \underline{\mathbf{d}}_l^H}{|\underline{\mathbf{d}}_l|^2}\right) \tag{4.117}$$

$$= \frac{1}{2\sigma_w^2}\left(\mathbf{I}_N - \frac{\underline{\mathbf{d}}_l \underline{\mathbf{d}}_l^H}{|\underline{\mathbf{d}}_l|^2} - \frac{\sigma_w^2/|\underline{\mathbf{d}}_l|^2}{z_i + \sigma_w^2/|\underline{\mathbf{d}}_l|^2} \frac{\underline{\mathbf{d}}_l \underline{\mathbf{d}}_l^H}{|\underline{\mathbf{d}}_l|^2}\right) \tag{4.118}$$

leads to an expression of the same form as (4.57).

$$p(z_l|\mathbf{y}, \{h_{j\neq l}\}) = \frac{\frac{z_l^{k-1}}{z_l+\eta} \exp\left(-\frac{\beta}{z_l+\eta} - \frac{z_l}{\theta}\right)}{\int_0^\infty \frac{z_l^{k-1}}{z_l+\eta} \exp\left(-\frac{\beta}{z_l+\eta} - \frac{z_l}{\theta}\right) dz_l} \tag{4.119}$$

However, the parameters are now

$$\eta' = \frac{\sigma_w^2}{|\underline{\mathbf{d}}_l|^2} \tag{4.120}$$

$$\beta' = \frac{\left(\mathbf{y} - \tilde{\mathbf{D}}_l \tilde{\mathbf{h}}_l\right)^H \left(\underline{\mathbf{d}}_l \underline{\mathbf{d}}_l^H\right) \left(\mathbf{y} - \tilde{\mathbf{D}}_l \tilde{\mathbf{h}}_l\right)}{2|\underline{\mathbf{d}}_l|^4} \tag{4.121}$$

$$= \frac{|\underline{\mathbf{d}}_l^H \left(\mathbf{y} - \tilde{\mathbf{D}}_l \tilde{\mathbf{h}}_l\right)|^2}{2|\underline{\mathbf{d}}_l|^4}. \tag{4.122}$$

Finally, the MMSE estimate of h_l given $\{h_{j\neq l}\}$ is

$$\hat{h}_l = \frac{\underline{\mathbf{d}}_l^H \left(\mathbf{y} - \tilde{\mathbf{D}}_l \tilde{\mathbf{h}}_l\right)}{|\underline{\mathbf{d}}_l|^2} E\left[\frac{z_l}{z_l + \eta'}\right] \tag{4.123}$$

$$= \frac{\underline{\mathbf{d}}_l^H \left(\mathbf{y} - \tilde{\mathbf{D}}_l \tilde{\mathbf{h}}_l\right)}{|\underline{\mathbf{d}}_l|^2} \frac{\zeta(\beta', \eta'; k, \theta)}{\zeta(\beta', \eta'; k, \theta) + \eta\zeta(\beta', \eta'; k-1, \theta)}. \tag{4.124}$$

4.3.3 Results

The Matlab code used to implement the vector channel simulation and estimation techniques can be found in Appendix C. The channel taps and noise were assumed to each have independent components. The random variable realizations were calculated in the same manner as the scalar case, and the MSE was estimated from a sample average over 5000 trials. The matrix $\mathbf{D}$ was assumed to be the identity matrix.

The MAP estimate was calculated using the grouped coordinate descent algorithm described in Section 4.3.1. The convergence of Newton's Method was analyzed for different initial values of $\mathbf{r}_0$. Figure 4-5 displays the error as measured from the current estimate to the final estimate over each iteration. Quadratic convergence begins immediately when the initial point is set to $\sqrt{\mu_z}$, where μ_z is the mean of the Gamma distribution. This corresponds to the left most curve in the figure with $r_0 = 0.0071$. The convergence of the grouped coordinate descent is dependent upon the number of iterations allocated to Newton's Method in each iteration of the coordinate descent (CD). This is demonstrated in Figure 4-6. Ultimately, a maximum of 20 iterations of the coordinate descent were performed, with one Newton iteration at each step.

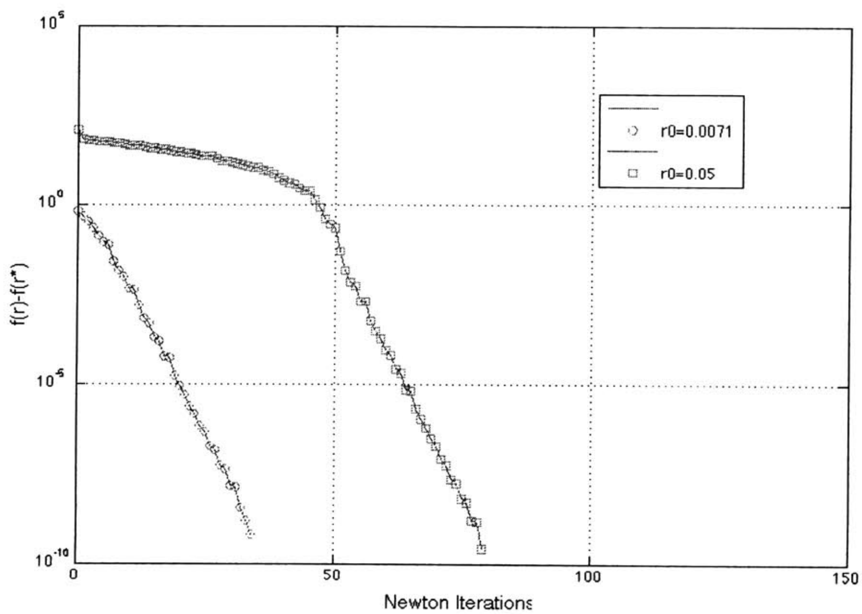


Figure 4-5: The convergence of Newton's method in estimating $\mathbf{r}$ for different initial values

The MMSE estimate was calculated via the coordinate descent approach outlined in section 4.3.2, using 20 iterations. As in the scalar case, the estimators were tested on channel simulations governed by statistics that corresponded to realistic parameters seen in the data of Chapter 2, as well as extreme values of the K-distribution parameters.

The distributions that were observed in the SPACE02 data are displayed in Figure 4-1. When the estimators derived in this chapter were implemented with M=4 and N=4, the

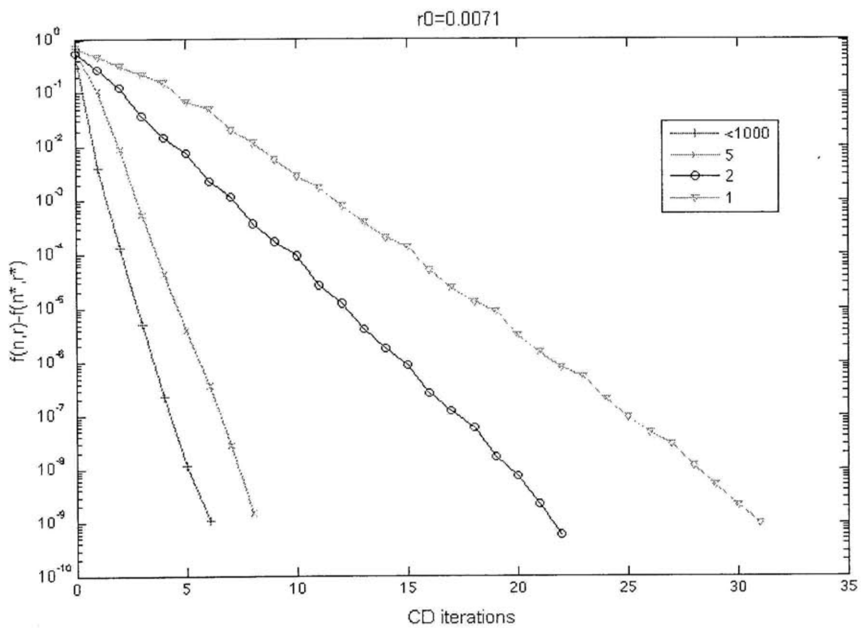


Figure 4-6: Coordinate descent convergence rates for various Newton iterations

MSE behavior seen in Figure 4-7 was obtained. The resultant MSE curves for $\nu = -0.95$ can be seen in Figure 4-8. There is a maximum improvement in MSE of 1.54 dB by using the MAP estimate, and 2.96 dB using the MMSE estimate. The probability distributions associated with this simulation can be found in Figure 4-3.

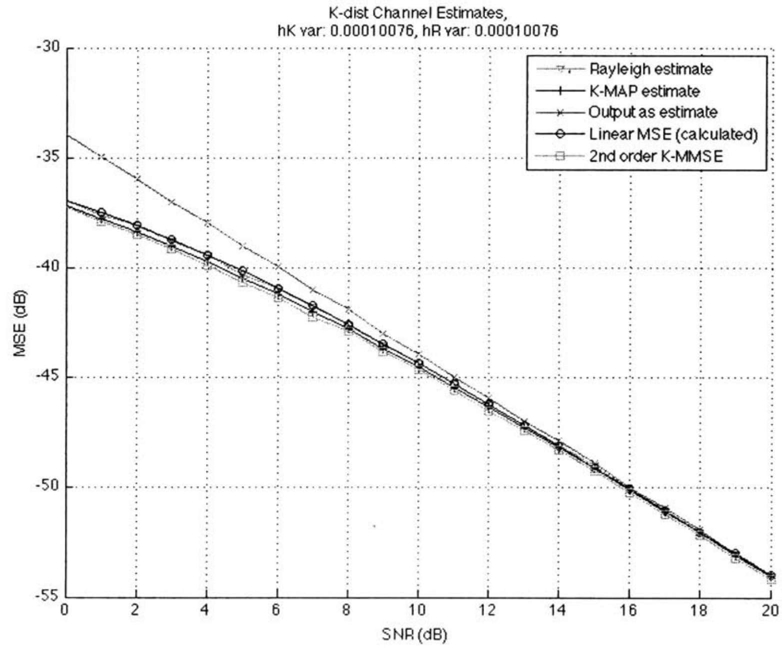


Figure 4-7: K-distribution vector channel estimate MSE: $\nu = -0.3447$, $\alpha = 0.0062$

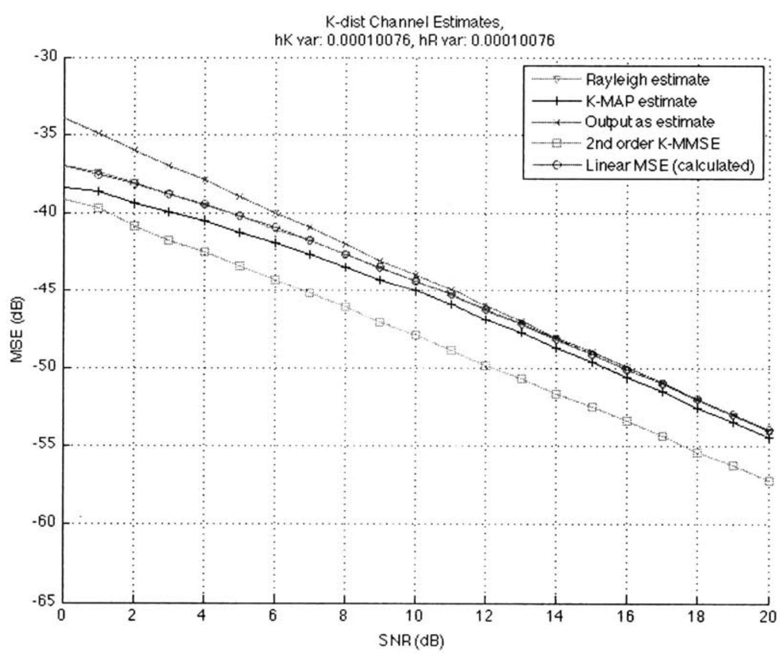


Figure 4-8: K-distribution vector channel estimate MSE: $\nu = -0.95$, $\alpha = 0.0224$

Chapter 5

Conclusions

Analysis of data from the SPACE02 experiment confirms that the channel taps associated with reflections from the ocean surface can exhibit fading that is better modeled as K-distributed than Rayleigh or Rician. The reflections from a single surface interaction were analyzed in detail. The results suggested that these channel taps became more K-distributed, with smaller shape parameters, during periods of high wind and wave activity. The channel taps became more Rayleigh-like at lower bandwidths and longer source-to-receiver distances. Both of these scenarios were attributed to more scatterers contributing to a resolvable arrival in delay.

The components of the scattering process were empirically examined. The "peak arrival" amplitude was determined to be well fit by a gamma distribution. Simulations of the scattering process were conducted, and it was possible to recreate the envelope statistics observed in the data. The simulation results also indicated that, under a discrete scatterer model, the resulting distribution is most sensitive to the number of scatterers used and their arrival's separation in delay.

Although there is a substantial performance improvement using the K-distribution estimates with certain distribution pairs, there is no evidence that such extreme distribution disparities exist in a realistic environment. The performance gained by using the K-Distribution model with realistic parameters was small compared with the increased computational complexity of the estimates. However, should fading models or other estimation scenarios arise with distributions comparable to those shown in Figure 4-3, this approach might prove advantageous.

Appendix A

MATLAB Code: Fading Statistics

A.1 Histogram Calculations

```
%adapted from code written by Dr. James C. Preisig
curdir = pwd;
rxsigdir =['/Users/alisonlaferriere/Documents/WHOI/Summer_Research/'...
    'SPACE02_Pulse_Data/'];
savedir = uigetdir

fs = 5e6/112;
siglen = 56;
fc = 14e3;
numpulsereps = 1307;
Nr = 1786;
Tb = 0.5;
numsampperfile = fs*siglen;

chnum = 18;

Δcycles = Nr/fs*fc;
Δcycles = Δcycles - floor(Δcycles);
Δomega = 2*pi*Δcycles;
phaseadjust = repmat(exp(-sqrt(-1)*Δomega*[0:numpulsereps-1]),Nr,1);

numblank = round(fs*Tb); clear Tb
```

```

expvec = exp((-sqrt(-1)*2*pi*fc/fs)*[0:numsampperfile-1].');
daxis = ([0:Nr-1]-59)*1000/fs;

sdi = min(find(daxis≥-1));
edi = max(find(daxis≤18));


epochvec = [3310000 3330000 3330200 3330400 3331600 3340200 ...
    3341200 3341400 3370000 3370200 3370800 3371400];
segnumvec = [13:19];
numsegs = length(segnumvec);

taxis = [0:numpulsereps-1]*Nr/fs;
lta = length(taxis);

numbins = 80;
dB_range = 30;

for epoch = epochvec
   for system_number = [1 2 4]
    if (system_number==1)
      startindex = 7404; % 60 samples in front of peak of first arrival
    elseif (system_number==2)
      if (chnum≥16)
        startindex = 14508;
      else
        startindex = 4472;
      end
    elseif (system_number==4)
      startindex = 28029;
    else
      % error('unrecognized system number.')
    end

    for lpfiltlen = [4 8 16 32 64 128 256]

      desired_baseband = ones(lpfiltlen,1)/lpfiltlen;
      numprior = 0;
```

```
    mv_abs = 0;
    numgoodpulses = zeros(1,numsegs);
    suspectpulses_saved = ones(numsegs,lta);
    impresp = complex(zeros(Nr,numpulsereps,numsegs));

    for segnum = segnumvec
      fname = ['sys',int2str(system_number),'ch',int2str(chnum) ...
           ,'_',int2str(epoch),int2str(segnum),'.rsig']
      cd(rxsigdir);
      cd(int2str(epoch))
      rsig = load_SPACE02_single_chan_rsig_fn(fname, ...
                             startindex+numblank,numsampperfile);
      cd(curdir)
      bbsig1 = rsig.*expvec;
      bbsig2 = freqdomainconv(desired_baseband,bbsig1);
      bbsig3 = bbsig2([1:Nr*numpulsereps]);
      bbsig3 = reshape(bbsig3,Nr,numpulsereps);
      bbsig3 = bbsig3 .* phaseadjust;
        % this compensates for phase discontinuity in tx sig
        % at the end of each pulse cycle.

      numprior = numprior + 1;

      impresp(:,:,numprior) = bbsig3;
      diffabs = diff(abs(bbsig3).').'/max(max(abs(bbsig3)));
      temppos = zeros(size(diffabs));
      tempneg = temppos;
      temppos(find(diffabs>0.3)) = 1;
      tempneg(find(diffabs<-0.3)) = 1;
      nr = size(tempneg,2);
      tempselect = max(temppos(:,1:nr-1).*tempneg(:,2:nr));
      suspectpulses_saved(numprior,2:numpulsereps-1) = tempselect;

      mv_abs=max([mv_abs,max(max(abs(bbsig3(:,find(tempselect==0)))))]);
      numgoodpulses(numprior) = sum(tempselect==0);
    end

    max_abs = double(zeros(numsegs,Nr));
    min_abs = double(zeros(numsegs,Nr));
```

```
    mean_abs = double(zeros(numsegs,Nr));
    std_abs = double(zeros(numsegs,Nr));
    mean_abssq = double(zeros(numsegs,Nr));
    std_abssq = double(zeros(numsegs,Nr));
    mean_dB = double(zeros(numsegs,Nr));
    std_dB = double(zeros(numsegs,Nr));
    mean_real = double(zeros(numsegs,Nr));
    mean_imag = double(zeros(numsegs,Nr));
    std_real = double(zeros(numsegs,Nr));
    std_imag = double(zeros(numsegs,Nr));
    mean_realsq = double(zeros(numsegs,Nr));
    mean_imagsq = double(zeros(numsegs,Nr));
    cc_real_imag = double(zeros(numsegs,Nr));

    moment_2_abs = double(zeros(numsegs,Nr));
    moment_2_dB = double(zeros(numsegs,Nr));
    moment_4_abs = double(zeros(numsegs,Nr));
    moment_4_dB = double(zeros(numsegs,Nr));
    var_abs = double(zeros(numsegs,Nr));
    var_dB = double(zeros(numsegs,Nr));

    mv_dB = 20*log10(mv_abs);
    edges_abs = [0:numbins]*mv_abs*1.0001/(numbins);
    edges_dB = [-(numbins):0]/(numbins)*dB_range + mv_dB+0.001;
    Δ_abs = mv_abs*1.0001/(numbins);
    Δ_dB = dB_range/(numbins);

    hist_abs = zeros(numbins,Nr,numsegs);
    hist_dB = zeros(numbins,Nr,numsegs);

    for ii=1:numsegs

      curresp = impresp(:,find(suspectpulses_saved(ii,:)==0),ii).';
      mean_real(ii,:) = mean(real(curresp));
      mean_imag(ii,:) = mean(imag(curresp));
      std_real(ii,:) = std(real(curresp));
      std_imag(ii,:) = std(imag(curresp));
      mean_realsq(ii,:) = mean(real(curresp).^2);
      mean_imagsq(ii,:) = mean(imag(curresp).^2);
```

```
        cc_real_imag = ...
          mean((imag(curresp) - repmat(mean_imag(ii,:),...
          numgoodpulses(ii),1)).* (real(curresp) - ...
          repmat(mean_real(ii,:),numgoodpulses(ii),1)));

        curresp = abs(curresp);

        max_abs(ii,:) = max(curresp);
        min_abs(ii,:) = min(curresp);
        mean_abs(ii,:) = mean(curresp);
        std_abs(ii,:) = std(curresp);
        moment_2_abs(ii,:) = sum(curresp.^2)/numgoodpulses(ii);
        moment_4_abs(ii,:) = sum(curresp.^4)/numgoodpulses(ii);
        var_abs(ii,:) = var(curresp);
        temphist = histc(curresp,edges_abs);
        hist_abs(:,:,ii) = temphist(1:numbins,:);


        curresp = curresp.^2;
        mean_abssq(ii,:) = mean(curresp);
        std_abssq(ii,:) = std(curresp);

        curresp = 10*log10(curresp);
        mean_dB(ii,:) = mean(curresp);
        std_dB(ii,:) = std(curresp);
        moment_2_dB(ii,:) = sum(curresp.^2)/numgoodpulses(ii);
        moment_4_dB(ii,:) = sum(curresp.^4)/numgoodpulses(ii);
        var_dB(ii,:) = var(curresp);
        temphist = histc(curresp,edges_abs);
        hist_dB(:,:,ii) = temphist(1:numbins,:);


    end
    cd([savedir])
    eval(['save hist_fading_lpfiltlen',int2str(lpfiltlen),'_sys'...
      ,int2str(system_number),'_channel',int2str(chnum),'_',...
      int2str(epoch),'  hist_abs hist_dB edges_abs edges_dB max_abs '...
      'min_abs mean_abs std_abs mean_abssq std_abssq mean_dB '...
      'std_dB numpulsereps mv_abs mv_dB Δ_abs Δ_dB mean_real '...
```

```
            'mean_imag std_real std_imag mean_realsq mean_imagsq '...
            'cc_real_imag suspectpulses_saved numgoodpulses moment_2_abs '...
            'moment_2_dB moment_4_abs moment_4_dB'])
          cd(curdir)
          pause(0.5)
      end
    end
end
```

```
function rsig = ...
    load_SPACE_single_chan_rsig_file_function(filename,startindex,numsamp)
%following code provided by Dr. James C. Preisig
%filename = input('file name? ','s');
%startindex = input('starting sample index (first sample index = 0)? ');
%numsamp = input('number samples? ');

numbuffersamp = 200000;
rsig = zeros(numsamp,1);

fid = fopen(filename,'r','ieee-le');

[endian_flag,count] = fread(fid,1,'int32');
if (endian_flag == -402456576)
  [fname,permission,fileformat] = fopen(fid);
  if (fileformat == 'ieee-be')
    iostat = fclose(fid);
    if (iostat ≠ 0)
      error('problem closing file')
    end
    fid = fopen(filename,'r','ieee-le');
    if (fid<0)
      error('problem opening file')
    end
    [a,count] = fread(fid,4,'int32');
    endian_flag = a(1);
    if (endian_flag ≠ 1000)
      fclose(fid)
      error(['file does not return proper endian flag in either'...
```

```
            'big or little endian format'])
    end
  elseif (fileformat == 'ieee-le')
    iostat = fclose(fid);
    if (iostat ≠ 0)
      error('problem closing file')
    end
    fid = fopen(filename,'r','ieee-be');
    if (fid<0)
      error('problem opening file')
    end
    [a,count] = fread(fid,4,'int32');
    endian_flag = a(1);
    if (endian_flag ≠ 1000)
      fclose(fid)
      error(['file does not return proper endian flag in either big'...
          ' or little endian format'])
    end
  else
    fclose(fid)
    error('unknown file format returned.')
  end
elseif (endian_flag ≠ 1000)
  error(['unknown endian flag variable: endian_flag = ',endian_flag])
end

numdatabyteskip = startindex*2; % number of bytes to skip
skipstatus = fseek(fid,numdatabyteskip,'cof');

numsegs = ceil((numsamp-startindex)/numbuffersamp);
numdataread = numbuffersamp;

for mm=0:numsegs-2
  [data,count] = fread(fid,numdataread,'int16');
  e = 8.^(3-mod(data,4));
  m = bitshift(data,-2);
  rsig(mm*numbuffersamp+[1:numbuffersamp]) = m.*e;
end
mm= numsegs-1;
```

```
lastsamp = mm*numbuffersamp;
numbuffersamp = numsamp - lastsamp - startindex;
numdataread = numbuffersamp;
[data,count] = fread(fid,numdataread,'int16');
e = 8.^(3-mod(data,4));
m = bitshift(data,-2);
rsig(lastsamp+[1:length(e)]) = m.*e;
rsig = rsig * (2.^(-22));

fclose(fid);
```

```
function fout = freqdomainconv(in1,in2)
% fout = freqdomainconv(in1,in2)

foutlen = length(in1) + length(in2) -1;
fftlen = 2^nextpow2(foutlen);
fout = ifft(fft(in1,fftlen).*fft(in2,fftlen));
fout = fout(1:foutlen);
```

```
function [ls_s,ls_g,ls_x,ls_px]=grid_search_fcn(data_hist,s,g,Δ_abs,type_dist)
  %data_hist= histogram to compare
  %s = parameter 1
  %g = parameter 2
  %x = data points
  %type_dist = type of distribution (k-dist, rician, rayleigh,log)

  save_form_param=0; %set to 1 to create record of parameters

  num_x=length(data_hist);
  Δ_scale=10; %scales number of data points for distribution calc
  x_scaled = ([0:2*num_x*Δ_scale-1]+0.0001)*Δ_abs/Δ_scale;
  ls_x = [0:2*num_x-1]*Δ_abs;

  if type_dist=='k' %(k-dist) s=nu, g=a
      px = k_dist_pdf(s,g,x_scaled);
  elseif type_dist=='c' %(rician) sigma=s, mu=g
```

```
        px = rician_pdf(s,g,x_scaled);
    elseif type_dist=='r' %(rayleigh) sigma=s, mu=0
        px = rician_pdf(0,g,x_scaled);
    elseif type_dist=='l' %log-normal sigma=s, mu=g
        px=lognpdf(x_scaled,g,s);
    else
          error('invalid distribution type')
    end

    %segmented integration
    int_px = zeros(1,2*num_x);
    int_px(1)=(Δ_abs/Δ_scale)*(sum(px(1:Δ_scale-1))+ ...
        sum(px(2:Δ_scale)))/2;
    index=0;
    for ii=2:160
        index=index+Δ_scale;
        int_px(ii)=(Δ_abs/Δ_scale)*(sum(px(index:index+Δ_scale-1))+ ...
            sum(px(index+1:index+Δ_scale)))/2;
    end

    ls_s = s;
    ls_g = g;
    grid_spacing = 0.1;
    num_grid = 11;

     numiterations = 0;
     if save_form_param==1
          former_param_px=zeros(10,2);
          former_error_px=zeros(10,1);
     end
    while (grid_spacing > 0.001)
      old_s = ls_s;
      old_g = ls_g;
      g_grid = old_g*(1+[-(num_grid-1)/2: ...
                                  (num_grid-1)/2]*grid_spacing);
      s_grid = old_s*(1+[-(num_grid-1)/2:(num_grid-1)/2]*grid_spacing);
      %changed = 0;

      if type_dist=='k' %(k-dist) s=nu, g=a
```

```
        cur_px = grid_k_dist_pdf(g_grid,s_grid,x_scaled);
    elseif type_dist=='c' %(rician) s=sigma, g=mu
        cur_px = grid_rician_pdf(g_grid,s_grid,x_scaled);
    elseif type_dist=='r' %(rayleigh) s=sigma, g=0
        cur_px = grid_rician_pdf(g_grid,s_grid,x_scaled);
    elseif type_dist=='l' %log-normal sigma=s, mu=g
        cur_px = grid_log_pdf(g_grid,s_grid,x_scaled);
    else
        error('invalid distribution type')
    end

    cur_int_px = zeros(11,160,11);
    index=0;

    cur_int_px(:,1,:)=Δ_abs/Δ_scale*(sum(cur_px(:,1:Δ_scale-1,:),2) ...
        + sum(cur_px(:,2:Δ_scale,:),2))/2;
    for ii=2:160
        index=index+Δ_scale;

        cur_int_px(:,ii,:) = ...
         (Δ_abs/Δ_scale)*(sum(cur_px(:,index:index+Δ_scale-1,:),2) ...
         + sum(cur_px(:,index+1:index+Δ_scale,:),2))/2;
    end

    %create matrix of errors for all values of sigma and mu
    cur_error_px = squeeze(sum(abs(cur_int_px(:,1:num_x,:) - ...
        repmat(data_hist,[11 1 11])),2) ...
        + sum(cur_int_px(:,num_x+1:2*num_x,:),2));

    [g_index,s_index]=find(cur_error_px==min(min(cur_error_px)));

    g_index = g_index(round(length(g_index)/2));
    s_index=s_index(round(length(s_index)/2));

    ls_px = cur_int_px(g_index,:,s_index);
    ls_s = s_grid(s_index);
    ls_g = g_grid(g_index);

    ls_error_px = cur_error_px(g_index,s_index);
```

```
    if type_dist == 'r'
        if abs(s_index-5) < 3
            grid_spacing = grid_spacing / 2;
        end
    else
        if abs(s_index-5) < 3 && abs(g_index-5) < 3
            grid_spacing = grid_spacing / 2;

        end
    end

    numiterations = numiterations + 1;
    %record former parameters
    if save_form_param==1
        former_param_px(numiterations,1) = ls_s;
        former_param_px(numiterations,2) = ls_g;
        former_error_px(numiterations) = ls_error_px;
    end
end
```

A.2 Distribution Fitting

```
%uses abs error to find best fit
%can choose option of displaying calculated arrivals
%can choose option of running environmental parameter/error correlations
savefolder1=uigetdir;

epochvec = [3310000 3330000 3330200 3330400 3331600 3331800 3340200 ...
    3341200 3341400 3370000 3370200 3370800 3371400];
lpfiltlen_vec =[4 8 16 32 64 128 256];

chnum =2;
dist = 80; %source to receiver distance
D=16; %water depth
c=1450; %speed of sound in water
del_delay=4; % will avg statistics over +-del_delay samples
```

```

segnumvec = [13:19];
system_vec = [1 2 4];
numsegs = length(segnumvec);

fs=5e6/112;
Nr=1786;
daxis=([0:Nr-1]-59)*1000/fs;

start_band_vec=zeros(length(epochvec),1);
end_band_vec=zeros(length(epochvec),1);
band_length_vec=zeros(length(epochvec),1);

idx=0;
for lpfiltlen=lpfiltlen_vec
    idx=idx+1;
    close all
    clear out, clear intervals, clear output
    mkdir(savefolder1,['/lpfilt',num2str(lpfiltlen)])
    savefolder=[savefolder1,'/lpfilt',num2str(lpfiltlen)];

    taud=1000*sqrt(dist^2+(6.25-3.3)^2)/c;
    taus=1000*sqrt(dist^2+(D-6.25+D-3.3)^2)/c-taud;
    taub=1000*sqrt(dist^2+(6.25+3.3)^2)/c-taud;
    taubs=1000*sqrt(dist^2+(D-3.3+D+6.25)^2)/c-taud;
    tausb=1000*sqrt(dist^2+(D+3.3+D-6.25)^2)/c-taud;

    curdir = pwd;
    out = [];
    intervals = [];

    for ep=1:length(epochvec)
        epoch=epochvec(ep);
        for system_number = system_vec
            savedir = ['/Users/alisonlaferriere/Documents/WHOI/'...
                'Re-evaluated/histograms'];
            fname = ['hist_fading_lpfiltlen',int2str(lpfiltlen),'_sys', ...
                int2str(system_number),'_channel',int2str(chnum),...
                '_',int2str(epoch)];
```

```
cd(savedir)
load(fname)
cd(curdir)

numbins = length(edges_abs)-1;
numtaps = length(mean_abs);

centers_abs = (edges_abs(1:numbins)+edges_abs(2:numbins+1))/2;
max_x = max(centers_abs);
min_x = min(centers_abs);
num_x = length(centers_abs);
sample_rician = [0:2*num_x-1]*Δ_abs;

combined_mean_abssq = numgoodpulses*mean_abssq/...
    sum(numgoodpulses);
combined_mean_imag = numgoodpulses*mean_imag/...
    sum(numgoodpulses);
combined_mean_real = numgoodpulses*mean_real/...
    sum(numgoodpulses);
combined_mean_abs = numgoodpulses*mean_abs/...
    sum(numgoodpulses);
combined_max_abs = max(max_abs);
combined_abs_pdf = sum(hist_abs,3)/sum(numgoodpulses);
combined_std_imag = sqrt(numgoodpulses*(std_imag.^2 +...
    mean_imag.^2)/sum(numgoodpulses) - combined_mean_imag.^2);
combined_std_real = sqrt(numgoodpulses*(std_real.^2 +...
    mean_real.^2)/sum(numgoodpulses) - combined_mean_real.^2);
combined_std_abs = sqrt(numgoodpulses*(std_abs.^2+...
    mean_abs.^2)/sum(numgoodpulses)-combined_mean_abs.^2);
combined_moment_2_abs = numgoodpulses*moment_2_abs/...
    sum(numgoodpulses);
combined_moment_4_abs = numgoodpulses*moment_4_abs/...
    sum(numgoodpulses);

data_plot=figure(1); clf, hold on, grid on
[minerr,zerdel]=min(abs(daxis));
plot(daxis,10*log10(combined_max_abs.^2))
plot(daxis,10*log10(combined_mean_abssq),'r')
aa = axis;
```

```
        axis([daxis(zerdel) daxis(600) aa(3:4)])  %30-600
        xlabel('Relative Delay (milliseconds)')
        ylabel('Intensity (dB)')

        figure(data_plot)

        if idx==1

        %*************************************************************
        %if uncommented the following lines of code will mark the
        % calculated arrival times of the
        %direct/surface/surface-bottom/bottom-surface arrivals
            %pause
            %[dir_arrival,yd,nextd]=ginput(2);%input for delay band
            %[err,start_d]=min(abs(daxis-dir_arrival(1)));
            %[err,end_d]=min(abs(daxis-dir_arrival(2)));
            %taud_meas=find(combined_mean_abs==...
            %    max(combined_mean_abs(start_d:end_d)));
            %taud_meas=daxis(taud_meas);
            %aa=axis;

            %plot([(taud_meas) (taud_meas)],aa(3:4),'k','linewidth',2);
            %plot([(taus+taud_meas) (taus+taud_meas)],aa(3:4),...
            %'c','linewidth',2);
            %plot([taub+taud_meas taub+taud_meas],aa(3:4),'g',...
            %'linewidth',2);
            %plot([tausb+taud_meas tausb+taud_meas],aa(3:4),'m',...
            %'linewidth',2);
            %plot([taubs+taud_meas taubs+taud_meas],aa(3:4),'y',...
            %'linewidth',2);
            %cur_aa=gca;
            %legend('max','avg','d','s','b','sb','bs')
            pause
            %axes(cur_aa);
        %*************************************************************

            [delay_band,y,next]=ginput(2);%user input for delay band
            [err,start_band]=min(abs(daxis-delay_band(1)));
            [err,end_band]=min(abs(daxis-delay_band(2)));
```

```
        maxmeanpoint=find(combined_mean_abs==...
            max(combined_mean_abs(start_band:end_band)));
        start_band_vec(ep)=maxmeanpoint-del_delay;
        end_band_vec(ep)=maxmeanpoint+del_delay;
    end

    start_band=start_band_vec(ep);
    end_band=end_band_vec(ep);
    band_length=length(start_band:end_band);

    band_mean_abssq=...
        sum(combined_mean_abssq(start_band:end_band))/band_length;
    band_mean_imag = ...
        sum(combined_mean_imag(start_band:end_band))/band_length;
    band_mean_real = ...
        sum(combined_mean_real(start_band:end_band))/band_length;
    band_max_abs = ...
        sum(combined_max_abs(start_band:end_band))/band_length;
    band_abs_pdf = ...
       sum(combined_abs_pdf(:,start_band:end_band),2)./band_length;
    band_std_imag = ...
        sum(combined_std_imag(start_band:end_band))/band_length;
    band_std_real = ...
        sum(combined_std_real(start_band:end_band))/band_length;
    band_std_abs = ...
        sum(combined_std_abs(start_band:end_band))/band_length;
    band_moment_2_abs = ...
        sum(combined_moment_2_abs(start_band:end_band))/band_length;
    band_moment_4_abs = ...
       sum(combined_moment_4_abs(start_band:end_band))/band_length;

    m= band_moment_4_abs/(band_moment_2_abs^2);
    band_nu=(m-4)/(2-m);
    if band_nu<-1
        epoch
        band_nu
        warning('K parameter estimate invalid, set to abs(.)')
        band_nu=100;
    elseif band_nu>100
```

```
      warning('K parameter estimate exceeds threshold, set to 100')
      band_nu=100;
    end
    band_a=sqrt(band_moment_2_abs/(4*(1+band_nu)));

    band_raysigma = sqrt(band_moment_2_abs/2);

    cur_abs_pdf =  band_abs_pdf';



    band_sigma = sqrt((band_std_imag^2 + band_std_real^2)/2);
    band_mu=abs(sqrt(band_moment_2_abs-2*band_sigma.^2));
    rp = rician_pdf(band_mu,band_sigma,sample_rician);
    kp = k_dist_pdf(band_nu,band_a,sample_rician);
    ray = rician_pdf(0,band_raysigma,sample_rician);

    rp_total_prob = sum(rp)*Δ_abs;
    kp_total_prob = sum(kp)*Δ_abs;
    ray_total_prob = sum(ray)*Δ_abs;

    cap_total_prob = sum(cur_abs_pdf);

    [ls_raysigma,ls_raymu,¬,ls_ray]=...
        grid_search_fcn(cur_abs_pdf,band_raysigma,0,Δ_abs,'r');
    [ls_sigma,ls_mu,¬,ls_rp]=...
     grid_search_fcn(cur_abs_pdf,band_sigma,band_mu,Δ_abs,'c');
    [ls_nu,ls_a,ls_x,ls_kp]=...
        grid_search_fcn(cur_abs_pdf,band_nu,band_a,Δ_abs,'k');


    pdf_plot=figure(2);clf;grid on;hold on;
    figure(pdf_plot);
    plot(centers_abs,cur_abs_pdf,'linewidth',1);
    grid on;hold on
    plot(sample_rician,rp*Δ_abs,'--r','linewidth',1)
    plot(sample_rician,kp*Δ_abs,'--k','linewidth',1)
    plot(sample_rician,ray*Δ_abs,'--c','linewidth',1)
    plot(ls_x,ls_rp,'m','linewidth',1)
    plot(ls_x,ls_kp,'g','linewidth',1)
```

```
        plot(ls_x,ls_ray,'--y','linewidth',1)
        legend('Histogram','Rician Theory','K Theory',...
            'Rayleigh Theory','Rician Best Fit','K Best Fit',...
            'Rayleigh Best Fit')
        xlabel('Intensity')
        ylabel('Probability Density')
          title({['Ch' num2str(chnum) ' System ' num2str(sysnum)], ...
          ['Epoch: ' num2str(epoch)]})

          printfile =[savefolder,'/ch',num2str(chnum),'sys',...
              num2str(sysnum),'_pdf_',num2str(epoch)];
          print('-dpdf',printfile)

          saveas(pdf_plot,printfile,'fig')



          figure(data_plot)
          [minerr,zerdel]=min(abs(daxis));
          plot(daxis,10*log10(combined_max_abs.^2))
          plot(daxis,10*log10(combined_mean_abssq),'r')
          aa = axis;
          plot([daxis(start_band) daxis(start_band)],aa(3:4),'g',...
              'linewidth',2);
          plot([daxis(end_band) daxis(end_band)],aa(3:4),'g',...
              'linewidth',2);
          aa = axis;
          axis([daxis(zerdel) daxis(600) aa(3:4)])  %30-600
          xlabel('Relative Delay (milliseconds)')
          ylabel('Intensity (dB)')
          title({['Ch' num2str(chnum) ' System ' num2str(sysnum)], ...
              ['Epoch: ' num2str(epoch)]})

          printfile =[savefolder,'/ch',num2str(chnum),'sys',...
              num2str(sysnum),'_signal_',num2str(epoch)];
          print('-dpdf',printfile)

          saveas(data_plot,printfile,'fig')

          env_data=figure(3);
```

```
                [h,p,s,d]=plot_wind_and_wave_data(epoch);
            end
            out=[out;epoch, ls_nu, ls_a, min_error_kp, D_KL_kp, ls_mu, ...
                ls_sigma,min_error_rp, D_KL_rp, ls_raysigma, min_error_ray, ...
                D_KL_ray, h,p,s,cur_abs_pdf];
            intervals=[intervals; delay_band';];
        end
        output = {'Epoch','K-Dist Nu','K-Dist a','K-Dist Error',...
            'K-Dist K-L Diff','Rician Mu','Rician Sigma','Rician Error',...
            'Rician K-L Diff','Ray Sigma','Rayleigh Error',...
            'Rayleigh K-L Diff','Wave Height','Wave Period','Wind Speed',...
            'Data Histogram'};
%*********************************************************************
 %if uncommented the following lines of code will save the data to a file
 %and run the environmental parameter vs. error and parameter analysis
        % fileinfo = ['/lpfiltlen',int2str(lpfiltlen),'_ch',...
        %int2str(chnum),'_sys',int2str(system_number)];
        % save([savefolder,fileinfo,'_alldata'])
        % plotresultsv2_partialcorr
 %*********************************************************************

end
```

Appendix B

MATLAB Code: Simulations

```
%scatterer simulation

close all

N=1307*7;
T=1786;
start = 149; %for tau
X=zeros(N,T);

fs=5e6/112;
Nr=1786;
daxis = ([0:Nr-1]-59)*1000/fs;


pulsewidth=0.1; %miliseconds
alpha=1/pulsewidth^2;
Rs=exp(-alpha*daxis.^2);
Rs0=0.04*Rs;

%normal tau using time in ms
tau_mu=2.787;
tau_sigma=0.18113;

%Gamma amplitude
z_shape = 14.4086;
```

```
z_scale= 0.008;
A_shape = 3.79;
A_scale=0.0038;

d1=90;b1=105;    % stationary arrival delays at
Rs_center=60;    % ie daxis=0

Rsd1= [zeros(1,d1-Rs_center) Rs0];
Rsd1(T+1:T+d1-Rs_center)=[];

Rsb1 = [zeros(1,b1-Rs_center) Rs0];
Rsb1(T+1:T+b1-Rs_center)=[];

m=1;% # of surface scatterers
r=5;% # scatterers per surface scatterer
Rst=zeros(m,T);
for n=1:N

    Ai=gamrnd(A_shape,A_scale,m,r); %gamma A

    if r>1
        del_tau=randi([0 14],1,r-1);
    end

    sig_phase=unifrnd(0,2*pi,m,r);

    tauii = normrnd(tau_mu,tau_sigma,m,1);
    taui=59+find(abs(daxis(60:end)-tauii)==min(abs(daxis(60:end)-tauii)));
    %find closest number of units, zero starts at daxis(60)

    for ii=1:m %calculate scattered arrival for each arrival time
        Rsti = (Ai(ii,:).*exp(-1i*sig_phase))'*Rs;
            for k=1:r
                if k>1
                    %independent interval process:
                    taui(ii,k)=taui(ii,k-1)+del_tau(k-1);
                end

                Rsti(k,:)=[zeros(1,taui(ii,k)-Rs_center) ...
```

```
                        Rsti(k,1:T-taui(ii,k)+Rs_center)];


            end
            if r>1
                Rst(ii,:)=sum(Rsti);
            else
                Rst=Rsti;
            end
    end

    X(n,:)=Rsb1+Rsd1+Rst;

    desired_baseband=ones(4,1)/4;
    s=freqdomainconv(desired_baseband,X(n,:)');
    X(n,:)=s(1:T);
end

mean_real = mean(real(X));
mean_imag = mean(imag(X));
std_real = std(real(X));
std_imag = std(imag(X));
mean_realsq = mean(real(X).^2);
mean_imagsq = mean(imag(X).^2);

X=abs(X);

numbins=80;
mv=max(max(X));
edges_abs=[0:numbins]*mv*1.0001/numbins;
Δ_abs = mv*1.0001/(numbins);

pcolor(daxis,0:1299,X(1:1300,:));shading('flat')
axis([0 5 0 1300])

max_abs = max(X);
min_abs = min(X);
mean_abs = mean(X);
std_abs = std(X);
```

```
moment_2_abs = sum(X.^2)/size(X,1);
moment_4_abs = sum(X.^4)/size(X,1);
var_abs = var(X);
temphist = histc(X,edges_abs);
hist_abs = temphist(1:numbins,:);
mean_abssq = mean(X);
std_abssq = std(X);

combined_mean_abssq = mean_abssq;
combined_mean_imag = mean_imag;
combined_mean_real = mean_real;
combined_mean_abs = mean_abs;
combined_max_abs = max_abs;
combined_abs_pdf = hist_abs/N;
combined_std_imag = std_imag;
combined_std_real = std_real;
combined_std_abs = std_abs;
combined_moment_2_abs = moment_2_abs;
combined_moment_4_abs = moment_4_abs;
```

Appendix C

MATLAB Code: Channel Estimation

```
clear all

%setting this to 1 will cause random number generators to use same seed
%for each trial
rand_off=0;
%set to zero to supress plot output
plots_on=0;

num_mmse_it=20;%number of iterations in coordinate descent, MMSE estimate
M=4; %number of channel taps
Num=5000; %number of trials
K=10; %number of observations
CD_iterations=10; %number of iterations in coordinate descent, MAP estimate
Newton_it=1;  % number of iterations in Newton's method for MAP estimate


%K-dist parameters
thetax=2*0.0062^2*(1-0.3447)/0.05;
a=sqrt(thetax/2);
nu=-0.95;%
%nu=-0.3447;
%a=0.0062;
```

```
%gamma dist parameters
theta=2*a^2;
k=nu+1;

% Rayleigh distribution sigma parameter
sig=sqrt(k*theta);

% K-dist var, 2*sig_hK^2 is var of complex channel tap
sig_hK=sqrt(k*theta);

% Covariance matrix for h
Rh=2*sig^2*eye(M);

%vector of SNR in dB
noise_lev=5;%-5:20;

%initialize MSE vectors
hKk_mse=zeros(1,length(noise_lev));
hKy_mse=zeros(1,length(noise_lev));
hKr_mse=zeros(1,length(noise_lev));

hK_var=zeros(1,length(noise_lev));

hRk_mse=zeros(1,length(noise_lev));
hRy_mse=zeros(1,length(noise_lev));
hRr_mse=zeros(1,length(noise_lev));

min_MSE=zeros(1,length(noise_lev));

for noise_lev_index=1:length(noise_lev)

        % initialize MSE vectors for each "look", i.e. trial
        hKk_mse_look=zeros(M,Num);
        hKy_mse_look=zeros(M,Num);
        hKr_mse_look=zeros(M,Num);

        hRk_mse_look=zeros(M,Num);
        hRy_mse_look=zeros(M,Num);
```

```
    hRr_mse_look=zeros(M,Num);

    hK_var_look=zeros(M,Num);
    D=2*(randn(K,M)>0)-1;%eye(K,M);
    %Calculate noise variance based on SNR
    %noise in dB = 10log10(sig_h^2/sig_w^2)
    sig_u=sqrt((sig_hK^2)*10^(-noise_lev(noise_lev_index)/10));
    Cuu=diag(sig_u^2*ones(K,1));
    Cvu=0;
    Cw=2*(Cuu+1i*Cvu); %complex noise covariance

    %calculate MSE expected using LMMSE estimate
    min_MSE(noise_lev_index)=trace((eye(M)/Rh+D'/Cw*D)\eye(M));

    for look_index = 1:Num

        %generate complex quassian noise
        if rand_off==1
        randn('state',0);
        end
        u=mvnrnd(zeros(K,1),sig_u^2);
        if rand_off==1
        randn('state',1);
        end
        v=mvnrnd(zeros(K,1),sig_u^2);
        w=u+1i*v;

        %generate K channel
        if rand_off==1
        randn('state',1);
        end
        [hK,z,nX,nY]=genKchannel(nu,a,M);
        if rand_off==1
        randn('state',0);
        end
        %generate Rayleigh channel
        hR=genRayleighchannel(sig,M);

        yK=D*hK+w;
```

```
        %initialize r_hat=r0
        r0=1*ones(M,1);%sqrt(k*theta)*ones(M,1);
        R0=diag(r0);
        n0=(R0'*D'/Cw*D*R0+eye(M)/2)\(R0'*D'/Cw*yK);

        y=yK;

        if M==1

        betak=abs(yK)^2/(2*D^2);
        etak=sig_u^2/D^2;
        Karg=2*sqrt(betak/theta);

        %1st order MMSE estimate
        t1_1st=-k*besselk(k,Karg)/betak+sqrt(betak*theta)*...
            besselk(k+1,Karg)/(theta*betak);
        t2_1st=(1-k)*besselk(k-1,Karg)/(betak*sqrt(betak*theta))+...
            besselk(k,Karg)/(betak*theta);
        hK_mmse1=(y/D)*t1_1st/(t1_1st+etak*t2_1st);

        %second order MMSE estimate
        hK_mmse_hat=(y/D)*t_fcn(betak,etak,k,theta)/...
            (t_fcn(betak,etak,k,theta)+etak*t_fcn(betak,etak,k-1,theta));

        else

            hK_mmse_hat=zeros(M,1); %initialize algorithm
            for hkhat_it=1:num_mmse_it
                for kidx=1:M
                    dk=D(:,kidx);
                    Dtilde=D;
                    Dtilde(:,kidx)=[];
                    htilde=hK_mmse_hat;
                    htilde(kidx)=[];

                    etak=sig_u^2/norm(dk)^2;

                    betak=norm(dk'*(y-Dtilde*htilde))^2/(2*norm(dk)^4);
```

```
                t_hat=t_fcn(betak,etak,k,theta)/...
                    (t_fcn(betak,etak,k,theta)+...
                    etak*t_fcn(betak,etak,k-1,theta));
                hK_mmse_hat(kidx)=t_hat*dk'*(y-Dtilde*htilde)/...
                    norm(dk)^2;
            end
        end
    end


    r=r0;
    R=diag(r0);
    vals_CD=[];
    for CD_index=1:CD_iterations


        R=diag(r);
        n(:,CD_index)=(R'*D'/Cw*D*R+eye(M)/2)\(R'*D'/Cw*y);
        N=diag(n(:,CD_index));
        val_CD=real((y-D*N*r)'/Cw*(y-D*N*r))+r'*r/theta-(2*k-1)*...
            sum(log(r))+n(:,CD_index)'*n(:,CD_index)/2;
        vals_CD=[vals_CD, val_CD];

        newton_opt; %sets opt r using newton method

    end
    R=diag(r);
    n(:,CD_index)=(R'*D'/Cw*D*R+eye(M)/2)\(R'*D'/Cw*y);
    N=diag(n(:,CD_index));
    val_CD=real((y-D*N*r)'/Cw*(y-D*N*r))+r'*r/theta-(2*k-1)*...
        sum(log(r))+n(:,CD_index)'*n(:,CD_index)/2;
    vals_CD=[vals_CD, val_CD];
    optval=val_CD;


    r_khat=r;
    n_khat=n(:,CD_index);
    R_khat=diag(r_khat);
```

```
%K channel estimates
hK_khat=r_khat.*n_khat;
hK_rhat=(D'/Cw*D+eye(M)/Rh)\D'/Cw*yK;
hKk_err=hK-hK_khat;
hKy_err=hK-yK(1:M);
hKr_err=hK-hK_rhat;
hKmmse1_err=hK-hK_mmse1;
hKmmsehat_err=hK-hK_mmse_hat;


%Rayleigh channel estimates
yR=D*hR+w;

y=yR;
r=r0;
vals_CD=[];
for CD_index=1:CD_iterations
    R=diag(r);
    n(:,CD_index)=(R'*D'/Cw*D*R+eye(M)/2)\(R'*D'/Cw*y);
    N=diag(n(:,CD_index));
    val_CD=real((y-D*N*r)'/Cw*(y-D*N*r))+r'*r/theta-(2*k-1)*...
        sum(log(r))+n(:,CD_index)'*n(:,CD_index)/2;
    vals_CD=[vals_CD, val_CD];

    newton_opt; %sets opt r using newton method

end

r_rhat=r;
n_rhat=n(:,CD_index);
R_rhat=diag(r_rhat);

hR_khat=r_rhat.*n_rhat;
hR_rhat=(D'/Cw*D+eye(M)/Rh)\D'/Cw*yR;
hRk_err=hR-hR_khat;
hRy_err=hR-yR(1:M);
hRr_err=hR-hR_rhat;
```

```
        hKk_mse_look(:,look_index)=abs(hKk_err).^2;
        hKy_mse_look(:,look_index)=abs(hKy_err).^2;
        hKr_mse_look(:,look_index)=abs(hKr_err).^2;
        hKmmse1_mse_look(:,look_index)=abs(hKmmse1_err).^2;
        hKmmsehat_mse_look(:,look_index)=abs(hKmmsehat_err).^2;

        hRk_mse_look(:,look_index)=abs(hRk_err).^2;
        hRy_mse_look(:,look_index)=abs(hRy_err).^2;
        hRr_mse_look(:,look_index)=abs(hRr_err).^2;

    end
    hKk_mse(noise_lev_index)=sum(sum(hKk_mse_look,2)/Num);
    hKy_mse(noise_lev_index)=sum(sum(hKy_mse_look,2)/Num);
    hKr_mse(noise_lev_index)=sum(sum(hKr_mse_look,2)/Num);
    hKmmse1_mse(noise_lev_index)=sum(sum(hKmmse1_mse_look,2)/Num);
    hKmmsehat_mse(noise_lev_index)=sum(sum(hKmmsehat_mse_look,2)/Num);

    hK_var(noise_lev_index)=sum(sum(hK_var_look,2)/Num);

    hRk_mse(noise_lev_index)=sum(sum(hRk_mse_look,2)/Num);
    hRy_mse(noise_lev_index)=sum(sum(hRy_mse_look,2)/Num);
    hRr_mse(noise_lev_index)=sum(sum(hRr_mse_look,2)/Num);
end

if plots_on ==1
    figure, hold on, grid on
    plot(noise_lev,10*log10(hKr_mse),'r')
    plot(noise_lev,10*log10(hKk_mse),'k')
    plot(noise_lev,10*log10(hKy_mse))
    plot(noise_lev,10*log10(min_MSE),'m')
    %plot(noise_lev,10*log10(hK_var),'c')
    %plot(noise_lev,10*log10(hKmmse1_mse),'g')
    plot(noise_lev,10*log10(hKmmsehat_mse),'-c')

    legend('Rayleigh estimate','K-MAP estimate','Output as estimate',...
        'Linear MSE (calculated)','2nd order K-MMSE')

    xlabel('SNR (dB)')
```

```
    ylabel('MSE (dB)')
    title({'K-dist Channel Estimates,'; ['hK var: ' num2str(2*sig_hK^2) ...
        ', hR var: ' num2str(2*sig^2)]})

    figure, hold on, grid on
    plot(noise_lev,10*log10(hRr_mse),'r')
    plot(noise_lev,10*log10(hRk_mse),'k')
    plot(noise_lev,10*log10(hRy_mse))
    plot(noise_lev,10*log10(min_MSE),'m')
    legend('Rayleigh estimate','K-dist estimate',...
        'Output as estimate','MMSE')
    xlabel('SNR (dB)')
    ylabel('MSE (dB)')
    title({'Rayleigh Channel Estimates,'; ['hK var: ' ...
        num2str(2*sig_hK^2),', hR var: ' num2str(2*sig^2)]})

end
```

```
% Newton method
% adopted from course material for Convex Optimization I at Stanford
% University, taught by Professor Stephen Boyd
ALPHA = 0.01;
BETA = 0.5;
MAXITERS =200;% Newton_it;
NTTOL = 1e-9;
GRADTOL = 1e-6;


vals = [];
steps =[];

for iter = 1:MAXITERS
    val =real((y-D*N*r)'/Cw*(y-D*N*r))+r'*r/theta-(2*k-1)*sum(log(r))+...
        n(:,CD_index)'*n(:,CD_index)/2;
    vals = [vals, val];
    grad= -2*real(N'*D'/Cw*y)+2*real(N'*D'/Cw*D*N)*r+2*r/theta-(2*k-1)./r;
    hess = 2*real(N'*D'/Cw*D*N)+2/theta+(2*k-1)*(diag(1./r.^2));
    v = -hess\grad;
```

```
    fprime = grad'*v ;
    if abs(fprime) < NTTOL, break; end;
        t = 1;
        while (min(r+t*v) < 0)
            t = BETA*t;
        end;
        while  ((y-D*N*(r+t*v))'/Cw*(y-D*N*(r+t*v))+(r+t*v)'*(r+t*v)/...
                theta-(2*k-1)*sum(log(r+t*v))+n(:,CD_index)'...
                *n(:,CD_index)/2>val + ALPHA*t*fprime )
            t=BETA*t;
        end;

    r = r+t*v;
    steps = [steps,t];

end;

optval = vals(length(vals));
steps = []; % don't output figures
if isempty(steps)≠1
    figure(3)
    semilogy([0:(length(vals)-2)], vals(1:length(vals)-1)-optval, '-', ...
        [0:(length(vals)-2)], vals(1:length(vals)-1)-optval, 'o');
    xlabel('r'); ylabel('z');

    figure(4)
    plot([1:length(steps)], steps, '-', [1:length(steps)], steps, 'o');
    axis([0, length(steps), 0, 1.1]);
    xlabel('r'); ylabel('z');
end
```

```
function t=t_fcn(betak,etak,k,theta)

Karg=2*sqrt(betak/theta);

t=2*(betak*theta)^(k/2)*(-k*besselk(k,Karg)/betak+etak*besselk(k-1,Karg)/...
    (theta*sqrt(betak*theta))-k*etak*besselk(k-2,Karg)/(betak*theta)+...
    sqrt(betak*theta)*besselk(k+1,Karg)/(theta*betak));
```

Bibliography

[1] Douglas A. Abraham and Anthony P. Lyons. Novel physical interpretations of k-distributed reverberation. *IEEE Journal of Oceanic Engineering*, 27(4):800–812, October 2002.

[2] J.C. Bezdek, R.J. Hathaway, R.E. Howard, C.A. Wilson, and M.P. Windham. Local convergence analysis of a grouped variable version of coordinate descent. *Journal of Optimization Theory and Applications*, 54(3):471–477, 1987.

[3] D.H. Brandwood. A complex gradient operator and its application in adaptive array theory. *IEE Proc. F and H*, 130(1):11–16, 1983.

[4] Paul C. Etter. *Underwater Acoustic Modeling and Simulation.* Spon Press, 2003.

[5] I.S. Gradshteyn and I.M. Ryzhik. *Table of Integrals, Series, and Products.* Academic Press, 7th edition, 2007.

[6] R.J. Hathaway and J.C. Bezdek. Grouped coordinate minimization using Newton's method for inexact minimization in one vector coordinate. *Journal of Optimization Theory and Applications*, 71(3):503–516, 1991.

[7] E. Jakeman. On the statistics of K-distributed noise. *Journal of Physics A: Mathematical and General*, 13:31–48, 1980.

[8] E. Jakeman and P. N. Pusey. A model for non-Rayleigh sea echo. *IEEE Transactions on Antennas and Propagation*, AP-24(1976):806–814, November 1976.

[9] I Joughin and D Percival. Maximum likelihood estimation of K distribution parameters for SAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 1993.

[10] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall PTR, 1993.

[11] J. Preisig. Acoustic propagation considerations for underwater acoustic communications network development. *ACM SIGMOBILE Mobile Comp. Commun. Rev.*, 11(4):2–10, October 2007.

[12] Jan Sijbers, Arnold J. den Dekker, Paul Scheunders, and Dirk Van Dyck. Maximum-likelihood estimation of Rician distribution parameters. *IEEE Transactions on Medical Imaging*, 17(3):357–361, June 1998.

[13] Marvin K. Simon and Mohamed-Slim Alouini. *Digital Communication over Fading Channels.* John Wiley and Sons, 2 edition, 2005.

[14] G. R. Valenzuela and M. B. Laing. On the statistics of sea clutter. NRL Rept., Naval Research Lab, Washington, D.C., 1971.

[15] A. van den Bos. Complex gradient and hessian. *IEE Proceedings -Vision, Image, and Signal Processing*, 141(6), December 1994.

[16] K. D. Ward. Compound representation of high resolution sea clutter. *IET Electronics Letters*, 17(16):561–563, August 1981.

[17] K. D. Ward, R. J. A. Tough, and S. Watts. *Sea Clutter: Scattering, the K Distribution and Radar Performance.* The Institute of Engineering and Technology, London, UK, 2006.