Iury Bertollo Gomes Pôrto

Approximation and Higher Order Statistics for the Kappa-Mu Phase Fading Model

Aproximação e estatísticas de ordem superior para a fase do modelo de desvanecimento Kappa-Mu

Campinas
2012

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Iury Bertollo Gomes Pôrto

Approximation and Higher Order Statistics for the Kappa-Mu Phase Fading Model

Aproximação e estatísticas de ordem superior para a fase do modelo de desvanecimento Kappa-Mu

Masters thesis presented to the School of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of Master in Electrical Engineering. Concentration area: Telecomunications and Telematics.

Dissertação de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Advisor: Michel Daoud Yacoub

Orientador: Michel Daoud Yacoub

Este exemplar corresponde à versão final da tese defendida pelo aluno, e orientada pelo Professor Michel Daoud Yacoub.

____

Campinas
2012

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Pôrto, Iury Bertollo Gomes

P838a Aproximação e estatísticas de ordem superior para a fase do modelo de desvanecimento Kappa-Mu / Iury Bertollo Gomes Pôrto. --Campinas, SP: [s.n.], 2012.

Orientador: Michel Daoud Yacoub.

Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Rádio. 2. Modelos estatísticos. 3. Ajuste de curvas. 4. Aproximaçãoes matemáticas. I. Yacoub, Michel Daoud, 1955-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Approximation and higher order Statistics for the Kappa-Mu phase fading model
Palavras-chave em Inglês: Radio, Statistical models, Curve fitting, Aproximaçãoes mathematical
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora: Ugo Silva Dias, José Cândido Silveira Santos Filho
Data da defesa: 10-08-2012
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Iury Bertollo Gomes Pôrto

Data da Defesa: 10 de agosto de 2012

Título da Tese: "Approximation and Higher Order Statistics for the Kappa-Mu Phase Fading Model"

Prof. Dr. Michel Daoud Yacoub (Presidente): ______________________________

Prof. Dr. Ugo Silva Dias : ______________________________

Prof. Dr. José Candido Silveira Santos Filho: ______________________________

Acknowledgments

I would like to thank everyone who helped making this work possible, in special to

Professor Michel Yacoub, my advisor, for the all the precious guidance provided;

my friend Guilherme Rabelo, whose recommendation was the starting point of this journey;

my friends Tiago Ricciardi and Rafael Ando, for all the valuable conversations;

my other friends, for all the good moments together;

my family, my mother Dilva, my father Natal, my brother Victor and my sister Larissa, for the support;

the School of Electrical and Computer Engineering/University of Campinas (FEEC/UNICAMP), for the very good infrastructure;

all the professors whose classes I had the pleasure to attend to, for the great courses that were offered;

the members of the examining committee, professors Ugo Dias and José Candido, for the suggestions and contributions to the final version of this thesis;

all the people who contributed in any way to my development.

This thesis was supported by the National Council for Scientific and Technological Development (CNPq) under the grant #143629/2010-7.

Abstract

This thesis concerns some important results regarding the phase statistics of the kappa-mu fading model. In particular, the phase crossing rate is obtained in an exact manner. In addition, in order to circumvent the intricacy of the exact formulations, approximate solutions for the following statistics are proposed: phase crossing rate and probability density function of the phase. Furthermore, a simulation methodology is developed so as to validate the formulations. Finally, field data obtained by measurements conducted elsewhere are used to fit the phase statistics for both kappa-mu and generalized Nakagami-m models.

Key-words: Fading models, kappa-mu, Nakagami-m, phase statistics.

Resumo

Esta tese apresenta alguns resultados importantes das estatísticas de fase do modelo de desvanecimento kappa-mu. Em particular, a taxa de cruzamento de fase é obtida de maneira exata. Adicionalmente, para evitar a complexidade das formulações exatas, foram propostas soluções aproximadas para as seguintes estatísticas: taxa de cruzamento de fase e função densidade de probabilidade de fase. Além disso, uma metodologia de simulação foi desenvolvida para validar as formulações. Finalmente, dados de campo obtidos através de medidas conduzidas por outros pesquisadores foram usados para adequar as estatísticas de fase, tanto para o modelo kappa-mu quanto para Nakagami-m generalizado.

Palavras-chave: Modelo de desvanecimento, kappa-mu, Nakagami-m, estatísticas de fase.

List of Figures

List of Acronyms

CPU	Central Processing Unit
FM	Frequency Modulation
IID	Independent and Identically Distributed (Random Variates)
LoS	Line-of-Sight
NLoS	Non-Line-of-Sight
PCR	Phase Crossing Rate
PDF	Probability Density Function
RMS	Root Mean Square
RV	Random Variate

Contents

Introduction

The Mobile Radio Channel

The modeling of the wireless channel is a rather intricate problem. The wireless signal is subjected to various phenomena, such as shadowing and multiple reflections, caused by obstacles in the propagation path. The signal transmitted to and received by mobile stations is affected by a Doppler shift related to the relative velocity between the transmitter and receiver. In addition, changes in the environment greatly affect the wireless transmissions, this occurring with an intensity greater than that in guided communications such as wired and fiber optics transmissions. In this sense, an accurate modeling of the propagation channels is needed for a better system design and performance analysis.

Accounting for all of the interactions suffered by the wireless signal is an impossible task. Hence, the wireless channel is often modeled statistically, with the phase and the envelope of the signal being represented by random variables (RV). The fading process is usually divided in two types: the slow fading, caused by shadowing, and the fast fading, caused by multipath propagation. The envelope of a signal undergoing slow fading is generally well represented by the log-normal distribution [1], whereas several different distributions model the fast fading. Some of those distributions are Rayleigh, Rice, Nakagami-m and κ-μ. As their names imply, the time scale of signal strength variation in the slow fading case is much greater than that of the fast fading.

The Phase Crossing Rate

In order to have a better understanding of the dynamics of the fading process some second order statistics are studied. Such an investigation was pioneered by Rice in his classical paper [2], in which he studied the click noise in FM systems. The second order statistics take in account not only the distribution of the envelope and phase, but also how the signal changes with respect to the time. To compute those statistics, a detailed description of the physical processes is required.

In several fading models, such as Rayleigh, Rice, Nakagami-m, κ-μ and others, the physical process is modeled as the sum of squared gaussian variables. Those Gaussians in turn are modeled as the sum of an infinitude of sinusoidal waves, representing the scattered waves received by the mobile station. Each sinusoidal wave has a phase that is related to the distance traveled by the wave before arriving at the receiver, and a shift in frequency due to the Doppler effect, which is related to its angle of arrival and the velocity of the station [3]. With the signal physically modeled, it is possible to compute the derivatives of the envelope and phase with respect to the time.

One important second order statistics is the phase crossing rate (PCR). It measures the average number of crossings in a single direction (upward or downward) of a given phase. The study of the phase behavior is useful, for instance, for the design of optimal carrier recovery schemes needed in the synchronization subsystem of coherent receivers [4].

Fading Models

The fading models comprised in this work are described next. The main contribution is relative to the phase statistics of the κ-μ model, which in turn requires some understanding of its particular cases, namely Nakagami-m, Rice and Rayleigh [5]. The Generalized Nakagami-m model [6], which isn't a particular case of the κ-μ model, is adjusted to experimental data, so a brief introduction to this model is also given.

Nakagami-m

The Nakagami-m model is one of the most used fading models for modeling the fast fading [7]. It has a wide range of applicability and has been found to fit very well to measurements of the mobile radio channel [8]. While it was first inferred by inspection over measured data [7], a physical model that obtained the exact expression was proposed in [9]. A generalized version is discussed in [6], with the addition of a phase parameter p that models a power imbalance between the in-phase and quadrature components.

Rayleigh

The Rayleigh fading is a fast fading model in which the envelope of the wireless signal follows a Rayleigh distribution. It is well suited for studying non-line-of-sight (NLoS) propagation conditions, in which the received signal is scattered by a large number of objects [10]. It is a special case of both the Nakagami-m and Rice models.

Rice

The Ricean fading models a propagation condition in which the energy of the direct component is larger than that of the scattered components. It adequately models the fast fading under line-of-sight (LoS) propagation [11], yielding better fits under this condition than the Nakagami-m model.

κ-μ

The κ-μ distribution is a general fading distribution that represents the small-scale variations of the fading signal under a LoS condition [5]. As its name implies, it is written in terms of two physical parameters: κ and μ. The parameter $\kappa > 0$ concerns the ratio between the total power of the dominant components and the total power of the scattered waves, whereas the parameter $\mu > 0$ is related to the multipath clustering. It encompasses several important fading distributions as special cases, including Nakagami-m and Rice. Its flexibility renders it suitable to better fit field measurements data in a variety of scenarios, both for low- [5] and high-order statistics [12]. Although several first and second order statistics of the envelope of the κ-μ model have been derived [5,12], the phase crossing rate of this model remains unknown. Furthermore, no closed-form expression has been found for the probability density function (PDF) of the phase [13]. This hinders the practicality of using the κ-μ for studying phase related phenomena.

Contributions

The contributions of this thesis are many-fold:

- The PCR of the κ-μ fading model is found in an exact manner;
- a simple approximation for the κ-μ PCR and phase probability density function is proposed, which yields results that are very close to the exact expression, but that are much faster to compute;
- a simulation methodology to validate the results is presented;
- the PCR is fitted to field data, for both the κ-μ and the Nakagami-m models.

Structure

In the first chapter the κ-μ, Generalized Nakagami-m, Rice and Rayleigh models are revisited. The results presented there are important for the derivations that follow. The second chapter details the derivation of the PCR of the κ-μ model. The third chapter presents the method used to find the approximate function and the proposed approximation for the κ-μ PCR and phase PDF. The fourth chapter comprises a discussion regarding the simulation of the phase of the κ-μ process with some simulated results. Finally, in the fifth chapter the PCR of the κ-μ and Nakagami-m models is adjusted to field data obtained elsewhere.

Chapter 1

Fading Models Revisited

In this chapter, the κ-μ, Rice and Generalized Nakagami-m models are revisited. The main characteristics of these models are discussed. The physical modeling of each fading model is given, along with the mathematical expression of some important statistics. The relationships between these models are also explored.

1.1 The κ-μ fading model

The κ-μ envelope is defined as [5]

$$R^2 = \sum_{i=1}^{\mu}(X_i + p_i)^2 + \sum_{i=1}^{\mu}(Y_i + q_i)^2, \tag{1.1}$$

in which X_i and Y_i are the independent Gaussian random variables resulting from the processes

$$X_i = \sqrt{2\sigma}\sum_{j=1}^{N}\cos(\omega_{ij}t - \phi_{ij}) \tag{1.2}$$

and

$$Y_i = \sqrt{2\sigma}\sum_{j=1}^{N}\sin(\omega_{ij}t - \phi_{ij}). \tag{1.3}$$

In Equations (1.2) and (1.3), ϕ_{ij} is the phase, ω_{ij} is the Doppler shift and N is the number of scatterers in an isotropic environment [3]. Note that $E(X_i) = E(Y_i) = 0$ and $E(X_i^2) = E(Y_i^2) = \sigma^2$. Each X_i and Y_i component can be viewed as a sample of a stochastic process, with auto-correlation [3, 14]

$$\mathbb{R}(\tau) = J_0(2\pi f_d \tau), \tag{1.4}$$

in which $J_0(x)$ is the zeroth-order Bessel function of the first kind, f_d is the maximum Doppler shift and τ is the time delay. The maximum Doppler shift is found as v/λ, with $v_{t,r}$ being the velocity of the transmitter relative to the receiver and λ being the wavelength of the signal. The theoretical power spectrum of such a stochastic process is

$$S(f) = \frac{1}{\pi f_d \sqrt{1 - (f/f_d)^2}}, \tag{1.5}$$

in which f is the frequency in Hertz.

The in-phase and quadrature parts of the signal are given, respectively, as

$$X^2 = \sum_{i=1}^{\mu} (X_i + p_i)^2 \tag{1.6}$$

and

$$Y^2 = \sum_{i=1}^{\mu} (Y_i + q_i)^2. \tag{1.7}$$

The terms p_i and q_i are the mean values of the in-phase and quadrature components of the multipath cluster i. The power of the in-phase and quadrature components are given as

$$p^2 = \sum_{i=1}^{\mu} p_i^2 \qquad \text{and} \qquad q^2 = \sum_{i=1}^{\mu} q_i^2, \tag{1.8}$$

and the total power of the dominant components is

$$d^2 = \sum_{i=0}^{\mu} \left(p_i^2 + q_i^2\right). \tag{1.9}$$

The parameter κ is the ratio of the total power of the dominant components and the power of the scattered waves,

$$\kappa = d^2/2\mu\sigma^2. \tag{1.10}$$

The variance of the Gaussian processes can be written as a function of the parameters κ and μ and the root mean square (RMS) value of the envelope $\hat{r} = \sqrt{E[R^2]}$,

$$\sigma^2 = \frac{\hat{r}^2}{2\mu(1+\kappa)}. \tag{1.11}$$

In [15], a phase parameter $\phi \triangleq \arg(p + jq)$ was introduced. It allows the terms p and q to be written as a function of κ, μ and ϕ,

$$p = \sqrt{\frac{\kappa}{1+\kappa}}\hat{r}\cos(\phi) \tag{1.12}$$

and

$$q = \sqrt{\frac{\kappa}{1+\kappa}}\hat{r}\sin(\phi). \tag{1.13}$$

Denote X or Y by Z and p or q by λ as required. The PDF of Z is given as [15]

$$f_Z(z) = \frac{|z|^{\frac{\mu}{2}} \exp\left(-\frac{(z-\lambda)^2}{2\sigma^2}\right) I_{\frac{\mu}{2}-1}\left(\frac{|\lambda z|}{\sigma^2}\right)}{2\sigma^2 |\lambda|^{\frac{\mu}{2}-1} \cosh\left(\frac{\lambda z}{\sigma^2}\right)}. \tag{1.14}$$

Observe that although the probability distribution of the in-phase or quadrature components is known, the complex envelope is not. Equation (1.14) was achieved by combining the PDF of $|Z|$, the absolute value of the RV Z, with the knowledge of the PDF of Z at a particular case.

More details on this derivation are found in [15]. The joint PDF of the normalized envelope P and the phase Θ of the κ-μ model is given as [15]

$$
\begin{aligned}
f_{P,\Theta}(\rho,\theta) &= \frac{1}{2}\mu^2\kappa^{1-\frac{\mu}{2}}(1+\kappa)^{\frac{\mu+2}{2}}\rho^{\mu+1}|\sin 2\theta|^{\frac{\mu}{2}}|\sin 2\phi|^{1-\frac{\mu}{2}} \\
&\times \exp\left(-\mu(1+\kappa)\rho^2-\kappa\mu+2\mu\sqrt{\kappa(1+\kappa)}\rho\cos(\theta-\phi)\right) \\
&\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\cos\theta\cos\phi|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho\cos\theta\cos\phi\right) \\
&\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\sin\theta\sin\phi|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho\sin\theta\sin\phi\right),
\end{aligned}
\tag{1.15}
$$

$\rho \geq 0$, $-\pi < \theta \leq \pi$.

In this work P denotes the envelope normalized by its RMS value, rather than the scaled version found in [15]. Two particular cases of the joint envelope were also presented in [15], for $\phi = \pm n\pi$ and for $\phi = \pm\frac{(2n+1)\pi}{2}$. They are

$$
\begin{aligned}
f_{P,\Theta}(\rho,\theta)|_{\phi=\pm n\pi} &= \frac{1}{\Gamma\left(\frac{\mu}{2}\right)}\mu^{1+\frac{\mu}{2}}\kappa^{\frac{1}{2}-\frac{\mu}{4}}(1+\kappa)^{\frac{1}{2}+\frac{3\mu}{4}}\rho^{\frac{3\mu}{2}}|\sin\theta|^{\mu-1}|\cos\theta|^{\frac{\mu}{2}} \\
&\times \exp\left(-\mu(1+\kappa)\rho^2-\kappa\mu+2\mu\sqrt{\kappa(1+\kappa)}\rho\cos\theta\cos\phi\right) \\
&\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\cos\theta|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho\cos\theta\right)
\end{aligned}
\tag{1.16}
$$

and

$$
\begin{aligned}
f_{P,\Theta}(\rho,\theta)|_{\phi=\pm\frac{(2n+1)\pi}{2}} &= \frac{1}{\Gamma\left(\frac{\mu}{2}\right)}\mu^{1+\frac{\mu}{2}}\kappa^{\frac{1}{2}-\frac{\mu}{4}}(1+\kappa)^{\frac{1}{2}+\frac{3\mu}{4}}\rho^{\frac{3\mu}{2}}|\cos\theta|^{\mu-1}|\sin\theta|^{\frac{\mu}{2}} \\
&\times \exp\left(-\mu(1+\kappa)\rho^2-\kappa\mu+2\mu\sqrt{\kappa(1+\kappa)}\rho\sin\theta\sin\phi\right) \\
&\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\sin\theta|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho\sin\theta\right),
\end{aligned}
\tag{1.17}
$$

in which $\Gamma(\cdot)$ is the complete gamma function [16]. The PDF of the normalized envelope P can be found by integrating (1.15) with respect to Θ from $-\pi$ to π or it can be derived in a straightforward fashion as shown in [5]. It is given as

$$
f_P(\rho) = \frac{2\mu(1+\kappa)^{\frac{\mu+1}{2}}}{\kappa^{\frac{\mu-1}{2}}\exp(\mu\kappa)}\rho^{\mu}\exp\left[-\mu(1+\kappa)\rho^2\right]I_{\mu-1}\left[2\mu\sqrt{\kappa(1+\kappa)}\rho\right],
\tag{1.18}
$$

$\rho \geq 0$.

The phase PDF is found by integrating either one of Equations (1.15), (1.16) and (1.17) with respect to ρ from 0 to ∞. No closed-form expression was found for this integral and it can only be calculated numerically.

1.2 The Generalized Nakagami-m model

The envelope of the generalized Nakagami-m model [6] is calculated as

$$
R^2 = \sum_{i=1}^{m_X} X_i^2 + \sum_{i=1}^{m_Y} Y_i^2,
\tag{1.19}
$$

in which X_i and Y_i are Gaussian random variables with the same variance and zero mean and m_X and m_Y are the number of multipath clusters in the in-phase and quadrature components. The phase parameter $-1 < p < 1$ measures the phase imbalance and is given as

$$p = \frac{m_X - m_Y}{m_X + m_Y}, \tag{1.20}$$

in which m_X and m_Y relates to the m parameter of the original Nakagami-m model by

$$2m = m_X + m_Y. \tag{1.21}$$

The fading is said to be balanced if $p = 0$, in which case the generalized Nakagami-m model reverts back to its classical formulation. With this condition, it is a special case of the κ-μ model when $\kappa \to 0$.

The generalized joint phase-envelope PDF is given as [6]

$$f_{R,\Theta}(r,\theta) = \frac{m^m |\sin\theta\cos\theta|^{m-1} r^{2m-1}}{\Omega^m \Gamma\left(\frac{1+p}{2}m\right)\Gamma\left(\frac{1-p}{2}m\right)|\tan\theta|^{pm}} \exp\left(-\frac{mr^2}{\Omega}\right), \tag{1.22}$$

$r \geq 0$, $-\pi < \theta \leq \pi$. The parameter Ω is the mean value of the square of the envelope,

$$\Omega = E[R^2]. \tag{1.23}$$

The envelope is given as

$$f_R(r) = \frac{2m^m r^{2m-1}}{\Omega^m \Gamma(m)} \exp\left(-\frac{mr^2}{\Omega}\right). \tag{1.24}$$

The phase PDF is

$$f_\Theta(\theta) = \frac{\Gamma(m)}{2^m \Gamma\left(\frac{1+p}{2}m\right)\Gamma\left(\frac{1-p}{2}m\right)} \frac{|\sin 2\theta|^{m-1}}{|\tan\theta|^{pm}}. \tag{1.25}$$

The PCR of the generalized Nakagami-m model is also found in an exact manner [6] as

$$N_\Theta(\theta) = \frac{\sqrt{\pi} f_d |\sin 2\theta|^{m-1} |\tan\theta|^{-pm} \Gamma\left(m - \frac{1}{2}\right)}{2^{m+\frac{1}{2}} \Gamma\left(\frac{1+p}{2}m\right)\Gamma\left(\frac{1-p}{2}m\right)}. \tag{1.26}$$

1.3 Rice

The complex envelope of the Ricean process is given as

$$\mathbb{Z} = (X + p) + j(Y + q), \tag{1.27}$$

in which X and Y are Gaussian variates with zero mean and variance σ^2. Note that the Rice process is a particular case of the κ-μ process when $\mu = 1$ and $\kappa = k$. The phase-envelope joint PDF of Rice is obtained by substituting $\mu = 1$ in (1.15)

$$f_{P,\Theta}(\rho,\theta) = \frac{1+k}{\pi} \rho \exp\left(-(1+k)\rho^2 + 2\sqrt{k(1+k)}\rho\cos(\theta - \phi) - k\right), \tag{1.28}$$

$\rho \geq 0$, $-\pi < \theta \leq \pi$. As in (1.15), ρ denotes the RV obtained by normalizing the envelope r by its RMS value, $\hat{r}$.

The normalized envelope PDF can be calculated directly by substituting $\mu = 1$ in (1.18)

$$f_P(\rho) = 2(1+k)\rho \exp\left[-(1+k)\rho^2 - k\right] I_0 \left[2\sqrt{k(1+k)}\rho\right], \tag{1.29}$$

$\rho \geq 0$.

The phase PDF is calculating by integrating (1.28) with respect to ρ from $-\infty$ to ∞,

$$f_\Theta(\theta) = \frac{\exp(-k)}{2\pi}(1 + \sqrt{k\pi}\exp(k\cos^2(\theta-\phi))\cos(\theta-\phi)[1+\mathrm{erf}(\sqrt{k}\cos(\theta-\phi))]), \tag{1.30}$$

$-\pi < \theta \leq \pi$, in which $k = (p^2+q^2)/(2\sigma^2)$ and $\mathrm{erf}(\cdot)$ denotes the error function [17].

The PCR of the Rice fading model is [18]

$$N(\theta) = \frac{f_d}{2\sqrt{2}} \exp\left(-\frac{d^2\sin^2(\theta-\phi)}{2\sigma^2}\right)\left[1+\mathrm{erf}\left(\frac{d\cos(\theta-\phi)}{\sqrt{2}\sigma}\right)\right], \tag{1.31}$$

in which d is defined as

$$d^2 \triangleq p^2 + q^2. \tag{1.32}$$

1.4 Rayleigh

The Rayleigh distribution is a particular case of the κ-μ, Nakagami-m and Rice distributions. It is obtained by setting $\kappa = 0$ and $\mu = 1$ in the κ-μ model, $m = 1$ and $p = 0$ in the generalized Nakagami-m model and $k = 0$ in the Rice fading model. The complex envelope of the Rayleigh distribution is given as

$$\mathbb{Z} = X + jY, \tag{1.33}$$

in which X and Y are Gaussian variates with zero mean and variance σ^2. In the Rayleigh distribution, the phase and the envelope are independent. The PDF of the normalized Rayleigh envelope is

$$f_P(\rho) = 2\rho\exp(-\rho^2), \tag{1.34}$$

$r \geq 0$, and the phase PDF is constant in the $(-\pi, \pi]$ interval,

$$f_\Theta(\theta) = \frac{1}{2\pi}, \tag{1.35}$$

$-\pi < \theta \leq \pi$. The PCR of Rayleigh is

$$N_\Theta(\theta) = \frac{f_d}{2\sqrt{2}}. \tag{1.36}$$

All the expressions for Rayleigh can be obtained by making the corresponding substitutions in the expressions of either the κ-μ, Nakagami-m or Rice models.

Chapter 2

The κ-μ Phase Crossing Rate

The PCR is a second order statistics that measures the average number of upward (or downward) crossings per second of a given phase θ. The PCR of a continuous process is given as

$$N_\Theta(\theta) = \int\limits_0^\infty \dot{\theta} f_{\theta,\dot{\theta}}(\theta, \dot{\theta}) d\dot{\theta}. \tag{2.1}$$

In which θ is the phase of the process and $\dot{\theta}$ the time derivative of the phase. In order to obtain the PCR, it is thus necessary to know the joint distribution $f_{\theta,\dot{\theta}}(\theta, \dot{\theta})$.

2.1 Derivation of the κ-μ PCR

The derivation of the PCR first involves the derivation of $f_{\Theta,\dot{\Theta}}(\theta, \dot{\theta})$. This can be accomplished in three steps. First the joint distribution of $f_{X,Y,\dot{X},\dot{Y}}(x, y, \dot{x}, \dot{y})$ is found. Then, the Cartesian $X, Y, \dot{X}, \dot{Y}$ variables are changed to the polar coordinates $R, \dot{R}, \Theta, \dot{\Theta}$. To simplify the notation, the joint distribution $f_{R,\dot{R},\Theta,\dot{\Theta}}(r, \dot{r}, \theta, \dot{\theta})$ is normalized with respect to the RMS values of R and $\dot{R}$, yielding the joint distribution $f_{P,\dot{P},\Theta,\dot{\Theta}}(\rho, \dot{\rho}, \theta, \dot{\theta})$. Finally, the resulting expression is integrated over the domains of P and $\dot{P}$, resulting in $f_{\Theta,\dot{\Theta}}(\theta, \dot{\theta})$. Unfortunately, no closed-form expression of the latter was found and hence an intermediate step was introduced, in which $N_\Theta(\theta, \rho)$ is found, so that $N_\Theta(\theta) = \int_0^\infty N_\Theta(\theta, \rho) d\rho$. This will be further explained in the next subsections.

2.1.1 Proof that Z and $\dot{Z}$ are independent

In this section, Z denotes either X or Y and λ denotes either p or q, as required. The PDF of Z is given in Equation (1.14). From Equations (1.6) or (1.7),

$$Z^2 = \sum_{i=1}^{\mu} (Z_i + \lambda_i)^2. \tag{2.2}$$

By differentiating both sides of (2.2) with respect to time and rearranging the terms, it is possible to write

$$\dot{Z} = \frac{\sum_{i=1}^{\mu}(Z_i + \lambda_i)\dot{Z}_i}{Z}. \tag{2.3}$$

Let $\dot{Z}|Z$ represent the random variable $\dot{Z}$ conditioned on the value of all Z_i , $i = 1, ..., \mu$. In [1], it is shown that, for an isotropic environment, $\dot{Z}_i$ is a Gaussian random variable with zero mean and variance

$$\dot{\sigma}^2 = 2\pi^2 f_d^2 \sigma^2. \tag{2.4}$$

Because $\dot{Z}|Z_i$ is the sum of μ Gaussian random variables, it is itself a Gaussian random variable, and its PDF can be fully characterized by its first and second moments. The moments of each individual Gaussian component $\dot{Z}_i$ are

$$E[\dot{Z}_i] = 0 \tag{2.5}$$

and

$$E[\dot{Z}_i \dot{Z}_j] = \begin{cases} \dot{\sigma}^2 & \text{if } i = j \\ 0 & \text{elsewise} \end{cases}. \tag{2.6}$$

The mean value of $\dot{Z}|Z$ is

$$E[\dot{Z}|Z] = \frac{\sum\limits_{i=1}^{\mu} (Z_i + \lambda_i) E[\dot{Z}_i]}{Z} = 0, \tag{2.7}$$

and its second moment is

$$E[\dot{Z}^2|Z] = \frac{\sum\limits_{i=1}^{\mu}\sum\limits_{j=1}^{\mu} (Z_i + \lambda_i)(Z_j + \lambda_j) E[\dot{Z}_i \dot{Z}_j]}{Z^2} = \frac{\sum\limits_{i=1}^{\mu} (Z_i + \lambda_i)^2 E[\dot{Z}_i^{\,2}]}{Z^2} = \dot{\sigma}^2. \tag{2.8}$$

The PDF of $\dot{Z}|Z$ is then

$$f_{\dot{Z}|Z}(\dot{z}|z) = \frac{1}{\sqrt{2\pi\dot{\sigma}^2}} \exp\left(-\frac{\dot{z}^2}{2\dot{\sigma}^2}\right). \tag{2.9}$$

Since $f_{\dot{Z}|Z}(\dot{z}|z)$ is not a function of z, $f_{\dot{Z}|Z}(\dot{z}, z) = f_{\dot{Z}}(\dot{z})$, which means that Z is independent of $\dot{Z}$.

2.1.2 Joint PDF of $X, \dot{X}, Y, \dot{Y}$

Because X and Y are independent processes and because X is independent of $\dot{X}$ and Y is independent of $\dot{Y}$, the variables $X, \dot{X}, Y, \dot{Y}$ are jointly independent. The joint PDF can be calculated by simply multiplying the marginal distributions in (1.14) and (2.9), substituting X or Y for Z and $\dot{X}$ or $\dot{Y}$ for $\dot{Z}$, as required. That is,

$$f_{X,\dot{X},Y,\dot{Y}}(x, \dot{x}, y, \dot{y}) = \\ \frac{|xy|^{\frac{\mu}{2}}}{8\pi\sigma^4\dot{\sigma}^2|pq|^{\frac{\mu}{2}-1}} \exp\left(-\frac{(x-p)^2+(y-q)^2}{2\sigma^2}\right) \exp\left(-\frac{\dot{x}^2+\dot{y}^2}{2\dot{\sigma}^2}\right) I_{\frac{\mu}{2}-1}\left(\frac{|px|}{\sigma^2}\right) I_{\frac{\mu}{2}-1}\left(\frac{|qy|}{\sigma^2}\right) \operatorname{sech}\left(\frac{px}{\sigma^2}\right) \operatorname{sech}\left(\frac{qy}{\sigma^2}\right). \tag{2.10}$$

2.1.3 Joint PDF of $P, \dot{P}, \Theta, \dot{\Theta}$

In this section, P denotes the normalized envelope. First, a variable transformation is made from the Cartesian coordinates (X, Y) to the polar coordinates (R, Θ). The transformation is

$$f_{R,\dot{R},\Theta,\dot{\Theta}}(r, \dot{r}, \theta, \dot{\theta}) = |J| f_{X,\dot{X},Y,\dot{Y}}(x, \dot{x}, y, \dot{y}), \tag{2.11}$$

in which J is the Jacobian of the transformation. X and Y can be written as functions of R and Θ,

$$\begin{aligned} x &= r\cos\theta, & (2.12) \\ y &= r\sin\theta. & (2.13) \end{aligned}$$

Therefore

$$\begin{aligned} \dot{x} &= \dot{r}\cos\theta - r\dot{\theta}\sin\theta, & (2.14) \\ \dot{y} &= \dot{r}\sin\theta + r\dot{\theta}\cos\theta. & (2.15) \end{aligned}$$

The Jacobian of this transformation is easily found to be $|J| = r^2$. Thus

$$\begin{aligned} &f_{R,\dot{R},\Theta,\dot{\Theta}}(r, \dot{r}, \theta, \dot{\theta}) = \\ &\tfrac{r^{\mu+2}|\sin 2\theta|^{\frac{\mu}{2}}}{2^{3+\frac{\mu}{2}}\pi\sigma^4\dot{\sigma}^2|pq|^{\frac{\mu}{2}-1}} \exp\left(-\tfrac{r^2-2r(p\cos\theta+q\sin\theta)+p^2+q^2}{2\sigma^2}\right) \exp\left(-\tfrac{\dot{r}^2+r^2\dot{\theta}^2}{2\dot{\sigma}^2}\right) \\ &\times I_{\frac{\mu}{2}-1}\left(\tfrac{|pr\cos\theta|}{\sigma^2}\right) I_{\frac{\mu}{2}-1}\left(\tfrac{|qr\sin\theta|}{\sigma^2}\right) \operatorname{sech}\left(\tfrac{pr\cos\theta}{\sigma^2}\right) \operatorname{sech}\left(\tfrac{qr\sin\theta}{\sigma^2}\right). \end{aligned} \tag{2.16}$$

The expression (2.16) contains the variances σ^2 and $\dot{\sigma}^2$, which is an inconvenience. Ideally, this expression should be given only in terms of the parameters κ, μ and ϕ. It is possible to get rid of this unwanted terms by normalizing the envelope r with respect to its RMS value $\hat{r}$. The normalized envelope is

$$\rho = \frac{r}{\hat{r}}, \tag{2.17}$$

in which the term $\hat{r}$ is defined in (1.11). This transformation is performed in a similar fashion as (2.11), with the Jacobian being $|J| = \hat{r}^2$. After making the necessary algebraic manipulations, the desired joint PDF is obtained

$$\begin{aligned} &f_{P,\dot{P},\Theta,\dot{\Theta}}(\rho, \dot{\rho}, \theta, \dot{\theta}) = \frac{\kappa^{1-\frac{\mu}{2}}\mu^3}{4\pi^3 f_d^2}\rho^{\mu+2} \\ &\times|\sin 2\theta|^{\frac{\mu}{2}}|\sin 2\phi|^{1-\frac{\mu}{2}} \exp\left(-\tfrac{\mu(1+\kappa)}{2\pi^2 f_d^2}(\dot{\rho}^2 + \rho^2\dot{\theta}^2)\right) \\ &\times\exp\left(-\mu(1+\kappa)\rho^2 - \kappa\mu + 2\mu\sqrt{\kappa(1+\kappa)}\rho\cos(\theta-\phi)\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\cos\theta\cos\phi|\right) \operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\cos\theta\cos\phi|\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\sin\theta\sin\phi|\right) \operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\sin\theta\sin\phi|\right). \end{aligned} \tag{2.18}$$

2.1.4 The Phase Crossing Rate

The next step in the derivation of the PCR would be finding $f_{\Theta,\dot{\Theta}}(\theta,\dot{\theta})$, which is obtainable by integrating (2.18) with respect to ρ and $\dot{\rho}$. However, this integration does not produce a closed-form expression, which means that the PCR can only be achieved by numerical calculation. To reduce the number of integrations, as an intermediate step the PCR is calculated as a function of ρ and θ, *i.e.*

$$N_\Theta(\rho,\theta) \triangleq \int_{\dot{\Theta}}\int_{\dot{P}} \dot{\theta} f_{P,\dot{P},\Theta,\dot{\Theta}}(\rho,\dot{\rho},\theta,\dot{\theta})d\dot{\rho}d\dot{\theta}, \tag{2.19}$$

for which there is a closed-form expression,

$$\begin{aligned} N_\Theta(\rho,\theta) &= \frac{\sqrt{\pi} f_d}{2\sqrt{2}}\mu^{3/2}\kappa^{1-\frac{\mu}{2}}(1+\kappa)^{\frac{\mu+1}{2}}\rho^\mu |\sin 2\theta|^{\frac{\mu}{2}}|\sin 2\phi|^{1-\frac{\mu}{2}} \\ &\times \exp\left(-\mu(1+\kappa)\rho^2 - \kappa\mu + 2\mu\sqrt{\kappa(1+\kappa)}\rho\cos(\theta-\phi)\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\cos\theta\cos\phi|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\cos\theta\cos\phi|\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\sin\theta\sin\phi|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho|\sin\theta\sin\phi|\right). \end{aligned} \tag{2.20}$$

Finally, the PCR is found by integrating $N_\Theta(\rho,\theta)$ over the domain of ρ,

$$N_\Theta(\theta) = \int_0^\infty N_\Theta(\rho,\theta)d\rho. \tag{2.21}$$

2.2 Special Cases

As discussed before, the κ-μ distribution encompasses several other distributions, including Nakagami-m and Rice. This section shows the compatibility of Equation (2.20) with those models.

2.2.1 Rice

The Rice fading model is obtained from the κ-μ fading model by setting $\mu = 1$. After using the identity

$$I_{-\frac{1}{2}}(x) = \sqrt{2/(\pi x)}\cosh(x) \tag{2.22}$$

in (2.20) and making the necessary algebraic manipulations, we arrive at

$$\begin{aligned} N_\Theta(\rho,\theta) &= \frac{f_d}{\sqrt{2\pi}}\sqrt{1+\kappa} \\ &\times \exp\left(-(1+\kappa)\rho^2 - \kappa + 2\sqrt{\kappa(1+\kappa)}\rho\cos(\theta-\phi)\right). \end{aligned} \tag{2.23}$$

Integrating Equation (2.23) with respect to ρ leads to

$$N_\Theta(\theta) = \frac{f_d}{2\sqrt{2}}\exp\left(-\kappa\sin^2(\theta-\phi)\right)\left[1+\operatorname{erf}\left(\sqrt{\kappa}\cos(\theta-\phi)\right)\right]. \tag{2.24}$$

After substituting (1.10) in (2.24) Equation (1.31) is obtained.

2.2.2 Nakagami-m

The Nakagami-m model is a special case of the κ-μ model when $\kappa \to 0$. It can be shown that

$$\lim_{x\to 0}\left(x^{\frac{1}{2}-\frac{m}{4}} I_{\frac{\mu}{2}-1}(ax^{1/2})\right) = \frac{(a/2)^{\frac{m}{2}-1}}{\Gamma\left(\frac{m}{2}\right)}. \tag{2.25}$$

Using Equation (2.25), the limit of 2.20 when $\kappa \to 0$

$$N_\Theta(\rho,\theta) = \lim_{\kappa\to 0} N_\Theta(\rho,\theta)_{\kappa-\mu} \tag{2.26}$$

can be calculated as

$$N_\Theta(\rho,\theta) = \frac{\sqrt{\pi} f_d 2^{\frac{1}{2}-\mu}}{\Gamma^2\left(\frac{\mu}{2}\right)} \mu^{\mu-\frac{1}{2}} \rho^{2\mu-2} |\sin 2\theta|^{\mu-1} \exp(-\mu\rho^2). \tag{2.27}$$

After performing the integration we arrive at the expression

$$N_\Theta(\theta) = \frac{\sqrt{\pi} f_d |\sin 2\theta|^{\mu-1} \Gamma\left(\mu - \frac{1}{2}\right)}{2^{\mu+\frac{1}{2}} \Gamma^2\left(\frac{\mu}{2}\right)}, \tag{2.28}$$

which is the exact PCR of the Nakagami-m fading process [19], with $\mu = m$.

2.3 Sample plots of the κ-μ PCR

In this section some sample plots of this new statistics are presented. They should provide the reader some familiarity with the general shape of the PCR, which is very similar to the κ-μ phase PDF, as in [15]. In Figure 2.1 it can be seen that as κ increases the PCR becomes more impulsive. Figure 2.2 shows how the PCR changes with varying μ. In particular, if $\mu > 1$ the PCR is always zero at integer multiples of $\pi/2$, whereas if $\mu < 1$ the PCR goes to infinity at those points. This means that when $\mu \neq 1$ the PCR can be divided in four different lobes. As shown in Figure 2.3, the parameter ϕ determines where the global maximum or minimum will occur. If ϕ is an integer multiple of $\pi/2$, the PCR will be symmetric in relation to ϕ. If ϕ is an odd multiple of $\pi/4$ the maximum will occur at $\theta = \phi$. Otherwise, the maximum will generally occur in the neighborhood of ϕ, except in the cases in which ϕ is very close to $n\pi/2$. This behavior is depicted in Figure 2.4.

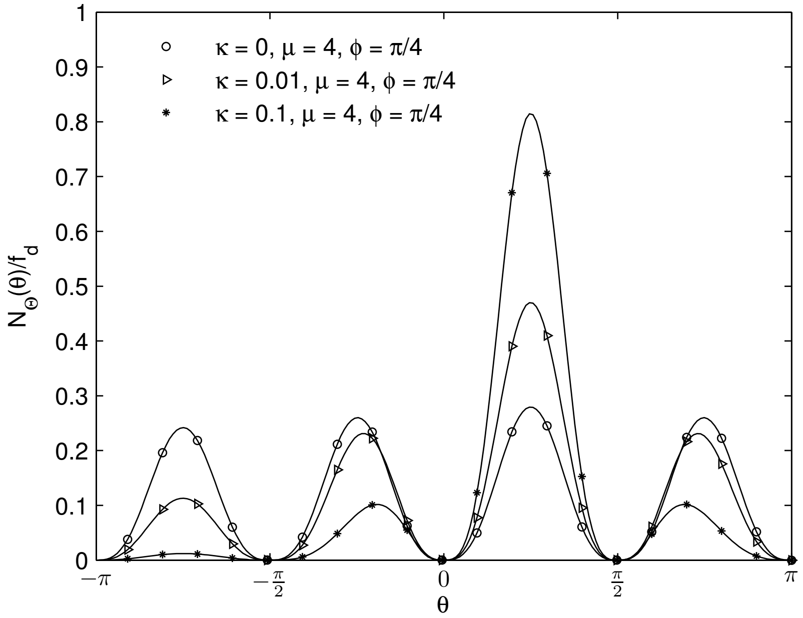


Figure 2.1: Sample plots of the PCR of the κ-μ model for changing κ.

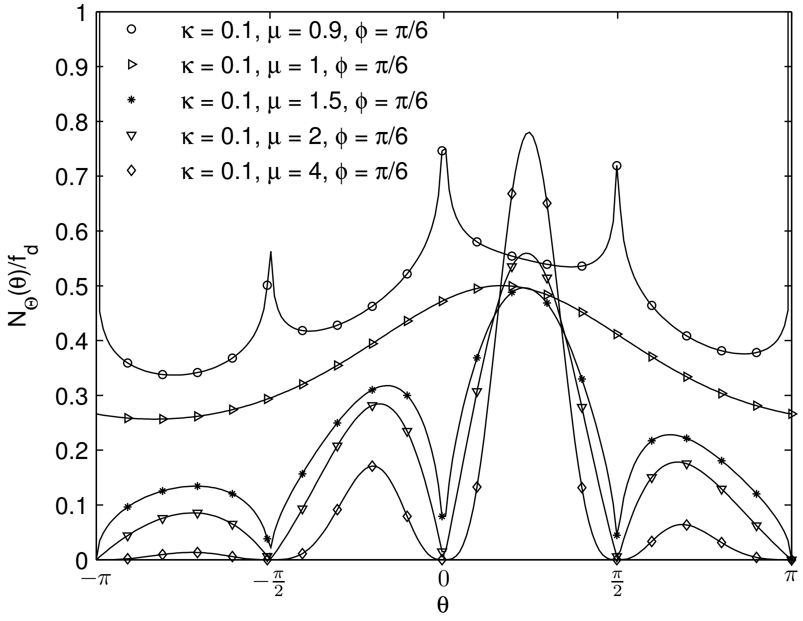


Figure 2.2: Sample plots of the PCR of the κ-μ model for changing μ.

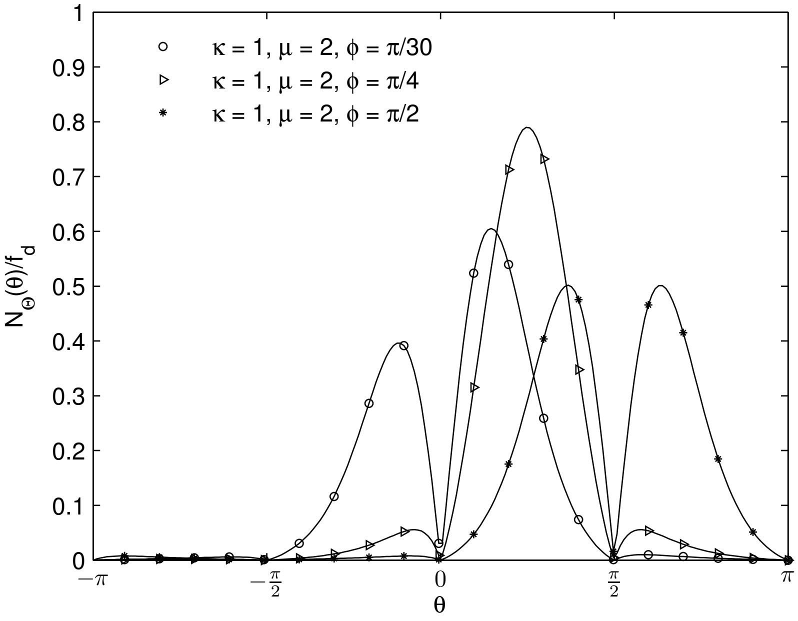


Figure 2.3: Sample plots of the PCR of the κ-μ model for changing ϕ.

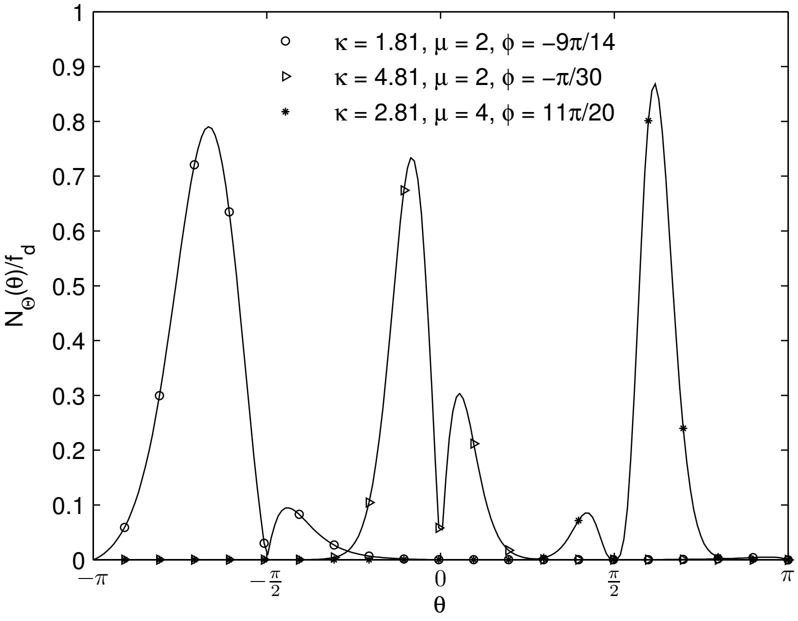


Figure 2.4: Sample plots of the PCR of the κ-μ model.

2.4 Conclusions

The main contribution of this chapter is the derivation of the κ-μ PCR. A proof of the independence of the in-phase or quadrature components of the κ-μ process and its corresponding time derivative is also provided. Moreover, sample plots of the κ-μ PCR illustrate how the parameters κ, μ and ϕ change the shape of the curves.

Chapter 3

Approximations to the κ-μ PCR and Phase PDF

In many applications, some statistics of the model used to study the signal propagation needs to be evaluated several times. For instance, finding an optimal set of parameters to fit experimental data is a non-linear optimization problem that requires multiple evaluations of the statistics being considered. The lack of a closed-form expression for the PCR and the phase PDF of the κ-μ model hinders the practicality of its use to study phase related phenomena. A suitable simpler approximation can therefore make the model more readily usable. This chapter presents the derivation of such an approximation for the PCR, which is then refined. A similar approach also provides an useful approximation to the phase PDF.

3.1 Approximate PCR

In general, it is possible to approximate a function by truncating its Taylor series expansion [20]. Since the PCR, given by Equation (2.21), is in an integral form, the terms of its Taylor expansion will themselves be in an integral form, which does not reduce the complexity as desired. Instead, if the integrand is expanded and truncated prior to the integration, a more suitable approximation is found, with the terms of the expansion having a closed-form expression. Unfortunately it was not possible to find an analytic solution to that integral. A workaround to address this problem is discussed here. First, the interval of integration is changed, as follows

$$N_\Theta(\theta) = \int_0^1 N_{P,\Theta}(x,\theta)dx + \int_1^\infty N_{P,\Theta}(x,\theta)dx. \tag{3.1}$$

Next, the variable x of the second integral is changed to $y = 1/x$, so that $y = 0$ when $x = \infty$, $y = 1$ when $x = 1$ and $dx = -\frac{dy}{y^2}$. Accordingly,

$$N_\Theta(\theta) = \int_0^1 N_{P,\Theta}(x,\theta)dx + \int_0^1 \frac{N_{P,\Theta}\left(\frac{1}{y},\theta\right)}{y^2}dy. \tag{3.2}$$

The dummy variable y is then changed back to x and both integrals are regrouped under a single integral from 0 to 1.

$$N_\Theta(\theta) = \int_0^1 \left(N_{P,\Theta}(x,\theta) + \frac{N_{P,\Theta}\left(\frac{1}{x},\theta\right)}{x^2}\right)dx. \tag{3.3}$$

Finally, the integrand can be approximated by a truncated Taylor series around x, with θ fixed. When this method is applied to Equation (2.20) an analytic solution if found. Equation (3.4) shows the result of that approximation for the Taylor polynomial of degree 1 around $\rho_0 = 1$.

$$\begin{aligned}
N_\Theta(\theta)_{app1} &= \frac{3f_d\sqrt{\pi}}{2\sqrt{2}} \frac{(1+\kappa)^{\frac{1+\mu}{2}}}{\kappa^{\frac{\mu}{2}-1}} \mu^{\frac{3}{2}} |\sin 2\theta|^{\frac{\mu}{2}} |\sin 2\phi|^{1-\frac{\mu}{2}} \\
&\times \exp\left(-\mu(1+2\kappa) + 2\mu\sqrt{\kappa(1+\kappa)}\cos(\theta-\phi)\right) \\
&\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\cos\theta\cos\phi|\right) \operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}|\cos\theta\cos\phi|\right) \\
&\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\sin\theta\sin\phi|\right) \operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}|\sin\theta\sin\phi|\right)
\end{aligned} \tag{3.4}$$

Numerical simulations have shown that although the shape of Equation (3.4) is very similar to the shape of the exact PCR, the resulting curves are scaled by an unknown factor. By multiplying (3.4) by a conveniently chosen factor, a very good approximation can be found. Define $N^*(\theta)$ as

$$\begin{aligned}
&N^*_\Theta(\theta) = \\
&|\sin 2\theta|^{\frac{\mu}{2}} |\sin 2\phi|^{1-\frac{\mu}{2}} \exp\left(2\mu\sqrt{\kappa(1+\kappa)}\cos(\theta-\phi)\right) \\
&\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\cos\theta\cos\phi|\right) \operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}|\cos\theta\cos\phi|\right) \\
&\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\sin\theta\sin\phi|\right) \operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}|\sin\theta\sin\phi|\right).
\end{aligned} \tag{3.5}$$

Note that Equation (3.5) is Equation (3.4) multiplied by a function of κ and μ. We want a scalar value C^* so that $C^* \cdot N^*_{max} = N_{max}$. Since the shape of $N^*_\Theta(\theta)$ is closely related to the shape of $N_\Theta(\theta)$, the maximum of both functions will occur at approximately the same argument and we can use $\theta_{max} \approx \theta^*_{max}$. The approximate PCR is found by the following method:

1. Find $\theta^*_{max} = \arg\max\{N^*_\Theta(\theta)\}$
2. Use θ^*_{max} to compute $N_\Theta(\theta^*_{max})$ numerically as an approximation for $\max\{N_\Theta(\theta)\}$
3. Compute $C^* = \frac{N_\Theta(\theta^*_{max})}{\max\{N^*_\Theta(\theta)\}}$
4. The approximate PCR is $N_\Theta(\theta)_{app} = C^* \cdot N^*_\Theta(\theta)$

This approximation will only work if $\mu \geq 1$. If $\mu < 1$, the PCR admits no point of maximum, as it goes to infinity as θ approaches $n\pi/2$, for n integer. Instead, in this case the approximation can be alternatively normalized by the points of minimum.

The advantage of the approximation is that it requires the computation of just one numerical integral for each curve, instead of one for each point used in the exact solution.

If $\mu \neq 1$ the PCR will have four lobes, going either to 0 or ∞ as θ approaches $n\pi/2$. This suggests one enhancement to the approximation: by normalizing the PCR in a piecewise manner, a tighter fit can be achieved. The PCR is divided in the following four intervals: $(-\pi, -\pi/2]$, $(-\pi/2, 0]$, $(0, \pi/2]$ and $(\pi/2, \pi]$. Within each interval, the approximation previously described is used, using the local maximum or minimum to normalize the approximate PCR. The trade-off is some added complexity, as this enhanced approximation demands the evaluation of four numerical integrals instead of one.

3.1.1 Sample Plots of the Approximate PCR

Figures 3.1-3.4 show samples curves of the approximation resulting from the normalization with respect global maximum or minimum, plotted with the exact PCR. In comparison, Figures 3.5-3.8 show the plots of the piecewise approximation. The advantage of using the piecewise approximation is evident for low values of κ, specially when $\mu < 1$, as illustrated by Figures 3.1, 3.3, 3.5 and 3.7. In this case, the piecewise approximation provides very good fits, even if the normalization by the global extremum is insufficient.

Approximation by Global Extremum Normalization

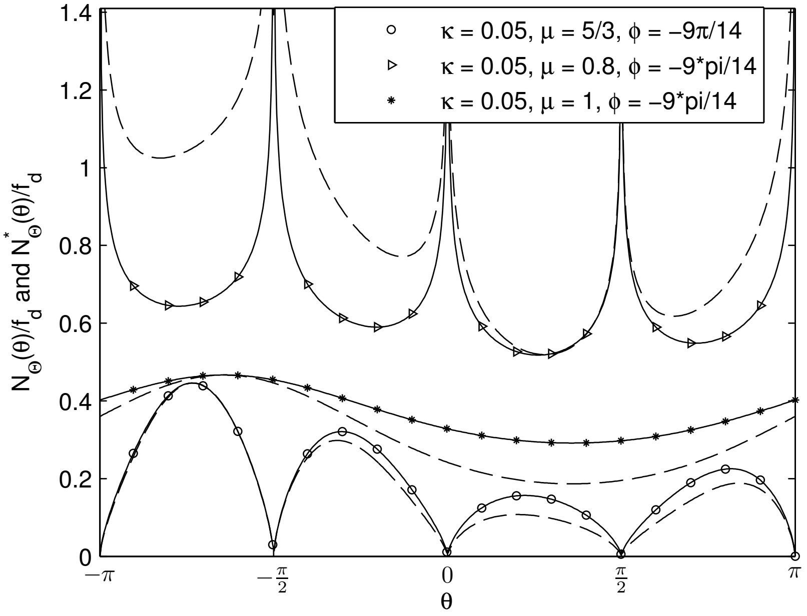


Figure 3.1: Comparison between exact (solid line) and approximate (dashed line) PCR of the κ-μ fading channel.

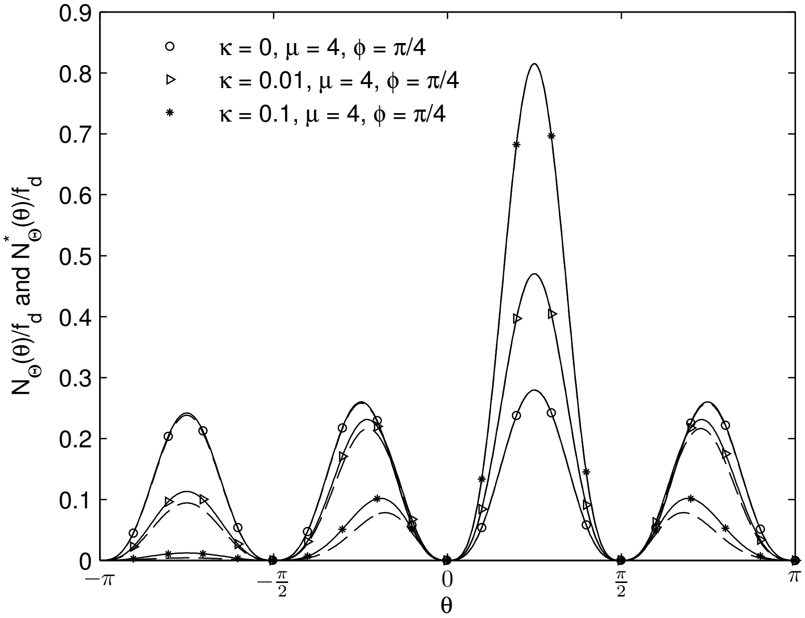


Figure 3.2: Comparison between exact (solid line) and approximate (dashed line) PCR of the κ-μ fading channel.

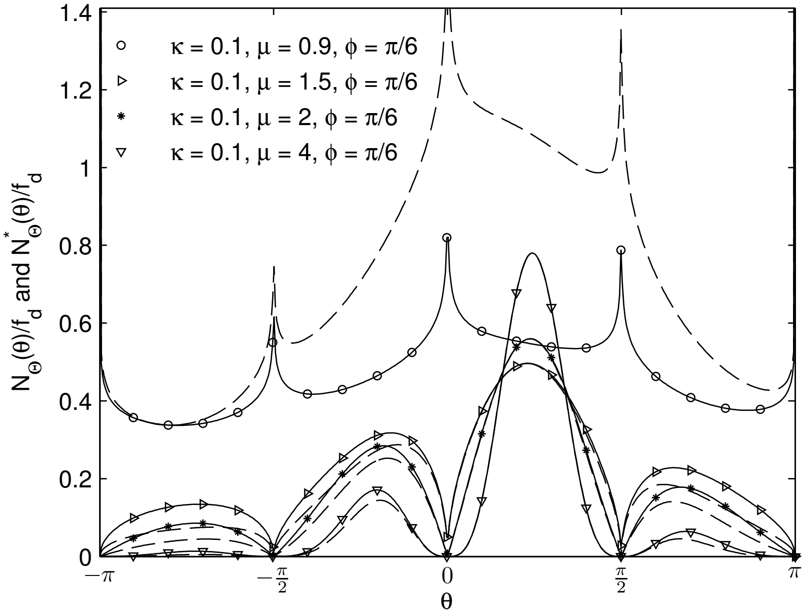


Figure 3.3: Comparison between exact (solid line) and approximate (dashed line) PCR of the κ-μ fading channel.

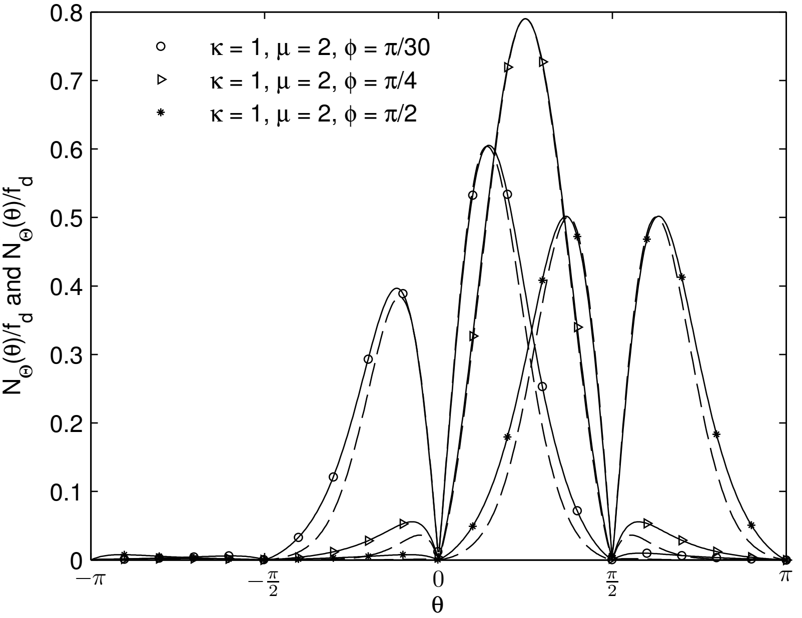


Figure 3.4: Comparison between exact (solid line) and approximate (dashed line) PCR of the κ-μ fading channel.

Approximation by Piecewise Normalization

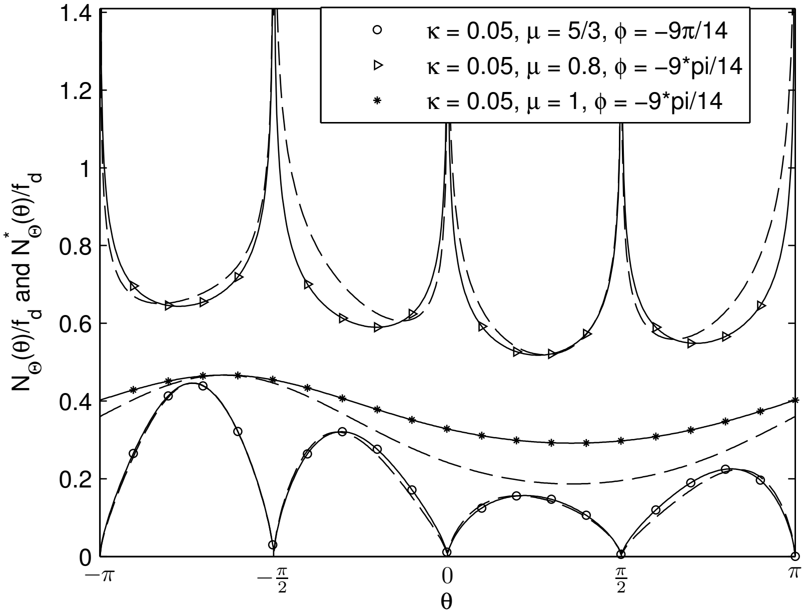


Figure 3.5: Comparison between exact (solid line) and approximate (dashed line) PCR of the κ-μ fading channel.

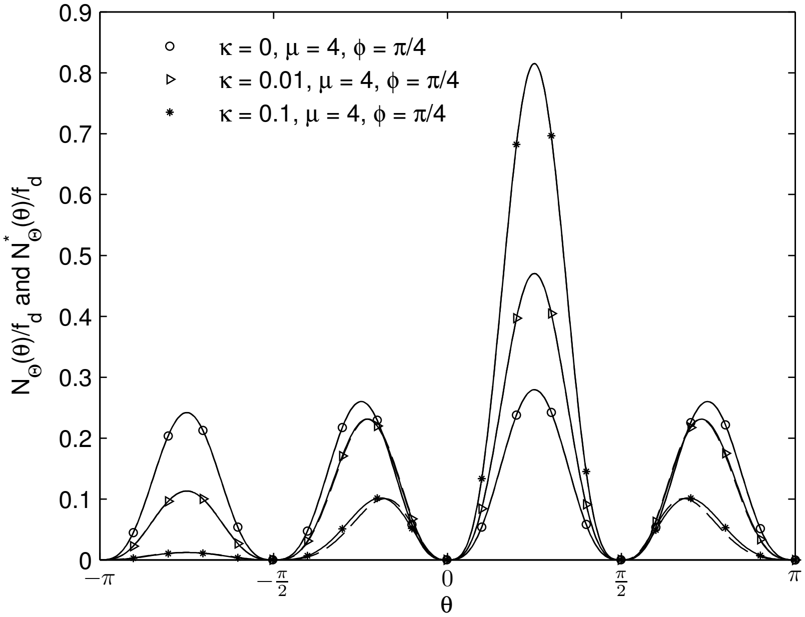


Figure 3.6: Comparison between exact (solid line) and approximate (dashed line) PCR of the κ-μ fading channel.

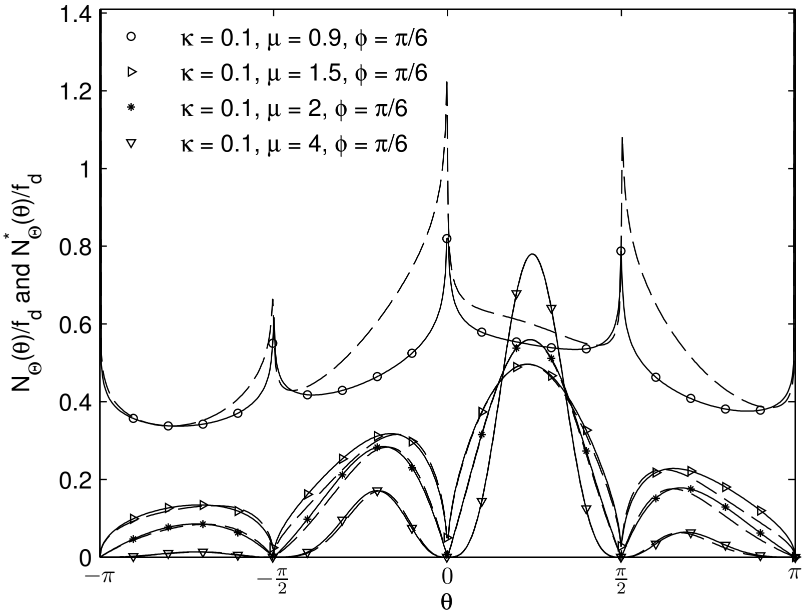


Figure 3.7: Comparison between exact (solid line) and approximate (dashed line) PCR of the κ-μ fading channel.

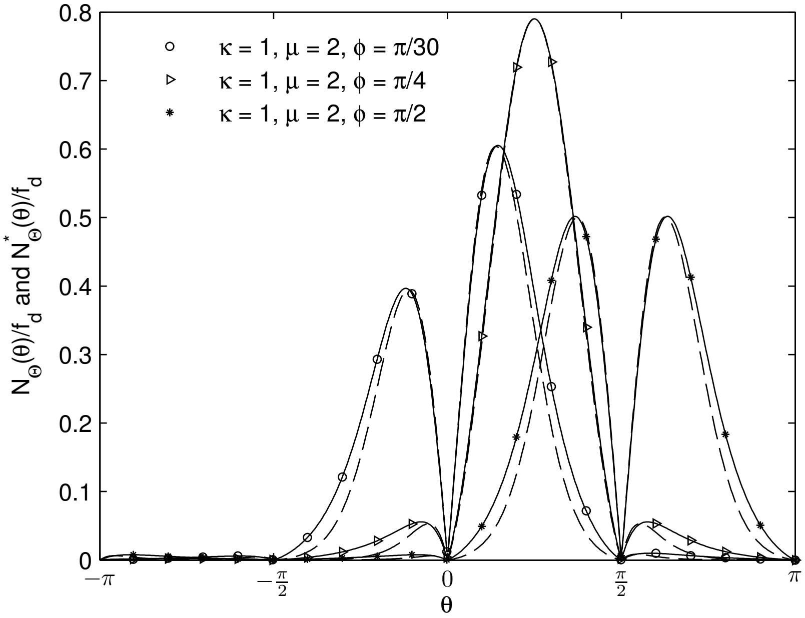


Figure 3.8: Comparison between exact (solid line) and approximate (dashed line) PCR of the κ-μ fading channel.

3.2 Approximate Phase PDF

The numerical integral needed to evaluate the phase PDF of the κ-μ model is very similar to the one needed to evaluate the PCR, as it can be recognized from expressions (1.15) and (2.20).

This suggests that an approximation of the phase PDF can be achieved in a similar manner. However, since this case deals with a probability density function, the approximation will need to satisfy some constraints, as it will be discussed.

First the same method used to achieve (3.4) is followed, by substituting $f_\Theta(\theta)$ for $N_\Theta(\theta)$ and $f_{P,\Theta}(\rho,\theta)$ for $N_\Theta(\rho,\theta)$, which results in

$$\begin{aligned} g(\theta) &= \frac{3}{2}\mu^2\kappa^{1-\frac{\mu}{2}}(1+\kappa)^{1+\frac{\mu}{2}}|\sin 2\theta|^{\frac{\mu}{2}}|\sin 2\phi|^{1-\frac{\mu}{2}} \\ &\times \exp\left(-\mu(1+2\kappa)+2\mu\sqrt{\kappa(1+\kappa)}\cos(\theta-\phi)\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\cos\theta\cos\phi|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\cos\theta\cos\phi\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\sin\theta\sin\phi|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\sin\theta\sin\phi\right). \end{aligned} \tag{3.6}$$

By definition, a continuous probability density function is a non-negative function that integrates to unitary area over its domain [21]. In order to be a PDF, Equation (3.6) needs to be multiplied by some scaling constant to normalize its area to one. Hence, the terms that are not function of θ can be dropped. The simplified version of $g(\theta)$ is given by (3.7).

$$\begin{aligned} g^*(\theta) &= \kappa^{1-\frac{\mu}{2}}|\sin 2\theta|^{\frac{\mu}{2}}\exp\left(2\mu\sqrt{\kappa(1+\kappa)}\cos(\theta-\phi)\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\cos\theta\cos\phi|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\cos\theta\cos\phi\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\sin\theta\sin\phi|\right)\operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\sin\theta\sin\phi\right). \end{aligned} \tag{3.7}$$

The term $\kappa^{1-\frac{\mu}{2}}$ was maintained because otherwise the indeterminacy obtained by setting $\kappa \to 0$ cannot be resolved. For $\mu \geq 0$, $g^*(\theta)$ is always positive. Let $S(\kappa,\mu,\phi) = \left(\int_{-\pi}^{\pi} g^*(\theta)d\theta\right)^{-1}$. The approximate PDF is found by multiplying (3.7) by $S(\kappa,\mu,\phi)$. Denote by $\mathbb{O}$ the random variable with such PDF. Then,

$$\begin{aligned} f_{\mathbb{O}}(\theta) &= S(\kappa,\mu,\phi)|\sin 2\theta|^{\frac{\mu}{2}}\exp\left(2\mu\sqrt{\kappa(1+\kappa)}\cos(\theta-\phi)\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\cos\theta\cos\phi|\right) \\ &\times I_{\frac{\mu}{2}-1}\left(2\mu\sqrt{\kappa(1+\kappa)}|\sin\theta\sin\phi|\right) \\ &\times \operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\cos\theta\cos\phi\right) \\ &\times \operatorname{sech}\left(2\mu\sqrt{\kappa(1+\kappa)}\sin\theta\sin\phi\right). \end{aligned} \tag{3.8}$$

This approximation is similar to the one found for the PCR. In this case, however, no information of the exact PDF is needed. Equation (3.8) is a stand-alone function and is the description of a new probability density function.

3.2.1 Special Cases

As it was discussed, the κ-μ distribution encompasses several other distributions, including Nakagami-m and Rice. In this section, the behavior of the approximate expression for those two special cases is explored.

Rice

The approximate phase PDF for Rice is obtained by setting $\mu = 1$ in (3.8). Surprisingly, this leads to a rather simple and closed-form expression, given by (3.9).

$$f^*_\Theta(\theta)_{Rice} = \frac{\exp\left(2\sqrt{\kappa(1+\kappa)}\cos(\theta-\phi)\right)}{2\pi I_0(2\sqrt{\kappa(1+\kappa)})}. \tag{3.9}$$

The expression (3.9) is the PDF of the Von Mises distribution, which can be viewed as a circular analog of the normal distribution [22]. This result shows that the Von Mises distribution can be used to approximate the Ricean phase distribution, by substituting the κ parameter of the Von Mises distribution by $2\sqrt{\kappa(1+\kappa)}$.

Nakagami-m

Nakagami-m is a particular case of the κ-μ distribution with $\kappa \to 0$. Its phase distribution was derived in [23] and is given by

$$f_\Theta(\theta)_{nak-m} = \frac{|\sin 2\theta|^{m-1}\Gamma(\mu)}{2^\mu\Gamma^2(\mu/2)}. \tag{3.10}$$

Obtaining the approximate expression for the Nakagami-m case is more complicated, since setting $\kappa = 0$ leads to an indeterminacy. To solve the indeterminacy, the limit (2.25) is used in (3.6) and then the resulting expression is scaled to unitary area. The approximation of the phase PDF of the Nakagami-m model obtained this way reduces to the exact Nakagami-m phase PDF. This is an interesting result that shows one particular case in which the proposed approximate phase PDF reduces to the exact phase PDF.

3.2.2 Sample Plots of the Approximate Phase PDF

In this section, exact and approximate solutions of the phase PDF are compared, over a selected range of parameters. Figures 3.9-3.12 show some plots of the general case, whereas Figure 3.13 show a comparison of the Ricean and Von Mises distributions. In this last case, the parameter ϕ only shifts the distributions horizontally. In all cases the approximate solution is very close to the exact solution

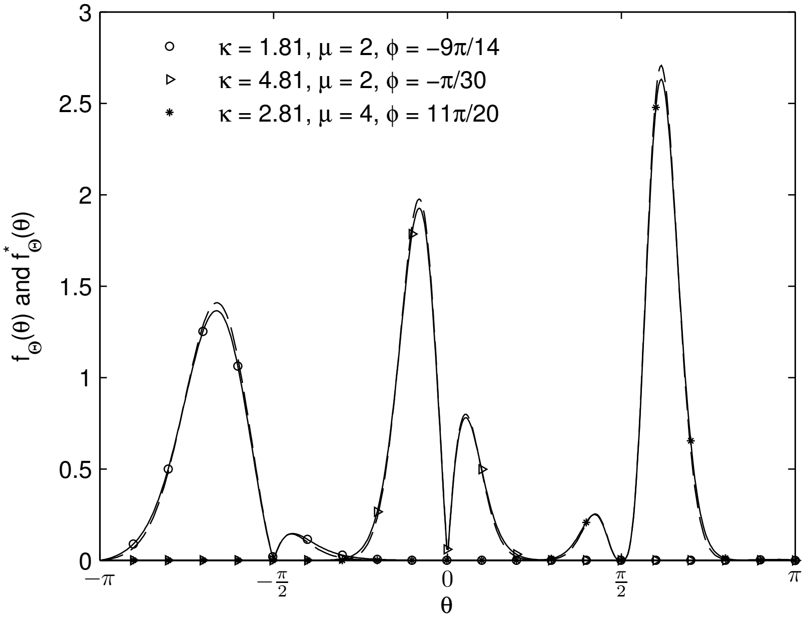


Figure 3.9: Comparison between exact (solid line) and approximate (dashed line) phase PDF of the κ-μ fading channel.

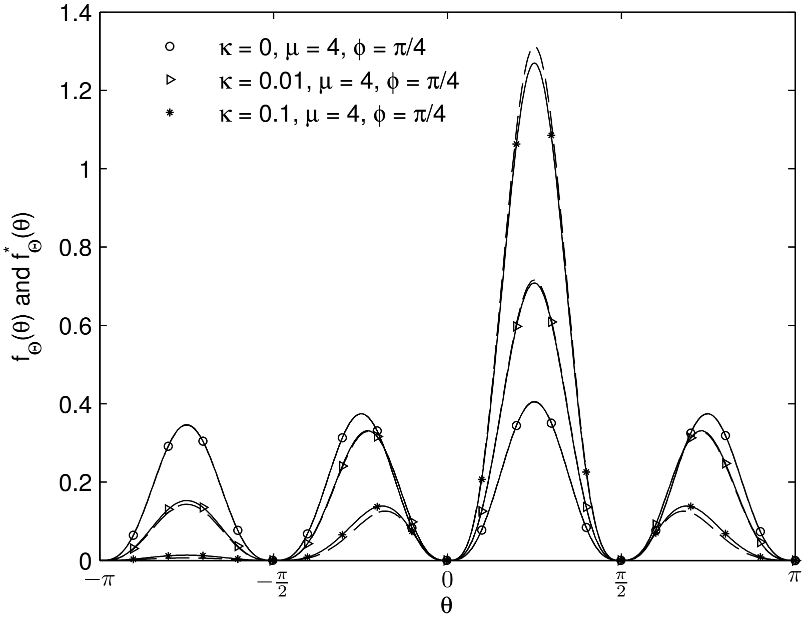


Figure 3.10: Comparison between exact (solid line) and approximate (dashed line) phase PDF of the κ-μ fading channel.

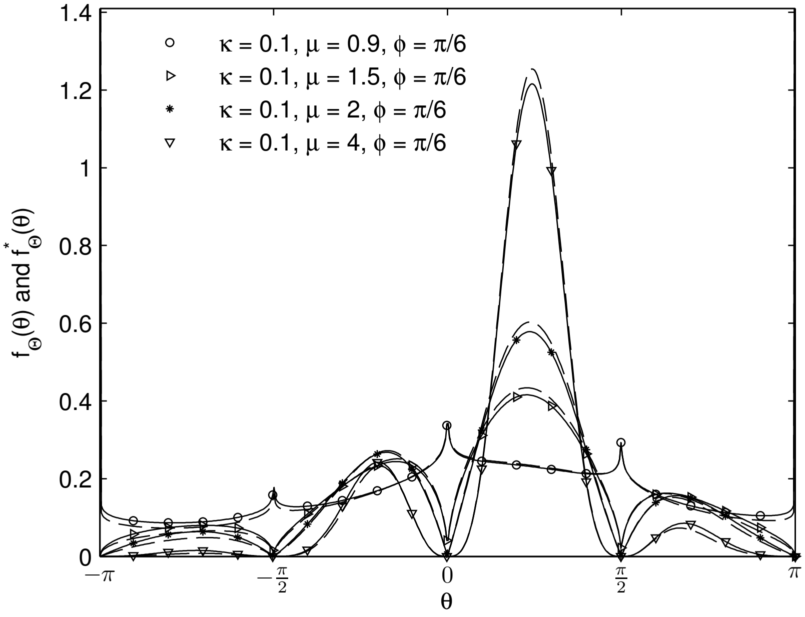


Figure 3.11: Comparison between exact (solid line) and approximate (dashed line) phase PDF of the κ-μ fading channel.

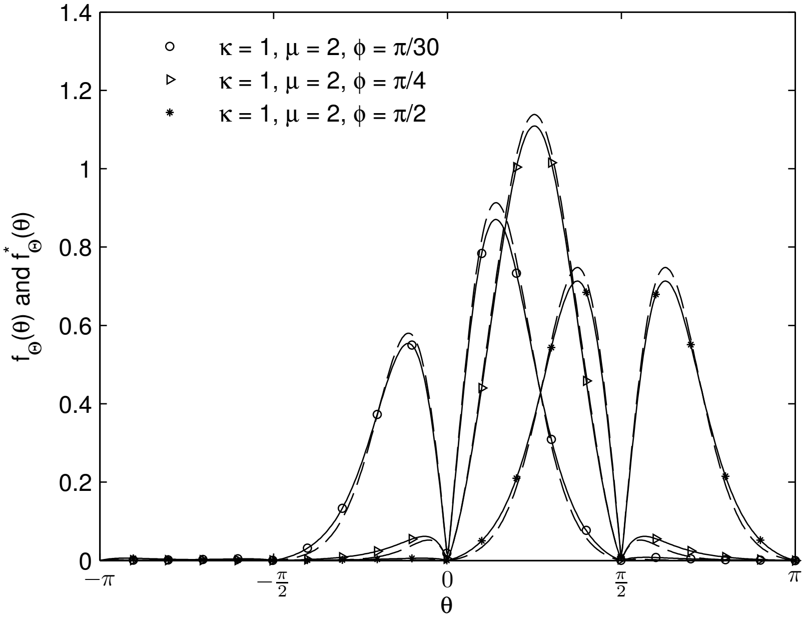


Figure 3.12: Comparison between exact (solid line) and approximate (dashed line) phase PDF of the κ-μ fading channel.

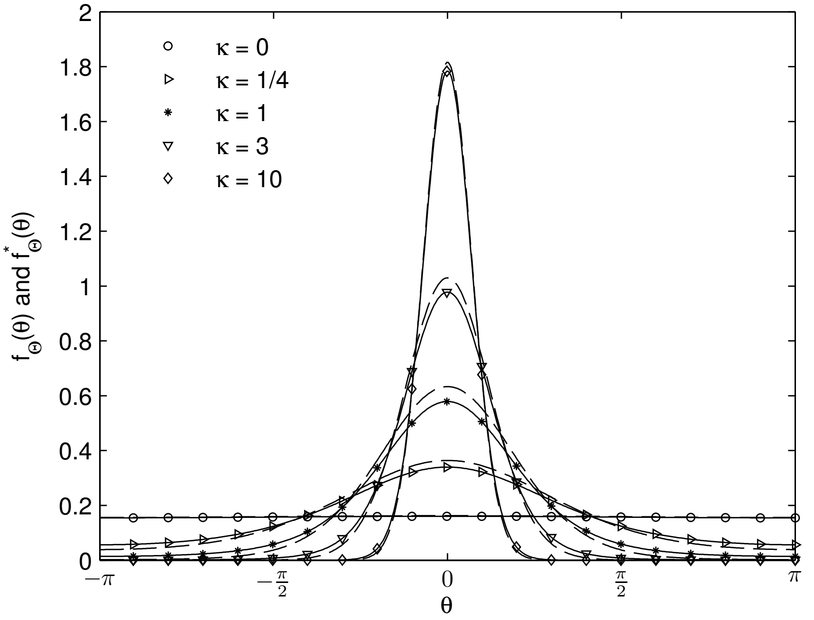


Figure 3.13: Comparison between exact (solid line) and approximate (dashed line) phase PDF of the Ricean fading channel ($\mu = 1$, with $\phi = 0$). In this case, the approximate phase PDF follows a Von Mises distribution.

3.2.3 Comparison between approximate and exact solutions

To evaluate how well the proposed new $\mathbb{O}$ RV approximates the κ-μ phase PDF the solutions can be compared to each other. In particular, two measures of the efficiency of the approximation are explored: the error between exact and approximate solutions and the time needed to compute each formulation. These two aspects are important to quantify the usefulness of the approximation. Whereas the latter shows the main advantage of using an approximation, the former reveals the quality of the fit between the curves.

The error was measured as the energy of the difference between the two probability density functions, as given in (3.11). Figures (3.14 - 3.16) show how the error behaves as each parameter is changed. It reaches its maximum value at $\kappa = 1$ and $\mu = 1$, and it goes down as κ and μ increase. In relation to ϕ, the error is periodic with period $\pi/2$, reaching its lowest value at odd multiples of $\pi/4$.

$$\text{total squared error} = \int_{-\pi}^{\pi} \left[f_{\kappa-\mu}(\theta) - f_{\mathbb{O}}(\theta)\right]^2 d\theta \tag{3.11}$$

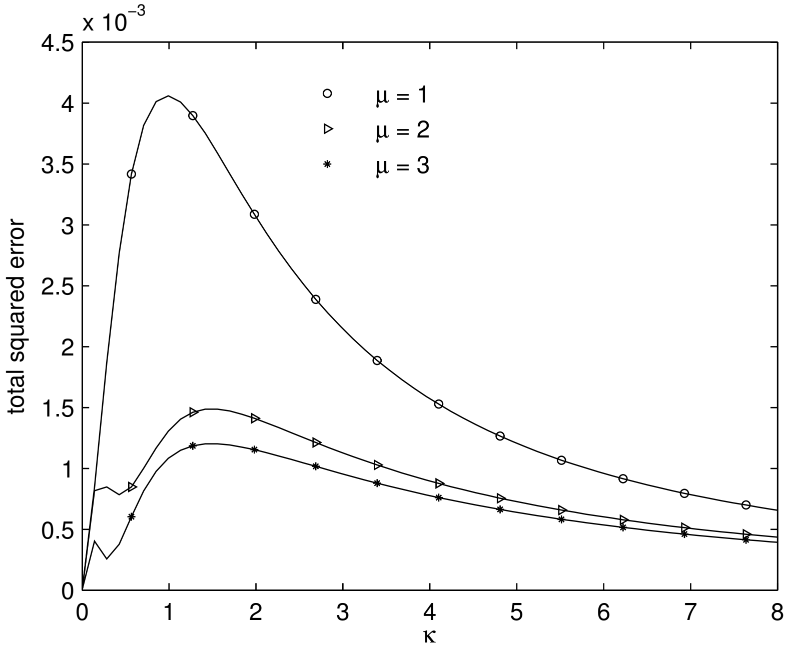


Figure 3.14: Total squared error between exact and approximate solutions of the PDF of the phase of the κ-μ process as a function of κ ($\phi = 0$).

The CPU time was measured with assistance from the Mathematica software [24]. All the times were measured on the same computational system. Figure 3.17 shows the time needed to map the exact and approximate solutions as a function of the number of points used. As it can be seen, the time needed to numerically evaluate the exact expression is around two orders of magnitude greater than the time needed to calculate the approximation, for the same restrictions on the precision of the results.

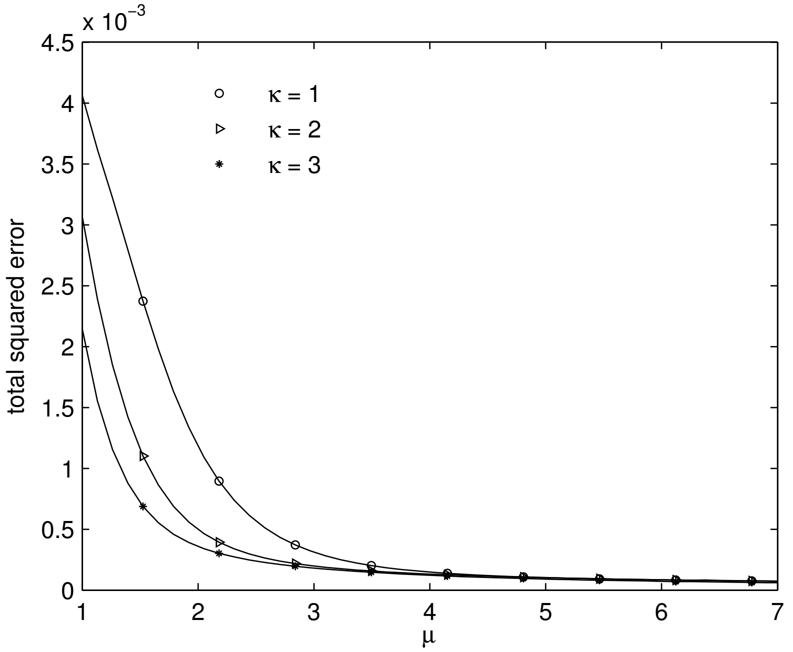


Figure 3.15: Total squared error between exact and approximate solutions of the PDF of the phase of the κ-μ process as a function of μ ($\phi = \pi/5$).

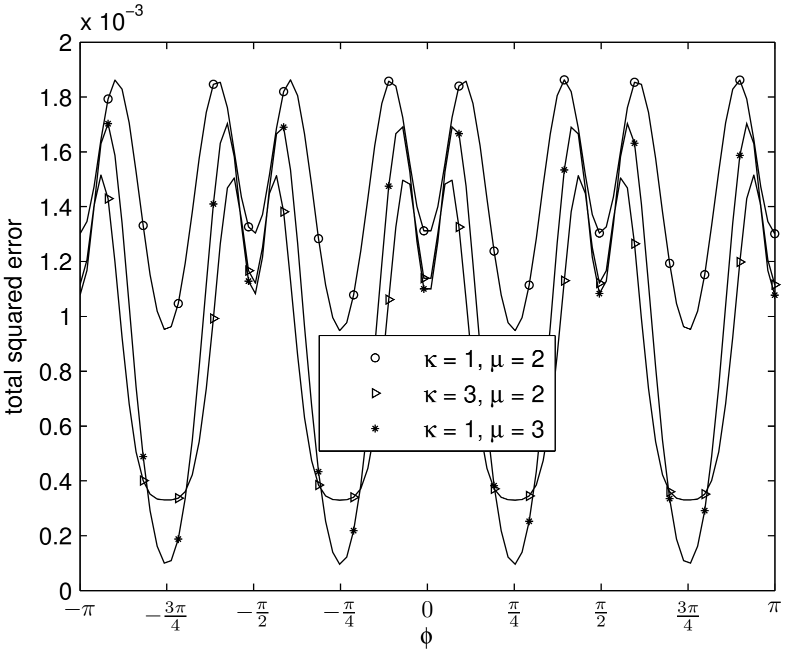


Figure 3.16: Total squared error between exact and approximate solutions of the PDF of the phase of the κ-μ process as a function of ϕ.

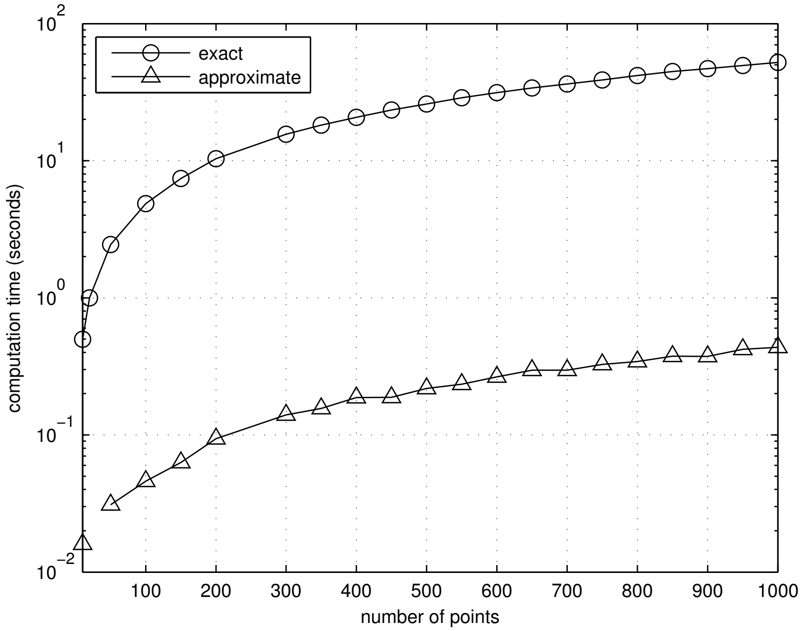


Figure 3.17: CPU time needed to map the approximate and exact solutions. The κ-μ parameters used were $\kappa = 2$, $\mu = 2$ and $\phi = \pi/4$.

3.3 Conclusions

In this chapter, two approximations are described, one for the PCR and one for the phase PDF of the κ-μ model. These approximations will help make the κ-μ model more attractive to the study of the phase of wireless signals, as they require significantly less computational resources than the exact expressions. Whereas the expression of the approximate PCR requires the knowledge of the value of the exact PCR at some points, a stand-alone random variable that approximates the phase PDF is described. This new RV, denoted by $\mathbb{O}$, reduces to the Von Mises distribution when $\mu = 1$ and to the Nakagami-m phase distribution when $\kappa \to 0$. Finally, a comparison is made between the PDF of $\mathbb{O}$ and the exact phase PDF of the κ-μ model, which shows the overall quality of the fit and the superiority of the approximation in terms of computational performance.

Chapter 4

Simulation

The envelope of the κ-μ fading model can be simulated in a straightforward way with the aid of the physical model in (1.1), for integer μ. The variables X_i and Y_i can be modeled in time as the discrete samples $X_i[k]$ and $Y_i[k]$ of a stochastic process, with the auto-correlation in (1.4). Let Z be X or Y and λ be p or q as required. The series $|Z[k]|$ of the κ-μ model can be obtained with the aid of the inverse Fourier transform, according to the block diagram in Figure (4.1) [25].

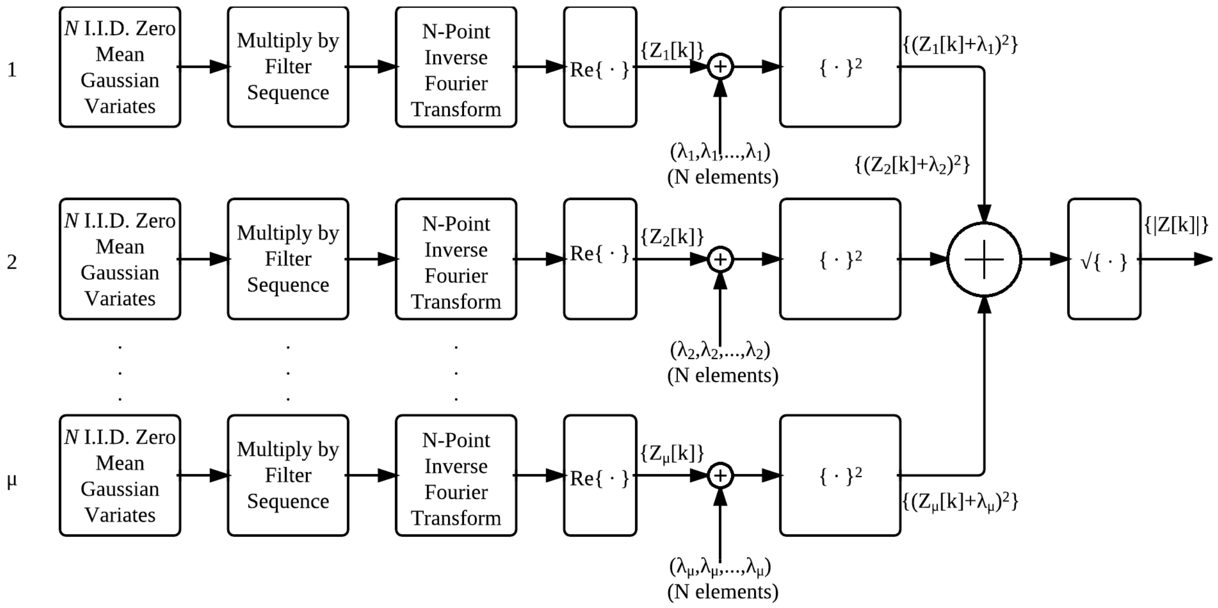

1
N I.I.D. Zero Mean Gaussian Variates
Multiply by Filter Sequence
N-Point Inverse Fourier Transform
Re{ · }
$\{Z_1[k]\}$
$(\lambda_1,\lambda_1,\ldots,\lambda_1)$ (N elements)
{ · }^{2}
$\{(Z_1[k]+\lambda_1)^2\}$
2
N I.I.D. Zero Mean Gaussian Variates
Multiply by Filter Sequence
N-Point Inverse Fourier Transform
Re{ · }
$\{Z_2[k]\}$
$(\lambda_2,\lambda_2,\ldots,\lambda_2)$ (N elements)
{ · }^{2}
$\{(Z_2[k]+\lambda_2)^2\}$
μ
N I.I.D. Zero Mean Gaussian Variates
Multiply by Filter Sequence
N-Point Inverse Fourier Transform
Re{ · }
$\{Z_\mu[k]\}$
$(\lambda_\mu,\lambda_\mu,\ldots,\lambda_\mu)$ (N elements)
{ · }^{2}
$\{(Z_\mu[k]+\lambda_\mu)^2\}$
√{ · }
{|Z[k]|}

Figure 4.1: Block diagram of the simulation of Z.

The discrete filter sequence is [25]:

$$F[k] = \begin{cases} 0, & k = 0, \\ \sqrt{\frac{1}{1\sqrt{1-\left(\frac{k}{Nf_d}\right)^2}}}, & k = 1, 2, \ldots, k_d - 1, \\ \sqrt{\frac{k_d}{2}\left[\frac{\pi}{2} - \arctan\left(\frac{k_d-1}{\sqrt{2k_d-1}}\right)\right]}, & k = 1, 2, \ldots, k_d - 1, \\ 0, & k = k_d + 1, \ldots, N - k_d - 1, \\ \sqrt{\frac{1}{1\sqrt{1-\left(\frac{k}{Nf_d}\right)^2}}}, & k = N - k_d, \\ \sqrt{\frac{1}{1\sqrt{1-\left(\frac{N-k}{Nf_d}\right)^2}}}, & k = N - k_d + 1, \ldots, N - 2, N - 1. \end{cases} \tag{4.1}$$

in which $k_d = \lfloor f_d N \rfloor$.

The κ-μ model imposes only one constraint over the terms p_i and q_i, given in Equation (1.8). There are infinite many partitions of $p_1, p_2, ..., p_\mu$ and $q_1, q_2, ..., q_\mu$ that satisfy the constraints. One of such partitions is to divide them equally among the μ Gaussian processes, so that

$$p_i = p/\sqrt{\mu} \qquad \text{and} \qquad q_i = q/\sqrt{\mu}. \tag{4.2}$$

The terms p_i and q_i can be expressed in terms of the κ-μ parameters for the normalized envelope ($\hat{r} = 1$) using Equation (4.2) in (1.12) and (1.13), so that

$$p_i = \sqrt{\frac{\kappa}{\mu(1+\kappa)}}\cos(\phi) \tag{4.3}$$

and

$$q_i = \sqrt{\frac{\kappa}{\mu(1+\kappa)}}\sin(\phi). \tag{4.4}$$

The time series of the envelope is then obtained by

$$|R[k]| = \sqrt{X[k]^2 + Y[k]^2}. \tag{4.5}$$

The simulation of the phase is more complicated. So far, the sign of the processes $X[k]$ and $Y[k]$ have not been considered, as they are not needed to simulate the envelope. Let $s_X[k]$ and $s_Y[k]$ denote respectively the sign of the in-phase and quadrature components. The complex envelope of the κ-μ model is given as

$$R[k] = s_X[k]|X[k]| + js_Y[k]|Y[k]|. \tag{4.6}$$

The terms $s_X[k]$ and $s_Y[k]$ are not defined by the κ-μ model and their physical description remains an open problem. This makes the simulation of the κ-μ phase specially problematic, which in turn makes it difficult to validate the theoretical expressions for the phase statistics. One solution is to estimate those terms, leading to an approximate simulation. In the next sections, two different approaches to this estimation are discussed.

4.1 Sign Estimation

In the simulation model used in [13], the signs were estimated as

$$s_X[k] = \operatorname{sign}\left(\sum_{i=1}^{\mu} X_i[k] + p_i\right) \qquad \text{and} \qquad s_Y[k] = \operatorname{sign}\left(\sum_{i=1}^{\mu} Y_i[k] + q_i\right). \tag{4.7}$$

This is an interesting approach that reduces exactly to the simulation of the Ricean process when $\mu = 1$. It also provides an excellent fit when the theoretical curve is symmetric, for both the PCR and phase PDF, being almost exact in the Nakagami-m case, when $p_i = q_i = 0$. The fit deteriorates as the curves become asymmetric.

One particularity of this model is that there are discrete jumps in the simulated phase for integer $\mu > 1$. These jumps exist because the estimations s_X and s_Y may change sign, even if the envelope is non zero. The terms s_X and s_Y are the sign of a sum of Gaussian variables, which is itself a Gaussian variable with mean $\sqrt{\mu}p$ or $\sqrt{\mu}q$. They are, in fact, the sign of Ricean processes, which have a non zero probability of crossing any angle as evidenced by the PCR of Rice. On the other hand, the PCR of the κ-μ model is zero at the angles $\theta = n\pi/2$, which means that those angles are crossed, in average, zero times per second. Therefore, to be computed in finite time, any practical simulation of the κ-μ model will have to introduce phase jumps in some way. Figure 4.2 illustrate this behavior.

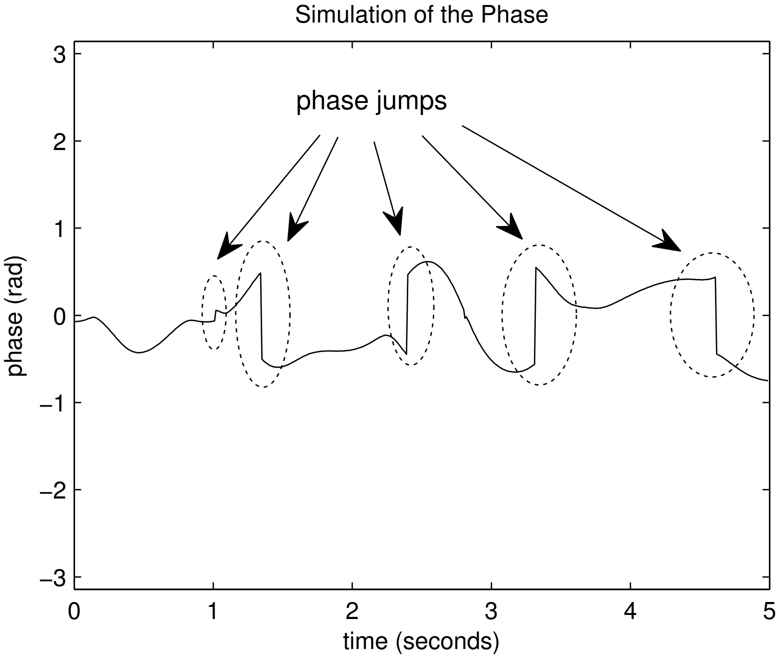


Figure 4.2: Simulation of the κ-μ phase. The phase jumps are highlighted by the dashed ellipses.

4.1.1 Simulation Results - Sign Estimation

Figure 4.3 shows the simulation and exact PCR for the Ricean case, which has an excellent fit, as expected. Figure 4.4 shows the Nakagami-m case, which has also a very good fit. For

$\mu > 1$, the PCR is symmetric in relation to ϕ when $\phi = n\pi/2$. That is, when either p or q is equal to zero. In this case, the process s_X or s_Y will be the sign of a zero mean Gaussian variable, which is also symmetric, providing a very good fit, as evidenced in Figure 4.5. Even in this case, though, the fit deteriorates when κ is near zero, as shown in Figure 4.6. The fit is also not perfect when the PCR is asymmetric, as shown if Figures 4.7-4.8. This is an expected result, given that the unknown exact sign of the κ-μ in-phase and quadrature components has a different nature from the estimation used. The correlated Gaussian time sequences used in the simulation were 10^6 points long, with maximum Doppler shift $f_d = 1Hz$ and sampling frequency $F_s = 100Hz$.

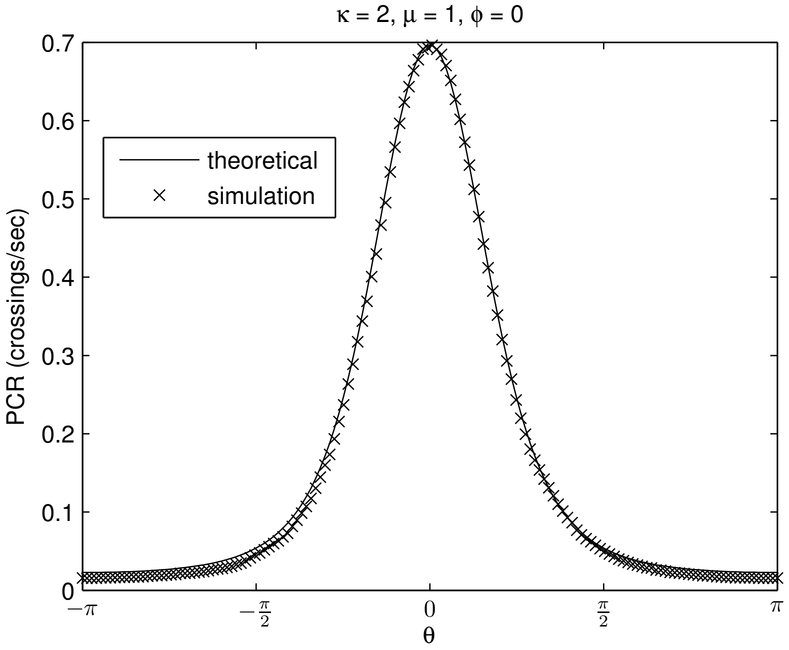


Figure 4.3: Simulation of the PCR of the κ-μ model, Ricean case.

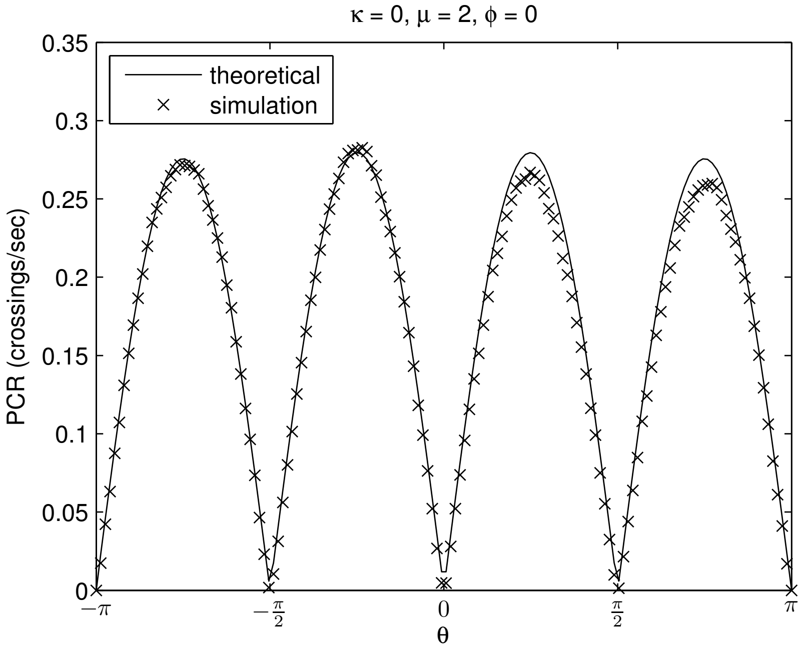


Figure 4.4: Simulation of the PCR of the κ-μ model, Nakagami-m case.

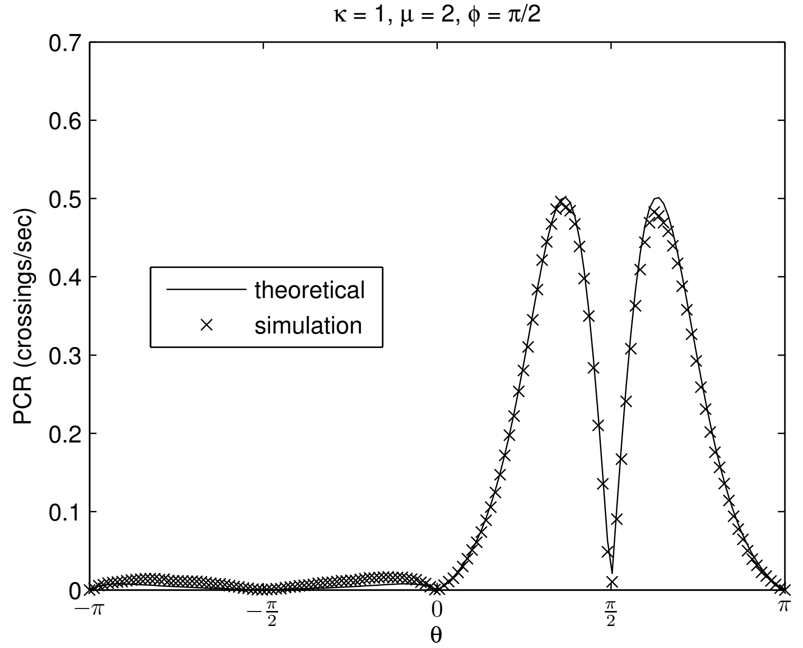


Figure 4.5: Simulation of the PCR of the κ-μ model, symmetric case.

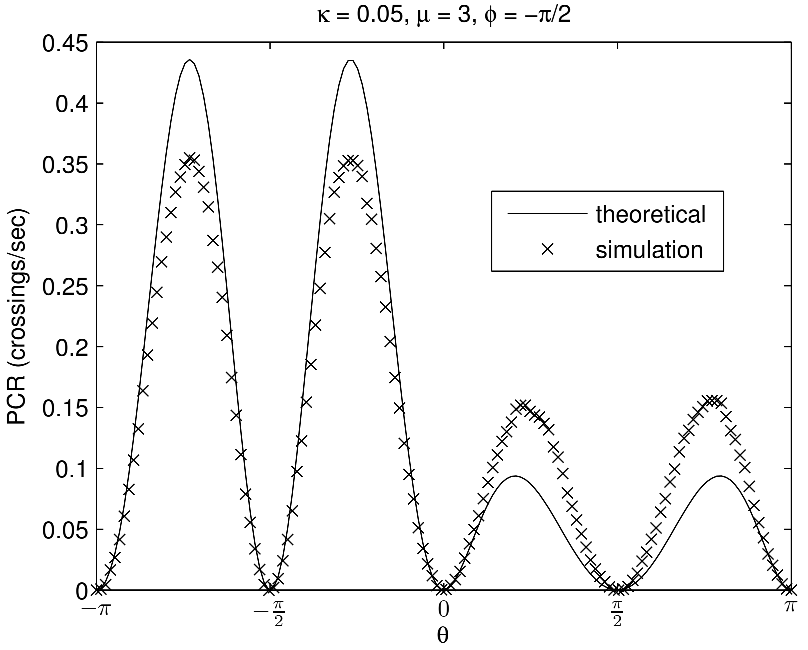


Figure 4.6: Simulation of the PCR of the κ-μ model, symmetric case with low κ.

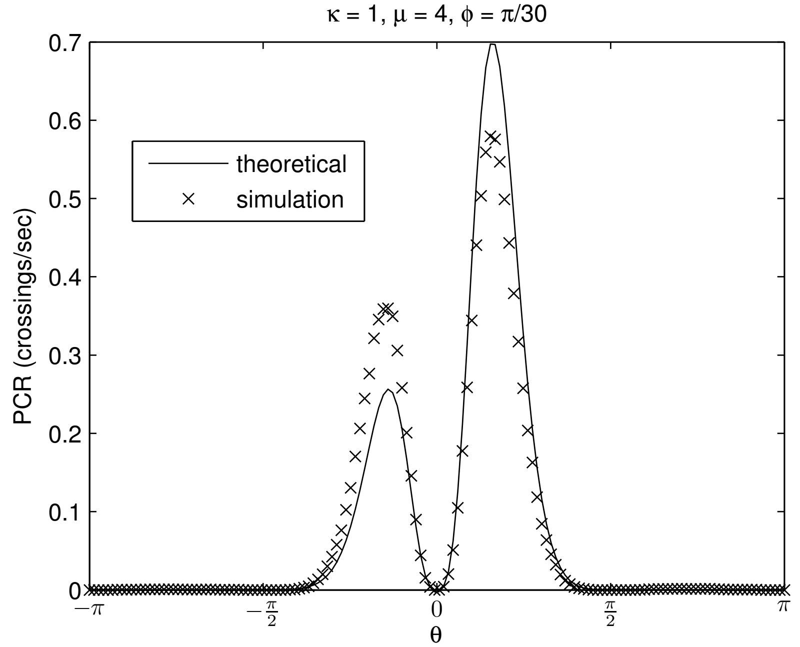


Figure 4.7: Simulation of the PCR of the κ-μ model, asymmetric case.

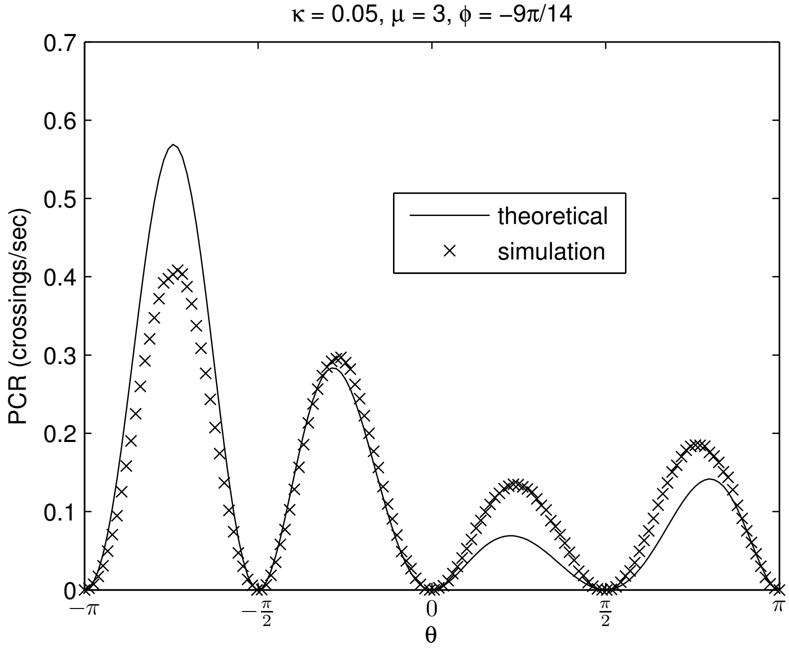


Figure 4.8: Simulation of the PCR of the κ-μ model, asymmetric case with low κ.

4.2 Alternative approach

So far, the new expression proposed for the κ-μ PCR could not be perfectly validated via simulation. Although the good approximate simulation results allied to the fact that the PCR deteriorates to the exact Ricean and Nakagami-m cases are good indication that the expression found is indeed the correct one, an exact simulation would help to consolidate the PCR expression. In this section, an alternative approach is presented, for which an exact simulation is possible.

4.2.1 The Ψ phase variable

The problem in simulating the phase of the κ-μ model is that its complex envelope is an open problem, which, in turn, demands the estimation of the signs s_X and s_Y. This problem can be avoided if the simulation is constrained to the first quadrant, so that the signs never change. This is achieved by defining a new phase variable Ψ, that is obtained from Θ by the following transformation:

$$\Psi = \begin{cases} \pi + \Theta, & \text{if } -\pi < \Theta < -\pi/2, \\ -\Theta, & \text{if } -\pi/2 < \Theta < 0, \\ \Theta, & \text{if } 0 < \Theta < \pi/2, \\ \pi - \Theta, & \text{if } \pi/2 < \Theta < \pi. \end{cases} \tag{4.8}$$

The transformation from Θ to Ψ is non-injective, so there is an inherent loss of information in this transformation. However, since Ψ is constrained to the first quadrant, it can be easily simulated.

4.2.2 Relationship between the statistics of Ψ and Θ

The simulation of Ψ cannot be directly compared to the κ-μ model. In order to obtain a meaningful comparison, the theoretical statistics of Ψ must be obtained in an exact manner. These statistics can be calculated in relation to those of Θ by using the transformation in (4.8) and some results from the Probability Theory.

Represent Ψ as $\Psi = g(\Theta)$. $g(\Theta)$ is a piecewise function with a support that consists of a union of four intervals. Let $g_i(\Theta), i = 1, ..., 4$ be $g(\Theta)$ in each of these intervals. Then [26],

$$f_\Psi(\psi) = \sum_{i=1}^{4} f_\Theta(g_i^{-1}(\psi)) \left| \frac{dg_i^{-1}(\psi)}{d\psi} \right|, \tag{4.9}$$

which leads to

$$f_\Psi(\psi) = f_\Theta(\psi - \pi) + f_\Theta(-\psi) + f_\Theta(\psi) + f_\Theta(\pi - \psi), \qquad 0 < \psi < \pi/2. \tag{4.10}$$

From (4.8), the relationship between $\dot{\Theta}$ and $\dot{\Psi}$ can be derived.

$$\dot{\Psi} = \begin{cases} \dot{\Theta} & \text{if } -\pi < \Theta < 0 \text{ or } 0 < \Theta < \pi/2, \\ -\dot{\Theta} & \text{if } -\pi/2 < \Theta < 0 \text{ or } \pi/2 < \Theta < \pi. \end{cases} \tag{4.11}$$

Since the PDF of $\dot{\Theta}$ is an even function, it follows

$$f_{\dot{\Psi}}(\psi) = f_{\dot{\Theta}}(\theta). \tag{4.12}$$

With Equations (4.8) and (4.12) the joint distribution of Ψ and $\dot{\Psi}$ can be calculated, and is given as

$$f_{\Psi,\dot{\Psi}}(\psi, \dot{\psi}) = f_{\Theta,\dot{\Theta}}(\psi-\pi, \dot{\psi}) + f_{\Theta,\dot{\Theta}}(-\psi, \dot{\psi}) + f_{\Theta,\dot{\Theta}}(\psi, \dot{\psi}) + f_{\Theta,\dot{\Theta}}(\pi-\psi, \dot{\psi}), \quad 0 < \psi < \pi/2. \tag{4.13}$$

The PCR of Ψ is given by

$$N_{\Psi}(\psi) = \int_0^{\infty} \dot{\psi} f_{\Psi,\dot{\Psi}}(\psi, \dot{\psi}) d\dot{\psi}. \tag{4.14}$$

Using (4.13) in (4.14) the relation between N_{Ψ} and N_{Θ} is finally found to be

$$N_{\Psi}(\psi) = N_{\Theta}(\psi - \pi) + N_{\Theta}(-\psi) + N_{\Theta}(\psi) + N_{\Theta}(\pi - \psi), \qquad 0 < \psi < \pi/2. \tag{4.15}$$

4.2.3 Simulation Results - PCR of Ψ

Figures (4.9 - 4.12) show the PCR of the simulation of the Ψ RV of the κ-μ model. The theoretical PCR is the solid line, whereas the markers represent the simulated values. The simulation agrees excellently to the theoretical fit, over all range of parameters. As in the plots of the simulation of Dias's model, the correlated Gaussian time sequences used in the simulation were 10^6 points long, with maximum Doppler shift $f_d = 1Hz$ and sampling frequency $F_s = 100Hz$.

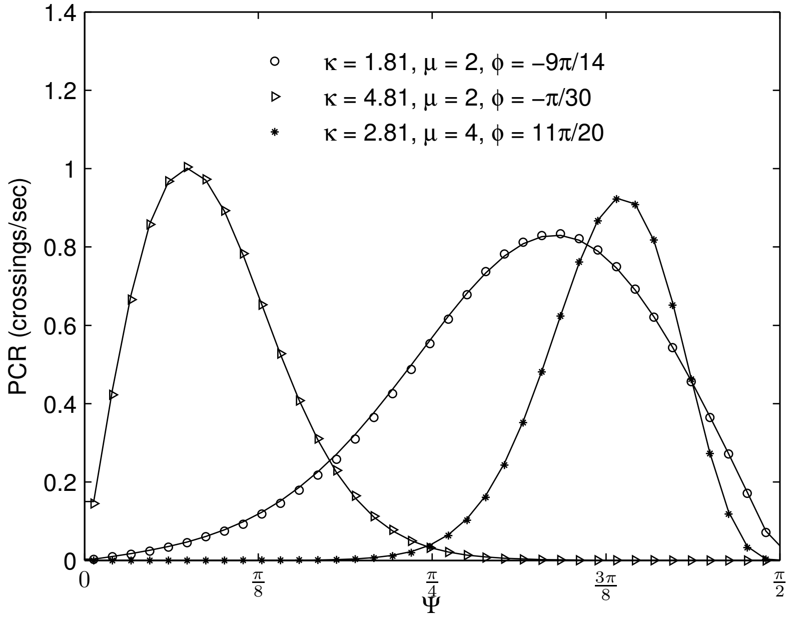


Figure 4.9: Simulation of the PCR of Ψ for the κ-μ model. Solid line is the theoretical value, markers represent the simulated value.

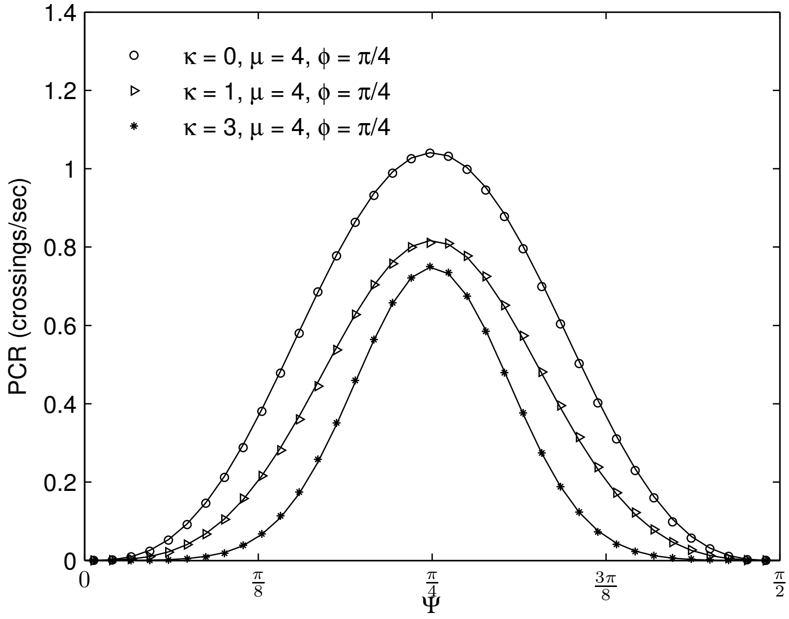


Figure 4.10: Simulation of the PCR of Ψ for the κ-μ model. Solid line is the theoretical value, markers represent the simulated value.

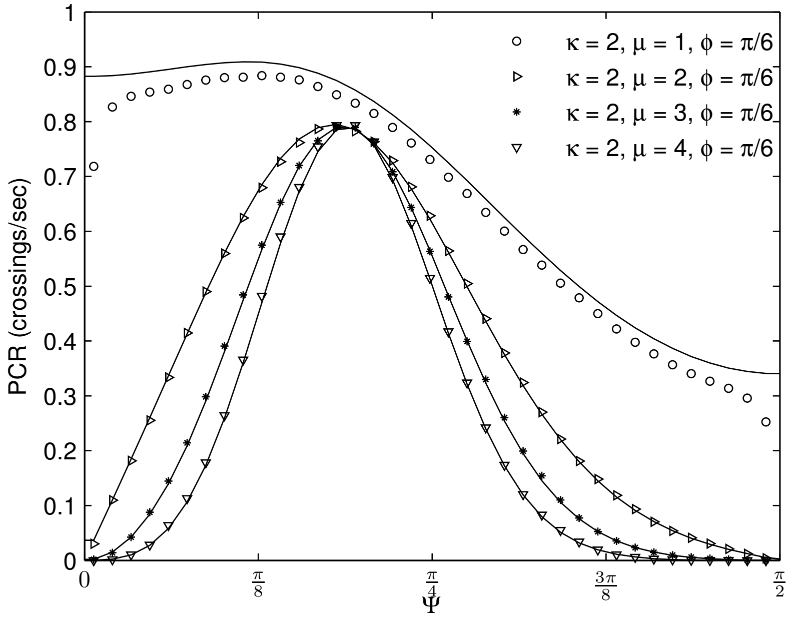


Figure 4.11: Simulation of the PCR of Ψ for the κ-μ model. Solid line is the theoretical value, markers represent the simulated value.

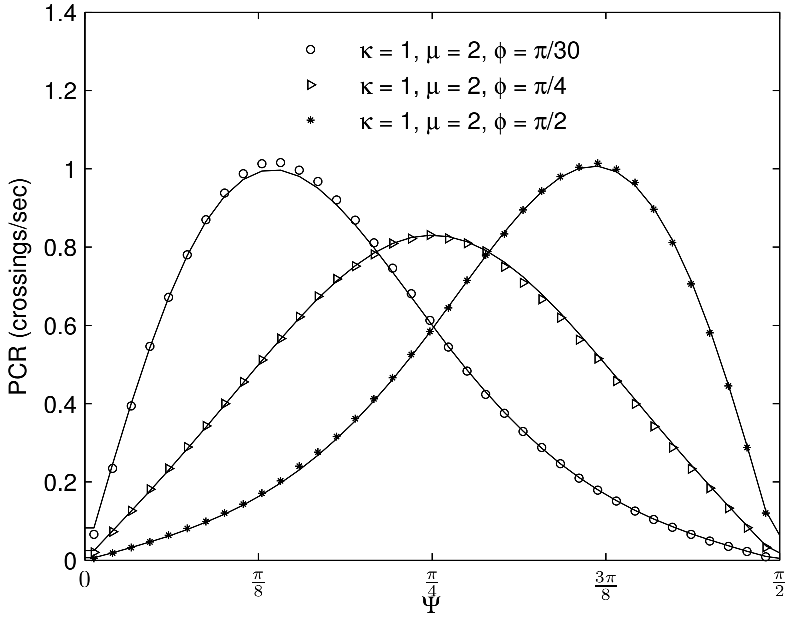


Figure 4.12: Simulation of the PCR of Ψ for the κ-μ model. Solid line is the theoretical value, markers represent the simulated value.

4.3 Conclusions

In this chapter, an efficient simulation method for the generation of correlated Gaussian RV was revisited. It was seen that the absence of a theoretical model of the complex envelope presents a challenge to the phase simulation. Two alternatives were presented to facilitate such a simulation. The first one used an approximation of the complex envelope, resulting in plots that are compatible with the theoretical PCR but are not exact. The second alternative used a variable transformation that made an exact simulation possible, resulting in an excellent agreement between the theory and simulation. The trade-off in this case was a loss of information in the transformation that generated this RV. These two approaches combined help validate the theoretical expression obtained for the κ-μ PCR.

Chapter 5

Curve Fitting

In 2010, Cotton and Scanlon conducted experiments in which they measured the reception of a wireless signal in a wireless body area network context [27]. Two separate environments were studied: an anechoic chamber and a reverberation chamber. The test subject, an adult male of height 1.82m and mass 90kg, carried one transmitter at the left side of the waist and 7 receivers at his right side. The receivers were located at the head, the chest, the elbow, the waist, the wrist, the knee and the ankle. The radiating patch element of the antennas were mounted parallel to the surface of the body and the experiment was conducted at 2.45 GHz. All elements were connected to a vector analyzer Rhode & Schwartz ZVB-8 VNA using calibrated low loss coaxial cables. Figure (5.1) show the setup of the transmitter and receivers. In each scenario, 3000 data points were used, at a sampling frequency of 200 Hz, totalling 15 seconds of data gathering.

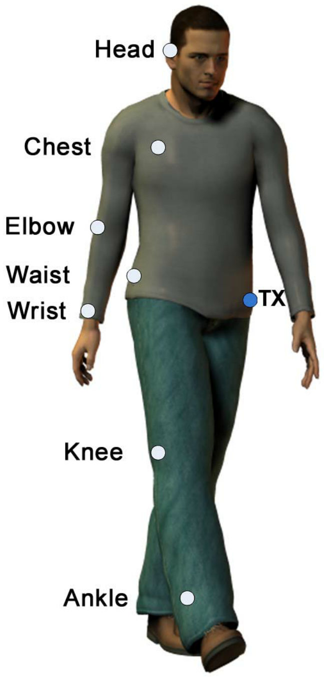


Figure 5.1: Location of the transmitters and receivers.

The data collected by Cotton and Scanlon were used so as to test the validity of the formulations for the κ-μ PCR developed here. In some cases, the data were also fitted to the generalized Nakagami-m distribution. To the best of the author's knowledge this is the first time that field data have been used to support this theoretical model. To find an optimal fit, first a non-linear search of the parametric space was performed. In addition to the parameters κ,μ and ϕ, the maximum Doppler shift f_d appears as a multiplicative term in the PCR expression and is also a degree of freedom to be considered in the search. Using the parameters found by the non-linear optimization as a starting point, a manual search was used to refine the fit. The goal of this optimization was to find theoretical curves that well represented the features of the PCR of the experimental data, while maintaining a tight fit. The measure of tightness that was used was the energy of the error, as shown in Equation (3.11). To aid the manual search, a graphic interface was developed. This interface is shown in Appendix A. Both the manual search and the non-linear optimization were implemented in MATLAB [28]. The maximum Doppler shift f_d was estimated as the multiplicative constant that minimizes the error between the data field and the theoretical curves. To find a better fit, the phase reference of the experimental data was shifted as needed.

5.1 Adjusted κ-μ PCR

This section shows the plots of the PCR of the data fitted to the exact κ-μ model. Figures 5.2-5.8 show the measures from the reverberation chamber and Figures 5.9-5.15 show the measures from the anechoic chamber. In many cases, the optimum μ is close to one, which suggests Ricean fading. Even then, the extra degrees of freedom allowed by the κ-μ model make it possible to find tighter adjustments with this model.

In the reverberation chamber all the estimated κ were low, always below the unity. This is in accordance with the κ-μ model, suggesting that in this scenario the energy of the scattered components is greater than the energy of the LoS component, as expected. In contrast, in the anechoic chamber, some sensors had a clear LoS to the transmitter, as evidenced in Figures 5.10, 5.12 and 5.14. In these cases, the high κ denotes that most energy came from the direct component.

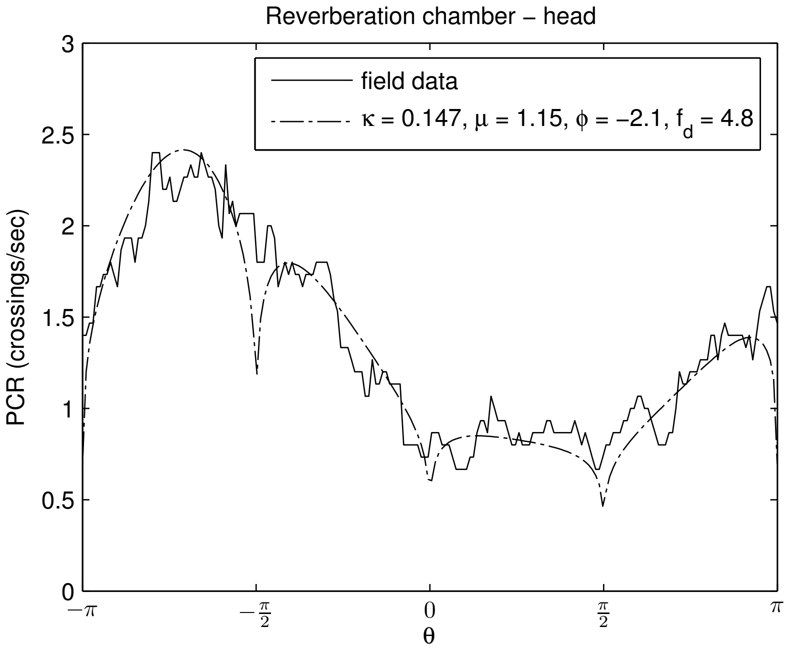


Figure 5.2: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

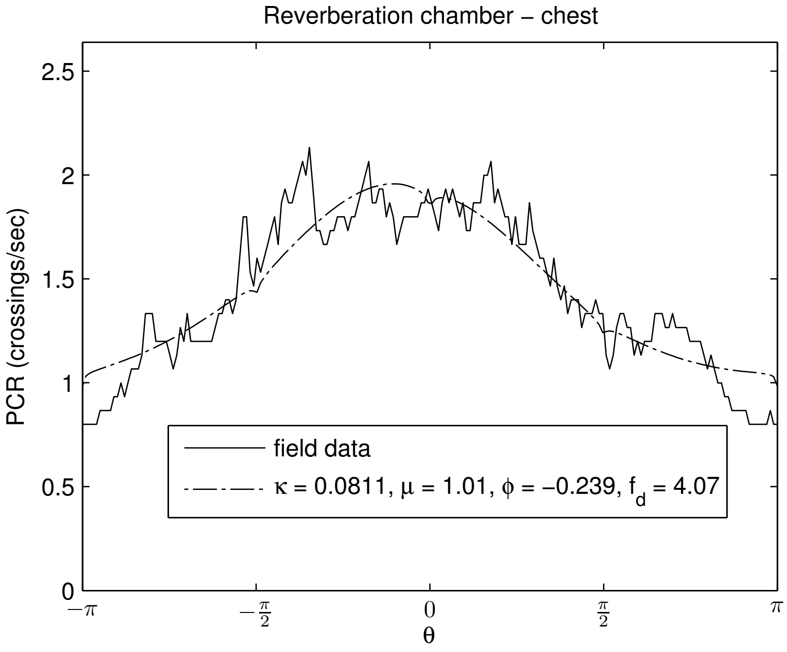


Figure 5.3: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

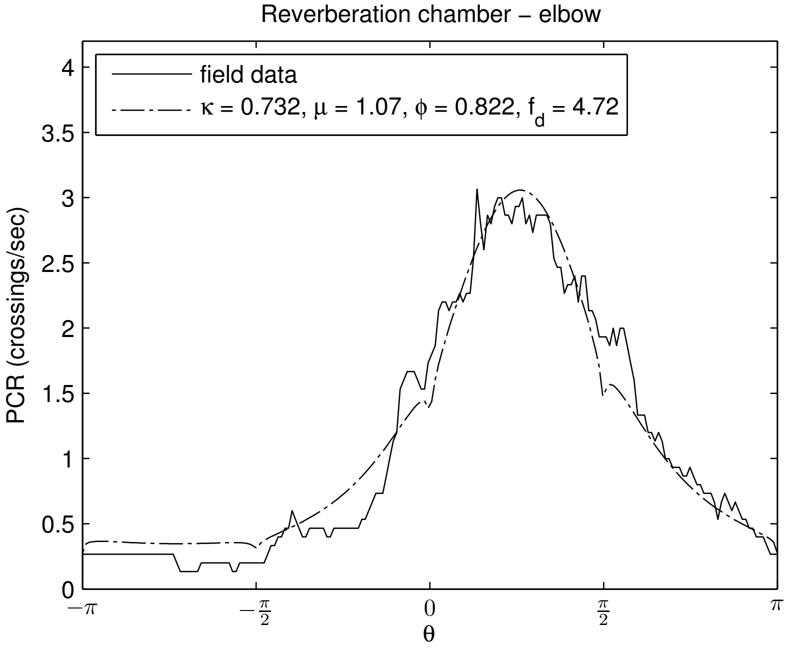


Figure 5.4: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

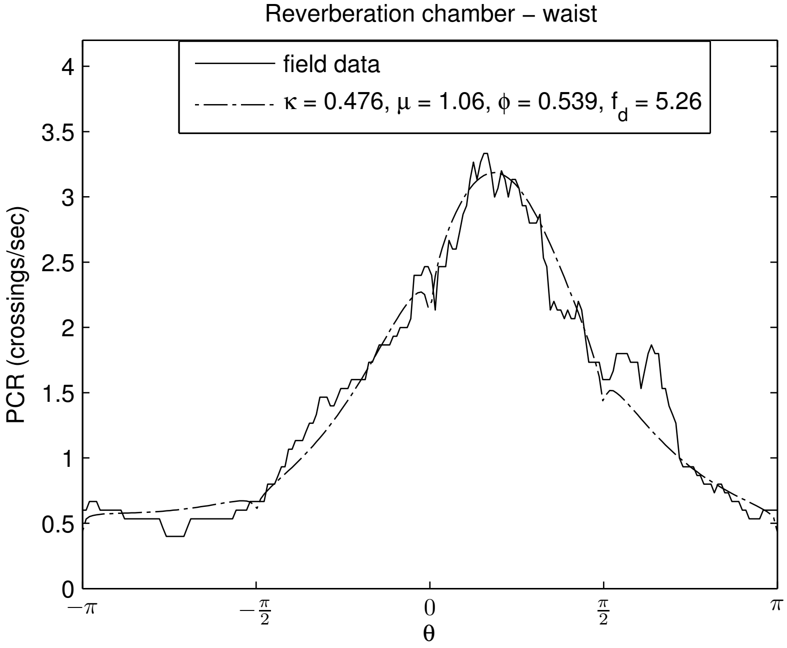


Figure 5.5: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

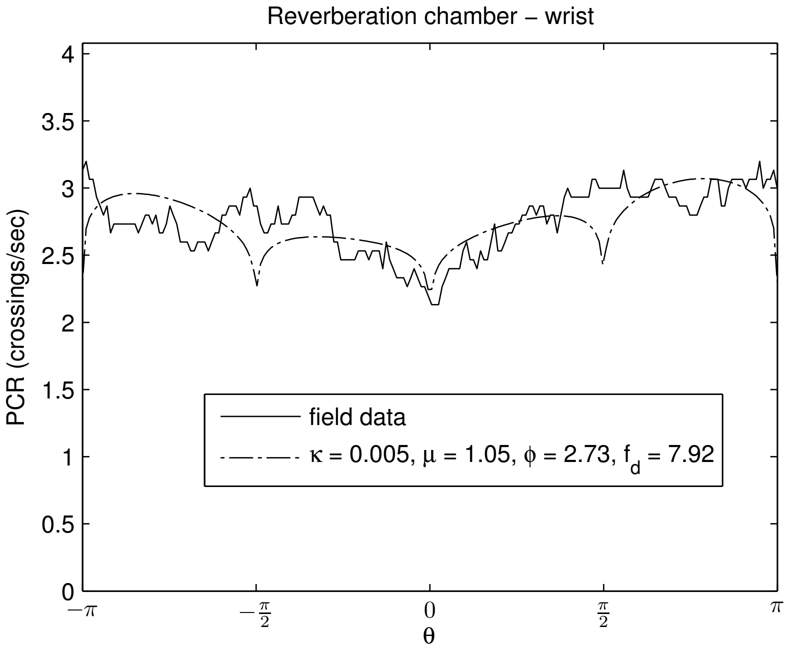


Figure 5.6: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

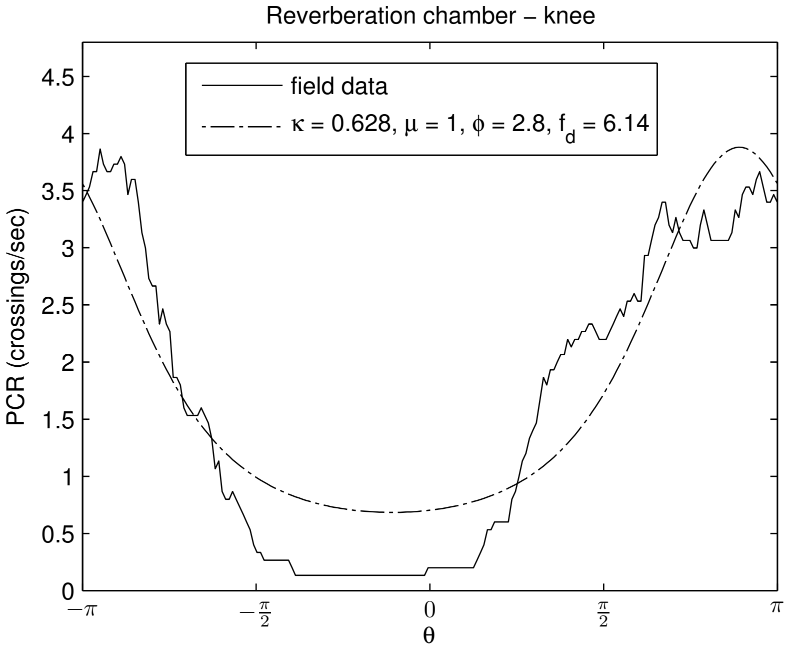


Figure 5.7: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

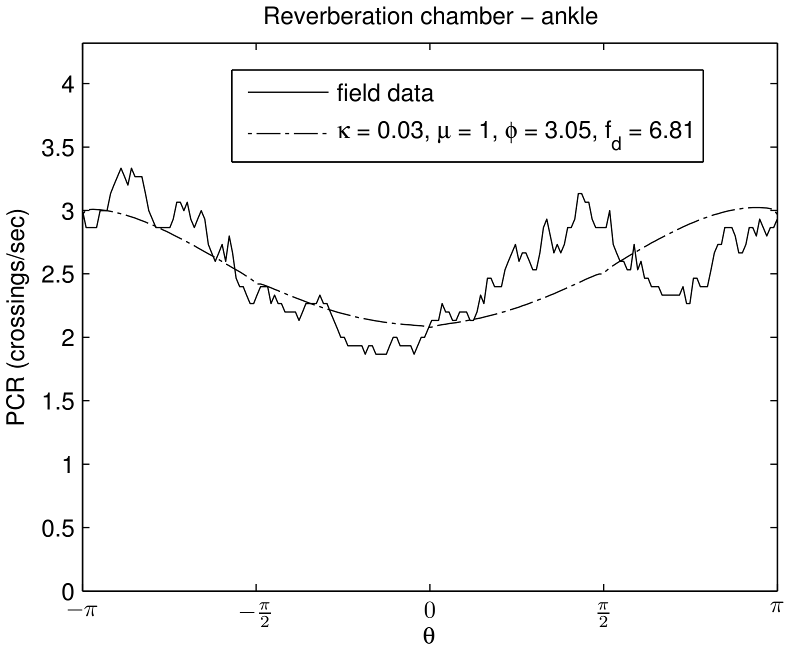


Figure 5.8: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

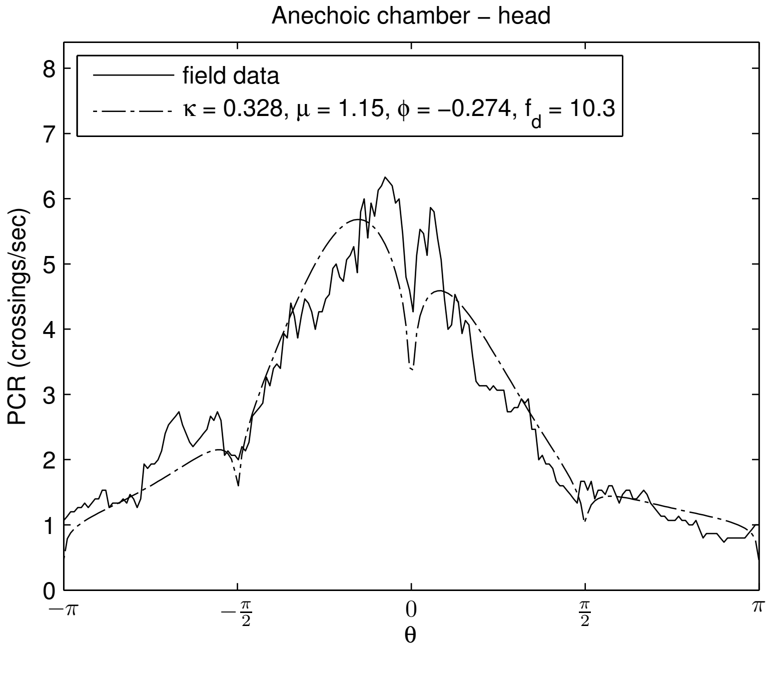


Figure 5.9: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

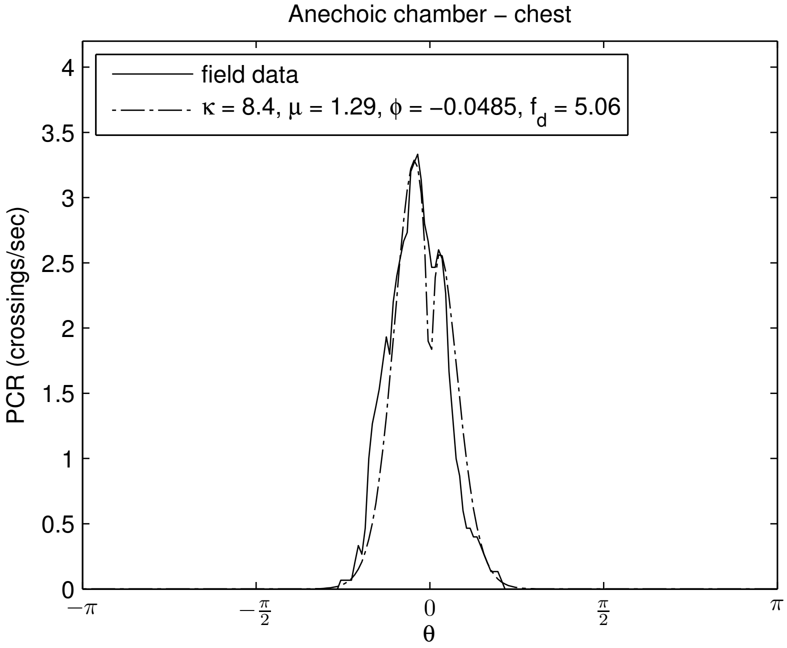


Figure 5.10: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

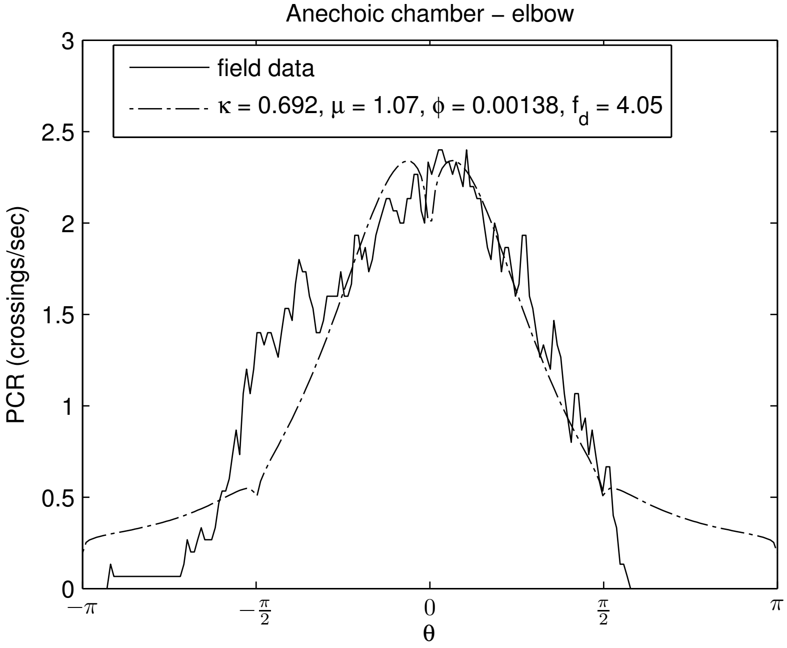


Figure 5.11: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

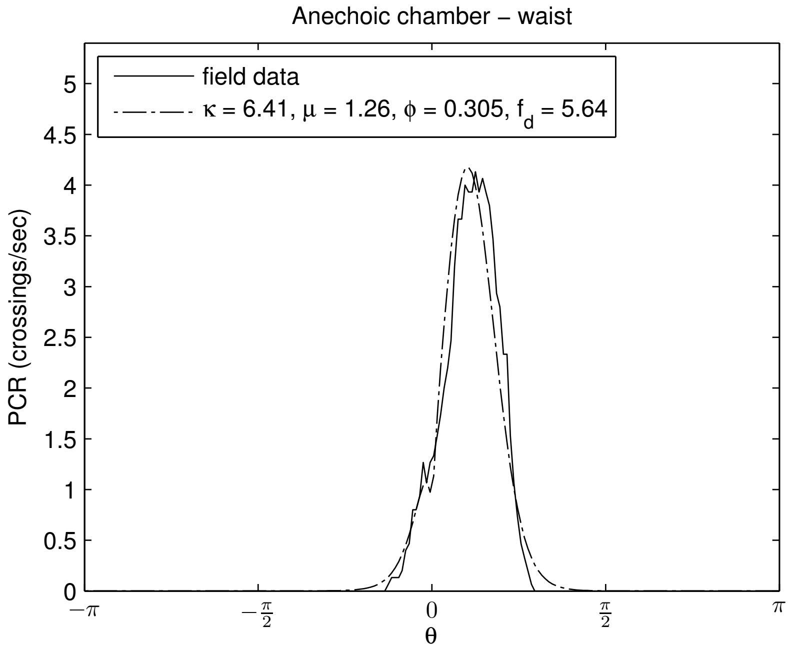


Figure 5.12: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

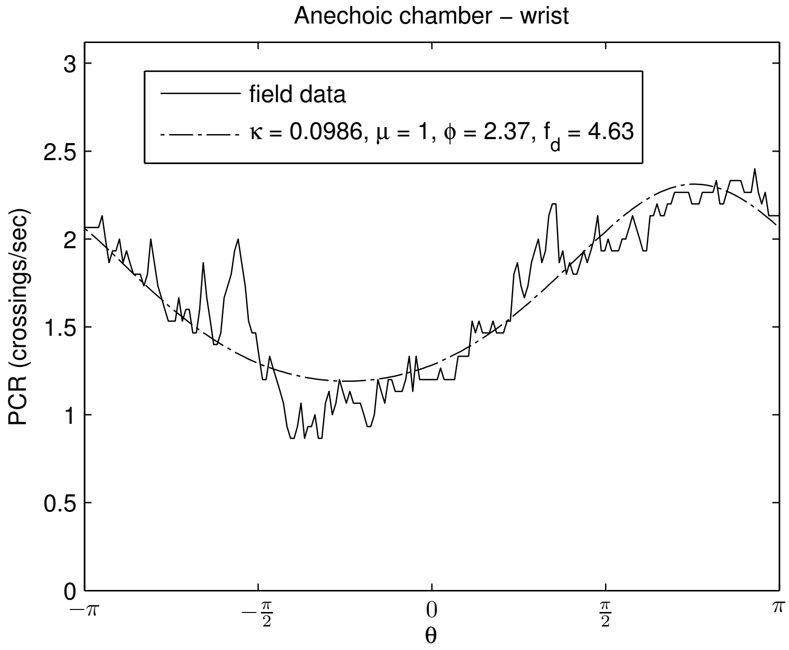


Figure 5.13: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

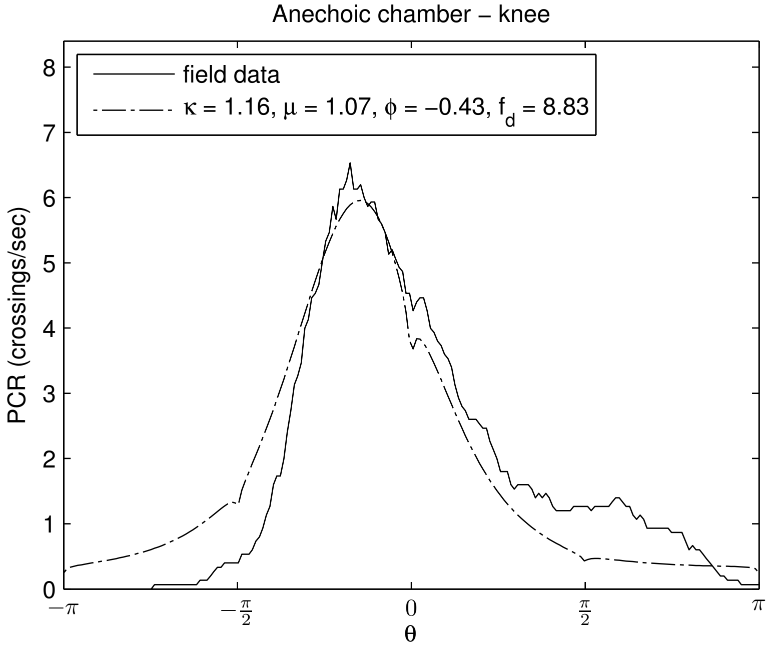


Figure 5.14: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

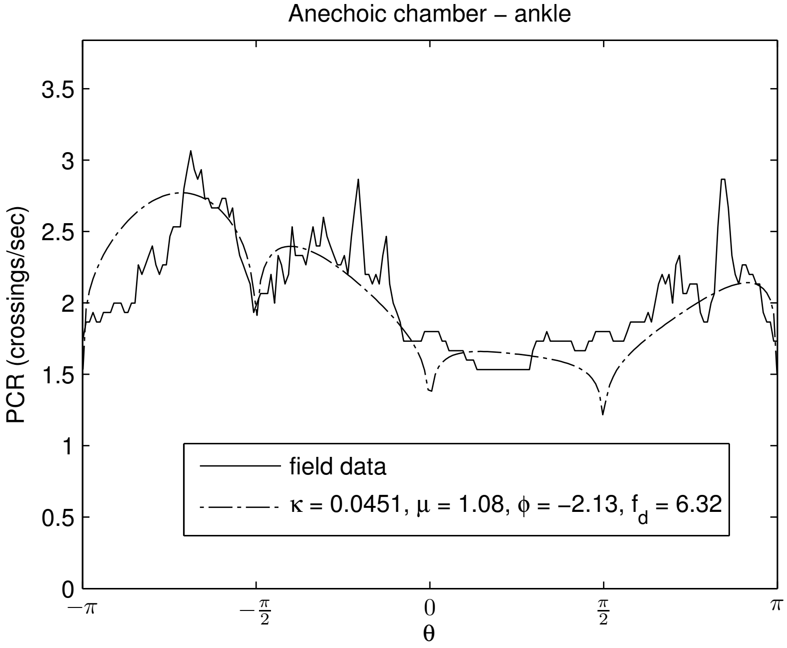


Figure 5.15: PCR of the κ-μ model fitted to field data obtained in WBAN experiments.

5.2 Adjusted Generalized Nakagami-m PCR

Three cases were found to adjust well to the Nakagami-m model. Two of these cases were measures from the reverberation chamber, as shown in Figures 5.16 and 5.17, and one from the anechoic chamber, as shown in Figure 5.18. Unsurprisingly, all of theses cases had a very low κ estimated in the κ-μ fitting. The best fit was obtained for the ankle receiver in the reverberation chamber, as shown in Figure 5.17. The measured PCR shows two main lobes, which are well modeled with the generalized Nakagami-m PCR but not with the classical Nakagami-m PCR. In the classical case, there are always four main lobes.

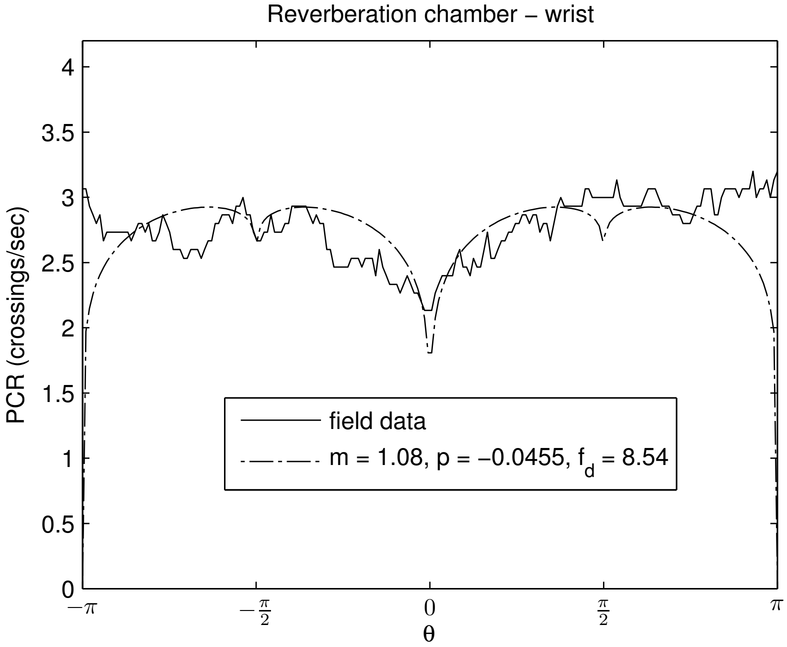


Figure 5.16: PCR of the Generalized Nakagami-m model fitted to field data obtained in WBAN experiments.

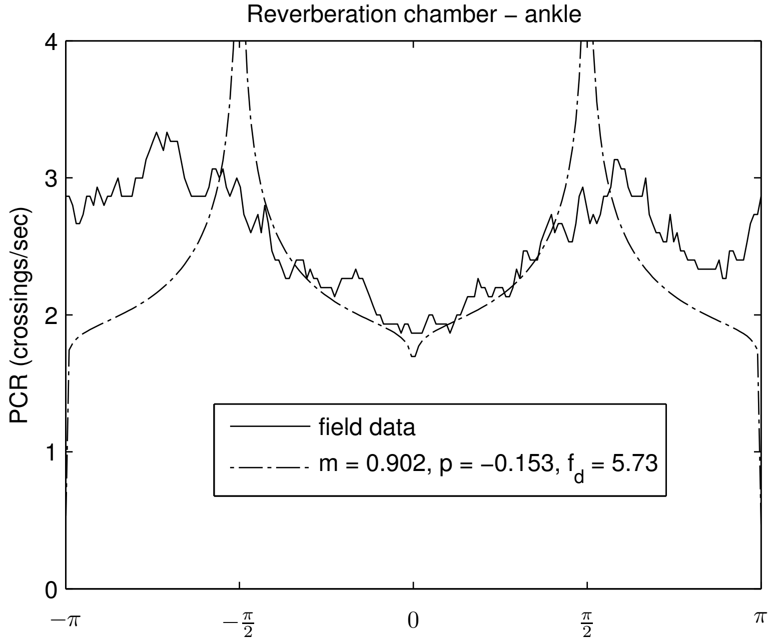


Figure 5.17: PCR of the Generalized Nakagami-m model fitted to field data obtained in WBAN experiments.

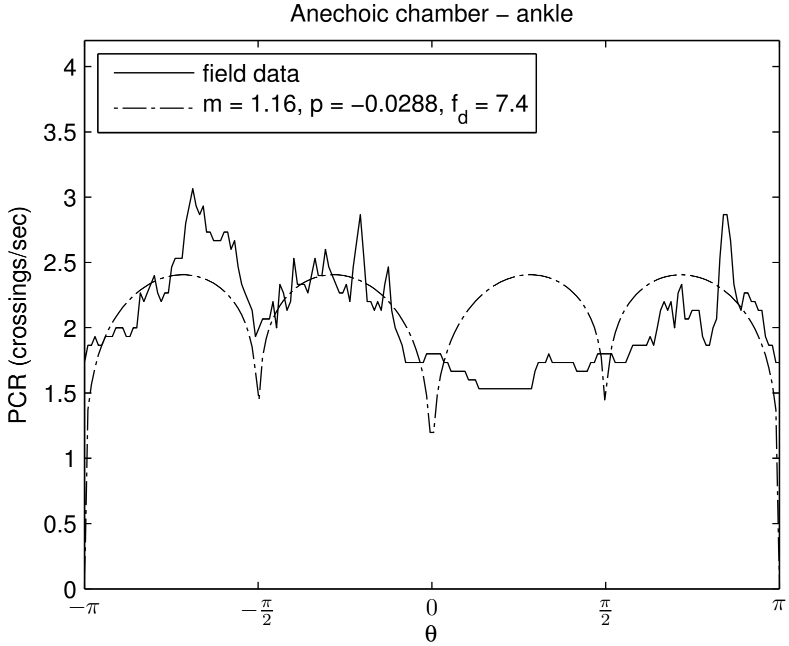


Figure 5.18: PCR of the Generalized Nakagami-m model fitted to field data obtained in WBAN experiments.

5.3 Conclusions

This chapter concerns the adjustment of the κ-μ and Nakagami-m PCRs to field data. The parameters were estimated using a non-linear optimization algorithm, that searched for the parameters that gave an optimum fit. Overall, very good agreement with the theoretical κ-μ PCR was obtained. The generalized Nakagami-m model also provided a good fit in some scenarios. These results indicate that there are practical propagation conditions that can be physically described by those fading models.

Chapter 6

Conclusions and Further Work

This work presented important results regarding the phase statistics of the κ-μ fading channel. It provided the derivation of a higher-order phase statistic, the PCR, in an exact manner. This theoretical expression was later validated through simulations. Tight approximations of the exact expressions of both PCR and phase PDF of the κ-μ model were proposed. Finally, field data were adjusted with the κ-μ and generalized Nakagami-m PCRs.

The main contribution of this work was to deepen the knowledge of the κ-μ phase related processes. This goal was achieved by working in three different fronts: firstly, a previously unknown phase statistics was derived in an exact manner; secondly, the approximations derived for the PCR and the phase PDF demand significantly less computational resources to be evaluated than their exact counterparts, which facilitates the use of the model; thirdly and finally, it was shown that there exist indeed real world fading conditions in which the phase of the received signal can be well represented by the κ-μ model. In that sense, this work also contributed to the generalized Nakagami-m model.

Further work remains to be done concerning the simulation of the κ-μ phase process. A theoretical model of the complex envelope would greatly simplify such simulations. Other approaches can also be explored, such as the rank-matching approach that was proposed to simulate Nakagami-m channels [29]. Other possible area of interest is to further investigate the proposed $\mathbb{O}$ random variable. Specifically, the calculation of some kind of estimator for that RV would help in the process of fitting data.

Appendices

Appendix A

MATLAB User Interface

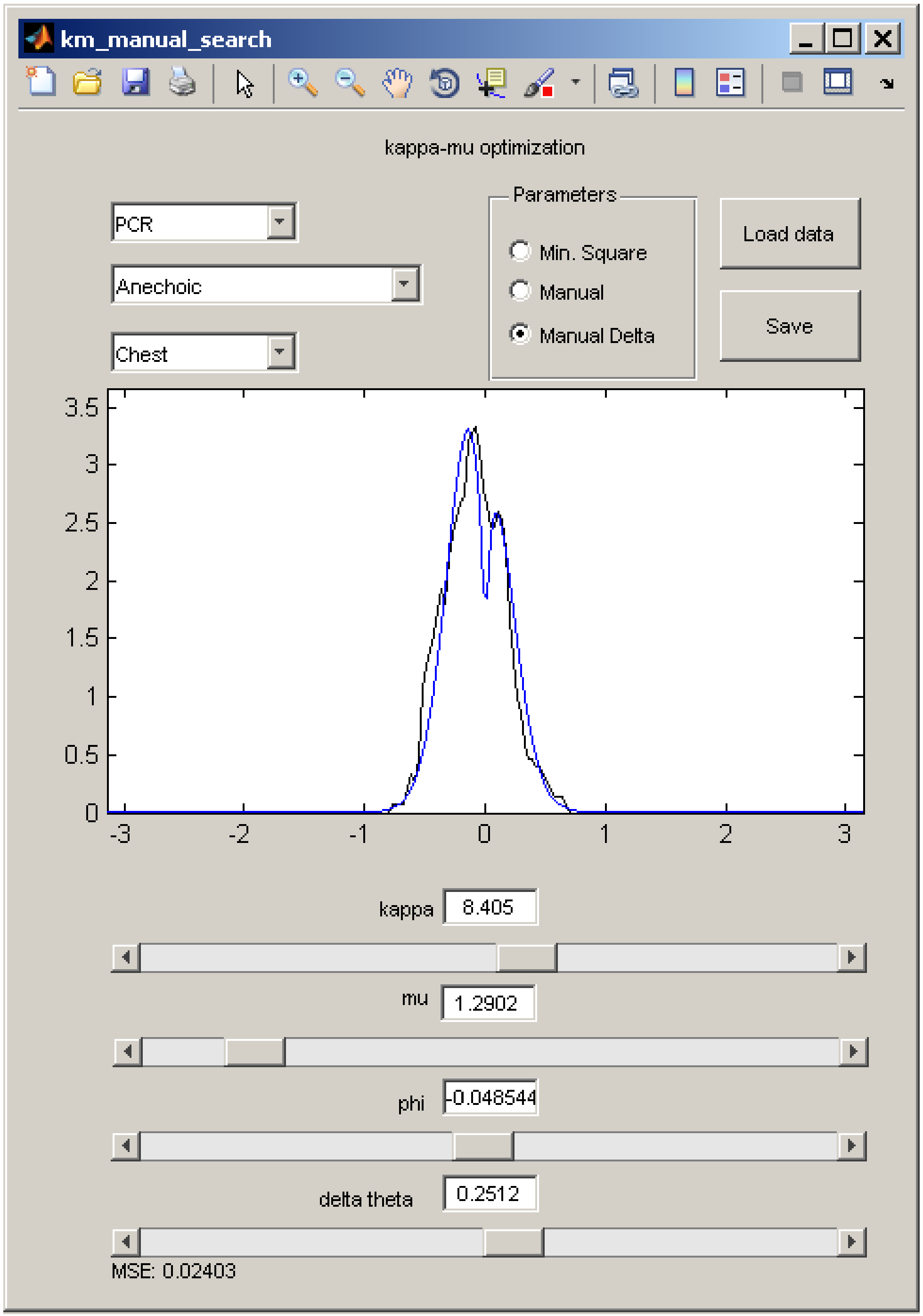
In order to aid the fitting of data to both the κ-μ and Nakagami-m models two graphical interfaces were created, which are shown in Figures A.1 and A.2. They were created as a MATLAB GUI with the aid of GUIDE, the graphical user interface development environment of MATLAB. Their usage is explained below.

κ-μ

In Figure A.1 the graphical interface created to aid the fitting of κ-μ curves is shown. Three drop-down menus are used to select the curve to be adjusted, which are located in the top left part of the interface. The top one selects the nature of the measurement being adjusted, which can be either the PCR or the phase PDF. The middle one selects between the kind of scenario, either the Anechoic Chamber or the Reverberation Chamber. The bottom one selects one of the seven receivers used in the Cotton and Scanlon experiments: Ankle, Chest, Elbow, Head, Knee, Waist or Wrist. The selected curve is loaded from a MATLAB variable containing the corresponding data.

The parameters can be adjusted either by moving one of the four sliders in the bottom part of the user interface or by editing the associated text box. The top three sliders select the parameters, namely κ, μ and ϕ and the bottom slider shifts the phase reference of the experimental curve. The κ and μ parameters are adimensional, while ϕ and delta theta are measured in radians. When a parameter is changed, the plot is updated in real time. The mean squared error between the experimental and theoretical points is shown in the bottom left part of the interface, after "MSE:", and is also updated in real time. It is a useful measure to guide the manual adjustments.

The adjusted parameters can be saved and loaded by using the two buttons in the top right part of the interface. A set of parameters to load is chosen by selecting the desired set in the "Parameters" radio button group and then pressing the "Load" button. Likewise, the parameters can be saved by selecting the desired option and then pressing the "Save" button. There are three possibilities to choose from when loading: "Min. Square", "Manual" and "Manual Delta". The "Min. Square" option loads the parameters from the non-linear optimization, which can be used as a starting point. There are two sets of manual adjustments to choose from, via the



km_manual_search

kappa-mu optimization

PCR

Anechoic

Chest

Parameters

○ Min. Square

○ Manual

● Manual Delta

kappa 8.405

mu 1.2902

phi -0.048544

delta theta 0.2512

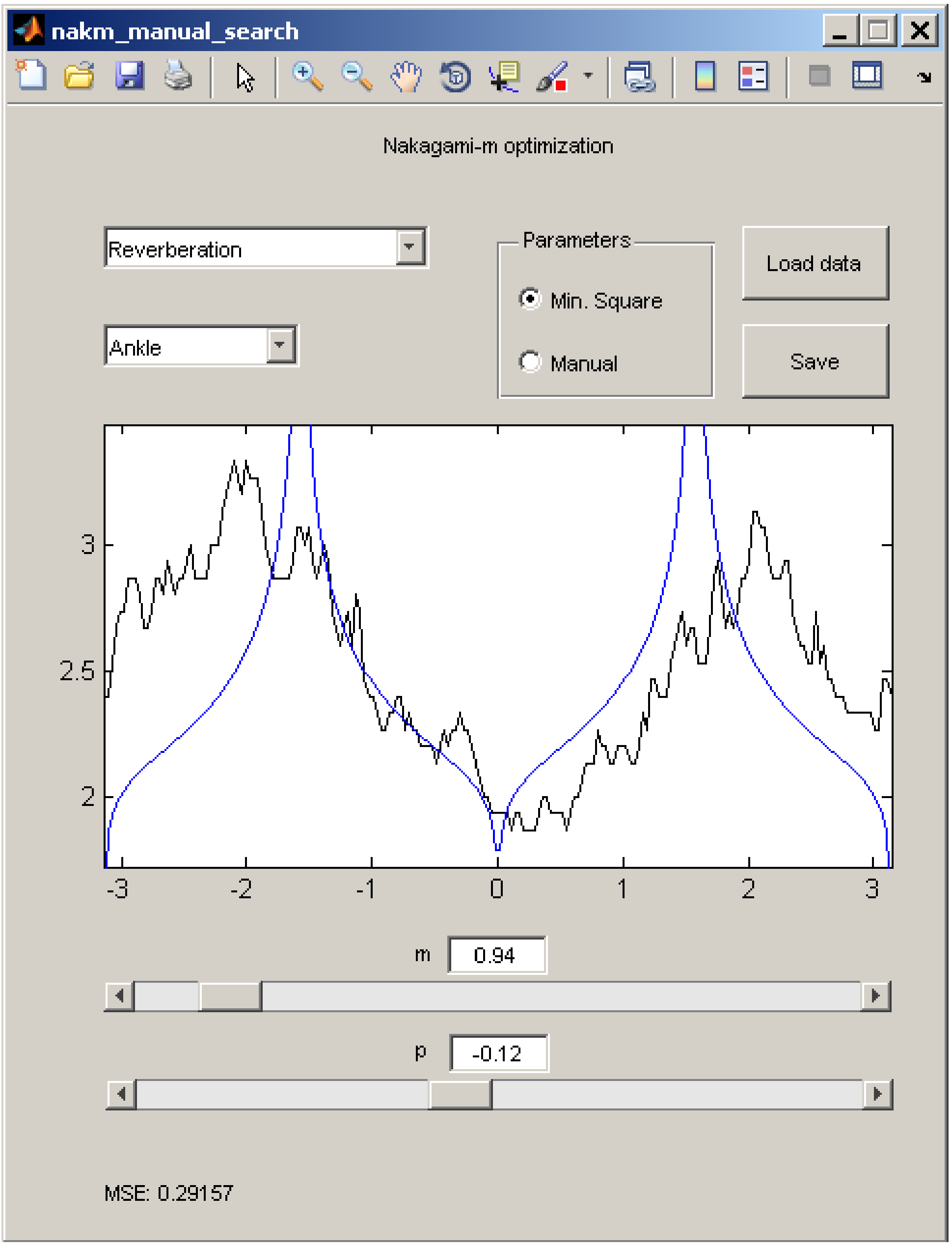
MSE: 0.02403

Figure A.1: Graphical interface created to manually adjust the experimental data to the κ-μ PCR and phase PDF.

”Manual” and ”Manual Delta” radio buttons. The difference between them is that the ”Manual Delta” option allows the user to save and load the parameters with a shift in the reference of the phase (delta theta), which is not allowed by the ”Manual” option. In the latter, the slider delta theta is always loaded as zero. When saving the parameters the user must first choose the desired radio button. If the radio button ”Min. Square” is selected, the parameters are saved to the ”Manual” variable.

Generalized Nakagami-m

The graphical interface created to aid the fitting of generalized Nakagami-m curves is shown in Figure A.2. It is very similar to the one created to fit the κ-μ curves. It was made to fit only the PCR curves, so it does not have the PCR - PDF drop down menu. It does not allow a shift in the reference of the phase, so neither the ”delta theta” slider nor the ”Manual Delta” option in the ”Parameters” radio button group are present. The generalized Nakagami-m distribution has only two parameters, the adimensional ”m” and ”p”, which can be adjusted by the corresponding sliders and text boxes.



Reverberation

Ankle

Parameters

● Min. Square

○ Manual

m 0.94

p -0.12

MSE: 0.29157

Figure A.2: Graphical interface created to manually adjust the experimental data to the generalized Nakagami-m PCR.

Appendix **B**

MATLAB code

This appendix presents the scritps used to generate the graphic interface.

B.1 κ-μ M-Files

B.1.1 Graphical Interface

```
function varargout = km_manual_search(varargin)
% KM_MANUAL_SEARCH M—file for km_manual_search.fig
%      KM_MANUAL_SEARCH, by itself, creates a new KM_MANUAL_SEARCH or raises the existing
%      singleton*.
%
%      H = KM_MANUAL_SEARCH returns the handle to a new KM_MANUAL_SEARCH or the handle to
%      the existing singleton*.
%
%      KM_MANUAL_SEARCH('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in KM_MANUAL_SEARCH.M with the given input arguments.
%
%      KM_MANUAL_SEARCH('Property','Value',...) creates a new KM_MANUAL_SEARCH or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before km_manual_search_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to km_manual_search_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help km_manual_search

% Last Modified by GUIDE v2.5 23—Jul—2012 19:24:21

% Begin initialization code — DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @km_manual_search_OpeningFcn, ...
    'gui_OutputFcn',  @km_manual_search_OutputFcn, ...
    'gui_LayoutFcn',  [] , ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code — DO NOT EDIT

% —— Executes just before km_manual_search is made visible.
function km_manual_search_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to km_manual_search (see VARARGIN)

% Choose default command line output for km_manual_search
handles.output = hObject;

set(hObject,'toolbar','figure');
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes km_manual_search wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% —— Outputs from this function are returned to the command line.
function varargout = km_manual_search_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% —— Executes on selection change in popupmenu_chamber.
function popupmenu_chamber_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_chamber (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_chamber contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_chamber
load_parameters(handles);
[theta y th_y] = get_curve_plots(handles);
plot_function(theta,y,th_y,handles);

% —— Executes during object creation, after setting all properties.
function popupmenu_chamber_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_chamber (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% —— Executes on selection change in popupmenu_position.
function popupmenu_position_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_position (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_position contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_position
load_parameters(handles);
[theta y th_y] = get_curve_plots(handles);
plot_function(theta,y,th_y,handles);

% —— Executes during object creation, after setting all properties.
```

```
function popupmenu_position_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_position (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% ——— Executes on button press in pushbutton_load.
function pushbutton_load_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_load (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
load_parameters(handles);
[theta y th_y] = get_curve_plots(handles);
plot_function(theta,y,th_y,handles);
%guidata(hObject,handles);

% ——— Executes on slider movement.
function slider_kappa_Callback(hObject, eventdata, handles)
% hObject    handle to slider_kappa (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
input = get(handles.slider_kappa,'Value');
handle_slider = handles.slider_kappa;
handle_text = handles.edit_kappa;
change_parameters(handle_slider, handle_text, handles, input);
plot_curve(handles);

% ——— Executes during object creation, after setting all properties.
function slider_kappa_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_kappa (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_kappa_Callback(hObject, eventdata, handles)
% hObject    handle to edit_kappa (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_kappa as text
%        str2double(get(hObject,'String')) returns contents of edit_kappa as a double
input = str2double(get(handles.edit_kappa,'String'));
if isempty(input)||isnan(input) input = get(handles.slider_kappa,'Value'); end
handle_slider = handles.slider_kappa;
handle_text = handles.edit_kappa;
change_parameters(handle_slider, handle_text, handles, input);
plot_curve(handles);

% ——— Executes during object creation, after setting all properties.
function edit_kappa_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_kappa (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on slider movement.
function slider_mu_Callback(hObject, eventdata, handles)
% hObject    handle to slider_mu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
input = get(handles.slider_mu,'Value');
handle_slider = handles.slider_mu;
handle_text = handles.edit_mu;
change_parameters(handle_slider, handle_text, handles, input);
plot_curve(handles);

% --- Executes during object creation, after setting all properties.
function slider_mu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_mu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end



function edit_mu_Callback(hObject, eventdata, handles)
% hObject    handle to edit_mu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_mu as text
%        str2double(get(hObject,'String')) returns contents of edit_mu as a double
input = str2double(get(handles.edit_mu,'String'));
if isempty(input)||isnan(input) input = get(handles.slider_mu,'Value'); end
handle_slider = handles.slider_mu;
handle_text = handles.edit_mu;
change_parameters(handle_slider, handle_text, handles, input);
plot_curve(handles);

% --- Executes during object creation, after setting all properties.
function edit_mu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_mu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider_phi_Callback(hObject, eventdata, handles)
% hObject    handle to slider_phi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
input = get(handles.slider_phi,'Value');
handle_slider = handles.slider_phi;
handle_text = handles.edit_phi;
change_parameters(handle_slider, handle_text, handles, input);
plot_curve(handles);

% --- Executes during object creation, after setting all properties.
function slider_phi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_phi (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_phi_Callback(hObject, eventdata, handles)
% hObject    handle to edit_phi (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_phi as text
%        str2double(get(hObject,'String')) returns contents of edit_phi as a double
input = str2double(get(handles.edit_phi,'String'));
if isempty(input)||isnan(input) input = get(handles.slider_phi,'Value'); end
handle_slider = handles.slider_phi;
handle_text = handles.edit_phi;
change_parameters(handle_slider, handle_text, handles, input);
plot_curve(handles);

% ——— Executes during object creation, after setting all properties.
function edit_phi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_phi (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% ——— Executes on slider movement.
function slider_delta_theta_Callback(hObject, eventdata, handles)
% hObject    handle to slider_delta_theta (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
input = get(handles.slider_delta_theta,'Value');
handle_slider = handles.slider_delta_theta;
handle_text = handles.edit_delta_theta;
change_parameters(handle_slider, handle_text, handles, input);
plot_curve(handles);

% ——— Executes during object creation, after setting all properties.
function slider_delta_theta_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_delta_theta (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_delta_theta_Callback(hObject, eventdata, handles)
% hObject    handle to edit_delta_theta (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_delta_theta as text
%        str2double(get(hObject,'String')) returns contents of edit_delta_theta as a double
input = str2double(get(handles.edit_delta_theta,'String'));
if isempty(input)||isnan(input) input = get(handles.slider_delta_theta,'Value'); end
handle_slider = handles.slider_delta_theta;
handle_text = handles.edit_delta_theta;
change_parameters(handle_slider, handle_text, handles, input);
```

```
plot_curve(handles);

% ——— Executes during object creation, after setting all properties.
function edit_delta_theta_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_delta_theta (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% ——— Executes on button press in pushbutton_save.
function pushbutton_save_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_save (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%newFolder = 'C:\Users\Iury\Documents\My Dropbox\MATLAB\workspace\Mestrado\Body Area Network\Data';
newFolder = 'D:\Dropbox\MATLAB\workspace\Mestrado\Body Area Network\Data';
oldFolder = cd(newFolder);
curve = get(handles.popupmenu_curve,'Value');
chamber = get(handles.popupmenu_chamber,'Value');
pos = get(handles.popupmenu_position,'Value');
[~, ~, ~, prmtrs, dt] = get_curve_plots(handles);
rb = get_parameters_radiobutton(handles);
switch curve
    case 1 %PDF
        switch rb
            case {1,2}
                load BAN_km_pdf_parameters_manual;
            case 3
                load BAN_km_pdf_parameters_manual_delta_theta;
        end
        switch chamber
            case 1 %anechoic
                delta_theta_ane(pos) = dt;
                km_pdf_par_ane(:,pos) = prmtrs';
            case 2 %reverberation
                delta_theta_rev(pos) = dt;
                km_pdf_par_rev(:,pos) = prmtrs';
        end
        switch rb
            case {1,2}
                save('BAN_km_pdf_parameters_manual','km_pdf_par_ane','km_pdf_par_rev',...
                    'delta_theta_ane','delta_theta_rev');
            case 3
                save('BAN_km_pdf_parameters_manual_delta_theta','km_pdf_par_ane',...
                    'km_pdf_par_rev','delta_theta_ane','delta_theta_rev');
        end
    case 2 %PCR
        switch rb
            case {1,2}
                load BAN_km_parameters_manual;
            case 3
                load BAN_km_parameters_manual_delta_theta;
        end
        switch chamber
            case 1 %anechoic
                delta_theta_ane(pos) = dt;
                km_par_ane(:,pos) = prmtrs';
            case 2 %reverberation
                delta_theta_rev(pos) = dt;
                km_par_rev(:,pos) = prmtrs';
        end
        switch rb
            case {1,2}
                save('BAN_km_parameters_manual','km_par_ane','km_par_rev',...
                    'delta_theta_ane','delta_theta_rev');
            case 3
```

```
                save('BAN_km_parameters_manual_delta_theta','km_par_ane',...
                    'km_par_rev','delta_theta_ane','delta_theta_rev');
        end
end
cd(oldFolder);

% --- Executes on selection change in popupmenu_curve.
function popupmenu_curve_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_curve (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_curve contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_curve
load_parameters(handles);
[theta y th_y] = get_curve_plots(handles);
plot_function(theta,y,th_y,handles);

% --- Executes during object creation, after setting all properties.
function popupmenu_curve_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_curve (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function change_parameters(handle_slider, handle_text, handles, input)
slider_max = get(handle_slider,'Max');
slider_min = get(handle_slider,'Min');

if input>slider_max
    input = slider_max;
elseif input<slider_min
    input = slider_min;
end
set(handle_slider,'Value',input);
set(handle_text,'String',num2str(input));
%[theta y th_y] = get_curve_plots(handles);
%plot_function(theta,y,th_y,handles);

function plot_curve(handles)
[theta y th_y] = get_curve_plots(handles);
plot_function(theta,y,th_y,handles);

function load_parameters(handles)
curve = get(handles.popupmenu_curve,'Value');
chamber = get(handles.popupmenu_chamber,'Value');
pos = get(handles.popupmenu_position,'Value');
rb = get_parameters_radiobutton(handles);
%load BAN_data
switch curve
    case 1 %PDF
        switch rb
            case 1 %msd
                load BAN_km_pdf_parameters;
            case 2 %manual
                load BAN_km_pdf_parameters_manual;
            case 3
                load BAN_km_pdf_parameters_manual_delta_theta;
        end
        if chamber == 1 %anechoic
            prmtrs = km_pdf_par_ane;
        else %reverberation
            prmtrs = km_pdf_par_rev;
        end
    case 2 %PCR
        switch rb
```

```
            case 1 %msd
                load BAN_km_parameters;
            case 2 %manual
                load BAN_km_parameters_manual;
            case 3
                load BAN_km_parameters_manual_delta_theta;
        end
        if chamber == 1 %anechoic
            dt = delta_theta_ane(pos);
            prmtrs = km_par_ane;
        else %reverberation
            dt = delta_theta_rev(pos);
            prmtrs = km_par_rev;
        end
end
if rb ~= 3
    dt = 0;
end
prmtrs = prmtrs(:,pos);
kappa = prmtrs(1);
mu = prmtrs(2);
phi = prmtrs(3);
change_parameters(handles.slider_kappa, handles.edit_kappa, handles, kappa);
change_parameters(handles.slider_mu, handles.edit_mu, handles, mu);
change_parameters(handles.slider_phi, handles.edit_phi, handles, phi);
change_parameters(handles.slider_delta_theta, handles.edit_delta_theta, handles, dt);
plot_curve(handles);

function [theta y th_y prmtrs delta_theta] = get_curve_plots(handles)
load BAN_data;
curve = get(handles.popupmenu_curve,'Value');
chamber = get(handles.popupmenu_chamber,'Value');
pos = get(handles.popupmenu_position,'Value');
switch curve
    case 1 %PDF
        switch chamber
            case 1 %anechoic
                y = pdf_ane;
            case 2 %reverberation
                y = pdf_rev;
        end
    case 2 %PCR
        switch chamber
            case 1 %anechoic
                y = pcr_ane;
            case 2 %reverberation
                y = pcr_rev;
        end
end
kappa = get(handles.slider_kappa,'Value');
mu = get(handles.slider_mu,'Value');
phi = get(handles.slider_phi,'Value');
delta_theta = get(handles.slider_delta_theta,'Value');
theta = theta;
y = y(:,pos);
y = shift_y(theta, y, delta_theta)';
switch curve
    case 1 %PDF
        th_y = km_ppdf(theta, [kappa mu phi])';
        prmtrs = [kappa mu phi];
    case 2 %PCR
        th_y = km_pcr(theta, [kappa mu phi])';
        [th_y omega] = bestfit(th_y,y);
        prmtrs = [kappa mu phi omega];
end

function y_shifted = shift_y(theta,y,delta_theta)
delta_theta = delta_theta + pi;
if delta_theta > 2*pi || delta_theta < 0
    delta_theta = rem(delta_theta ,2*pi);
end
```

```
delta_theta = delta_theta - pi;
N = numel(theta);
N_shift = floor(N*delta_theta/2/pi);
y_index = 0:N-1;
y_index_shifted = rem(y_index-N_shift+N,N)+1;
y_shifted(y_index+1) = y(y_index_shifted);

function plot_function(theta,y,th_y,handles)
axes(handles.axes1);
y_variation = max(y) - min(y);
%ymin = min(y) - 0.1*y_variation;
ymin = 0;
ymax = max(y) + 0.1*y_variation;
plot(theta,y,'k');
axis([-pi pi ymin ymax]);
hold on;
plot(theta,th_y);
hold off;
mse = mean((th_y - y).^2); %mean squared error
set(handles.text_mse,'String',['MSE: ' num2str(mse)]);
%guidata(hObject,handles);

function [w a]  = bestfit(X, Y)
% Try to find a that best fit (minimum squares): X * a = Y
% Returns a and w = X * a;
m = size(X,1);
n = size(X,2);
if n>m
    a = Y * pinv(X);
    w = a * X;
else
    a = pinv(X) * Y;
    w = X * a;
end

function rb = get_parameters_radiobutton(handles)
if get(handles.radiobutton_msd,'Value') == 1
    rb = 1; %minimum square distance parameters selected
elseif get(handles.radiobutton_manual,'Value') == 1
    rb = 2; %manual parameters selected
else
    rb = 3;
end

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit_delta_theta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_delta_theta as text
%        str2double(get(hObject,'String')) returns contents of edit_delta_theta as a double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_delta_theta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

B.1.2 κ-μ PCR

```
function [ y ] = km_pcr( theta, parameters )
%pcr for f_d = 1, in which f_d is the maximum Doppler shift in Hz
```

```
K = parameters(1);
M = parameters(2);
P = parameters(3);
y = zeros(size(theta));
rho = linspace(0.01,15,100);
f = nrt(rho,theta,K,M,P);
y = sum(f,1)*(rho(2)-rho(1));
end


function [ f ] = nrt(rho, theta, kappa, mu ,phi)
%KM_NRT returns N(rho, theta), which if integrated from 0 to infinity in
%respect to rho yelds the PCR of the kappa-mu distribution.

rho = rho(:); % rho is a column vector
theta = theta(:)'; %theta is a row vector

f_const = (2*pi)*(1/sqrt(32*pi))*(mu^(3/2))*(kappa^(1-mu/2))*((1+kappa)^(1/2 + mu/2))*exp(-kappa*mu);
f_rho = (rho.^mu).*exp(-mu*(1+kappa)*rho.^2);
f_theta = ((abs(sin(2.*theta)).^(mu./2))./(abs(sin(2.*phi))).^((mu./2)-1));
f_rho_theta =  ...
    exp(2*mu*sqrt(kappa*(1+kappa))*rho*cos(theta-phi)).*k_func((mu./2)-1,2.*mu.*sqrt(kappa.*(1+kappa)).*rho*abs(c
f = f_const*(f_rho*f_theta).*f_rho_theta; %[size_rho x size_theta]

end
```

B.1.3 κ-μ Phase PDF

```
function [ y ] = km_ppdf( theta, parameters )
K = parameters(1);
M = parameters(2);
P = parameters(3);

theta = theta(:)';

st = numel(theta);
rho = linspace(0.005,1.0,200);
rho_inv = 1./rho;
rep_rho = repmat(rho, st,1)'; %[sr x st]
s = rho(2) - rho(1);
w1 = pdf_rt(rho,theta, K, M, P);
w2 = pdf_rt(rho_inv,theta, K, M, P)./(rep_rho.^2);
y = sum(w1+w2)*s;

t = isnan(y)|isinf(y);
y(t) = 0;
end


function [ f ] = pdf_rt(rho, theta, K, M ,P)
%PDF_RT returns the phase-enveloppe joint distribution f(rho, theta) for
%the kappa-mu model

%let size(theta) = [1 x st] and size(rho) = [1 x sr]
st = numel(theta);
sr = numel(rho);
b = exp(-K*M)*(1/(((2*K*M)^M)*8))/((abs(sin(2.*P))).^((M./2)-1)); %b is a multiplying constant
rep_theta = repmat(theta,sr,1); %size(f1) = [sr x st]
rep_rho = repmat(rho,st,1)';%size(f1) = [sr x st]
f1 = (abs(sin(2*rep_theta)).^(M./2)); % size(f1) = [sr x st];
f2 = (rep_rho.^(M+1)); %size(f2) = [sr x st]
f3 = exp(-(rep_rho.^2)/(4*K*M)+ rho'*cos(theta-P)); %size(f3) = [sr x st]
f4 = k_func((M/2)-1,rho'*abs(cos(theta)*cos(P))).*k_func((M/2)-1,rho'*abs(sin(theta)*sin(P))); ...
    %size(f4) = [sr x st]
f = b*f1.*f2.*f3.*f4;%size(f) = [sr x st]
end
```

B.2 Generalized Nakagami-m M-Files

B.2.1 Graphical Interface

```
function varargout = nakm_manual_search(varargin)
% NAKM_MANUAL_SEARCH M—file for nakm_manual_search.fig
%      NAKM_MANUAL_SEARCH, by itself, creates a new NAKM_MANUAL_SEARCH or raises the existing
%      singleton*.
%
%      H = NAKM_MANUAL_SEARCH returns the handle to a new NAKM_MANUAL_SEARCH or the handle to
%      the existing singleton*.
%
%      NAKM_MANUAL_SEARCH('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in NAKM_MANUAL_SEARCH.M with the given input arguments.
%
%      NAKM_MANUAL_SEARCH('Property','Value',...) creates a new NAKM_MANUAL_SEARCH or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before nakm_manual_search_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to nakm_manual_search_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help nakm_manual_search

% Last Modified by GUIDE v2.5 15—Dec—2011 17:22:13

% Begin initialization code — DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @nakm_manual_search_OpeningFcn, ...
                   'gui_OutputFcn',  @nakm_manual_search_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code — DO NOT EDIT

% ——— Executes just before nakm_manual_search is made visible.
function nakm_manual_search_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to nakm_manual_search (see VARARGIN)

% Choose default command line output for nakm_manual_search
handles.output = hObject;

set(hObject,'toolbar','figure');

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes nakm_manual_search wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = nakm_manual_search_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in popupmenu_chamber.
function popupmenu_chamber_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_chamber (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_chamber contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_chamber
load_parameters(handles);
[theta pcr th_pcr] = get_pcr_plots(handles);
plot_function(theta,pcr,th_pcr,handles);

% --- Executes during object creation, after setting all properties.
function popupmenu_chamber_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_chamber (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu_position.
function popupmenu_position_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_position (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu_position contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_position
load_parameters(handles);
[theta pcr th_pcr] = get_pcr_plots(handles);
plot_function(theta,pcr,th_pcr,handles);

% --- Executes during object creation, after setting all properties.
function popupmenu_position_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_position (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in load_button.
function load_button_Callback(hObject, eventdata, handles)
% hObject    handle to load_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
load_parameters(handles);
[theta pcr th_pcr] = get_pcr_plots(handles);
plot_function(theta,pcr,th_pcr,handles);
guidata(hObject,handles);

% --- Executes on slider movement.
function slider_m_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to slider_m (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider

input = get(hObject,'Value');
change_slider_m(input,handles);
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function slider_m_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_m (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider_p_Callback(hObject, eventdata, handles)
% hObject    handle to slider_p (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
input = get(hObject,'Value');
change_slider_p(input,handles);
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function slider_p_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_p (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_p_Callback(hObject, eventdata, handles)
% hObject    handle to edit_p (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_p as text
%        str2double(get(hObject,'String')) returns contents of edit_p as a double
input = str2double(get(hObject,'String'));

%checks to see if input is empty. if so, default input1_editText to zero
if (isempty(input))
     set(hObject,'String','0')
end
change_slider_p(input,handles);
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function edit_p_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_p (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end

function edit_m_Callback(hObject, eventdata, handles)
% hObject    handle to edit_m (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_m as text
%        str2double(get(hObject,'String')) returns contents of edit_m as a double

input = str2double(get(hObject,'String'));

%checks to see if input is empty. if so, default input1_editText to zero
if (isempty(input))
     set(hObject,'String','0')
end
change_slider_m(input,handles);
guidata(hObject, handles);

% ——— Executes during object creation, after setting all properties.
function edit_m_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_m (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function change_slider_m(input,handles)
slider_max = get(handles.slider_m,'Max');
slider_min = get(handles.slider_m,'Min');

if input>slider_max
    input = slider_max;
elseif input<slider_min
    input = slider_min;
end
set(handles.slider_m,'Value',input);
set(handles.edit_m,'String',num2str(input));
[theta pcr th_pcr] = get_pcr_plots(handles);
plot_function(theta,pcr,th_pcr,handles);

function change_slider_p(input,handles)
slider_max = get(handles.slider_p,'Max');
slider_min = get(handles.slider_p,'Min');
if input>slider_max
    input = slider_max;
elseif input<slider_min
    input = slider_min;
end
set(handles.slider_p,'Value',input);
set(handles.edit_p,'String',num2str(input));
[theta pcr th_pcr] = get_pcr_plots(handles);
plot_function(theta,pcr,th_pcr,handles);

function prmtrs = load_parameters(handles)
rb = get_parameters_radiobutton(handles);
switch rb
    case 1 %msd
        load BAN_nakm_parameters;
    case 2 %manual
        load BAN_nakm_parameters_manual;
end
chamber = get(handles.popupmenu_chamber,'Value');
pos = get(handles.popupmenu_position,'Value');
switch chamber
    case 1
        prmtrs = nakm_par_ane;
    case 2
        prmtrs = nakm_par_rev;
end
prmtrs = prmtrs(:,pos);
m = prmtrs(1);
p = prmtrs(2);
change_slider_m(m,handles);
change_slider_p(p,handles);

function [theta pcr th_pcr prmtrs] = get_pcr_plots(handles)
load BAN_data;
chamber = get(handles.popupmenu_chamber,'Value');
pos = get(handles.popupmenu_position,'Value');
switch chamber
    case 1
        pcr = pcr_ane;
    case 2
        pcr = pcr_rev;
end
m = get(handles.slider_m,'Value');
p = get(handles.slider_p,'Value');
theta = theta;
pcr = pcr(:,pos);
th_pcr = nakmp_pcr(theta,m,p);
[th_pcr omega] = bestfit(th_pcr,pcr);
prmtrs = [m p omega];

function plot_function(theta,pcr,th_pcr,handles)
axes(handles.axes1);
pcr_variation = max(pcr) - min(pcr);
ymin = min(pcr) - 0.1*pcr_variation;
ymax = max(pcr) + 0.1*pcr_variation;
plot(theta,pcr,'k');
axis([-pi pi ymin ymax]);
hold on;
plot(theta,th_pcr);
hold off;
mse = mean((th_pcr - pcr).^2); %mean squared error
hObject = handles.text_mse;
set(hObject,'String',['MSE: ' num2str(mse)]);
guidata(hObject,handles);

function [w a]  = bestfit(X, Y)
% Try to find a that best fit (minimum squares): X * a = Y
% Returns a and w = X * a;
m = size(X,1);
n = size(X,2);
if n>m
    a = Y * pinv(X);
    w = a * X;
else
    a = pinv(X) * Y;
    w = X * a;
end

% --- Executes on button press in pushbutton_save.
function pushbutton_save_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%newFolder = 'C:\Users\Iury\Documents\My Dropbox\MATLAB\workspace\Mestrado\Body Area Network\Data';
newFolder = 'D:\Dropbox\MATLAB\workspace\Mestrado\Body Area Network\Data';
oldFolder = cd(newFolder);
load BAN_data;
load BAN_nakm_parameters_manual;
chamber = get(handles.popupmenu_chamber,'Value');
pos = get(handles.popupmenu_position,'Value');
[~, ~, ~, prmtrs] = get_pcr_plots(handles);
switch chamber
    case 1
        nakm_par_ane(:,pos) = prmtrs';
    case 2
        nakm_par_rev(:,pos) = prmtrs';
end
save('BAN_nakm_parameters_manual','nakm_par_ane','nakm_par_rev');
cd(oldFolder);

function rb = get_parameters_radiobutton(handles)
if get(handles.radiobutton_msd,'Value') == 1
    rb = 1; %minimum square distance parameters selected
else
    rb = 2; %manual parameters selected
end
```

B.2.2 Generalized Nakagami-m PCR

```
function [ f ] = nakmp_pcr( theta, m,p )
f = (sqrt(pi)*(abs(sin(2*theta)).^(m-1)).*(abs(tan(theta)).^(-p*m)).*...
    gamma(m-1/2))/((2^(m+(1/2)))*gamma((1+p)*m/2)*gamma((1-p)*m/2));
end
```

B.2.3 Generalized Nakagami-m Phase PDF

```
function [ f ] = nakmp_pdf( phase, m , p )
f = (gamma(m)/((2^m)*gamma((1+p)*m/2)*gamma((1-p)*m/2)))*...
    (abs(sin(2*phase)).^(m-1))./(abs(tan(phase)).^(p*m));
end
```

References

[1] M. D. Yacoub, *Foundations of Mobile Radio Engineering*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1993.

[2] S. O. Rice, “Statistical properties of sine wave plus random noise,” *Bell System Technical Journal*, vol. 27, pp. 109 – 157, Jan. 1948.

[3] W. C. Jakes, *Microwave Mobile Communications.* John Wiley & Sons Inc, 1975.

[4] D. Hess, “Cycle slipping in a first-order phase-locked loop,” *Communication Technology, IEEE Transactions on*, vol. 16, no. 2, pp. 255 –260, april 1968.

[5] M. Yacoub, “The κ-μ distribution and the η- μ distribution,” *Antennas and Propagation Magazine, IEEE*, vol. 49, no. 1, pp. 68 –81, feb. 2007.

[6] ——, “Nakagami-m phase - envelope joint distribution: A new model,” *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 3, pp. 1552 –1557, march 2010.

[7] M. Nakagami, “The m-distribution, a general formula of intensity of rapid fading,” in *Statistical Methods in Radio Wave Propagation: Proceedings of a Symposium held June 18-20, 1958*, W. C. Hoffman, Ed. Pergamon Press, june 1960, pp. 3–36.

[8] H. Suzuki, “A statistical model for urban radio propogation,” *Communications, IEEE Transactions on*, vol. 25, no. 7, pp. 673 – 680, jul 1977.

[9] M. Yacoub, J. Vargas B., and L. De R. Guedes, “On the statistics of the nakagami-m signal,” in *Telecommunications Symposium, 1998. ITS ’98 Proceedings. SBT/IEEE International*, vol. 2, aug 1998, pp. 377 –382 vol.2.

[10] B. Sklar, “Rayleigh fading channels in mobile digital communication systems .i. characterization,” *Communications Magazine, IEEE*, vol. 35, no. 7, pp. 90 –100, jul 1997.

[11] G. L. Stüber, *Principles of Mobile Communication (2nd Edition).* Springer, 2000.

[12] S. Cotton and W. Scanlon, “Higher-order statistics for κ-μ distribution,” *Electronics Letters*, vol. 43, no. 22, 25 2007.

[13] U. Dias and M. Yacoub, “The κ-μ phase-envelope joint distribution,” *Communications, IEEE Transactions on*, vol. 58, no. 1, pp. 40 –45, january 2010.

[14] R. H. Clarke, “A statistical theory of mobile-radio reception,” *Bell Systems Technical Journal*, vol. 47, pp. 957–1000, 1968.

[15] U. Dias, M. Yacoub, and D. da Costa, “The κ-μ phase-envelope joint distribution,” in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, sept. 2008, pp. 1 –5.

[16] E. W. Weisstein, “Gamma function.” [Online]. Available: http://mathworld.wolfram.com/GammaFunction.html

[17] ——, “Erf.” [Online]. Available: http://mathworld.wolfram.com/Erf.html

[18] Y. Li, J. Chen, Y. Song, and R. Lin, “On the study of the crossing statistics of phase processes in rice mobile fading channels,” in *Wireless, Mobile and Sensor Networks, 2007. (CCWMSN07). IET Conference on*, dec. 2007, pp. 158 –161.

[19] D. da Costa, M. Yacoub, J. Filho, G. Fraidenraich, and J. Mendes, “Generalized nakagami-m phase crossing rate,” *Communications Letters, IEEE*, vol. 10, no. 1, pp. 13 – 15, jan 2006.

[20] E. W. Weisstein, “Taylor series.” [Online]. Available: http://mathworld.wolfram.com/TaylorSeries.html

[21] ——, “Probability density function.” [Online]. Available: http://mathworld.wolfram.com/ProbabilityDensityFunction.html

[22] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions.* Wiley-Interscience, 2000.

[23] M. Yacoub, G. Fraidenraich, and J. Santos Filho, “Nakagami-m phase-envelope joint distribution,” *Electronics Letters*, vol. 41, no. 5, pp. 259 – 261, march 2005.

[24] Wolfram Research, Inc., *Mathematica Edition: Version 8.0.1.* Champaign, Illinois: Wolfram Research, Inc., 2011.

[25] D. Young and N. Beaulieu, “The generation of correlated rayleigh random variates by inverse discrete fourier transform,” *Communications, IEEE Transactions on*, vol. 48, no. 7, pp. 1114 –1127, jul 2000.

[26] A. G. Glen, L. M. Leemis, and Berger, “A generalized univariate change-of-variable transformation technique,” 1997.

[27] S. Cotton, U. Dias, W. Scanlon, and M. Yacoub, “On the distribution of signal phase in body area networks,” *Communications Letters, IEEE*, vol. 14, no. 8, pp. 728 –730, august 2010.

[28] MATLAB, *version 7.10.0 (R2010a).* Natick, Massachusetts: The MathWorks Inc., 2010.

[29] J. Filho, M. Yacoub, and G. Fraidenraich, “A simple accurate method for generating autocorrelated nakagami-m envelope sequences,” *Communications Letters, IEEE*, vol. 11, no. 3, pp. 231 –233, march 2007.