A HARDWARE BASED RICEAN FADING

RADIO CHANNEL SIMULATOR

by

Ahmed Mohamed H. Badawy

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2010

APPROVAL

of a thesis submitted by

Ahmed Mohamed H. Badawy

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citation, bibliographic style, and consistency and is ready for submission to the Division of Graduate Education.

Dr. Richard Wolff

Approved for the Department of Electrical Engineering

Dr. Robert C. Maher

Approved for the Division of Graduate Education

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Ahmed Mohamed H. Badawy

April 2010

ACKNOWLEDGEMENTS

I would like to thank Prof. Richard Wolff for giving me the opportunity to be a part of an extraordinary team that made working at The Department of Electrical and Computer Engineering an amazing and wonderful research experience. It was a great honor to work under his supervision. I learned a lot about problem solving, management and leadership skills from him in addition to of course the academic knowledge. I would like to thank Mr. Andy Olson who was always there for me when I had a technical problem. I also gained a lot of technical experience from him. I would like to thank Mr. Aaron Traxinger for his help in the experimental set up.

I would like to thank my father, Mohamed Badawy, whose constant support made it possible for me to always succeed in both my personal and academic lives. I also would like to thank and dedicate this thesis my mother's soul whom I know would have been so proud of me. I would like to thank my brother, Sameh, and sister, Nilly, for their constant support.

Most of all, I would like to thank God, for the blessings and sound belief in Him, health, and sanity and for putting me in a path that allowed me to meet people that have been kind to me and allowing me the opportunity to reciprocate.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

ABSTRACT

As the signal propagates from the transmitter to the receiver, it is subjected to fading. One of the fading distributions is Ricean distribution that assumes a line of sight signal and an infinite number of multipath signals between the transmitter and the receiver. The key parameter in the Ricean distribution is the *K* factor which is a measure of the severity of fading; the lower the *K* value, the more severe the fading. The problem was to develop a hardware based fading simulator that is based on the Ricean distribution. The key component of the hardware is an eight channel beamformer board; each channel has a 6-bit attenuator and a 6-bit phase shifter controlled by a on board FPGA for beamforming purpose. To approximate the Ricean fading, one of the eight channels was designated the line of sight signal, and the seven other channels were used to emulate the multipath signals. Cables with different lengths were used as delay lines to emulate the actual delays that occur to the multipath signals. The signals were then normalized to account for the cable losses. The fade simulator was controlled using a MATLAB program that sent a serial bit stream to the attenuators and phase shifters on the beamformer board. The MATLAB program generated random attenuations and phase shifts according to the Ricean distribution for the inputted *K* value. For a constant amplitude input RF signal, as *K* was increased, the fluctuation in the signal at the output of the fade simulator decreased. The estimated *K* values from the collected data were close to the inputted *K* values. The probability density functions (PDFs), the cumulative density functions (CDFs) and the level crossing rates (LCRs) of the collected data compared closely to the theoretical Ricean fading distribution PDFs, CDFs and LCRs, which showed that the fade simulator provides an accurate simulation of Ricean fading. The fade simulator was used to evaluate the performance of Airspan and Harris radios. The throughput of the radios was tested under different *K* values and for different SNR. As *K* increased, the fluctuation in the signal decreased which led to higher throughput.

CHAPTER ONE

INTRODUCTION

In today's telecommunication networks, both wireless and wired communications equipments are deployed to serve different applications. Both wireless and wired communications have advantages and disadvantages. One of the main advantages of wireless communications over wired communication is cost. For users in rural areas, wireless communication becomes a favorable solution. One of the main advantages of wired communication over wireless communication is privacy and security. In wireless communication, the transmitted signal could be received by unwanted users.

Another main issue with wireless communication is fading. Fading causes the received signal strength to fluctuate such that this received signal might go below the receiver's sensitivity. The fading statistics are highly related to the environment where the signal propagates. In contrast to wired communication, loss in wireless systems is temporally and spatially variable.

A study of the fading statistics is very important for an efficient transceiver design as well as link budget calculations.

It's highly desirable to be able to reproduce the environment conditions in the lab to be able to evaluate the performance of different radios and to ensure a successful communication instead of field tests, which are time consuming, expensive and difficult to control.

The goal of this research was to develop a Ricean fading channel simulator to evaluate the performance of different radios under controllable fading conditions. My

contribution to this research was to assemble the simulator, develop the software needed to run the fade simulator, design hardware to normalize the signals, collect and analyze data and to validate the results.

CHAPTER TWO

BACKGROUND INFORMTION

In wireless communications, as the signal travels from the transmitter to the receiver, the environment or the channel between the transmitter and the receiver plays an important role in the success of the communication. The attenuation in the signal is related to the separation between the transmitter and the receiver and other factors.

Propagation Mechanisms

As the signal propagates through the channel, it's subjected to:

Reflection

When a smooth surface with dimensions much greater than signal's wavelength obstructs the travelling signal path, the signal is reflected. Parameters such as the angle of incidence and the surface material determine the percentage of the signal energy that gets reflected and the percentage that gets absorbed.

Diffraction

When a dense body with dimensions much greater than signal's wavelength obstructs the travelling signal, a secondary signal is formed behind the body. The secondary signal is formed in what's called the shadowed region, which is a function of the obstruction dimension and the radio signal frequency.

Scattering

When a rough surface with dimensions much greater than the signal's wavelength or a surface with dimensions comparable to the signal's wavelength obstructs the travelling signal, the signal's energy spreads out causing scattered signals. The scattered signals arrive at the receiver with different strengths and different phase shifts. At the receiver end, the scattered signals might add up or cancel out.

Figure 1 shows the multipath effect due to reflection, diffraction and scattering. The transmitter in this case is a fixed base station and the receiver is a mobile subscriber station.

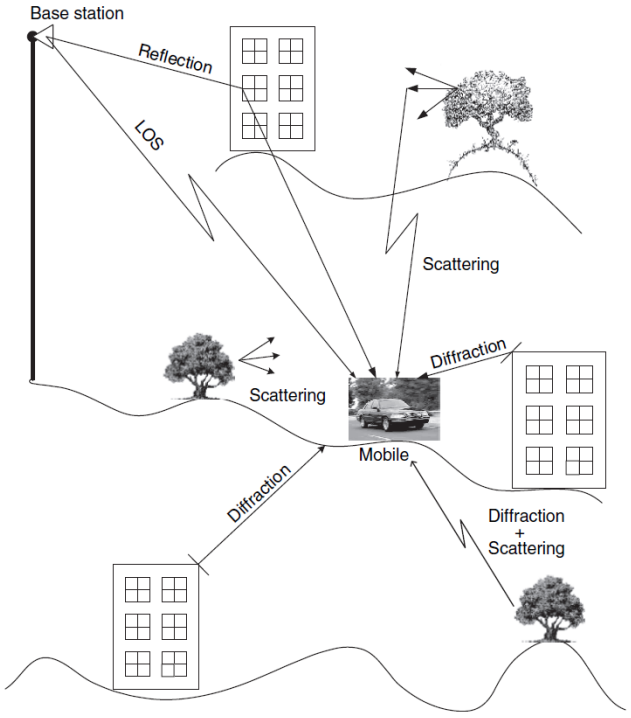


Figure 1: Multipath effect due to reflection, diffraction and scattering.

Attenuation due to distance, along with the propagation mechanisms explained above, cause the transmitted signal to fade. This fading can be categorized into two main categories:

Large Scale Fading

Large scale fading or large scale path loss is mainly due to attenuation to the propagating signal due to distance. Knowledge of path loss is a key to calculating the link budget. A communication between a transmitter and a receiver is considered successful if the received signal is higher than the receiver's sensitivity. Therefore, a careful link budget calculation plays a significant role in estimating the success of the communication. There are two main path loss models described in [1]:

Log-distance Path Loss Model. In this model, the received signal strength depends only on the distance between the transmitter and the receiver. The average received signal strength decreases logarithmically with distance (d) as shown in Equation (1).

$$\overline{PL}(d)\,\alpha\left(\frac{d}{d_0}\right)^n \tag{1}$$

$$\overline{PL}(d)=\overline{PL}(d_0)+10n\log\left(\frac{d}{d_0}\right) \tag{2}$$

Where n is the path loss exponent, d_0 is the reference distance. The path loss exponent depends on the environment or the channel between the transmitter and the receiver. Table 1 shows the path loss exponent for different environments. The path loss calculated using the free space model is considered the reference path loss. The free space model

treats the environment between the transmitter and the receiver as being free of all obstacles that might reflect or diffract the signal. In the free space model, the effect of atmospheric absorption is neglected. The propagating signal is also considered far away from the earth's surface. Table 1 also indicates that the more obstacles the environment has the higher the path loss exponent. The higher the path loss exponent the more path loss introduced for a fixed distance. The reference distance (d_0) differs based on the environment; for large coverage area d_0 is 1 km, for microcell systems d_0 is 100 m, and for indoor propagation d_0 is 1 m. d_0 must always be in the antenna's far field to avoid near field effects.

Table 1: Path Loss Exponents for Different Environments [1]

Environment	Path Loss Exponent, *n*
Free Space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Log-normal Shadowing. The received signal strength at two different locations having the same distance (d) between the transmitter and the receiver might be vastly different from the calculated average using Equation (1) because of the differences in the environment. For any particular d, at a particular location, the path loss is random and log-normally distributed about the mean distance dependant value.

$$PL(d)[dB] = \overline{PL}(d_0) + X_\sigma = \overline{PL}(d_0) + 10n\log\left(\frac{d}{d_0}\right) + X_\sigma \qquad (3)$$

Equation (3) states that the path loss at a particular distance *d* equals to the average path loss calculated using Equation (1) plus a random variable X_σ. X_σ is a zero mean Gaussian random variable in dB with a standard deviation σ also in dB. The received signal strength ($P_r(d)$) in *dBm* is given by

$$P_r(d)[dBm] = P_t[dBm] - PL(d)[dB] \tag{4}$$

Where $P_t(d)$ is the transmitted power in *dBm*.

From the actual measurements and using linear regression methods, the values of *n* and σ are adjusted so that the estimated path loss using Equation (3) is as close as possible to the actual measured path loss.

The Q-function or the error function (*erf*) can be used to determine the probability that received signal level will be above or below a particular given level for lognormal distribution. The Q-function is given by

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty \exp\left(-\frac{x^2}{2}\right) dx = \frac{1}{2}\left[1 - erf\left(\frac{z}{\sqrt{2}}\right)\right] \tag{5}$$

$$Q(z) = 1 - Q(-z) \tag{6}$$

The probability that the received signal will be above a particular given level γ is given by

$$\Pr[P_r(d) > \gamma] = Q\left(\frac{\gamma - \overline{P_r(d)}}{\sigma}\right) \tag{7}$$

The probability that the received signal will be below a particular given level γ is given by

$$\Pr[P_r(d) < \gamma] = Q\left(\frac{\overline{P_r(d)} - \gamma}{\sigma}\right) \quad (8)$$

Small Scale Fading

As explained at the beginning of the chapter, as the signal propagates, it is subjected to reflection, diffraction and scattering. The transmitted signal is scattered into a large number of signals travelling in different paths and arriving at the receiver at different times with different attenuations and phase shifts. The scattered signals are called the multipath waves. The received signal is a voltage summation of all the multipath waves. The multipath waves with different phase shifts can cancel each other out causing fading.

As explained in [1], the physical factors that can affect the small scale fading are:

Multipath Propagation. As explained before, the transmitted signal is scattered into a large number of signals with different amplitudes and phase shifts arriving at the receiver at slightly different times. The summation of these time variable multipath signals cause a rapid fluctuation in the received signal strength. Multipath propagation can also cause intersymbol interference since it causes a spread in the arrival time of a pulse into the next pulse's time interval.

Speed of the Mobile. When either the transmitter or the receiver moves, the motion causes a Doppler shift which results in frequency modulation. The Doppler shifts are positive in the case the motion is in direction toward the other end. But if either the

transmitter or the receiver is moving away from the other end, the Doppler shift is negative.

Speed of Surrounding Objects. If the motion of the objects in the environment between the transmitter and the receiver is at a greater rate than the motion of the transmitter or the receiver themselves, then this motion causes a time varying Doppler shift which dominates the effect of the small scale fading. Otherwise, the effect of the motion of the surrounding objects can be neglected.

The Transmission Bandwidth of the Signal. If the signal's bandwidth is smaller than the channel's bandwidth, distortion to the propagating signal will occur as discussed later in this chapter.

Before looking into the types of small scale fading, it's necessary to look at the meaning of coherence bandwidth, coherence time, maximum excess delay, rms delay spread and Doppler spread.

Coherence Bandwidth (B_C). The coherence bandwidth is a measure of the maximum frequency difference for which signals are still strongly correlated in amplitude. B_c is a measure of the range of frequencies that pass through the channel with approximately equal gain and linear phase.

Coherence Time (T_C). The coherence time is the period of time over which the response of the channel is invariant. Meaning that for this period of time, the channel behaves the same and that signals passing through the channel are subjected to the same attenuation and phase changes.

Maximum Excess Delay (T_m)(X dB). Maximum excess delay is the time between the arrival of the first multipath wave and the arrival of the multipath wave at which the received signal strength is X dB below the maximum.

RMS Delay Spread. Delay spread is the period of time between the arrival of the first multipath wave arriving at the receiver and the last one. The rms delay spread is given by Equation (9) where $\bar{\tau}$ is the mean excess delay which is calculated from the power delay profile.

$$\sigma_\tau = \sqrt{\overline{\tau^2} - (\bar{\tau})^2} \tag{9}$$

Doppler Spread (B_D). Doppler spread is defined in [1] as the range of frequencies over which the received Doppler spectrum is essentially non-zero. It's a measure of the spectral broadening caused by the time rate of change of the mobile radio channel. The Doppler spectrum is the received signal power spectrum, which is range of frequencies between $f_c - f_d$ and $f_c + f_d$, where f_c is the carrier frequency and f_d is the Doppler shift. The relation between the coherence time and the Doppler shift is given by

$$T_C \approx \frac{1}{f_m} \tag{10}$$

where f_m is the maximum Doppler shift. If the coherence time is defined as the time where the time coherence function is above 0.5 then

$$T_C \approx \frac{9}{16\pi f_m} \tag{11}$$

The most commonly used equation for the coherence time is the geometric mean of Equations (10) and (11) which yields to

$$T_C = \sqrt{\frac{9}{16\pi (f_m)^2}} = \frac{0.423}{f_m} \tag{12}$$

The relation between the coherence bandwidth and the rms delay spread depends on the structure of the multipath waves. An exact equation that relates them does not exist, although an approximation based on the frequency correlation function could be given by Equation (13) if the correlation function is above 0.9 and by Equation (14) if the correlation function is above 0.5.

$$B_c \approx \frac{1}{50\sigma_\tau} \tag{13}$$

$$B_c \approx \frac{0.276}{\sigma_\tau} \tag{14}$$

An exact relation that relates the coherence bandwidth with the maximum excess delay is given in [2] by

$$B_c = \frac{1}{T_m} \tag{15}$$

Equation (15) is not widely used because the maximum excess delay is not a good indicator of how the channel behaves since the same signal could propagate through two different channels and still have the same maximum excess delay.

Small scale fading is categorized into two main categories: fading due to the time spreading of the signal and fading due to time variance of the channel.

Fading Due to Time Spreading of the Signal

The transmitted signal is scattered into a large number of scattered signals arriving at the receiver at slightly different time causing time spreading of the received

power signal. Fading due to time spreading of the signal is categorized into two main categories:

Flat Fading. Flat fading means that the channel passes all the multipath waves with the same gain and with a linear phase. The time domain explanation for flat fading is that the propagating signal is subjected to flat fading when the signal's delay spread is smaller than the symbol time (Equation (16)). The frequency domain explanation of flat fading is that the propagating signal is subjected to flat fading when the signal's bandwidth (B_s) is smaller than the channel's coherence bandwidth (Equation (17)). Flat fading channels are also called narrowband channels. Flat fading causes fluctuations in the received signal of 20-30 dB which must be accounted for to assure successful communication. Flat fading does not cause intersymbol interference. Figure 2 shows the flat fading channel characteristics both in time and frequency domains. In the time domain, the channels induces a small delay as compared to the symbol time, which minimizes the chance that the received pulse will spread into the next pulse's time interval. In the frequency domain, the frequency components of the signal are subjected to the same gain.

$$\sigma_\tau << T_S \tag{16}$$

$$B_S << B_C \tag{17}$$

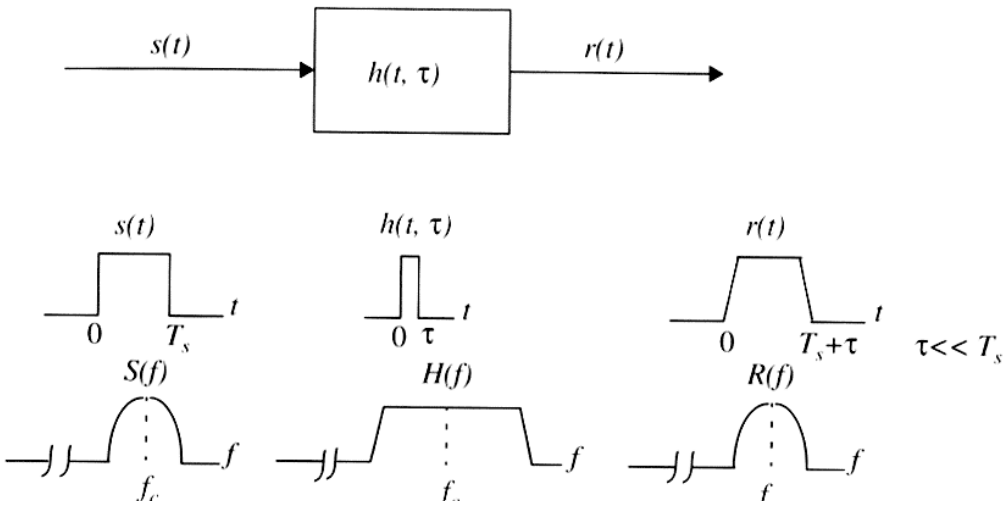


Figure 2: Flat fading channel characteristics [1]

Frequency Selective Fading. The time domain explanation for frequency selective fading is that the propagating signal is subjected to frequency selective fading when the signal's delay spread is larger than the symbol time (Equation (18)). The frequency domain explanation of frequency selective fading is that the propagating signal is subjected to frequency selective fading when the signal's bandwidth is larger than the channel's coherence bandwidth (Equation (19)). Frequency selective fading channels are also called wideband channels. Figure 3 shows the frequency selective fading channel characteristics in both the time and frequency domains. In the time domain, the channels induces a large delay as compared to the symbol time which causes intersymbol interference because the received pulse spread into the next pulse's time interval. In the frequency domain, the frequency components of the signal are subjected to different gains.

$$\sigma_\tau >> T_S \tag{18}$$

$$B_S >> B_C \tag{19}$$

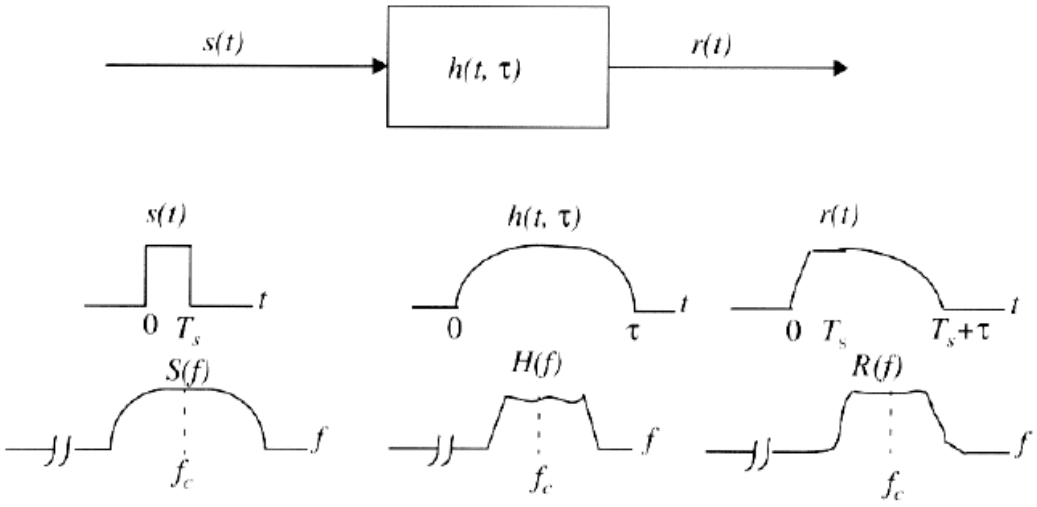


Figure 3: Frequency selective fading channel characteristics [1]

Fading Due to Time Variance of the Channel

As the objects in the channel environment move or change, the behavior of the channel changes causing the channel to be time variant. This type of fading is categorized into fast fading and slow fading depending on how fast the propagating signal changes compared to channel's rate of change.

Fast Fading. The time domain explanation of fast fading is that the channel is considered a fast fading channel if the channel's coherence time is smaller than the signal's symbol time (Equation (20)). The frequency domain explanation of fast fading is that the channel is considered a fast fading channel if the signal's bandwidth is smaller than the channel's Doppler spread (Equation 21).

$$T_C < T_s \tag{20}$$

$$B_S < B_D \tag{21}$$

Slow Fading. The time domain explanation of slow fading is that the channel is considered a slow fading channel if the channel's coherence time is longer than the signal's symbol time (Equation (22)). The frequency domain explanation of slow fading is that the channel is considered a slow fading channel if the signal's bandwidth is larger than the channel's Doppler spread (Equation (23)).

$$T_C >> T_s \tag{22}$$

$$B_S >> B_D \tag{23}$$

Flat or frequency selective fading is not correlated to fast or slow fading. Hence knowing if the signal is subjected to a flat or frequency selective fading does not tell if the channel is considered a fast or slow fading channel.

CHAPTER THREE

REVIEW OF THE LITERATURE AND RESEARCH

This chapter gives a detailed background about the literature which the research I conducted is based on. This chapter covers three different aspects. The first is a study of Ricean fading, which is the basis for the simulator implementation. The second is a study of over water path propagation, which is of particular interest for our current research. The last aspect is a study of different fade simulators designed and tested by researchers.

Ricean Fading

When a direct line of sight path between the transmitter and the receiver exists, the fading follows a Ricean distribution. Weaker multipath signals arrive at the receiver with different phase shifts and are superimposed on the line of sight dominant signal. When there is no line of sight signal, the fading follows a Rayleigh distribution. The parameter *K* is the main factor in Ricean fading that relates the line of sight signal power and the total power of the multipath signals The following is a discussion of the Ricean distribution based on a method developed by Greenstein [3] and [4] to estimate the value of *K* from actual received power versus time measurements. A Greenstein [3] model to estimate *K* is based on the season, antenna height, antenna beamwidth and distance is then presented.

Ricean Distribution

When the complex path gain consists of a constant component plus a zero-mean Gaussian fluctuating component, the resulting time varying envelope follows a Ricean distribution. The probability density function (PDF) of the Ricean distribution is given by Equation (24) as shown in [1] where A is the amplitude of the dominant signal, r is the received signal envelope voltage, σ^2 is the variance of the multipath signals and $I_0(*)$ is the modified Bessel function of the first kind and zero-order. The cumulative distribution function (CDF) of the Ricean distribution is given by Equation (25) where $Q_1(*,*)$ is the Marcum Q function. The Ricean PDF of the signal power is given by Equation (26) where P is the received voltage and $\bar{P}$ is the local mean power. The Ricean K parameter is the ratio between the power of the line of sight signal and the summation of power of the multipath signals (Equations (27), (28), and (29)). K is the defining parameter of the Ricean distribution and it is a measure of the severity of the fade.

$$p(r) = \begin{cases} \frac{r}{\sigma^2} e^{-\frac{(r^2+A^2)}{2\sigma^2}} I_0\left(\frac{Ar}{\sigma^2}\right) & \text{for } (A \geq 0, r \geq 0) \\ 0 & \text{for } (r < 0) \end{cases} \tag{24}$$

$$p(r < abscissa) = 1 - Q_1\left(\frac{A}{\sigma}, \frac{r}{\sigma}\right) \tag{25}$$

$$p(P) = \frac{1+K}{\bar{P}} \exp\left[-K - \frac{(1+K)P}{\bar{P}}\right] . I_0\left[\sqrt{\frac{4K(1+K)P}{\bar{P}}}\right] \tag{26}$$

$$K = \frac{\text{power of line of sight signal}}{\sum \text{power of multipath signals}} \tag{27}$$

$$K = \frac{A^2}{2\sigma^2} \tag{28}$$

$$K(dB) = 10\log\frac{A^2}{2\sigma^2} \quad \text{dB} \tag{29}$$

A function that plots the PDF and CDF of the Ricean distribution (Equations (24) and (25)) for different values of *K* was programmed into MATLAB. The MATLAB function code is shown in appendix A. The function's input parameters are *r*, *K* and σ. The PDF and CDF plots shown in Figure 4 and Figure 5 use σ = 2 and *r* as the vector [0:01:25]. When *K* = 0, the Ricean distribution degenerates to a Rayleigh distribution. Figure 4 and Figure 5 indicate that as the value of *K* increases, the Ricean distribution is distributed at a higher mean received signal envelope voltage. This confirms the idea shown in Equation (28) which indicates that as the value of *K* increases the peak amplitude of the LOS signal increases, which results in a higher received signal envelope voltage. Note also that as *K* increases, the PDF approaches a Gaussian distribution. Equation (26) was programmed into MATLAB to plot the Ricean PDF of the received power for different *K* values. The MATLAB code is shown in Appendix A. Figure 6 shows that as *K* increases the Ricean PDF of the received power gets narrower and sharper, which indicates that as *K* increases the fluctuation in the signal decreases. The reason for that is that as *K* increases the power of the constant line of sight signal increases as compared to the fluctuating multipath signals.

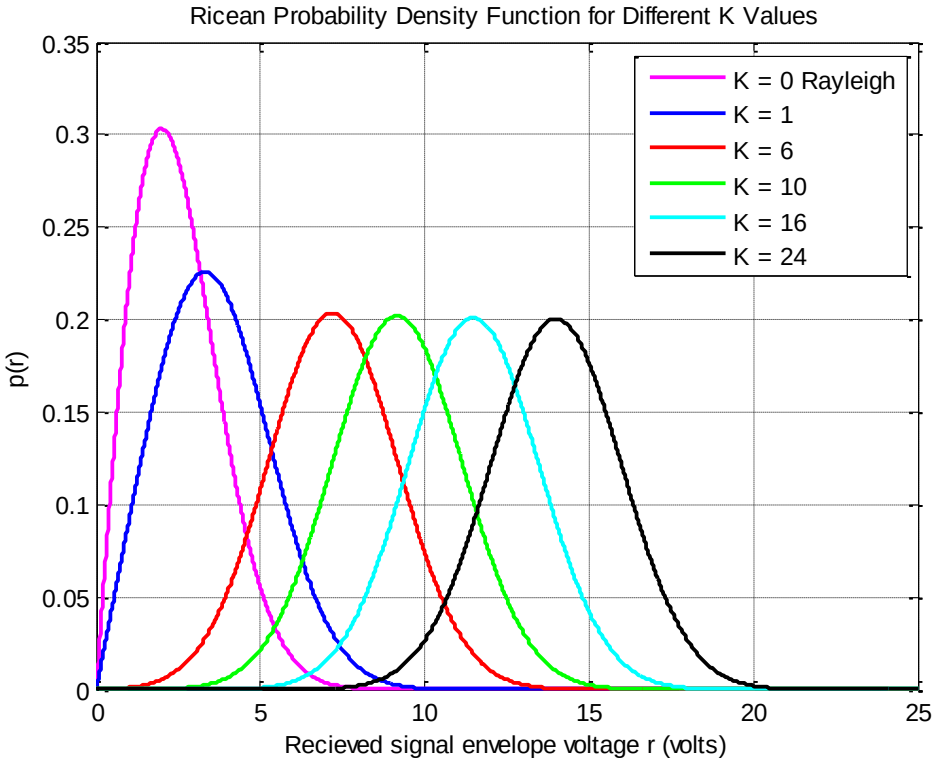


Figure 4: The Ricean probability density function (PDF) for different *K* values

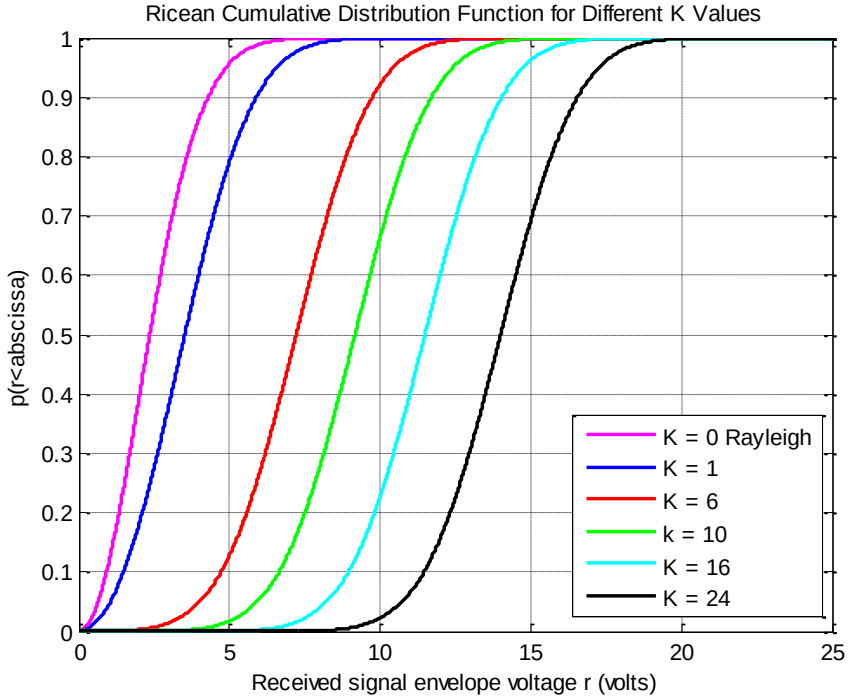


Figure 5: The Ricean cumulative distribution function (CDF) for different *K* values

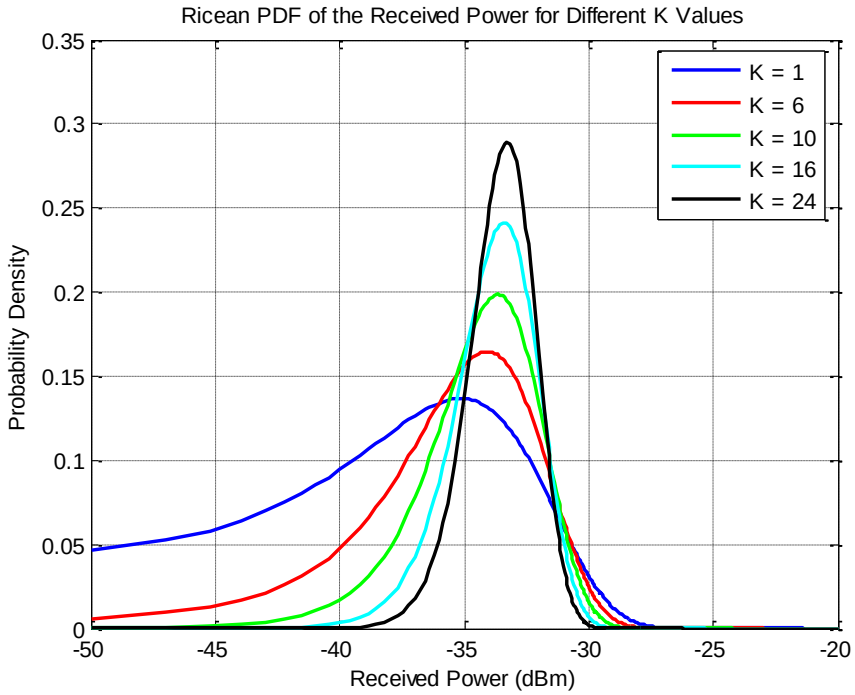


Figure 6: Ricean PDF of the received power for different *K* values

Estimation of the *K* Parameter

Different methods have been derived to the estimate the value of *K* for various channel environments. Some of these methods are based on the measured received power versus time data and some are based on the field strength data. A maximum likelihood estimator (MLE) using an expectation/maximization algorithm was derived by Marzetta [5]. This method was developed by Chen [6] to use samples from the fading envelope along with samples from the fading phase. The MLE method is complicated and time consuming. The collected data from a spectrum analyzer is power versus time data, making use of Chen method or the field strength method unfeasible.

An easy and rapid way (moment method estimation) developed by Greenstein [4] that is based on the received power versus time measurements is chosen to estimate the

value of *K* from the collected data. Greenstein and his group conducted extensive measurements in four different locations in New Jersey on Ricean fading. The theoretical moment based estimation method that he developed was compared to actual measurements and showed good agreement. Greenstein and his group also came up with a *K*-model that is discussed later.

Moment Method Estimation of the Ricean K-Factor. This method uses the first and second moments of the received power envelope [4]. The formulation of the method is as follows:

The complex path gain of a narrowband wireless channel is characterized by a frequency-flat response

$$g(t) = V + v(t) \tag{30}$$

where *V* is a complex constant (line of sight) and *v(t)* is a complex zero-mean random time variation (multipath signals). The corresponding power gain *G* is given by

$$G = |g(t)|^2 \tag{31}$$

The first moment or the time average of *G* is

$$G_a = |V|^2 + \overline{|v(t)|^2} + \overline{2\,\mathrm{Re}(\mathrm{V}v^*(t))} \tag{32}$$

Since *v(t)* is a zero-mean random process, the last term reduces to zero. Defining

$$\sigma^2 \equiv \overline{|v(t)|^2} \tag{33}$$

v(t) can be written as

$$G_a = |V|^2 + \sigma^2 \tag{34}$$

The second moment or the rms value of the fluctuation of G about Ga is

$$G_v = \left[\overline{(G - G_a)^2}\right]^{1/2} \tag{35}$$

$$G_v = \overline{|v(t)|^4} - \sigma^4 + \overline{(2\,\mathrm{Re}\,(\mathrm{V}v^*(t))^2} \tag{36}$$

Knowing that *v(t)* is a zero-mean complex Gaussian, G_a can be written as

$$G_v = \left[\sigma^4 + 2|V|^2\sigma^2\right]^{1/2} \tag{37}$$

Solving Equations (34) and (37) for $|V|^2$ and σ^2 yields

$$|V|^2 = \left[G_a^2 - G_v^2\right]^{1/2} \tag{38}$$

And

$$\sigma^2 = G_a - \left[G_a^2 - G_v^2\right]^{1/2} \tag{39}$$

K is defined by

$$K = \frac{|V|^2}{\sigma^2} \tag{40}$$

Where σ^2 is twice the RF power of the fluctuating term. That's why the factor of 2 in the denominator is not present.

To verify that the derived *K* estimation method is valid, Greenstein and his group compared the CDF of the extensive data they collected at four different locations in New Jersey for over paths with Ricean fading with a simulated ideal Ricean distribution (Figure 7). The simulated results used the *K* value estimated via their moments method. The comparison shows that their method works appropriately.

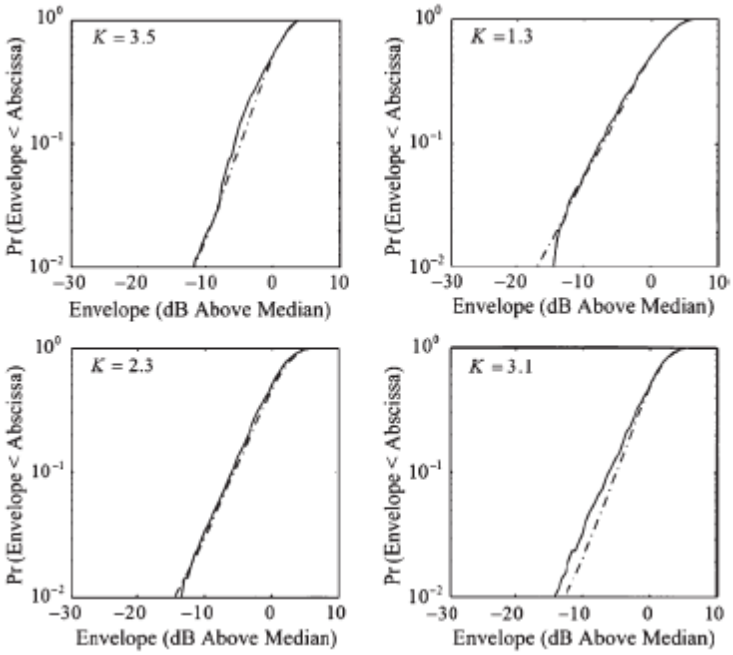


Figure 7: The CDF for the measured data (solid curves) and the Ricean simulated data (dashed curves) for different values of *K*. Greenstein [4].

K Model

To derive a model for *K*, Greenstein and his group used the extensive measurements they collected in four different locations (Table 2). The experiment parameters are antenna heights, antenna beamwidth, season and distance. the results collected from the first location are the richest in terms of parameter variation. The model represented is mainly based on the data collected from location1. The model was validated by checking these results with the results from the other three locations. For location 1, the measurment duration was 280 seconds and the data was collected every 0.4 second. Hence 750 snapshots of power versus frequency at 100 frequencies spaced by 100 kHz to cover 10 MHz bandwith were taken. The measurements were taken for 12

combinations of two seasons (summer and winter), two antenna heights (3 and 10 meters) and three beamwidths (17°, 30° and 65°).

The strength of the scattered signal increases in the summer over the winter due to the existence of leaves in the summer that cause more reflections and therefore the strength of the scattered signals increases. This means that the value of *K* in the winter is higher than the value of *K* in the summer. Greenstein and his group recorded a consistent increase in the value of *K* in winter by 3.5 to 4.5 dB with an average of 4.0 dB.

As the base station antenna height increases, the chance of having a direct line of sight will increase, the diffraction angle will decrease, the strength of the line of sight signal will increase, and the scattered components will vary less. This all means that the value of *K* increases as the antenna height increases. Greenstein and his group recorded a constant increase in the value of *K* of 2.0 to 3.0 dB with an average of 2.4 dB as the antenna height increased from 3m to 10m. If the power law relationship is assumed ($K \propto h^{\alpha}$), then $\alpha = 0.46$.

As the beamwidth of the receiving antenna decreases, less scattered signals are received. This means that the value of *K* increases as the beamwidth decreases. Greenstein and his group recorded a consistent increase in the value of *K* of 3.0 to 4.0 dB as the beamwidth changed from 65° to 17°. If the power law relationship is assumed ($K \propto b^{\beta}$) then $\beta = -0.62$. This law was then applied to the results of the 30° beamwidth data and the agreement was within 1dB.

As the distance between the transmitter and the receiver decreases, the strength of the line of sight signal increases. This means that as the distance decreases, the value of K increases. This follows the power law relationship $K \propto d^{\gamma}$.

Table 2: Summary of the four experiments [3]

Exp. No.	Measurement parameters	Measurement Duration	Terminal Antenna	Geography, Seasons	Uses for Reduced Data
1	9 MHz (90 frequencies) Base Antenna: Single-pol Panel h=15 and 30m	5-minute samples at each downlink location.	Single-pol; b = 17°, 30°, 65°; h = 3 m, 10 m	Three transmit sites in NJ. Tens of downlink locations for each site. Summer and Winter.	Devise first-order model for pdf of K at any time, frequency and location in NJ suburban environments.
2	CW (one frequency) Base Antenna: Single-pol Omni-directional h=25m	5-minute samples at each downlink location.	Single-pol; b = 45° and omni antenna; h = 3 m	One transmit site in WA, two in IL. Tens of downlink locations for each site. Summer only	Reinforce first-order model. Examine regional differences. Show effects of omni vs. directional antennas.
3	6 MHz (60 frequencies) Base Antenna: Single-pol Panel h=15 and 30m	Long runs (12 – 24 hrs) at each downlink location.	Dual-pol; b = 32°; h = 3 m	Two transmit sites in NJ. Seven and nine downlink sites. Summer only.	Reinforce first-order model. Examine variations with time and frequency.
4	CW (one frequency) Base Antenna: Single-pol Omni-directional h=25m	Long runs (2 – 12 hrs) at each downlink location.	Dual-pol; b = 32°; h = 3 m	Two transmit sites in IL. 14 and 21 downlink locations. Summer only.	Reinforce first-order model. Examine regional differences. Examine variations with time.

The derived formulation for K by Greenstein is:

$$K \cong F_s F_h F_b K_o d^{\gamma} \tag{41}$$

where the factors are

1- F_s is the seasonal factor

$$F_s = \begin{cases} 1.0 & \text{Summer(Leaves)} \\ 2.5 & \text{Winter(No Leaves)} \end{cases} \tag{42}$$

2- F_h is antenna height factor

$$F_h = \left(\frac{h}{3}\right)^{0.46} \tag{43}$$

3- F_b is the beamwidth factor

$$F_b = \left(\frac{b}{17}\right)^{-0.62} \tag{44}$$

K_o and γ are constants to be optimized via regression fitting.

Over-Water Path

If the communication between the transmitter and the receiver involves an over-water path, the signal might be subjected to sustained deep fades. The water surface functions as a mirror causing the reflected signals to be stronger than reflection from rough surfaces. The strong reflected signals when added with the line of sight signal could cause deeper fades than that caused by reflections from the earth's surface or objects in the channel environment. The water refractive index varies over time due to tidal effects, wave heights, ducting, temperature, pressure and humidity. An experiment that was performed by Norland [7] at the Norwegian coast, that studied the variation of the refractive index of the sea surface showed that the weather parameters cause the refractive index to change rapidly over time. The experiment sent an interferometric radar signal at 9.6 GHz over a 2.9 km over-water link to monitor unstable mountain blocks.

The measurements were taken for two different sites, each with two corner reflectors. Norland used the measured change in the received signal's phase ($\Delta\varphi$) (Equation (45)) to estimate the change in the refractivity index *N*.

$$\Delta\varphi = \frac{4\pi R}{\lambda}\Delta n \tag{45}$$

where

$$R = \frac{T_0 . C_0}{2\ n(\vec{r},t)}, \tag{46}$$

λ is the wavelength, T_0 is the travel time of the electromagnetic wave, C_0 is the velocity of light in vacuum and $n(\vec{r},t)$ is the spatial and temporal index of refraction.

$$N = (n-1)10^6 = 77.6\frac{p}{T} + 3.734*10^5\frac{e}{T^2} \tag{47}$$

where *P* is the pressure (mbar), *T* is the temperature (°*K*) and *e* is the partial water vapor pressure (mbar). The estimated refractivity index along with the recorded temperature, pressure, humidity and precipitation for a period of 45 days are shown in Figure 8. The four plots shown on the left of Figure 8 for the refractivity index *N* versus days (abscissa) show that *N* changes over days. The two main factors causing the change in *N* are the temperature and humidity (Figure 8 - right). More detailed measurements of *N* versus hours or minutes rather than days could have revealed a closer look at the effect of the weather factors as well as the variation of *N* over shorter periods of time.

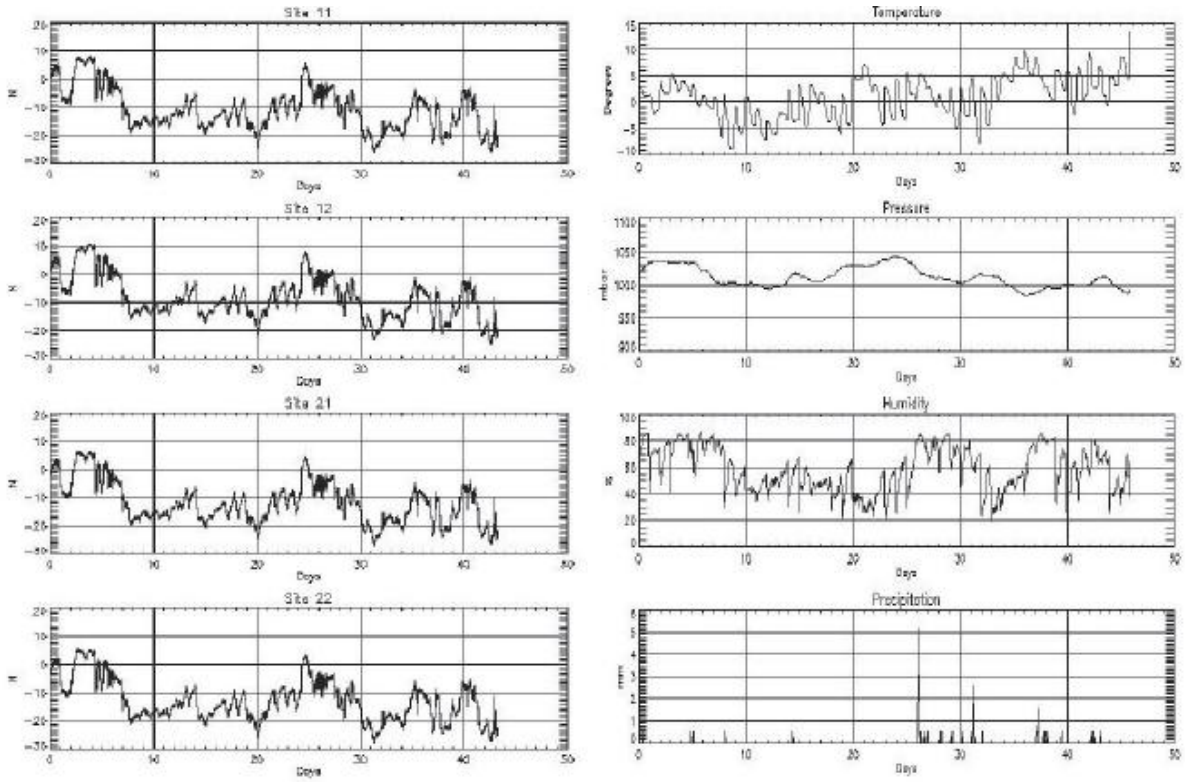


Figure 8: Relative variation of the refractivity index for the four sites and corresponding meteorological data [7]

Another reason for the increase of the refractive index which leads to sustained deep fades (SDF) was mentioned by Dockery [8] who examined advection of moist warm air over the cooler water. This creates an increase of the water temperature lapse rate and at the same time an increase of the saturation water vapor pressure with altitude. The resulted increase in the vapor pressure with altitude causes an increase of refractivity. Dockery and his group recorded extensive measurements for the signal fading statistics for two over water link with a line of sight in the mid-Atlantic coast of the United States for a one year period. A 4.7 GHz CW signal was sent from the transmitting tower at Parramore Island (PI), Virginia, to two receiving towers, lighthouse (LH) and lookout (LT) on Assateague Beach, Virginia, at distances of 44 and 39 km, respectively. The

weather conditions were recorded simultaneously along with the measured refractive profile. The propagation parameters for the PI-LH and PI-LT links are shown in Table 3.

Table 3: Propagation parameters for PI-LH and PI-LT links [8]

Parameter	Value
Antenna Transmit Power at PI, dBm	37.8
Gain of transmit antenna dB	25.0
Gain of receiver antenna dB	25.6
Nominal gaseous attenuation dB	0.26,0.23
Transmit antenna gain reduction in direction of PI-LY, dB	2.5
Estimated free space power	-50.7, -52.0
Diffraction losses for standard propagation, dB	2.0, 13.4
Standard power received, dBm	-52.7, -65.4
Multipath grazing angles, deg	0.0325, 0.0045
Reflection coefficient magnitudes	1.0, 1.0
Reflection coefficient phases, deg	180, 180

The fade is calculated relative to the free space power. The cumulative fade distribution for the two locations for the 1989-1990 year (Figure 9) shows that the received signal fluctuates over a 65 dB range. A SDF of 49 and 54 dB for the LH and LT links occurred less than 0.1% of the annual operating time.

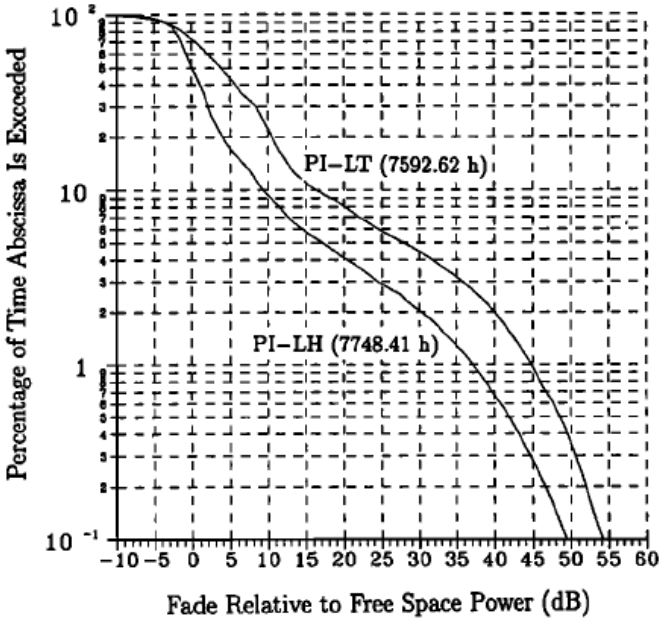


Figure 9: Cumulative fade distribution for the Pi-LH and PI-LT [8]

A monthly cumulative fade distribution for the two locations shown in Figure 10 (LH left and LT right) shows that the weaker fades occur during the fall and summer months and the deeper fades occur during the winter and spring months when the water is coldest.

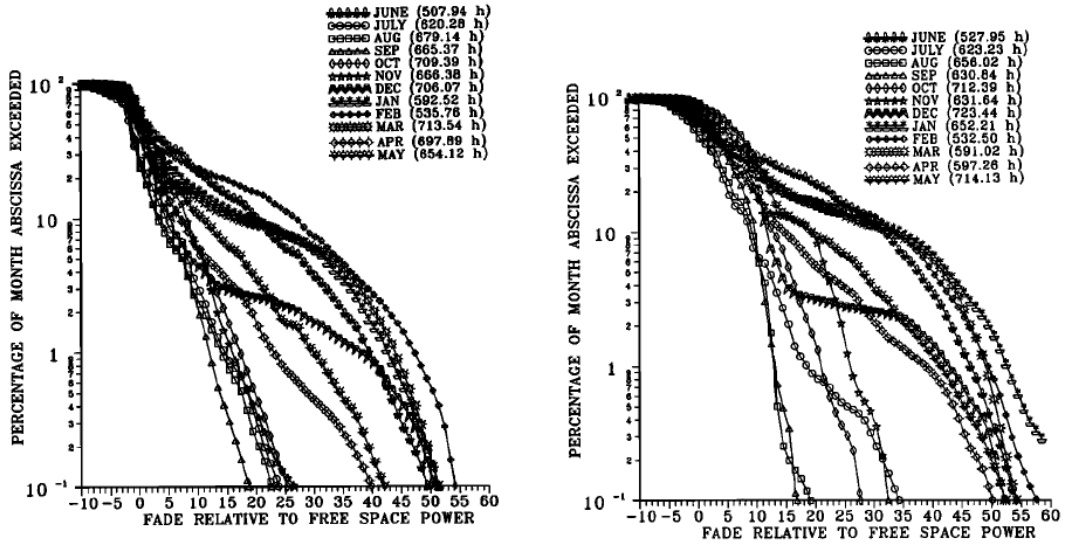


Figure 10: Monthly cumulative fade distribution [8]

Figure 11 shows the cumulative fade distribution for four contiguous 6-hour time slots corresponding to the period from June 1, 1989, to May 31, 1990 (LH right and LT left). It shows that there is no diurnal variation in the distribution of the fade. Analysis of synoptic meteorological conditions and wind speed data confirmed that sustained deep fades (example shown in Figure 12) occurred when warm, moist air over cold water existed causing a rise to positive lapse rates of refractivity with altitude.

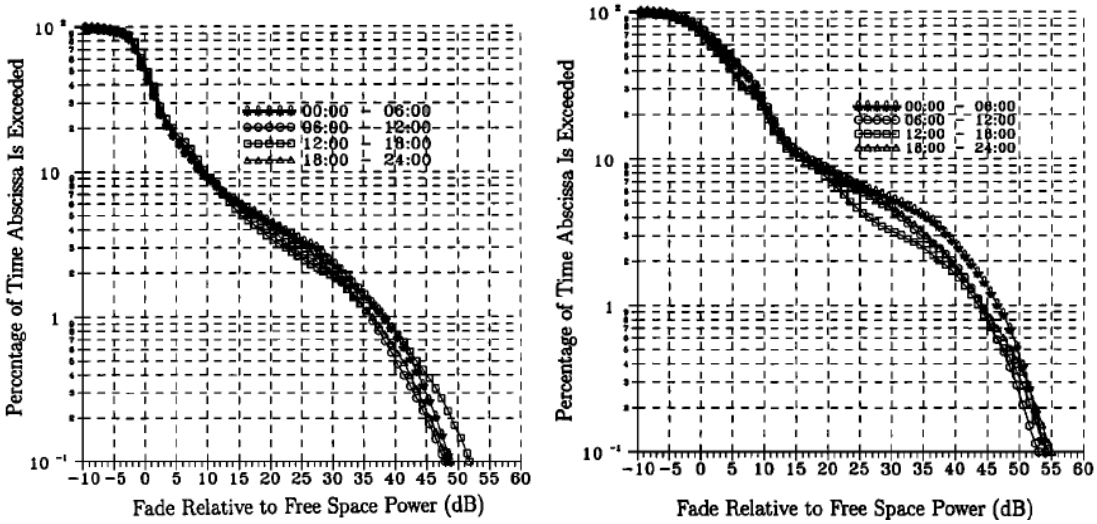


Figure 11: Cumulative fade distribution for four contiguous 6-hour time slots [8]

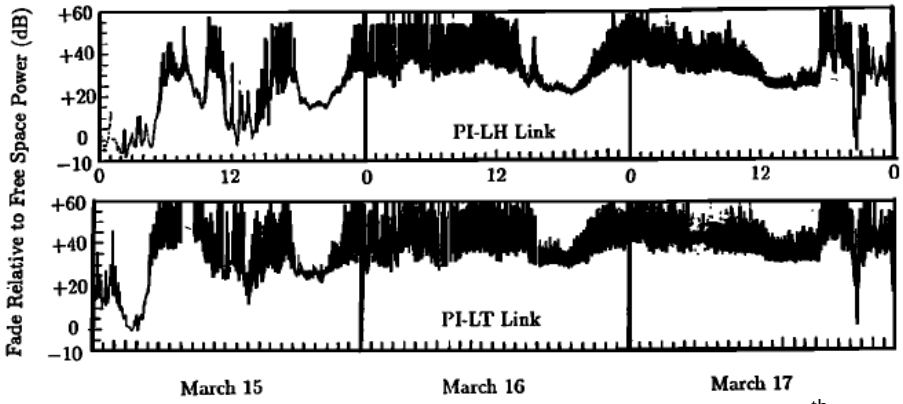


Figure 12: Recorded sustained deep fades during the period of March 15th to March 17th, 1990 [8]

A similar experiment to Dockery's was performed by Ong [9] through the year of December 1996 to November 1997. The link was 32.7 km over water with a line of sight path and operated at 8 GHz. The signal was transmitted from site T to site G. The antenna heights at the G location are 147 and 133 m. The antenna heights at the T locations were 76 and 61 m. The higher antennas were used as regular channels and the lower antennas were used as protection channels. The annual and worst month cumulative fade distributions for the T-G link are shown in Figure 13. The recorded deep fades fluctuated up to 40 dB relative to free space power.

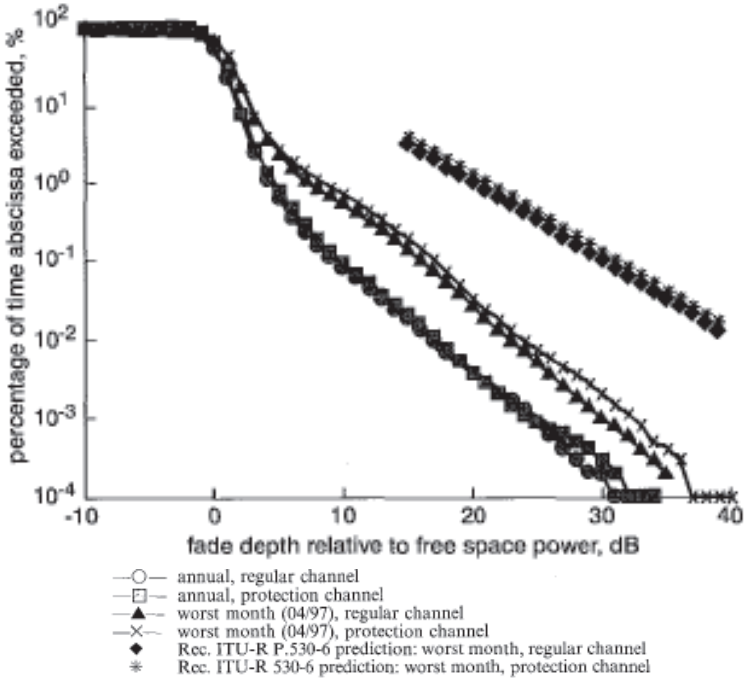


Figure 13: The annual and worst month cumulative fade distribution for the T-G link [9]

Ong stated in [10] that the deep fades were recorded at night during the inter-moon seasons when there was low wind activity and high humidity. The low wind activity was favorable for the formation of radio ducts. Also, under such conditions the sea surface was smooth which caused the reflected signals to be stronger.

Jin and his group [11] studied the correlation of the fading and weather conditions on an over-water digital microwave link. The link was 115 km and operated at 4 GHz. During December 1992, they recorded the fade statistics along with marine wave height, temperature, humidity, wind speed, wind direction and wind gust. They found that fading activities were related to low marine wave height, low wind speed, low wind gusts and low relative humidity. The reason is that the stated conditions cause the refractive index to increase which caused the reflected signals to be stronger.

Summary

The overwater path literature study showed that the over water link can suffer a SDF in the range of 40 dB. The water refractive index varies over time due to tidal effects, wave heights, ducting, temperature, pressure and humidity. Low wind activity causes the surface of the water to be smoother which results in stronger reflected signals. The advection of moist warm air over the cooler water causes an increase of water refractivity index which leads to higher SDF. The deeper fades occur during the winter and spring months when the water is coldest.

Previous Fade Simulators

We conducted a literature survey to study how researchers implemented fade simulators. A study of three different fade simulators is covered in this section. The first two are hardware-based and the third one is software-based. The three fade simulators studied show the developments made to the hardware and software associated with fade simulators from 1973 to 2000.

A Multipath Fading Simulator for Mobile Radio

This fade simulator was developed in 1973 by Gaston Arredondo and William Chriss [12]. They developed an analog hardware based fade simulator, and their goal was to simulate the Rayleigh distributed fast fading encountered in mobile radio. A schematic of their fade simulator design is shown in Figure 14. To simulate the Rayleigh fading that causes the received signal to fluctuate, two independent Gaussian low-pass noise sources with identical spectra are added to signal components in quadrature.

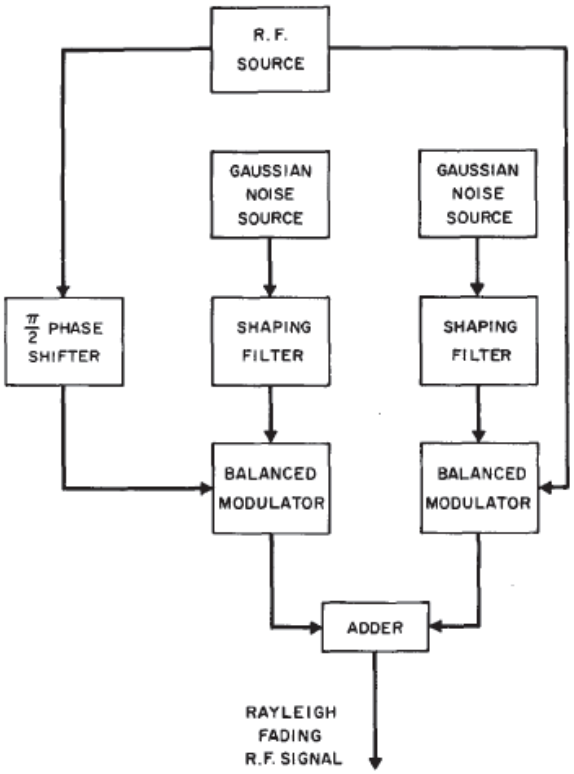


Figure 14: Analog Rayleigh fade simulator design (1973) [12]

The Gaussian noise source is a noisy Zener diode operating near cutoff shown in Figure 15. The output of the Zener diode is then passed through a shaping filter shown in Figure 16.

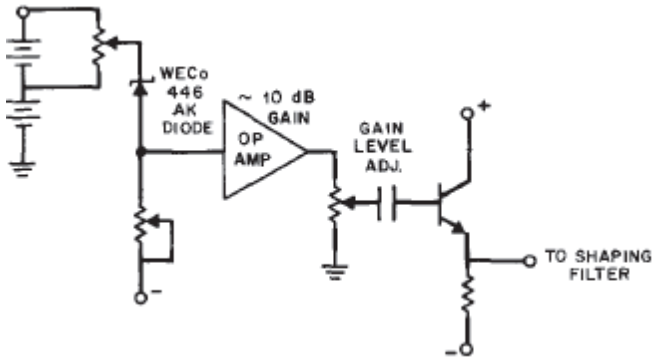


Figure 15: Low frequency Zener diode noise source [12]

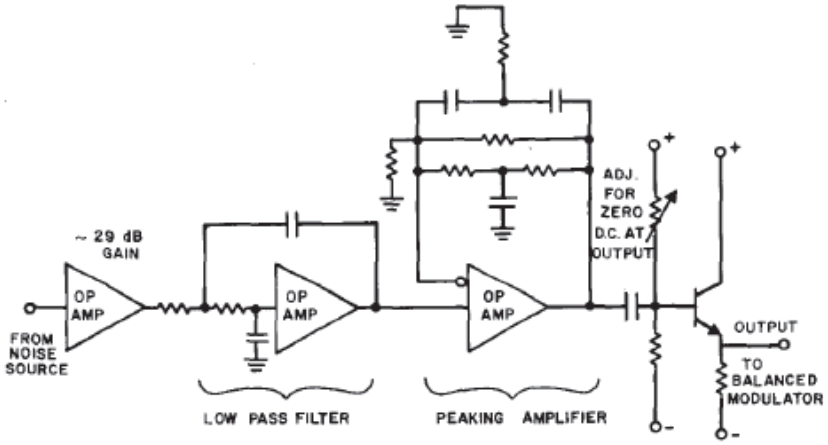


Figure 16: The noise source shaping filter [12]

The shaping filter is used to shape the spectrum of the noise source to match the theoretical spectral density of the complex envelope of the received signal, which is given by:

$$S(f)=\begin{cases}\dfrac{E^2}{2\pi\ f_D}\left[1-\left(\dfrac{f}{f_D}\right)\right]^{-1/2} & ,\ f \le f_D \\ 0 & f > f_D\end{cases} \tag{48}$$

where E is the rms value of the signal envelope and f_D is the Doppler shift given by:

$$f_D=\frac{V}{\lambda} \tag{49}$$

where V is the vehicle speed and λ is the carrier wavelength. Figure 17 shows the theoretical spectral density (left) and the simulated spectral density (right). A set of shaping filters were designed to be switched in to allow simulation of several vehicle speeds.

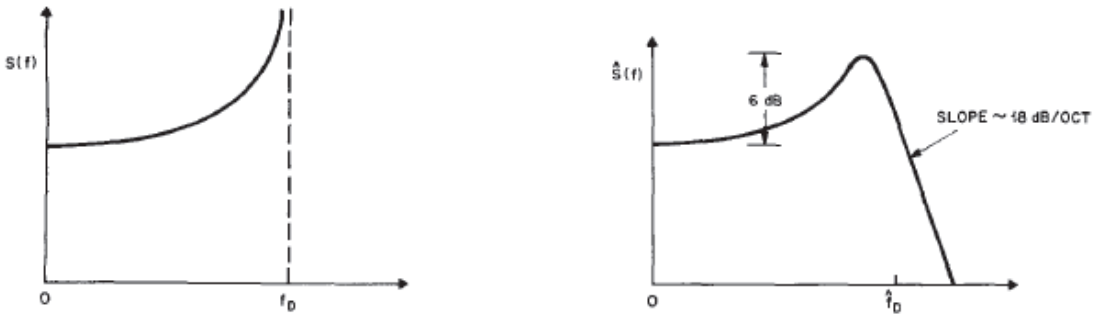


Figure 17: The theoretical and simulated spectral densities [12]

The statistics of the simulator output were evaluated to compare their fit to the Rayleigh distribution shown in Figure 18, where the Rayleigh variant R and the measured variant $\hat{R}$ are plotted. The simulated results fit closely with the theoretical predictions. To validate the fading envelope characteristics, the level cross rate, which is the rate at which the envelope makes crossing of a specified level, was plotted. Figure 19 shows that the simulated level cross rate (points) matches the theoretical prediction (solid).

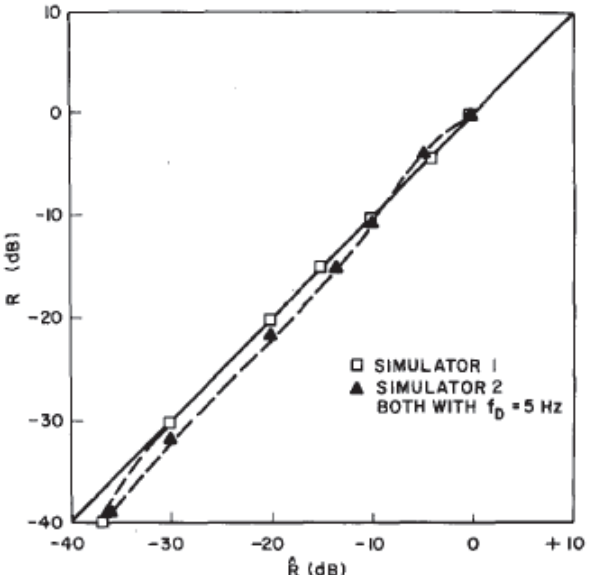


Figure 18: Closeness of fit to Rayleigh statistics [12]

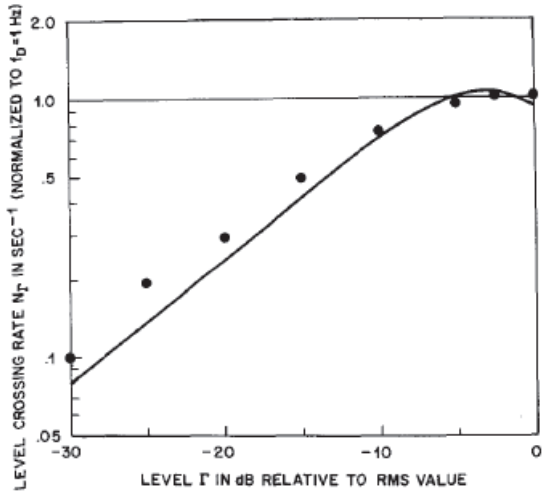


Figure 19: Level cross rate (Solid: theoretical) and (points: measured) [12]

A Simple Digital Fading Simulator for Mobile Radio

This fade simulator, which was developed in 1990 by Eduardo Casas and Cyril Leung [13], is the digital version of the analog fade simulator presented above. The basic theory that was implemented is Rayleigh fading for mobile radio, which is the same idea used in the analog version. The RF input signal is divided into in-phase and quadrature

phase components and then modulated with independent low-pass Gaussian random signals, as shown in Figure 20.

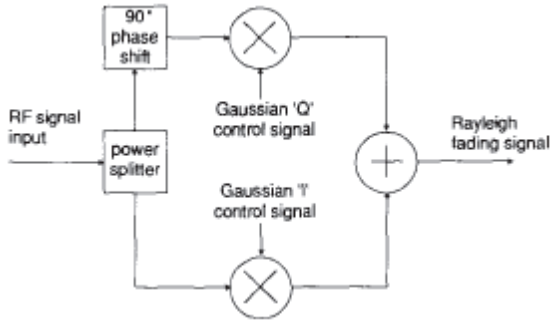


Figure 20: Rayleigh digital fade simulator [13]

The analog noise sources and analog filters were used to control the signal in the previous analog fade simulator. Casas and Leung used an Intel 8088 microprocessor instead of analog components to generate the two control signals. As shown in Figure 21, the control signals are the weighted sums of the outputs from nine sinusoidal oscillators. The amplitudes and frequencies of the sinusoids are chosen to give uncorrelated signals with power spectra approximating *S(f).* An 8088 assembly language program was used to generate the control signals. Doppler rates between 2 and 126 Hz in steps of two could be simulated.

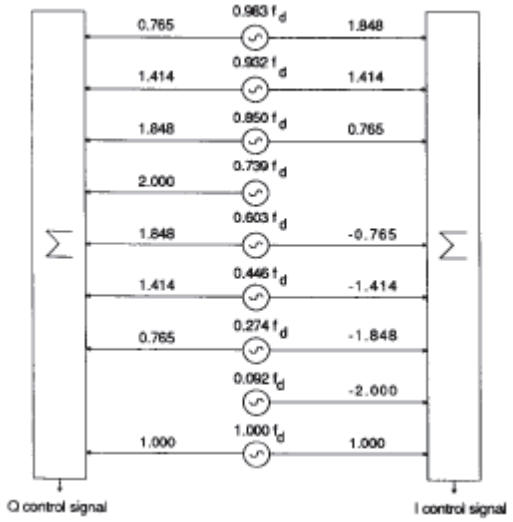


Figure 21: Generation of the control signals [13]

To validate the results, Casas and Leung plotted a comparison of a measured CDF for three different Doppler rates of 2, 20 and 120 Hz with an ideal Rayleigh CDF as shown in Figure 22. The measured CDFs are Rayleigh to within about 1 dB in the range from -40 to +10 dB. They also plotted a comparison of the measured level crossing rate for the three Doppler rates with the corresponding theoretical ones as shown in Figure 23. The measured results for their Rayleigh digital fade simulator showed good agreement with the theoretical results.

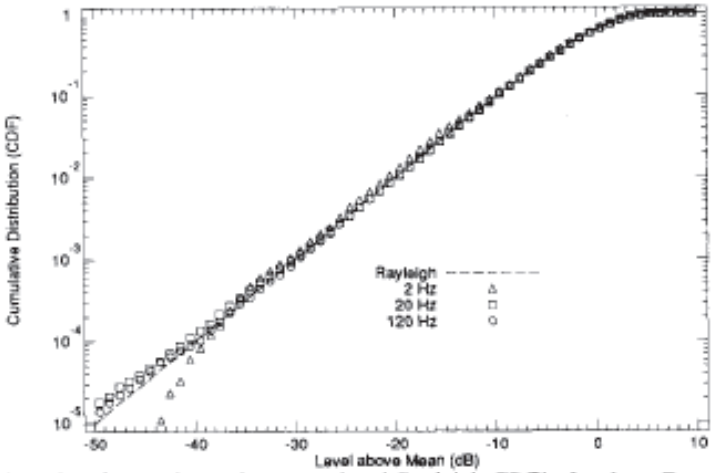


Figure 22: Measured and Rayleigh CDF for three Doppler rates [13]

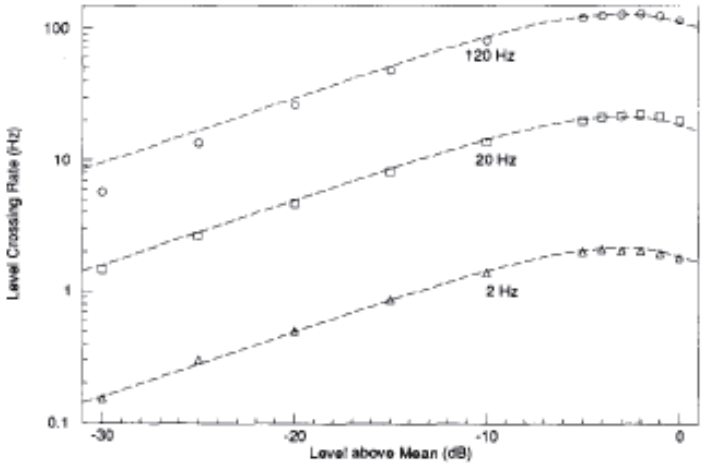


Figure 23: Comparison of theoretical and measured level crossing rate [13]

Ricean Fading within a Packet Simulator

In [14] Punnoose and Nikitin introduced work that has been done to model the effect of small scale fading (Rayleigh and Ricean) within a network simulator. The simulator they developed models Rayleigh fading as a special case of Ricean fading when $K = 0$. A dataset containing the in-phase and quadrature phase components, which are then combined with the appropriate K value, is pre-computed. The pre-computed dataset is considered a lookup table that could be used to model a wide range of parameters such as the time averaged power, the maximum Doppler frequency and the Ricean K factor. The steps used to generate the dataset are:

1- Specify the number of points, maximum Doppler frequency and the frequency spacing of the points.

2- Generate complex Gaussian random numbers for each of the frequency components.

3- Evaluate the fading spectrum at the frequency pointed selected above.

4- Multiply the frequency domain component with the fading spectrum.

5- Perform an IFFT on this data to yield a time series data.

6- Repeat steps 2, 3, 4 and 5 to generate a second dataset. The two generated datasets are the in-phase and quadrature phase datasets.

To validate the results of their packet fade simulator, Punnoose and Nikitin plotted the CDF and the level crossing rate for their results and the theoretical predictions, as shown in Figure 24 and Figure 25, respectively. The packet simulator results for the CDF and the level crossing rate for both the Rayleigh fading and the Ricean fading for K = 6 dB showed good agreement with the theoretical predictions.

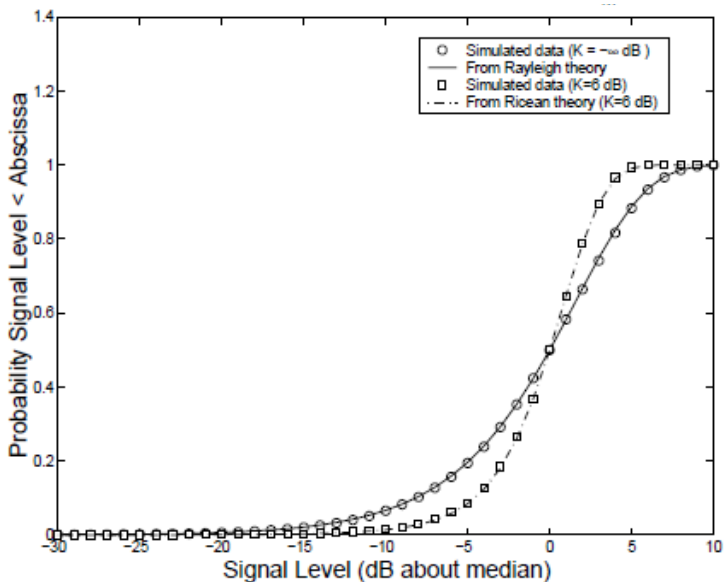


Figure 24: CDF for simulated Ricean and Rayleigh fading using the packet simulator [14]

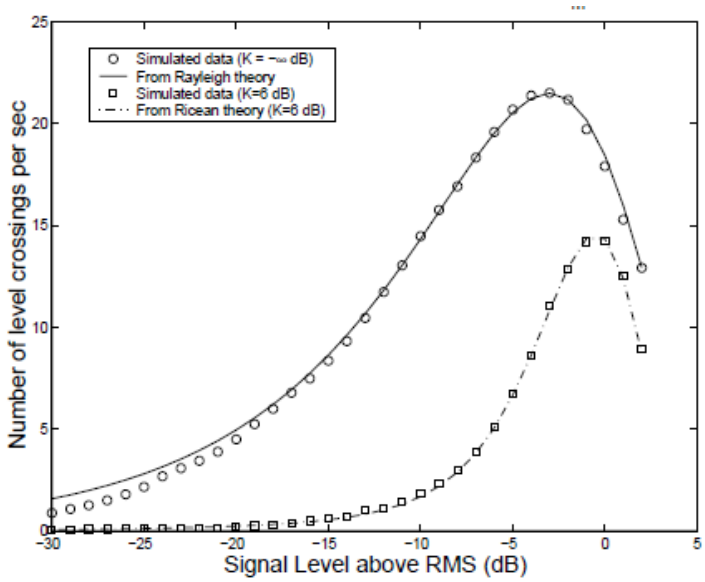


Figure 25: Level crossing rate for simulated Ricean and Rayleigh fading using the packet simulator [14]

The first two fade simulators presented above are hardware based, and were designed to simulate a Rayleigh fading for mobile radio where there is no line of sight signal. Instead of building analog hardware circuits to be switched in to simulate different Doppler frequencies, the digital version of the Rayleigh fade simulator introduced an easy and quick way to simulate a range of Doppler shifts. The third fade simulator presented above is a software-based and can simulate both Rayleigh and Ricean fading. An extensive literature review showed that a hardware based Ricean fade simulator for static radios has not been developed prior to the work reported here.

CHAPTER FOUR

EXPERIMENTAL PROCEDURE

The Ricean fade simulator developed here is hardware-based and is controlled with a MATLAB program. Figure 26 shows a block diagram of the Ricean fade simulator. The key component in the fade simulator is the beamformer board that was designed and built for antenna beamforming and directional of arrival estimation research conducted by Dr. Wolff's group. The beamformer board was designed to operate at 5.8 GHz and has eight channels, each with digital attenuators and phase shifters.

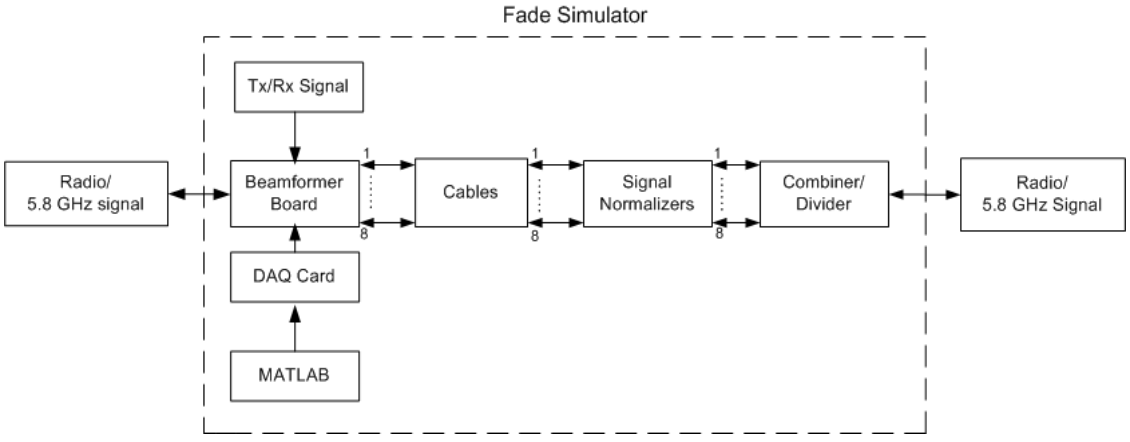


Figure 26: Block diagram of the Ricean fade simulator

How does the Ricean Fade Simulator Work?

The Ricean distribution theory assumes a line of sight signal and an infinite number of random time variant multipath signals superimposed on the line of sight signal. It is not practical to simulate an infinite number of signals. The approximated simulation for the theory is to use a line of sight signal and seven multipath signals since

the beamformer board has only eight channels. This means that one of the channels is assigned to be the fixed line of sight signal while the seven other channels are the varying multipath signals. A MATLAB program that generates a serial bit stream of random attenuations and phase shifts weights according to the Ricean distribution is first run. The serial bit stream is sent from MATLAB to the Field-Programmable Gate Array (FPGA) on the beamformer board through the Data Acquisition (DAQ) card. The FPGA then sends the serial bit stream to the 6-bit attenuators and 6-bit phase shifters. The fade simulator can work in either the transmitting or the receiving mode. The Tx/Rx signal assigns the beamformer to work in either the transmitting mode or the receiving mode by triggering the switches on each channel.

Fade Simulator in the Transmitting Mode

The 5.8 GHz input signal is divided into eight signals using the beamformer's microstrip line power divider/combiner. At the same time, the serial bit stream is sent to the attenuators and phase shifters so the eight signals can be attenuated and phase shifted to emulate a time variable Ricean channel. The eight signals are passed through eight cables with different lengths to be delayed. The eight signals are then passed through signal normalizers. The signal normalizers are required to normalize the signals to a specific level since the signals are subjected to different attenuations as they propagate through the cables. The eight signals are then combined using an eight channel signal combiner/divider.

Fade Simulator in the Receiving mode

The 5.8 GHz signal is first divided into eight signals using the eight channel signal combiner/divider. The eight signals are then normalized, delayed and passed through the beamformer board to be attenuated and phase shifted. The eight signals are then combined using the beamformer board microstrip line combiner/divider.

The development of the fade simulator proceeding two steps: developing the software to run the fade simulator according to the Ricean distribution and constructing the hardware.

Software Development

The software associated with the Ricean fade simulator controls the digital attenuators and phase shifters on the beamformer board by sending a serial bit stream from the computer running MATLAB to the FPGA, which controls the attenuators and phase shifters. The Ricean fade simulator software development requires three main steps:

1. Generating the serial bit stream of the random attenuations and phase shifts according to the Ricean fading ditstribution.
2. Using the DAQ card to send the serial bit stream to the FPGA on the beamformer board
3. Programming the FPGA to receive the serial bit stream and then sending it to the attenuators and the phase shifters.

Figure 27 shows a detailed block diagram of the three main steps of the software development of the Ricean fade simulator.

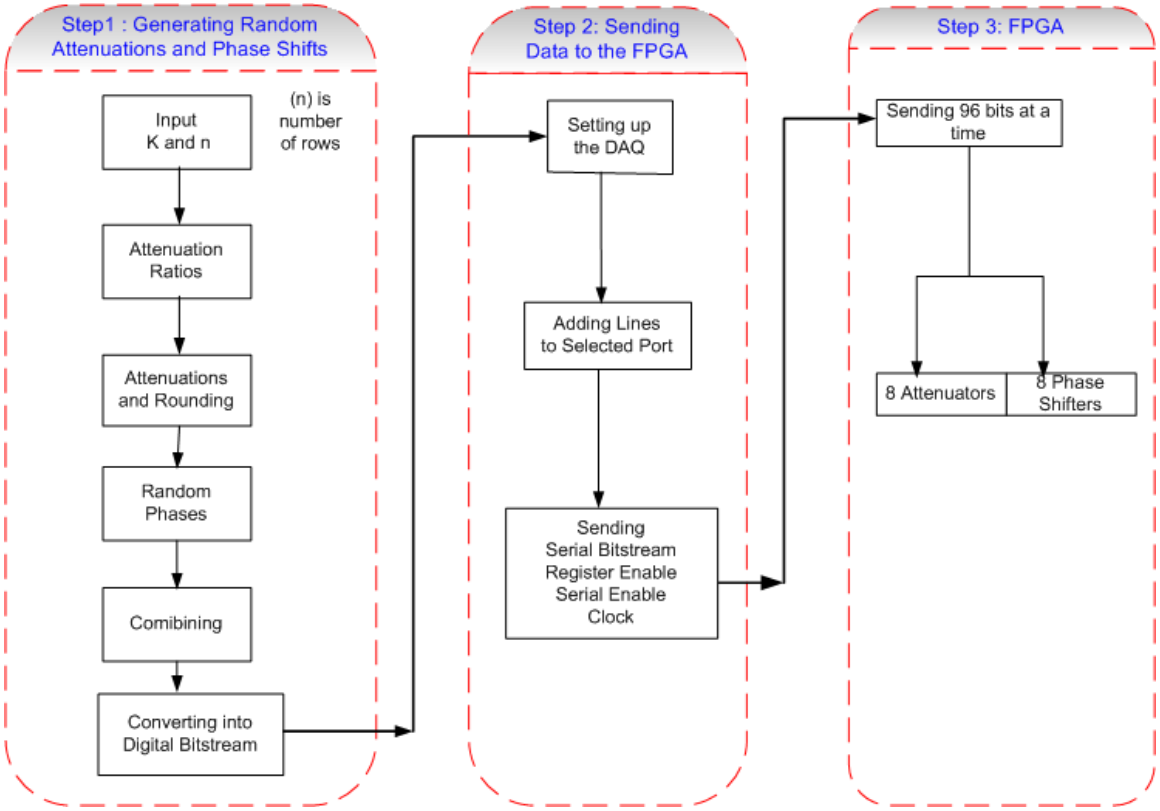


Figure 27: Software flow of the Ricean fade simulator

Since the beamformer board has only 8 channels; one for the line of sight signal and seven others for the multipath signals, Equation (27) is first rewritten as

$$K = \frac{P_{LOS}}{\sum_{i=1}^{7} P_i} \tag{50}$$

Where P_{LOS} is the received line of sight signal power and P_i is the average received multipath signal power. Equation (50) is then rewritten as

$$\frac{P_{LOS}}{P_i} = 7K \tag{51}$$

Converting Equation (51) into dB

$$10\log\left(\frac{P_{LOS}}{P_i}\right) = 10\log(7K) \tag{52}$$

Defining

$$(P_{LOS})_{dB} = 10\log(P_{LOS}) \tag{53}$$

$$(P_i)_{dB} = 10\log(P_i) \tag{54}$$

$$(7K)_{dB} = 10\log(7K) \tag{55}$$

Equation (52) is then rewritten as

$$(P_{LOS})_{dB} - (P_i)_{dB} = (7K)_{dB} \tag{56}$$

The received signal in (dB) is the difference of the transmitted signal in (dB) and the paths loss in (dB). Therefore,

$$(P_{LOS})_{dB} = (P_t - A_{LOS})_{dB} \tag{57}$$

and

$$(P_i)_{dB} = (P_t - A_i) \tag{58}$$

where P_t is the transmitted signal power, A_{LOS} is the attenuation in the line of sight path and A_i is the attenuation in one of the multipath paths. Equation (56) is rewritten as

$$(P_t - A_{LOS})_{dB} - (P_t - A_i) = (7K)_{dB} \tag{59}$$

which leads to

$$A_i - A_{LOS} = (7K)_{dB} \tag{60}$$

The attenuators have a maximum attenuation of 31.5 dB which leads to

$$x_i \,.31.5 - x_{LOS} \,.31.5 = (7K)_{dB} \tag{61}$$

$$x_i - x_{LOS} = \frac{(7K)_{dB}}{31.5} \tag{62}$$

where x_i and x_{LOS} are the attenuation weights in the multipath paths and the line of sight path, respectively. Both x_i and x_{LOS} are between 0 and 1. Defining

$$ratio = \frac{(7K)_{dB}}{31.5} \tag{63}$$

gives

$$x_i = x_{LOS} + ratio \tag{64}$$

Assume the line of sight path has a fixed x_{LOS} = 0.15 to have some attenuation in the line of sight path

$$x_i = 0.15 + ratio \tag{65}$$

The idea is to generate seven x_i weights that fluctuate between

$$(0.15 + ratio) \pm constant \tag{66}$$

The K value is chosen to be in the range of

$$0 < K \leq 24 \tag{67}$$

K = 24 indicates that the line of sight signal is very dominant as compared to the summation of the multipath signals. Simulating Rayleigh fading (K = 0) using this procedure is not possible because substituting K = 0 in Equation (63) gives (*ratio* = infinity). Substituting K = 24 in Equation (63) leads to (*ratio* = 0.7).

$$0 \leq (0.15 + 0.7) \pm constant \leq 1 \tag{68}$$

The *constant* is then equal to 0.15, which gives a range of multipath attenuations of 10 dB ($\pm$ 0.15 $\rightarrow$ 0.3 * 31.5 $\approx$ 10 dB). Note that 10 dB is one tenth of the power.

Step 1: Generating the Serial Bit Stream

The key parameter in the Ricean distribution is the *K* factor, which is the first input for the MATLAB code. The number of rows (*n*) (each row consists of eight different attenuations and eight different phase shifts) is the second input for the MATLAB code. The first step of the MATLAB code is to use *K* to calculate (*ratio*) and then generate seven random weights for *n* number of rows that fluctuate in the range of $(x_{LOS} + ratio) \pm 0.15 < 1$. x_{LOS} is assigned a fixed value of 0.15. The MATLAB function (rand) is used to generate the random weights for the *n* rows. This generates a matrix of a size (*n* * 8). The eight weights in each row are then multiplied by 31.5 to generate the actual attenuation values. Since the attenuators attenuate the signals in steps of 0.5 dB, the generated attenuations are rounded to the nearest zero or 0.5 dB value. So if the attenuation value is between X.0 to X.24, it is rounded to X.0 and if the attenuation value is between X.25 to X.5, it's rounded to X.5 because X.25 is used as the threshold. Also if the attenuation value is between X.51 to X.74, it is rounded to X.5 and if the attenuation value is between X.75 to X.99, it's rounded to X.0 + 1 because X.75 is used as the margin. The attenuators are controlled by a 6 bit binary word corresponding to attenuators ranging from 0 to 31.5 dB. The 6-bit binary number gives a decimal range of 0 to 63. The attenuation range is 31.5, which is equal to 63 doubled. The attenuators have the left bit as the most significant bit (MSB), which means that the attenuation numbers to be converted to serial bit stream are

$$\text{attenuation numbers} = 63 - 2 * (\text{attenuation values}) \tag{69}$$

The attenuation numbers matrix still has the size of (n * 8). The 6-bit phase shifters have a 360° coverage range in steps of 5.6° (5.6° * 63 ≈ 360°). The MATLAB (rand) function is used to generate seven random phase shift numbers for n number of rows between 0 and 63. The line of sight channel is assigned a fixed phase shift value for all the (n) rows. The phase shift number matrix size is (n * 8). The attenuation and phase shift numbers are then combined such that for each row the 16 attenuation and phase shift numbers are arranged in the order of attenuation then phase shift then attenuation then phase shift,… etc. the combined matrix has a size of (n * 16). Each attenuation and phase shift pair is assigned to a channel on the beamformer board. The first pair is assigned to channel one which is designated as the line of sight signal. The MATLAB (dec2binvec) function is then used to convert the attenuation and phase shift numbers to 6-bits binary numbers. The bits are then flipped left to right to have the MSB the left bit. The binary matrix has a size of (n * 96) where 96 is the 16 numbers multiplied by 6, yielding the number of bits. The matrix is then converted to the size of ((n*96) * 1) which is the serial bit stream. The MATLAB code that was written to generate the serial bit stream is shown in Appendix A.

Step 2: DAQ card

A DAQ card is used to send the serial bit stream from MATLAB to the FPGA. The first step is to install the DAQ card driver and the MATLAB data DAQ tool box on the computer running MATLAB. The serial bit stream is sent from MATLAB to the DAQ tool box, to the DAQ card driver, and then to the DAQ card. The first step in this MATLAB code is to set up the DAQ card. The set up process includes selecting whether

the DAQ card used is digital or an analog, if it is used as an input or an output, and the name of the DAQ card. The set up process also includes defining the DAQ vendor, e.g. Advantech, Agilent Technologies or National Instrument. The National Instruments NI USB-6501 DAQ card was used. After installing the DAQ card driver and connecting the DAQ to the computer, the DAQ card driver interface enables checking on the devices that are connected to the computer and the name of each one. The name of the DAQ card can be changed if desired. The second step is to select which ports and which lines in these ports are to be used to output the data. The beamformer FPGA requires four inputs: the clock bit stream, the serial data bit stream, the serial enable bit stream and the register enable bit stream. The clock signal is used to synchronize the FPGA, enabling it to read the serial data bit stream. The serial data bit stream is the transferred data from MATLAB to the FPGA that contains the controlling bits for the attenuators and the phase shifters. The serial enable bit stream informs the FPGA to save the serial data bit stream in a register as long as the serial enable bit is high. The register enable bit stream informs the FPGA to send the serial data bit stream saved in the register to the attenuators and the phase shifters when it becomes high. Four lines at one of the DAQ card's port are needed. The last step is to start sending the four bit streams simultaneously by sending a continuous clock signal, the generated serial data bit stream, the serial enable bit stream which is high during the transfer of each 96 bits then becomes low for one clock cycle then back high again and the register enable serial bit stream which is low during the transfer of each 96 bits and then becomes high for one clock cycle and back low again.

The MATLAB code that was written to transfer the data from MATLAB to the FPGA is shown in Appendix A.

Step 3: FPGA

A VHDL (VHSIC hardware description language, VHSIC: very high speed integrated circuit) program was written and installed on the beamformer FPGA. The VHDL program contains all the information about which pins of the FPGA are connected to the attenuators and the phase shifters. The VHDL program enables the FPGA to save the serial data bit stream in a register as long as the serial enable bit stream is high and the register enable bit stream is low. When the FPGA receives 96 bits, the serial enable bit goes low and the register enable bit goes high which triggers the FPGA to send the saved 96 bits from the register to the attenuators and the phase shifters. This is all synchronized with the input clock signal sent from MATLAB. Figure 28 shows the timing diagram for the four serial bit streams.

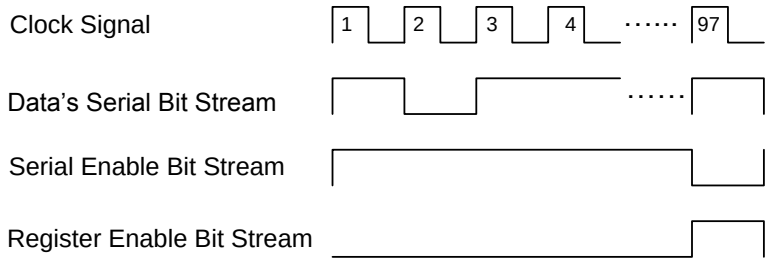


Figure 28: Timing diagram for the four serial bit streams

Hardware

The Ricean fade simulator's hardware components are: the beamformer board, cables, signal normalizers and signal combiner/divider.

Beamformer Board

Figure 29 shows the schematic of the beamformer board. The beamformer board has two inputs (Tx/Rx signal and serial bit streams), one input/output (5.8 GHz signal) and eight identical input/output channels (5.8 GHz signal). The board's input/output is the 5.8 GHz signal that is either divided into eight signals if the board is in the transmitting mode or combined from the eight channels if the board is in the receiving mode, using an eight way microstrip line power divider/combiner. The two inputs are the serial data bit stream that is sent from the FPGA to the phase shifters and attenuators and the Tx/Rx signal that instructs the beamformer to work in either the transmitting or the receiving mode. The Tx/Rx signal is either driven from a radio if the input signal is coming from a radio or from a DC power supply if the input signal is coming from the network analyzer or other CW RF signal source. The beamformer board is in the transmitting mode if the Tx/Rx signal is low and in the receiving mode if the Tx/Rx signal is high. Each of the eight channels has an attenuator, a phase shifter, two switches, a power amplifier, a low-noise amplifier and a band-pass filter. When the board is in the transmitting mode, which means that the two switches on each channel are triggered to enable the signal to propagate towards the cables, the 5.8 GHz input signal is divided to eight signals. The signal propagating through each channel is first attenuated with the appropriate attenuation using the 6-bit attenuator then phase shifted with the appropriate phase shifter using a 6-bit phase shifter and finally amplified. The attenuators and phase shifters used on the beamformer board are HMC425LP3 and MAPCGM0002, respectively. The datasheets for the attenuators and phase shifters are shown in Appendix

B. The HMC425LP3 is a 6-bit attenuator that operates at 2.4 to 8 GHz and has an attenuation range between 0 and 31.5 dB in steps of 0.5 dB. The MAPCGM0002 is a 6-bit phase shifter that operates at 2.4 to 8 GHz and has a phase shifting range of 360° in steps of 5.6°. When the board is in the receiving mode which means that the two switches on each channel are triggered to enable the signal to propagate towards the attenuators and phase shifters, the eight 5.8 GHz input signals are band-pass filtered, amplified, phase shifted, attenuated and then combined using the beamformer's eight channel microstrip line power combiner.

Cables

The cables are used as delay lines to simulate the actual delay that occurs to the multipath signals as they propagate from a transmitter to a receiver. The cables used are LMR-400, low attenuation at 5.8 GHz. The LMR-400 cable datasheet is shown in Appendix B. From the datasheet, for a 100 ft cable, the typical attenuation at 5.8 GHz is 10.8 dB. The propagation velocity of the LMR-400 cables is 85 % of the free space propagation velocity. This gives a time delay of 1.2 ns/ft. There is no delay cable associated with the line of sight path. The seven multipath signal paths have seven different length cables to simulate different delays. Table 4 shows the cable lengths and the associated time delays.

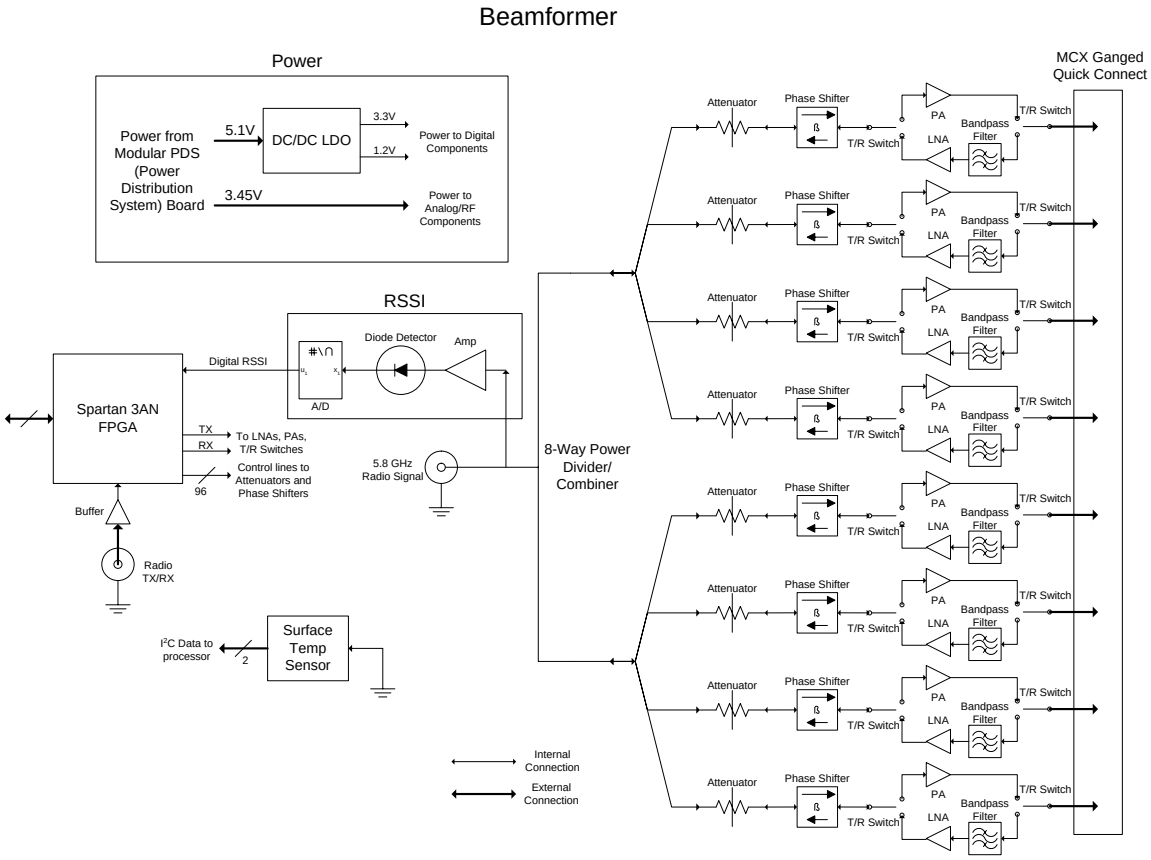


Figure 29: The beamformer board schematic

Table 4: Cable lengths and associated time delays

Cable lengths	Time Delay (ns)
100 ft	120
90 ft	108
75 ft	90
60 ft	72
45 ft	54
30 ft	36
15 ft	18

After the cables were cut to the appropriate lengths, SMA connectors were attached to the two ends of each cable. The cables were tested using the network analyzer to measure the return loss (S11 and S22 in dB) to assure that the SMA connectors were connected appropriately and the insertion loss (S21 in dB) to measure the exact attenuation in each

cable. Table 5 shows the measured characteristics of the cables at 5.8 GHz. The highest return loss measured is (-12.5) dB which means that associated reflection coefficient is 0.237. A (0.237) reflection coefficient means that ((1 - $(0.237)^2$) * 100 = 94.37 %) of the incident power is transmitted while 5.6% of the incident power is reflected. The measured insertion for the 100 ft cable at 5.8 GHz is 11.35 dB, which is within a 1-dB range of the value given in the datasheet of 10.8 dB. Table 5 also shows the measured voltage standing wave ratio (VSWR) for the cables.

Table 5: The measured characteristics of the cables at 5.8 GHz

Cables	S11 (dB)	S22 (dB)	S21 (dB)	VSWR
100 ft	-24.5	-12.5	-11.35	1:1.6
90 ft	-16.76	-19.9	-9.76	1:1.2
75 ft	-19	-21	-8.7	1:1.2
60 ft	-17.38	-18.87	-6.65	1:1.1
45 ft	-23	-21.7	-5.65	1:1.2
30ft	-13	-13	-3.7	1:1.6
15 ft	-23.79	-24.6	-1.8	1:1.1

Signal Normalizers

The signals are attenuated by the attenuators on the beamformer board to the appropriate levels derived from the Ricean distribution. Signals normalizers are required to normalize the signal after they are attenuated in the cables to a specific level so the cables function as lossless delay lines. Two levels were chosen to normalize the signals; either the high level of the line of sight path that has no cables and hence no attenuation, or the low level of the 100 ft cable. The high level would require uni-directional or bi-directional amplifiers to amplify the signals, and the latter would require attenuators to attenuate the signals on all but the longest path channel. The use of amplifiers enables the

fade simulator to have a higher dynamic range than the use of attenuators, but at the same time it's a more expensive solution, and also requires a DC power supply to drive the amplifiers. It is necessary to ensure that the amplifiers operate in the linear region below the 1-dB compression point.

Uni-directional Amplifier. When the 5.8 GHz input signal is drived from the function generator or the network analyzer, the beamformer board operated in either the transmitting or the receiving mode. Uni-directional amplifiers can be used to normalize the signals. The first version of the fade simulator used a ZX60-6013E+. The datasheet for this amplifier is shown in Appendix B. The ZX60-6013E+ operates between 20 MHz to 6 GHz with a noise figure of 3.3 dB. The typical gain at 5.8 GHz is 12 dB. The measured gain is 11.55 dB. For different cable lengths, different attenuations are associated which means the different gains are required to normalize the signals. The ZX60-6013E+ amplifiers are used along with attenuators to get different gains. Table 6 shows the values of the required attenuators that were used with the ZX60-6013E+ amplifiers to achieve different gains. The values of the attenuators are the amplifier gain subtracted from the cable losses rounded to the nearest integer.

Table 6: Required attenuators with the uni-directional amplifier.

Cable lengths	Cable loss (dB)	Amplifier gain (dB)	Attenuator (dB)
100 ft	11.35	11.55	0
90 ft	9.76	11.55	2
75 ft	8.7	11.55	3
60 ft	6.65	11.55	5
45 ft	5.65	11.55	6
30 ft	3.7	11.55	8
15 ft	1.8	11.55	10

Bi-directional Amplifier. When the 5.8 GHz input signal is derived from a radio transceiver, bi-directional amplifiers that enable the signals to be amplified in the two directions are required to normalize the signals. A complete design of a bi-directional amplifier that operates between 5 to 6 GHz and has a typical gain of 12 dB was completed. Figure 30 shows a schematic of the designed bi-directional amplifier.

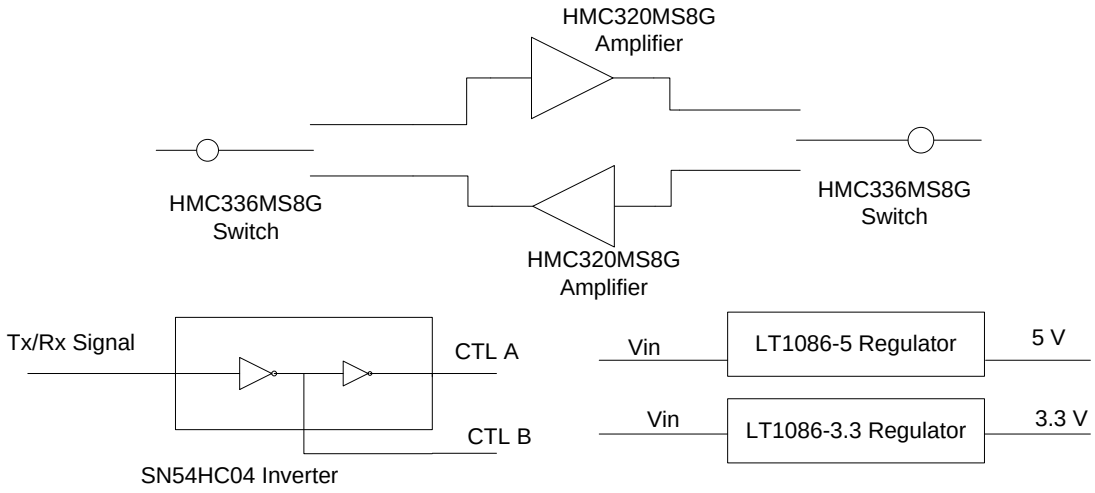


Figure 30: Bi-directional amplifier schematic

The components of the bi-directional amplifier are: two HMC336MS8G switches, two HMC320MS8G amplifiers, a LT1086-5 voltage regulator, a LT1086-3.3 voltage regulator and an SN54HC04 inverter. The datasheets for all the components are shown in Appendix B. The HMC336MS8G switch is an eight lead MSOP8G surface mount package that operates from DC up to 6 GHz with a typical insertion loss of 1.6 dB and a typical isolation of 42 dB. The switch operates with a control voltage of 5V. The switch is triggered using two control signals, CTL A and CTL B. The HMC320MS8G amplifier is an eight lead MSOP ground base surface mount package that operates between 5 to 6 GHz. It is a low noise amplifier that has a noise figure of 2.5 dB, a typical gain of 12 dB at 3V bias and a typical return loss of 10 dB. The SN54HC04 inverter has 6 inverters and

operates using a bias voltage between 2V to 6V. The LT1086 is a series of voltage regulator that can provide up to 1.5A. It is a 3-pin T0-220 package.

The Tx/Rx signal is first converted into CTL A and CTL B signals using the SN54HC04. The CTL A and CTL B signals trigger the switches to operate either in the transmitting mode or the receiving mode. The 5.8 GHz signal is passed through the first switch, amplified and then passed through the second switch. The input voltage (V_{in}) is inputted to the two voltage regulators LT1086-5 and LT1086-3.3. The LT1086-5 generates 5V that drives the switches and the inverter. The LT1086-3.3 generates a 3.3V that drives the amplifiers. AppCAD was used to calculate the width of the RF signal transmission line. The inputs for the AppCAD tool are the operating frequency and the substance prosperities information. At 5.8 GHz and with RF-4 material, the calculated width of the transmission line is 23 mils. Figure 31 shows a detailed schematic of the bi-directional amplifier.

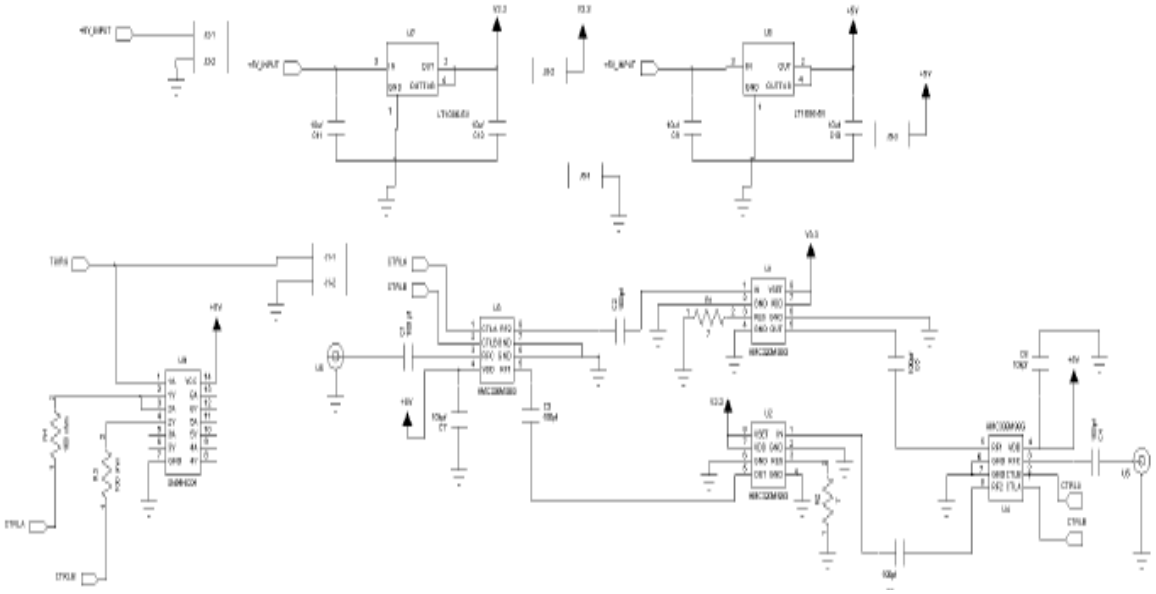


Figure 31: Detailed schematic of the bi-directional amplifier.

Figure 32 shows the layout of the bi-directional amplifier. The layout has two layers with dimensions of (2.6 * 2.1) inches.

Attenuators (only). The second solution of normalizing the signals is attenuating all the signals to the level of the 100 ft cable. The attenuators are bi-directional and do not require power to operate making them an easier and quicker solution. The main drawback is that adding attenuators reduces the fade simulator's dynamic range. Table 7 shows the attenuators required to normalize the signals in the seven multipath paths and the line of sight path. The current implementation of the fade simulator uses attenuators to normalize the signals.

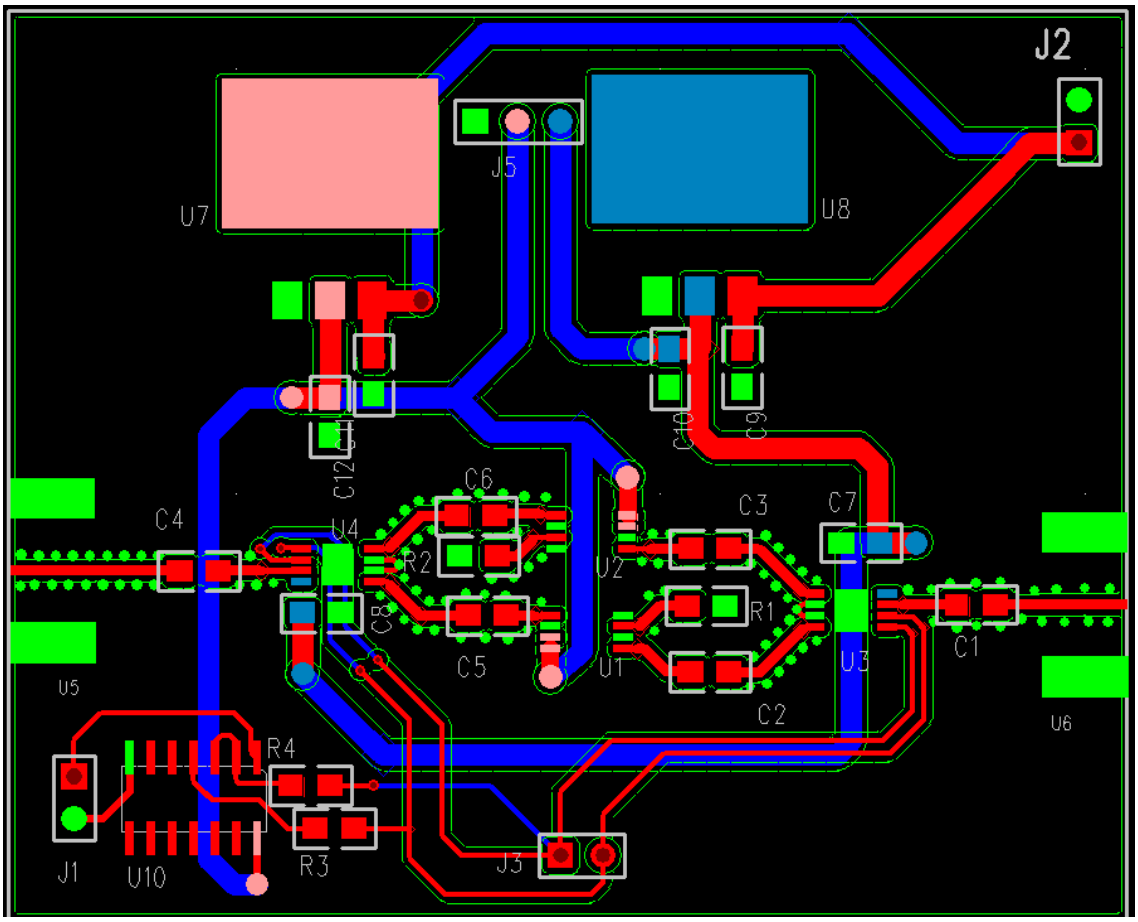


Figure 32: Bi-directional amplifier layout

Table 7: Attenuators required to normalize the signals (attenuators only)

Cable lengths	Cable loss (dB)	Attenuator (dB)
100 ft	11.35	0
90 ft	9.76	2
75 ft	8.7	3
60 ft	6.65	5
45 ft	5.65	6
30 ft	3.7	8
15 ft	1.8	10
LOS Path	0	12

Signal Combiner/Divider

An eight channel combiner/divider is used to combine the eight signals in case the fade simulator is used in the transmitting mode or to divide the input signal to eight signals in case the fade simulator is used in the receiving mode. The combiner/divider used is the ZBP8PD-6.4 that operates between 5.6 to 6.4 GHz. It has a typical insertion loss of 9.8 dB and a typical isolation of 23 dB. The datasheet of the combiner/divider is shown in Appendix B.

Experimental Setup

The first experiment was to use the fade simulator in the transmitting mode or the receiving mode and collect power versus time measurements to study the fading characteristics of the fade simulator and to test their agreement with the Ricean fading distribution. The second experiment was to use the fade simulator in the bi-directional mode to evaluate the performance of broadband radios. Chapter Six details the second experiment.

To perform the first experiment, a 5.8 GHz input signal generated by either a vector network analyzer or a synthesized signal generator was used. The lab HP 870D vector network analyzer was typically used as the signal source. The network analyzer 870 D was set in the CW mode at 5.8 GHz. The output of the fade simulator was then fed to the lab's Advantest R3273 spectrum analyzer. The center frequency of the spectrum analyzer was set to 5.8 GHz. The spectrum analyzer was used in the zero-span mode to collect power versus time measurements.

Coherence Time Measurement

It was important to measure the simulated channel coherence time, which is the time it takes the fade simulator to move from one set of fade values to the next one. The coherence time is determined by the time it takes MATLAB to send the 96 attenuators and phase shifter controlling bits. The time it takes the FPGA to send the 96 bits to the beamformer board is in the microseconds range which could be neglected. To measure the coherence time, the MATLAB program was run to generate the serial bit streams and then send them to the DAQ card. The register enable line of the DAQ card was observed on the oscilloscope. The time between two consecutive peaks is the measured coherence time. The measured coherence time depends on the speed of the processor of the computer used as well as the speed of the DAQ card. The measured coherence time is between 1.4 to 1.5 seconds. The fade simulator measured coherence time is in agreement with the measured coherence time of actual Ricean channel measurements. Greenstein [3] stated that relatively shallow fluctuation occurred on a time scale of 1 or 2 seconds. In [15], Feick and his group collected field results for fading characteristics at 3.5 GHz and

5.8 GHz for line of sight and non-line of sight urban fixed wireless channels. They stated that the measured coherence time was 0.1 to 2 seconds.

Troubleshooting Codes

To measure the attenuation range of each channel of the beamformer board, a MATLAB code (shown in Appendix A) was written. The MATLAB code sends consecutive combinations of a series of 96 ones followed by a series of 96 zeros to measure the range between the minimum and maximum fluctuations of the signals. The 5.8 GHz signal with 0 dBm power was inputted from the network analyzer and the output of each beamformer board's channel was observed on the spectrum analyzer. The Tx/Rx input was grounded to enable the fade simulator to work in the transmitting mode.

Table 8 shows the measured maximum and minimum powers as well as the attenuation range for each channel of the beamformer board. It shows that the beamformer board's channels are not consistent since they output different maximum and minimum powers and that the attenuation range is different from one channel to another. The minimum output powers for the channel ranged between -28.55 dBm for channel 2 to 36.66 dBm for channel 6, which is a range of 8.11 dB. The maximum output powers ranged between -2.79 dBm for channel 2 to -7.65 dBm for channel 4 which is a range of 4.86 dB. The typical attenuator range is 31.5 dB. The measured attenuator range was between 24.67 dB for channel 5 to 31.96 dBm for channel 6, which is a range of 7.29 dB. The differences in the maximum and minimum channel powers are due to the variability in the channel amplifiers gains plus the variability in the attenuators and phase shifters insertion and return losses.

Other MATLAB codes were written to troubleshoot the beamformer board. Some of these codes send all ones or all zeros or a combination of ones and zeros. The inputted series of bits is then tested at the attenuators and phase shifter control bits input using the logic analyzer to ensure that what is being sent is what the attenuators and phase shifters are actually reading and in the right order.

Table 8: Beamformer board's attenuators range

Channels	Minimum Power (dBm)	Maximum Power (dBm)	Range (dB)
Channel 1	-35.7	-5.37	30.33
Channel 2	-28.52	-2.79	25.73
Channel 3	-32.91	-7.48	25.43
Channel 4	-35.64	-7.65	27.99
Channel 5	-28.42	-3.66	24.67
Channel 6	-36.66	-4.7	31.96
Channel 7	-31	-5.26	25.74
Channel 8	-34.38	-3.42	30.96

MATLAB GUI

A MATLAB graphical user interface (GUI) was designed to control the fade simulator. The GUI is shown in Figure 33. The GUI was designed to run the fade simulator using the Greenstein *K*-model explained in Chapter Three or using a specific *K* value. The inputs for the Greenstein model are the season factor, the antenna height, the antenna beamwidth, the separation between the transmitter and the receiver, K_o and γ. The GUI first runs the *K*-model and presents the estimated *K* value. The *K* value should not exceed 24 as explained before. If the calculated *K* value exceeded 24, the GUI will display an error message informing the user that the value of *K* cannot exceed 24. If the *K* value is less than 24, the (n) value is then inputted and the Run (the fade simulator

Greenstein model) button is then pushed to run the fade simulator. The second option is to input a specific *K* value and the value of (n) and push the Run (the fade simulator with specific *K* value) button to run the fade simulator. Running the fade simulator include generating the serial bit stream and sending it to the FPGA. The GUI then presents Time1 which is the time MATLAB took to generate the serial bit stream, Total time which is the time MATLAB took to send the serial bit stream and Coherence Time, which is the Total time divided by (n). The GUI has hints for the user for easier usage.

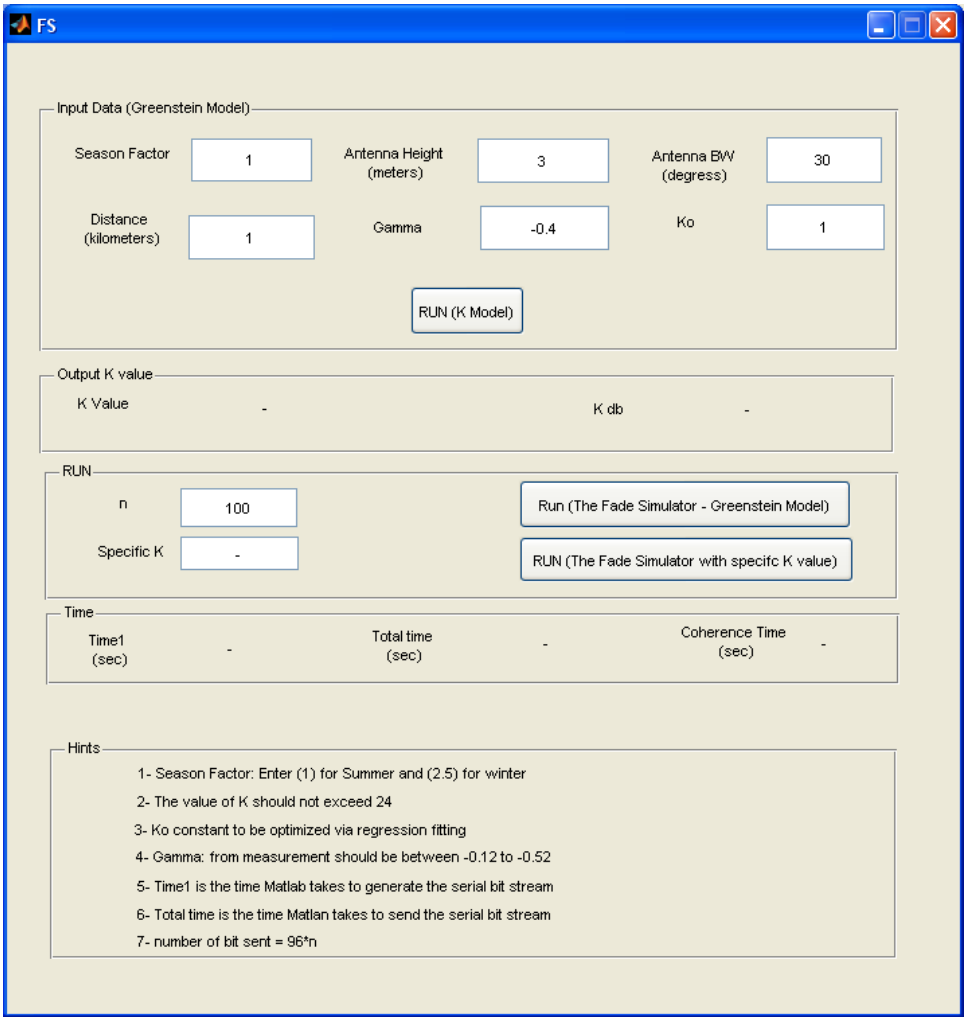


Figure 33: MATLAB GUI to control the fade simulator

Collecting the Results

A Labview interface program that was written by Raymond Weber, a Montana State University graduate student, was used to collect the power versus time measurements from the spectrum analyzer and save it as an Excel sheet. The Labview program parameters are the center frequency, the span time and the name of the Excel sheet and where to save it. The Labview interface program details are shown in Appendix C.

CHAPTER FIVE

RESULTS AND DISCUSSION

The first experiment was performed to study the characteristics of the fade simulator. The fade simulator was set in the transmitting mode by grounding the Tx/Rx signal. The 5.8 GHz signal was provided from the network analyzer. The MATLAB program was run for n = 100 which covers a time span of 150 seconds, since the measured coherence time is 1.4 to 1.5 seconds. The MATLAB program was run for five K values that cover the range between 0 and 24. The five K values are: K = 1, K = 6, K = 10, K =16 and K = 24. For each K value, the power versus time measurements were collected five times each with different random attenuations and phase shifts. Figure 34 shows the collected power vs. time measurements for five different K values for Run 1. The power vs. time plots for the other four runs are shown in Appendix D. Table 9 shows the measured minimum, maximum and range of fluctuation of the received power for the five K values for the five runs.

Both Figure 34 and Table 9 show that the fluctuation occurred to the received signal decreases as K increases. The fluctuation for K = 1 ranged between 28 to 30 dB for four runs and a deeper fade of 40 dB for run3. The fluctuation for K = 6 ranged between 13 to 20 dB. The fluctuation for K = 10 ranged between 10 to 15 dB. The fluctuation for K = 16 ranged 8 to 10 dB. The fluctuation for K = 24 was equal to 7 dB. The results matched the expectations since K is a measure of the severity of fade which means that the higher the K value the stronger the constant line of sight the signal as compared to the

summation of the fluctuating multipath signals and the less fluctuating the received signal power becomes.

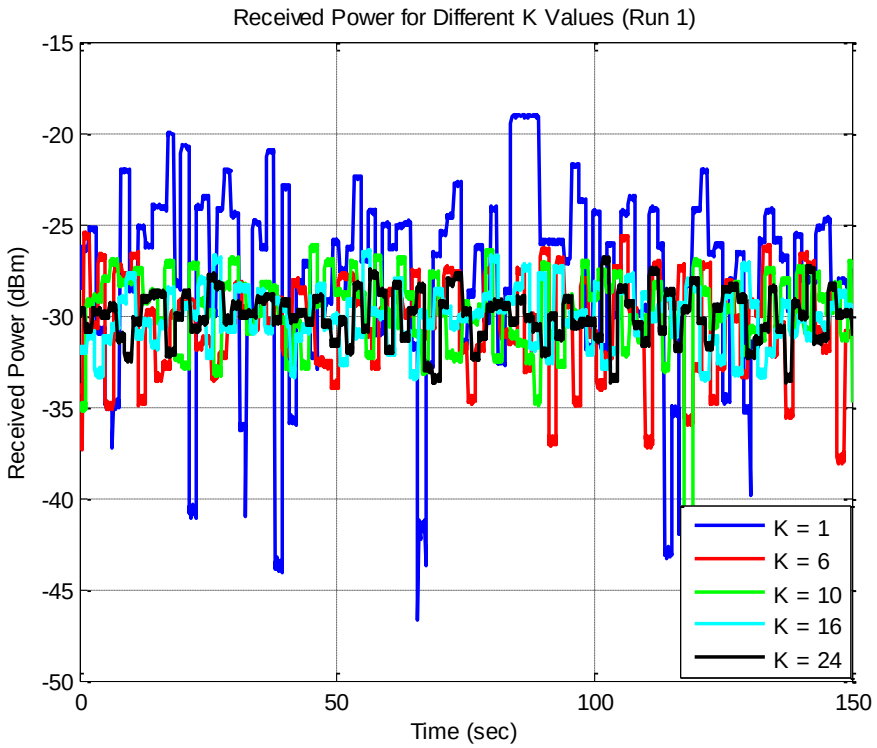


Figure 34: The collected power vs. time measurements for 5 different *K* values for Run 1

Table 9: Maximum, minimum and range of power fluctuation for the five *K* values for the five runs

		Run 1	Run 2	Run 3	Run 4	Run 5
K = 1	Min (dBm)	-47	-47	-59	-47	-49
	Max (dBm)	-19	-18	-19	-29	-29
	Range (dB)	28	29	40	28	30
K = 6	Min (dBm)	-38	-42	-42	-42	-44
	Max (dBm)	-25	-25	-25	-25	-25
	Range (dB)	13	17	16	17	20
K = 10	Min (dBm)	-41	-39	-36	-39	-37
	Max (dBm)	-26	-27	-25	-26	-26
	Range (dB)	15	13	10	13	12
K = 16	Min (dBm)	-34	-36	-36	-37	-35
	Max (dBm)	-26	-26	-27	-27	-26
	Range (dB)	8	9	9	10	9

K = 24	Min (dBm)	-34	-34	-34	-34	-34
	Max (dBm)	-27	-27	-27	-27	-27
	Range (dB)	7	7	7	7	7

Validating the Results

It was important to validate that the collected data are in agreement with the Ricean fading distribution. Validation was accomplished by estimating the *K* value from the collected data, plotting their PDFs and CDFs, and plotting the level crossing rates.

Estimating the Value of *K*

The moment method estimation developed by Greenstein explained in Chapter Three was used to estimate the value of *K* from the collected power vs. time measurements. The Greenstein method was coded into MALAB to process the collected data to estimate the value of *K*. The MATLAB code is shown in Appendix A. Table 10 shows the estimated values of *K* for the five inputted values of *K* = 1, 6, 10, 16 and 24 for the five runs as well as their means and standard deviations. The calculated means for the five runs are 9%, 7%, 3%, 7% and 11% less than the inputted value for *K* = 1, 6, 10, 16 and 24, respectively. The estimated *K* value is smaller than the inputted *K* value because the measured attenuators range of attenuation was between 24.67 dB to 31.96 dB. The MATLAB program was written under the assumption that the attenuators have a range of attenuation of 31.5 dB as stated in the attenuators datasheet. The measured low attenuation range means that the signals are stronger than they should be which leads to higher fluctuation range in the received signal; the higher the signal fluctuation the lower the estimated *K* value. The lower the standard deviation values as compared to the mean,

the better the results. This means that the results are closer to the mean value and less spread which makes them closer to the inputted values. For K = 1, the results showed a relatively higher standard deviation than for high K values. One standard deviation means that around 68% of the estimated values are within the range of (mean ± standard deviation). One standard deviation represents 40%, 10.5%, 10%, 6.8% and 10% of the means for the results of K = 1, 6, 10, 16 and 24, respectively.

Table 10: Estimated K values

Inputted (K)	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Standard Deviation
K = 1	0.594	1.2051	0.5387	1.3766	0.8301	0.9089	0.37044
K = 6	5.3482	5.6539	6.5153	4.9608	5.3577	5.5672	0.5844
K = 10	9.7866	10.0574	8.4987	11.1522	9.3454	9.7681	0.9732
K = 16	13.733	16.2358	15.7351	15.1319	13.8917	14.9455	1.0172
K = 24	19.9727	22.0759	19.6114	20.5008	24.6249	21.3571	2.055

PDF and CDF

MATLAB was used to plot the PDF and CDF of the collected data. Figure 35 shows the PDF of the received power for different K values for the first run. The PDF plots of the received power for the other four runs are shown in Appendix D. The measured PDF plots for the five runs show good agreement with the simulated results shown in Figure 6. The measured and simulated PDFs indicate that as the value of K increases, the PDF gets narrower and sharper which indicates that the signal fluctuates less. This is in agreement with Table 9 that showed the measured fluctuation in the received signal for the five K values. Figure 36 shows the CDF of the measured received power for different K values for the first run. The CDF plots for the other four runs are shown in Appendix D.

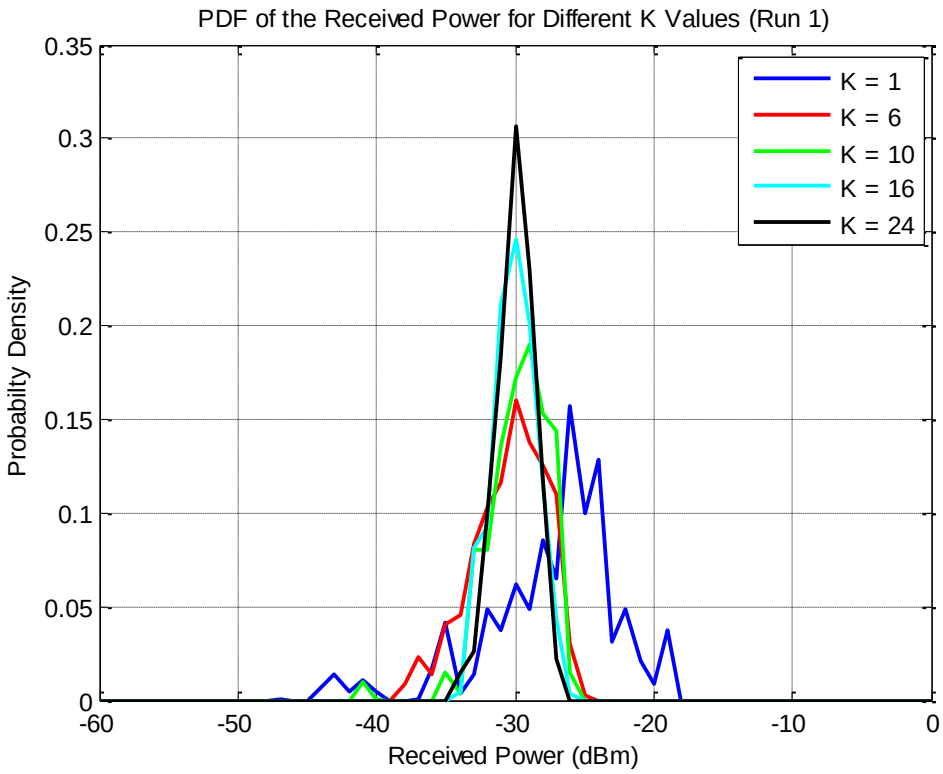


Figure 35: PDF of the received power for different *K* value for the first run.

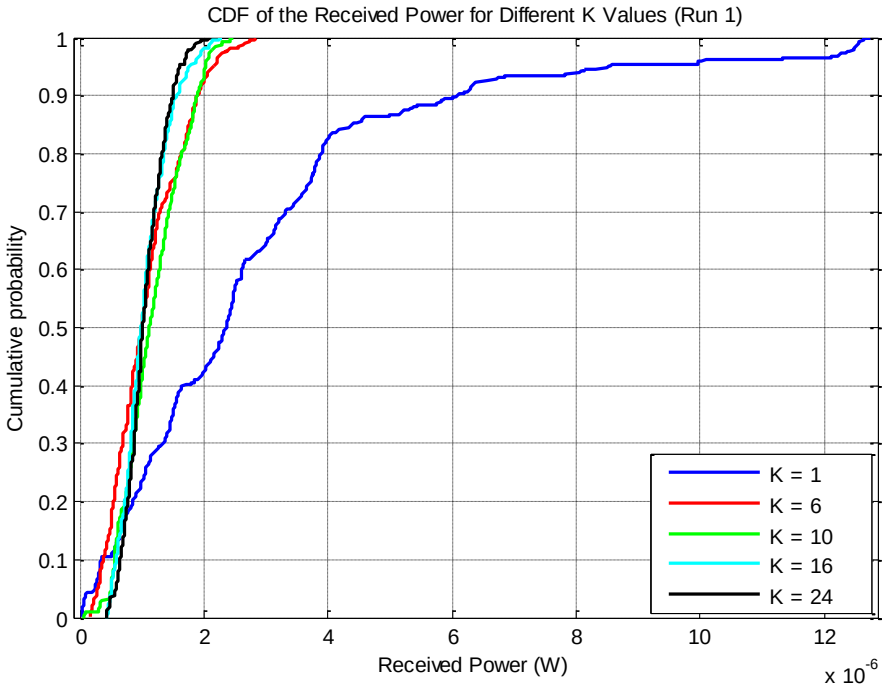


Figure 36: CDF of the received power for different *K* values for the first run

A MATLAB distribution fit tool (dfittool) was used to fit the CDF of the collected data to an ideal Ricean CDF. Figure 37, Figure 38, Figure 39, Figure 40 and Figure 41 show the CDF Ricean fit for the collected received power for the first run for K = 1, 6, 10, 16 and 24, respectively. The CDF Ricean fit for the collected received power for the other four runs for all K values is shown in Appendix D. The presented fitted data show that the collected results are in a good agreement with the Ricean distribution.

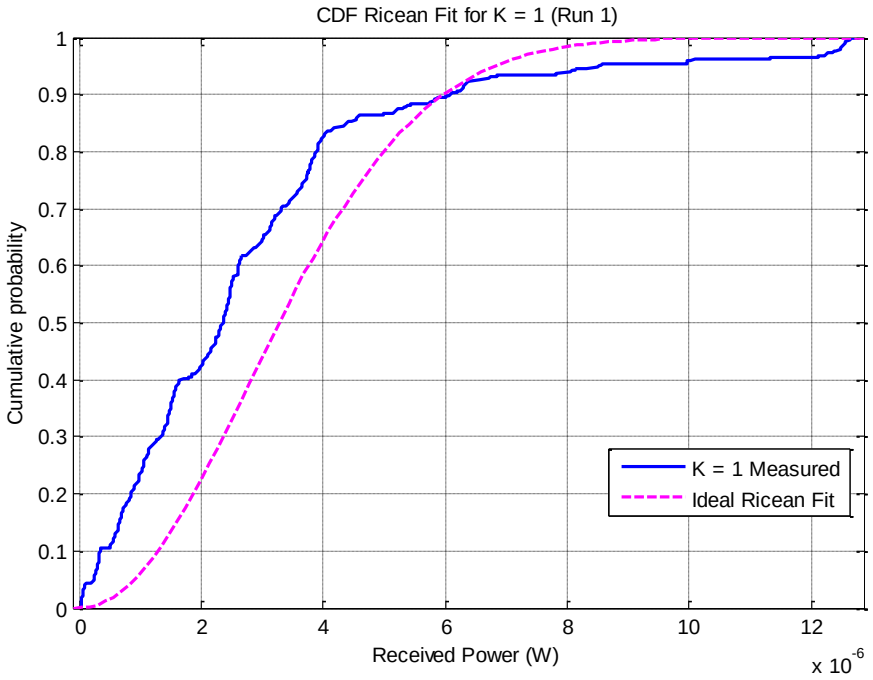


Figure 37: CDF Ricean fit for the collected received power for K = 1 of the first run

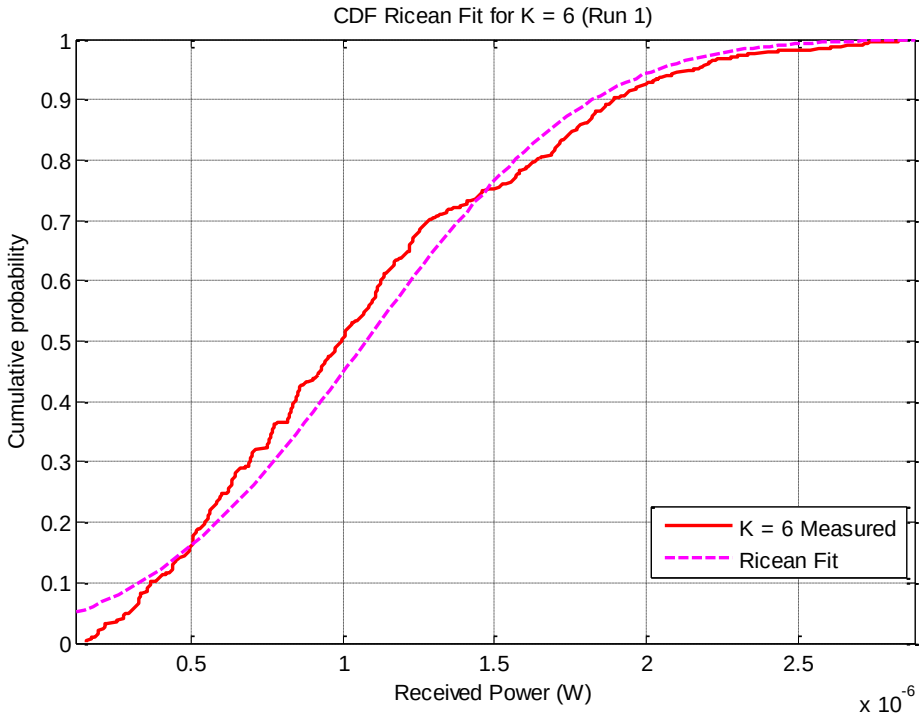


Figure 38: CDF Ricean fit for the collected received power for $K = 6$ of the first run

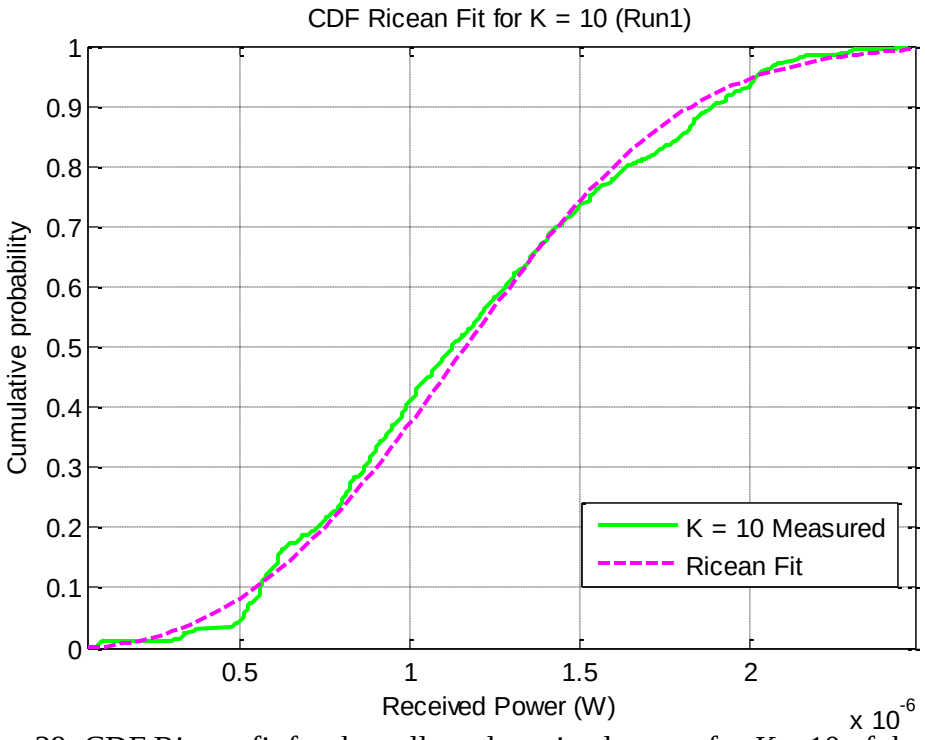


Figure 39: CDF Ricean fit for the collected received power for $K = 10$ of the first run

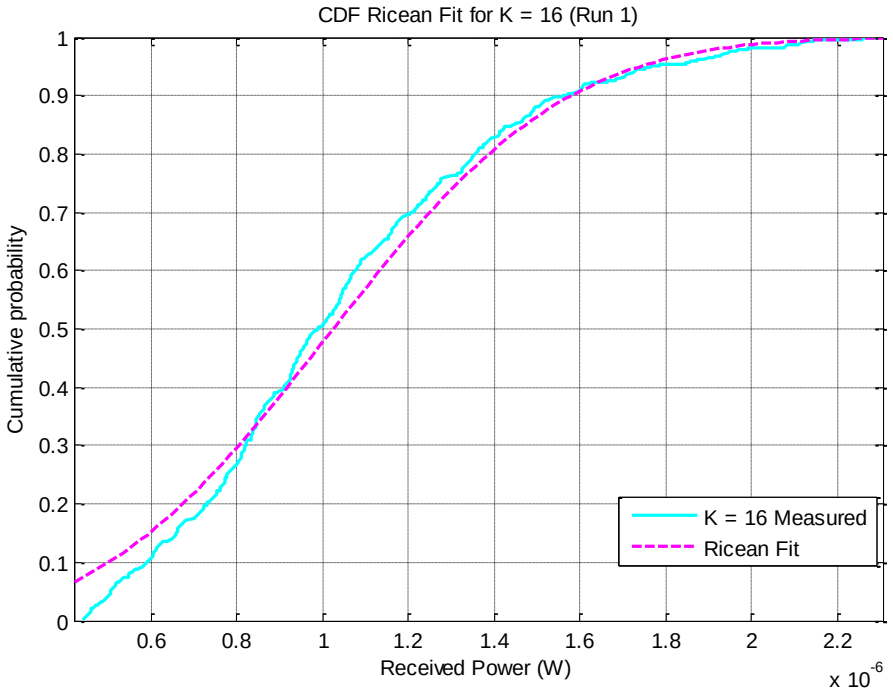


Figure 40: CDF Ricean fit for the collected received power for K = 16 of the first run

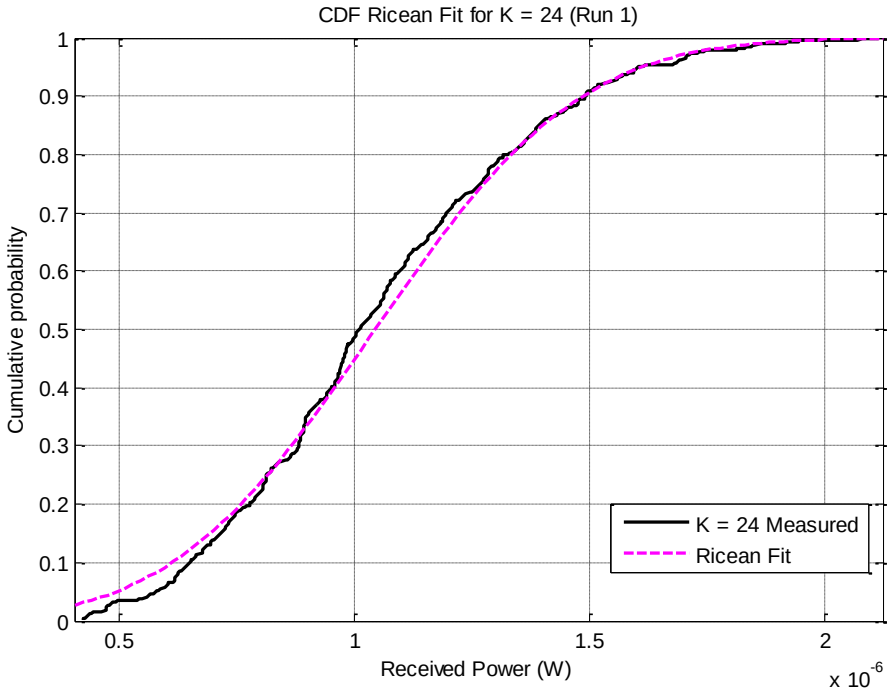


Figure 41: CDF Ricean fit for the collected received power for K = 24 of the first run

Level Crossing Rate

The level crossing rate (LCR) is defined as the number of crossings in a given interval of 1 second at which the power envelope crosses a specified level in the positive or negative going direction. A MATLAB program (shown in Appendix A) was written process the collected power vs. time measurement to plot the level crossing rate for the five *K* values for the five runs. Figure 42 shows a plot of the LCR of the received power for the five different *K* values for the first run. The LCR plots for the other four runs are shown in Appendix D. The LCR provides useful information on how many times a second the signal's fading envelope passes through a specific level. The MATLAB code that was written to plot the LCR divided the fluctuation range into 2 dB levels. The LCR plot shows that as the value of *K* increases, the curve gets sharper and narrower which indicates that for a specific level (L) the LCR decreases as *K* increases. The reason for that is that as *K* increases, less fluctuation occurs to the fading envelope. The measured LCR plot is in good agreement with the LCR plot presented by Xiao in [16]. Xiao and his group presented a novel statistical simulation model for Ricean fading channels. To validate their results, Xiao and his group plotted the LCR (Figure 43) for their simulated results along with the theoretical results for different values of *K*. The LCR plot presented by Xiao verifies the same idea presented in the measured LCR plot that as the value of *K* increases, the LCR curve gets narrower which indicates that for a specific level (L) the LCR decreases as *K* increases due to less fluctuation occurs to the fading envelope

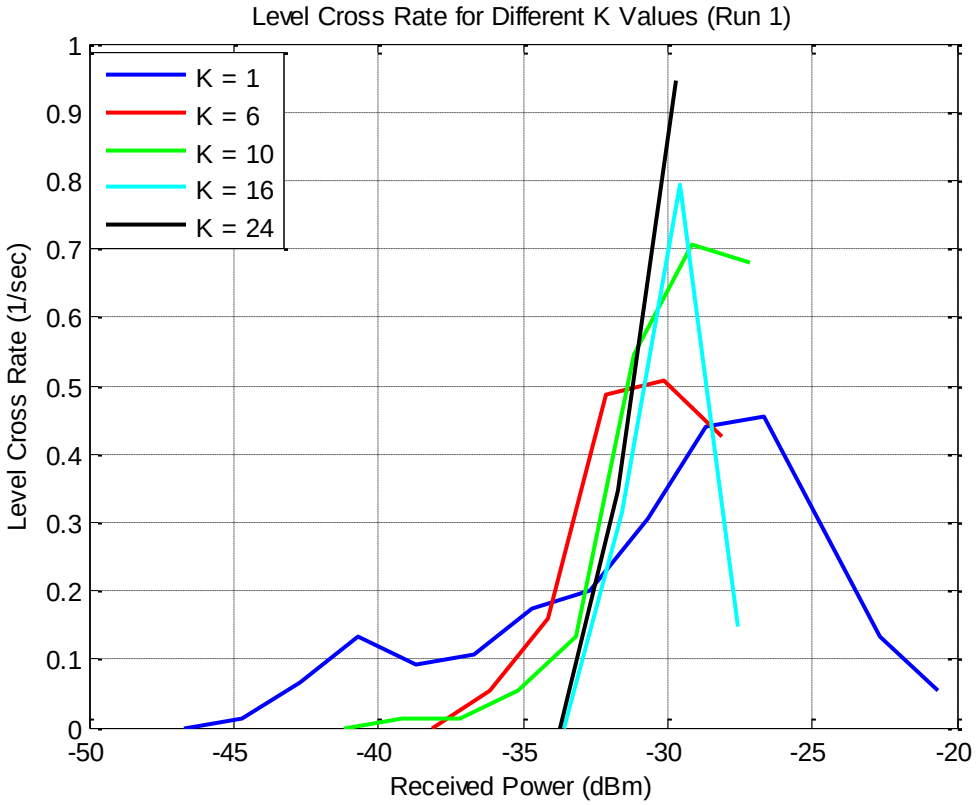


Figure 42: LCR of the received power for the five different *K* values for the first run.

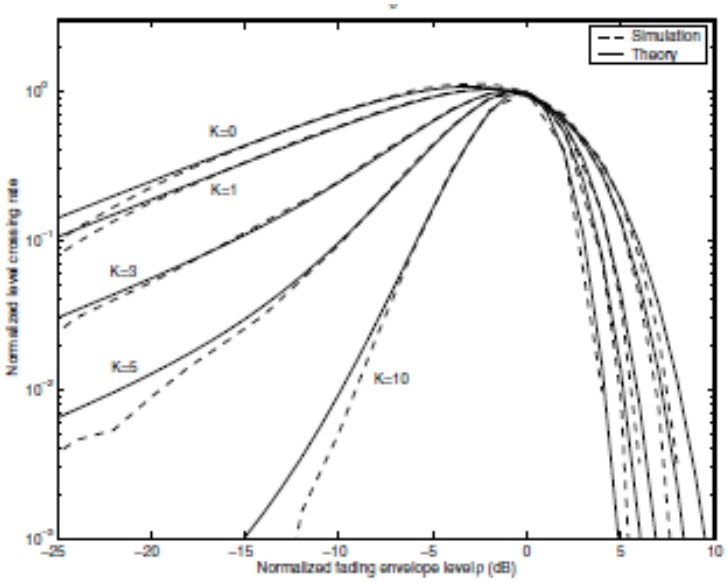


Figure 43: Theoretical LCR for different *K* values [16]

Summary

We conclude that the fade simulator provides an accurate simulation of the Ricean fading over the range of $K = 1$ to $K = 24$ with a systematic bias of about 8% due to the difference in the measured attenuators range (24.67 dB to 31.96 dB) from the value given in the datasheet of 31.5 dB.

CHAPTER SIX

EVALUATING THE PERFORMANCE OF BROADBAND RADIOS

After studying the fading characteristics produced by the fade simulator and validating that it yields a Ricean fading distribution, the fade simulator was then used to evaluate the performance of broadband radios under different varying fading conditions. The radios under test were Airspan (MicroMAX 5.8 GHz TDD Ext base station (BS) and ProST 5.8 GHz TDD Ext subscriber station (SS)) radios and Harris (SeaLancet RT-1944/U) radios.

The Airspan radios have a web-interface which is used to fully control all the parameters of the radios such as the transmitted power, the operating frequency, the channel bandwidth, and the IP configuration as well as monitoring in real time parameters such as received signal to noise ratio (SNR) and received signal strength indicator (RSSI). The Airspan base station manages the communication between itself and any subscriber station or any two subscriber stations. Meaning that, in order for two subscriber stations to communicate with each other, the base station has to be the intermediate mode.

The Harris (SeaLancet RT1944/U) radios are still under development. The interface program used to control the radios does not provide access to many parameters. The parameters that can be changed using the interface program are input power, operating frequency and which of the two antennas ports is used. The only parameter that

can be monitored is the received SNR. To monitor the SNR, the refresh button has be pushed manually whenever a SNR measurement is needed. Each Harris radio can be configured as either a master or a slave. The master is considered the base station and the slave is considered the subscriber station.

The fade simulator was used between the radios (base station and subscriber station or master and slave) to simulate Ricean fading for different values of *K*. The throughput of the radios under different varying fading condition was tested using Jperf which is a network performance measurement tool explained later in the chapter.

Test Setup

Airspan Radios

The setup procedure for the Airspan radios involves four steps: hardware setup, internet protocol (IP) configuration, frequency setup and Jperf setup.

Hardware Setup. Figure 44 shows the Airspan radios test hardware setup. The fade simulator was run for different values of *K*. The beamformer's 5.8 GHz RF input signal and the Tx/Rx signal were driven from the Airspan's base station. The base station used was the Airspan MicroMAX 5.8 GHz TDD Ext. The base station has two input/output ports, which are the antenna port and the power/data port. The base station operates in the 5.8 GHz band and employs the WiMAX technology that uses orthogonal frequency division multiplexing (OFDM) as the physical layer. The maximum transmitted power for the base station is 28 dBm and the receiver sensitivity is -103 dBm. The available channel bandwidths are 1.75 MHz, 3.5 MHz, 5 MHz and 10 MHz. The

supported modulation schemes are binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), 16 quadrature amplitude modulation (16 QAM) and 64 QAM. The duplex method is time division duplex (TDD) with 50 % for the downlink and 50 % for the uplink. The base station can support four scheduling schemes which are the unsolicited grant service (UGS), the real time polling (RTP), non-real time polling (nRTP) and best effort (BE).

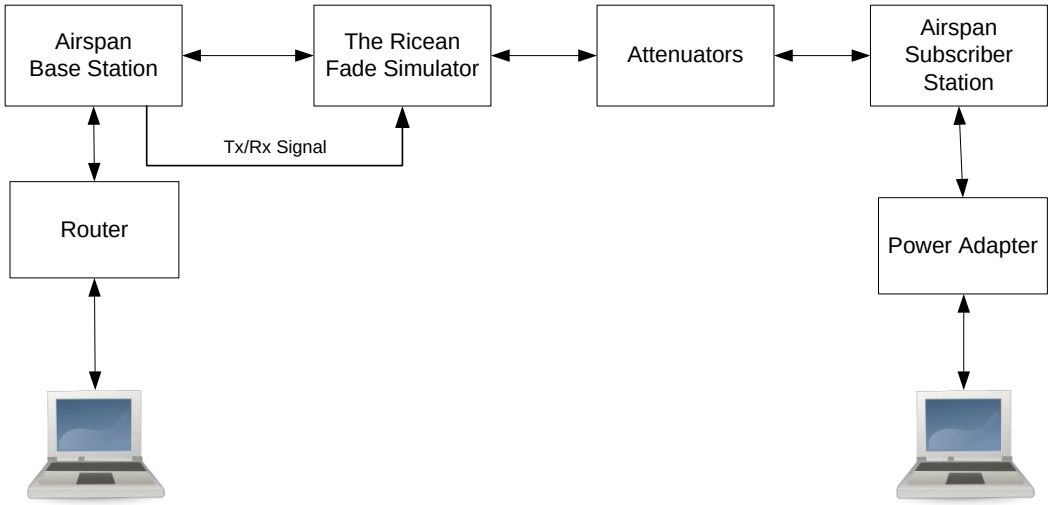


Figure 44: The Airspan radios test hardware setup

The subscriber station used was the Airspan ProST 5.8 GHz TDD Ext. The subscriber station's maximum transmitted power is 24 dBm and its receiver sensitivity is -103 dBm. The subscriber station employs TDD as the duplex method. The available channel bandwidths are the same as the base station.

The router was used to provide 12 voltages to the BS to power it up and to set up the network. The base station laptop was connected to the router through an Ethernet cable. The Airspan SDA-1 Type 1 power adapter (PA) was used. The PA has two

input/output ports; one of them is for the radio and the second one is for LAN/PC connection. The power adapter supplies 12 V to the subscriber station as well as connecting a laptop to the subscriber station through an Ethernet cable. The laptops are used to monitor and control the BS and the SS as well as for running Jperf. The attenuators are used to attenuate the signal to a reasonable level so the received signal will not harm the radios.

IP Configuration. The next step was to setup the network through configuring the IP addresses. The BS, the SS, and each laptop should have a unique IP address that does not conflict with any other IP address in the network. The IP addresses of the BS and the SS were unknown and therefore, the internal factory-set IP address is used to access each SS and the BS. The internal IP address for the SS is:

IP address 10.0.0.1

Subnet address 255.255.240.0

The laptop connected to the SS IP address had to be changed to:

IP address 10.0.0.X

Subnet address 255.255.240.0

Where X is {0, 9} except 1. After changing the IP addresses of the laptop connected to the SS to the factory-set network, the laptops can access the SS through the web-interface. To access the BS or the SS through the web-interface, a web-browser should be opened and the IP address of the BS or the SS (which is 10.0.0.1) is typed in the URL.

After accessing the SS through the web-interface, the IP address of the SS can be changed to the BS network. The following steps used to change the SS IP addresses from the factory-set to the selected network address:

1- Open a web-browser

2- Type 10.0.0.1

3- Go to advance

4- Go to IP Mode

5- Change IP mode to “static”

6- Type in the new IP address, which is:

IP address 192.168.215.xxx

Subnet address 255.255.255.0

Where xxx {0,255}

7- Click submit.

The laptops IP addresses must then be changed to the network addresses.

Frequency Setup. After configuring the IP addreses of the BS, the SS and the laptops and ensuring that each device has a unique IP address that does not conflict with any other IP address in the network, the next step is to ensure that the BS and the SS operate at the same frequency. To check the operational frequency, the following steps are performed at the SS and the BS:

1- Open a web-browser from each laptop

2- Type the IP address of the device connected to the laptop whether it’s a SS or a BS.

3- Check the signal parameters and find out what the operational frequency

4- The selected operating frequency should be 5.8 GHz. If the frequency does not match at any SS or the BS, the following steps are to be followed to change the frequency:

   a. Go to advance

   b. Channel setting

   c. Pick 5.8 GHz from the channel table

   d. Click submit

   e. Click restart

Harris Radios

Figure 45 shows the Harris radios test hardware setup. The SeaLancet RT1944/U radios were designed for severe maritime applications and were optimized for the ocean environment. They employ a proprietary Wi-Fi based technology and can provide a throughput of up to 54 Mbps. They support all IP traffic such as video, file transfer, voice over IP (VoIP) and email. They have a wide operating frequency band selection. The radios are powered using an external 12 V DC power supply. They are connected directly to the laptops using an Ethernet cable. The beamformer's Tx/Rx signal is provided from the radio connected to the beamformer board whether it's a master or a slave.

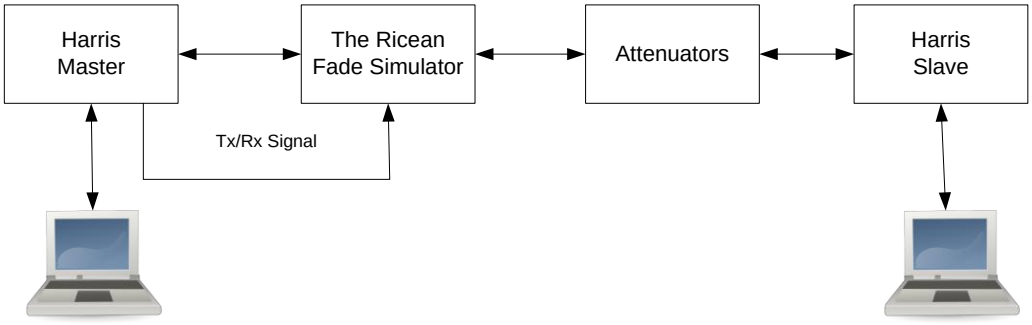


Figure 45: The Harris radios test hardware setup

The SeaLancet RT1944/U radios are controlled using a simple interface program. The parameters that can be controlled are the operating frequency, input power and which path is used. The frequency was set to 5.8 GHz and the input power to -10 dBm. The path that was used is path 1. The interface program does not provide information regarding the channel bandwidth. The radio output was connected to the spectrum analyzer to measure the channel's bandwidth, which was measured to be 20 MHz.

The interface program enables the user to configure the IP address of the radio connected to the laptop running the interface program. The IP address is configured by simply entering the IP address of the radio connected to the laptop. The IP address of the radio is

IP address 192.168.1.xx

Subnet address 255.255.255.0

where xx is the radio's serial number which is found on the top of the radios. The IP address of the laptop connected to the radio must be changed to the same network address

IP address 192.168.1.xxx

Subnet address 255.255.255.0

where xxx is between {0,255}.

Both Airspan and Harris radio throughput was tested under varying fading conditions. The parameters that affect the throughput are the channel bandwidth, the SNR, and the *K* factor. The higher either or both the channel bandwidth and SNR the higher the throughput. The Airspan radio channel bandwidth is changeable but can go up to 10 MHz. The Harris radios have a fixed channel bandwidth of 20 MHz. Under

different varying fading conditions, the signal strength changes which means that the SNR changes. Both Airspan and Harris radios can adapt to the change in SNR by adapting their modulation schemes. As the modulation index increases from BPSK, QPSK, 16-QAM and 64-QAM, higher SNR is required to achieve the same bit error rate (BER). When the SNR drops, the radios adapt by reducing the modulating index to achieve the same BER. This reduces the throughput since the lower the modulation index the lower the throughput. Both Airspan and Harris radios can adapt to the environment which means that they can change their gains automatically to minimize the fluctuation in the received signal. The tests were performed at three SNR ranges, which were SNR = 30 dB, SNR = 15 dB and SNR = 5 dB.

Jperf

Jperf is the GUI front end of Iperf. Figure 46 shows the Jperf GUI. Iperf is a tool to measure the bandwidth and the quality of a network link. Jperf has to be installed on two machines; one serves as the Jperf server and the second one as the client. For the radio tests, Jperf is first installed on the two laptops connected to the radios. Jperf's client sends random packets to the Jperf server. Both the rate of the inputted load as well as the period can be controlled. The Jperf server collects the packets and measures the throughput, jitter and datagram loss. Figure 47 shows the configuration of Jperf to measure the throughput of both the uplink and the downlink for the two radios. To measure the downlink, the master or the base station should be sending the packets and the slave or base station should be receiving. Since in Jperf the client sends the packets,

the master or the base station should be configured as the client and the slave or the base station should be configured as the server and vice versa for the uplink.

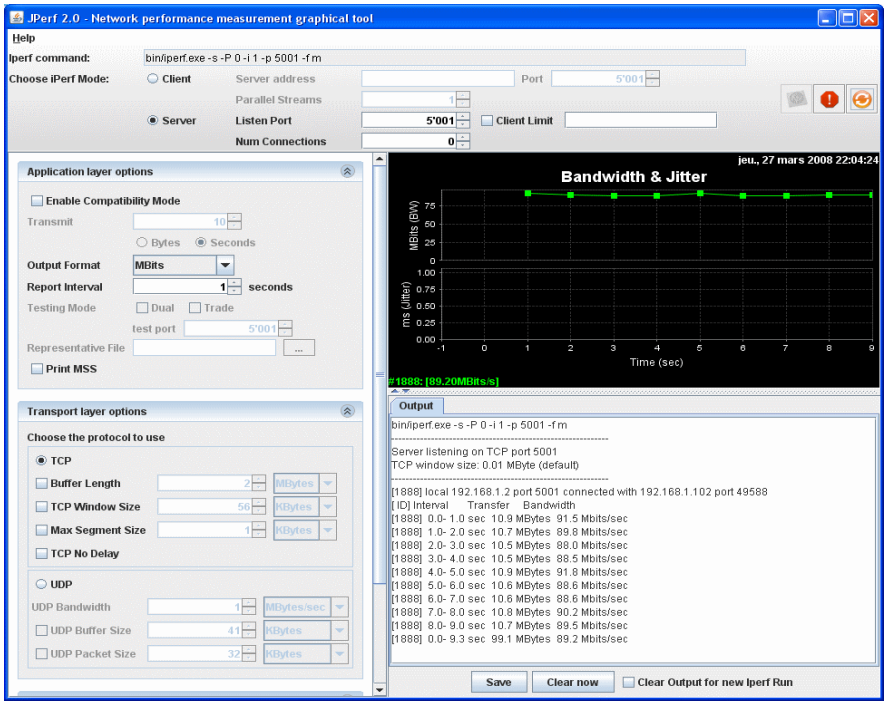


Figure 46: Jperf screenshot

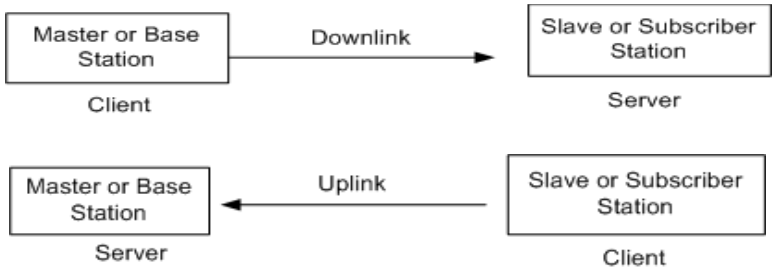


Figure 47: Jperf uplink and downlink configuration for both radios

In Jperf, the user can select between transmission control protocol (TCP) or user datagram protocol (UDP). TCP requires setting up the connection before sending the packets but ensures that the packets are correctly sent to the receiver, whereas UDP just sends the packets without checks. UDP is faster than TCP. For the radio tests, UDP was

used as the transmission protocol. The load, or the number of packets sent from the client to the server, was increased gradually and the throughput was measured at the server. There is an option in Jperf to enable use of a specific file to generate the packets instead of generating random packets each time it is run. A picture file on the laptop running Jperf was used to generate the packets throughout the entire radio testing. Jperf was run for 150 seconds which is the time the fade simulator was run for n = 100. Jperf was set to measure the throughput every second.

Results

For each value of *K*, the fade simulator was run simultaneously with Jperf, the load was increased in steps until the radios reached throughput saturation at which point increasing the load does not increase the throughput. The collected data is the average throughput for the 150 seconds for both the uplink and downlink for different values of *K* as the load increased for both radios, for the three SNR ranges. The throughput was measured without the fade simulator to measure the maximum throughput the radios can achieve.

Airspan Radio Results

Table 11 shows the measured throughput for the Airspan radios for both the downlink and the uplink for SNR = 30 and 15 dB for different values of *K*. As the value of *K* increases, the fluctuation in the signal decreases. The deeper the fades the less the SNR which leads to less throughput. The Airspan radios adapt to the signal fluctuations by increasing the gain. SNR = 30 dB is considered a high SNR at which the radios can

achieve the highest throughput since it enables the radios to transmit at the highest modulation index of 64-QAM. For a lower SNR of 15 dB, the throughput is expected to drop. For SNR = 30 dB, the measured Airpsan radio maximum throughput without the fade simulator is 25.7 Mbps for the downlink and 20.8 Mbps for the uplink. For SNR = 30 dB, the lowest measured throughput with the fade simulator connected between the transmitter and the receiver for the lowest *K* value (maximum fluctuation) of 1 is 22.8 Mbps for the downlink and 16.4 Mbps for the uplink. For *K* = 1, the signal passed through the fade simulate fluctuates in the range of 30 dB. The Airspan radio interface provides data to plot the measured SNR in real time. The measured fluctuation in the SNR for *K* = 1 is 9 to 10 dB, which means for the high SNR of 30 dB, the throughput is not affected much since the SNR remains in the twenties range.

For SNR = 15 dB, the measured Airpsan radio maximum throughput without the fade simulator is 8.59 Mbps for the downlink and 9.43 Mbps for the uplink. For SNR = 15 dB the lowest measured throughput with the fade simulator connected between the transmitter and the receiver for the lowest *K* value of 1 is 2.82 Mbps for the downlink and 5.19 Mbps for the uplink.

For SNR less than 5 dB, the radios could not communicate. The lowest measured SNR at which the radios can communicate is 7 dB.

Table 11: Airspan radios measured throughput for both downlink and uplink for different *K* values for SNR = 30 and 15 dB

Aispan Radios Downlink							
	Load (Mbps)	Throughput (Mbps)					
		K =1	K = 6	K = 10	K = 16	K = 24	w/o FS
SNR = 30 dB	10	9.58	6.64	9.7	9.76	9.8	9.86
	20	17.2	17.1	17.1	19.3	19.3	19.4
	30	22.6	25.5	24.1	25.6	25.7	25.7
	40	22.8	25.5	25.6	25.5	25.5	25.7
SNR = 15 dB	5	2.76	2.8	2.76	2.76	3.88	4.91
	10	2.8	5.66	5.76	6.2	6.4	8.62
	15	2.82	5.7	5.8	6.44	7.66	8.59
Airspan Radios Uplink							
SNR = 30 dB	10	8.1	9.2	9.3	9.3	9.4	9.72
	20	14.4	18.7	18.4	18.4	18.6	19.2
	30	16.7	19.7	19.4	20.3	20.7	20.9
	40	16.4	20	20.2	20.3	20.9	20.8
SNR = 15 dB	5	3.83	4.5	4.56	4.83	4.82	4.8
	10	5.72	5.72	6.28	6.63	6.9	9.39
	15	5.19	5.95	6.44	6.59	6.85	9.43

The collected throughput was processed using MATLAB. The measured throughput was divided by the maximum achievable throughput without the fade simulator for each SNR to estimate the range of throughput ratio as *K* changes. The MATLAB code is shown in Appendix A. Figure 48 and Figure 49 show the throughput ratio for different values of *K* for the downlink as the load increases. Figure 50 and Figure 51 show the throughput ratio for different values of *K* for the uplink as the load increases. For the high SNR of 30 dB, for $K = 1$, the achieved throughput was 90 % for the downlink and 80% for the uplink of the maximum throughput. As *K* increases the throughput increases to get closer to the maximum achievable throughput. For the SNR of 15 dB, for $K = 1$, the achieved throughput was 30 % for the downlink and 55% for the

uplink of the maximum throughput. As *K* increases the throughput increases to 80% for the downlink 75 % of the maximum throughput for *K* = 24.

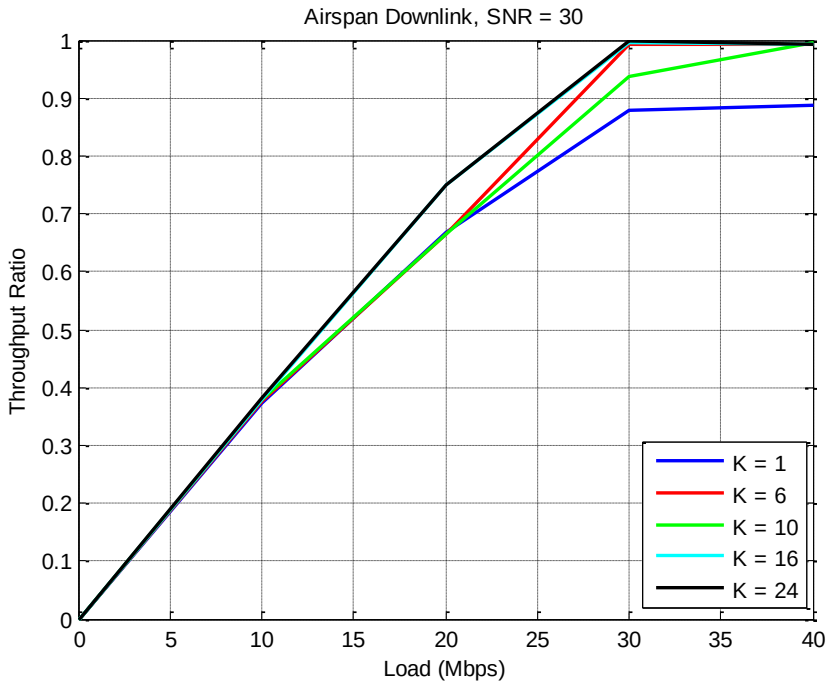


Figure 48: Throughput ratio for Airspan radios downlink for SNR = 30 for all *K* values

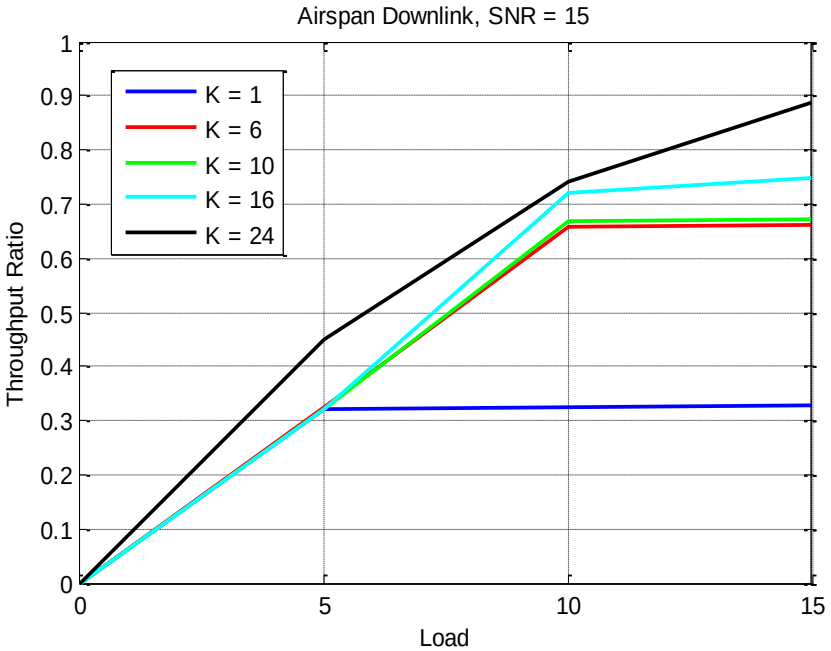


Figure 49: Throughput ratio for Airspan radios downlink for SNR = 15 for all *K* values

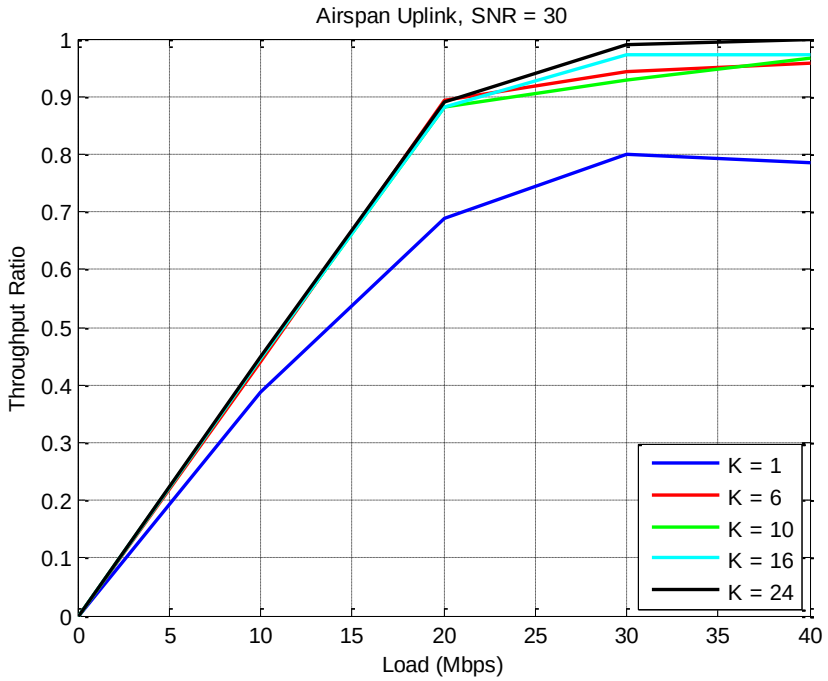


Figure 50: Throughput ratio for Airspan radios uplink for SNR = 30 for all *K* values

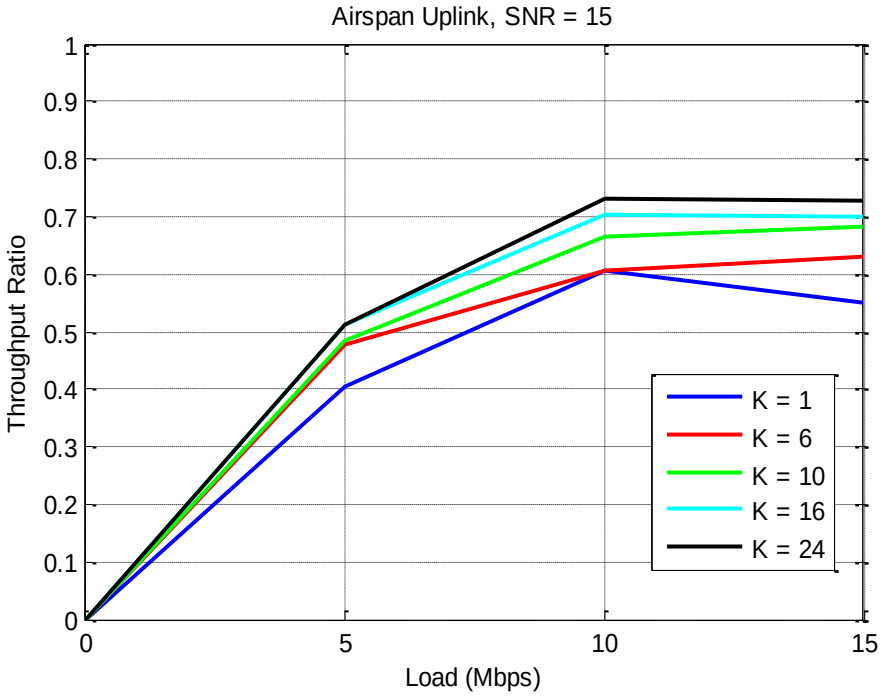


Figure 51: Throughput ratio for Airspan radios uplink for SNR = 15 for all *K* values

Harris Radios Results

Table 12 shows the measured throughput for the Harris radios for both the downlink and the uplink for SNR = 30 and 15 dB for different values of *K*. For SNR = 30 dB, the measured Harris radio maximum throughput without the fade simulator is 35 Mbps for the downlink and 32.7 Mbps for the uplink. For SNR = 30 dB the lowest measured throughput with the fade simulator connected between the transmitter and the receiver for the lowest *K* value of 1 is 29.5 Mbps for the downlink and 21.7 Mbps for the uplink.

For SNR = 15 dB, the measured Harris radios maximum throughput without the fade simulator is 17.4 Mbps for the downlink and 15.6 Mbps for the uplink. For SNR = 15 dB the lowest measured throughput with the fade simulator connected between the transmitter and the receiver for the lowest *K* value of 1 is 9.64 Mbps for the downlink and 4.02 Mbps for the uplink. For SNR less than 5 dB, the radios could not communicate. As with the Airspan radios, the Harris radios can adapt to the fluctuation in the received signal by increasing the gain. Unlike Airspan, the Harris radios interface program does not enable a real time measurement of the received SNR.

Table 12: Harris radios measured throughput for both downlink and uplink for different *K* values for SNR = 30 and 15 dB

Harris Radios Downlink							
	Load (Mbps)	Throughput (Mbps)					
		K =1	K = 6	K = 10	K = 16	K = 24	w/o FS
SNR = 30 dB	10	9.68	9.69	9.67	9.8	9.65	9.73
	20	19.8	19.3	19.4	19.4	19.4	19.3
	30	28.4	28.9	28.9	28.8	28	29
	40	29.2	34.5	34.7	34.5	34.8	34.4
	50	29.5	34.2	34.8	34.8	35	35

SNR = 15 dB	10	9.26	9.73	9.7	9.73	9.73	9.69
	20	10.4	12.7	17.3	16.7	16.5	16.8
	30	9.64	13	17.4	16.7	16.5	17.4
Harris Radios Uplink							
SNR = 30 dB	10	9.71	9.55	9.59	9.55	9.58	9.58
	20	19.2	19.3	19.2	19.2	19.3	19.5
	30	21.6	25.6	23.9	26.9	25.7	28.7
	40	21.7	23.9	25.3	25.6	26.9	32.7
	50	21.7	23.9	25.3	25.6	26.9	32.7
SNR = 15 dB	10	3.57	4.24	6.26	6.11	6.57	9.75
	20	3.9	4.23	6.09	6.1	5.91	15.8
	30	4.02	4.22	6.14	6.01	6.44	15.6

The measured throughput was processed using MATLAB to plot the throughput ratio for different values of *K*. The MATLAB code is shown in Appendix A. Figure 52 and Figure 53 show the throughput ratio for different values of *K* for the downlink as the load increases. Figure 54 and Figure 55 show the throughput ratio for different values of *K* for the uplink as the load increases. For the high SNR of 30 dB, for $K = 1$, the achieved throughput was 85 % for the downlink and 65% for the uplink of the maximum throughput. As *K* increases the throughput increases to get closer to the maximum achievable throughput. For the SNR of 15 dB, for $K = 1$, the achieved throughput was 55 % for the downlink and 25% for the uplink of the maximum throughput. As *K* increases the throughput increases to 98% for the downlink 40 % of the maximum throughput for $K = 24$.

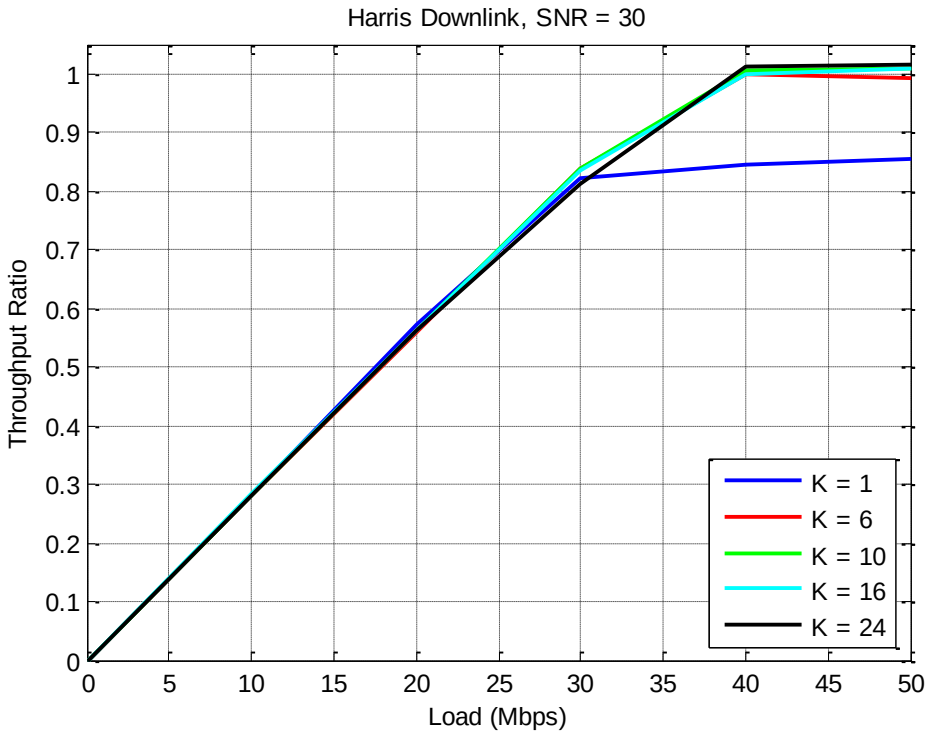


Figure 52: Throughput ratio for Harris radios downlink for SNR = 30 for all *K* values

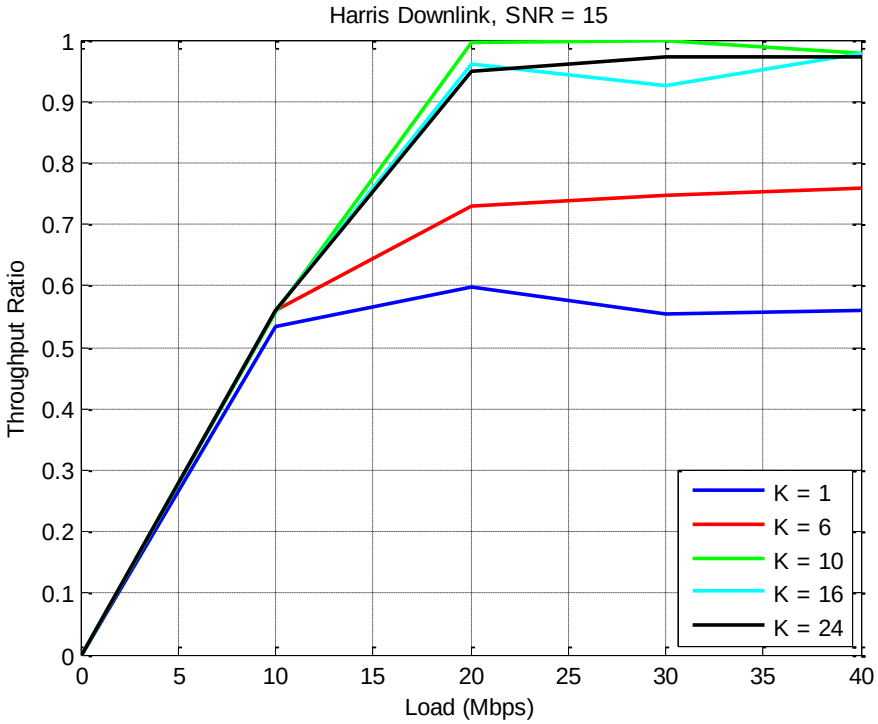


Figure 53: Throughput ratio for Harris radios downlink for SNR = 15 for all *K* values

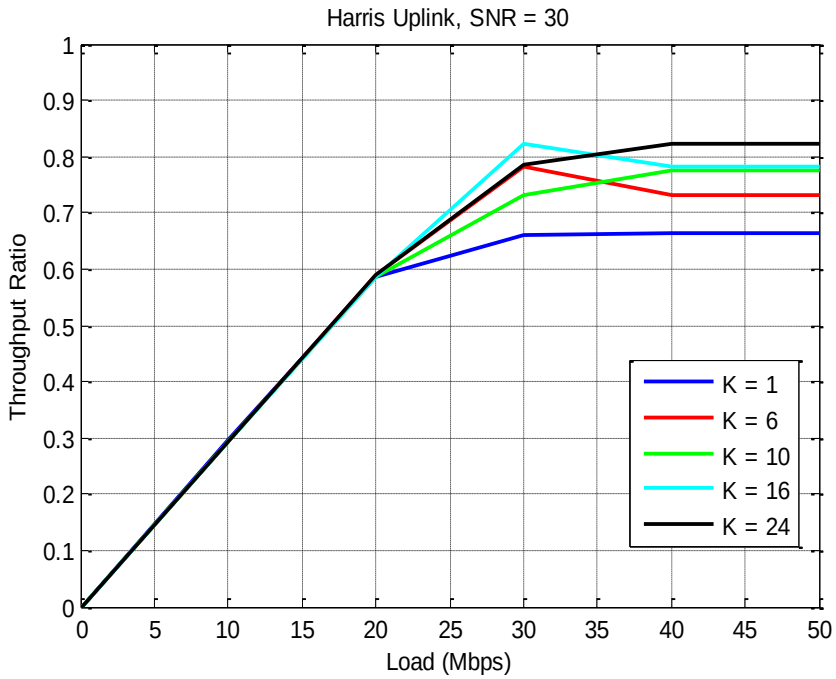


Figure 54: Throughput ratio for Harris radios uplink for SNR = 30 for all *K* values

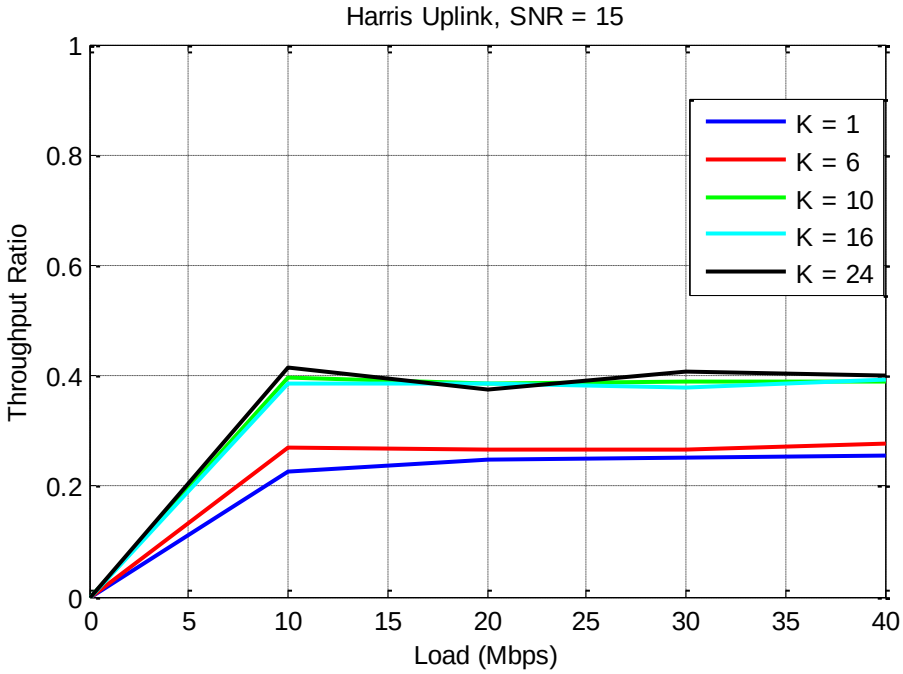


Figure 55: Throughput ratio for Harris radios uplink for SNR = 15 for all *K* values

The measured maximum throughput for different values of *K* was divided by the maximum throughput the radios can achieve (without the fade simulator) for different SNR to examine the radios robustness to Ricean fading. Figure 56 shows the Airspan radio downlink percentage of maximum throughput for different *K* values for SNR = 30 dB and 15 dB. Figure 57 shows the Harris radio downlink percentage of maximum throughput for different *K* values for SNR = 30 dB and 15 dB. Figure 58 shows the Airspan radio uplink percentage of maximum throughput for different *K* values for SNR = 30 dB and 15 dB. Figure 59 shows the Harris radio uplink percentage of maximum throughput for different *K* values for SNR = 30 dB and 15 dB.

At SNR = 30, both radios (for both uplink and downlink) achieved almost the maximum throughput the radios can achieve for *K* = 6 to 24. For *K* = 1, the measured throughput was 90% of the maximum throughput. At SNR = 15 dB, for the downlink, the Harris radio achieved higher percentage of throughput than the Airspan radio. For each *K* run, the measured throughput of the Harris radio was fluctuating, which led to higher average throughput. Unlike Harris, the Airspan radio showed a stable throughput but with a lower average. At SNR = 15 dB, for the uplink, both Harris and Airspan radios showed almost the same percentage of the maximum throughput. The Airspan radio was more robust with respect to decrease in throughput at low *K* values. While the Harris radio throughput began to decline for *K* less than ten, the Airspan radio throughput was almost unchanged until *K* fell below five.

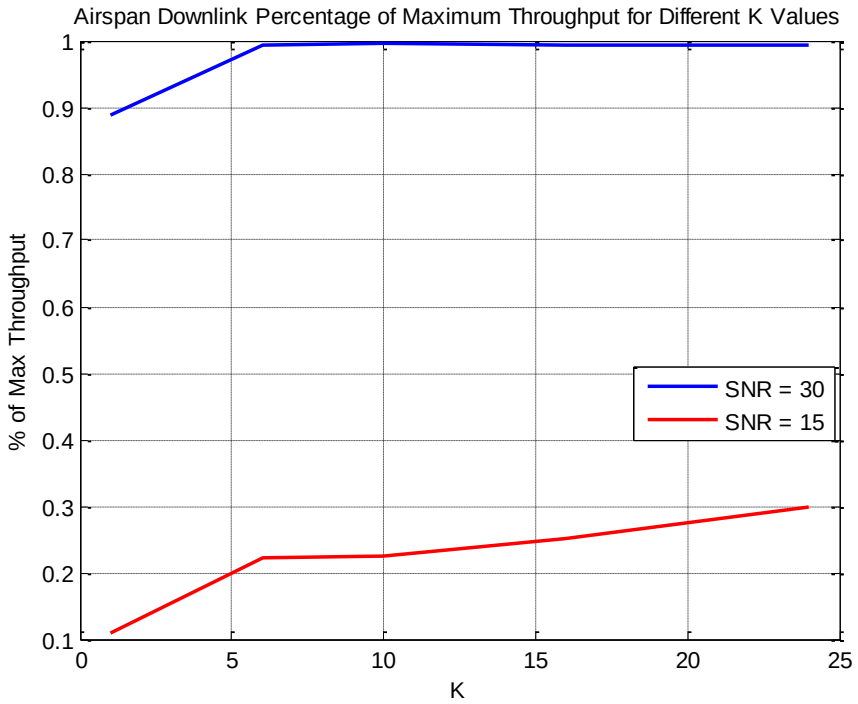


Figure 56: Airspan downlink percentage of maximum throughput for different *K* values

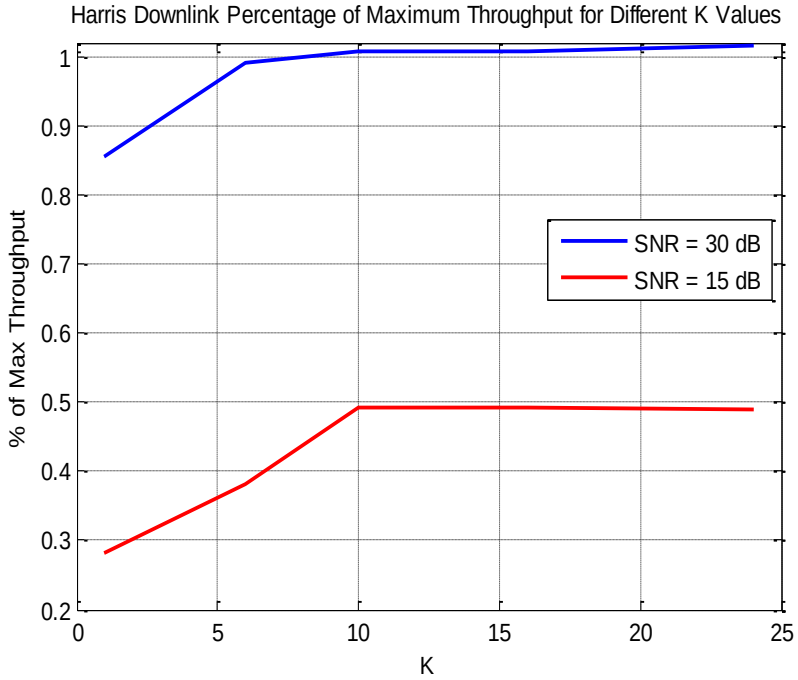


Figure 57: Harris downlink percentage of maximum throughput for different *K* values

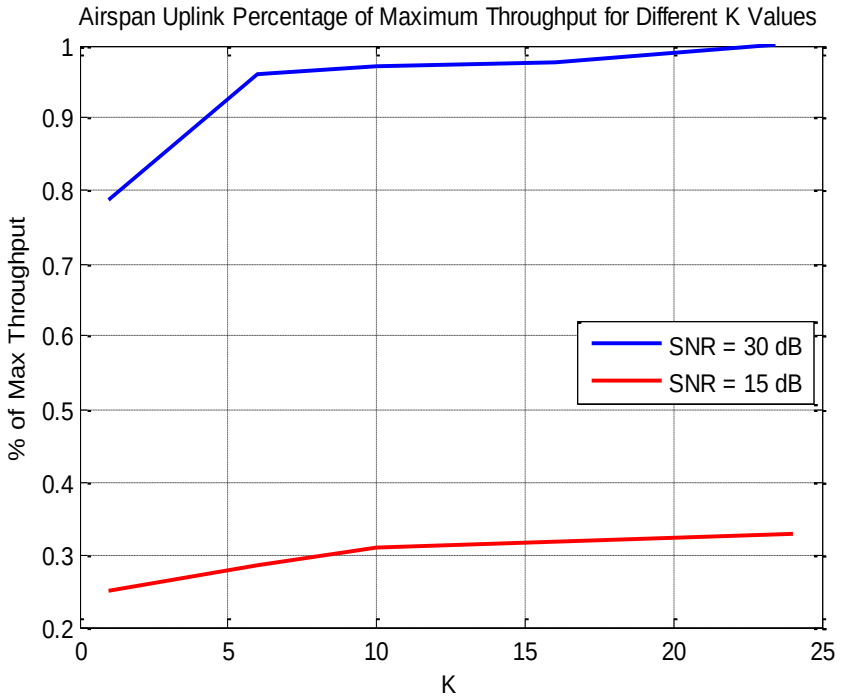


Figure 58: Airspan uplink percentage of maximum throughput for different *K* values

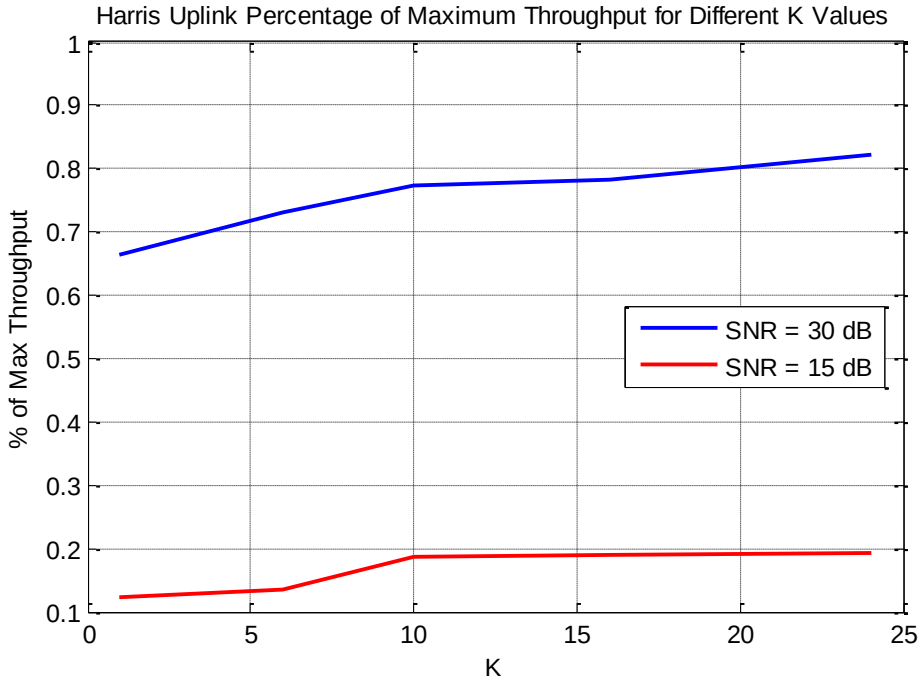


Figure 59: Harris uplink percentage of maximum throughput for different *K* values

CHAPTER SEVEN

CONCLUSION AND FUTURE WORK

This thesis introduced a detailed description of a digital hardware based Ricean fading channel simulator. The fade simulator was run using a MATLAB program that sends a serial bit stream to the attenuators and phase shifters on the beamfomer board. The serial bit stream was generated according to the Ricean fading distribution. The first experiment was to run the fade simulator for different values of *K*. The input signal was driven from the vector network analyzer and the output signal was received by the spectrum analyzer in the zero span mode to collect power versus time measurements. The collected power versus time measurements showed that as *K* increased the fluctuation in the received signal decreased which is in agreement with the Ricean distribution since *K* is a measure of the severity of the fade and as *K* increases the ratio of the constant line of sight signal to the fluctuating multipath signal increases. To validate that the fading characteristics produced by the fade simulator are Ricean, the collected power versus time measurements were processed using MATLAB. The estimated value of *K* was close to the inputted *K* value and in most of the cases was in an average of 10% lower than the inputted value. The plotted PDF, CDF and LCR showed a good agreement with the theoretical PDF, CDF and LCR of the Ricean distribution.

The second experiment was to use the fade simulator to evaluate the performance of broadband radios under different varying fading conditions. The tested radios were the Airspan (MicroMAX 5.8 GHz TDD Ext base station (BS) and ProST 5.8 GHz TDD Ext subscriber station (SS)) and Harris SeaLancet RT1944/U. The fade simulator was

connected between the BS and the SS and the master and the slave. The Tx/Rx signal was derived from the radio connected to the beamformer board. Jperf was used to measure the throughput for the radios. The throughput was measured for different SNR and for different *K* values as the load increased for both the uplink and the downlink. The throughput of the two radios showed that as *K* increased, the throughput increased due to less fluctuation that occurred to the signal. Both radios utilize adaptive modulation and adaptive gain to minimize the fluctuation and increase the throughput. For higher SNR, the throughput increased due to higher modulation index. The Airspan radios showed a more stable (less fluctuating) throughput than the Harris radios but with less average throughput.

The GUI was designed to use the *K*-model developed by Greenstein and his group and can be used to control the fade simulator. The radios can be tested under different antenna heights, antenna beamwidths, separations between the transmitter and the receiver and seasons. This will enable to test of the performance of the radios under different conditions in the laboratory environment instead of real life tests that can be expensive and hard to control. The model has a constant to be optimized via regression fitting. This constant should be in the around 1.1 since the estimated *K* value in most cases was 10% lower than the inputted *K* value.

The overwater path fading literature study showed that an over water path can have deep fading in the range of 30-40 dB. For $K = 1$, the fade simulator can introduce a fluctuation in the same range which means that the fade simulator can be used to test radios under maritime conditions.

The fade simulator can be modified to simulate Rayleigh fading. The MATLAB code used to generarte the serial bit stream can be modified for Rayleigh fading. The key parameter in Rayleigh fading is the Doppler shift which is related to the coherence time. The current fade simulator has a coherence time of 1.4 to 1.5 seconds (depends on the speed of the computer used), which is in a good agreement with the measured coherence time for Ricean fading. To change the coherence time (different Doppler shifts) the serial bit stream can be sent all at once to the FPGA memory. The FPGA can be programmed to send the serial bit stream to the attenuators and the phase shifters at the selected coherence time (Doppler shift).

The massive amount of Ricean fading data collected by Greenstein group showed that *K* changes log-normally overtime. The MATLAB program can be modified to generate random values of *K* between (1 and 24) according to the lognormal distribution and use them to run the fade simulator instead of one constant value throughout the entire run.

The range of attenuation of each channel of the beamformer board was measured. The measurement showed that there is up to 6-7 dB difference in the attenuation between the channels. Calibrating the beamformer board before running the fade simulator should give better results.

It will be desirable to integrate the Labview program that collects the data from the spectrum analyzer with the GUI that runs the fade simulator and then to add the code that estimates *K*, plots the PDF, CDF and LCR to the GUI. This will make the fade simulator more powerful since the user can use the GUI to run the fade simulator and to

observe the power versus time measurements in real time as well as studying the fading characteristics.

The bi-directional amplifier should be used instead of attenuators to normalize the signals. The use of the bi-directional amplifier will increase the dynamic range of the fade simulator as well as improving the SNR.

REFERENCE CITED

[1] Rappaport, Theodore S. *Wireless Communications, Principles and Practice.* s.l. : Prentice Hall , 2001.

[2] *Rayleigh Fading Channels in Mobile Digital Communication Systems Part 1: Charcterization.* Sklar, Bernard. 7, s.l. : IEEE, 1997, Vol. 35.

[3] *Ricean K-Factors in Narrow-Band Fixed Wireless Channels: Theory, Experiment, and Statistical Models.* Larry J. Greenstein, Saeed S. Ghassemzadeh, Vinko Erceg and David G. Michelson. 8, s.l. : IEEE Transaction on Vehicular Technology, 2009, Vol. 58.

[4] *Moment-Method Esimation of the Ricean K-Factor.* L.J Greenstein, V. Erceg. 6, s.l. : IEEE Communications Letters, 1999, Vol. 3.

[5] *EM Algorithm for Estimating the Parameters of a Multivariate Complex Rician Density for Polarimetric SAR.* Marzwetta, Thomas L. s.l. : IEEE, 1995.

[6] *Maximum Likelihood Estimation of the K Factor in Ricean Fading Channels.* Beaulieu, Yunfei Chen and Norman C. 12, s.l. : IEEE Communication Letters, 2005, Vol. 9.

[7] *Temporal Variation of the Refractive Index in Coastal Water.* Norland, Richard. s.l. : 2nd Microwave and Radar Week in Poland , 2006, Vols. International Radar Symposium, IRS 2006, Proceedings.

[8] *Fade Statistics and Propagation Events at C Band for Two overwater, Line-of-Sight Propagation Pathe over a 1-year period.* Julius Goldhirsh, G. Daniel Dockery, and Bert H. Musiani. 6, s.l. : Radio Science, 1992, Vol. 27.

[9] *Fade Depth and Diversity Statistics for 8 GHz Over-Water Line-of-Sight Link in Singapore.* Hu, J.T Ong and C.F. 2, s.l. : IEEE Electronics Letters, 2001, Vol. 37.

[10] *Propagation Measurements on an Over-Water, Line-of-Sight Link in Singapore.* Hu, Jin-Teong Ong and Chun-Feng. s.l. : Proceedings of the International Conference on Information, Communications and Signal Processing, ICICS, 1997, Vol. 3.

[11] *Correlation Analysis of the Fading and Weather Conditions on All-Water Digital Microwave Radio Hop.* X. Jin, D. Brown and S. Periyalwar. s.l. : Conference proceedings - Canadian Conference on Electrical and Computer Engineering, 1994, Vol. 1.

[12] *A Multipath Fading Simulator for Mobile Radio .* Gaston A. Arrendondo, William H. Chriss and Edward H. Walker. 11, s.l. : IEEE Transaction on Communications, 1973, Vol. 21.

[13] *A simple Digital Fading Simulator for Mobile Radio.* Leung, Eduardo Casas and Cyril. 3, s.l. : IEEE Transaction on Vehicular Technology, 1990, Vol. 39.

[14] *Efficient Simulation of Ricean Fading within a Packet Simulator.* Ratish J. Punnoose, Pavel V. Nikitin and Daniel D. Stancil. 52, s.l. : IEEE Vehicular Technology Conference, 2000, Vol. 2.

[15] *Experimental Results on the Level Crossing Rate and Average Fade Duration for Urban Fixed Wireless Channels.* Rodolfo Feick, Reinaldo A. Valenzuela and Luciano Ahumada. 1, s.l. : IEEE Transaction on Wireless Communication, 2007, Vol. 6.

[16] *Statistical Simulation Models for Rayleigh and Rician Fading.* Chengshan Xiao, Yahong R. Zheng, Norman C. Beaulieu. s.l. : IEEE International Conference on Communications, 2003, Vol. 5.

APPENDICES

APPENDIX A

MATLAB CODE

MATLAB code to plot the PDF and CDF of the received voltage

```
%%%%%% PDF FUNCTION
A_square= 2*K*sigma^2;
A=sqrt(A_square);

omega=A_square + sigma^2;
x1 = r./(sigma)^2;
x2= exp(-((r.^2 + A^2)/(2*sigma^2)));
bessel_input = (A.*r)/sigma^2;
bessel_eq= besseli(0, bessel_input);
PDF = x1.*x2.*bessel_eq;


%%% CDF FUNCTION
x3=A/sigma;                 %%% First input argument
x4=r./sigma;                %%% Second input argument
Q=marcumq(x3,x4);           %%% Marcum Q-Function
CDF=1-Q;
%%% PLOTTING THE PDF AND CDF
 figure (1)
 plot(r,PDF,'k','linewidth',1.5),grid on
 figure (2)
 plot(r,CDF,'k','linewidth',1.5),grid on
 %%%%%%%
```

MATLAB code to plot the PDF of the received power

```
% MATLAB code to plot the Ricean PDF of the received power
K = 10;      % Input K Value
p=[.01:.01:10]; % power
p_mean=mean(p)/10;% mean of the power

y1 = ((1+KK)/p_mean)*exp(-KK-((KK+1)/p_mean).*p);

bessel_input2=sqrt(((4*KK*(1+KK))/p_mean).*p);
bessel_eq2= besseli(0, bessel_input2);
PDF2 = y1.*bessel_eq2; % Ricean PDF power formula

plot(10*log10(p./1000),PDF2/10,'k','linewidth',2),grid on
```

MATLAB code to generate the serial bit stream

```
% FDS program
% The idea is to generate a number of phase shifts and attenuation
% values such that the receiver will see 8 signals
% one of them is the LOS signal and 7 other signals.
```

```
% The LOS signal's power equals K * the summation of all 7 multipath
% signals

% Step 1 :    generate the LOS signal attenuation value
%             The maximum percenatge should be used for the
%             LOS signal depends on the maximum K value used
%             xi = xo + ratio where xi is the average attenuation
%             for the multipath signals and xo is the LOS attenuatio
%             ratio = 10 log (7*k)/31.5
%             ratio + xo + .15 < 1
%             for k =20, (xo)max = 1 - ratio - .15
k = 1;
tic;
n=100;
log = 10*log10 (7*k);
ratio = log/31.5;
LOS =.15*ones(1,n);
c=LOS(1)-0.15; d=LOS(1)+0.15;
    for m = 1:n;
        per(m,:)= ratio + c + (d-c) * rand(1,7);
    end
     for m =1:n
        perc(m,:) = sort(per(m,:));
     end
    percent=[LOS' perc];
    percent;
    magnitude = percent.*31.5;

  %  phase = 360.*rand(n,8)-180;
    ratio
    for i =1 :n
    test (i)= (sum(percent(i,:))-percent(i,1))/7 - percent(i,1);
    end
    testavg=sum(test)/n
 % -------------- Step 2:   Magnitude and Phase decoders ------
 %
 %              The attenuation values that will be downloaded to
 %              the FBGA can not exceed 31.5 and in steps of .5 dB
 %              The phase shifts values are in between -180 to 180
degrees
 %              in steps of 5.625 degress


 % ---------- Mag decoder

 ro=round(magnitude);
 xo=ro-magnitude;
 xp=abs(xo);
 for i=1:n
    for h=1:8
    if xp(i,h)<.25
     magnitude(i,h)= ro(i,h);
    end
```

```
        if xp(i,h)>.25
            if xp(i,h)<.75
              if xo(i,h)>0
                magnitude(i,h)=ro(i,h)-.5;
              else
                 magnitude(i,h)=ro(i,h)+.5;
              end
            end
        end
        if xp(i,h)>.75
         magnitude(i,h)=ceil(magnitude(i,h));
        end
     end
 end
magnitude;


 %------- Phase decoder
outputp=[22*ones(n,1) round(63.*rand(n,7))];

%%%%%%%%%%%%%  Step 3 Clutster & Array
for i=1:n
comb(i,2:2:16)=outputp(i,:);
comb(i,1:2:16)=outputm(i,:);
end
comb;
da=comb';
da=da(:);

%%%%%%%%%%%% step 4 (a) converting the decimal number into 6 bit binar
number

for ff=1:length(da)
    dat(ff,:)= fliplr(dec2binvec(da(ff),6));
end
dat;
adf=dat';
data=adf(:);
ti=toc
```

MATLAB code to send the serial bit stream to the DAQ card

```
%%%%%%%%%%%% step 5 setting up the daq card and sending the data
tic;
dio=digitalio('nidaq','Dev5');    % setting up the daq card, Dev1 is the
name of the daq card used,
%%    **check on the device name befor running the code**
hline = addline(dio,1:4,'out',{'data';'Reg_en';'Ser_en';'clk'}); %
adding lines to port 0 to be used as
%%%% (port0.1 as data; port0.2 as register enable; port0.3 as serial
```

```
%%%% enable; port0.4 as clk)

for ii=1:96*n

        putvalue(dio,[data(ii) 0 1 0]);
        putvalue(dio,[data(ii) 0 1 1]);


                if rem(ii,96)== 0

                putvalue(dio.line([2 3 4]),[1 0 0]);
                putvalue(dio.line(4),1)
                end


end
toc;
```

MATLAB code to send ones and zeros to the beamformer board

```
%%%%%%%%%%%% FCS getting started
%%%%%%%% all the attenuators and phase shifters are set to zero

c=[zeros(8,1) ones(8,1) ];
zros_ones=[c c c c c c c c ];
zros_ones=zros_ones(:)
tic;
dio=digitalio('nidaq','Dev2');   % setting up the daq card, Dev1 is the
name of the daq card used,
%%    **check on the device name befor running the code**
hline = addline(dio,1:4,'out',{'data';'Reg_en';'Ser_en';'clk'}); %
adding lines to port 0 to be used as
%%%% (port0.1 as data; port0.2 as register enable; port0.3 as serial
%%%% enable; port0.4 as clk)

for g=1:96

        putvalue(dio,[zros_ones(g) 0 1 0]);
        putvalue(dio,[zros(g) 0 1 1]);
end
putvalue(dio.line([2 3 4]),[1 0 0]);
putvalue(dio.line(4),1)
toc;
```

MATLAB code to estimate the value of K

```
a=k1_2;
G_db=a/270-65;
% Greenstein paper
G=(1e-3)*(10.^((a/270-65)/10)); % linear power
Gm=mean(G);                     % First Moment
                                % mean value of the collected data

Gv=sqrt(sum((G-Gm).^2)/length(G));% second moment
V_sqr=sqrt((Gm^2)-(Gv^2));
sigma_sqr=Gm-V_sqr;
K=V_sqr/sigma_sqr                % K value
```

MATLAB code to plot the LCR

```
%%% To plot the LCR

no_levels=round((range(G_db))/2); % number of levels
levels=min(G_db):2:max(G_db);     % 2 db is the difference between
                                  % consective levels

for i= 1:no_levels
    counter=0;
    counter1=0;
    n=0;
    for ii=1:(length(G_db))-1
        if G_db(ii)> levels(i)
            if G_db(ii+1) < levels(i)
                counter=counter+1;
                n=n+1;
            end
        end
         if G_db(ii) < levels(i)
            if G_db(ii+1) > levels(i)
                counter=counter+1;
            else
                counter1 = counter1 + 1;
            end
         end
    end
    LCR(i)=counter/150;

end

mn=round(min(G_db))
mx=round(max(G_db))
rng=round(range(G_db))

    plot(LCR,'b','linewidth',2),grid on
```

```
   plot(levels(1:length(LCR)),LCR,'k','linewidth',2),grid on,hold on
    figure()
    plot(levels(1:length(AFD)),AFD,'r'),grid on
    title('Level Cross Rate for Different K Values (Run 5)')
     xlabel('Received Power (dBm)')
     ylabel('Level Cross Rate (1/sec)')
```

MATLAB GUI Code

```
function varargout = FS(varargin)
% FS M-file for FS.fig
%      FS, by itself, creates a new FS or raises the existing
%      singleton*.
%
%      H = FS returns the handle to a new FS or the handle to
%      the existing singleton*.
%
%      FS('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in FS.M with the given input arguments.
%
%      FS('Property','Value',...) creates a new FS or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before FS_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to FS_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 FS

% Last Modified by GUIDE v2.5 27-Jan-2010 15:05:51

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @FS_OpeningFcn, ...
                   'gui_OutputFcn',  @FS_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 FS is made visible.
function FS_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 FS (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = FS_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;



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (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 edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as
a double


% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (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 edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (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 edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as
a double


% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (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 edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (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 edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as
a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (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 edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (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 edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as
a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (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 edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (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 edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as
a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (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 edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (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 edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as
a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (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 edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (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 edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as
a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit6 (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 togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton1



function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (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 edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as
a double


% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (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 pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

season_factor = str2double(get(handles.edit1,'String'));
antenna_height= str2double(get(handles.edit3,'String'));
antenna_beamwidth= str2double(get(handles.edit4,'String'));
distance= str2double(get(handles.edit10,'String'));
Gamma = str2double(get(handles.edit5,'String'));
ko= str2double(get(handles.edit6,'String'));

K =
K_model(season_factor,antenna_height,antenna_beamwidth,distance,ko,Gamm
a);

K_val_db = 10*log10(K);

set(handles.text15,'String',num2str(K));
set(handles.text16,'String',num2str(K_val_db));
if K > 24
    errordlg('K has to be in the range of 0.5 to 24','ERROR','error')
end
if K < .24
    errordlg('K has to be in the range of 0.5 to 24','ERROR','error')
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

n = str2double(get(handles.edit9,'String'));


season_factor = str2double(get(handles.edit1,'String'));
antenna_height= str2double(get(handles.edit3,'String'));
antenna_beamwidth= str2double(get(handles.edit4,'String'));
distance= str2double(get(handles.edit10,'String'));
Gamma = str2double(get(handles.edit5,'String'));
ko= str2double(get(handles.edit6,'String'));

K =
K_model(season_factor,antenna_height,antenna_beamwidth,distance,ko,Gamm
a);
[data,ti] = Fade_simulator_serial_bitstream (K,n);
set(handles.text23,'String',num2str(ti));
[Tim,Tc] = send_bits(data,n);
set(handles.text14,'String',num2str(Tim));
set(handles.text13,'String',num2str(Tc));
msgbox('The serial bit stream was sent to the DAQ card','Success')
```

```
function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (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 edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10
as a double


% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (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 pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
K = str2double(get(handles.edit1,'String'));
n = str2double(get(handles.edit9,'String'));
[data,ti] = Fade_simulator_serial_bitstream (K,n);
set(handles.text23,'String',num2str(ti));
[Tim,Tc] = send_bits(data,n);
set(handles.text14,'String',num2str(Tim));
set(handles.text13,'String',num2str(Tc));
msgbox('The serial bit stream was sent to the DAQ card','Success')

function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (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 edit12 as text
%        str2double(get(hObject,'String')) returns contents of edit12
as a double


% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (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
```

MATLAB code to plot the throughput for the Airspan Radio downlink and uplink

```
% Airspan Radios downlink data
% To measure the downlink the throughput
% The BS is the client
% The SS is the server

input_SNR30=[0 10 20 30 40];
input_SNR15=[0 5 10 15];
% Without fade simulator at SNR = 30
thrput_max_SNR30_DL=[0 9.86 19.4 25.7 25.7];
% Without fade simulator at SNR = 15
thrput_max_SNR15_DL=[0 4.91 8.62 8.59];
%%% With fade simulator at SNR = 30dB
k1_SNR30_DL=[0 9.58 17.2 22.6 22.8];
k6_SNR30_DL=[0 9.64 17.1 25.5 25.5];
k10_SNR30_DL=[0 9.7 17.1 24.1 25.6];
k16_SNR30_DL=[0 9.76 19.3 25.6 25.5];
k24_SNR30_DL=[0 9.8 19.3 25.7 25.5];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SNR =15 dB
k1_SNR15_DL=[0 2.76 2.8 2.82];
k6_SNR15_DL=[0 2.8 5.66 5.7];
k10_SNR15_DL=[0 2.76 5.76 5.8];
k16_SNR15_DL=[0 2.78 6.2 6.44];
k24_SNR15_DL=[0 3.88 6.4 7.66];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(1)
plot(input_SNR30,k1_SNR30_DL,'linewidth',2)
hold on
plot(input_SNR30,k6_SNR30_DL,'r','linewidth',2)
```

```
hold on
plot(input_SNR30,k10_SNR30_DL,'g','linewidth',2)
hold on
plot(input_SNR30,k16_SNR30_DL,'c','linewidth',2)
hold on
plot(input_SNR30,k24_SNR30_DL,'k','linewidth',2)
hold on
plot(input_SNR30,thrput_max_SNR30_DL,'m','linewidth',2),grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure (2)
plot(input_SNR15,k1_SNR15_DL,'linewidth',2)
hold on
plot(input_SNR15,k6_SNR15_DL,'r','linewidth',2)
hold on
plot(input_SNR15,k10_SNR15_DL,'g','linewidth',2)
hold on
plot(input_SNR15,k16_SNR15_DL,'c','linewidth',2)
hold on
plot(input_SNR15,k24_SNR15_DL,'k','linewidth',2)
hold on
plot(input_SNR15,thrput_max_SNR15_DL,'m','linewidth',2),grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(3)
plot(input_SNR30,k1_SNR30_DL/max(thrput_max_SNR30_DL),'linewidth',2)
hold on
plot(input_SNR30,k6_SNR30_DL/max(thrput_max_SNR30_DL),'r','linewidth',2)
hold on
plot(input_SNR30,k10_SNR30_DL/max(thrput_max_SNR30_DL),'g','linewidth',
2)
hold on
plot(input_SNR30,k16_SNR30_DL/max(thrput_max_SNR30_DL),'c','linewidth',
2)
hold on
plot(input_SNR30,k24_SNR30_DL/max(thrput_max_SNR30_DL),'k','linewidth',
2),grid on
xlabel('Load')
ylabel('Throughput Ratio')
title('Airspan Downlink, SNR = 30')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(4)
plot(input_SNR15,k1_SNR15_DL/max(thrput_max_SNR15_DL),'linewidth',2)
hold on
plot(input_SNR15,k6_SNR15_DL/max(thrput_max_SNR15_DL),'r','linewidth',2)
hold on
plot(input_SNR15,k10_SNR15_DL/max(thrput_max_SNR15_DL),'g','linewidth',
2)
hold on
```

```
plot(input_SNR15,k16_SNR15_DL/max(thrput_max_SNR15_DL),'c','linewidth',
2)
hold on
plot(input_SNR15,k24_SNR15_DL/max(thrput_max_SNR15_DL),'k','linewidth',
2),grid on
xlabel('Load')
ylabel('Throughput Ratio')
title('Airspan Downlink, SNR = 15')


% Airspan Radios uplink data
% To measure the downlink the throughput
% The BS is the server
% The SS is the client

input_SNR30=[0 10 20 30 40];
input_SNR15=[0 5 10 15];
% Without fade simulator at SNR = 30
thrput_max_SNR30_UL=[0 9.72 19.2 20.9 20.8];
% Without fade simulator at SNR = 15
thrput_max_SNR15_UL=[0 4.8 9.39 9.43];

%%% With fade simulator at SNR = 30dB
k1_SNR30_UL=[0 8.1 14.4 16.7 16.4];
k6_SNR30_UL=[0 9.2 18.7 19.7 20];
k10_SNR30_UL=[0 9.3 18.4 19.4 20.2];
k16_SNR30_UL=[0 9.3 18.4 20.3 20.3];
k24_SNR30_UL=[0 9.4 18.6 20.7 20.9];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SNR =15 dB
k1_SNR15_UL=[0 3.83 5.72 5.19];
k6_SNR15_UL=[0 4.5 5.72 5.95];
k10_SNR15_UL=[0 4.56 6.28 6.44];
k16_SNR15_UL=[0 4.83 6.63 6.59];
k24_SNR15_UL=[0 4.82 6.9 6.85];
figure(1)
plot(input_SNR30,k1_SNR30_UL)
hold on
plot(input_SNR30,k6_SNR30_UL,'r')
hold on
plot(input_SNR30,k10_SNR30_UL,'g')
hold on
plot(input_SNR30,k16_SNR30_UL,'y')
hold on
plot(input_SNR30,k24_SNR30_UL,'m')
hold on
plot(input_SNR30,thrput_max_SNR30_UL,'k'),grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure (2)
```

```
plot(input_SNR15,k1_SNR15_UL)
hold on
plot(input_SNR15,k6_SNR15_UL,'r')
hold on
plot(input_SNR15,k10_SNR15_UL,'g')
hold on
plot(input_SNR15,k16_SNR15_UL,'y')
hold on
plot(input_SNR15,k24_SNR15_UL,'m')
hold on
plot(input_SNR15,thrput_max_SNR15_UL,'k'),grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3)
plot(input_SNR30,k1_SNR30_UL/max(thrput_max_SNR30_UL),'linewidth',2)
hold on
plot(input_SNR30,k6_SNR30_UL/max(thrput_max_SNR30_UL),'r','linewidth',2)
hold on
plot(input_SNR30,k10_SNR30_UL/max(thrput_max_SNR30_UL),'g','linewidth',
2)
hold on
plot(input_SNR30,k16_SNR30_UL/max(thrput_max_SNR30_UL),'c','linewidth',
2)
hold on
plot(input_SNR30,k24_SNR30_UL/max(thrput_max_SNR30_UL),'k','linewidth',
2),grid on
xlabel('Load (Mbps)')
ylabel('Throughput Ratio')
title('Airspan Uplink, SNR = 30')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(4)
plot(input_SNR15,k1_SNR15_UL/max(thrput_max_SNR15_UL),'linewidth',2)
hold on
plot(input_SNR15,k6_SNR15_UL/max(thrput_max_SNR15_UL),'r','linewidth',2)
hold on
plot(input_SNR15,k10_SNR15_UL/max(thrput_max_SNR15_UL),'g','linewidth',
2)
hold on
plot(input_SNR15,k16_SNR15_UL/max(thrput_max_SNR15_UL),'c','linewidth',
2)
hold on
plot(input_SNR15,k24_SNR15_UL/max(thrput_max_SNR15_UL),'k','linewidth',
2),grid on
xlabel('Load (Mbps)')
ylabel('Throughput Ratio')
title('Airspan Uplink, SNR = 15')
```

MATLAB code to plot the throughput for the Harris Radio downlink and uplink

```
% Harris Radios downlink data
% To measure the downlink the throughput
```

```
% The Master is the client
% The Slave is the server

input=[0 10 20 30 40 50];
input15=[0 10 20 30 40]
% Without fade simulator at SNR = 30
thrput_max_SNR30_DL=[0 9.73 19.3 29 34.4 34.5];
% Without fade simulator at SNR = 15
thrput_max_SNR15_DL=[0 9.69 16.8 17.4 17 17.2];

%%% With fade simulator at SNR = 30dB
k1_SNR30_DL=[0 9.68 19.8 28.4 29.2 29.5];
k6_SNR30_DL=[0 9.69 19.3 28.9 34.5 34.2];
k10_SNR30_DL=[0 9.67 19.4 28.9 34.7 34.8];
k16_SNR30_DL=[0 9.8 19.4 28.8 34.5 34.8];
k24_SNR30_DL=[0 9.65 19.4 28 34.9 35];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SNR =15 dB
k1_SNR15_DL=[0 9.26 10.4 9.64 9.76];
k6_SNR15_DL=[0 9.73 12.7 13 13.2];
k10_SNR15_DL=[0 9.7 17.3 17.4 17];
k16_SNR15_DL=[0 9.73 16.7 16.1 17];
k24_SNR15_DL=[0 9.73 16.5 16.9 16.9];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1)
plot(input,k1_SNR30_DL)
hold on
plot(input,k6_SNR30_DL,'r')
hold on
plot(input,k10_SNR30_DL,'g')
hold on
plot(input,k16_SNR30_DL,'y')
hold on
plot(input,k24_SNR30_DL,'m')
hold on
plot(input,thrput_max_SNR30_DL,'k'),grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (2)
plot(input15,k1_SNR15_DL)
hold on
plot(input15,k6_SNR15_DL,'r')
hold on
plot(input15,k10_SNR15_DL,'g')
hold on
plot(input15,k16_SNR15_DL,'y')
hold on
plot(input15,k24_SNR15_DL,'m')
hold on
plot(input,thrput_max_SNR15_DL,'k'),grid on
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3)
plot(input,k1_SNR30_DL/max(thrput_max_SNR30_DL),'linewidth',2)
hold on
plot(input,k6_SNR30_DL/max(thrput_max_SNR30_DL),'r','linewidth',2)
hold on
plot(input,k10_SNR30_DL/max(thrput_max_SNR30_DL),'g','linewidth',2)
hold on
plot(input,k16_SNR30_DL/max(thrput_max_SNR30_DL),'c','linewidth',2)
hold on
plot(input,k24_SNR30_DL/max(thrput_max_SNR30_DL),'k','linewidth',2),gri
d on
xlabel('Load (Mbps)')
ylabel('Throughput Ratio')
title('Harris Downlink, SNR = 30')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(4)
plot(input15,k1_SNR15_DL/max(thrput_max_SNR15_DL),'linewidth',2)
hold on
plot(input15,k6_SNR15_DL/max(thrput_max_SNR15_DL),'r','linewidth',2)
hold on
plot(input15,k10_SNR15_DL/max(thrput_max_SNR15_DL),'g','linewidth',2)
hold on
plot(input15,k16_SNR15_DL/max(thrput_max_SNR15_DL),'c','linewidth',2)
hold on
plot(input15,k24_SNR15_DL/max(thrput_max_SNR15_DL),'k','linewidth',2),g
rid on
xlabel('Load (Mbps)')
ylabel('Throughput Ratio')
title('Harris Downlink, SNR = 15')

% Harris Radios uplink data
% To measure the downlink the throughput
% The Master is the server
% The Slave is the client

input=[0 10 20 30 40 50];
input15=[0 10 20 30 40];
% Without fade simulator at SNR = 30
thrput_max_SNR30_UL=[0 9.58 19.5 28.7 32.7 32.7];
% Without fade simulator at SNR = 15
thrput_max_SNR15_UL=[0 9.75 15.8 15.6 15.8];
%%% With fade simulator at SNR = 30dB
k1_SNR30_UL=[0 9.71 19.2 21.6 21.7 21.7];
k6_SNR30_UL=[0 9.55 19.3 25.6 23.9 23.9];
k10_SNR30_UL=[0 9.59 19.2 23.9 25.3 25.3 ];
k16_SNR30_UL=[0 9.55 19.2 26.9 25.6 25.6];
k24_SNR30_UL=[0 9.58 19.3 25.7 26.9 26.9];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SNR =15 dB
k1_SNR15_UL=[0 3.57 3.9 4 4.02];
```

```
k6_SNR15_UL=[0 4.24 4.23 4.22 4.37];
k10_SNR15_UL=[0 6.26 6.09 6.14 6.14];
k16_SNR15_UL=[0 6.11 6.1 6.01 6.2];
k24_SNR15_UL=[0 6.57 5.91 6.44 6.35];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1)
plot(input,k1_SNR30_UL)
hold on
plot(input,k6_SNR30_UL,'r')
hold on
plot(input,k10_SNR30_UL,'g')
hold on
plot(input,k16_SNR30_UL,'y')
hold on
plot(input,k24_SNR30_UL,'m')
hold on
plot(input,thrput_max_SNR30_UL,'k'),grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (2)
plot(input15,k1_SNR15_UL)
hold on
plot(input15,k6_SNR15_UL,'r')
hold on
plot(input15,k10_SNR15_UL,'g')
hold on
plot(input15,k16_SNR15_UL,'y')
hold on
plot(input15,k24_SNR15_UL,'m')
hold on
plot(input15,thrput_max_SNR15_UL,'k'),grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(3)
plot(input,k1_SNR30_UL/max(thrput_max_SNR30_UL),'linewidth',2)
hold on
plot(input,k6_SNR30_UL/max(thrput_max_SNR30_UL),'r','linewidth',2)
hold on
plot(input,k10_SNR30_UL/max(thrput_max_SNR30_UL),'g','linewidth',2)
hold on
plot(input,k16_SNR30_UL/max(thrput_max_SNR30_UL),'c','linewidth',2)
hold on
plot(input,k24_SNR30_UL/max(thrput_max_SNR30_UL),'k','linewidth',2),gri
d on
xlabel('Load (Mbps)')
ylabel('Throughput Ratio')
title('Harris Uplink, SNR = 30')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(4)
plot(input15,k1_SNR15_UL/max(thrput_max_SNR15_UL),'linewidth',2)
```

```
hold on
plot(input15,k6_SNR15_UL/max(thrput_max_SNR15_UL),'r','linewidth',2)
hold on
plot(input15,k10_SNR15_UL/max(thrput_max_SNR15_UL),'g','linewidth',2)
hold on
plot(input15,k16_SNR15_UL/max(thrput_max_SNR15_UL),'c','linewidth',2)
hold on
plot(input15,k24_SNR15_UL/max(thrput_max_SNR15_UL),'k','linewidth',2),g
rid on
xlabel('Load (Mbps)')
ylabel('Throughput Ratio')
title('Harris Uplink, SNR = 15')
```

APPENDIX B

DATASHEETS

Datasheets for HMC425LP03 Attenuator

Hittite MICROWAVE CORPORATION v00.0202

HMC425LP3

0.5 dB LSB GaAs MMIC 6-BIT DIGITAL POSITIVE CONTROL ATTENUATOR, 2.4 - 8.0 GHz

Typical Applications

The HMC425LP3 is ideal for:

- WLAN & Point-to-Multi-Point
- Fiber Optics & Broadband Telecom
- Microwave Radio & VSAT
- Military

Features

0.5 dB LSB Steps to 31.5 dB

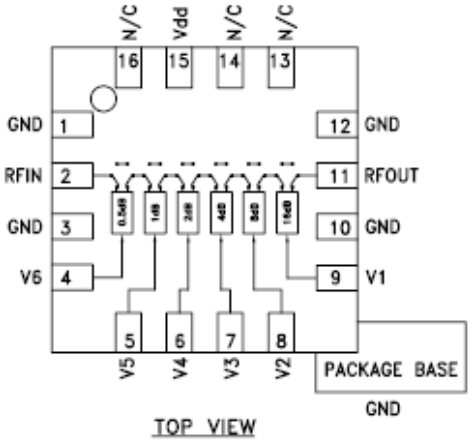
Single Control Line Per Bit

+/- 0.5 dB Typical Bit Error

Single +5V Supply

3 mm x 3 mm x 1 mm SMT Package

Functional Diagram



TOP VIEW

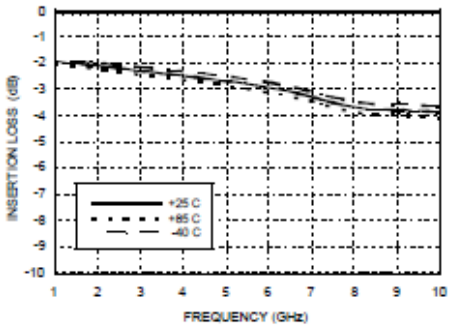
General Description

The HMC425LP3 is a broadband 6-bit GaAs IC digital attenuator in a low cost leadless surface mount package. Covering 2.4 to 8.0 GHz, the insertion loss is less then 3.8 dB typical. The attenuator bit values are 0.5 (LSB), 1, 2, 4, 8, and 16 dB for a total attenuation of 31.5 dB. Attenuation accuracy is excellent at ± 0.5 dB typical step error with an IIP3 of +40 dBm. Six control voltage inputs, toggled between 0 and +3 to +5V, are used to select each attenuation state. A single Vdd bias of +3 to +5V is required.

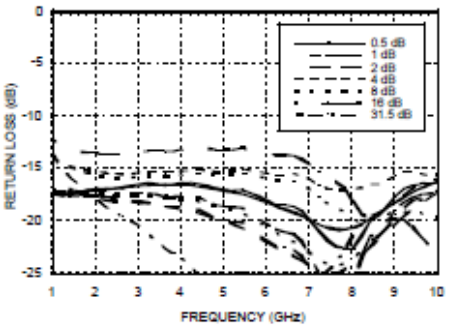
Electrical Specifications, T_A = +25° C, With Vdd = +5V & Vctl = 0/+5V (Unless Otherwise Noted)

Parameter		Frequency (GHz)	Min.	Typ.	Max.	Units
Insertion Loss		2.4 - 6.0 GHz 6.0 - 8.0 GHz		3.0 3.8	3.5 4.3	dB dB
Attenuation Range		2.4 - 8.0 GHz		31.5		dB
Return Loss (RF1 & RF2, All Atten. States)		2.4 - 8.0 GHz	11	15		dB
Attenuation Accuracy: (Referenced to Insertion Loss)	All States	2.4 - 8.0 GHz	± 0.5 + 5% of Atten. Setting Max.			dB
Input Power for 0.1 dB Compression	Vdd = 5V Vdd = 3V	2.4 - 8.0 GHz		22 19		dBm dBm
Input Third Order Intercept Point (Two-Tone Input Power= 0 dBm Each Tone)	REF - 16.0 dB States 16.5 - 31.5 dB States	2.4 - 8.0 GHz		45 35		dBm dBm
Switching Characteristics tRISE, tFALL (10/90% RF) tON, tOFF (50% CTL to 10/90% RF)		2.4 - 8.0 GHz		 160 180		 ns ns

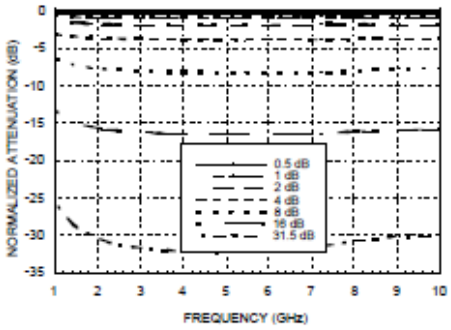
Insertion Loss



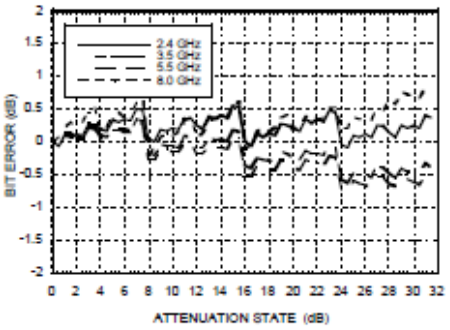
Return Loss RF1, RF2
(Only Major States are Shown)



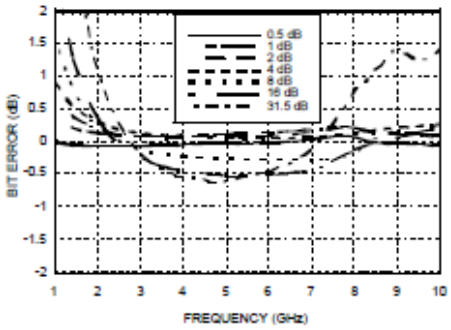
Normalized Attenuation
(Only Major States are Shown)



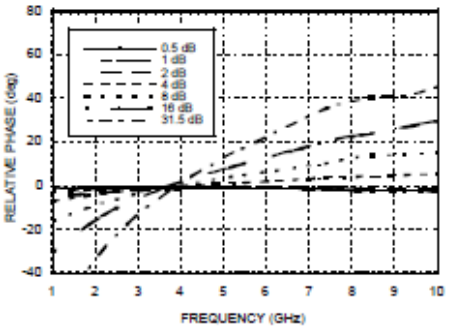
Bit Error vs. Attenuation State



Bit Error vs. Frequency
(Only Major States are Shown)



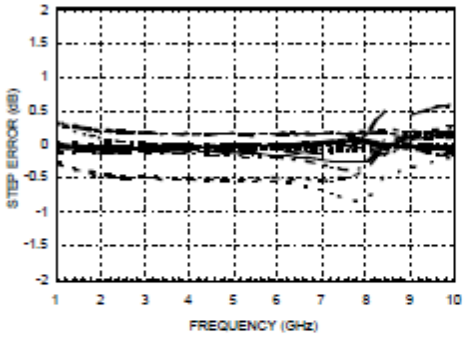
Relative Phase vs. Frequency
(Only Major States are Shown)



HMC425LP3

0.5 dB LSB GaAs MMIC 6-BIT DIGITAL POSITIVE CONTROL ATTENUATOR, 2.4 - 8.0 GHz

Worst Case Step Error Between Successive Attenuation States



Bias Voltage & Current

Vdd Range = 3.0 to +5.0 Vdc	
Vdd (VDC)	Idd (Typ.) (µA)
+3.0	10
+5.0	30

Control Voltage

State	Bias Condition
Low	0 to 0.2V @ 10 uA Typ.
High	Vdd ± 0.2V @ 5 uA Typ.
Note: Vdd = +3V to +5V	

Truth Table

Control Voltage Input						Attenuation State RF1 - RF2
V1 16 dB	V2 8 dB	V3 4 dB	V4 2 dB	V5 1 dB	V6 0.5 dB	
High	High	High	High	High	High	Reference I.L.
High	High	High	High	High	Low	0.5 dB
High	High	High	High	Low	High	1 dB
High	High	High	Low	High	High	2 dB
High	High	Low	High	High	High	4 dB
High	Low	High	High	High	High	8 dB
Low	High	High	High	High	High	16 dB
Low	Low	Low	Low	Low	Low	31.5 dB
Any combination of the above states will provide an attenuation approximately equal to the sum of the bits selected.						

Datasheets for MAPCGM002 Phase Shifter

tyco Electronics

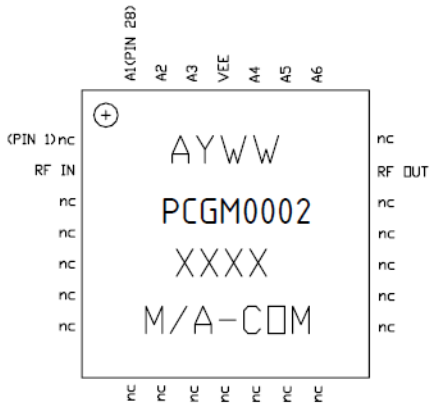
MACOM

Phase Shifter
3.5-6.0 GHz

MAPCGM0002
903214 —
Preliminary Information

Features

- 6 Bit Phase Shifter
- 360° Coverage, LSB = 5.6°
- TTL Control Inputs
- MSAG™ Process
- 6mm, 28 Lead, PQFN Package



Description

The MAPCGM0002 is a 6-bit Phase Shifter with Parallel TTL Input Control and is packaged in a micro lead package (MLP). This product is fully matched to 50 ohms on both the input and output. This part has 360° of phase coverage in 5.6° increments.

Fabricated using M/A-COM's repeatable, high performance and highly reliable GaAs Multifunction Self-Aligned Gate (MSAG™) Process, each device is 100% RF tested on wafer to ensure performance compliance.

M/A-COM's MSAG™ process features robust silicon-like manufacturing processes, planar processing of ion implanted transistors, multiple implant capability enabling power, low-noise, switch and digital FETs on a single chip, and polyimide scratch protection for ease of use with automated manufacturing processes. The use of refractory metals and the absence of platinum in the gate metal formulation prevents hydrogen poisoning when employed in hermetic packaging.

Primary Applications

- Satellite Communication
- Phased Array Radar

Absolute Maximum Conditions [1]

Parameter	Symbol	Absolute Maximum	Units
Input Power	P_{IN}	30	dBm
Digital Supply Voltage	V_{EE}	-6.0	V
Junction Temperature	T_J	180	°C
Storage Temperature	T_{STG}	-55 to +150	°C

1. Operation outside of these ranges may reduce product reliability. Operation at other than the typical values may result in performance outside the guaranteed limits.

Recommended Operating Conditions

Characteristic	Symbol	Min	Typ	Max	Unit
Control Voltage	A1 thru A6				
Logic High		2.8	5	5	V
Logic Low		0	0	0.8	V
Junction Temperature	T_J			150	°C
Digital Supply Voltage	V_{EE}	-5.2	-5.0	-4.8	V

tyco Electronics — MACOM

Phase Shifter
3.5-6.0 GHz

MAPCGM0002
903214 —
Preliminary Information

Electrical Characteristics: T_B = 25°C[2], Z_0 = 50Ω, V_{EE} = -5V

Parameter	Symbol	Typical	Units
Bandwidth	f	3.5-6.0	GHz
Insertion Loss	IL	6.5	dB
Input VSWR (At Reference)	VSWR	1.6:1	
Output VSWR (At Reference)	VSWR	1.7:1	
RMS Phase Error	RMS	9	°
RMS Phase Error — Calibrated	RMS	3	°
Phase Range	ΔΦ	360	°
Gain Variation over all Phase Shifter settings	ΔG	< 3	dB
Digital Supply Current	I_{EE}	< 10	mA
Input Third Order Intercept	ITOI	32	dBm
Input 1-dB Compression Point	P_{1dB}	26	dBm

2. T_B = Package Base Temperature

Truth Table[3]

Pin	Designation	Description	Level	State
22	A6	180° Phase Bit : MSB	Logic High	Phase Shift ≈ -180°
23	A5	90° Phase Bit	Logic High	Phase Shift ≈ -90°
24	A4	45° Phase Bit	Logic High	Phase Shift ≈ -45°
25	V_{EE}	DC Supply Voltage	-5V	ON
26	A3	22.5° Phase Bit	Logic High	Phase Shift ≈ -22.5°
27	A2	11.2° Phase Bit	Logic High	Phase Shift ≈ -11.2°
28	A1	5.6° Phase Bit : LSB	Logic High	Phase Shift ≈ -5.6°

3. All Phase Bits at Logic Low = Reference State.

Operating Instructions

This device is static sensitive. Please handle with care. To operate the device, follow these steps.

1. Apply V_{EE} = -5 V.
2. Apply Logic Voltages to control circuit as listed in Recommended Operating Conditions Table.
3. Power Down. Set V_{EE} = 0.

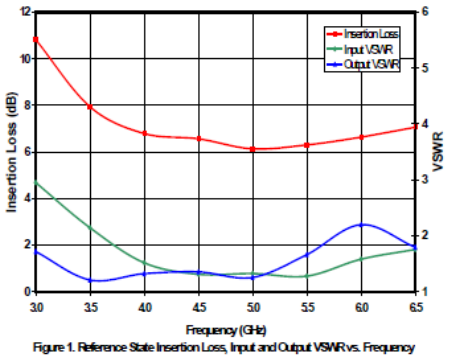


Figure 1. Reference State Insertion Loss, Input and Output VSWR vs. Frequency

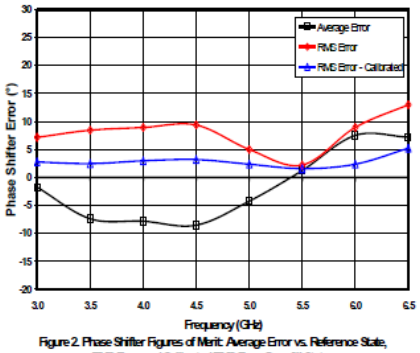


Figure 2. Phase Shifter Figures of Merit: Average Error vs. Reference State, RMS Error and Calibrated RMS Error Over All States

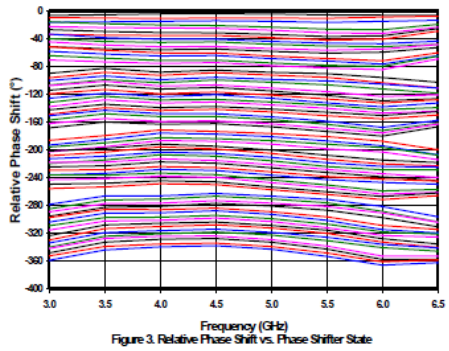


Figure 3. Relative Phase Shift vs. Phase Shifter State

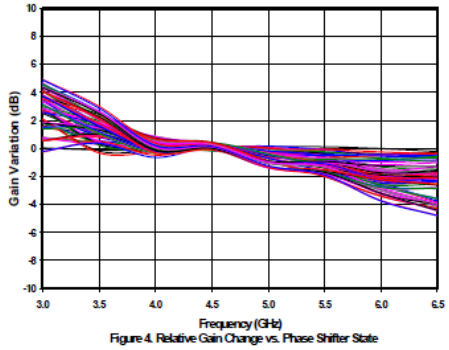


Figure 4. Relative Gain Change vs. Phase Shifter State

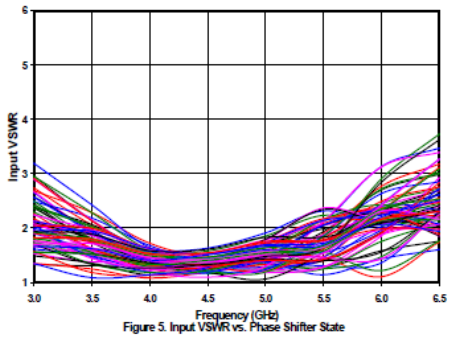


Figure 5. Input VSWR vs. Phase Shifter State

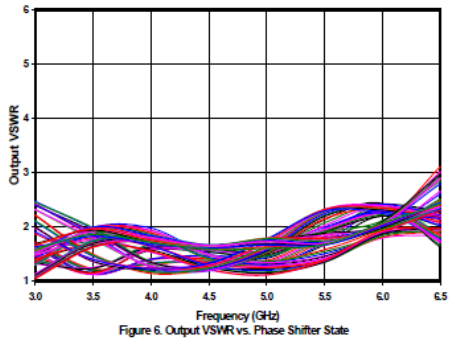


Figure 6. Output VSWR vs. Phase Shifter State

Datasheets for LMR-400 Cable

TIMES MICROWAVE SYSTEMS

LMR®-400
Flexible Low Loss Communications Coax

Ideal for...

- Drop-in replacement for RG-8/9913 Air-Dielectric type Cable
- Jumper Assemblies in Wireless Communications Systems
- Short Antenna Feeder runs
- Any application (e.g. WLL, GPS, LMR, WLAN, WISP, WiMax, SCADA, Mobile Antennas) requiring an easily routed, low loss RF cable

• **LMR®** standard is a UV Resistant Polyethylene jacketed cable designed for 20-year service outdoor use. The bending and handling characteristics are significantly better than air-dielectric and corrugated hard-line cables.

• **LMR®- DB** is identical to standard LMR plus has the advantage of being watertight. The addition of waterproofing compound in and around the foil/braid insures continuous reliable service should the jacket be inadvertently damaged during installation or in the future.

• **LMR®- FR** is a non-halogen (non-toxic), low smoke, fire retardant cable designed for in-building runs that can be routed anywhere except air handling plenums. LMR-FR has a UL/NEC & CSA rating of 'CMR' and 'FT4' respectively.

• **LMR®- FR-PVC** is a general-purpose indoor cable and has a UL/NEC & CSA rating of 'CMR' and 'FT4' respectively. It is less expensive than LMR-FR, however it emits toxic fumes (HCL) and greater smoke density when burned.

• **LMR®- PVC** is designed for low loss general-purpose indoor/outdoor applications and is somewhat more flexible than the standard polyethylene jacketed LMR.

• **LMR®- PVC-W** is a white-jacketed version of LMR-PVC for marine and other indoor/outdoor applications where color compatibility is desired.

• **Flexibility** and bendability are hallmarks of the LMR-400 cable design. The flexible outer conductor enables the tightest bend radius available for any cable of similar size and performance.

• **Low Loss** is another hallmark feature of LMR-400. Size for size LMR has the lowest loss of any flexible cable and comparable loss to semirigid hard-line cables.

• **RF Shielding** is 50 dB greater than typical single shielded coax (40 dB). The multi-ply bonded foil outer conductor is rated conservatively at > 90 dB (i.e. >180 dB between two adjacent cables).

• **Weatherability**: LMR-400 cables designed for outdoor exposure incorporate the best materials for UV resistance and have life expectancy in excess of 20 years.

• **Connectors**: A wide variety of connectors are available for LMR-400 cable, including all common interface types, reverse polarity, and a choice of solder or non-solder center pins. Most LMR connectors employ crimp outer attachment using standard hex crimp sizes.

• **Cable Assemblies**: All LMR-400 cable types are available as pre-terminated cable assemblies. Refer to the section on FlexTech for further details.

Part Description				Stock Code
Part No.	Application	Jacket	Color	
LMR-400	Outdoor	PE	Black	54001
LMR-400-DB	Outdoor/Watertight	PE	Black	54091
LMR-400-FR	Indoor -Riser CMR	FRPE	Black	54030
LMR-400-FR-PVC	Indoor -Riser CMR	FRPVC	Black	54073
LMR-400-PVC	Indoor/Outdoor	PVC	Black	54218
LMR-400-PVC-W	Indoor/Outdoor	PVC	White	54204

Construction Specifications			
Description	Material	In.	(mm)
Inner Conductor	Solid BCCAl	0.108	(2.74)
Dielectric	Foam PE	0.285	(7.24)
Outer Conductor	Aluminum Tape	0.291	(7.39)
Overall Braid	Tinned Copper	0.320	(8.13)
Jacket	(see table above)	0.405	(10.29)

TIMES MICROWAVE SYSTEMS

Mechanical Specifications

Performance Property	Units	US	(metric)
Bend Radius: installation	in. (mm)	1.00	(25.4)
Bend Radius: repeated	in. (mm)	4.0	(101.6)
Bending Moment	ft-lb (N-m)	0.5	(0.68)
Weight	lb/ft (kg/m)	0.068	(0.10)
Tensile Strength	lb (kg)	160	(72.6)
Flat Plate Crush	lb/in. (kg/mm)	40	(0.71)

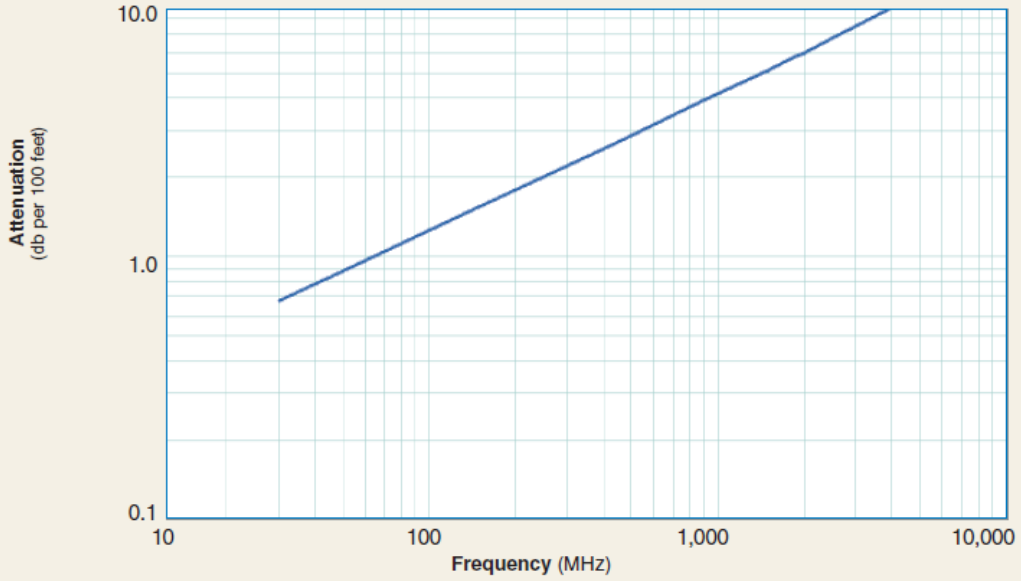
Environmental Specifications

Performance Property	°F	°C
Installation Temperature Range	-40/+185	-40/+85
Storage Temperature Range	-94/+185	-70/+85
Operating Temperature Range	-40/+185	-40/+85

Electrical Specifications

Performance Property	Units	US	(metric)
Cutoff Frequency	GHz	16.2	
Velocity of Propagation	%	85	
Dielectric Constant	NA	1.38	
Time Delay	nS/ft (nS/m)	1.20	(3.92)
Impedance	ohms	50	
Capacitance	pF/ft (pF/m)	23.9	(78.4)
Inductance	uH/ft (uH/m)	0.060	(0.20)
Shielding Effectiveness	dB	>90	
DC Resistance			
Inner Conductor	ohms/1000ft (/km)	1.39	(4.6)
Outer Conductor	ohms/1000ft (/km)	1.65	(5.4)
Voltage Withstand	Volts DC	2500	
Jacket Spark	Volts RMS	8000	
Peak Power	kW	16	

Attenuation vs. Frequency (typical)



Frequency (MHz)	30	50	150	220	450	900	1500	1800	2000	2500	5800
Attenuation dB/100 ft	0.7	0.9	1.5	1.9	2.7	3.9	5.1	5.7	6.0	6.8	10.8
Attenuation dB/100 m	2.2	2.9	5.0	6.1	8.9	12.8	16.8	18.6	19.6	22.2	35.5
Avg. Power kW	3.33	2.57	1.47	1.20	0.83	0.58	0.44	0.40	0.37	0.33	0.21

Calculate Attenuation =
$(0.122290) \bullet \sqrt{FMHz} + (0.000260) \bullet FMHz$ (interactive calculator available at http://www.timesmicrowave/telecom)

Attenuation:
VSWR=1.0 ; Ambient = +25°C (77°F)

Power:
VSWR=1.0; Ambient = +40°C; Inner Conductor = 100°C (212°F); Sea Level; dry air; atmospheric pressure; no solar loading

Datasheets for ZX60-6013E+ Amplifier

Connectorized

Amplifier

ZX60-6013E+

50Ω 20 MHz to 6 GHz

Features

- Wide Bandwidth, 20 MHz to 6 GHz
- Low Noise Figure, 3.3 dB Typ.
- Protected by US Patent 6,790,049

Applications

- Buffer Amplifier
- Cellular
- PCS
- Lab
- Instrumentation
- Test Equipment

CASE STYLE: GC957

Connectors	Model	Price	Qty.
SMA	ZX60-6013E-S+	$49.95 ea.	(1-9)

+ RoHS compliant in accordance with EU Directive (2002/95/EC)

The +Suffix has been added in order to identify RoHS Compliance. See our web site for RoHS Compliance methodologies and qualifications.

Electrical Specifications at T_{AMB} = 25°C

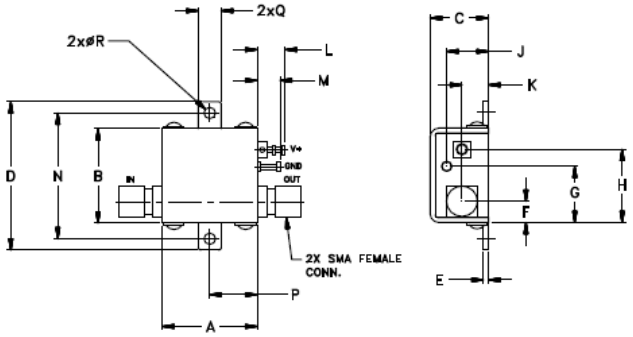
MODEL NO.	FREQ. (GHz) f_L - f_U	DC VOLTAGE @ Pin V+ (V)	GAIN over frequency in GHz Typ (dB) 0.1	1.0	2.0	3.0	4.0	5.0	6.0	Min.at 2 GHz	MAXIMUM POWER (dBm) Output (1 dB Comp.) Typ. f_L	f_U	DYNAMIC RANGE NF (dB) Typ.	IP3 (dBm) Typ.	VSWR (:1) Typ. In f_L-3 GHz	In 3-f_U GHz	Out f_L-3 GHz	Out 3-f_U GHz	ACTIVE DIRECTIVITY (dB) Isolation-Gain Typ.	DC OPERATING CURRENT @ Pin V+ (mA) Typ.	Max.
ZX60-6013E+	0.02-6	12.0	16.2	15.9	15.2	14.3	13.4	12.7	12.1	13.0	13.4	5.8	3.3	28.7	1.4	1.6	1.2	1.2	3-9	39	50

Maximum Ratings

Operating Temperature	-45°C to 80°C case
Storage Temperature	-55°C to 100°C
DC Voltage	12.5V
Input Power(no Damage)	15dBm
Power	650mW

Permanent damage may occur if any of these limits are exceeded.

Outline Drawing



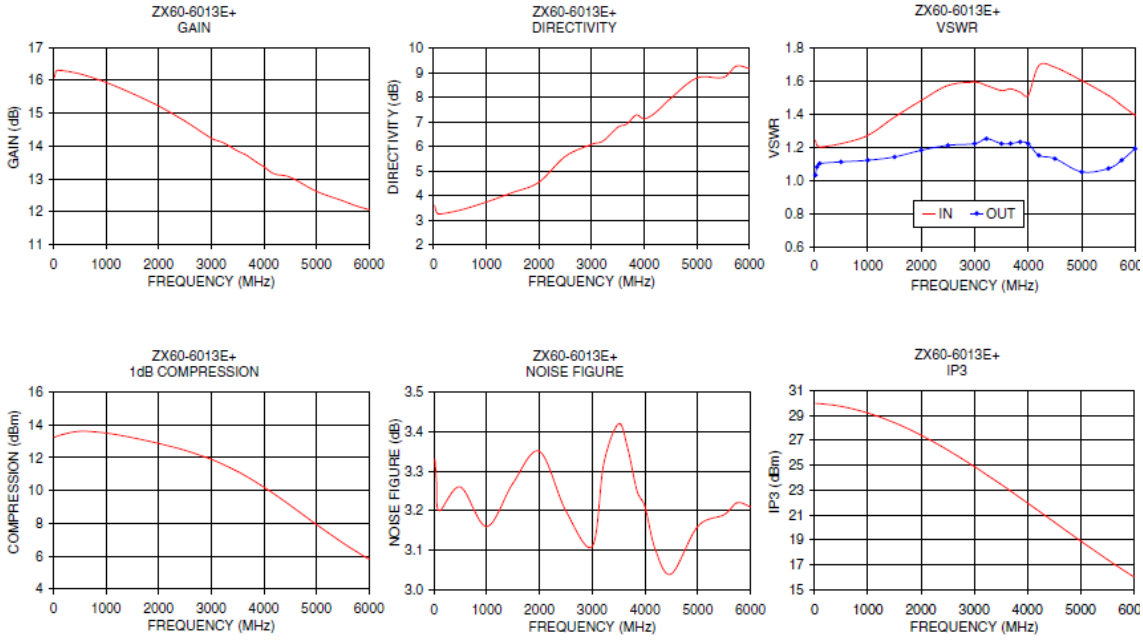
Outline Dimensions (inch / mm)

A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	WT.
.74	.75	.46	1.18	.04	.17	.45	.59	.33	.21	.22	.18	1.00	.37	.18	.106	GRAM
18.80	19.05	11.68	29.97	1.02	4.32	11.43	14.99	8.38	5.33	5.59	4.57	25.40	9.40	4.57	2.69	23.0

RE
M1
ED
ZX
RA
09
Pa

Typical Performance Data & Curves at 25°C ZX60-6013E+

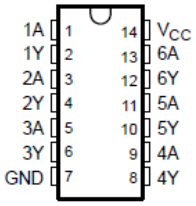
FREQUENCY (MHz)	GAIN (dB)	DIRECTIVITY (dB)	VSWR IN (:1)	VSWR OUT (:1)	POWER OUT @1dB COMPRESSION (dBm)	IP3 (dBm)	NF (dB)
20	16.09	3.60	1.24	1.03	13.24	29.97	3.33
50	16.28	3.33	1.21	1.08	13.28	29.96	3.27
100	16.31	3.24	1.20	1.10	13.35	29.95	3.20
500	16.20	3.40	1.22	1.11	13.61	29.73	3.26
1000	15.94	3.73	1.27	1.12	13.50	29.20	3.16
1500	15.60	4.12	1.38	1.14	13.22	28.42	3.27
2000	15.22	4.54	1.48	1.18	12.87	27.42	3.35
2500	14.76	5.59	1.57	1.21	12.45	26.23	3.20
3000	14.24	6.08	1.59	1.22	11.90	24.91	3.11
3220	14.11	6.22	1.57	1.25	11.60	24.28	3.32
3500	13.85	6.78	1.54	1.22	11.16	23.48	3.42
3670	13.72	6.91	1.55	1.22	10.86	22.97	3.36
3850	13.52	7.28	1.53	1.23	10.51	22.43	3.25
4000	13.37	7.13	1.51	1.22	10.21	21.98	3.21
4200	13.15	7.35	1.69	1.15	9.78	21.37	3.10
4500	13.05	7.95	1.68	1.13	9.10	20.45	3.04
5000	12.62	8.79	1.60	1.05	7.93	18.92	3.16
5500	12.33	8.81	1.51	1.07	6.81	17.44	3.19
5750	12.18	9.27	1.45	1.12	6.30	16.72	3.22
6000	12.06	9.18	1.39	1.19	5.81	16.04	3.21



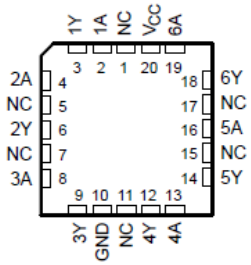
Datasheets for SN54HC04 Inverter

- Wide Operating Voltage Range of 2 V to 6 V
- Outputs Can Drive Up To 10 LSTTL Loads
- Low Power Consumption, 20-μA Max I_{CC}
- Typical t_{pd} = 8 ns
- ±4-mA Output Drive at 5 V
- Low Input Current of 1 μA Max

SN54HC04 . . . J OR W PACKAGE
SN74HC04 . . . D, N, NS, OR PW PACKAGE
(TOP VIEW)



SN54HC04 . . . FK PACKAGE
(TOP VIEW)



NC – No internal connection

description/ordering information

The '_HC04 devices contain six independent inverters. They perform the Boolean function $Y = \overline{A}$ in positive logic.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
–40°C to 85°C	PDIP – N	Tube of25	SN74HC04N	SN74HC04N
	SOIC – D	Tube of 50	SN74HC04D	HC04
		Reel of 2500	SN74HC04DR	
		Reel of 250	SN74HC04DT	
	SOP – NS	Reel of 2000	SN74HC04NSR	HC04
	TSSOP – PW	Tube of 90	SN74HC04PW	HC04
		Reel of 2000	SN74HC04PWR	
		Reel of 250	SN74HC04PWT	
–55°C to 125°C	CDIP – J	Tube of 25	SNJ54HC04J	SNJ54HC04J
	CFP – W	Tube of 150	SNJ54HC04W	SNJ54HC04W
	LCCC – FK	Tube of 55	SNJ54HC04FK	SNJ54HC04FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

FUNCTION TABLE
(each inverter)

INPUT A	OUTPUT Y
H	L
L	H

logic diagram (positive logic)

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{CC} . . . . . –0.5 V to 7 V
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1) . . . . . ±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) (see Note 1) . . . . . ±20 mA
Continuous output current, I_O ($V_O = 0$ to V_{CC}) . . . . . ±25 mA
Continuous current through V_{CC} or GND . . . . . ±50 mA
Package thermal impedance, θ_{JA} (see Note 2): D package . . . . . 86°C/W
N package . . . . . 80°C/W
NS package . . . . . 76°C/W
PW package . . . . . 113°C/W
Storage temperature range, T_{stg} . . . . . –65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions (see Note 3)

			SN54HC04			SN74HC04			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage		2	5	6	2	5	6	V
V_{IH}	High-level input voltage	V_{CC} = 2 V	1.5			1.5			V
		V_{CC} = 4.5 V	3.15			3.15			
		V_{CC} = 6 V	4.2			4.2			
V_{IL}	Low-level input voltage	V_{CC} = 2 V			0.5			0.5	V
		V_{CC} = 4.5 V			1.35			1.35	
		V_{CC} = 6 V			1.8			1.8	
V_I	Input voltage		0		V_{CC}	0		V_{CC}	V
V_O	Output voltage		0		V_{CC}	0		V_{CC}	V
$\Delta t/\Delta v$	Input transition rise/fall time	V_{CC} = 2 V			1000			1000	ns
		V_{CC} = 4.5 V			500			500	
		V_{CC} = 6 V			400			400	
T_A	Operating free-air temperature		–55		125	–40		85	°C

NOTE 3: All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

Datasheets for HMC336 MS8G Switch

MICROWAVE CORPORATION v03.0607

HMC336MS8G / 336MS8GE

GaAs MMIC SPDT NON-REFLECTIVE POSITIVE CONTROL SWITCH, DC* - 6 GHz

Typical Applications

This switch is suitable for usage in DC - 6.0 GHz 50-Ohm or 75-Ohm systems:

- Broadband
- Fiber Optics
- Switched Filter Banks
- Wireless below 6.0 GHz

Features

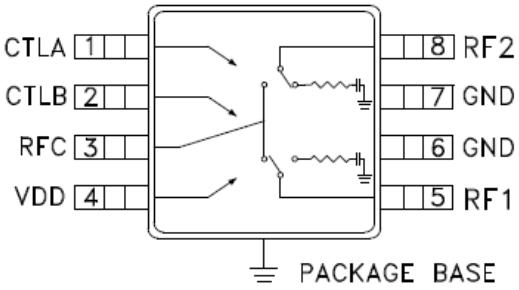
Broadband Performance: DC - 6 GHz

High Isolation: 42 dB@ 6 GHz

Low Insertion Loss: 1.6 dB@ 6 GHz

MSOP8G SMT Package

Functional Diagram



General Description

The HMC336MS8G & HMC336MS8GE are broadband non-reflective GaAs MESFET SPDT switches in low cost 8 lead MSOP8G surface mount packages with an exposed ground paddle. Covering DC to 6 GHz, this switch offers high isolation and low insertion loss. The switch operates using a positive control voltage of 0/+5 Volts, and requires a fixed bias of +5V. This switch is suitable for usage in 50-Ohm or 75-Ohm systems.

Electrical Specifications, T_A = +25° C, With 0/+5V Control, 50 Ohm System

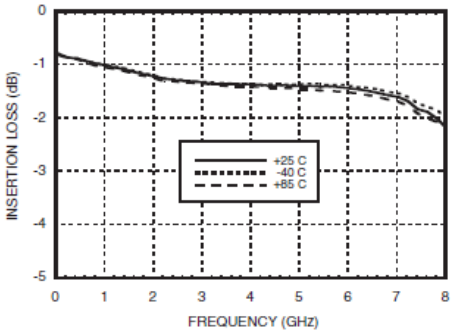
Parameter		Frequency	Min.	Typ.	Max.	Units
Insertion Loss		DC - 2.0 GHz DC - 4.0 GHz DC - 6.0 GHz		1.2 1.4 1.6	1.6 1.8 2.0	dB dB dB
Isolation		DC - 2.0 GHz DC - 4.0 GHz DC - 6.0 GHz	42 39 37	47 44 42		dB dB dB
Return Loss	"On State"	DC - 2.0 GHz DC - 6.0 GHz	9 6	12 9		dB dB
Return Loss (RF1, RF2)	"Off State"	2.0 - 6.0 GHz	13	18		dB
Input Power for 1 dB Compression		0.5 - 6.0 GHz	20	25		dBm
Input Third Order Intercept (Two-Tone Input Power = +7 dBm Each Tone, 1 MHz Tone Spacing)		0.5 - 6.0 GHz	37	42		dBm
Switching Characteristics tRISE, tFALL (10/90% RF) tON, tOFF (50% CTL to 10/90% RF)		DC - 6.0 GHz		 8 20		 ns ns

** DC blocking capacitors are required at ports RFC, RF1 and RF2. Their value will determine the lowest transmission frequency.*

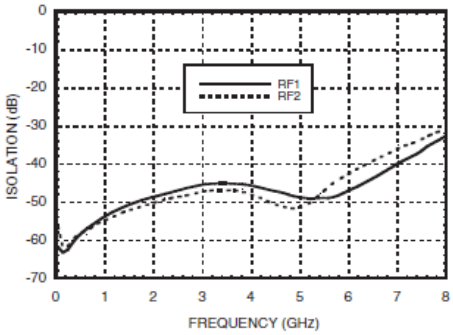
HMC336MS8G / 336MS8GE

GaAs MMIC SPDT NON-REFLECTIVE POSITIVE CONTROL SWITCH, DC* - 6 GHz

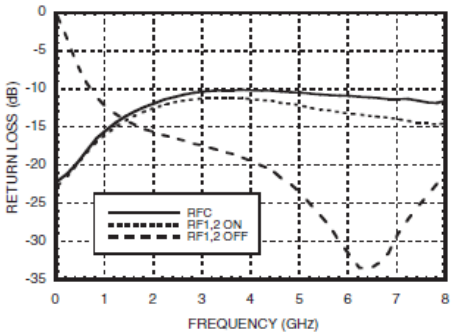
Insertion Loss vs. Temperature



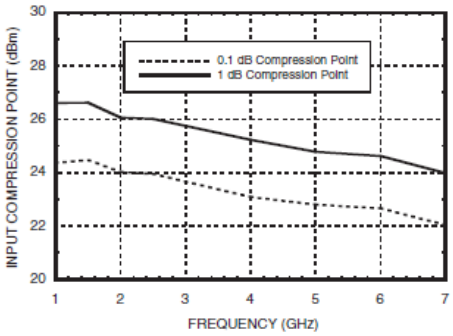
Isolation



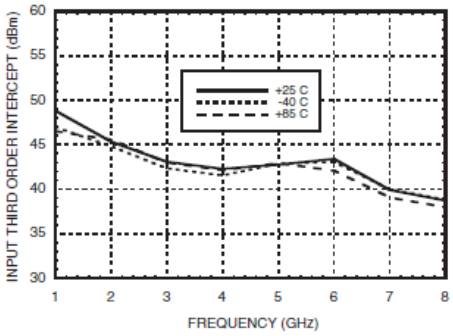
Return Loss



0.1 and 1 dB Input Compression Point



Input Third Order Intercept Point



Datasheets for ZB8PD-6.4 Power Combiner/Divider

Coaxial

Power Splitter/Combiner

ZB8PD-6.4

8 Way-0° 50Ω 5600 to 6800 MHz

CASE STYLE: Z41

Connectors	Model	Price	Qty.
SMA	ZB8PD-6.4-S	$138.95	(1-9)

Maximum Ratings

Operating Temperature	-55°C to 100°C
Storage Temperature	-55°C to 100°C
Power Input (as a splitter)	10W max.
Internal Dissipation	0.875W max.

Permanent damage may occur if any of these limits are exceeded.

Coaxial Connections

SUM PORT	S
PORT 1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8

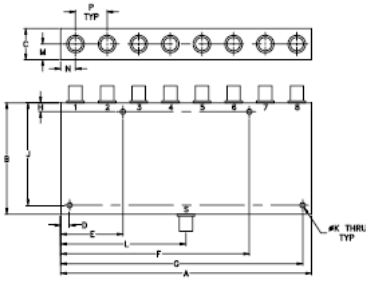
Features

- wideband, 5600 to 6800 MHz
- low insertion loss 0.8 dB typ.
- good isolation, 23 dB typ.
- up to 10 power input
- rugged shielded case

Applications

- SHF
- defense communication
- cable TV relay
- instrumentation

Outline Drawing



Outline Dimensions (inch / mm)

A	B	C	D	E	F	G	H
7.06	3.13	.88	.250	1.750	5.310	6.810	.250
179.32	79.50	22.35	6.35	44.45	134.87	172.97	6.35

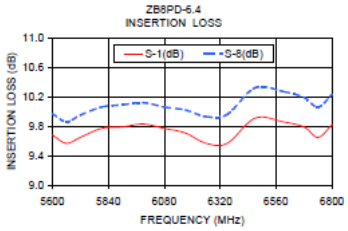
J	K	L	M	N	P	wt
2.875	.144	3.53	.44	.415	.89	grams
73.03	3.66	89.66	11.18	10.54	22.61	800

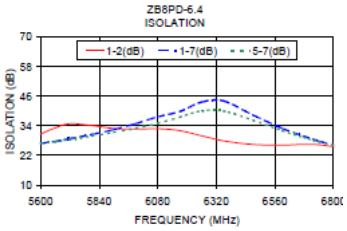
Electrical Specifications

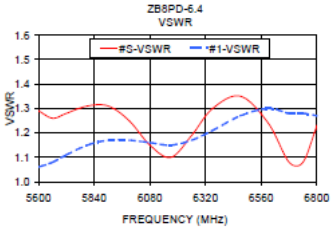
FREQ. RANGE (MHz)	ISOLATION (dB)		INSERTION LOSS (dB) ABOVE 9.0 dB		PHASE UNBALANCE (Degrees)	AMPLITUDE UNBALANCE (dB)
f_L-f_U	Typ.	Min.	Typ.	Max.	Max.	Max.
5600-6800	23	18	0.8	1.7	15	0.7

Typical Performance Data

Frequency (MHz)	Insertion Loss (dB)						Amplitude Unbalance (dB)		Isolation (dB)			VSWR S	VSWR 1	VS
	S-1	S-2	S-3	S-4	S-6	S-8			1-2	1-7	3-4	5-7			
5600.00	9.68	9.81	9.83	9.79	9.84	9.97	0.29		30.74	26.92	31.08	26.94	1.29	1.06	1.0
5660.00	9.57	9.70	9.72	9.70	9.72	9.86	0.29		33.18	27.70	33.96	27.55	1.26	1.08	1.1
5720.00	9.65	9.75	9.79	9.78	9.77	9.95	0.30		34.84	28.81	36.63	28.48	1.28	1.11	1.1
5810.00	9.77	9.85	9.85	9.89	9.86	10.06	0.29		34.09	30.51	36.68	29.84	1.31	1.15	1.1
5900.00	9.79	9.88	9.85	9.93	9.86	10.09	0.30		32.95	32.49	34.69	31.39	1.31	1.17	1.1
5990.00	9.83	9.93	9.87	10.00	9.88	10.12	0.30		32.66	34.91	33.36	33.05	1.25	1.17	1.1
6080.00	9.77	9.89	9.84	9.96	9.79	10.06	0.29		32.84	37.66	32.30	35.02	1.15	1.16	1.1
6170.00	9.71	9.87	9.80	9.92	9.73	10.02	0.31		32.05	39.79	30.84	37.54	1.10	1.15	1.1
6260.00	9.57	9.69	9.65	9.80	9.59	9.93	0.36		29.92	43.65	28.77	40.11	1.19	1.17	1.1
6350.00	9.57	9.66	9.64	9.80	9.59	9.96	0.39		27.90	44.08	27.26	39.95	1.30	1.21	1.1
6470.00	9.91	9.98	9.97	10.14	9.90	10.32	0.42		26.45	38.35	26.27	36.28	1.35	1.27	1.2
6590.00	9.86	9.92	9.91	10.04	9.86	10.27	0.42		26.19	32.97	26.42	32.02	1.24	1.30	1.2
6680.00	9.79	9.83	9.80	9.92	9.77	10.19	0.41		26.51	29.69	26.91	29.22	1.08	1.28	1.2
6740.00	9.65	9.74	9.68	9.80	9.65	10.06	0.41		26.33	27.82	26.64	27.52	1.08	1.28	1.2
6800.00	9.82	9.87	9.81	9.90	9.80	10.23	0.43		25.49	26.22	25.51	26.12	1.23	1.27	1.2







electrical schematic

REV. OR
M94964
ZB8PD-6.4
ED-5337
HY/TD/CP/
090827

APPENDIX C

LABVIEW PROGRAM TO COLLECT THE MEASUREMENTS FROM THE SPECTRUM ANALYZER

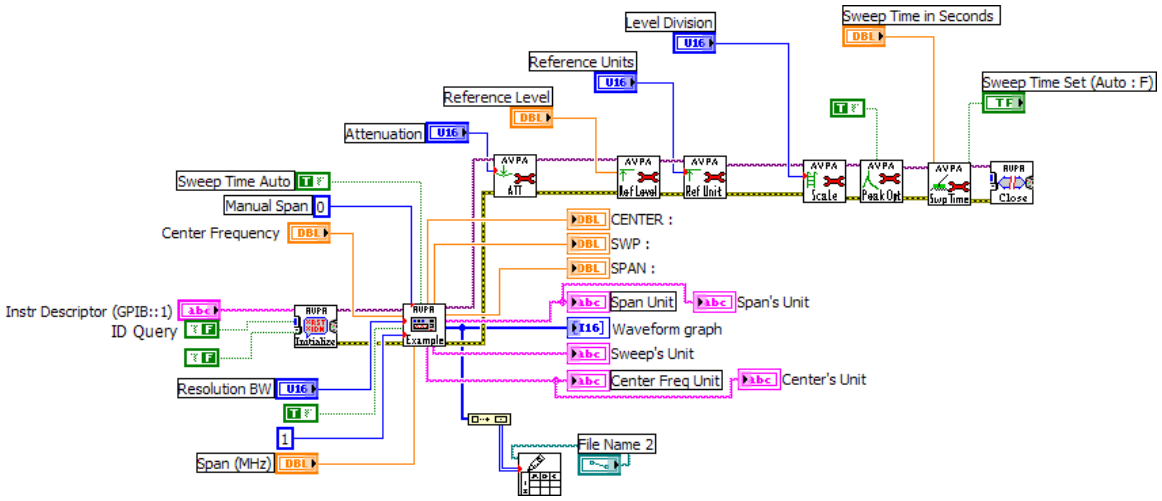


Figure C-1: LabView program used to collect the spectrum analyzer measurements

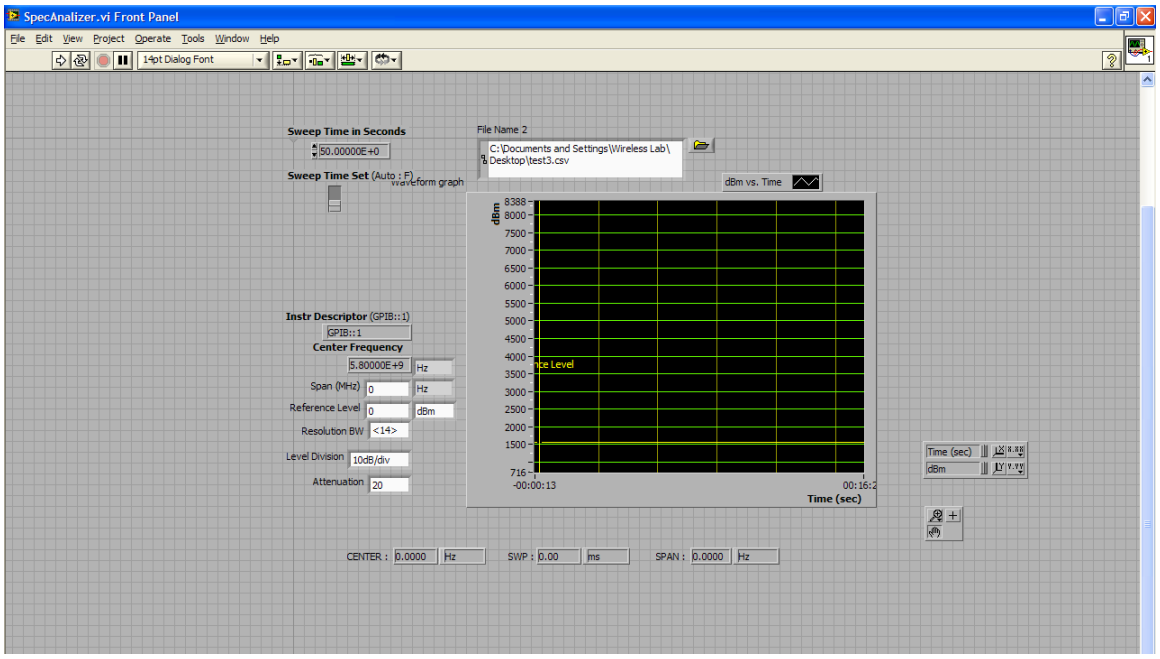


Figure C-2: frontend LabView program used to collect the spectrum analyzer measurements

APPENDIX D

COLLECTED DATA PRCOESSING

Power versus time measurements for different *K* values for the other four runs

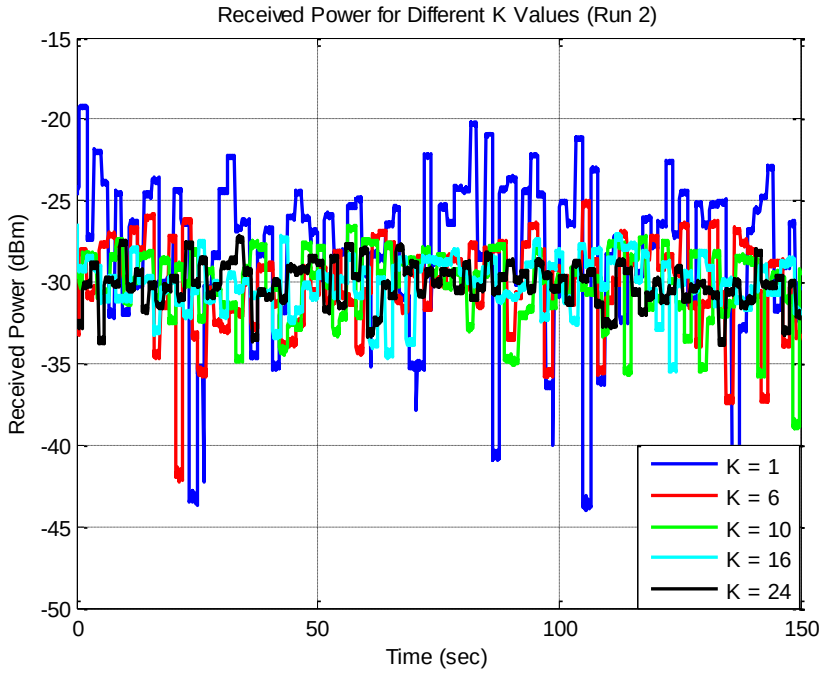


Figure D-1: Collected power versus time measurements for different *K* values (Run 2)

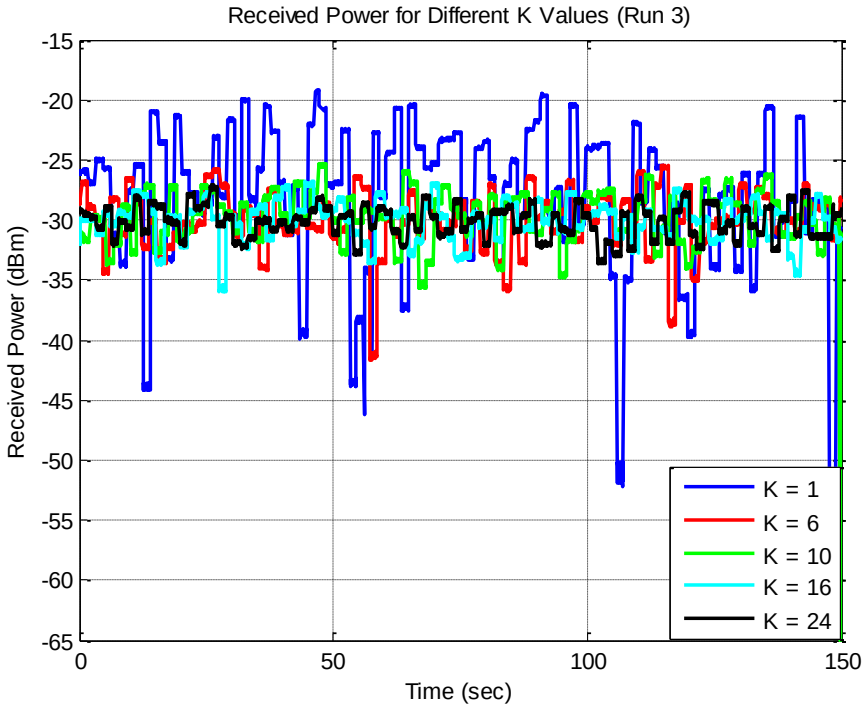


Figure D-2: Collected power versus time measurements for different *K* values (Run 3)

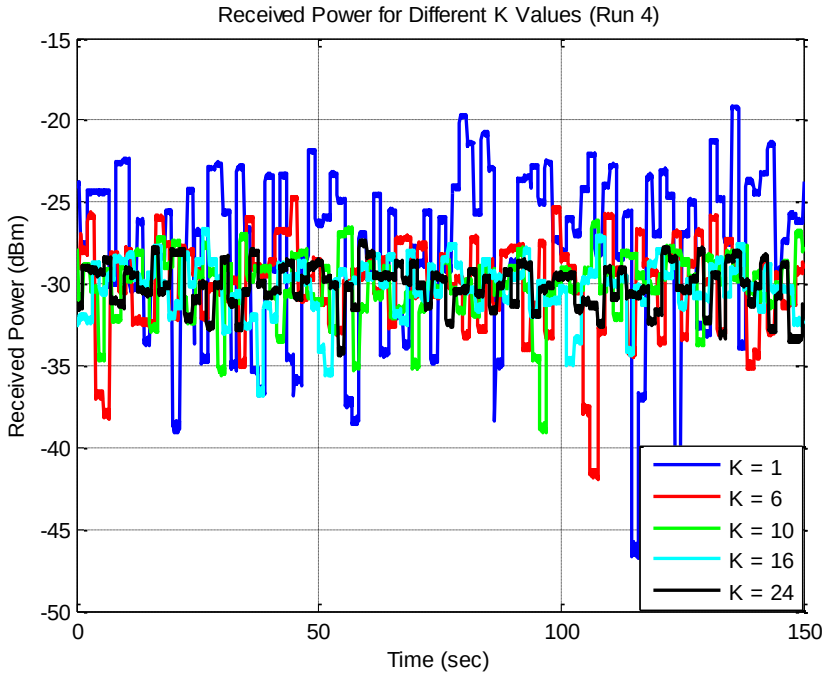


Figure D-3: Collected power versus time measurements for different *K* values (Run 4)

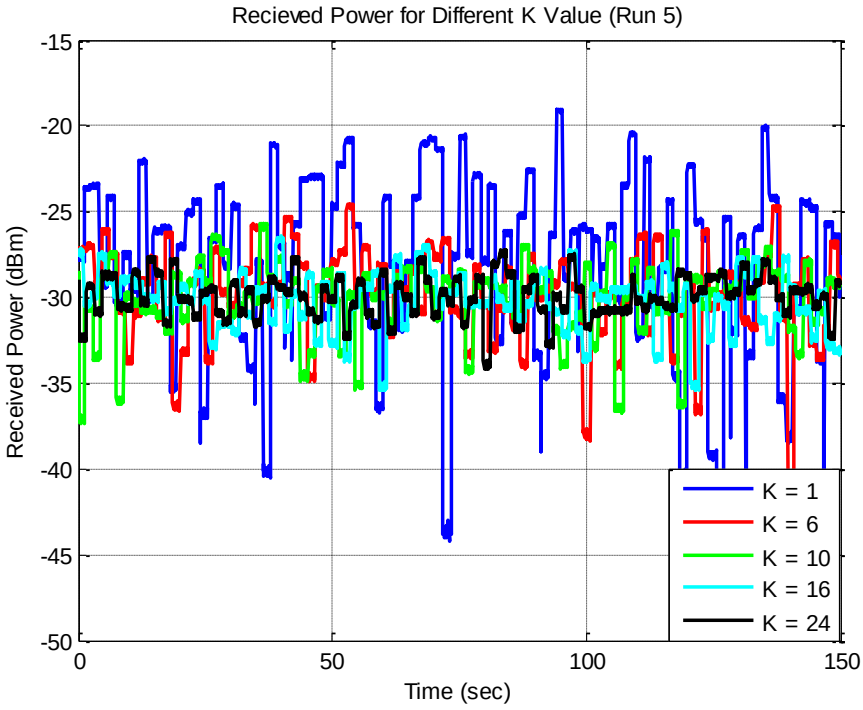


Figure D-4: Collected power versus time measurements for different *K* values (Run 5)

PDF of the received power for different *K* values for the other four runs

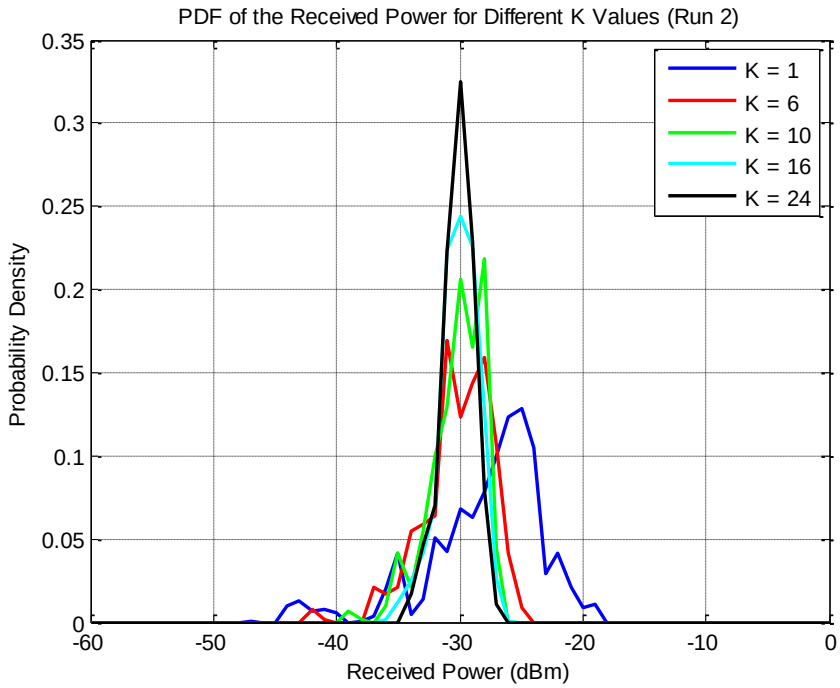


Figure D-5: PDF of the received power for different *K* values (Run 2)

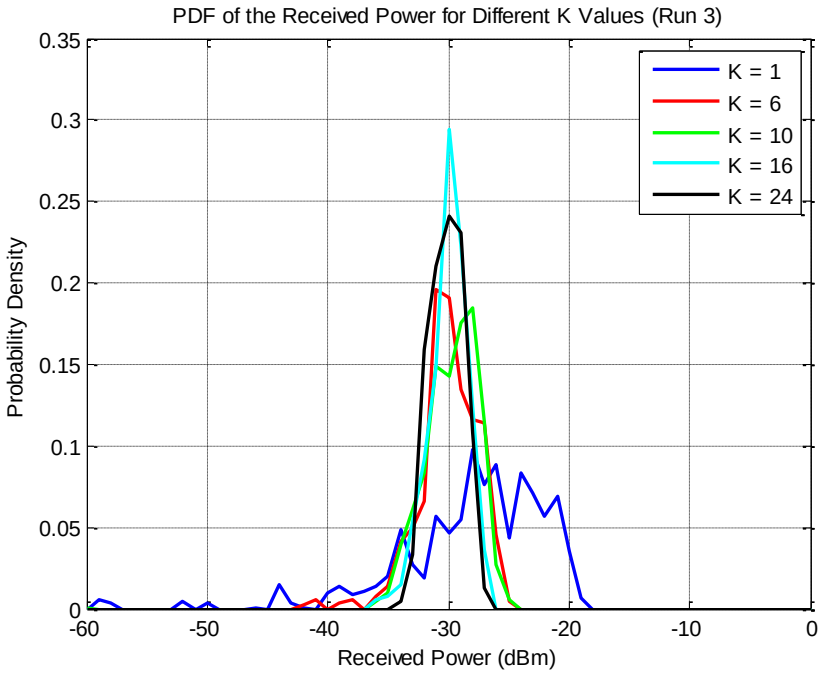


Figure D-6: PDF of the received power for different *K* values (Run 3)

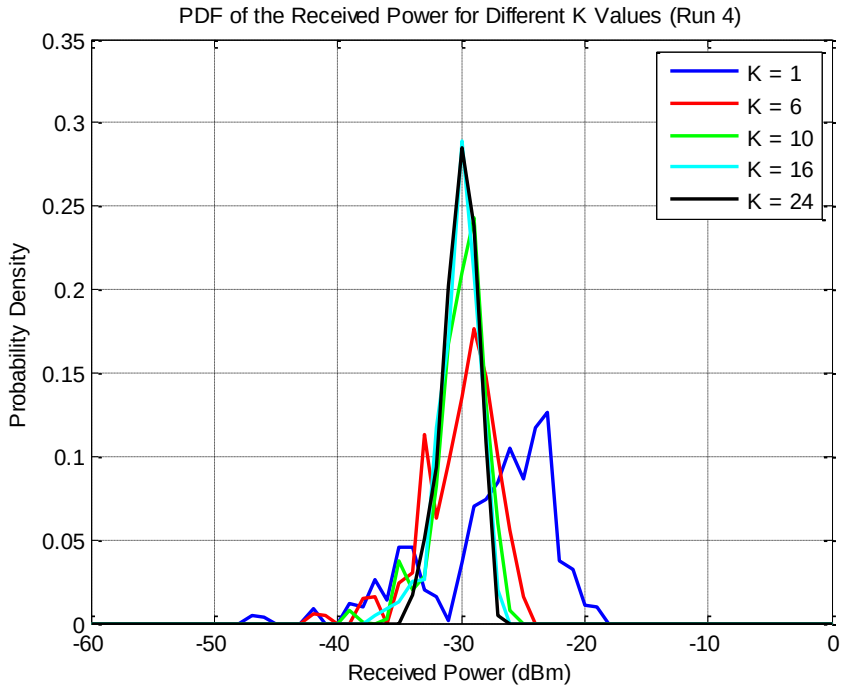


Figure D-7: PDF of the received power for different *K* values (Run 4)

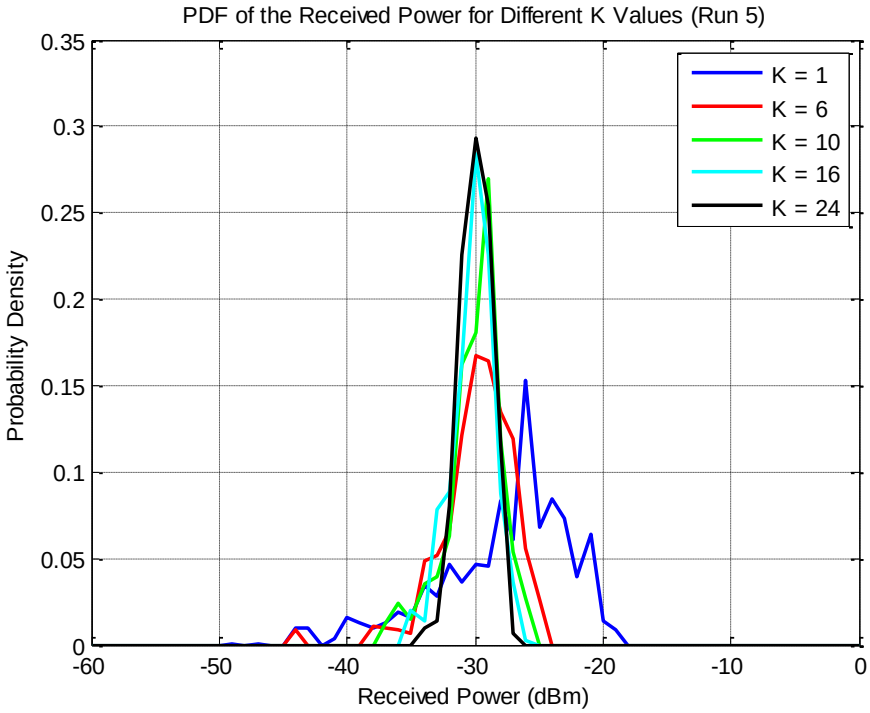


Figure D-8: PDF of the received power for different *K* values (Run 5)

CDF of the received power for different *K* values for the other four runs

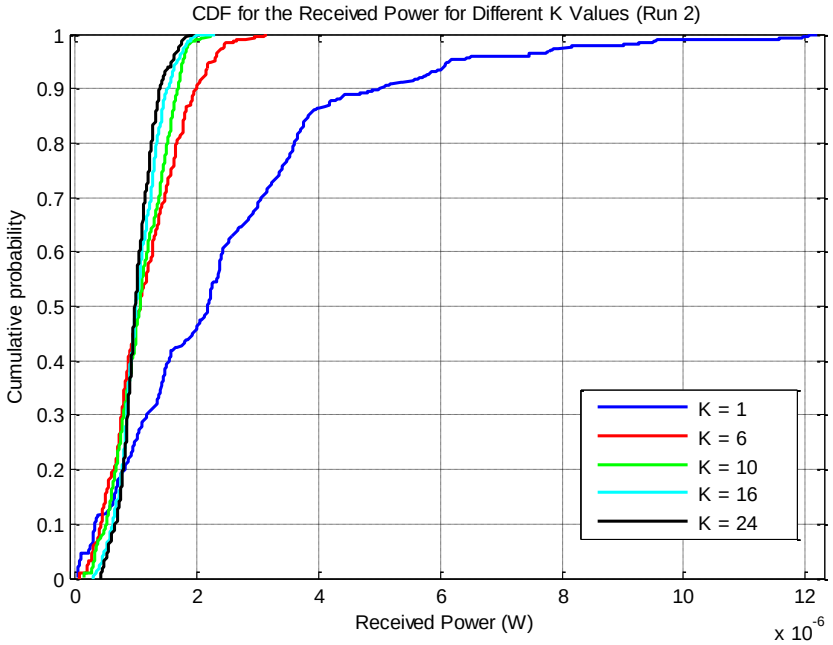


Figure D-9: CDF of the received power for different *K* values (Run 2)

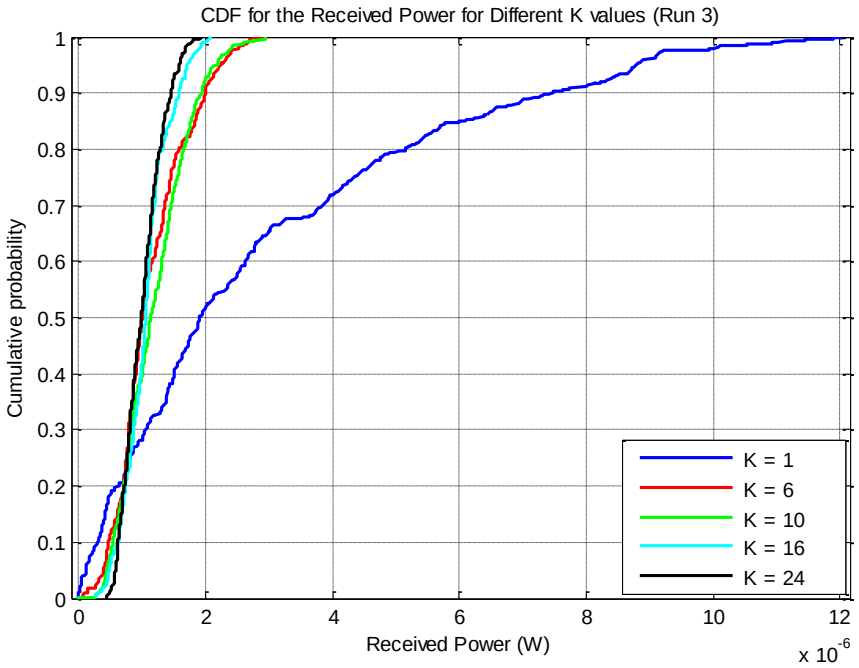


Figure D-10: CDF of the received power for different *K* values (Run 3)

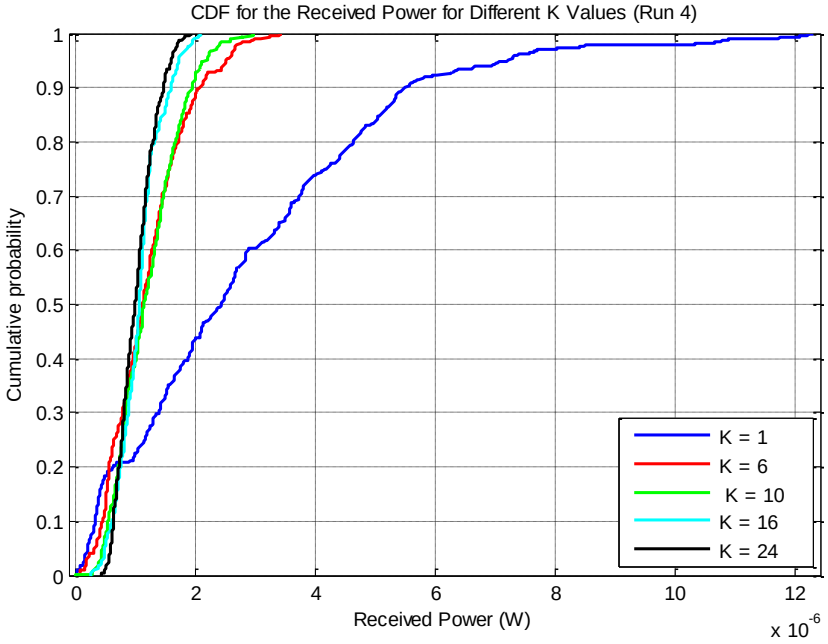


Figure D-10: CDF of the received power for different *K* values (Run 4)

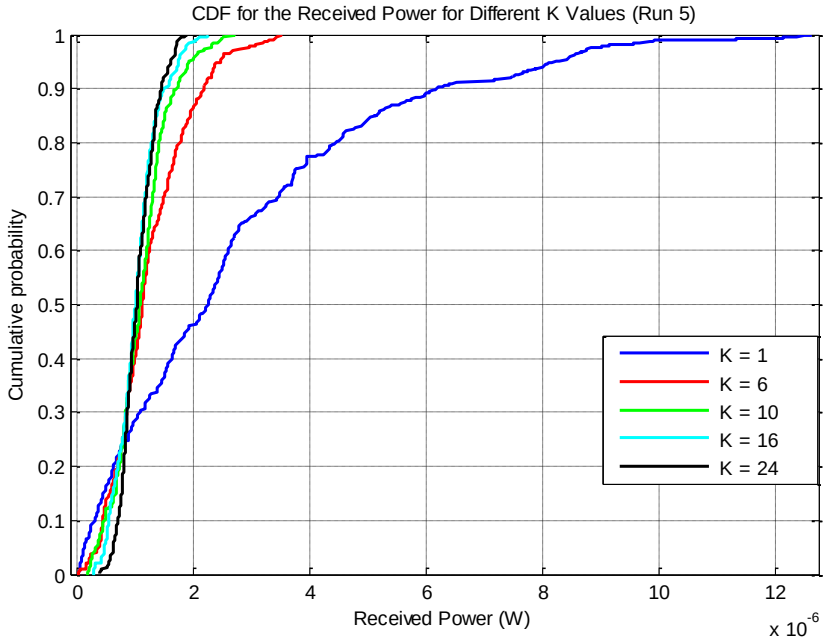


Figure D-11: CDF of the received power for different *K* values (Run 5)

Ricean Fit of the CDF of the received power versus time measurements for the other four runs

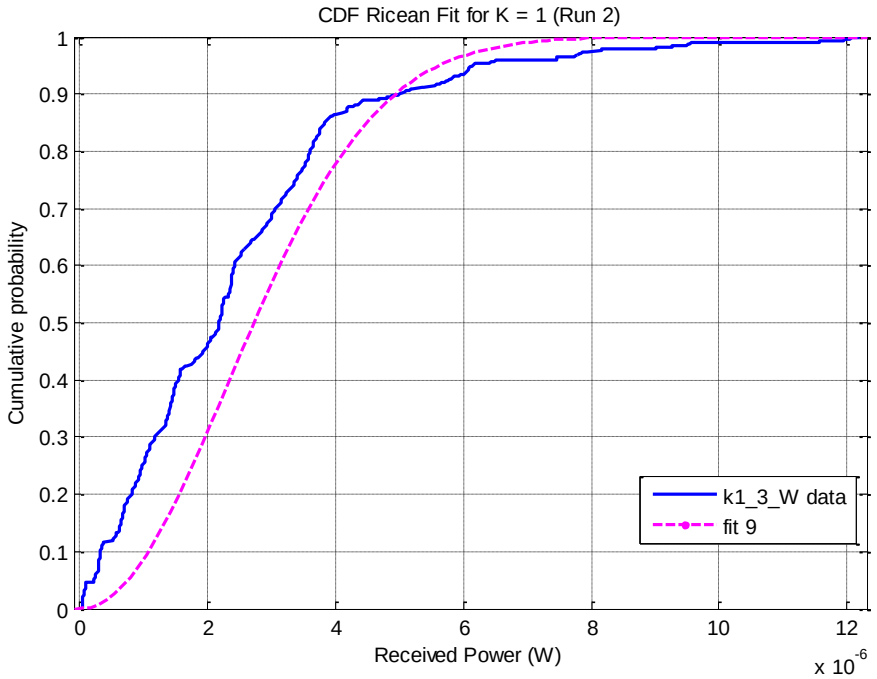


Figure D-12: CDF Ricean fit of the received power for K = 1 (Run 2)

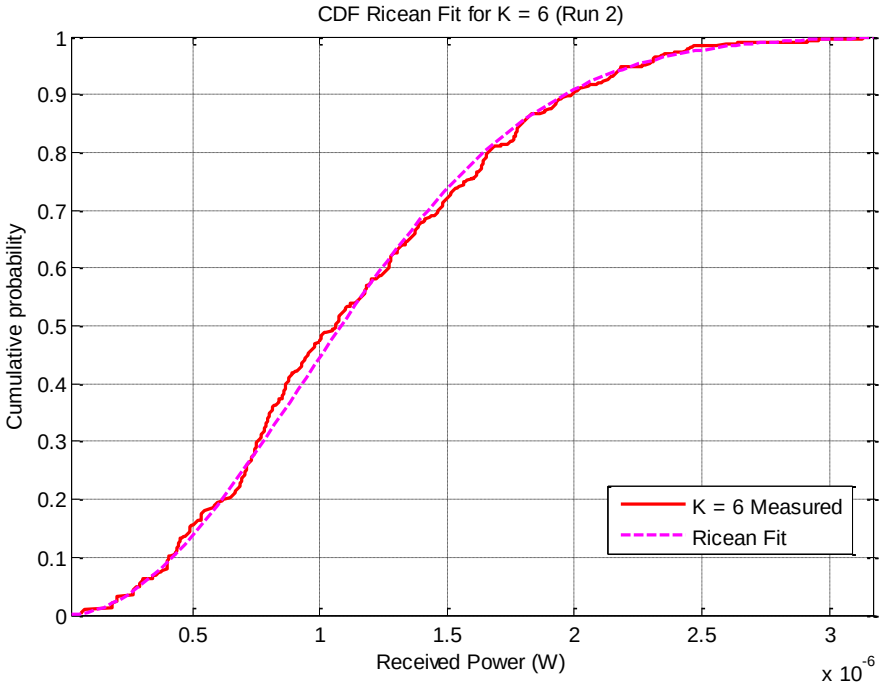


Figure D-13: CDF Ricean fit of the received power for K = 6 (Run 2)

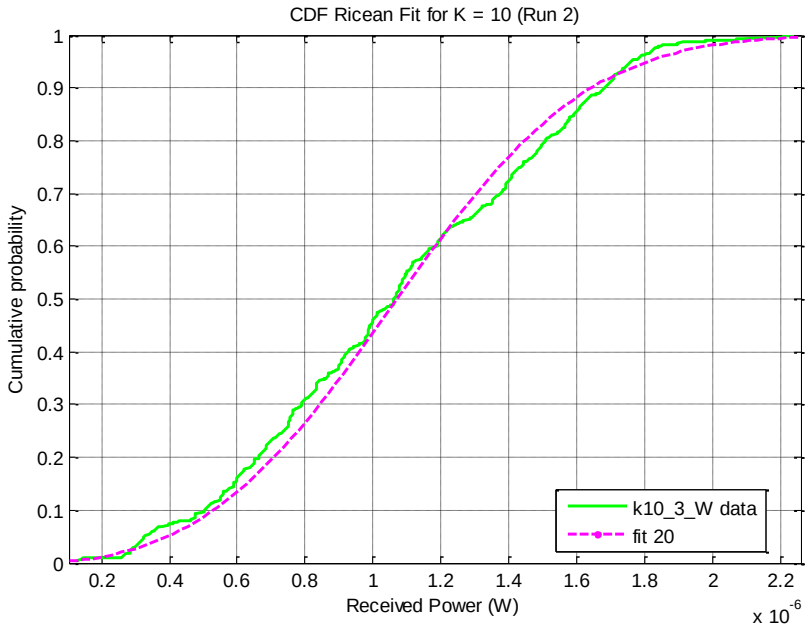


Figure D-14: CDF Ricean fit of the received power for K = 10 (Run 2)

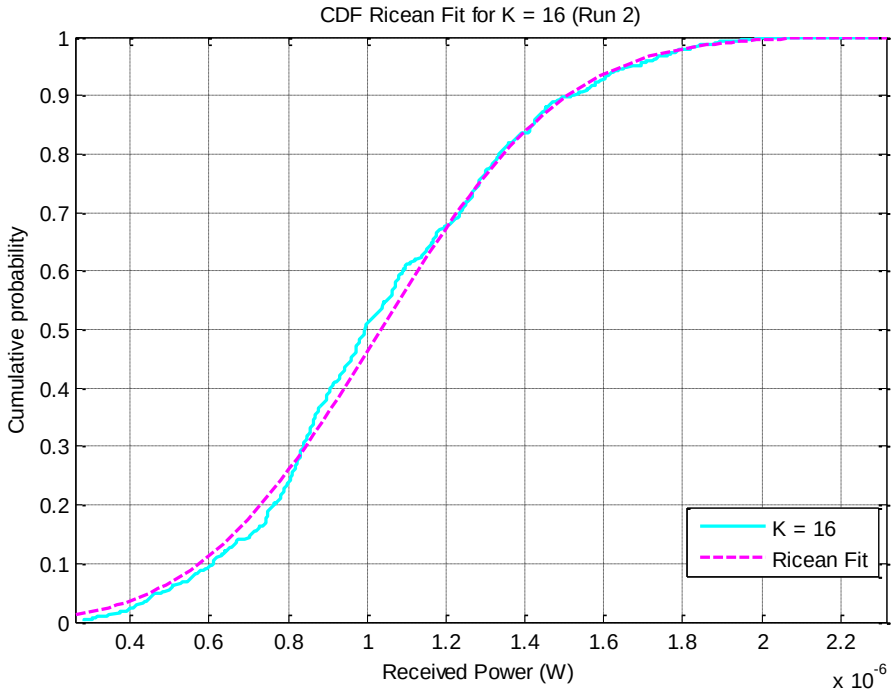


Figure D-15: CDF Ricean fit of the received power for K = 16 (Run 2)

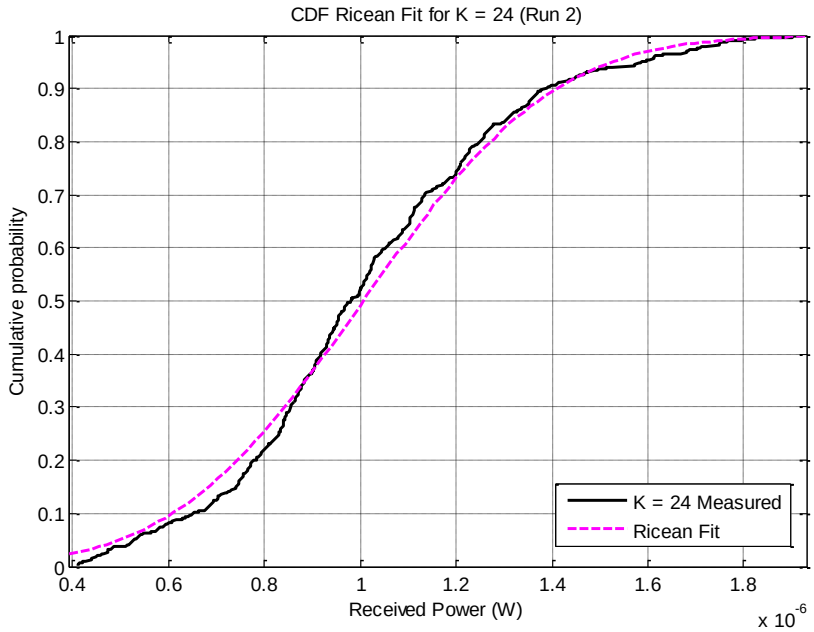


Figure D-16: CDF Ricean fit of the received power for K = 24 (Run 2)

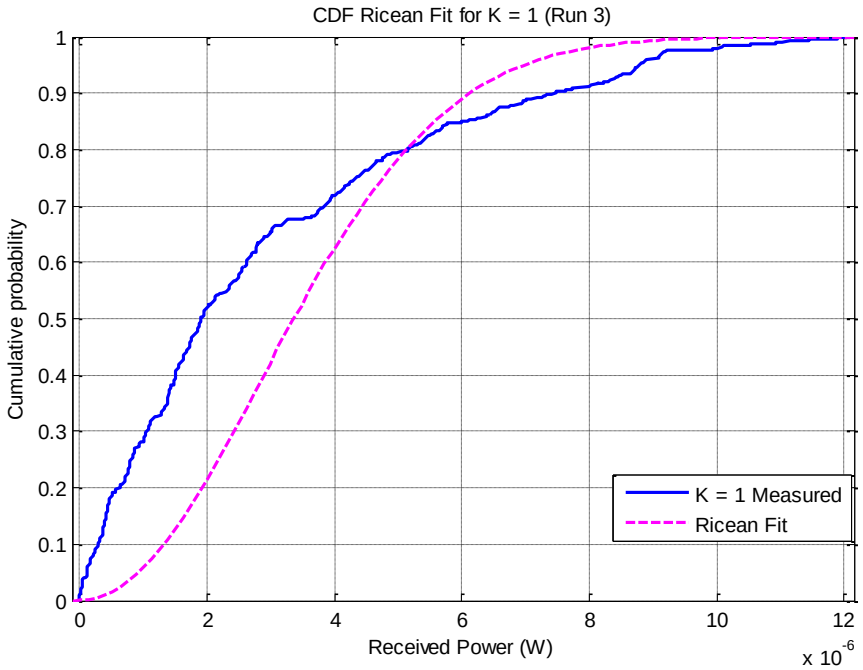


Figure D-17: CDF Ricean fit of the received power for K = 1 (Run 3)

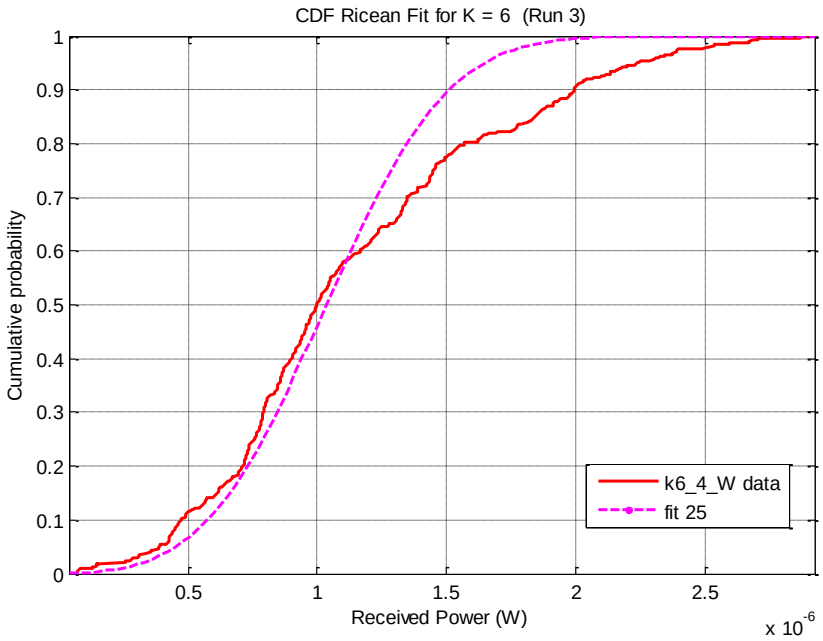


Figure D-18: CDF Ricean fit of the received power for K = 6 (Run 3)

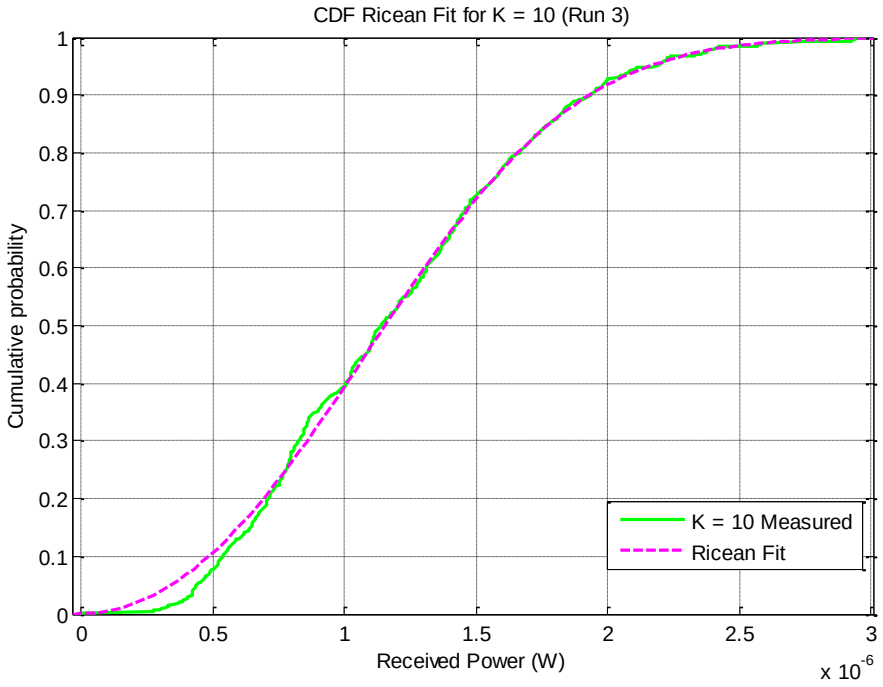


Figure D-19: CDF Ricean fit of the received power for K = 10 (Run 3)

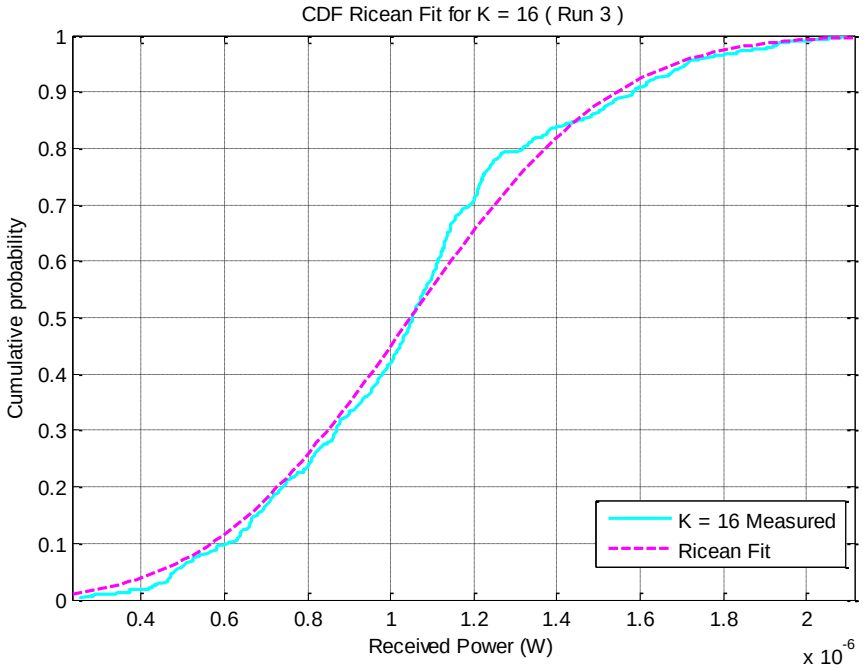


Figure D-20: CDF Ricean fit of the received power for K = 16 (Run 3)

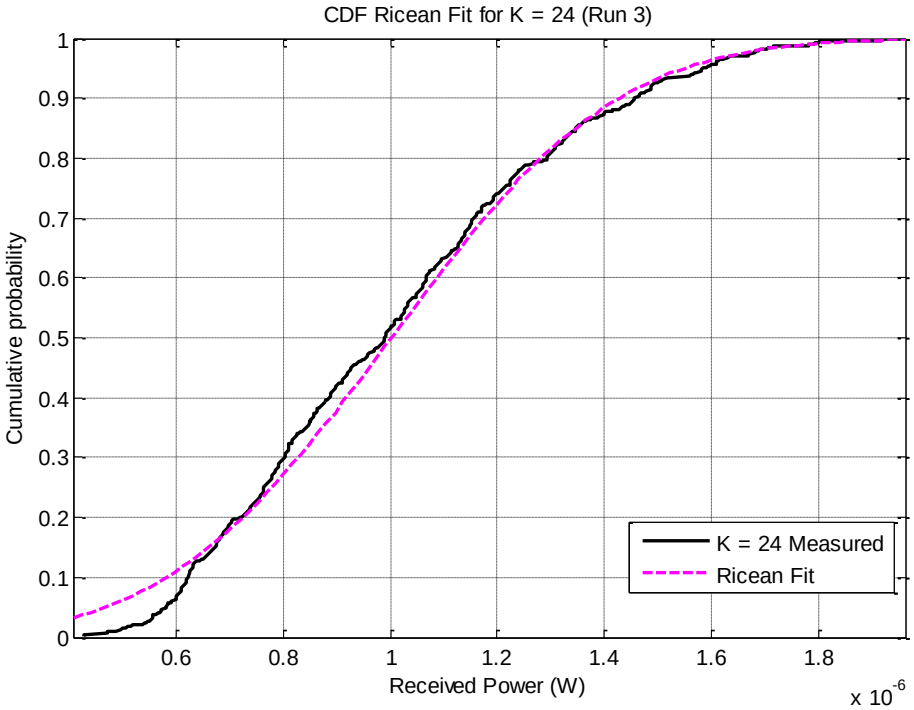


Figure D-21: CDF Ricean fit of the received power for K = 24 (Run 3)

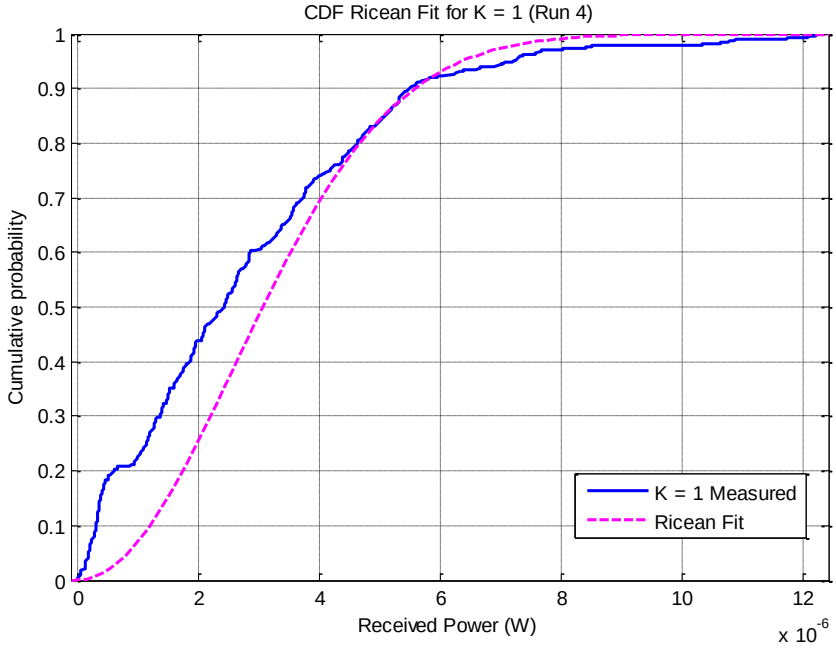


Figure D-22: CDF Ricean fit of the received power for K = 1 (Run 4)

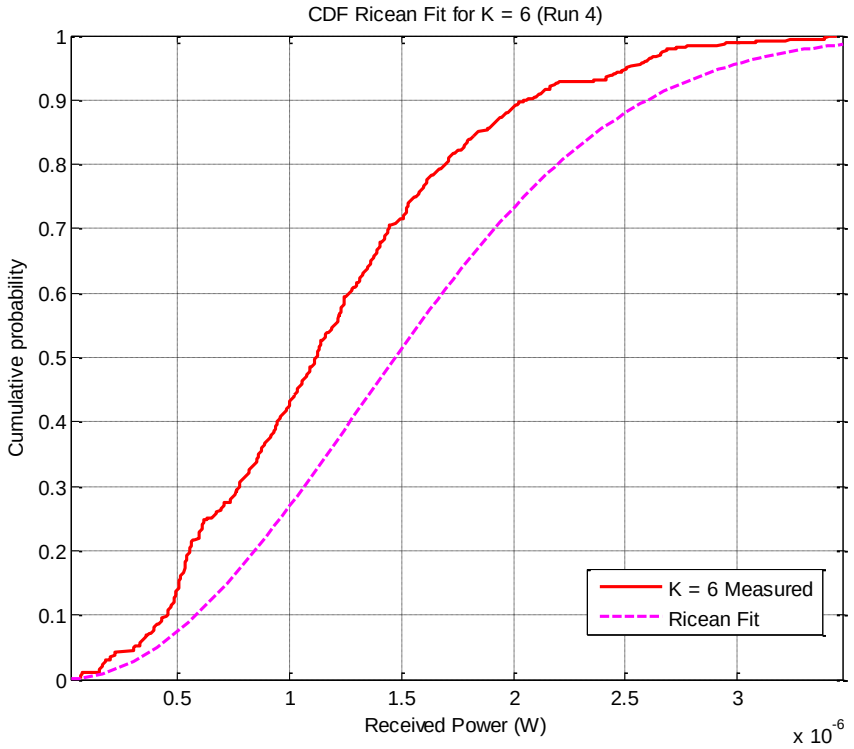


Figure D-23: CDF Ricean fit of the received power for K = 6 (Run 4)

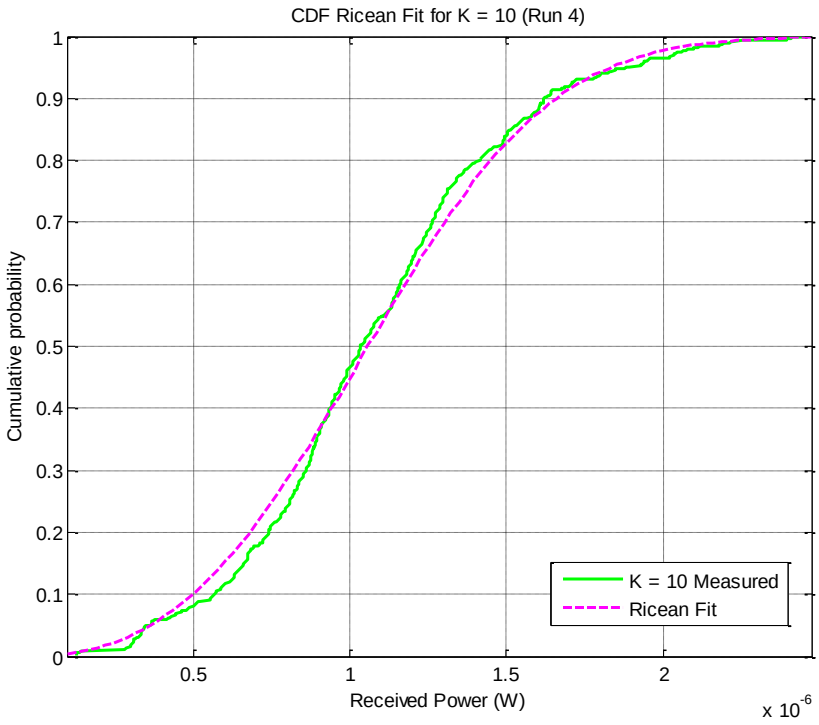


Figure D-24: CDF Ricean fit of the received power for K = 10 (Run 4)

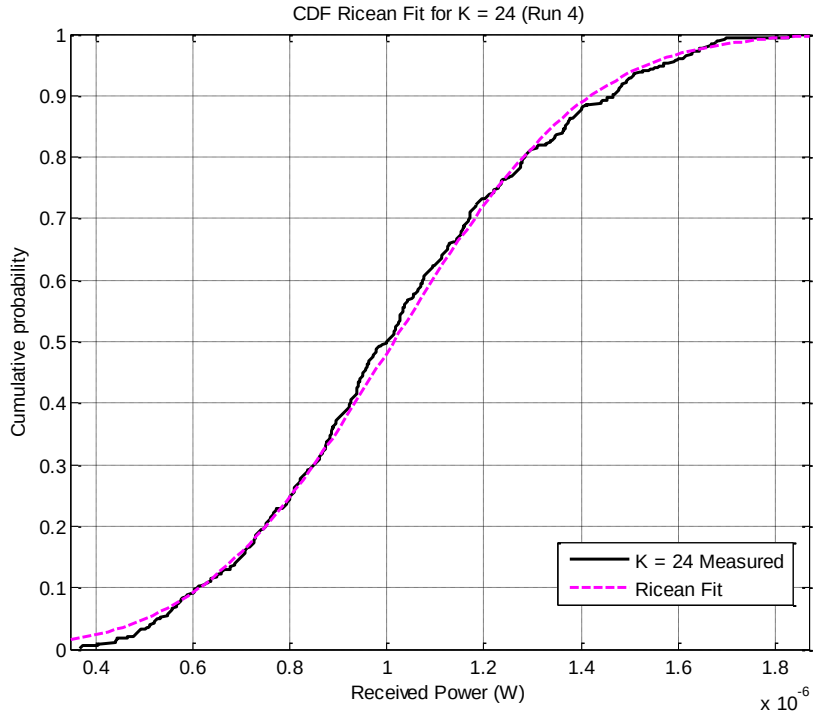


Figure D-25: CDF Ricean fit of the received power for K = 24 (Run 4)

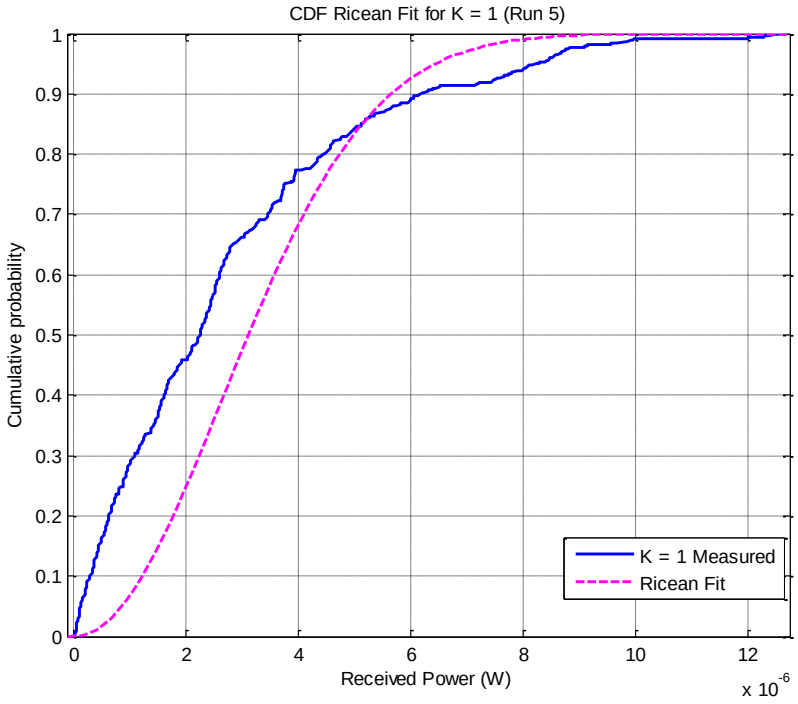


Figure D-26: CDF Ricean fit of the received power for K = 1 (Run 5)

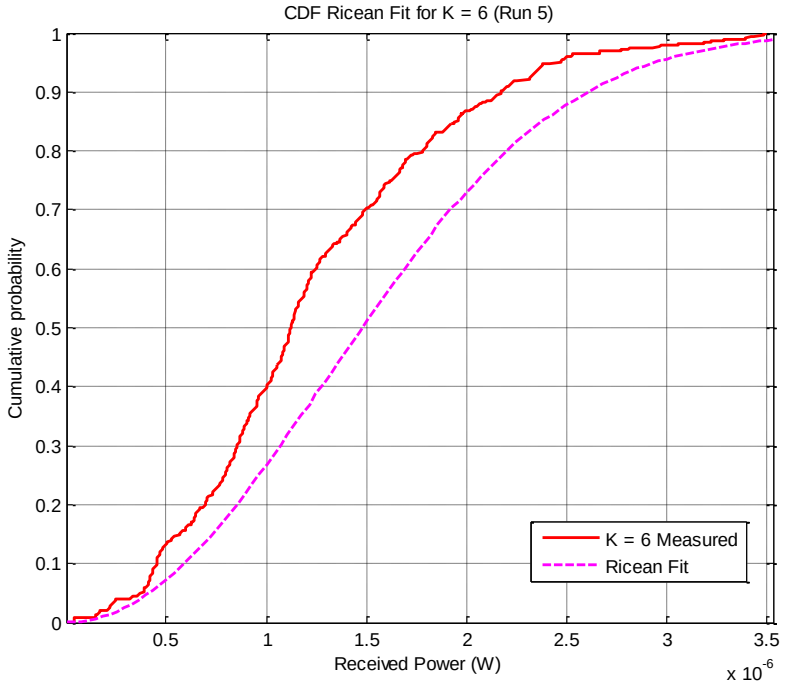


Figure D-27: CDF Ricean fit of the received power for K = 6 (Run 5)

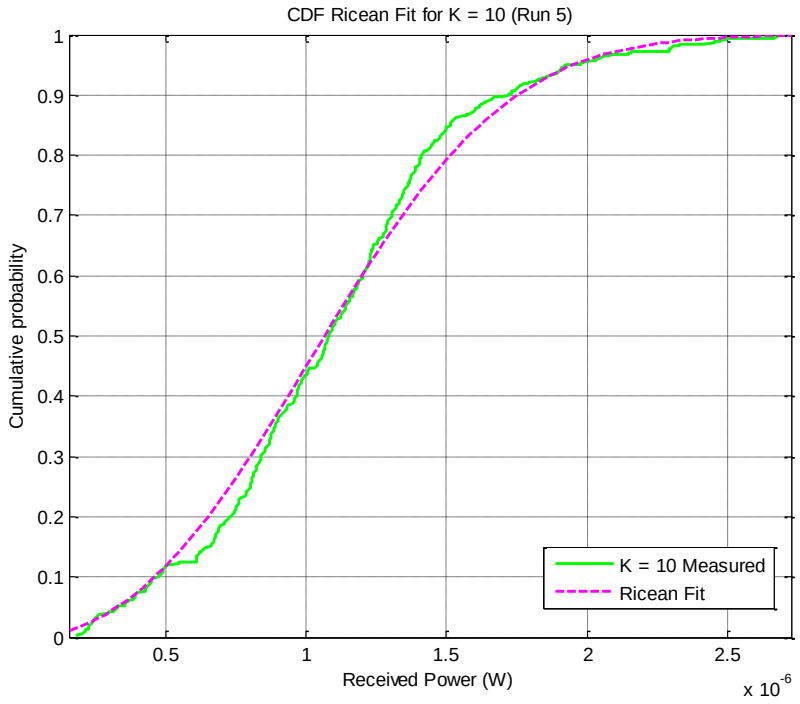


Figure D-28: CDF Ricean fit of the received power for K = 10 (Run 5)

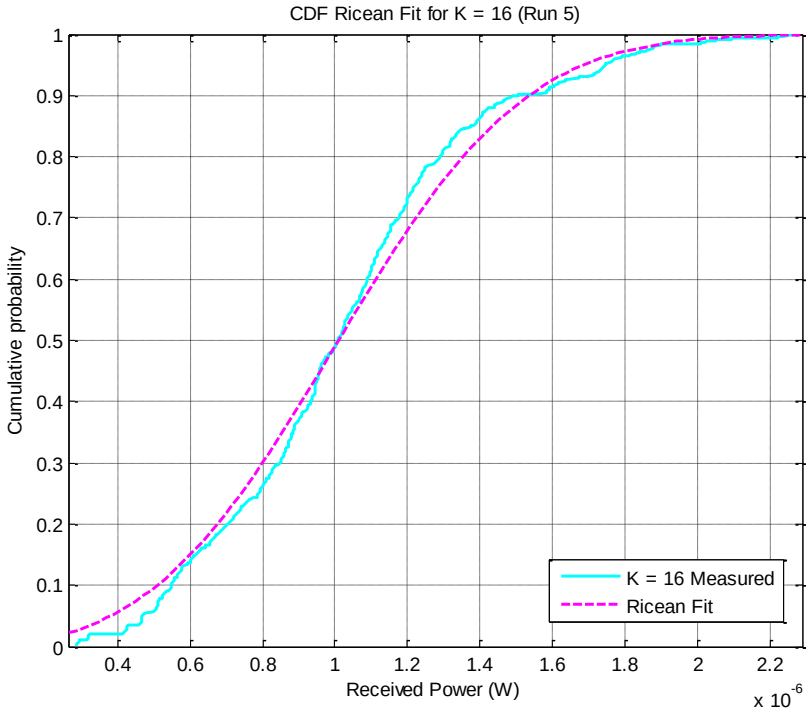


Figure D-29: CDF Ricean fit of the received power for K = 16 (Run 5)

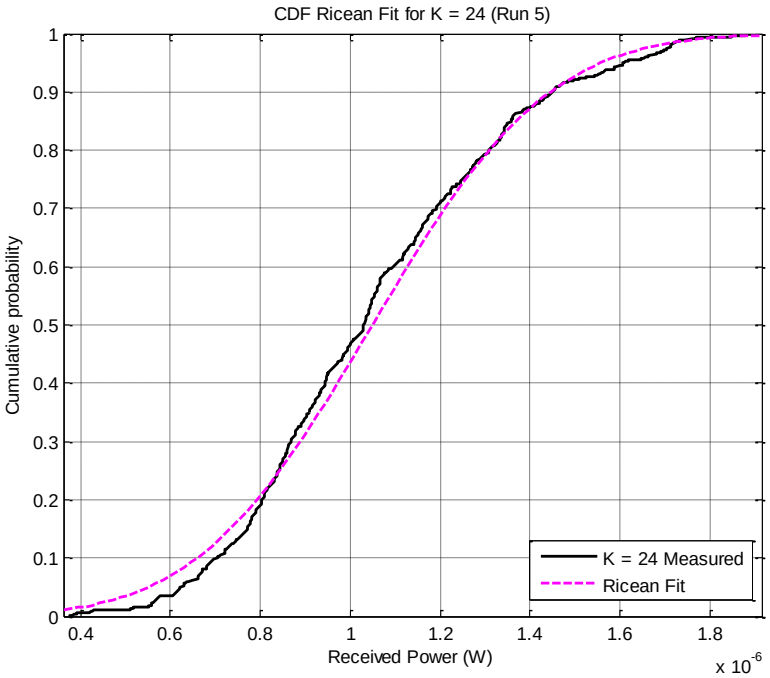


Figure D-30: CDF Ricean fit of the received power for K = 24 (Run 5)

Level Crossing Rate for different K values for the other four runs

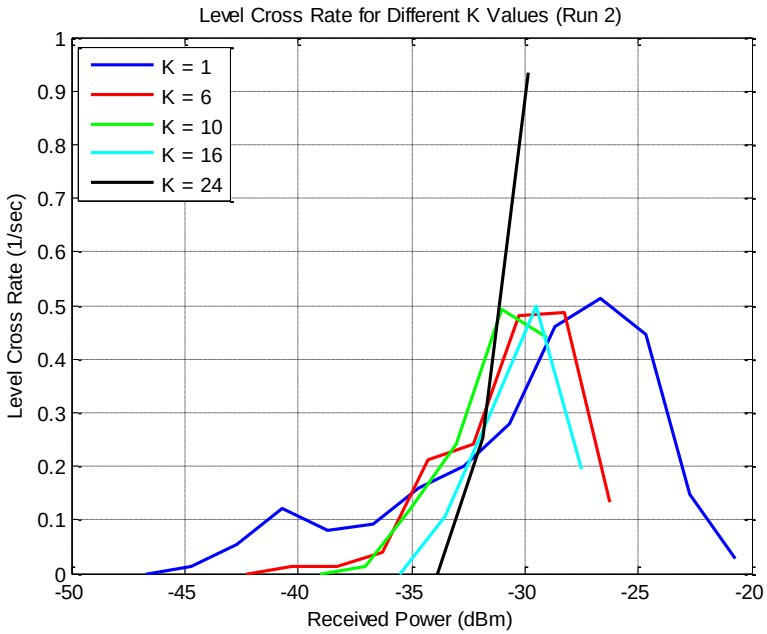


Figure D-31: LCR for different K values (Run 2)

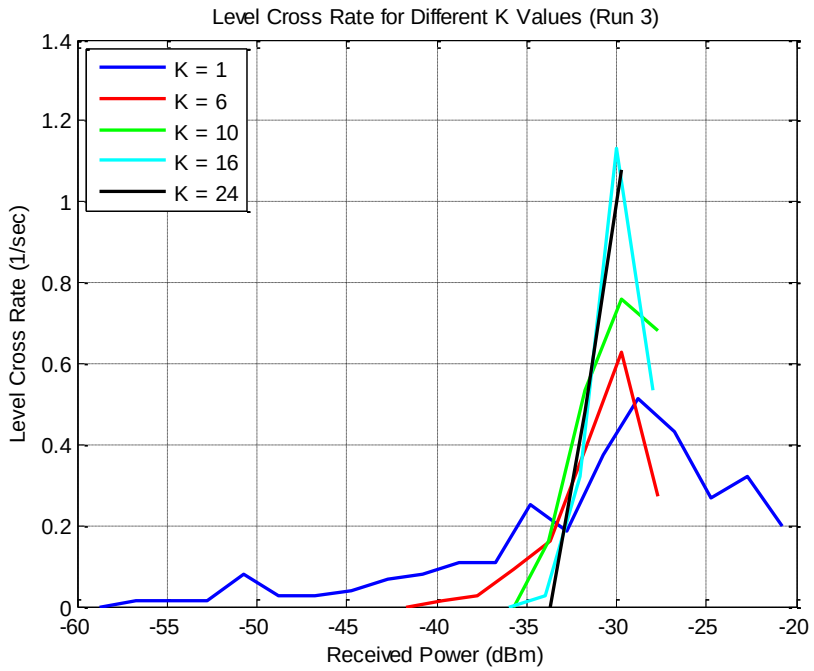


Figure D-32: LCR for different K values (Run 3)

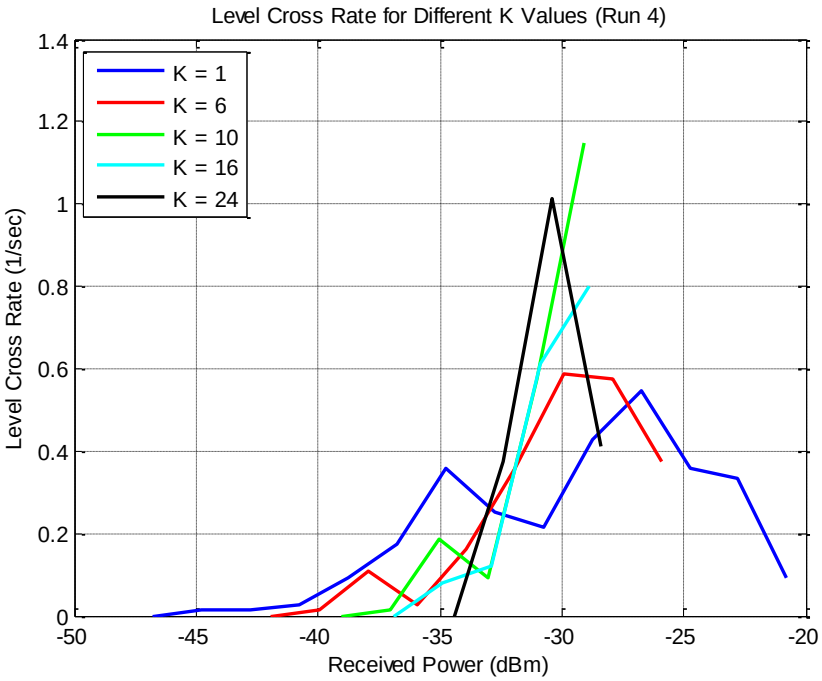


Figure D-33: LCR for different K values (Run 4)

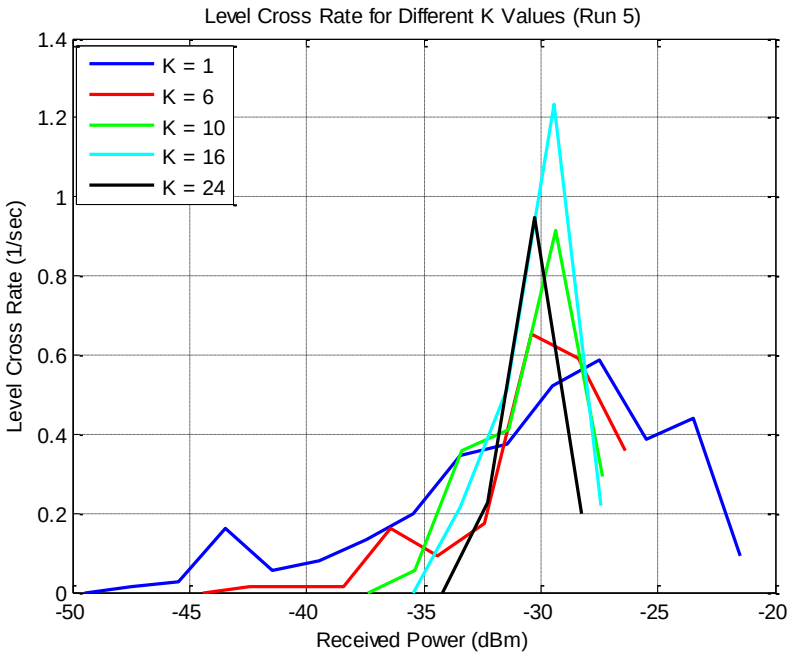


Figure D-34: LCR for different K values (Run 5)