Adaptive, Turbo-coded OFDM

by
Lou I. ILUNGA

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Dr. Annamalai Annamalai, Chair
Dr. Jeffrey Reed
Dr. Ira Jacobs

June 30, 2005
Blacksburg, VA

Keywords: OFDM, Turbo Codes, Adaptive Modulation

Adaptive, Turbo-coded OFDM

by
Lou. I. ILUNGA

ABSTRACT

Wireless technologies, such as satellite, cellular, and wireless internet are now commercially driven by ever more demanding consumers, who are ready for seamless integration of communication networks from the home to the car, and into the office. There is a growing need to quickly transmit information wirelessly and accurately. Engineers have already combine techniques such as orthogonal frequency division multiplexing (OFDM) suitable for high data rate transmission with forward error correction (FEC) methods over wireless channels.

In this thesis, we enhance the system throughput of a working OFDM system by adding turbo coding and adaptive modulation (AD). Simulation is done over a time varying, frequency selective Rayleigh fading channel. The temporal variations in the simulated wireless channel are due to the presence of Doppler, a sign of relative motion between transmitter and receiver. The wideband system has 48 data sub-channels, each is individually modulated according to channel state information acquired during the previous burst. The end goal is to increase the system throughput while maintaining system performance under a bit error rate (BER) of 10^{-2}. The results we obtained are preliminary. The lack of resources prevented us from producing detailed graphs of our findings.

Acknowledgments

I would like to take this opportunity to express my sincere thanks to my mother, Kabamba N. Ilunga and my two sisters Didi and Tete for without their unwaning support throughout this entire experience, I would not have succeeded.

Table of Contents

Chapter 1 Introduction. 1

1.1 Motivation. 1

1.2 Thesis Organization. 2

Chapter 2 Orthogonal Frequency Division Multiplexing or OFDM. 3

2.1 OFDM message. 3

2.2 Interference. 5

2.3 The Cyclic Prefix. 7

2.4 Channel Estimation and Equalization. 9

2.5 Power Amplifiers and Peak to Average Power Ratio (PAPR). 11

2.6 OFDM Simulator. 15

2.7 OFDM vs Single Carrier Alternatives. 20

2.8 Summary. 23

Chapter 3 Turbo Codes. 24

3.1 Introduction. 24

3.2 Turbo Encoding. 25

3.2.1 Recursive Systematic Convolutional (RSC) Codes. 25

3.2.2 Encoding of Parallel Concatenated Convolutional Codes. 26

3.3 Decoding Algorithms. 28

3.3.1 The Soft Output Viterbi Algorithm (SOVA). 32

3.3.2 The Modified Maximum A Posteriori (MAP) or BCJR. 34

3.4 Turbo Codes Performance. 41

3.4.1 Turbo Codes Implementation. 41

3.4.2 Performance of Turbo Codes 43

3.5 Summary. 46

Chapter 4 The Mobile Wireless Channel and Adaptive Modulation. 47

4.1 The Mobile Wireless Channel. 47

4.2 Simulating the Wireless Mobile Channel. 50

4.3 Channel Estimation. 58

4.4 Signal to Noise Ratio (SNR) 60

4.5 Adaptive Modulation. 62

Chapter 5 Conclusion. 72

5.1 Summary. 72

5.2 Future Work. 73

References. 75

LIST OF FIGURES

Figure 2.1	Basic OFDM system architecture	4
Figure 2.2	Effect of Frequency Offset (maintaining orthogonality)	6
Figure 2.3	Cyclic prefix	7
Figure 2.4	64QAM signal constellation diagrams for a 64-subcarrier OFDM system with flat Rayleigh fading. (a) The cyclic prefix is long enough to cover the delay spread. (b) The cyclic prefix is closer to being matched be the delay spread.	8
Figure 2.5	Illustrations of Class A, B, and C amplifier operating points	11
Figure 2.6	Power transfer function	13
Figure 2.7	BPSK BER performance of OFDM in an AWGN channel	16
Figure 2.8	BPSK BER performance of OFDM in an AWGN channel	17
Figure 2.9	BPSK BER performance of OFDM over a fast Rayleigh faded channel (perfct channel knowledge)	19
Figure 2.10	OFDM performance in a fast flat Rayleigh faded channel (perfect channel knowledge)	20
Figure 2.11	OFDM performance in a fast Rayleigh faded channel with frequency error of 700Hz (perfect channel knowledge)	22
Figure 3.1	Constraint length K = 2 convolutional encoder	25
Figure 3.2	Recursive Systematic Convolutional encoder	26
Figure 3.3	Turbo encoding scheme	27
Figure 3.4	Soft-input soft-output module: z_i, the a priori values for information bits, $y_i^{(p)}$ the parity observations, $y_i^{(s)}$ the systematic observations, and Λ_i the a posteriori values	29
Figure 3.5	Schematic of the Turbo Decoder	30
Figure 3.6	Feed-forward representation of r=1/2 RSC	41
Figure 3.7	Pseudorandom interleaver 2-D scatter plot	42

Figure 3.8	Puncturing pattern of form r=1/2 TC	43
Figure 3.9	Rate 1/3, constraint length K = 2, Turbo coded BPSK performance.	45
Figure 4.1	Power delay profile of a fast Rayleigh fading channel	49
Figure 4.2	PDF of the simulated Rice process is approximately Gaussian when $N_i = 7$	52
Figure 4.3	The PDF of envelope of the simulated Rice process is Rayleigh distributed	53
Figure 4.4	Doppler power spectra according to Jakes' Model	54
Figure 4.5	The PDF of a Rayleigh Channel	55
Figure 4.6	Original Rayleigh faded envelope for one packet	56
Figure 4.7	Upsampled Rayleigh faded envelope for one packet	57
Figure 4.8	Effect of Doppler on Rayleigh envelope	58
Figure 4.9	SNR estimation mean square error	61
Figure 4.10	AWGN BER curves	63
Figure 4.11	Adaptive OFDM in AWGN	65
Figure 4.12	Bits per symbol as a function of SNR	66
Figure 4.13	Adaptive OFDM in slow Rayleigh fading channel	67
Figure 4.14	Bits per symbol as a function of SNR	68
Figure 4.15	Turbo OFDM AWGN performance curves (3 iterations)	69
Figure 4.16	Adaptive Turbo OFDM in slow Rayleigh fading channel (3 iterations)	70

Figure 4.17 BPS as a function of SNR 71

LIST OF TABLES

Table 4.1	AWGN Switching Thresholds	64
Table 4.2	Turbo-AWGN Switching Thresholds	69

Chapter 1. Introduction

The telecommunications' industry is in the midst of a veritable explosion in wireless technologies. Once exclusively military, satellite and cellular technologies are now commercially driven by ever more demanding consumers, who are ready for seamless communication from their home to their car, to their office, or even for outdoor activities. With this increased demand comes a growing need to transmit information wirelessly, quickly, and accurately. To address this need, communications engineer have combined technologies suitable for high rate transmission with forward error correction techniques. The latter are particularly important as wireless communications channels are far more hostile as opposed to wire alternatives, and the need for mobility proves especially challenging for reliable communications.

1.1 Motivation

For the most part, Orthogonal Frequency Division Multiplexing (OFDM) is the standard being used throughout the world to achieve the high data rates necessary for data intensive applications that must now become routine.

This thesis enhances the throughput of an existing OFDM system by implementing adaptive modulation and turbo coding. The new system guarantees to reach a target performance BER of 10^{-2} over a slow time-varying fading channel. The system automatically switches from lower to higher modulation schemes on individual subcarriers, depending on the state of the quasi-stationary channel.

In conjunction with the adaptive design, forward error correction is performed by using turbo codes. The combination of parallel concatenation and recursive decoding allows these codes to achieve near Shannon's limit performance in the turbo cliff region.

1.2 Thesis Organization

This thesis presents the implementation and of an adaptive, turbo-coded OFDM system. It is presented as follows:

Chapter 2 introduces the theory behind OFDM as well as some of its advantages and functionality issues. We discuss basic OFDM transceiver architecture, cyclic prefix, intersymbol interference, intercarrier interference and peak to average power ratios. We also present a few results in both Additive White Gaussian Noise, and Rayleigh environments

Chapter 3 focuses on turbo codes. We explore encoder and decoder architecture, and decoding algorithms (especially the maximum a posteriori algorithm). We elaborate on the performance theory of the codes and find out why they perform so well.

Chapter 4 ties both technologies together. First we introduce the slow time-varying Rayleigh fading channel. We proceed by finding how to estimate channel state information and apply that knowledge to our adaptive modulation scheme. Then, we present our results on the combination of turbo coding and adaptive OFDM. The core of our simulation results are found here.

Chapter 5 consists in a summary of our work and a few suggestions are made on how to improve our system.

Chapter 2. OFDM

Orthogonal frequency division multiplexing (OFDM) is nowadays widely used for achieving high data rates as well as combating multipath fading in wireless communications. In this multi-carrier modulation scheme data is transmitted by dividing a single wideband stream into several smaller or narrowband parallel bit streams. Each narrowband stream is modulated onto an individual carrier. The narrowband channels are orthogonal vis-à-vis each other, and are transmitted simultaneously. In doing so, the symbol duration is increased proportionately, which reduces the effects of inter-symbol interference (ISI) induced by multipath Rayleigh-faded environments. The spectra of the subcarriers overlap each other, making OFDM more spectral efficient as opposed to conventional multicarrier communication schemes.

2.1. OFDM message

The OFDM message is generated in the complex baseband. Each symbol is modulated onto the corresponding subcarrier using variants of phase shift keying (PSK) or different forms of quadrature amplitude modulation (QAM). The data symbols are converted from serial to parallel before data transmission. The frequency spacing between adjacent subcarriers is $\frac{2\pi}{N}$, where N is the number of subcarriers. This can be achieved by using the inverse discrete Fourier transform (IDFT), easily implemented as

the inverse fast Fourier transform (IFFT) operation. As a result, the OFDM symbol generated for an N-subcarrier system translates into N samples, with the ith sample being

$$x_i = \sum_{n=0}^{N-1} C_n \exp\left\{ j \frac{2\pi i n}{N} \right\}, 0 \le i \le N-1 \qquad (2.1)$$

At the receiver, the OFDM message goes through the exact opposite operation in the discrete Fourier transform (DFT) to take the corrupted symbols from a time domain form into the frequency domain. In practice, the baseband OFDM receiver performs the fast Fourier transform (FFT) of the receive message to recover the information that was originally sent.

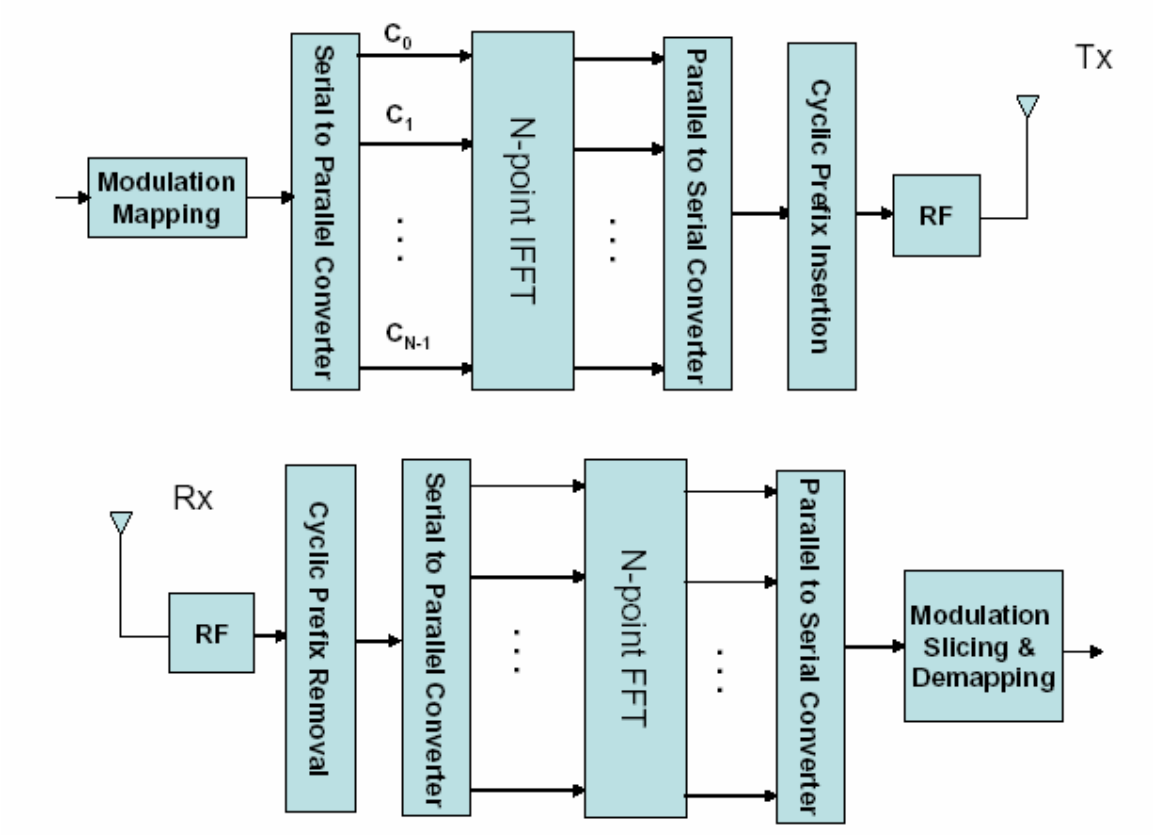

Figure 2.1 Basic OFDM system architecture

2.2. Interference

In a multipath environment, different versions of the transmitted symbol reach the receiver at different times. This is due to the fact that different propagation paths exist between transmitter and receiver. As a result, the time dispersion stretches a particular received symbol into the one following it. This symbol overlap is called inter-symbol interference, or ISI. It also is a major factor in timing offset. One other form of interference is inter-carrier interference or ICI. In OFDM, successful demodulation depends on maintaining orthogonality between the carriers. We demodulate a specific subcarrier N at its spectral peak, meaning that all the other carriers must have a corresponding zero spectra at the Nth center frequency (frequency domain perspective). Frequency offsets lead to this criterion not being met. This condition can seriously hinder the performance of our OFDM system. Figure 2.2 below shows that when the decision is not taken at the correct center frequency (i.e. peak) of carrier considered, adjacent carriers factor in the decision making, thus reducing the performance of the system.

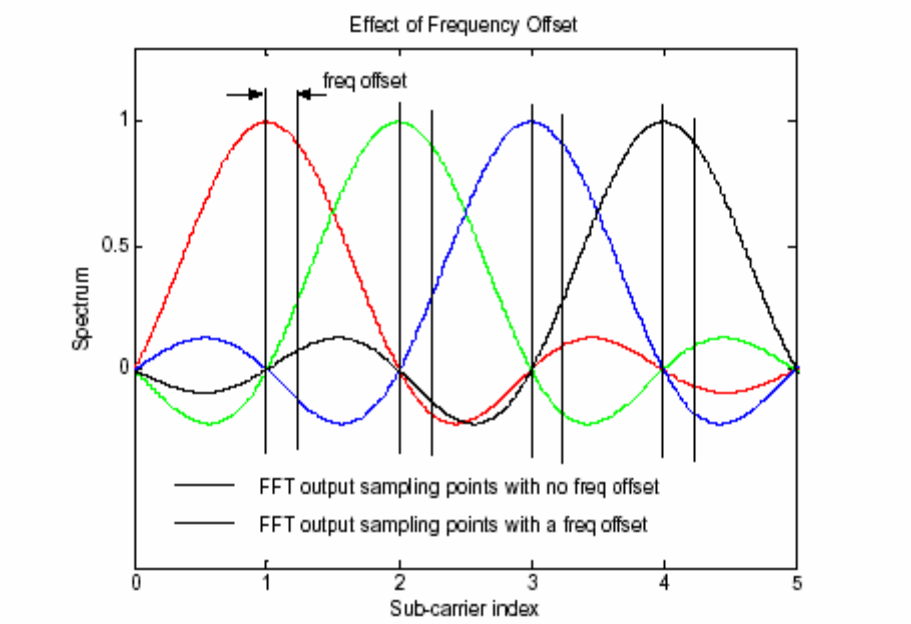

Figure 2.2 Effect of Frequency Offset (maintaining orthogonality)

2.3. The Cyclic Prefix

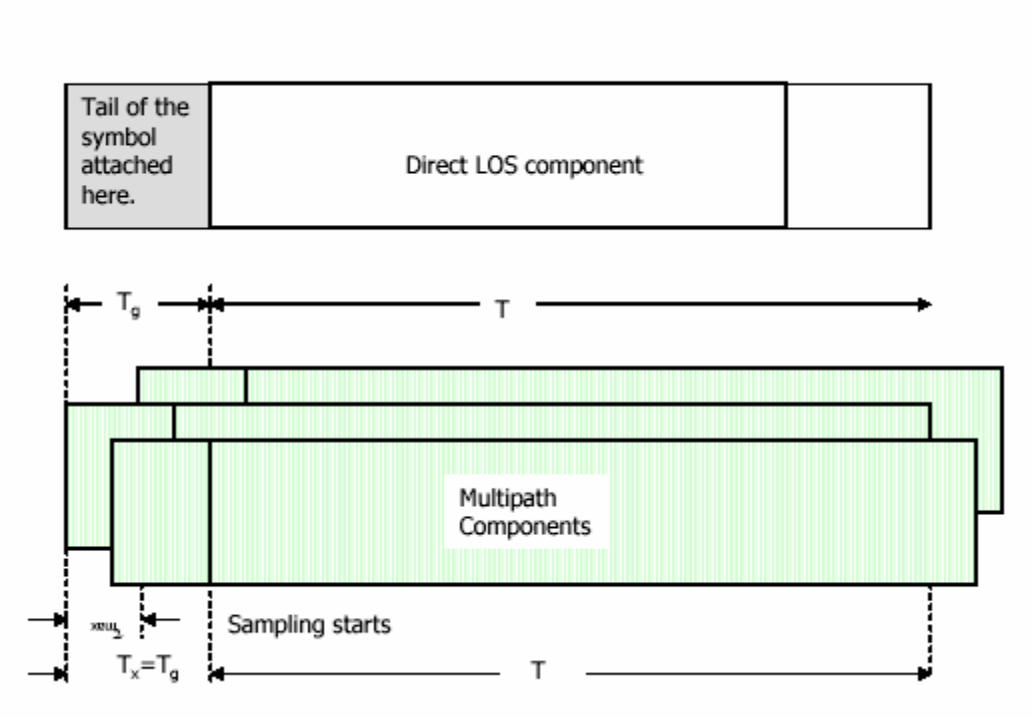

Figure 2.3 Cyclic prefix

OFDM demodulation must be synchronized both in the time domain as well as in the frequency domain. Engineers have found a way to ensure that goal by adding a guard time in the form of a cyclic prefix (CP) to each OFDM symbol. The CP consists in duplicates of the end samples of the OFDM message relocated at the beginning of the OFDM symbol. This increase the length T_{sym} of the transmit message without altering its frequency spectrum.

$$T_{sym} = CP + T_{data} N \tag{2.2}$$

where T_{data} is the duration of one data symbol, and N the number of carriers. The receiver is set to demodulate over a complete OFDM symbol period, which maintains orthogonality. As long as the CP, is longer than the channel delay spread, τ_{max}, the system will not suffer from ISI. The CP is to be added after the FFT operation at the transmitter

and removed prior to demodulation. The figure below whose the deteriotiation in performance when the CP is closely matched by the delay spread. The signal constellation is less tightly grouped, no doubt a sign of less than accurate decoding.

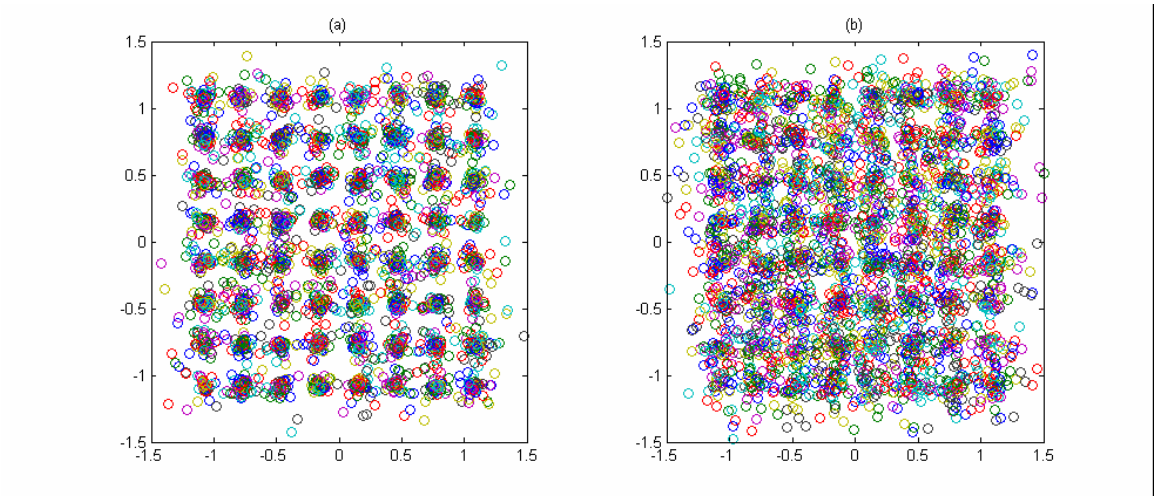

Figure 2.4 64QAM signal constellation diagrams for a 64-subcarrier OFDM system with flat Rayleigh fading. (a) The cyclic prefix is long enough to cover the delay spread. (b) The cyclic prefix is closer to being matched be the delay spread.

2.4. Channel Estimation and Equalization

Typically OFDM systems have known pilots symbols, or training data, inserted on the subcarriers before the IFFT operation at the transmitter. These symbols have been added to mitigate the interference between replicas of the data at the receiver. This data is to be used to estimate the channel. There is a real tradeoff in utilizing this technique. Indeed, pilots could potentially be used to send additional information thus increasing the bandwidth efficiency. On the other hand, the more pilots we include in our message, the more accurately we will be able to track and estimate the frequency response of the channel. We need to identify the minimum pilot spacing, Δp, for our OFDM system. In the frequency domain, the channel variation corresponds to maximum Doppler frequency f_{max}. According to [1],

$$\Delta p \leq \frac{1}{2 f_{\max} T_{sym}} \tag{2.3}$$

where T_{sym} is the OFDM symbol period. One must also note that the frequency domain correlation of the channel frequency response can be used to estimate the channel. The coherence bandwidth is defined as

$$\Delta f \approx \frac{1}{\tau_{\max}} \tag{2.4}$$

With τ_{max} being the maximum channel delay spread. When the subcarrier spacing is much less compared to the coherence bandwidth, neighboring carriers will be highly correlated. We discuss this in greater detail in a later chapter (4).

Once we have the channel information estimated, we can remove the negative effects of the channel from the receive signal by using one of three general equalization techniques: the maximum likelihood sequence estimation (MLSE), linear equalizers, and decision feedback equalizers. We only need a one tap equalizer for each subcarrier. This makes the linear equalizer method the logical choice. We can determine the coefficient of the equalizer by using either the MMSE or the zero forcing (ZF) criteria. The latter works as follows:

$$\hat{H}_n = \frac{Y_n}{P_n} = H_n + \frac{N_o}{P_n} \tag{2.5}$$

where Y_n is the receive signal, P_n represents the pilot symbols and N_o, additive white Gaussian noise. Using the pilot symbols to arrive at a channel estimate is also referred as pilot symbol aided modulation or PSAM. We will cover channel estimation again in a later chapter.

2.5. Power amplifiers and Peak to average power ratio (PAPR)

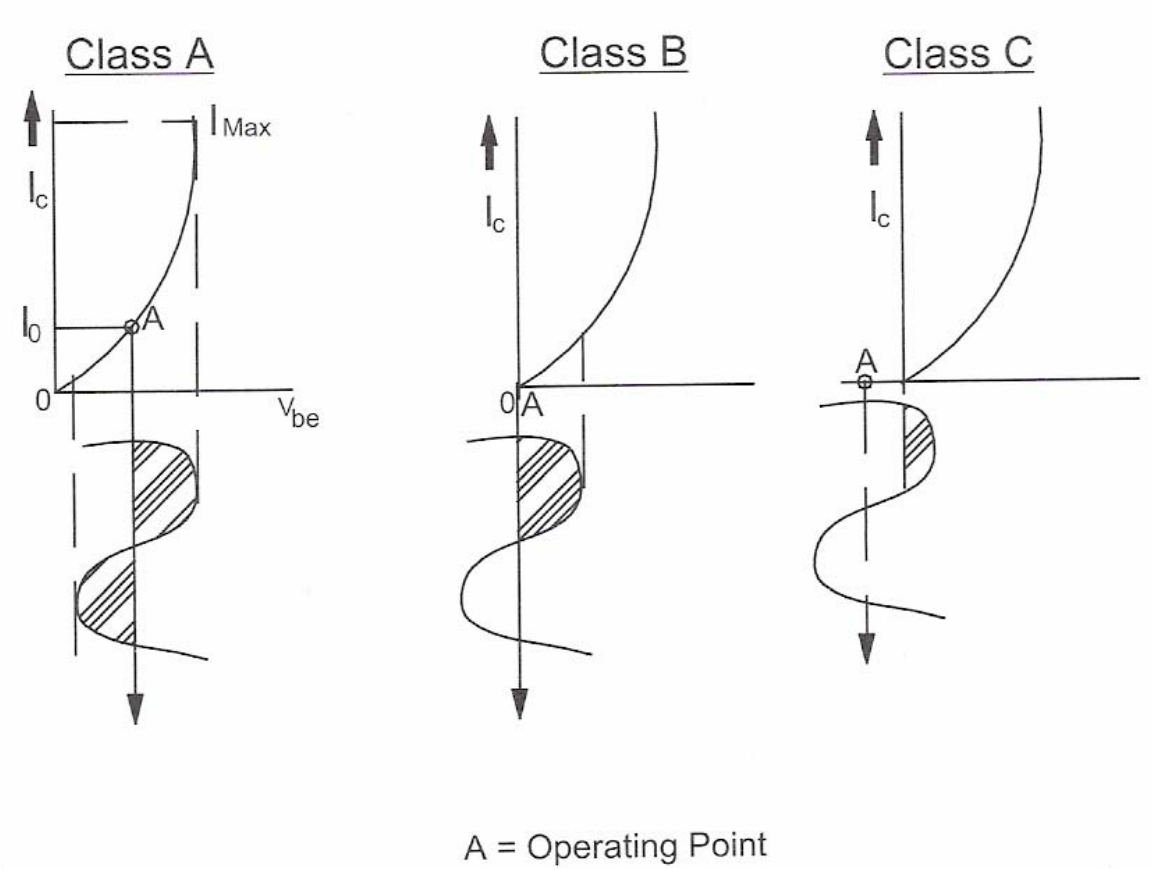

Figure 2.5 Illustrations of Class A, B, and C amplifier operating points

Power amplifiers are commonly classified under 4 classes: A, AB, B, and C. Class A amplifiers are unique as current continuously flows through the device at all times. They essentially operate over the linear region of the power transfer characteristic. Consequently, their input and output powers are related to each other by a positive or negative gain (scalar). In addition, class A amplifiers have very poor conversion efficiency, i.e. the ability to convert input DC power to output AC power (25%). Class B operation features an improved conversion efficiency but loss of linearity is unavoidable.

The amplifier functions somewhat like a rectifier as it only allows current flow during half of the signal cycle. Finally, class C rectifiers have a zero output current for more than half of the signal cycle. Conversion efficiency is unparallel but the output suffers from critical levels of harmonic distortions. Knowing these, characteristics, we can now understand how amplifiers can affect an OFDM system.

The main drawback of OFDM systems is the large PAPR caused by summing the carriers together. The maximum peak power increases proportionally to the number of carriers used in the system. The problem surfaces because amplifiers cannot function in a wide linear region to accommodate the large PAPR required by an OFDM system. Indeed, today's amplifiers have a relatively short linear region where the output power is a scalar version of the input power.

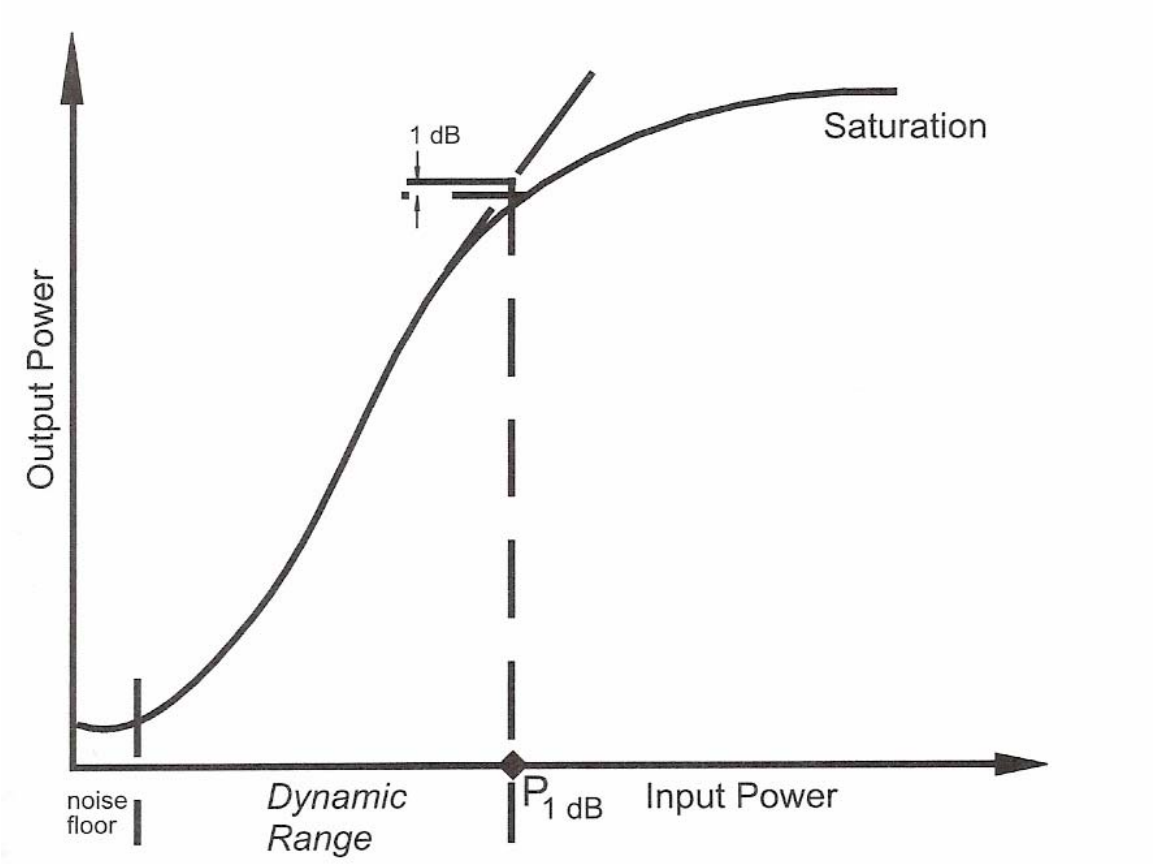

Figure 2.6 Power transfer function

Once you leave that linear region, the output of the power amplifier goes into a saturation region where the scalar relationship is lost. The use of amplifiers in the saturation region leads to the emergence of intermodulation products (signal distortion), something that cannot be tolerated.

Methods to mitigate this phenomenon include pre-distortion techniques, and coding. Distortion techniques attempt to alleviate non-liner distortions by altering the input signals characteristics in an adaptive or non-adaptive scheme. One of the most commonly used of these methods is clipping. Amplitude clipping can also be viewed as a noise source. The goal is to limit the amplitude of the input signal of the system to a preset

maximum value. The technique comes at a price. Indeed, the end result is an increase in in-band noise/distortion which cannot be reduced and leads to a degradation of the bit error rate BER performance. Also there is some out of band spectral leakage which can be reduced by using windowing or filtering. [4] and [5] talk in more details about potential clipping mitigating techniques.

Coding and/or scrambling techniques focus on selective transmission of symbols or data sequences based on the PAPR. These include but are not limited to partial transmit sequences (PTS) [6], selective mapping (SLM) [7], and block coding.[8].

2.6. OFDM Simulator

For simulation purposes, we based our work on the simulation tool provided online in [9]. It's a complete OFDM WLAN physical layer simulation in MATLAB. The program simulates a 64 subcarrier OFDM system. The system supports up to 2 transmit and 2 receive antennas, a convolutional code generator with rates ½, 2/3, and 3/4. The code is punctured to IEEE specifications. As an option, one can chose to interleave the transmit bits for added protection. The system supports 4 modulation schemes, binary phase shift keying, quadrature phase shift keying, sixteen quadrature amplitude modulation, and sixty four quadrature amplitude modulation. Frequency jitter can also be added to a system that supports two channel models, namely additive white Gaussian noise, AWGN and flat Rayleigh fading. One can input the desired length of the delay spread. The cyclic prefix is 16 samples long. You can also request a specific average signal to noise ratio. Transmit power amplifier effects and phase noise distortion can be added to the transmit signal. The simulator also comes with a series of synchronization algorithms including packet detection, fine time synchronization, frequency synchronization, pilot phase tracking, channel estimation, all of that if you wish to simulate IEEE 802.11 standards. There is also a switch to add a receiver timing offset. We drastically modified the simulator to study the aspects relevant to the scope of our research. For this chapter, we removed most of the options already present in the tool and have made some assumptions worthy to be noted.

For now, we purposely omit any type of coding or interleaving. We also omit transmit/receive diversity. We have removed the effects of the transmit power amplifier, as we focus the simulation in baseband, and phase noise.

We assume perfect synchronization both in time and in frequency. We also include perfect channel estimation for the time being.

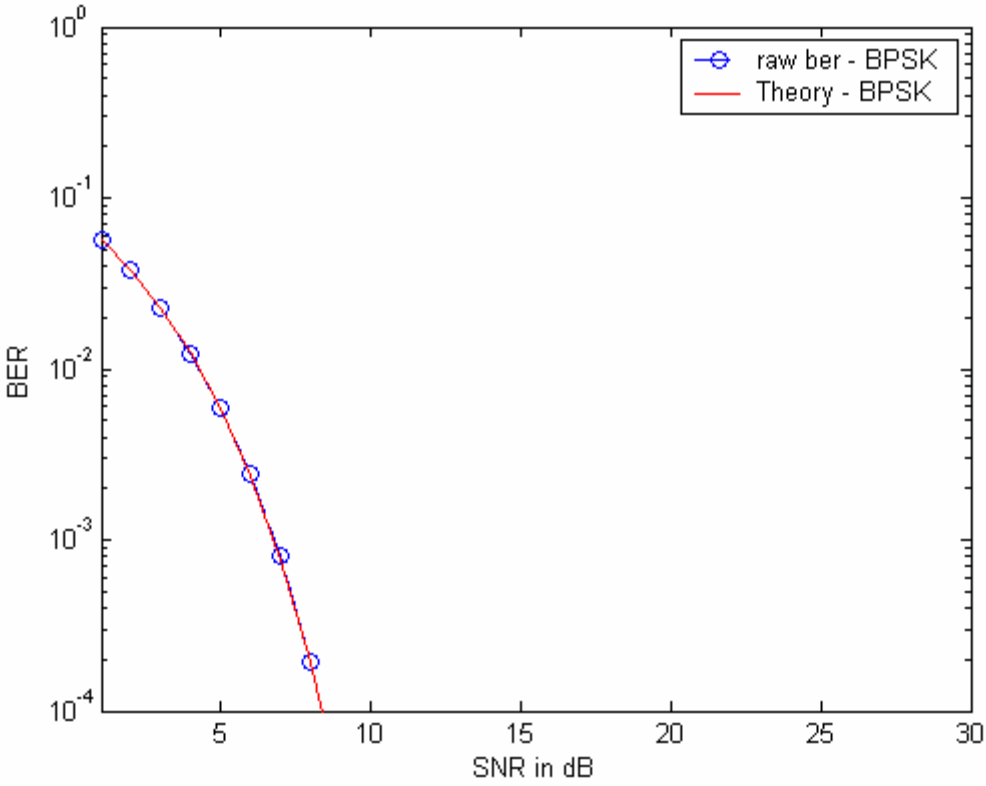

Figure 2.7 BPSK BER performance of OFDM in an AWGN channel

As seen in Figure 2.7, the binary phase shift keying raw BER obtained through simulation matches perfectly with the theoretical curve obtain by using the equation (2.5)

$$Pe = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \tag{2.5}$$

where Q( ) represents the Q function given by,

$$Q(z) = \int_{z}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} dx \tag{2.6}$$

and implemented in MATLAB by the complementary error function,

$$Q(z) = \frac{1}{2} erfc\left(\frac{z}{\sqrt{2}}\right). \tag{2.7}$$

We extend our simulation to include the modulation schemes we plan to use in the OFDM simulator. These schemes are two variants of multiple phase shift keying (MPSK), and two variant of multiple quadrature amplitude modulation (MQAM). Actually, all 4 schemes can be considered quadrature amplitude modulation.

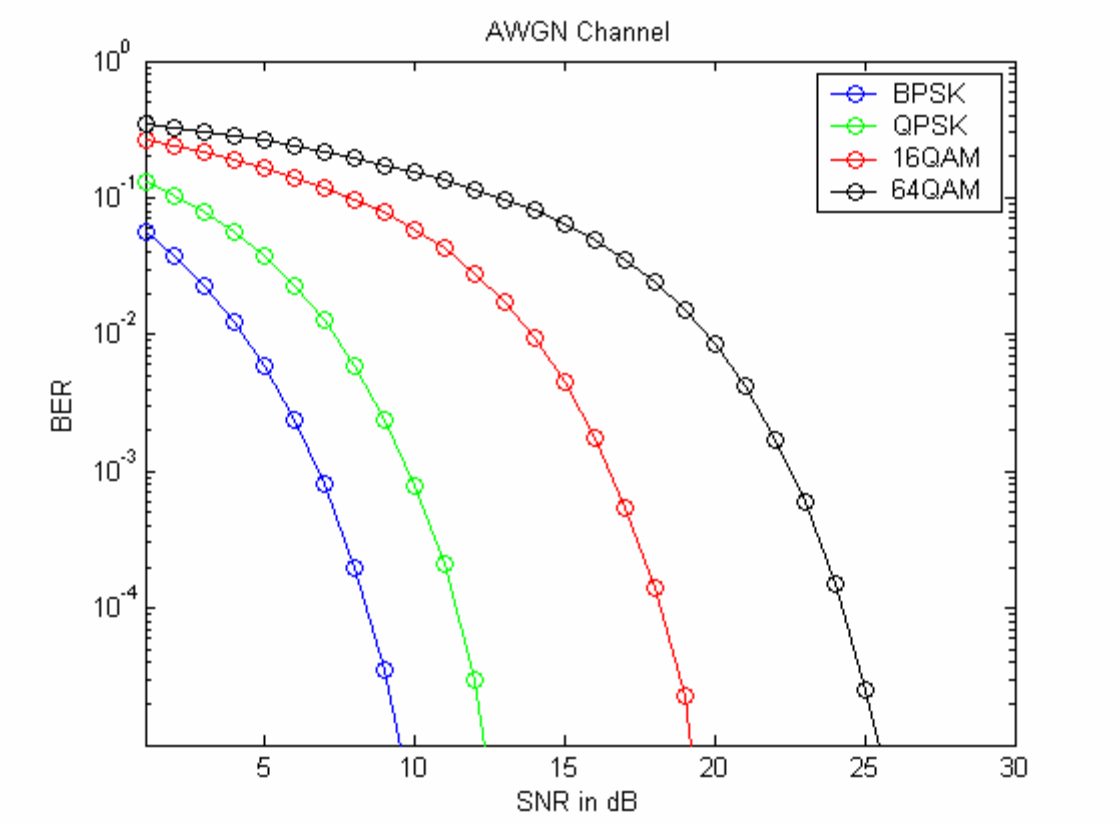

Figure 2.8 BPSK BER performance of OFDM in an AWGN channel

We follow up our experiment with a study of BER performance of our system in Rayleigh environments. Rayleigh fading emerges when multiple time-shifted or delayed versions of the originally transmitted signal emerge at the receiver. This phenomenon is due to the existence of various paths the signal can take before arriving at destination. These replicas interfere with one another, causing Rayleigh fading. When the difference between the delays is negligible, we can ignore it and model the signal as having only one delayed path. This is called flat Rayleigh fading. When the delays are clearly separated, the system suffers from frequency selective Rayleigh fading. Because of the properties of OFDM, each subcarrier is considered flat. In this thesis, we limit the scope of the simulation to frequency selective Rayleigh fading.

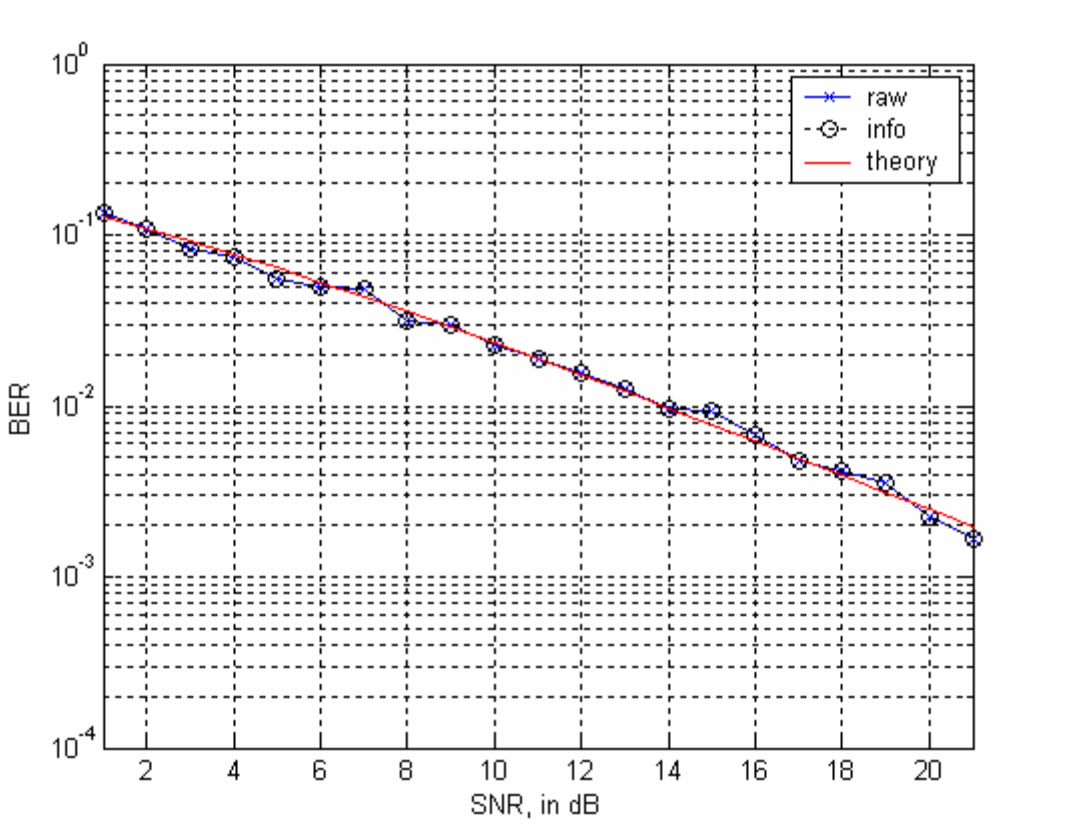

Figure 2.9 BPSK BER performance of OFDM over a fast Rayleigh faded channel (perfct channel knowledge)

As seen in above, our simulation results match the theoretical BPSK BER curve which is

$$Pe = \frac{1}{2}\left(1 - \sqrt{\frac{\overline{\beta}}{1+\overline{\beta}}}\right) \tag{2.8}$$

where $\overline{\beta}$ is the average signal to noise ratio (SNR) of the channel. This is a case of fast fading. We will explore slow fading in chapter 4. We expand our simulation to encompass all four modulation scheme cited above in this same fast Rayleigh fading channel. The results can be seen in Figure 2.10 below.

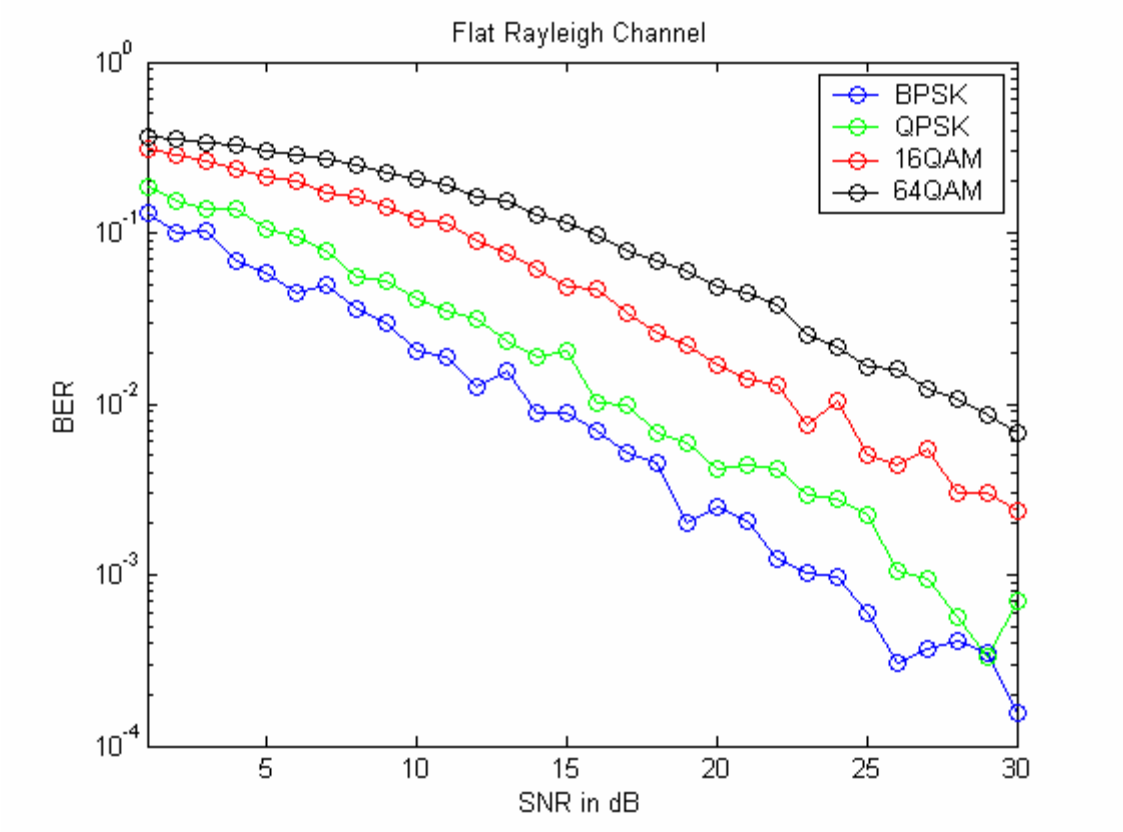

Figure 2.10 OFDM performance in a fast flat Rayleigh faded channel (perfect channel knowledge)

2.7. OFDM versus Single Carrier Alternative

The main differences between OFDM based systems and single carrier systems with the same data rate are their resiliency to fading and how susceptible they are to synchronization errors. As seen above, single carrier schemes and OFDM based systems are equivalent for AWGN and flat Rayleigh channels. There appears to be no inherent advantage to either technique for recovering information. However if you consider a frequency selective environment, the single carrier method requires a equalizer to compensate for the channel effects. This is a source of error as equalization can never be perfect and the operation could possible enhance noise amplitude in some part of the

signal to be demodulated. The OFDM equalizer is subject exactly the same limitations but the scheme perform 1-tap equalization on each subcarrier while, the single carrier approach must utilize multi-tap equalizers. The complexity of the latter is proportional to the square of its number of taps, which complicates implementation.

For time synchronization, OFDM systems possess known pilot signals that are transmitted in conjunction with the data. These symbols can be used for channel estimation as well as to compensate the phase distortion of the signal at the receiver. Single carrier lacks of a comparable mechanism. When the receiver is not synchronized to the transmitted data, the detected SNR suffers. The output SNR is given by

$$\rho = \frac{\Lambda(\tau)}{\Lambda(0)} \tag{2.9}$$

where Λ is the autocorrelation function and τ is the optimum sampling time and the time of the received signal.

OFDM based systems are vulnerable to poor frequency synchronization. Frequency errors can be introduced by Doppler shift, i.e. relative motion between the transmitter and receiver, or unreliable oscillators at the transmitter and/or receiver. The multicarrier scheme must maintain orthogonality between subcarriers to successfully transmit/receive data.

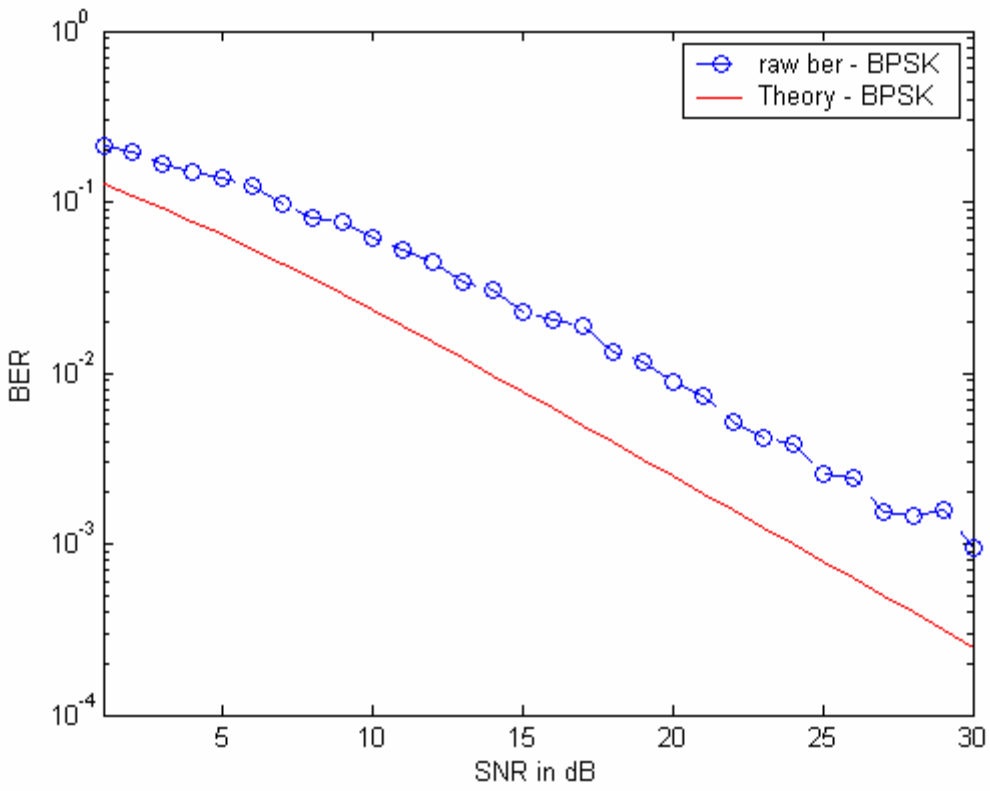

Figure 2.11 OFDM performance in a fast Rayleigh faded channel with frequency error of 700Hz (perfect channel knowledge)

As seen above, the system's performance dramatically worsens when sizeable frequency error is introduced. To maintain a BER or 10^{-2} the system must compensate by boosting the average SNR by approximately 5 dB. In contrast, single carrier can easily overcome this problem by using techniques such as first order Costa loop.

2.8. Summary

In this chapter, we familiarized ourselves with different aspects of OFDM. We learn the basic concepts that make OFDM work including ISI, ICI; and how to overcome such interference with the use of a cyclic prefix. We introduced the system's susceptibility to PAPR and frequency jitter. We also presented results from our stripped down OFDM simulator based on an online resource. Now that it has been validated, it is ready to be used in more involved simulations.

Chapter 3. Turbo codes

3.1 Introduction

Turbo codes were first presented at the International Conference on Communications in 1993. Until then, it was widely believed that to achieve near Shannon's bound performance, one would need to implement a decoder with infinite complexity or close. Parallel concatenated codes, as they are also known, can be implemented by using either block codes (PCBC) or convolutional codes (PCCC). PCCC resulted from the combination of three ideas that were known to all in the coding community:

- The transforming of commonly used non-systematic convolutional codes into systematic convolutional codes.
- The utilization of soft input soft output decoding. Instead of using hard decisions, the decoder uses the probabilities of the received data to generate soft output which also contain information about the degree of certainty of the output bits.
- Encoders and decoders working on permuted versions of the same information. This is achieved by using an interleaver.

An iterative decoding algorithm centered around the last two concept would refine its output with each pass, thus resembling the turbo engine used in airplanes. Hence, the name Turbo was used to refer to the process.

3.2 Turbo Encoding

3.2.1 Recursive Systematic Convolutional Codes (RSC)

Convolutional encoding results from passing the information to be encrypted through a linear shift register as shown in Figure 3.1 below. The encoder shown here is nonsystematic because no version of the uncoded input is part of the output. Convolutional encoder can be represented by their generator polynomials. For the encoder below, g(1) = [111] and g(2) = [101].

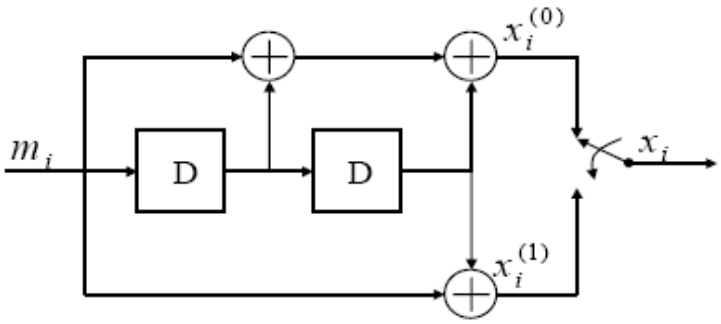

Figure 3.1 Constraint length K = 2 convolutional encoder

Convolutional encoding is a continuous process where the output depends on the K previous inputs of the encoder. The linear shift register introduces a deterministic component to the randomly generated input. This component can be tracked through a trellis, which we will introduce shortly. For Turbo codes, the recursive systematic convolutional codes were chosen as they exhibit better performance at low signal to noise ratios (SNR).

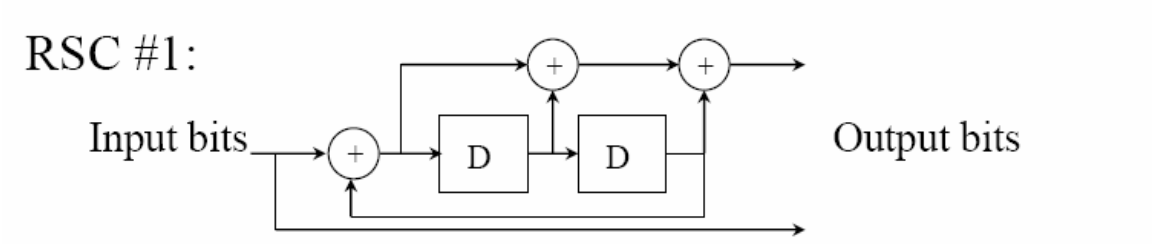

Figure 3.2 Recursive Systematic Convolutional encoder

For decoding purposes, convolutional encoders can be seen as finite state machines (FSM) with the content of the shift register indicating the state of the machine, and the outputs being a function of the current state as well as the input to the encoder. The code's behavior can also be described by using a trellis diagram. In a trellis diagram, all possible transitions between states are shown along side with the input and output associated with it. Transitions not drawn on the trellis do not represent valid codewords and are therefore classified as errors.

3.2.2 Encoding of Parallel Concatenated Convolutional Codes

Turbo codes were presented by Berrou, Glavieux and Thitimajshima [11] in 1993. They are the result of the parallel concatenation of two or more RSC. For the scope of this thesis, we will only consider the case where two RSC are used. The information is encoded by the first recursive systematic encoder, interleaved and then encoded by the second RSC at the same time. The size of the interleaver determines the length of the codeword.

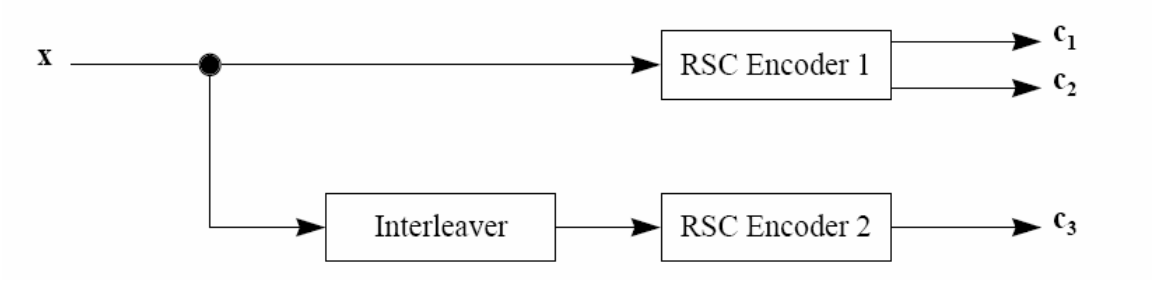

Figure 3.3 Turbo encoding scheme

A block MxN interleaver can be used. In that case, the M bits would be fed into the interleaver column-wise and N bits would be read out row-wise. The interleaver would then alleviate burst errors by spreading them so that one error occurs every M bits and thus reduce the correlation between it's input and output. The presence of the interleaver adds to a difficult trellis termination problem. The trellis of a conventional convolutional encoder can be terminated by appending a few zeros at the end of the input sequence. For the recursive variety of encoders, the termination bits depend on the on the state of the encoder as we are trying to force it back to the zero state. Therefore, the tails bits cannot be known until the encoder has completely encoded the data. Moreover, the additional bits used for trellis termination of RSC #1 will be interleaved and therefore useless in terminating RSC #2. They become data for the latter. One can see how difficult it becomes to successfully compute a sequence of tail bits that will terminate both trellis [13]. One solution is to only terminate the trellis of RSC #1 and leave the other open [12]. This alternative is perhaps the easiest to implement and we will chose it in our simulation. One can modify a turbo code (mother code) in order to achieve different code rates with the resulting code (punctured code). Puncturing patterns decide which parity bits are to be retained after puncturing. Commonly used patterns include selecting the x^{th}

bit every 2*k parity bits, k > 0. For most rates, when commonly used patterns are applied to both parity sequences, turbo codes exhibit very good performance [30].

3.3 Decoding Algorithms

When using turbo codes, the decoding process, unlike other coding schemes is iterative. The algorithms feature two soft-input, soft-output (SISO) decoding blocks working in conjunction with one another. The following section describes the two major class of turbo decoding algorithms currently used to implement the SISO decoders: the soft output Viterbi algorithm [14 - 15] (VA/SOVA), and the maximum a posteriori algorithm [16-28] (MAP) also known as the BCJR algorithm, after Bahl, Cocke, Jelinek and Raviv, the authors of [16]. Both solutions are trellis-based.

The VA was designed to find the most probable *sequence* of states **s** given the received symbols sequence **y**

$$\hat{s} = \arg\left\{\max_{s} P[s \mid y]\right\} \tag{3.1}$$

The states found by the VA must form a connected path through the trellis. For this reason, the VA is known to be better suited to minimize the frame error rate (FER) in communications systems. Conversely, the MAP is geared towards finding the most probable state s_i given the received symbol **y**

$$\hat{s}_i = \arg\left\{\max_{s} P[s_i \mid y]\right\} \tag{3.2}$$

The estimates generated by the MAP need not be connected at all. The MAP is best suited to minimize the bit error rate (BER). Turbo decoding works by independently estimating two individual processes. The two processes operate on the same data albeit

the second decoder uses an interleaved version of the original information. The decoding algorithm must take advantage of this fact by using one process output as a priori information for the other through soft-bits decisions. The end goal is to form log-likelyhood ratios (LLRs) which can be used to estimate the bit sequence by perfoming hard decision on them.

$$LLR = \Lambda_i = \ln\left(\frac{P[m_i = 1 \mid y]}{P[m_i = 0 \mid y]}\right) \tag{3.3}$$

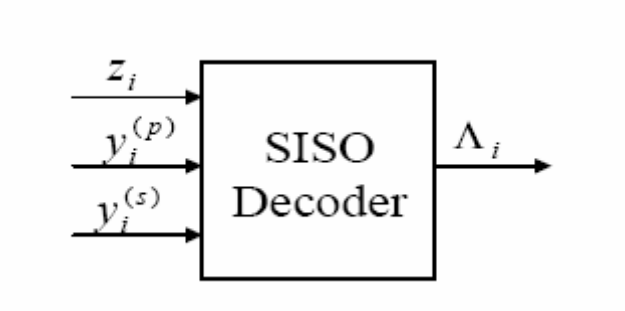

Figure 3.4 Soft-input soft-output module: z_i, the a priori values for information bits, $y_i^{(p)}$ the parity observations, $y_i^{(s)}$ the systematic observations, and Λ_i the a posteriori values

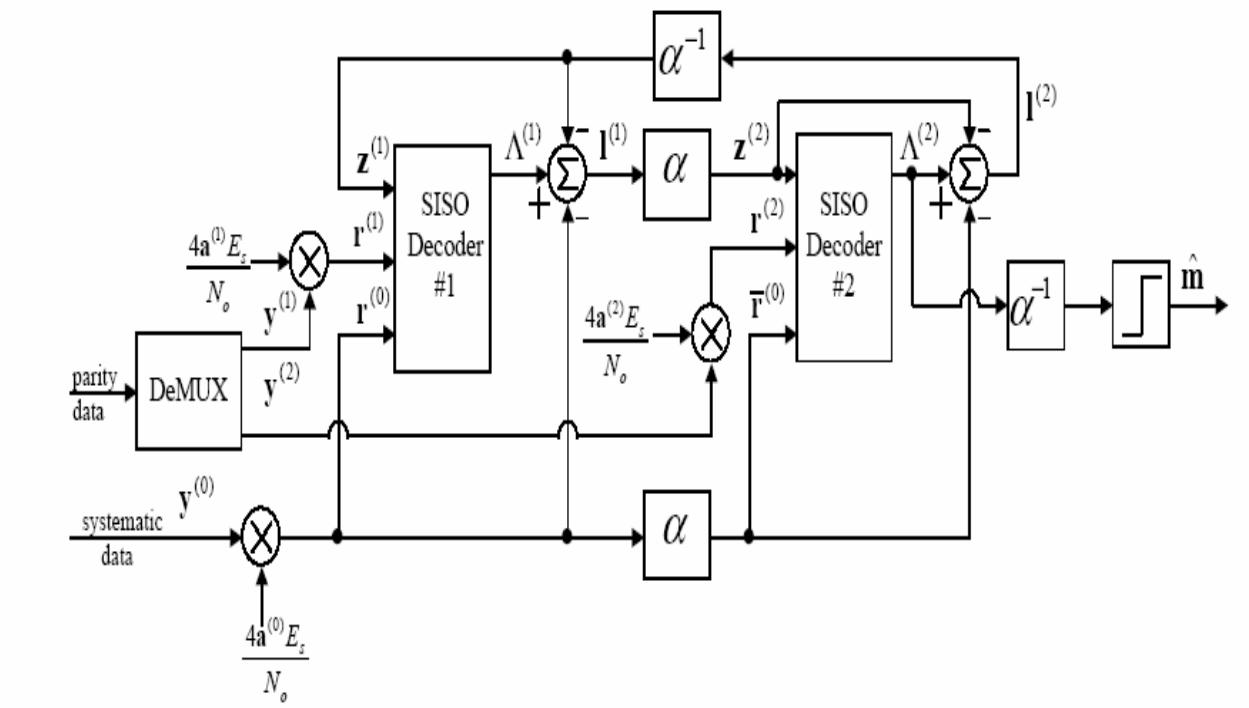

Figure 3.5 Schematic of the Turbo Decoder

The turbo decoder architecture is shown in Figure 3.5 above. Let use derive an expression for a binary phase shift keying (BPSK) modulated signal over a fading channel. Let $x = [010101....]$ be the binary sequence to be sent over the fading channel. Once modulated, the binary sequence becomes the BPSK modulated symbols sequence $X = [[-1111-1-1...]$. The channel impulse response is such that y, the signal observed at the receiver has the form

$$y(t) = aX + n(t) \tag{3.4}$$

Where *a* represent the fading amplitude and *n(t)* is the additive white Gaussian noise (AWGN) detected at the receiver with variance $\sigma^2 = \frac{N_0}{2E_s}$. According to [29], the output of the SISO decoder, the LLR, and be expressed as the sum of three entities

$$\Lambda_i = \frac{2a_i}{\sigma^2} y_i^{(s)} + z_i + l_i \tag{3.5}$$

With l_i being the extrinsic information, z_i, the information derived by the other decoder used as a priori information and the y term corresponding the systematic observations. One must note that to successfully decode the information, one must be careful to have the two SISO module exchange extrinsic information exclusively. We can use (3.5) to derive the expression for l_i

$$l_i = \Lambda_i - \frac{2a_i}{\sigma^2} y_i^{(s)} - z_i \tag{3.6}$$

One must also note that in the case of an AWGN channel, the a_i's, the channel statistics need not be included. Indeed, for the AWGN case, $a_i = 1$, and thus can be dropped from the equation.

The iterative turbo decoding algorithm functions as follows. The first decoder receives the systematic observations $y_i^{(s)}$, weighted by the appropriate channel statistics when it applies, along side with the systematic parity bits (generated by the first RSC) and the extrinsic information from the other decoder. Given those three pieces of information, SISO decoder 1 generates the first set of LLRs. Using (3.6) the extrinsic values are isolated and fed into an interleaver identical to the one used at the turbo encoder stage. The output of the interleaver is fed as a priori information to SISO decoder 2. The latter also takes as inputs interleaved versions of the systematic observations $y_i^{(s)}$, weighted by the appropriate channel statistics when it applies, and the parity bits (generated by RSC #2). The second decoder generates a seconds set of LLRs which are used to obtain the extrinsic values to be used as a priori information for decoder 1. These values must be deinterleaved so that they can be useful to decoder 1. The whole process is repeated for N

iterations. If the turbo code is punctured, zeros must be inserted at their corresponding spot in the parity bits sequences according to the puncturing pattern. The puncturing pattern must be know in advance and must match the one used at the encoder. This step is to be performed before the first iteration is to occur.

The output of the decoding process can be found by deinterleaving the output of the second decoder and performing hard-decisions. As (3.3) describes, the LLRs are the natural log of the ratio between the probability of the original bit m_i being equal to "1" given the observed bit y_i and the probability of the original bit m_i being equal to "0" given the observed bit y_i. From this statement, it can be inferred that if the LLR is positive, the ratio inside the natural log is greater than 1. Hence, the probability of the message bit being "1" was greater than the alternative. This would lead to a hard decision equal to "1". On the other hand, if the LLR is negative, the fraction inside the natural log is less than 1. The probability of the message bit being "0" was greatest.

3.3.1 The soft output Viterbi Algorithm (SOVA)

In this thesis, we have limited the scope of our research to the MAP algorithm. However, before elaborating on our findings, we will briefly present the SOVA.

The VA is a decoding algorithm originally derived to ensure maximum likelihood detection of convolutional coded schemes. A trellis can fully represent the decoding process. As (2.1) describes,, the VA finds the most likely path through the trellis, given the received signal bits. Using Bayes theorem, (3.1) becomes

$$\hat{s} = \arg\left\{ \max_{s} \frac{P[y \mid s]P[s]}{P[y]} \right\} \tag{3.7}$$

We can ignore the denominator in (3.7) as it is the same for all $\hat{s}$.

$$\hat{s} = \arg\left\{\max_{s} P[y \mid s]P[s]\right\} \tag{3.8}$$

To solve (3.8), we could calculate the joint probability of y and s for all possible states in the trellis or instead, determining the best code sequence by elimination. The conditional probability that the process is in a state s_i given all the previous states is the same as the conditional probability that the process is in s_i given the last state

$$P[s_{i+1} \mid s_0 \ldots . s_i] = P[s_{i+1} \mid s_i] \tag{3.9}$$

Also, the conditional property of the observation y_i given the entire state sequence is the same as the conditional property of the observation y_i given the last state transition

$$P[y_i \mid s] = P[y_i \mid s_i \rightarrow s_{i+1}] \tag{3.10}$$

Given the last two equations, (3.8) can be written as

$$\hat{s} = \arg\left\{\max_{s} \sum_{i=0}^{L-1} \lambda(s_i \rightarrow s_{i+1})\right\} \tag{3.11}$$

$$\lambda(s_i \rightarrow s_{i+1}) = \ln P[y_i \mid s_i \rightarrow s_{i+1}] + \ln P[s_{i+1} \mid s_i] \tag{3.12}$$

$\lambda(s_i \rightarrow s_{i+1})$ is the branch metric associated with the transition $s_i \rightarrow s_{i+1}$. We can write the branch metric in terms of the transmitted symbols and messages that produce the state transitions.

$$\lambda(s_i \rightarrow s_{i+1}) = \ln P[y_i \mid x_i] + \ln P[m_i] \tag{3.13}$$

Where m_i and x_i are the message and output associated with the given state transition $s_i \rightarrow s_{i+1}$. P[m_i] is obtained from the a priori information z_i

$$P[m_i] = \begin{cases} \dfrac{e^{z_i}}{1+e^{z_i}} & for \quad m_i = 1 \\ \dfrac{1}{1+e^{z_i}} & for \quad m_i = 0 \end{cases} \tag{3.14}$$

$$\ln P\left[m^i\right] = z_i m_i - \ln\left(1 + e^{z_i}\right) \tag{3.15}$$

In a flat fading environment, the branch metric is

$$\lambda\left(s_i \to s_{i+1}\right) = z_i m_i - \left\{\ln\left(1 + e^{z_i}\right) + \frac{1}{2}\ln\left(\frac{\pi N_0}{E_s}\right)\right\} - \frac{E_s}{N_0}\sum_{q=0}^{n-1}\left[y_i^{(q)} - a_i^{(q)}\left(2x_i^{(q)} - 1\right)\right]^2 \tag{3.16}$$

When the signal to noise ratio of the noisy channel is small, the third term doesn't factor heavily into the determination of λ. The first two factors rely on the extrinsic information produced by the other decoder which is used as a priori information in the current decoder. They influence the branch metric heavily. On the other hand, when the signal to noise ratio of the noisy channel is large, the λ calculation relies heavily on the channel observations, the third term in (3.16). We end our analysis of the SOVA here as this derivation of the branch metric will prove to be useful to us in understanding the MAP algorithm. [14] explores the VA in more detail. It can be modified according to [32] to produce a priori probabilities of the state transitions.

3.3.2 The Modified Maximum A Posteriori (MAP) or BCJR

There are two versions to the MAP. The type-I MAP, presented in [16] requires forward and backward recursion. The type-II MAP is the forward only variant of the algorithm. It is the most complex of the two and is best suited for continuous processing. We will present the type-I MAP. The MAP algorithm computes the a posteriori probability of each state transition given the noisy observation at the receiver. There is a one to one correspondence between a state transition and its corresponding code symbol. The states connected by the MAP-found state transition need not form a continuous path. The algorithm computes the a posteriori probabilities (APP) of each possible state transition

and chooses the one which is more likely (highest probability). In turbo decoding, the MAP finds the probabilities an individual message bit being either 1 or 0 given its noisy observation y. It puts them into LLR form and this information is exchanged between the two decoders until the last iteration, at which point a hard decision is performed.

The MAP algorithm starts by finding the probability of each valid state transition $P[s_i \to s_{i+1} \mid y]$ given the noisy observation y. If we use the definition of conditional probability

$$P[s_i \to s_{i+1} \mid y] = \frac{P[s_i \to s_{i+1}, y]}{P[y]} \tag{3.17}$$

Where $P[s_i \to s_{i+1}, y]$ is the joint probability of the state transition $s_i \to s_{i+1}$ and y, the observation corrupted by noise. The numerator of the right term of (3.17) can be partitioned into

$$P[s_i \to s_{i+1}, y] = \alpha(s_i)\gamma(s_i \to s_{i+1})\beta(s_{i+1}) \tag{3.19}$$

Where

$$\alpha(s_i) = P[s_i, (y_0 \ldots . y_{i-1})] \tag{3.20}$$

$$\gamma(s_i \to s_{i+1}) = P[s_{i+1}, y_i \mid s_i] \tag{3.21}$$

$$\beta(s_{i+1}) = P[(y_{i+1}, \ldots, y_{L-1}) \mid s_{i+1}] \tag{3.22}$$

$\gamma(s_i \to s_{i+1})$ is the branch metric associated the transition $s_i \to s_{i+1}$.

$$\gamma(s_i \to s_{i+1}) = P[s_{i+1} \mid s_i]P[y_i \mid s_i \to s_{i+1}] = P[m_i]P[y_i \mid x_i] \tag{3.23}$$

We have already derived $P[m_i]$ in equation (3.14). $P[y_i \mid x_i]$ is a function of modulation and the channel model. According to Baye' s theorem,

$$P[y_i \mid x_i] = \frac{\Pr\{x_i \mid y_i\}\Pr\{x_i\}}{\Pr\{y_i\}} = C \cdot \Pr\{x_i \mid y_i\} \tag{3.24}$$

Where C is constant for a particular codeword and as a result is to be ignored from future calculations. The channel reliability factor $R_i^{(q)}$ is

$$R_i^{(q)} = \ln\left[\frac{\Pr\left\{x_i^{(q)} = 1 \mid y\right\}}{\Pr\left\{x_i^{(q)} = 0 \mid y\right\}}\right] \tag{3.25}$$

with $q \in \{0,...n-1\}$ of a 1/n RSC encoder. Baye's theorem dictates that the a posteriori probability $\Pr\left\{x_i^{(q)} = b \mid y\right\}, \quad b \in \{0,1\}$, can be expressed in terms of the a priori probability $\Pr\left\{y_i^{(q)} \mid x_i^{(q)} = b\right\}$. Since

$$\Pr\left\{y_i \mid x_i\right\} = \frac{1}{\sqrt{\pi \frac{N_0}{E_s}}} \exp\left\{-\frac{E_s}{N_0}\left[y_i^{(q)} - a_i^{(q)}\left(X_i^{(q)}\right)\right]\right\} \tag{3.26}$$

Then the channel reliability factor simplifies down to

$$R_i^{(q)} = 4a_i^{(q)} \frac{E_s}{N_0} y_i^{(q)} \tag{3.27}$$

The AP probabilities $\Pr\left\{x_i^{(q)} = b \mid y_i^{(q)}\right\}, \quad b \in \{0,1\}$ can be written as

$$\Pr\left\{x_i^{(q)} \mid y_i^{(q)}\right\} = \frac{\left(e^{R_i^{(q)}}\right)^{x_i^{(q)}}}{1 + e^{R_i^{(q)}}} \tag{3.28}$$

Consequently,

$$\gamma\left(s_i \to s_{i+1}\right) = \frac{\left(e^{Z_i}\right)^{m_i}}{1 + e^{Z_i}} \prod_{q=0}^{n-1} \frac{\left(e^{R_i^{(q)}}\right)^{x_i^{(q)}}}{1 + e^{R_i^{(q)}}} \tag{3.29}$$

The denominator of (3.29) remains constant for a given codeword. (3.29) simplifies down to

$$\gamma\left(s_i \to s_{i+1}\right) = \left(e^{Z_i}\right)^{m_i} \prod_{q=0}^{n-1} \left(e^{R_i^{(q)}}\right)^{x_i^{(q)}} \tag{3.30}$$

The probability $\alpha(s_i)$ is found by forward recursion

$$\alpha(s_i) = \sum_{s_{i-1} \in A} \alpha(s_{i-1}) \gamma(s_{i-1} \to s_i) \tag{3.31}$$

A is the set of all the states s_{i-1} connected to state s_i. $\beta(s_i)$ is found through backward recursion.

$$\beta(s_i) = \sum_{s_{i-1} \in B} \beta(s_{i+1}) \gamma(s_{i+1} \to s_i) \tag{3.32}$$

Where *B* is the set of states s_{i+1} connected to state s_i.the log-likelihood equation in (3.3) becomes

$$LLR = \Lambda_i = \ln\left(\frac{\sum_{S_1} \alpha(s_i) \gamma(s_i \to s_{i+1}) \beta(s_{i+1})}{\sum_{S_0} \alpha(s_i) \gamma(s_i \to s_{i+1}) \beta(s_{i+1})} \right) \tag{3.33}$$

The MAP can calculate the a posteriori probabilities for each bit. Unfortunately the algorithm is computational intensive, and susceptible to round off errors. We can alleviate these problems by performing the MAP in the log domain. It becomes the log-MAP algorithm. Indeed, the LLRs consist in a sum of logarithms so we can apply the logs much earlier in the computation, changing what used to be multiplications operations into additions and divisions into subtractions. There are two algorithms that take advantage of this property of the logarithm, the log-MAP and the max-log-MAP. Consider the equation below from [33]

$$\ln\left(e^{\delta_1} + ... + e^{\delta n}\right) \approx \max_{i \in \{1 \ldots n\}} \delta_i \tag{3.34}$$

The logarithm of the sum of exponentials is replaced by n-1 maximum operations on the arguments of the exponentials. This approximation is characteristic of the max-log-MAP algorithm, for which the branch metric now looks like

$$\begin{aligned}\bar{\gamma}(s_i \to s_{i+1}) &= \ln P[m_i] + \ln P[y_i \mid x_i] = \ln\left(e^{Z_i}\right)^{m_i} + \sum_{q=0}^{n-1} \ln\left(e^{R_i^{(q)}}\right)^{x_i^{(q)}} \\ &= m_i Z_i + \sum_{q=0}^{n-1} x_i^{(q)} R_i^{(q)}\end{aligned} \tag{3.35}$$

or

$$\bar{\gamma}(s_i \to s_{i+1}) = \begin{cases} \sum_{q=1}^{n-1} x_i^{(q)} R_i^{(q)}, & m_i = x_i^{(0)} = 0 \\ Z_i + R_i^{(0)} + \sum_{q=1}^{n-1} x_i^{(q)} R_i^{(q)}, & m_i = x_i^{(0)} = 1 \end{cases} \tag{3.36}$$

The $\bar{\alpha}(s_i)$ and $\bar{\beta}(s_i)$ become

$$\begin{aligned}\bar{\alpha}(s_i) &= \ln \alpha(s_i) \\ &= \ln\left\{ \sum_{s_{i-1} \in A} \exp\left[\bar{\alpha}(s_{i-1}) \bar{\gamma}(s_{i-1} \to s_i)\right] \right\} - \max_{s_{i-1} \in A}\left[\bar{\alpha}(s_{i-1})\right] \\ &\approx \max_{s_{i-1} \in A}{}^{*}\left[\bar{\alpha}(s_{i-1}) + \bar{\gamma}(s_{i-1} \to s_i)\right]\end{aligned} \tag{3.37}$$

Here, the max* operator is simply equal to the maximum of the arguments. In a similar fashion

$$\begin{aligned}\bar{\beta}(s_i) &= \ln \beta(s_i) \\ &= \ln\left\{ \sum_{s_{i-1} \in B} \exp\left[\bar{\beta}(s_{i+1}) \bar{\gamma}(s_1 \to s_{i+1})\right] \right\} - \max_{s_{i-1} \in B}\left[\bar{\beta}(s_{i+1})\right] \\ &\approx \max_{s_{i-1} \in B}{}^{*}\left[\bar{\beta}(s_{i+1}) + \bar{\gamma}(s_i \to s_{i+1})\right]\end{aligned} \tag{3.38}$$

Once the $\bar{\alpha}(s_i)$ and $\bar{\beta}(s_i)$ can be found for all the states in the trellis, the LLR has the following form

$$\begin{aligned}\Lambda_i &= \ln\left\{\sum_{S_1}\exp\left[\overline{\alpha}(s_i)\overline{\gamma}(s_i \to s_{i+1})\overline{\beta}(s_{i+1})\right]\right\} \\ &\quad - \ln\left\{\sum_{S_0}\exp\left[\overline{\alpha}(s_i)\overline{\gamma}(s_i \to s_{i+1})\overline{\beta}(s_{i+1})\right]\right\} \\ &\approx \max_{S_1}{}^{*}\left[\overline{\alpha}(s_i)\overline{\gamma}(s_i \to s_{i+1})\overline{\beta}(s_{i+1})\right] \\ &\quad - \max_{S_0}{}^{*}\left[\overline{\alpha}(s_i)\overline{\gamma}(s_i \to s_{i+1})\overline{\beta}(s_{i+1})\right]\end{aligned} \tag{3.39}$$

Because of the approximation we applied (3.34), the max-log-MAP is sub-optimal and yields inferior soft results compared to the MAP algorithm. The problem is to calculate exactly the logarithm of the sum of exponentials. We can refine our calculation of (3.34) by using the Jacobian logarithm [33]

$$\begin{aligned}\ln\left(e^{\delta_1} + e^{\delta_2}\right) &= \max\left(\delta_1 + \delta_2\right) + \ln\left(1 + e^{-|\delta_2 - \delta_1|}\right) \\ &= \max\left(\delta_1 + \delta_2\right) + f_c\left(|\delta_1 - \delta_2|\right)\end{aligned} \tag{3.40}$$

In (3.40), f_c is called the correction function, the difference in implementation between the max-log-MAP and the log-MAP. At each step made by the max-log-MAP, the correction function is applied, in effect loosing some of the max-log-MAP lower complexity. This can be alleviated by storing values of f_c in a look up table. The table would only be a short, one dimensional because the computation is a function of the absolute value of the difference between δ_1 and δ_2.

The algorithm for the max-log-MAP and the log-MAP is computed in three steps. Perform the forward recursion to calculate the αs. Perform the backward recursion to calculate the βs. Use those results to find the LLRs. The details are listed below.

For the derivation below, j represents the number of states s_i from 0 to 2^M-1, where 2^M is the total number of states in the trellis.

$$j \in \left\{0 \ldots 2^M - 1\right\}$$

1. Initialize $\overline{\alpha}(s_i) = \begin{cases} 0 & if \quad j = 0 \\ -\infty & otherwise \end{cases}$

2. Calculate $\overline{\alpha}(s_i) = \underset{s_{i-1}=s_j \in A}{\max^*}\left[\overline{\alpha}(s_{i-1}) + \overline{\gamma}(s_{i-1} \rightarrow s_i)\right]$ where A is the set of all states s_{i-1} connected to state s_i (until you've reached the end of the trellis). For the max-log-MAP, $\max^*(\delta_1, \delta_2) = \max(\delta_1, \delta_2)$. For the log-MAP,

   $\max^*(\delta_1, \delta_2) = \max(\delta_1, \delta_2) + f_c(|\delta_1 - \delta_2|)$

3. Initialize $\overline{\beta}(s_i) = \begin{cases} 0 & if \quad j = 0 \\ -\infty & otherwise \end{cases}$ if the trellis is terminated. If the trellis is not terminated $\overline{\beta}(j, L) = 0 \quad for \quad all \quad j$

4. Calculate $\overline{\beta}(s_i) = \underset{s_{i+1}=s_j \in B}{\max^*}\left[\overline{\beta}(s_{i+1}) + \overline{\gamma}(s_i \rightarrow s_{i+1})\right]$ where B is the set of states s_{i+1} that are connected to state s_i (until you've reached the beginning of the trellis). The same rules detailed on point 2 about the max* operator apply here.

5. Finally, the LLR is computed in the following fashion

   $$\Lambda_i = \underset{S_1}{\max^*}\left[\overline{\alpha}(s_i) + \overline{\gamma}(s_i \rightarrow s_{i+1}) + \overline{\beta}(s_{i+1})\right] - \underset{S_0}{\max^*}\left[\overline{\alpha}(s_i) + \overline{\gamma}(s_i \rightarrow s_{i+1}) + \overline{\beta}(s_{i+1})\right]$$

   where S_1 is the set of all state transitions associated with the bit "1" and S_0 is the set of all state transitions associated with the bit "0".

3.4 Turbo codes performance analysis

3.4.1 Turbo code implementation

The following section describes the steps taken in our turbo code simulation. We used two identical half rate recursive systematic convolutional encoders with constraint length K = 3 and generator polynomial shown below in matrix form

$$g = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \tag{3.41}$$

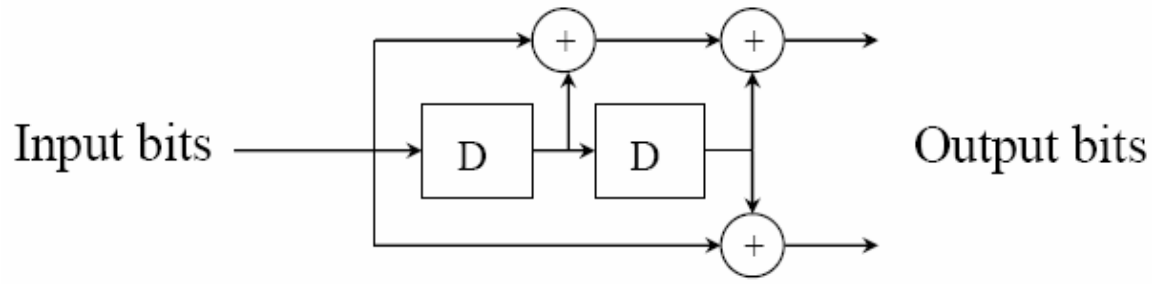

Figure 3.6 Feed-forward representation of r=1/2 RSC

The constraint length of our turbo code is a parameter we can change in the simulation but the program is extremely time-consuming. We tried to gain time where we could by keeping the complexity of the simulator to a near minimum. It was said earlier that turbo codes were not easily terminated due to the presence of the interleaver. For our implementation, we chose to terminate the trellis of the first RSC (Figure 3.3) and to leave the second trellis open. [35] confirms that there is no true performance difference between terminated and non-terminated schemes as soon as the block size used reaches a few hundred bits, which we have always exceeded in simulation work.

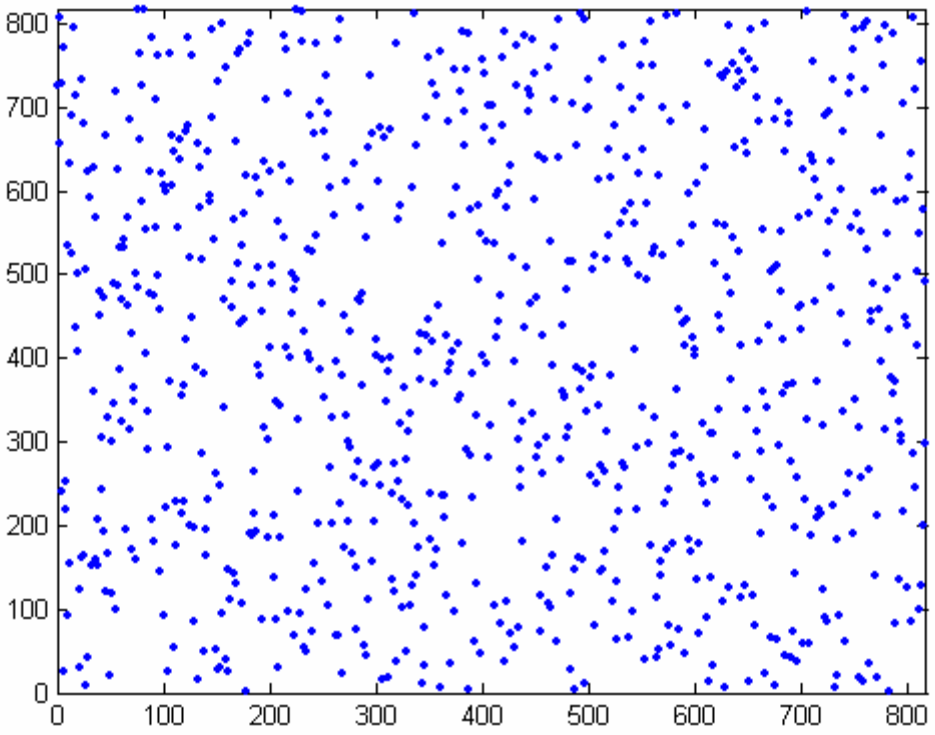

Figure 3.7 Pseudorandom interleaver 2-D scatter plot

The type of interleaver used in a turbo encoder can seriously hinder the performance of the constructed turbo code [36, p. 317]. The output of the interleaver must be as close to random as possible. In our simulation, we chose an interleaver depth equal to the number of bits to be encoded. We implemented the interleaver by using the Gaussian random number generator already available in MATLAB. Figure 3.7 shows the scatter plot of the interleaver we were able obtain. The interleaver function generates outputs with no apparent correlation between them. The pseudo-random generator is reliable and well suited for this task. We also implemented a puncturing function which allowed us to increase the turbo code rate from 1/3 to 1/2. The puncturing function modified the parity bits sequence of both RSCs. The first RSC's odd parity bits are punctured, while the second RSC's even parity bits are punctured. This leaves us with two sets of parity bits

have the size to the systematic bits sequence. We then interleave the two parity bits sequences to obtain a mixed sequence with parity information from both RSC and the only parity sequence. This forms the rate 1/2 code.

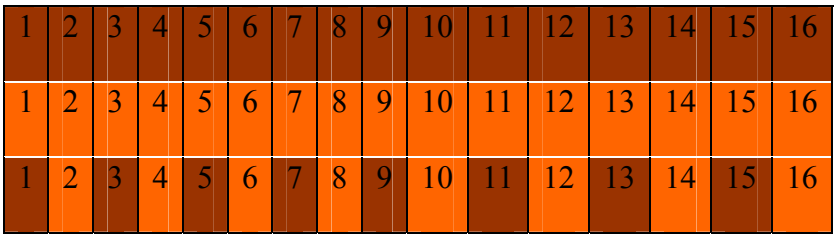

Figure 3.8 Puncturing pattern of form r=1/2 TC

At the receiver, we used the max-log-MAP to implement each SISO decoder. The choice of the max-log-map was also an attempt to reduce computational intensity by choosing the scheme that would require the least amount of processor power possible.

3.4.2 Performance of Turbo Codes

In an AWGN channel, the BER of a convolutional code can be bounded (upper) using the union bound technique

$$P_b \le \sum_{i=1}^{2^N} \frac{w_i}{N} Q\left(\sqrt{d_i \frac{2RE_b}{N_0}}\right) \tag{3.42}$$

Where w_i and d_i are the ith codeword information weight and total Hamming weight. The average information weight per codeword

$$\widetilde{w}_d = \frac{W_d}{N_d} \tag{3.43}$$

Where W_d is the total information weight of all codewords of weight d and N_d is the total number of codewords (multiplicity) of weight d. This yields

$$P_b \le \sum_{d=d_{free}}^{2(v+N)} \frac{N_d \widetilde{w}_d}{N} Q\left(\sqrt{d \frac{2RE_b}{N_0}}\right) \tag{3.44}$$

with d_{free} being the free distance of the code. The performance of a turbo code can also be bounded by (3.44). However, for moderate and high SNRs, (3.44) is dominated by the free distance term. Hence, the asymptotic performance of a turbo code can be approximated by

$$P_b \approx \frac{N_{free}\widetilde{w}_{free}}{N} Q\left(\sqrt{d_{free}\frac{2RE_b}{N_0}}\right) \tag{3.45}$$

Where N_{free} is the multiplicity of free distance codewords and $\widetilde{w}_{free}$ is the average weight of the information sequences causing free distance codewords. The free distance of a turbo code can be calculated by simulation [37]. The curve described by (3.45) is called the free distance asymptote. This limit is also referred to in turbo code literature as an *error floor* for turbo code performance. The phenomenon can be manipulated in two ways. While keeping the free distance and multiplicity of the code, we can increase the length of the interleaver N, which would in turn reduce the overall value of P_b. This would result in a collective lowering of the asymptote without changing the slope of the region. Conversely, decreasing the interleaver size while keep the other parameters constant would result in raising the error floor. The turbo code performance curve would flatten out at lower SNRs and yield worse BER performance. To change the slope of the asymptote, we must keep the size of the interleaver and multiplicity constant while tampering with the free distance. The slope of the error floor could be made steeper by increasing d_{free} while decreasing d_{free} results in a flatter slope. Typically, the performance curve of turbo codes is fairly flat in the moderate to high SNR region, thus confirming the codes relatively small free distance. Turbo codes exhibit near capacity performance at very low SNRs. The use of a pseudorandom interleaver in conjunction with a parallel

concatenation scheme yields a sparse distance spectrum, which in turn is responsible for turbo codes near capacity performance at low values of SNRs. Each term in the union bound equation (3.44) represents the distance spectrum information, or spectral line, associated with a particular distance d. For turbo codes, the free distance asymptote dominates the performance of the code for all SNRs. Such distance spectra is referred to as sparse or thin. The ability of turbo codes to follow the free distance asymptote is the cause for such a steep performance curve at low SNRs.

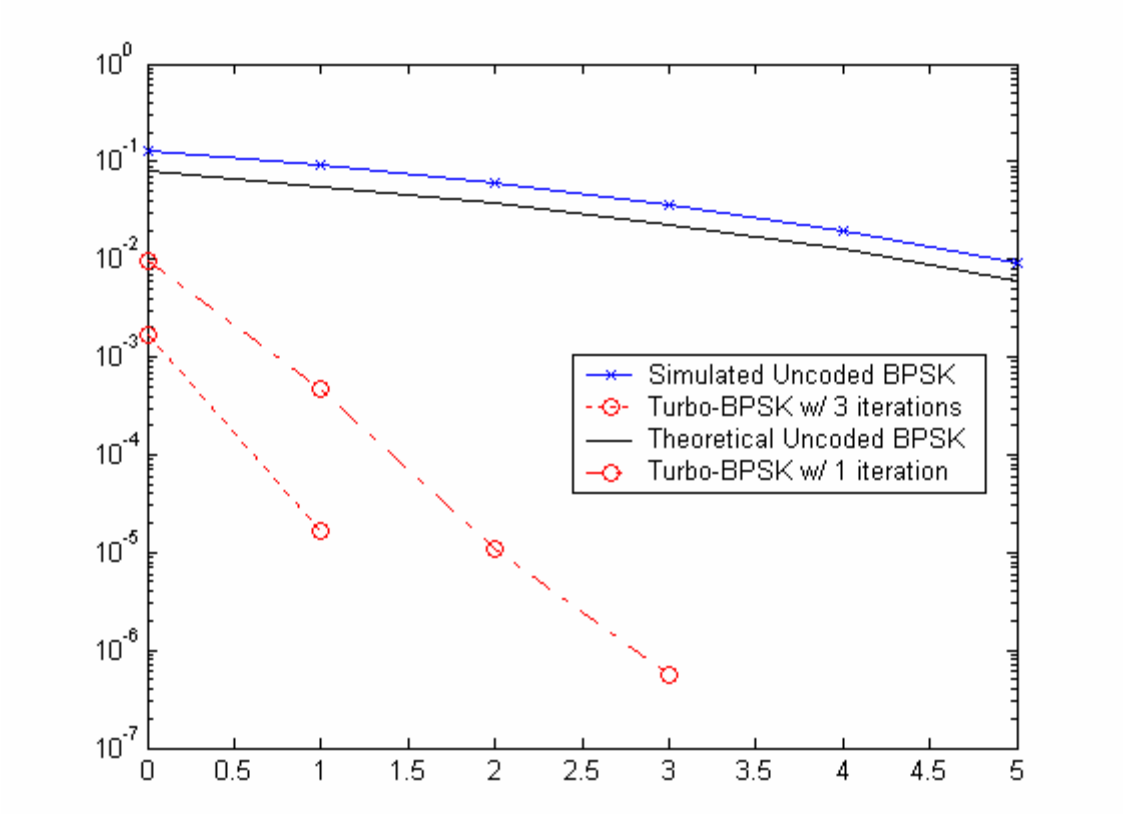

Figure 3.9 Rate 1/3, constraint length K = 2, Turbo coded BPSK performance.

3.5 Summary

In this chapter, we introduced the latest class of block codes called turbo codes. We provided an overview of turbo encoding. We discussed the two classes of recursive decoding algorithms used for decoding. We briefly covered the SOVA but mainly focused on the MAP. We derived a step by step algorithm for the MAP and proceeded to apply it. We presented our implementation of the turbo codes by showcasing the pseudo-random interleaver we chose, as well as the puncturing pattern we picked. We then explored the theoretical performance of turbo codes and uncovered why the work so well at low SNRs. We also were able to determine the reason why they should not be considered for applications with moderate to high SNRs.

Chapter 4. The mobile wireless channel and Adaptive Modulation

4.1 The mobile wireless channel

Wireless information exchange is highly limited by the state of the channel. Traditional communication systems are designed with an adequate margin in the link budget to guarantee functionality even under worse case scenarios, as the medium used is out of our control. This method hardly takes advantage of the channel's full capacity. To avoid such waste of scarce resources, communications engineers have moved towards adaptive schemes. Such systems automatically vary the transmit power and/or rate, loading the medium when favorable conditions allow it, and avoiding the squander of resources when deep fades hinder successful signal transmission. One must keep in mind that such schemes only works as long as the channel does not change faster than it can be estimate and fed back to the transmitter. If it does, other means should be explored to mitigate the effects of fading.

Different factors can make signal detection at the receiver problematic. There is of course a natural loss of signal power (path loss) proportional to the inverse of the square of the distance between transmitter and receiver. The existence of reflective material along the signal path leads to the emergence of multipaths. The phenomenon occurs when the transmit signal bounces off reflective surfaces on its way to the receiver. This leads to the presence of multiple versions of the transmit message at the receiver. Each replica has a time delay associated with it. The longer the distance the signal had to travel, the more obstacles along the way, the larger the time delay associated with a particular multipath.

Consequently, the times of arrival are scattered over an interval called the delay spread of the channel. When the difference between all the delays is negligible, the system experiences flat fading. When the delay spread is sizeable, the weighted versions of the original transmit signal interfere with one another, thus causing frequency selective fading. If the received signal consists of a dominant line of sight (LOS) component and several weighted replicas delayed in time, the system suffers from Rician fading. If unfortunately no LOS component is detected, the receive signal consists of multipaths exclusively. We have Rayleigh fading. The wideband nature of an OFDM based system guarantees that the OFDM symbol period is greater than coherence time. The impulse response of a wireless mobile channel can be modeled as a discrete time finite impulse response filter [39-47]

$$h(\tau;t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n(t)) \tag{4.1}$$

For wireless local area network (WLAN) applications, the channel is assumed to be quasistationnary, i.e. it does not change during the duration of the packet. Such an assumption allows us to remove the time dependency in (3.1) which becomes

$$h(\tau) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n) \tag{4.2}$$

The time delays are assumed to have infinite granularity. For the purpose of the model, the time axis has been divided into bins or time slots. All the multipaths present in a bin or slot are averaged and represented by a single impulse with magnitude α_n associated with a single arrival time τ_n. By plotting equation (4.2), we can obtain the channel power delay profile (PDP). Figure 4.1 below shows the negative exponential PDP model of a fast Rayleigh fading channel.

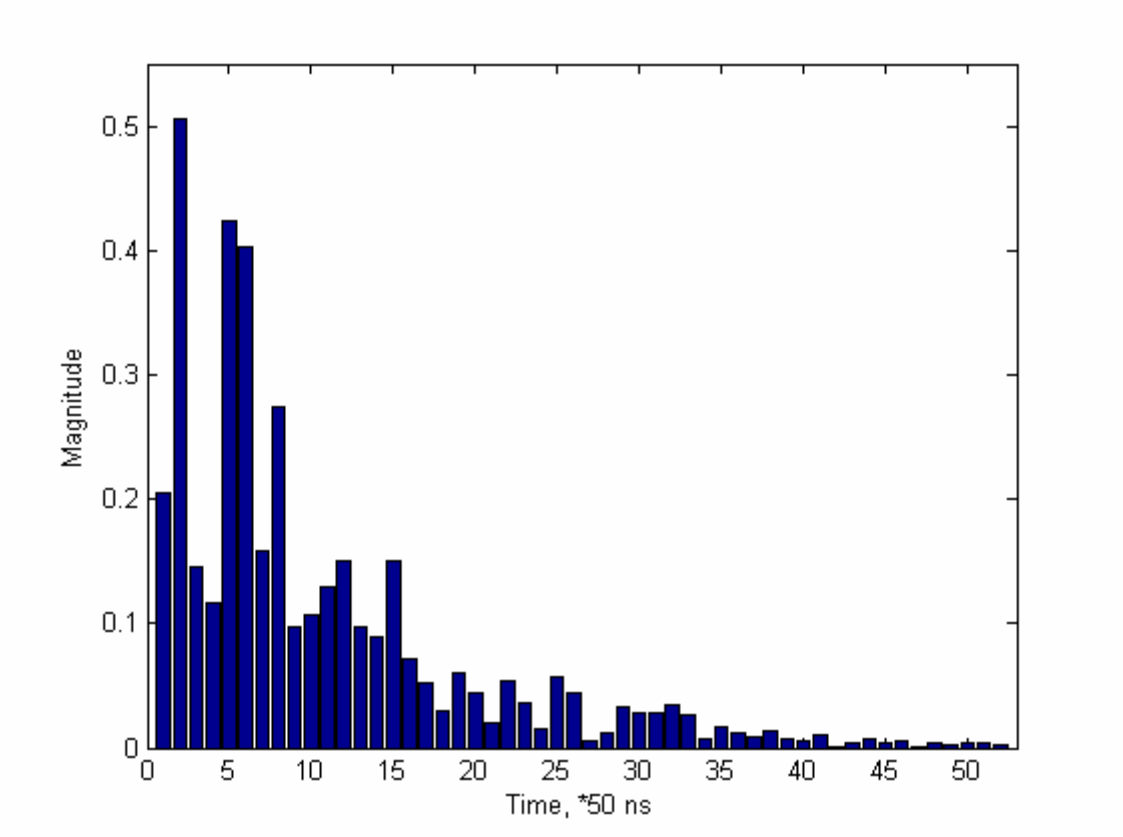

Figure 4.1 Power delay profile of a fast Rayleigh fading channel

The distribution of the arrival times is also important. In [67-68], the delays form a Poisson arrival-time sequence, used to model mobile urban channels. The time variant nature of the channel is due to the introduction of motion from a mobile antenna or motion of the scatterers (change in the environment) between the transmitter and the receiver. We will focus on the former. The maximum Doppler frequency generated by the relative motion between transmitter and receiver $f_{d,max}$ is defined as

$$f_{d,\max} = \frac{\Delta v}{\lambda}\cos\theta_i \tag{4.3}$$

Where Δv represents the difference in speed between the transmitter and the receiver, λ is the wavelength of the system, and θ_i is the angle between the direction of motion of the

mobile and the direction of arrival of the wave. Positive Doppler shifts occur when the distance between the transmitter and receiver is shortened. At the receiver, the operating frequency would be shifted up by $+f_{d,\,max}$ as a result. Negative Doppler shifts occur when the distance between the transmitter and receiver is increased. The receiver sees a downshift in operating frequency of $-f_{d,max}$ induced by the speed difference.

4.2 Simulating the wireless mobile channel

There are two general approaches to simulate the effects of a wireless mobile channel. The fastest way is to generate the channel impulse response (CIR) of the channel. It only requires that n multipath amplitudes and delays be created. The receive signal is generated by convolving this CIR with the transmit message. This process simulates the fading effects the desired channel. The received bits are obtained by adding additive white Gaussian noise AWGN to the former. This method is computationally efficient as few channel samples need to be generated to obtain the necessary results. However, the technique is not suited to study Doppler induced time variations. Doppler is a frequency phenomenon and for one to be able to control and/or vary it, one must take a frequency domain approach to channel generation.

A considerable body of work has been done is that area. Like most techniques, [39-46] revolve around two commonly used algorithms: Smith's frequency domain method and Rice's sum of sinusoids method. When generating the Rayleigh fading random variables using Rice's method, we use the following superposition of harmonic functions

$$\tilde{\mu}_i(t) = \sum_{n=1}^{N_i} c_{i,n} \cos\left(2\pi f_{i,n} t + \Theta_{i,n}\right), \; i = 1,2 \tag{4.4}$$

Where N_i designates the number of sinusoids, $\Theta_{i,n}$ is the phase variable equally distributed on the interval $(0,2\pi]$. The Doppler coefficients $c_{i,n}$, and the discrete Doppler frequencies $f_{i,n}$ are determined in the following fashion.

$$c_{i,n} = \sigma_0 \sqrt{\frac{2}{N_i}} \tag{4.5}$$

$$f_{i,n} = f_{\max} \sin\left[\frac{\pi n}{2N_i}\right] \tag{4.6}$$

The process can be changed from deterministic to stochastic by using randomly generated the $\Theta_{i,n}$ and discrete Doppler frequencies $f_{i,n}$. The central limit theorem states that in the limit $N_i \rightarrow \infty$, the probability density function of $\tilde{\mu}_i(t)$ tends to the normal distribution. However, for a finite number of sinusoids, it is merely an approximation (Figure 4.2). Finally, to generate the slow time varying channel, we add two such independent processes in quadrature. The resulting envelope has a Rayleigh PDF as shown in Figure 4.3.

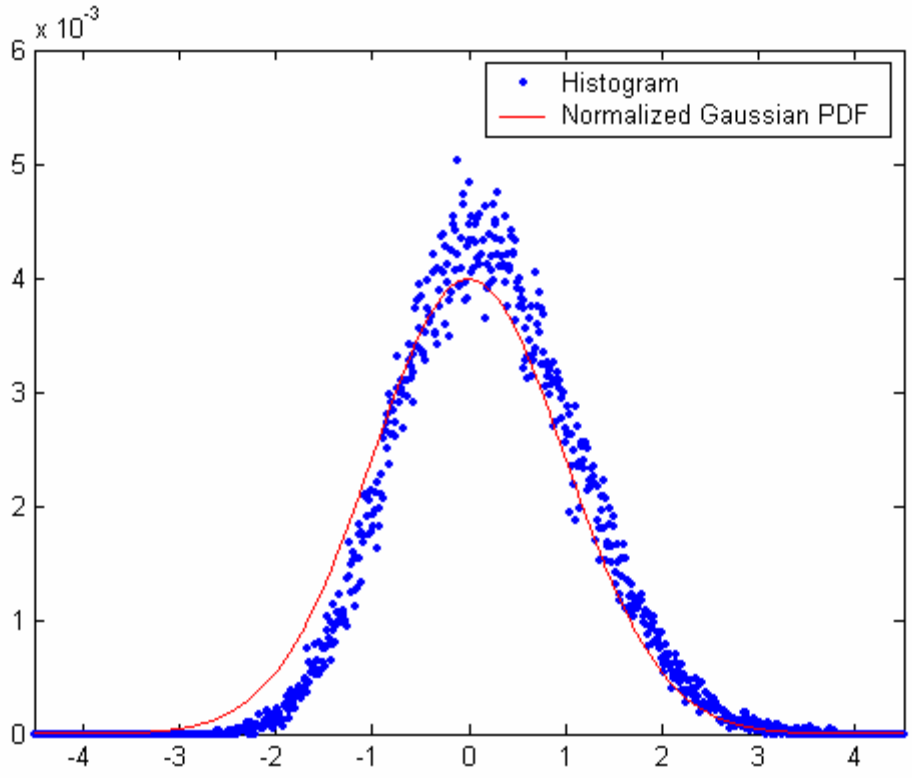

Figure 4.2 PDF of the simulated Rice process is approximately Gaussian when $N_i = 7$

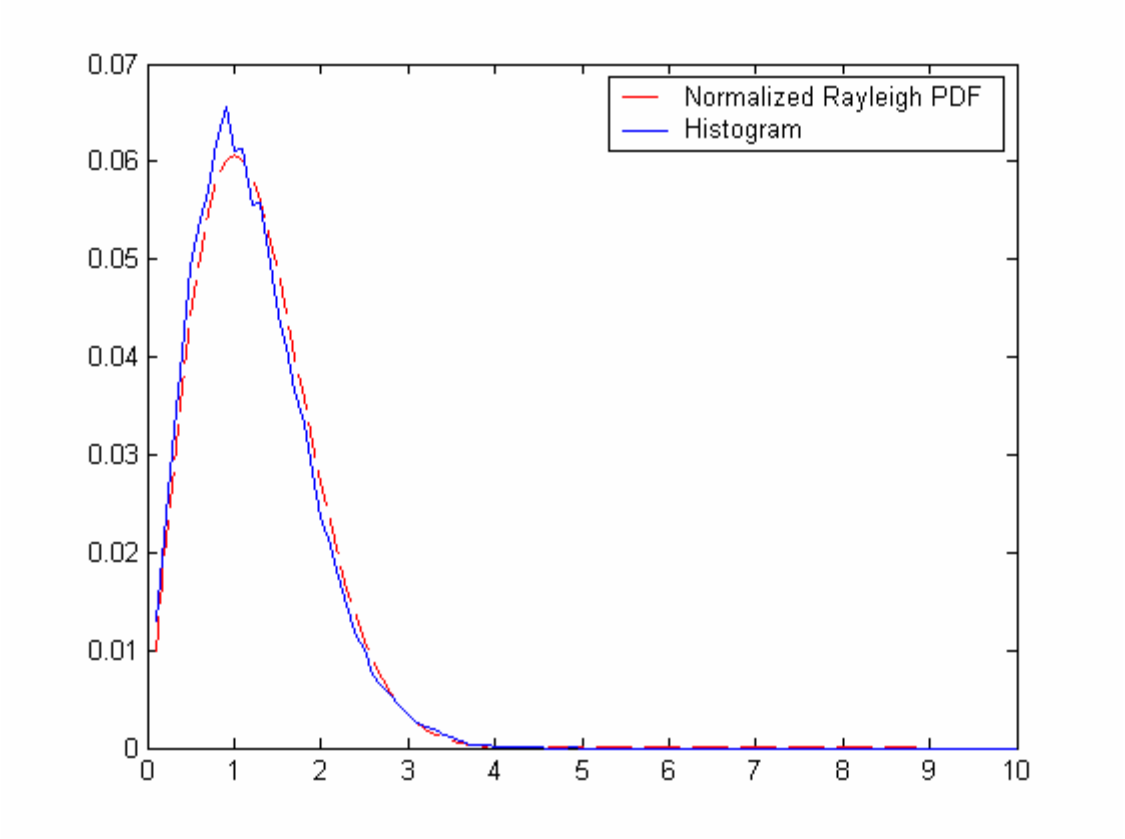

Figure 4.3 The PDF of envelope of the simulated Rice process is Rayleigh distributed

Rice's sum of sinusoids method has only one flaw. Multiple sinusoids must be generated for every channel realization. This process requires computer resources that cannot be neglected in a computationally intensive simulation.

In Smith's method, the algorithm starts by generating two independent, uncorrelated sequences of zero mean, unit variance, Gaussian random variables. According to Jakes' fading model, the autocorrelation of the time-variant transfer function is given by

$$\Phi_c(\Delta t) = J_0(2\pi f_m \Delta t) \tag{4.7}$$

where J( ) is the zeroth order Bessel function of the first kind and the maximum Doppler frequency is described by

$$f_m = \frac{v f_0}{c} \tag{4.8}$$

with v being the vehicular speed, f_0 the carrier frequency and c the speed of light, about $3*10^8$ m.s^{-1}. The Fourier transform of the autocorrelation function yields the following Doppler power spectrum

$$S_c(\lambda) = \begin{cases} \dfrac{1}{\pi f_m \sqrt{1-\left(\dfrac{f}{f_m}\right)^2}}, (|f| \le f_m) \\ 0. (|f| > f_m) \end{cases} \qquad (4.9)$$

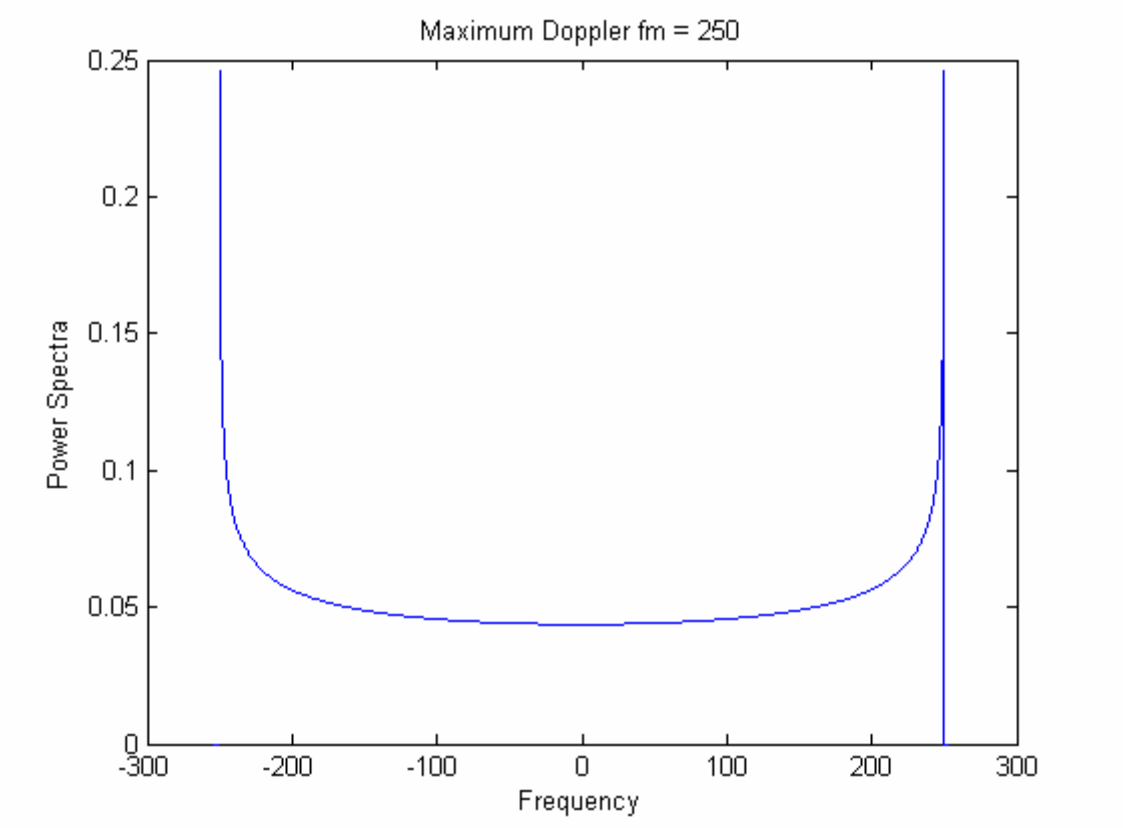

Figure 4.4 Doppler power spectra according to Jakes' Model

We then apply this filter to the Gaussian random variables previously generated. Because to the Doppler filter seen in Figure 4.4, the Gaussian variable obtain post-filtering no longer have zero mean and unit variance. We renormalize them before proceeding with any further step in the algorithm. We then take the inverse Fourier transform (implemented as the inverse fast Fourier transform, IFFT, in Matlab) of the resulting two sequences of variables. Finally the sequences are squared and added to each other, thus forming a set of Rayleigh distributed random variables. The latter is what should be used as the desired fading envelope. We verify our work by generating a histogram of our envelope and comparing it to the theoretical Rayleigh probability density function (PDF) as seen below.

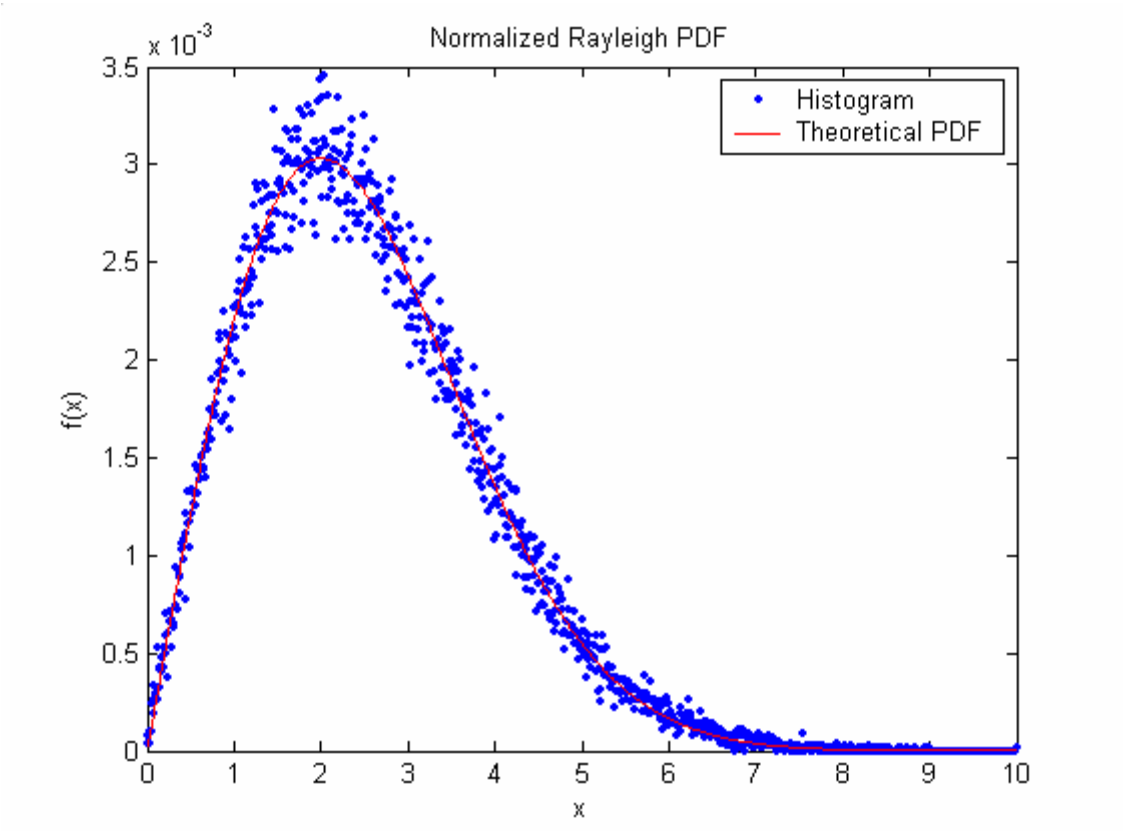

Figure 4.5 The PDF of a Rayleigh Channel

To successfully implement Smith's frequency domain method to generate the slow time varying channel, one must match the time scale of the channel with that of the system's symbol period. One way of doing so is to use a smaller number of variables to generate the Doppler filter. This is not practical. The filter must have a small enough resolution for the generated envelope to be reliable. The only alternative is to significantly upsample at the IFFT stage. This would allow us to truncate the envelope over the desired number of samples, thus maintaining the slow characteristic for the fading channel.

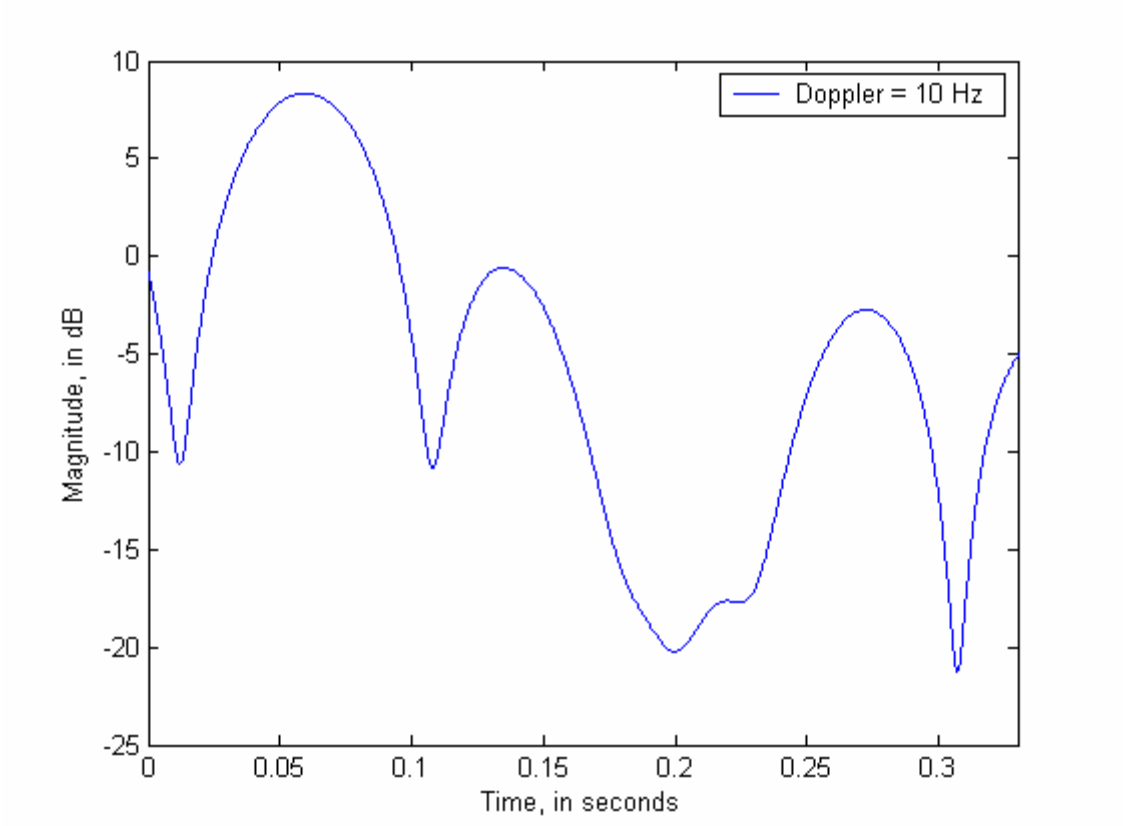

Figure 4.6 Original Rayleigh faded envelope for one packet

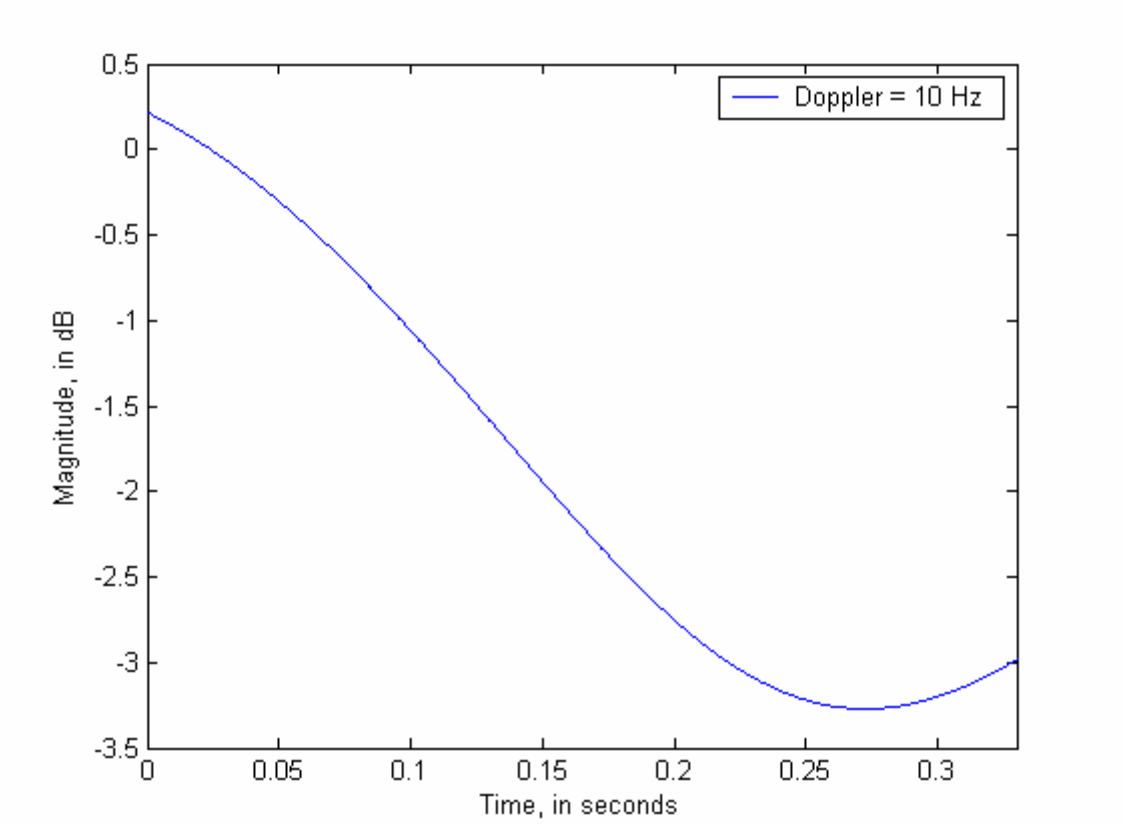

Figure 4.7 Upsampled Rayleigh faded envelope for one packet

Figure 4.6 shows the envelope of a slow, Rayleigh faded, time-varying channel as generated using Smith's method. One can see the relatively deep fades the envelope goes into. If such an envelope is applied to a single packet, the assumption of quasi-stationarity of the channel could never be made. On the other hand if we upsample as advised above (Figure 4.7), the range of the channel magnitude becomes much smaller, thus permitting the assumption of quasi-stationarity. One must also note that the channel looses is quasi-stationary properties as Doppler increases.

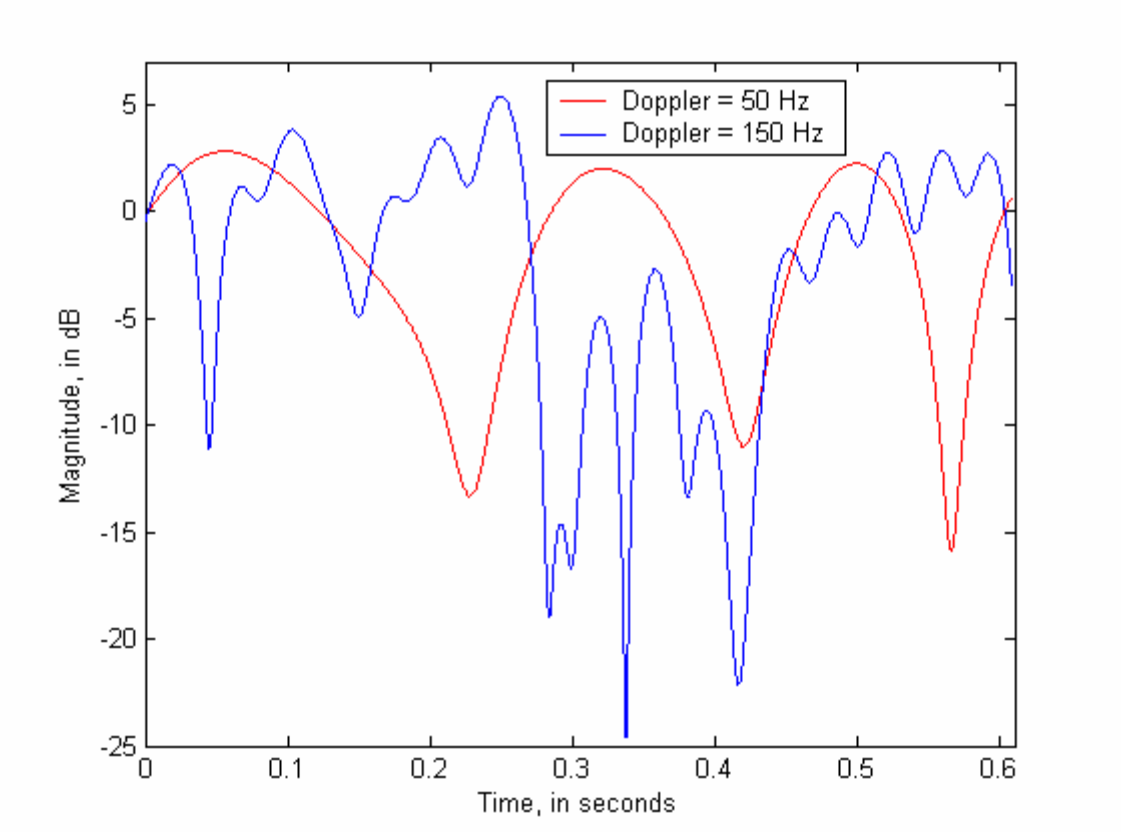

Figure 4.8 Effect of Doppler on Rayleigh envelope

As Figure 4.8 shows, the higher the Doppler, the deeper the fades over one symbol period. At 50 Hz Doppler, the range of magnitude reaches roughly 20 dBs. When the channel experiences 150 Hz Doppler, the magnitude range reached about 30 dBs. Higher Doppler values are synonymous with the channel varying faster than we can track it. It can not be considered quasi-stationary in such cases.

4.3 Channel Estimation

The goal of channel estimation is to estimate the frequency response of the channel the transmitted signal travels through before reaching the receiver. As we have previously seen, the impulse response of a time varying radio channel is typically described as a discrete time finite impulse response filter [47]

$$h(\tau;t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n(t)) \tag{4.10}$$

For wireless local area network (WLAN) applications, the channel is assumed to be quasi-stationnary, i.e. it does not change during the duration of the packet. Such an assumption allows us to remove the time dependency in (4.10) which becomes

$$h(\tau) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n) \tag{4.11}$$

The discrete time frequency response of the channel is obtained by taking the discrete Fourier transform (DFT) of the CIR.

$$H_k = DFT\{h_n\} \tag{4.12}$$

For data applications, most of the available bandwidth is dedicated to information transmission as the latter is done by short bursts. A quick, computationally inexpensive way of finding the channel estimate $\hat{H}_k$ must be implemented so as to free up resources for data exchange. To that effect, training data is included in the preamble and is transmitted using all available subcarriers. The preamble contains two identical long training sequences C_1 and C_2.

$$R_{l,k} = H_k C_k + W_{l,k} \tag{4.13}$$

Where l = 1,2. The received training sequences are the product between the training symbols and the channel response, corrupted by white Gaussian noise (4.13). The enhanced channel estimate $\hat{H}_k$ is obtained by averaging the two received sequences and multiplying the result by the conjugate of the original training symbols.

$$\hat{H}_k = \frac{1}{2}(C_1 + C_2)X_k^* \tag{4.14}$$

The coherence bandwidth can be described as the statistical measure of the range of frequencies over which the channel can be considered flat [69]. Such channel passes all frequency components of a signal with comparable gains and linear phases. The coherence bandwidth is approximately proportional the inverse of the length of the delay power spectrum (DPS) of the channel impulse response

$$B_{coherence} \propto \frac{1}{\tau_{\max}} \tag{4.15}$$

As long as the pilot subcarrier spacing is less than $B_{coherence}$, the channel response of neighboring OFDM subcarriers is highly correlated. Broadcast OFDM systems use this to their advantage and dedicate a select group of carriers (pilots) for continuous training data transmission. The final channel estimation is performed by interpolating the frequency response over the subcarriers used for data transmission. This technique is referred to as Pilot Symbol Aided Modulation or PSAM. [1] explains the procedure in more detail.

4.4 Signal to noise ratio (SNR)

Once we have obtained a reliable channel estimate, we are ready to calculate the signal to noise ratio (SNR). If we assume perfect knowledge of the noise variance, the SNR is fairly simple to derive. The SNR is equal to the square of the energy associated with a received symbol, divided by twice the noise variance (4.22).

$$\rho = \frac{|Y_k|^2}{2\sigma^2} \tag{4.22}$$

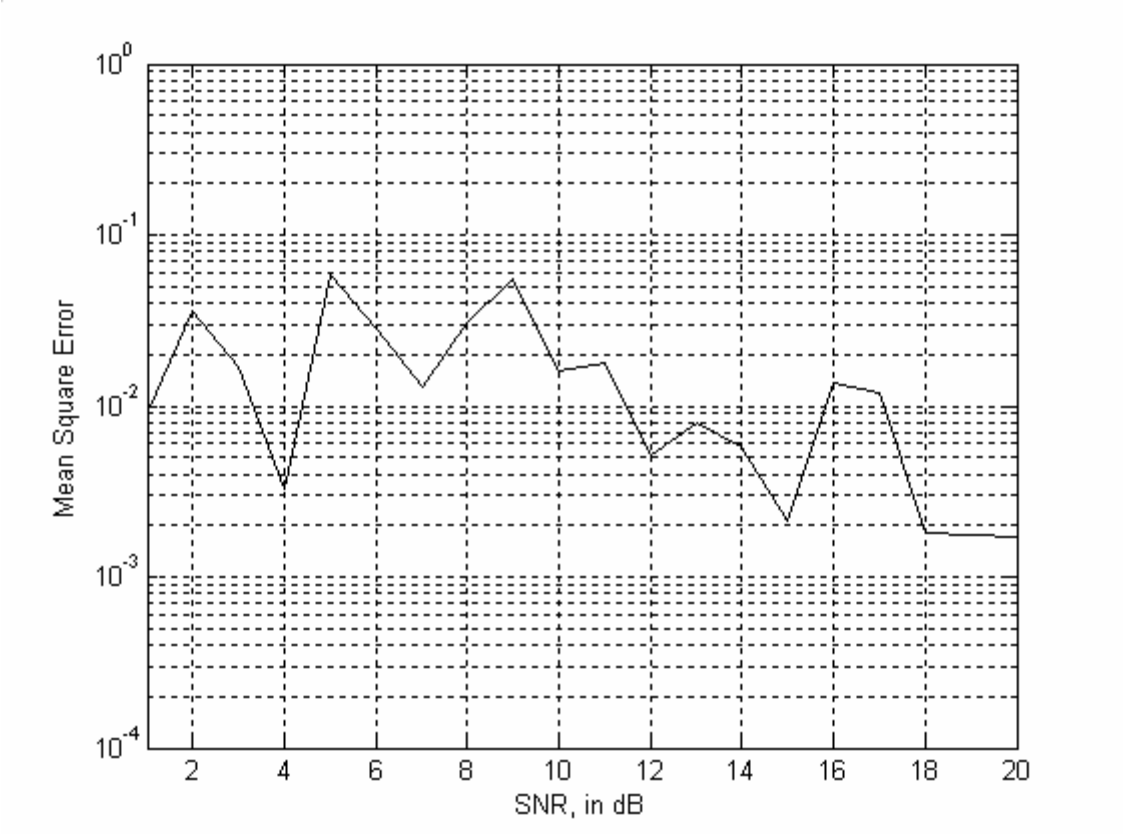

Figure 4.9 SNR estimation mean square error

The figure above show the mean square error of our SNR estimate (with known variance) on the subcarriers dedicated to data, in an AWGN channel. The algorithm guarantees that the estimate is at most one tenth of a dB away from the actual SNR.

Another issue one must keep in mind is whether the chosen scheme for channel estimation is consistently accurate. Our adaptive modulation algorithm (described in a later section) will greatly suffer if we are not able to successfully and consistently estimate the signal to noise ratio. The worse case scenario would be to overshoot the SNR every time we have to come up with an estimate. This would result in switching to a higher modulation scheme, from QPSK to 16QAM for example, at a moment in time when channel conditions are deleterious to high rate communication. On the other hand, under estimating the SNR will also have nefarious effects on system performance. The

system would continue transmitting in a lower modulation scheme, i.e. BPSK, even though the channel state would allow a switch to a higher rate, i.e. 16QAM, without suffering in bit error rate performance (not exceeding the target BER). However, the latter overall loss in system performance is preferable to sending irrecoverable information over an auspicious medium. Clearly, the quality of the channel estimation technique plays a key role in successfully and accurately determining the signal to noise ratio.

4.5 Adaptive modulation

The purpose of adaptive modulation is to maximize the resources of the wireless radio channel. As described before, traditional communication systems are built with an adequate link margin, which guarantees system functionality even in worse case scenarios. It is safe to say that those kind of extreme conditions are not the standard for slow time-varying channels. At any given time, the system could be enjoying particularly good channel conditions and would not be able to take full advantage of them if it weren't for adaptive schemes. There exists a significant variety of adaptive strategies [54-66]. Some are geared to minimize power consumption by adding an additional bit with the smallest additional transmit power possible for a requested BER [66]. Others focus on rate distribution given the capacity of the individual subchannels [60]. Some use proven adaptive strategies [64] but rely on metrics different from the SNR to perform rate switching. The goal of the adaptive modulation algorithm we used in our simulation is to reach and maintain a target BER irrespective of the SNR levels that each individual subcarrier experiences.

In OFDM, the frequency selective radio channel is perceived at the receiver as N parallel, frequency non-selective channels (flat) with different SNRs [65]. For that reason, we use the AWGN performance curves to derive the switching thresholds necessary to maintain a BER of 10^{-2} (we use a high BER target to minimize simulation runtime). Indeed, to maximize the channel capacity, it does not make sense to force transmit information through a channel which is experiencing a deep fade. In such a case, the target BER can not be guaranteed. On the other hand, we want to be able to increase the modulation scheme on a specific carrier or group of carriers if doing so does not cause the system to perform poorly. We plot the AWGN performance curves of BPSK, QPSK, 16QAM, and 64QAM.

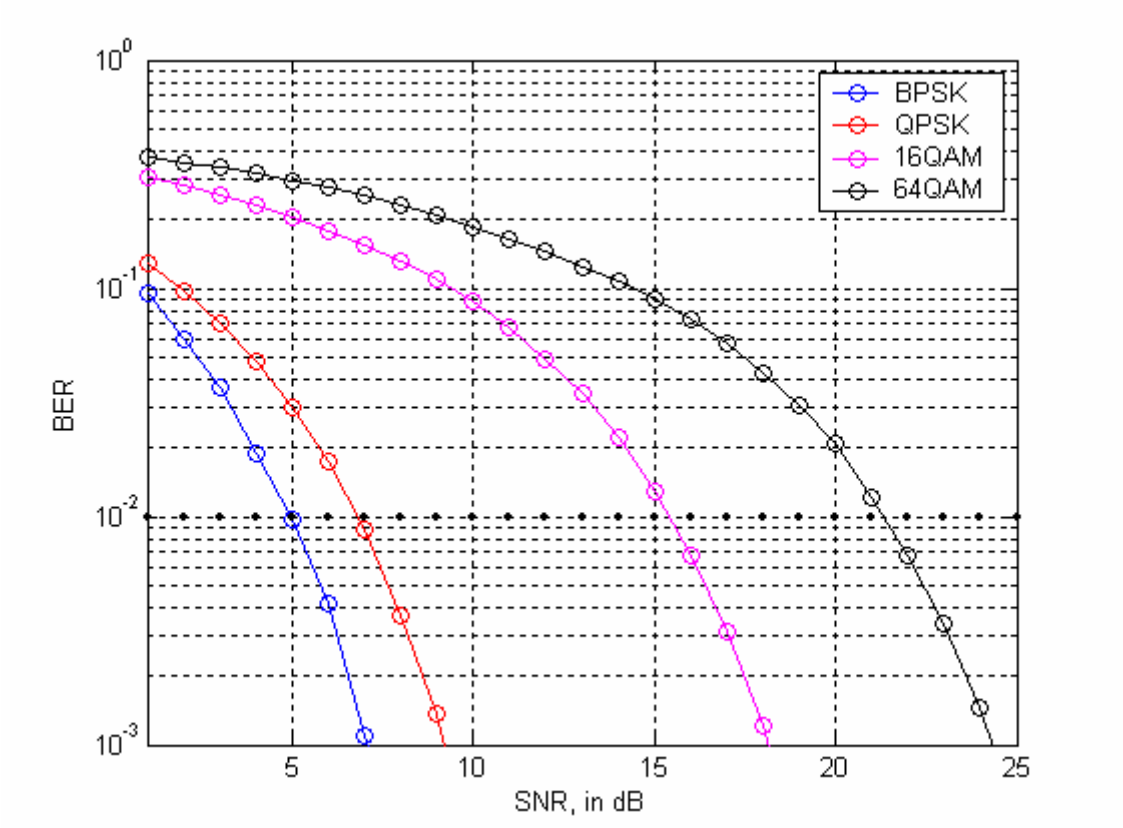

Figure 4.10 AWGN BER curves

Modulation Scheme	SNR Threshold
No TX	-
BPSK	5
QPSK	7
16QAM	16
64QAM	22

Table 4.1 AWGN Switching Thresholds

When the SNR of a specific subcarrier exceeds one of the switching thresholds, that carrier's modulation scheme is updated according to Table 4.1. No signal transmission is performed at SNRs lower than 5 dB as that would put us above the target BER we want to achieve. The size of each packet of information is maintained constant.

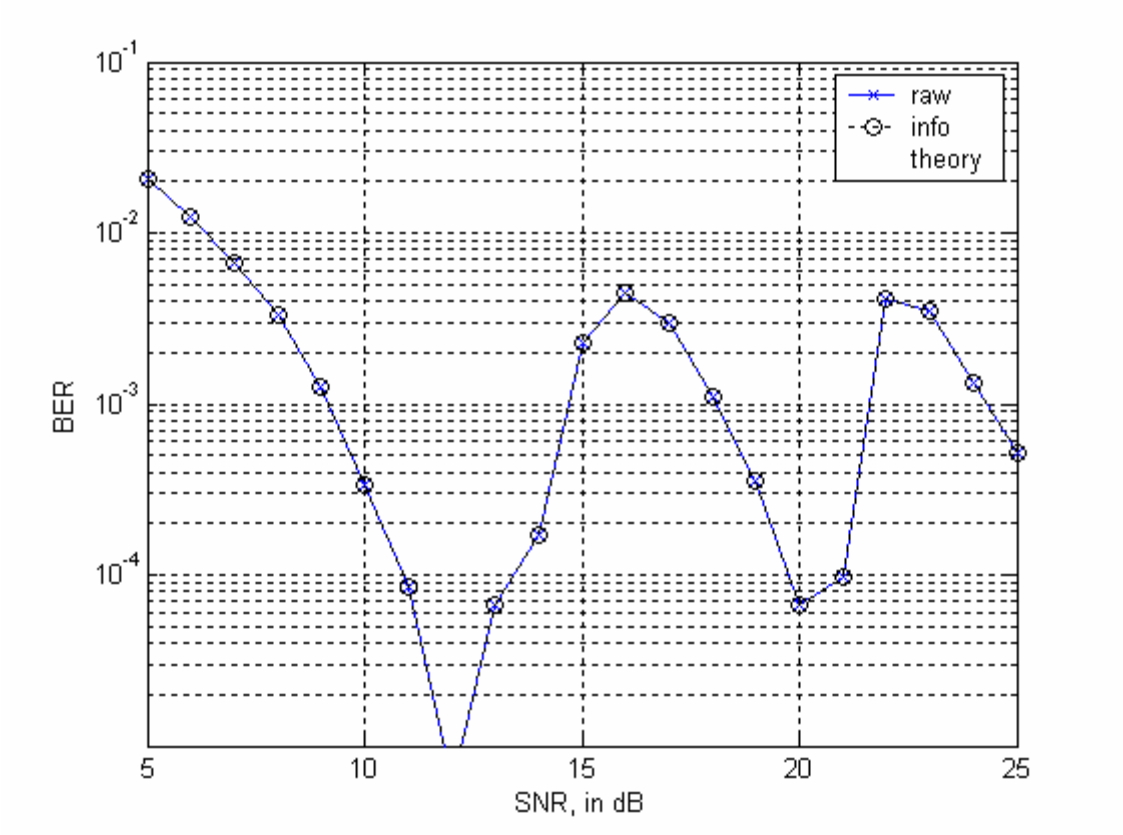

Figure 4.11 Adaptive OFDM in AWGN

The last two peaks correspond to the algorithm switching to 16QAM and then 64QAM. One can notice that there is not peak confirming the switch from BPSK to QPSK. It would seem that both modulation schemes thresholds are too close to each other for the system to be able to make a distinction.

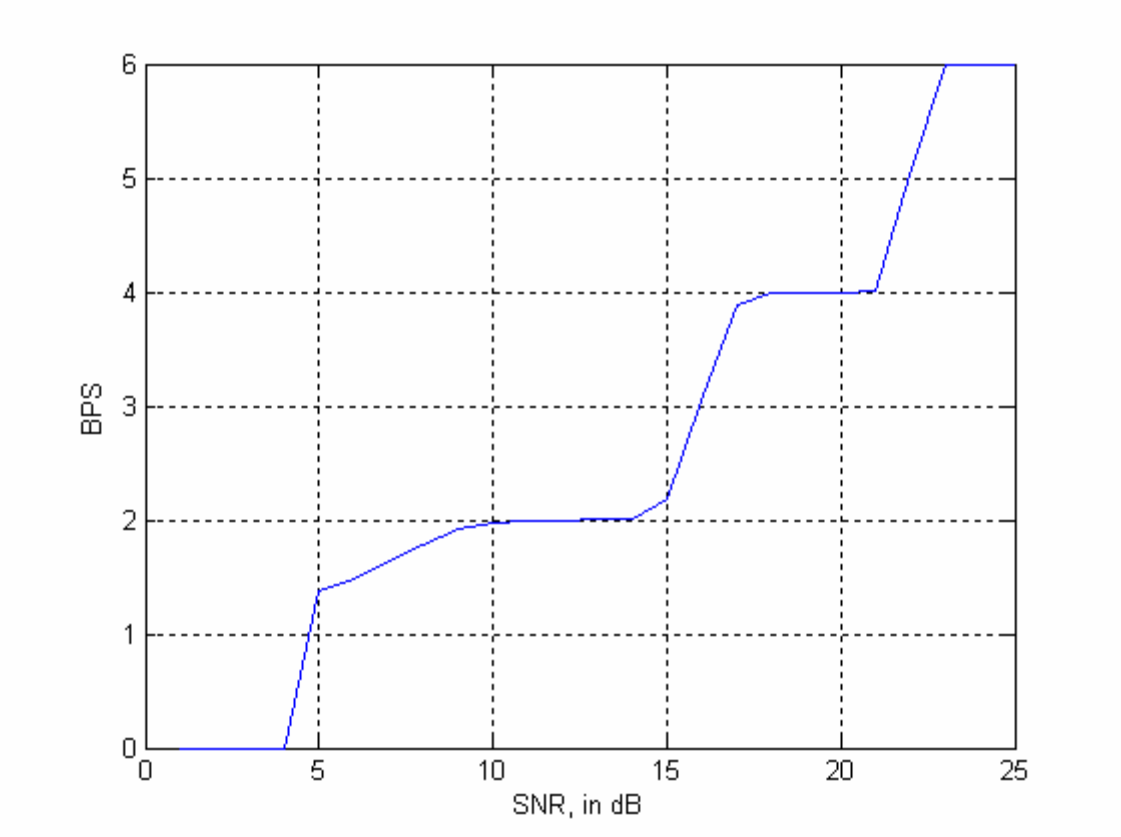

Figure 4.12 Bits per symbol as a function of SNR

Figure 4.12 show the increase in bits per symbol (BPS) as a function of the signal to noise ratio. Notice the close correlation between figure 4.12 and Table 4.1. The hikes in BPS correspond to the switching thresholds we determined earlier. This experiment allowed us to validate our algorithm. Let us now perform the same simulation over a slow Rayleigh fading channel. The results are displayed in Figure 4.13 below. The adaptive OFDM performance curve is much more continuous as opposed to the jagged one generated in the AWGN environment. In a real adaptive system, engineers must constantly worry about the speed at which the channel changes. Indeed if the channel changes more rapidly than the channel state information can be updated, the adaptive part of the system has no use. In simulation, we model that problem by manipulating the sampling period T_s and the maximum Doppler frequency f_d. For the particular system we

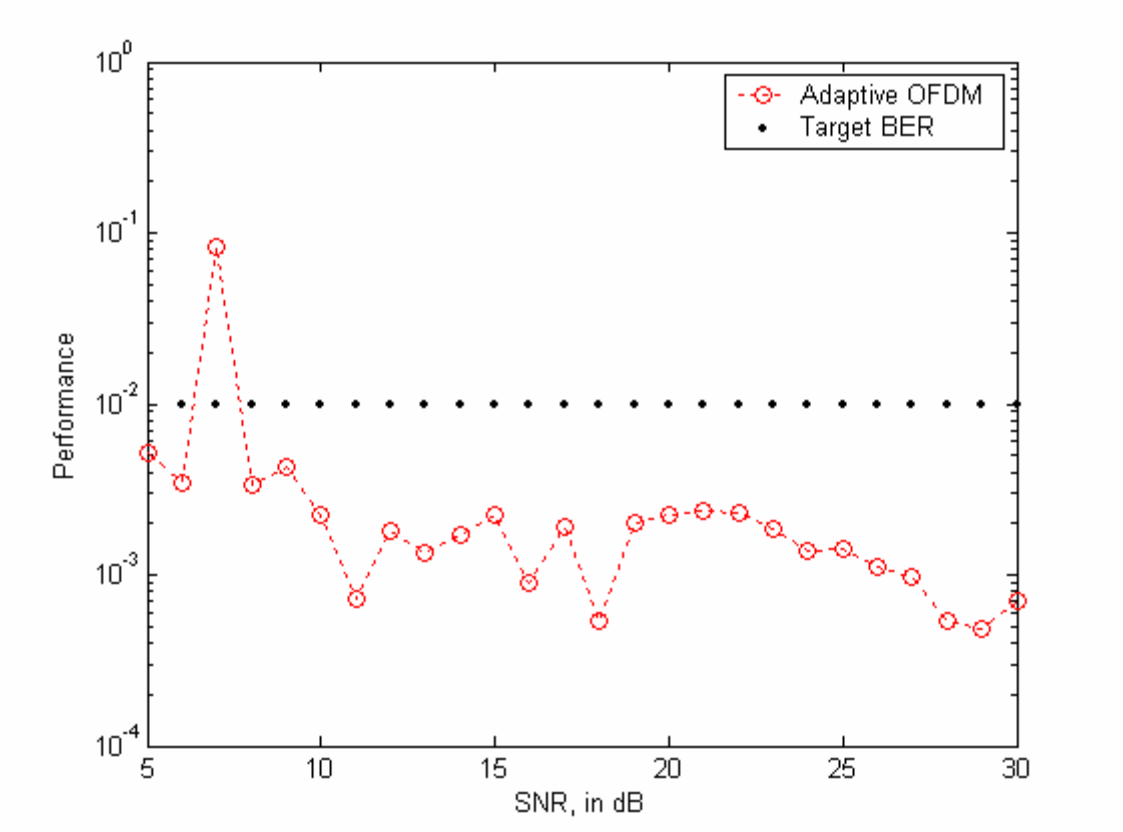

Figure 4.13 Adaptive OFDM in slow Rayleigh fading channel

simulated, we chose not to include a channel predictor. This limited the maximum Doppler frequencies and sampling frequencies we could use while maintaining the adaptive system functional. After experimenting with different values, we settled on a normalized fade rate of $f_d T_s = 0.0001$. Greater normalized fade rates result in performance curves well over the target (10^{-2}), whereas smaller rates cause the channel to vary extremely slowly. In the latter case, performance approaches that of an AWGN channel, thus defeating the purpose of the entire experiment. Figure 4.14 shows the bit per symbol (BPS) curve of the simulation.

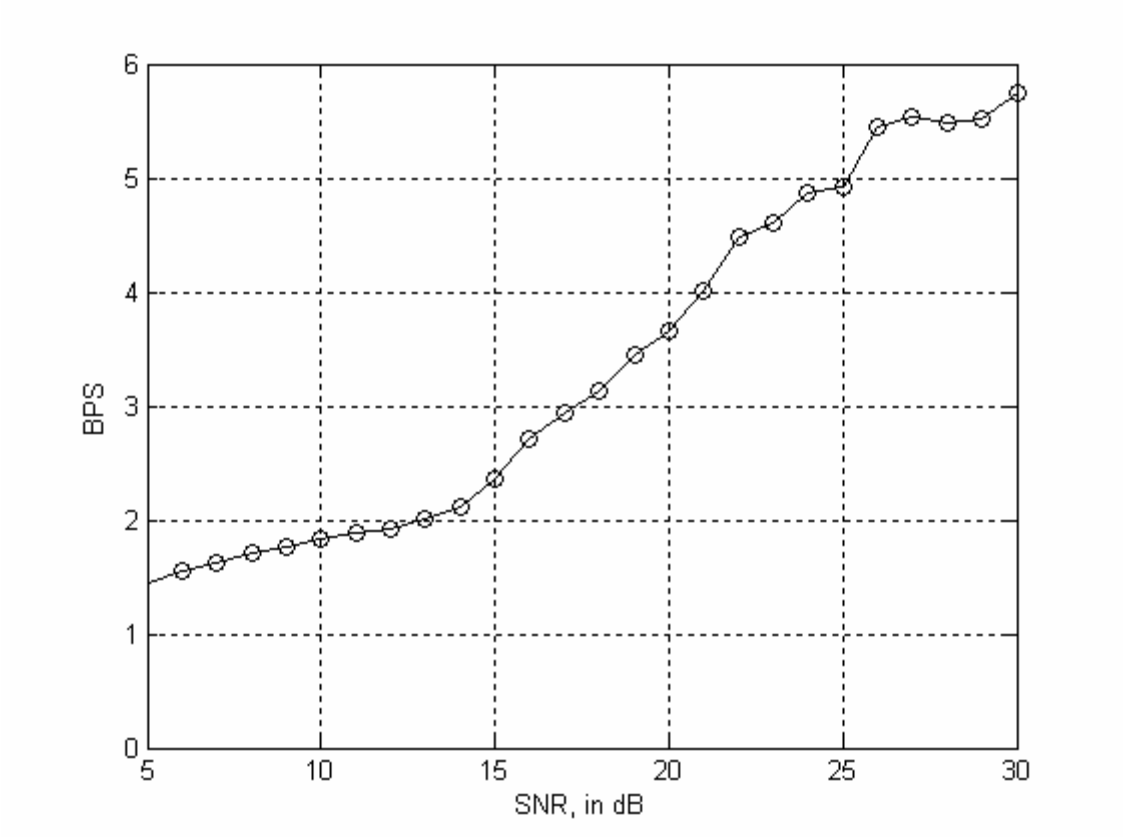

Figure 4.14 Bits per symbol as a function of SNR

Now let's repeat the last few steps using turbo codes to encrypt the information to be sent. Figure 4.12 and Table 4.1 show how close the performance thresholds for uncoded BPSK and QPSK really are. In a turbo coded scheme, we will not have the need to keep both modulation schemes. We therefore restrict our subsequent analysis to QPSK, 16QAM, and 64QAM. We also maintain the normalized fade rate to $f_d T_s = 0.0001$. The turbo code used is the 1/3 rate code described in Chapter 3.

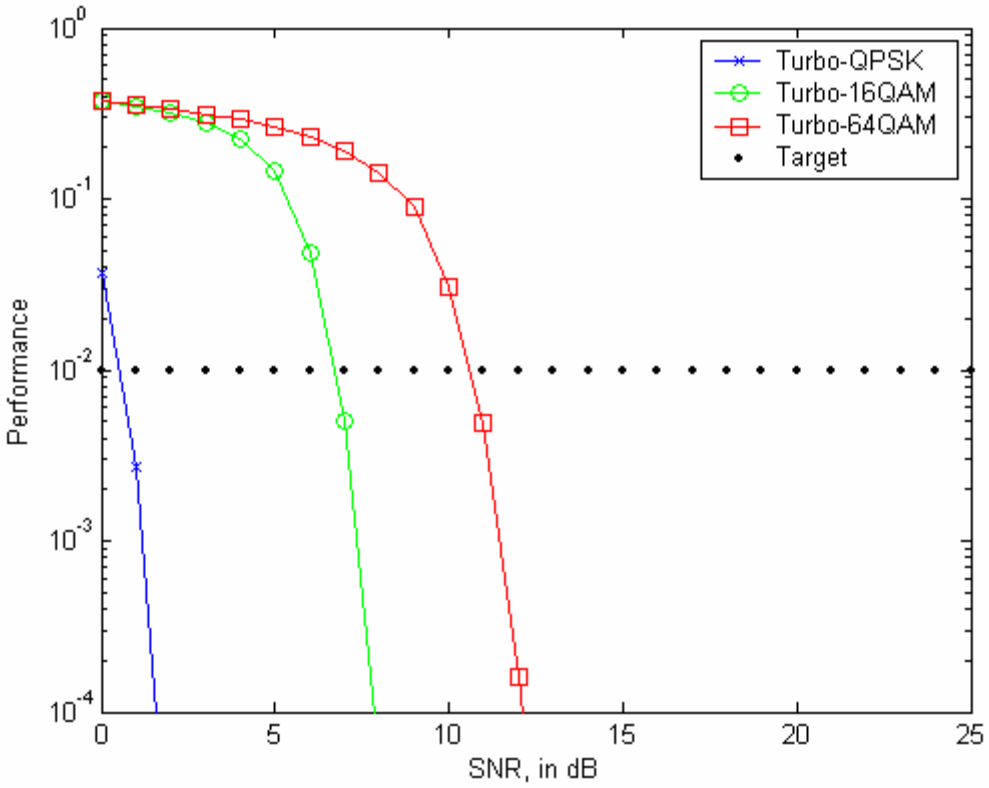

Figure 4.15 Turbo OFDM AWGN performance curves (3 iterations)

Modulation Scheme	SNR Threshold
No TX	-
QPSK	1
16QAM	7
64QAM	11

Table 4.2 Turbo-AWGN Switching Thresholds

As we had predicted, the use of turbo codes has allowed us to significantly reduce the switching thresholds.

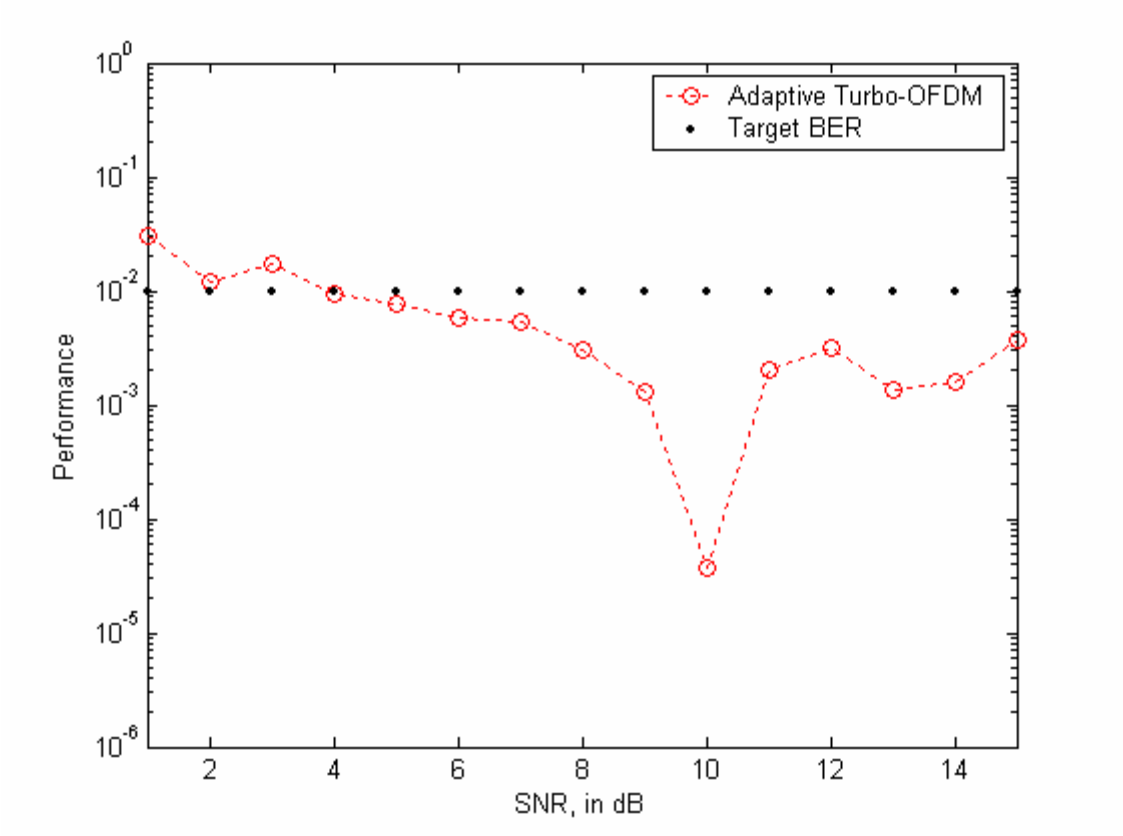

Figure 4.16 Adaptive Turbo OFDM in slow Rayleigh fading channel (3 iterations)

Figure 4.16 shows the performance of our adaptive turbo-coded OFDM scheme. The curve hovers above the targeted 10^{-2} between 1 and 4 dB. We were unable to determine the reason behind this behavior. However we are able to draw some conclusions from the simulation. The use of forward error correction has allowed us to save considerable transmit power. We reach the 64QAM threshold 11dB faster than the non coded adaptive scheme. Also, Figure 4.18 below shows that we were able to increase the system throughput. Indeed, the use of turbo codes has allowed us to increase the bits per symbol (BPS) sent through the frequency selective Rayleigh, slow fading channel. We exceed the 5 BPS at 12 dB, more than 13 dB faster than for the non-coded adaptive scheme.

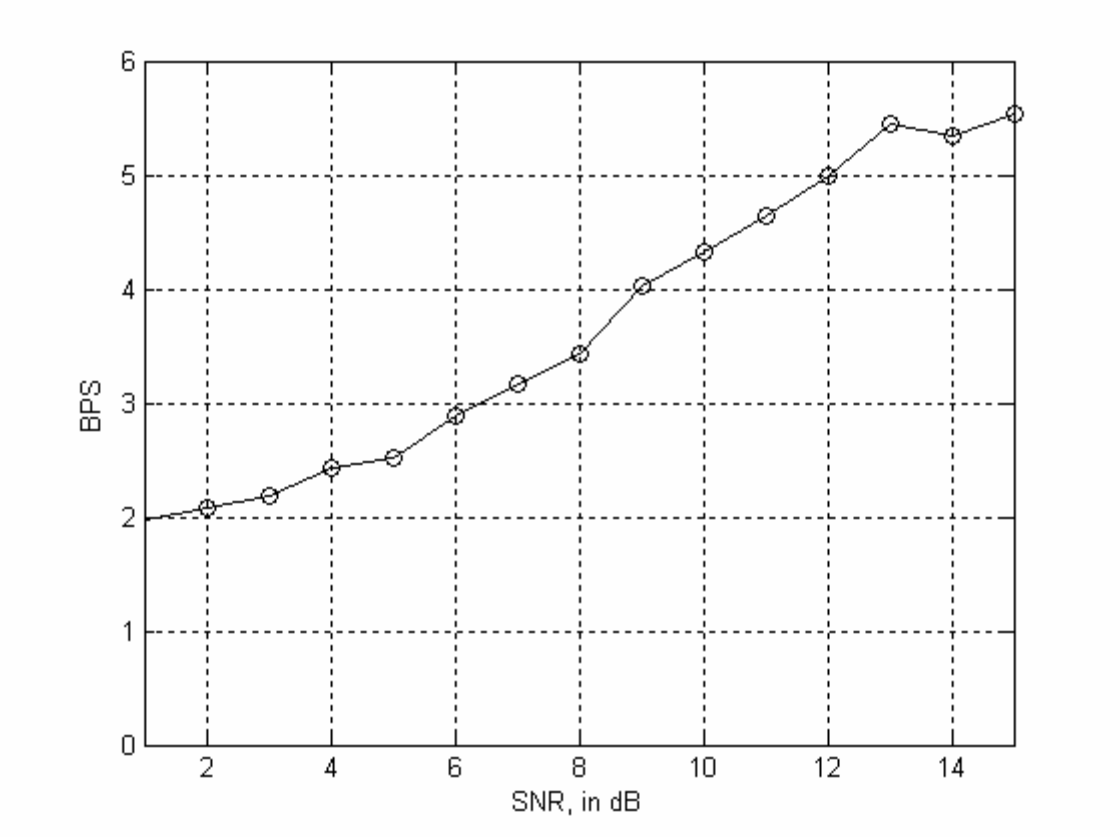

Figure 4.17 BPS as a function of SNR

By combining adaptive modulation with turbo coding, we exceeded performance expectations while enhancing the system's throughput. Figure 4.17 clearly shows how the inflection in the BPS curve occurs much sooner, at a lower SNR. This means that higher modulation schemes can be enjoyed at lower SNRs through forward error correction techniques such as turbo coding. This enhanced throughput can be achieved without breaching the target BER.

Chapter 5. Conclusion

5.1 Summary

This thesis presented the implementation and results for the adaptive, turbo-coded OFDM system as introduced in Chapter 1.

In chapter 2, we introduced the theory behind OFDM and discussed basic OFDM transceiver architecture. We identified some factors that could result in the OFDM system not performing to its potential. These factors included ISI caused by a dispersive channel, ICI and its deleterious effects, and the issue of PAPR which is crucial for proper functionality. We explored techniques to combat some of these problems such as the use of a cyclic prefix (longer than the channel delay spread), and equalization made easy thanks to the wideband nature of the OFDM. As long as the subcarrier spacing is kept smaller than the coherence bandwidth, we can take advantage of the high correlation between adjacent subcarriers. We also presented a few results in both AWGN and Rayleigh environments, as we needed to validate our modified, simplified simulator.

In chapter 3, we focused our attention on turbo codes and their implementation. We described the encoder architecture. In our case, the code is the result of the parallel concatenation of two identical RSCs. The code can be punctured in order to fulfill bit rate requirements. The decoder succeeded in its duty thanks to the decoding algorithms that it is built around. We focused mainly on the study of the MAP. We discovered that the power of the scheme came from the two individual decoders performing the MAP on interleaved versions of the input. Each decoder used information produced by the other as a priori information and outputted a posteriori information. We elaborated on the

performance theory of the codes and find out the key to explaining the two distinct performance regions was by examining the distance spectra of the code.

Chapter 4 tied concepts from chapter 2 and 3 with a target-based, adaptive modulation scheme. First we introduced and simulated the wireless slow time-varying Rayleigh fading channel. We showed how its' time-varying nature (due to motion between the transmitter and receiver) could be exploited to refine the system performance and/or throughput. Once we were able to estimate the channel, we used a fairly simple target-BER adaptive modulation algorithm to achieve our goal. Then, we presented our results on the combination of turbo codes and adaptive OFDM. The lack of powerful machines has not allowed us to generate more bits and therefore better graphs. For this reason, our results should be considered preliminary.

5.2 Future work

This thesis showed that the combination of turbo codes and adaptive OFDM can be powerful. However, a complete coded, adaptive system would include a few more wrinkles. First the system we implemented can be enhanced by improving the MAP implementation from max-log-map to log-map. Such changes would only require minimal changes to the MAP decoder modules. We believe that greater control over the BER fluctuations in the adaptive mode can be achieved by adding a 3 bit modulation scheme between QPSK and 16QAM. Even more control can be achieved by adding a module to vary the turbo code's rate and puncturing patterns such that multiple data rates can be achieved using the same modulation scheme (i.e. 16QAM). Not shown in our work is the lack of utility of turbo-codes when the target BER is lower than the "error

floor" of the code. In the future, it would be highly beneficial to implement a convolutional or trellis encoder that could be used when the turbo code use is no longer the better alternative. Spectrally, we could use the energy saved from carriers in the no-transmission zone to boost performance of carriers near a switching threshold for instance. Also, improving the channel estimation technique by integrating it with the turbo decoding process could yield some greater gains. Finally, to support greater user speeds, one could implement a channel predictor.

References

[1] P. Hoeher, S. Kaiser, and P. Robertson, "Two-dimensional Pilot-symbol-aided Channel Estimation by Wiener Filtering," IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 1845-1848, 1997

[2] M. Speth, S. Fechtel, G. Fock, H. Meyr, "Optimum Receiver Design for OFDM-Based Broadband Transmission-Part II: A Case Study," IEEE Transactions. On Communications, vol. 49, no. 4, pp. 571-578, April 2001

[3] X. Li and L.J. Cimini, "Effects of clipping and filtering on the performance of OFDM," IDDD Communications Letters, vol. 2, no. 5, May 1998

[4] D.Declercq and G.B. Giannakis, "Recovering clipped OFDM symbols with Bayesian interference," IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000

[5] D. Kim and G.J. Stuber, "Clipping noise mitigation for OFDM by decision-aided reconstruction," IEEE Communications Letters, vol. 2, no. 5, May 1998

[6] S.H. Muller and J.B. Huber, "OFDM with reduced peak-to-average power ratio by optimum combination of partial transmit sequences," Electronic Letters, vol. 33, no. 5, 1997

[7] R.W. Baumi, R.F.H. Fischer, and F.B. Huber, "Reducing the peak-to-average power ratio of multicarrier modulation by selective mapping," Electronic Letters, vol. 20, no. 22, 1996

[8] H. Ahn, y. Shin, and S. Im, "A block coding scheme for peak-to-average power ratio reduction in an orthogonal frequency division multiplexing system," VTC 2000, vol. 2, pp. 56-60

[9] J. Terry, and J. Deiskala, *OFDM Wireless LANs: A Theoretical and Practical Guide*, Sams Publishing, Indiana, 2002

[10] A.J. Goldsmith and S. Chua, "Adaptive coded modulation for fading channels," IEEE Trans. Commun., vol 46, pp. 595-602, May 1998

[11] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," Proceedings, 1993 IEEE International Conference on Communication, Geneva, Switzerland, pp.1064-1070.

[12] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," Proceedings, IEEE GLOBECOM Conf., 1994, pp. 1298-1303.

[13] J.W. Blakert, E.K. Hall, S.G. Wilson, “Turbo code termination and interleaver conditions,” Electronics Letters, Vol. 31, Issue 24, 1995, pp. 2082-2084.

[14] G.D. Forney, “The Viterbi Algorithm,” Proceedings, IEEE, vol. 61, pp 268-278, March, 1973

[15] H.-L. Lou, “Implementing the Viterbi algorithm,”, Signal Processing Magazine, IEEE, vol. 12, Issue 5, pp. 42-52, Sept. 1995

[16] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” IEEE Trans. Inform. Theory, pp. 284-287, March 1974.

[17] E.K. Hall, S.G. Wilson, “Design and analysis of turbo codes on rayleigh fading channels,” IEEE Journal, vol. 16, no. 2, February 1998

[18] I.L. Turner, “A modified Bahl algorithm for recursive systematic convolutional cods on rayleigh fading channels,” Vehicular Technology Conference, vol. 1, pp.75-76, May 1999

[19] Haixa Zhang, Feng Zhao, Dongfeng Yuan, Mingyan Jiang, “Performance of turbo code an WOFDM system on rayleigh fading channels,” Proceedings, IEEE, vol. 2, pp.1570-1573, Sept 2003

[20] S. Benedetto, G. Montorsi, “Design of parallel concatenated convolutional codes,” IEEE Transactions, vol. 44, no. 5, pp. 591-600

[21] P. Robertson, E. Villebrun, P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” International Conference, IEEE, vol. 2, pp. 1009-1013, June 1995

[22] R. Achiva, Booz, M. Mortazavi, W. Fizell, “Turbo code performance and design trade-offs,” MILCOM 2000, vol. 1, pp. 174-180, October 2000

[23] M.C. Valenti, B.D. Woerner, “Performance of turbo codes in interleaved flat fading channels with estimated channel state information”, Vehicular Technology Conference, vol. 1, pp. 66-70, May 1998

[24] A. Shaheem, M. Caldera, H.-R. Zepernick, “Channel reliability metrics for flat rayleigh fading channels without channel state information,” Sympo TIC 2004, pp. 58-61, October 2004

[25] P. Frenger, “Turbo Decoding for Wireless Systems with Imperfect Channel Estimates,” IEEE Transactions, vol. 48, no. 9, pp.1437-1440, September 2000

[26] Hyundong Shin, Sunghwan Kim, Jae Hong Lee, “Turbo decoding in a rayleigh fading channel with estimated channel state information,” Vehicular Technology Conference, IEEE, vol. 3, pp. 1358-1363

[27] Yufei Wu, B.D. Woerner, William J. Ebel, “A simple stopping criterion for turbo decoding,” IEEE Letters, vol. 4, no. 8, pp.258-260, August 2000

[28] M.C. Valenti, J. Sun, “The UMTS turbo code and an efficient decoder implementation suitable for software-defined radios”, International Journal of Wireless Information Networks, vol. 8, no. 4, October 2001

[29] J. Hagenauer, E. Offer, L. Papke, “Iterative decoding of binary black and convolutional codes,” IEEE Transactions on Information Theory, vol. 42, Issue 2, pp 429-445, March 1996.

[30] F. Mo, S.C. Kwatra, Junghwan Kim, “Analysis of puncturing patterns for high rate turbo codes,” MILCOM 1999, IEEE, vol. 1, pp. 547-550, November 1999

[31] A. Viterbi, “Effor bounds for convolutional codes and an asymptotically optimum decoding algorithm,” IEEE Transactions on Information Theory, vol. 13, Issue 2, pp.260-269

[32] J. Hagenauer, P. Hoeher, “ A viterbi algorithm with soft-decision outputs and its applications,” GLOBECOM 1989, IEEE, vol 3, pp. 1680-1686, November 1989

[33] P. Robertson, E. Villebrun, P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” ICC ’95, IEEE, vol. 2, pp. 1009-1013

[34] J.A. Erfanian, S.Pasupathy, G. Gulak, “Reduced complexity symbol detectors with parallel structures for ISI channels,” IEEE Transactions on Communications, vol. 42, pp.1661-1671, April 1994

[35] A Gueguen, D. Castelain, “Performance of frame oriented turbo codes on UMTS channel with various termination schemes”, Vehicular Technology Conference 1999, IEEE, vol. 3, pp. 1550-1554, September 1999

[36] C.B. Schlegel, L.C. Perez, “Trellis and Turbo Coding,” IEEE Press, Hoboken, NJ: John Wiley & Sons , INC

[37] R. Garello, P. Pierleoni, S. Benedetto, “Computing the free distance of turbo codes and serially concatenated codes with interleavers: algorithms and applications,” IEEE journal on selected areas in communications, vol. 19, no. 5, May 2001

[38] J. Seghers, “On the free distance of TURBO codes and related product codes,” Final Rep., Diploma Project SS 1995, no. 6613, Swiss Federal Institute of Technology, Zurich, Switzerland, August 1995

[39] D.J. Young and N.C. Beaulieu, “The generation of correlated Rayleigh random variates by inverse discrete Fourier transform,” IEEE Transactions on Communications, vol. 48, no. 7, July 2000

[40] J. I. Smith, “A computer generated multipath fading simulation for mobile radio,” IEEE Trans. Veh. Technol., vol. VT-24, pp. 39–40, August 1975.

[41] W. Jakes, *Microwave mobile communications*, New York, Wiley, 1974

[42] Patzold M., Szczepanski A., Youssef N., “Methods for Modeling of specified and measured Multipath power-delay profiles,” IEEE Transactions on vehicular technology, vol. 51, no. 5, September 2002

[43] Wang Cheng-Xiang, Patzold M., “Methods of generating multiple uncorrelated rayleigh fading processes,” Vehiculatr Technology Conference, vol. 1, pp. 510-514, April 2003

[44] Hoehrer P., “A statistical discrete-time model for the WSSUS multipath channel,” IEEE Transactions on vehicular technology, vol.41, no. 4, November 1992

[45] Lei-Lei Lock, Xiangming Kong, Barton R.J., “Simulation of time-varying, frequency-selective multipath fading channels for spread spectrum waveforms,” 33rd Asilomar Conference on signals, systems, and computers, vol. 2, pp. 1675-1679, October 1999

[46] Prabhu G.S., Shankar P.M., “Simulation of flat fading using Matlab for classroom instruction,” IEEE Transactions on Education, vol. 45, Issue 1, pp. 19-25, February 2002

[47] J. Heiskala, J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*, Sams Publishing, Indianapolis, Indiana, 2002

[48] Ming-Xian Chang, Su Y.T, “Blind and semiblind detections of OFDM signals in fading channels,” IEEE Transactions on Communications, vol. 52, Issue 5, pp. 744-754, May 2004

[49] Ming-Xian Chang, Su Y.T, “Blind joint channel and data estimation for OFDM signals in Rayleigh fading,” Vehicular Technology Conference, vol. 2, pp. 791-795, May 2001

[50] S. Boumard, “Novel noise variance and SNR estimation algorithm for wireless MIMO OFDM systems,” GLOBECOM ’03, IEEE vol. 3, pp. 1330-1334, December 2003

[51] W. Lee, J. Zhu, “Channel estimation for high-speed packet-based OFDM communication systems,” Proc. WPMC ’02, pp. 1293-1298, October 2002

[52] Z. Cheng, D. Dahlhaus, “Time versus frequency domain channel estimation for OFDM systems with antenna array,” Proc. International Conference on Signal Processing ’02, pp. 1340-1343, August 2002

[53] J.J. van de Beek, O. Edfors, M. Sandell, S.K. Wilson, P.O. Borjesson, “On channel estimation in OFDM systems,” Proc. Vehicular Technology Conference’95, pp.815-819, July 1995

[54] L. van der Perre, S. Thoen, P. Vandenameele, B. Gyselinckx, M. Engels, “Adaptive loading strategy for a high speed OFDM-base WLAN,” Global Telecommunications Conference 1998, vol. 4, pp. 1936-1940, November 1998

[55] Huiyun Kang, Byungik Son, Yunho Jung, Junghyuck Lee, Jaeseok Kim, “An efficient bit loading algorithm for OFDM-based wireless LAN systems and hardware implementation results,” ASIC 2003, vol. 2, pp. 1054-1057, October 2003

[56] Xiaoming She, Shidong Zhou, Xibin Xu, Yan Yao, “Constant Throughput Adaptive OFDM employing rate-compatible turbo coded modulation,” IEEE proceedings, PIMRC 2003, vol. 1, pp. 355-359, September 2003

[57] Xiaoming She, Shidong Zhou, Xibin Xu, Yan Yao, “Adaptive Turbo Coded Modulation for OFDM transmissions,” IEEE Proceedings, ICCT 2003, vol. 2, pp. 1491-1495, April 2003

[58] J. Andrea Goldsmith, Soon-Ghee Chua, “Variable-rate variable-power MQAM for fading channels,” IEEE Transactions on communications, vol. 45, no. 10, October 1997

[59] Xiaoming She, Shidong Zhou, Xibin Xu, Yan Yao, “Power and bit allocation for adaptive turbo coded modulation in OFDM systems,” Global Telecomunications Conference 2003, vol. 2, pp. 903-907, December 2003

[60] P.S. Chow, J.M. Cioffi, J.A.C. Bingham, “A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels,” IEEE Transactions on Communications, vol. 43, no. 2/3/4, pp. 773-775, February/March/April 1995

[61] Andreas Czylwik, “Adaptive OFDM for wideband radio channels,” GLOBECOM ’96, vol. 1, pp. 713-718, November 1996

[62] Clive Tang, Victor Stolpman, “An adaptive learning approach to adaptive OFDM,” Wireless Communications an Networking Conference 2004, vol. 3, pp. 1406-1410, March 2004

[63] R.F.H. Fischer, J.B. Huber, "A new loading algorithm for discrete multitone transmission," GLOBECOM '96, vol. 1, pp.724-728

[64] R.Grunheid, E. Bolinth, Pr. Dr. H. Rohling, "A blockwise loading algorithm for the adaptive modulation technique in OFDM systems," Vehicular Technology Conference 2001, vol. 2, pp.948-951, October 2001

[65] Hughes Harthogs. "Ensemble modem structure for imperfect transmission media," U.S. Patentes Nos. 4,679,227 (July 1987), 4,731,816 (March 1988), and 4,833,706 (May 1989)

[66] A.G. Burr, "Wide-band channel modeling using a spatial model," 5th International symposium on spread spectrum techniques and applications, IEEE proceedings, vol. 1, pp. 255-257, September 1998

[67] A.A.M. Saleh, R.A. Valenzuela, "A statistical model for indoor multipath propagation," IEEE Journals, Selected Areas in Communications, vol. SAC-5, no. 2, pp. 128-137, February 1987

[68] T. S. Rappaport, *Wireless communications, principles and practice*, Prentice Hall, New Jersey, 1996