School of Electrical and Electronic Engineering, University of Newcastle

MSc Wireless Embedded Systems 2013-2014, Semester 1

**EEE8077**: Simulation of Wireless Communications

Formal Project Report

Students:

* Nitisha Garole (105861025)
* Abhijeet Patil (130580339)
* Guanqiao Ren (130621045)
* Michael Walker (130623935)

## Absract

*The Bit Error Rate (BER) performance of a digital communication link utilizing OFDM with 16-QAM modulation in the presence of AWGN and through a Multipath Channel was investigated using MATLAB simulation.*

# Introduction

Wireless communication systems, when considered against wired systems, are of increasing importance in the modern world; just one use of wireless; that of mobile telephony has radically altered the way the world works in the space of only two decades. However, wireless communication suffers from disadvantages compared to that of wired communication in terms of limits to its data-rate and to its range.

Techniques used to mitigate the unique problems of the wireless communications regime have been developed over years and decades and have significantly improved the technology. Our objective in this course has been to utilize some of the most modern of these techniques in a MATLAB simulation to explore and compare their effects and gain an understanding and insight into digital communication systems in general.

This system to be simulated can be divided into the three parts, namely the **Transmitter**, the **Channel** and the **Receiver**, this is because ‘Any communication system moves information from a source to a destination through a channel [1]. The role of transmitter is to take binary data input and simulate the modulation of that data it onto 16-QAM OFDM subcarriers. The channel combines the original signal with fixed multipath interference and a variable intensity of additive white Gaussian noise. The receiver performs the demodulation of the signal back into binary data.

Certain techniques are explored to increase the performance of the system. These include the uses of and cyclic prefixes and a one-tap equalizers in the receiver to mitigate the effects of the multipath channel and limit the errors in the resulting data.

The ratio of the number of errors in the resultant binary data to the amount of data sent is always calculated for any particular setup of the simulated system. This is the Bit Error Rate or **BER**. Depending on the strength of the signal or the effectiveness of any performance enhancing techniques the BER will alter. Investigating how it is altered and why is the topic of this report.

# Background Theory

## Orthogonal Frequency Division Multiplexing (OFDM):

Orthogonal Frequency Division Multiplexing (OFDM) is a parallel transmission scheme. Whereby a high-rate serial data stream is split up into a set of lower-rate substreams, each of which is modulated on a separate Subcarrier [2].

Thereby, the bandwidth of the Subcarriers becomes small compared with the coherence bandwidth of the overall channel. This gives efficient usage of the spectrum of that channel, and the individual Subcarriers experience flat fading, which allows for simple (one-tap) equalization in the receiver [3]. This also implies that the symbol period of the substreams is longer compared to the delay spread of a time-dispersive radio channel, and thus it is less sensitive to symbol timing and multipath impulse noise.

However, due to its orthogonal subcarriers, the carrier frequency must be stable and avoid any offset (CFO) which would make the subcarriers non-orthogonal and cause intercarrier interference (which can also be caused by distortions due to the FFT). Likewise it is sensitive similar frequency deviations caused by Doppler shifting when transmitters and/or receivers are accelerating relative to each other. Also, exhibits a high crest factor (or PAPR) such that its average power output is quite low compared with single carrier schemes and this puts constraints on both its bandwidth and power efficiency (Shannon–Hartley theorem [4]). There is currently active research trying to find crest factor reduction techniques (CFR) for OFDM, as it is probably the prime disadvantage of the method and would lead to important economic effects [5].

### FFT-based OFDM

An Orthogonal Frequency Division Multiplexing (OFDM) system is a multi-carrier system which utilizes a parallel processing technique allowing the simultaneous transmission of data on many closely spaced, orthogonal sub-carriers. The Inverse Fast Fourier transform (IFFT) and Fast Fourier transform (FFT) in a conventional OFDM system are used to multiplex the signals together at the transmitter and decode the signal at the receiver respectively.

The system adds cyclic prefixes (CP) before transmitting the signal. The purpose of this is to increase the delay spread of the channel so that it becomes longer than the channel impulse response. The purpose of this is to reduce interference (ISI). However, the CP has the disadvantage of reducing the spectral containment of the channels.

## Digital Modulation Techniques

When we want to send signals over any distance, baseband signaling is not sufficient. We must therefore modulate the signals onto an RF carrier.The principle of the digital modulation is to modulate the analog carrier signal by a discrete signal. Digital modulation methods can be seen as a particular case of digital-to-analog conversion, and the corresponding demodulation or detection as a case of analog-to-digital conversion. How many possible discrete signals, or symbols are used is known as the modulation alphabet (the number of symbols given by the value ‘M’). After multiple modulations of the periodic analog carrier signal (usually a sine wave of high frequency) over discrete periods of time by the one of the discrete symbols in turn we will get a resulting complex analog modulated signal ready to transmit.

Digital information is transmitted by varying the information of either amplitude *a(t)* frequency *f(t)* or phase ɸ(t) of a sinusoidal RF signal[6]:

The most fundamental digital modulation techniques are based on keying:

**ASK (amplitude-shift keying)**: Discrete Amplitude levels are used as the symbols to achieve the aim of ASK. Pulse shaping can be employed to remove spectral spreading. ASK demonstrates poor performance as it is highly affected by noise and interference.

**FSK (frequency-shift keying):** One of the simplest and widely used analogue modulation techniques is Frequency Modulation (FM). The corresponding digital modulation scheme of FSK has the advantage of being very simple to generate and very simple to demodulate and due to constant amplitude, the probability of error is less as compared to ASK. Significant disadvantages, however are the poor spectral efficiency and BER performance.

**PSK (phase-shift keying):** An alternative to imposing the modulation onto carrier by varying the instantaneous frequency is to modulate the phase. This can be achieved simply by defining a relative phase shift from the carrier, usually equidistant from each required state. This method shows an improved BER performance. PSK and FSK can be considered special cases of each other, as both modulate the signal in the time domain.

[**QAM (quadrature amplitude modulation)**](http://en.wikipedia.org/wiki/Quadrature_amplitude_modulation)**:** A combination of both PSK and ASK, A number of at least two phases and at least two amplitudes are used, the resulting combinations of phase and amplitude each represent one symbol, and are conventionally plotted on a 2 dimensional constellation diagram.[6]

### Binary Phase Shift Keying (BPSK) Modulation

The simplest form of Phase Shift Keying (PSK) is BPSK (also called 2-Phase Reversal Keying, or 2PSK). The two phases encode 2 binary states and are separated by 180°. BPSK is the most utilized modulation scheme of all the PSKs since it is the simplest to implement and most resistant to error as takes the highest level of noise or distortion to make the demodulator reach an incorrect decision. It is, however, only able to modulate at 1 bit/symbol and so is unsuitable when high data-rates are required.

The pair of signals s1(t) and s2(t) used to represent symbols 1 and 0 defined respectively.

Where 0≤ t ≤ Tb with Tb denoting the bit duration and Eb donating the transmitted signal energy per bit.[6]

An Advantage of BPSK is that the transmitted energy per bit, Eb is constant equivalently and the average transmitted power is constant. However a disadvantage is that Demodulation of BPSK cannot be done using envelope detection, rather we have to look coherent.

### Quadrature Amplitude Modulation

**Quadrature amplitude modulation** (**QAM**) is a term used for both analog and digital modulation schemes. Two analog message signals or two digital bit streams are created by modulating the amplitudes of two carrier waves, using the amplitude-shift keying (ASK) technique. The carrier waves are sinusoidal signals of the same frequency which are out of phase with each other by 90° and are termed as quadrature components.

QAM can be viewed as an extension of multiphase PSK modulation wherein the two baseband signals are generated separately from each other. Thus, two completely independent channels are established, including the baseband coding and detection process.

After the modulation of the signals we get a waveform which is a combination of both phase-shift keying (PSK) and amplitude-shift keying (ASK). In the digital QAM a finite number of at least two phases and at least two amplitudes are used. The QAM principle is often used for designing the PSK modulator. Most of the digital telecommunication systems in use today use QAM to achieve the desired goal of modulation. By setting a suitable constellation size we can achieve high spectral efficiency with QAM. In the special case where two levels (± 1) are used on each channel, the system is identical to 4-PSK. Higher-level QAM systems, however, are distinctly different from higher-level PSK systems. The spectrum of a QAM system is determined by the spectrum of the baseband signals applied to the quadrature channels. Since these signals have the same basic structure as the baseband PSK signals. QAM spectrum shapes are identical to PSK spectrum shapes with equal numbers of signal points. Specifically, 16-QAM has a spectrum shape that is identical to 16-PSK. Even though the spectrum shapes are identical, the error performances of the two systems are quite different. With large number of signal points, QAM systems outperform PSK systems. The basic reason is that distance between signal points in PSK system is smaller than the distance between points in a comparable QAM system.

#### 16-QAM

16-state quadrature amplitude modulation involves four I values for amplitude and four Q values for phase. There are 16 states/symbols available because 24= 16. These 16 symbols can be arranged in a modulation scheme such that the constellation diagram takes the form of a 4 by 4 grid of points on the complex plane.

* Theoretical bandwidth efficiency is four bits/second/Hz.
* Data is split into two channels, I and Q and 4 values for them.
* As with QPSK, each channel can take on two phases. However, 16-QAM also accommodates two intermediate amplitude values.
* Two bits are routed to each channel simultaneously.
* The two bits to each channel are added, then applied to the respective channel’s modulator.

## Channel Noise

### AWGN Channel

The **Additive White Gaussian Noise** channel adds Gaussian noise to the signal that passes through it with a constant spectral density.To simulate background thermal noise (also known as Johnson noise) AWGN channels are used. This noise is flat spectrum and is caused by the thermal excitement of electrons in conductors[7] is an important contributing factor to the noise of the channel, and provides a lower limit on the noise in any detector[7]. But used on its own it is not a good model for terrestrial wireless communication links simply because it does not include any of the effects of wireless propagation through the atmosphere. Real world communication channels also contain other noise sources such as multipath interference.

### Signal to Noise Ratio (SNR)

For wireless (and all other) communications SNR is very important. An introduction to the concept of SNR is outside the scope of this report, but this table [8] was found useful by the group as a rough guide in analysing the performance of our simulation compared to real world mobile phone performances.

|  |  |
| --- | --- |
| 40dB SNR | Excellent signal (5 bars) |
| 25dB to 40dB SNR | Very good signal (3 - 4 bars) |
| 15dB to 25dB SNR | Low signal (2 bars) |
| 10dB - 15dB SNR | Very low signal (1 bar) |
| 5dB to 10dB SNR | No signal |

### Multipath Channel and Rayleigh Fading

In real world conditions, wireless signals do not follow a straight line from transmitter to receiver. Instead for most wireless applications, such as mobile phones or wireless computer networks the signal is generally sent from an omnidirectional dipole antenna and propagates in a toroidal shape [9]. Although most of the signal will be lost, a small proportion of the signal will take the direct route to the receiver, however the signal will also reach the receiver through other routes if it encounters a reflective surface that bounces the signal into the path of the receiver.

The signals that are reflected have a small corresponding delay due to the extra distance they have travelled causing them to arrive phase shifted from the primary signal, given they travel at the speed of light, and the distances involved are relatively short. They will also have lost greater energy than the direct, line-of-sight signal, thus these are ‘ghosts’, echoes of the main signal that interfere with its reception. When the energy levels of all the signals fit a probability distribution curve known as a Rayleigh Distribution (named after the English, Victorian physicist Lord Rayleigh) it is known as Rayleigh fading. This can be used for a statistical model of real-world multipath channel effects and is especially close fitting to the multipath effect of large built-up urban centers such as cities. For example the multipath channel propagation in the skyscraper populated island of Manhattan was found to be very similar to the theoretical model [10].

While Rayleigh Fading is a good statistical model for the distribution of multipath interference, it can only be used as a guiding basis for Multipath channel propagation to be mathematically modeled for simulation. Considering a static multipath channel for the simulation case, we can model the impulse response of all paths using the complex-baseband impulse response equation below.

The amplitude (a) of each path at any given time by the first section of the equation before the delta symbol, and involves using Euler’s formula applied to the complex plane. The Phase shifts (φ) of the reflected signals are both proportional to delay of that signal and given by (where fc is the frequency of the carrier wave). The second half (the part with the delta) defines the impulse response of the path through using the Dirac delta function.

## Bit error rate (BER):

With digital systems, it is the output quality of the information that is primary concern. Since the information is digital and usually has a binary representation, this quality is measured in terms of the average bit error rate (BER). A bit error occurs whenever the original bit prior to its transmission and received bit do not agree. This is a random event, but one that is probabilistically based on the noise of the channel. The BER is the primary measure of performance quality of digital communications system and it is typically a nonlinear function of Signal to Noise ratio.

Let n = number of bit errors observed in a sequence, and N = length of sequence.

Then, [6]

Naturally the required bit error rate of a digital system depends upon the application the system is intended to be used for:

* For general data transmission over a wireless channel, a bit error rate of 10-5 to 10-6 is often consider a reasonable objective.
* For important financial data, a BER of 10-11 may well be the requirement.
* For a coded speech signal, a BER of 10-2 to 10-3 is likely sufficient.

The BER is often expressed as a function of the normalized carrier-to-noise ratio measure denoted Eb/N0, (energy per bit to noise power spectral density ratio) in a noisy channel.

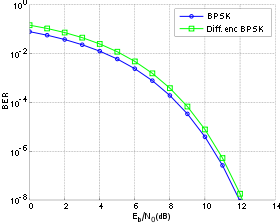


Figure : Example BER comparison between BPSK and Differentially encoded BPSK.

## Grey Codes

Grey codes are named after American Physicist Frank Gray and are an alternative binary numbering scheme for the integers such that each value is only dissimilar from it increment and decrement by one binary bit[11]. This has many practical applications, especially in modern digital communications for error detection and correction codes. By laying out a modulation schemes symbol constellation using grey coded values it can be ensured that no point is adjacent to another point that is more than 1-bit dissimilar to it. Thus, because if an error occurs and the symbol is misread it is most likely to be misread as an adjacent symbol, a one bit error will occur, which a simple error detection scheme such as a parity bit is more likely to be able to detect without collision, and a more complex error correction scheme such as a hamming code (the hamming(7,4) code for example) can actually correct the error.

### Constellation of Gray-Coded 16-QAM

For a constellation point z = x + jy , the bit allocation,b0b1b2b3, is as follows:

**First bit, b0 Second bit,b1**

b0=0 if Re {z}≤0 b1=0 if Re{z}≤−2 or Re {z}≥2

b0 = 1 if Re {z}>0 b1=1 if Re {z}>−2or Re {z}<2

**Third bit,b2 Forth bit,b3**

b2=0 if Im {z}≥0 b3 = 0 if Im{z}≤−2 or Im{z}≥2

b2= 1 if Im {z}< 0 b3=1 if Im {z} > −2 or Im {z}< 2.

# Implementation

A wireless communication system can be divided into three parts: the **transmitter** sending the signal, the **channel** the signal is sent through and the **receiver** that picks up the signal. Likewise, the simulation code is also divided into three functions each called from within a main simulation loop.

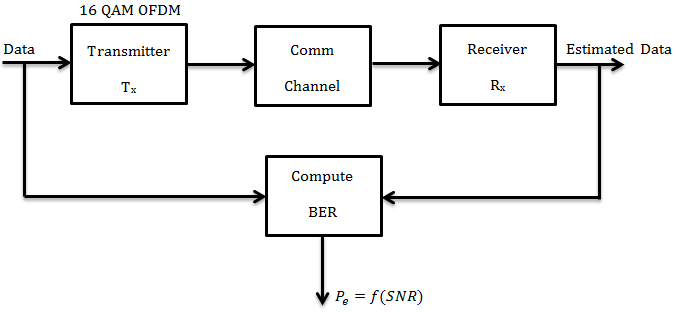


Figure : Overview of simulation structure

In testing the modular design of the system is useful in adjusting the simulation so that it in run in four different configurations. In the first type, a 16 QAM signal is produced by the transmitter to which various length cyclic prefixes are added and then to this transmitted signal the channel adds only AWGN. In this case no equalizer is required for the receiver. In the second type, the transmitter is the same but this time the signal passes through the Multipath channel with AWGN, this signal is then received by the receiver still without an equalizer. In the third setup an equalizer is added the receiver which unrealistically assumes perfect channel knowledge. Finally, in the fourth type the signal is modified in the transmitter to contain pilot values and it goes through the same channel as in second and third types but this time a different equalizer is used which uses the pilots to estimate the multipath channel effects is.

## MATLAB Programming Techniques

In implementing the simulation, our group found some common programming techniques useful, however it may cause the resulting code to appear unusual compared to other groups, so a quick explanation would seem to be useful here:

### Functional Decomposition

To help keep the program understandable and easily maintainable, the structure of the simulation was partitioned into functions rather than running it as one giant script. This had the advantage of allowing the general structure of the main simulation loop to be more apparent and compartmentalized the code into smaller more manageable groupings.

### Global Variables

Much of the data used in the simulation is required by more than one function. Unlike with a single script where all the data is global be necessity, the splitting of the code into discrete functions requires the specific declaration global data variables. MATLAB has provisions for declaring variables global and for declaring those global variables that are used by each function. These are used extensively in the code.

## Main Loop

The heart of the simulation is a doubly-nested for loop. It consists of an inner loop that runs a single simulation run, where data is generated, transmitted and received and checked for errors, and an outer loop that runs that simulation multiple times for different channel conditions.

There are 500 simulation runs used for each channel condition, so that the requirement the communication link runs for at least 106 symbols is met (i.e. 2048 \* 500 > 106). Running the simulation multiple times and averaging the result is known as Monte-Carlo Simulation[12], this is the inner loop. The channel conditions present during a particular inner loop are based on the SNR of the AWGN which is altered each outer loop to look at the results of simulation between 0-16dB in the case of AWGN only and 0-40db (in step of 2dB) in the case of multipath and AWGN.

The code for running using parallel processing is implemented, but left commented out by default as it was found that it does not induce much of a performance increase.

The bit errors generated from each simulation run need to be accumulated for each channel condition simulated the symbols received and the original symbols that were transmitted are compared, each discrepancy indicating a symbol error. The total number of errors divided by the total number of symbols is the symbol error rate. Which, as each symbol encodes 4 binary digits needs to be divided by 4 to get the BER for a particular simulation run, this is appended to a record BER which stores the BER for all runs at the different SNRs. At the end of all simulation runs the data is ready to be plotted and displayed.

The simulation also calculated the mean-squared error of the estimated frequency response when the equalizer for unknown channel is used.

‘tic’; a command added before the inner loop of simulation and ‘toc’; added after the inner loop are used so that the time taken to simulate a particular signal-to-noise ratio point can be obtained, this is useful when outputting the messages to the screen while the program runs and the times are accumulated so that the total time can be displayed against the final BER plot.

## Transmitter

The first part of the communication is the transmitter; the structure of the 16-QAM OFDM Transmitter is shown in Figure 2.

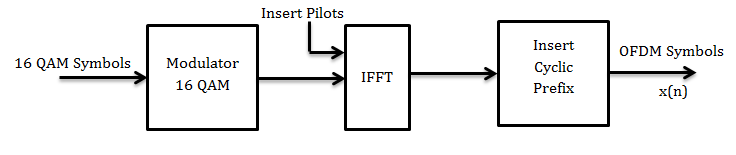


Figure : Transmitter structure

The signal to send has a length of 2048 Individual symbols and those symbols of data are randomly generated each time. The random generation needs to be seeded when the simulation starts, as implemented in the function below:

function [RN, RS] = initialise()

clear; clc; close all; % Clear workspace

RN = sum(100\*clock); % reset random number generator

RS = RandStream('mt19937ar','seed',RN);

RandStream.setGlobalStream(RS);

End

The function uses the computer clock to reset the random number generator at the start of the program so that different random values will be obtained in different MATLAB sessions, so that we can be sure by running the simulation more than once that the result is not dependent on some improbable starting condition that just so happens to be the one that we always use.

Then, within the main loop, the function Transmitter() is called to generate the 2048 symbols of random information (where N is a constant given the value 2048) which are put into a 1-2048 column matrix D. This is done for 16-QAM by first generating random integers between 0-15 that each correspond to 4 bits of binary information, These integers are used to mark position in the complex-plane QAM-16 constellation diagram, and then using a prepared lookup table to match the integer to the complex coordinate using gray encoding.

function [x, b] = Transmitter()

global N M cp L Pv C; % use globals from main function

D = zeros(1, N); % Initialize Frame (column matrix)

b = randi([0 (M-1)], 1, N);% Generate random symbols (0-15 for 16-QAM)

D = C(b+1); % Use C as look-up to modulate frame

D(1,1:L:end) = Pv; % Insert Pilots

d = sqrt(N)\*ifft(D); % Perform Inverse FFT and Normalize Energy

x = [d(end-(cp-1) : end) d];% Insert Cyclic Prefix

end

### Pilot Insertion

In order to detect the delay profile of the unknown multipath channel, 1 pilot of a known value (Pv = 3+3j) for every 8 subcarriers are inserted at the transmitter under normal operation, but pilot frequency can be increased or decreased as powers of 2 using the global variable L. To not generate pilots the single line of code D(1,1:L:end) = Pv; can be removed.

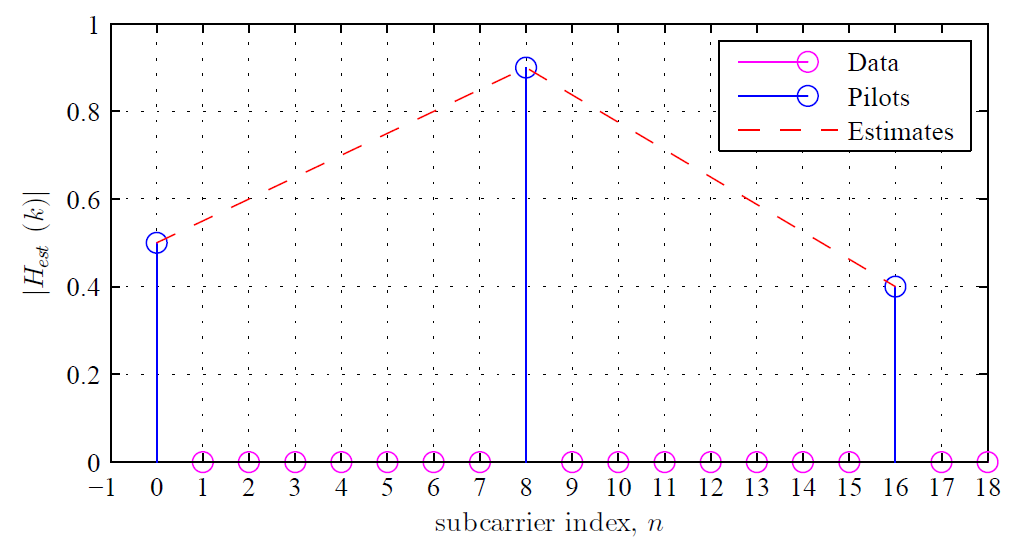


Figure : Pilot Insertion Diagram

After the symbols are generated, they are converted from the frequency domain into the time domain using the MATLAB Inverse Fast Fourier Transform function. The Fast Fourier Transform works on data where is size is a power of two, which is the case for a 2048 frame. If this was not the case the DFT would be employed, or the frame could be padded [13]. The IFFT then also needs to be normalized with respect to the Fast Fourier Transform used in the receiver, so that the energy of the transmission it kept constant, as the IFFT and FFT used have different scaling factors unless one does this. Finally the cyclic prefix is added by prepending the end of the frame to the start, the cyclic prefix length is set by the global variable cp.

## Channel

In this project, there are two different effects that can be applied in the channel: AWGN and Multipath Channel.

### AWGN

AWGN is the abbreviation of *additive white Gaussian noise*. It is random noise that follows the Gaussian distribution curve, which is a good statistical model of thermal noise.

In the program, the function Channel()is used to implement a channel of AWDN and Multipath noise, and is shown as follows:

function [z,zf] = Channel(x, stddev, h, zf)

global N cp sv; % use globals from main function

[y,zf] = filter (h, 1, x, zf); % Multipath noise added using fir filter

w = stddev\*(randn(1,N+cp) + (1j\*randn(1,N+cp))); % create AWGN of frame size

z = w+y; % add AWGN to the signal

end

The Variable stddev, the standard deviation required for appropriate scaling, and is pre-computed by the function SigmaVector()for each SNR during simulation initialization phase outside the main loop, random gaussian distributed noise is added to the real and imaginary parts for each symbol using the library function randn()resulting in the noise frame ‘w’.

The function combines the AWGN with multipath channel. A diagram of how the multipath channel with impulse response *h(n)* and the AWGN frame *w(n)* are applied to the signal *x(n)* is shown in Figure 4.

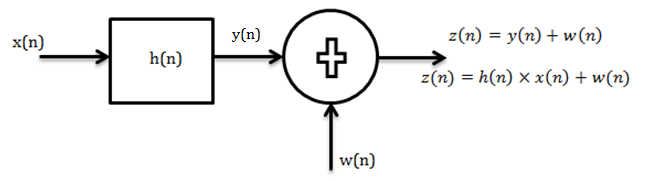


Figure : Model of multipath channel with AWGN

### Multipath

For the multipath the function filter() is used to apply the multipath channel impulse response to the signal using discrete convolution. The static multipath impulse response model used is the Pedestrian B Model defined by the ITU[14], which simulates the multipath noise encountered by a WiMax communications device used by pedestrian standing on the pavement in a built up city environment. The h parameter passed to the filter function in the code is the multipath magnitudes. This variable is created during initialization by the function multipath\_init() and contains the normalized magnitudes (normalized to sum to unity energy) that are generated from the relative negative gain values given in the model as P, which is stored in the code as an array containing 6 numbers: 0,-0.9,-4.9,-8.0,-7.8,-23.9. Appropriate zero-padding is required in h such that the normalized magnitudes appear at the indexes of the matrix corresponding to their delay spread also given in the Model, which is shown in Figure 5.

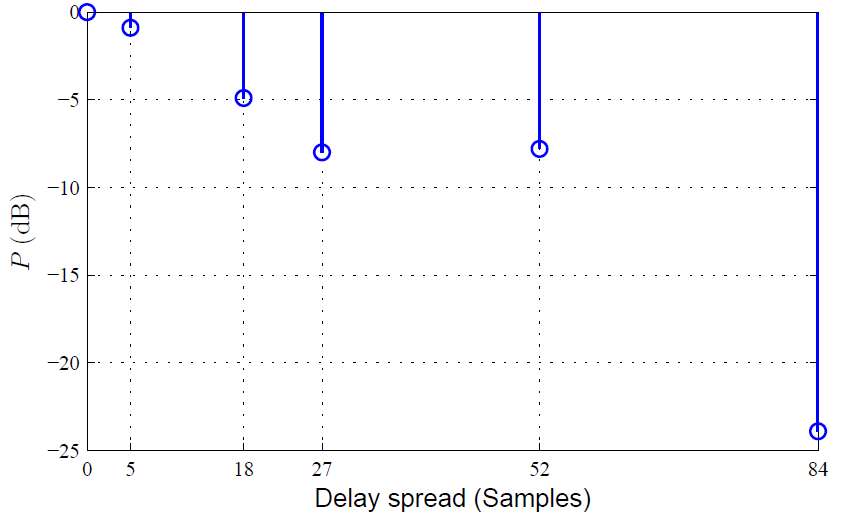


Figure : Impulse Reponses of ITU Pedestrian Model B Ch.103

The code below shows the implementation for channel normalization.

function [H, h] = multipath\_init()

global N P tau; % use globals from main function

h\_i = sqrt(10.^(P/10)); % Channel magnitude

U = sqrt(sum(abs(h\_i.^2))); % normalisation to unity i.e. sum(all)=1

h\_i = h\_i/U;

h = zeros(1,85); % initialize frequency response matrix (most zero)

for i = 1:length(tau); % generate full frequency response

h(tau(i)+1) = h\_i(i); % add each given multipath delay magnitudes

end

H = fft(h, N);

End

The Filter also uses a variable zf to keep filter state. This is critical to the use of the filter() function to produces discrete convolution of the Impulse Response to the original signal, as it allows for state feedback and keeps track of spillover between OFDM symbols and ensures that intercarrier/interblock interference effects are simulated. [15]

## Receiver

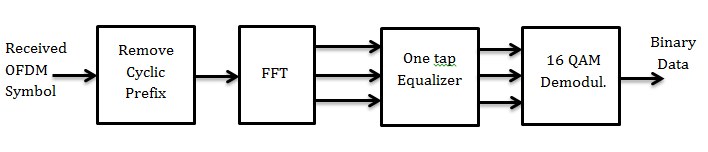


Figure : The receiver structure

Figure 6 shows the structure of the receiver for AWGN. For the multipath channel, it is necessary to add a one-tap equalizer between FFT and Demodulation to remove channel effect which is not shown.

The 16-QAM Demodulation procedure is same code for both AWGN and Multipath. In the code, after the normalized FFT is performed to convert the information back into the frequency domain, the minimum Euclidean distance approach is used to find the symbol that each received data point in the frame best matches. The code show as follows:

for i = 1:N % for each point in frame, find the

[val, pos] = min(abs(C-D(:,i))); % symbol with min euclidean distance

b\_est(i) = pos - 1; % and return index ie. demodulate

end

As for the known multipath channel, a one-tap equalizer is needed to add after the FFT demodulation.

one-tap equalizer

Where the H is produced from oversampling the h(n)impulse response of the channel, via a Fourier Transformation against the size of the transmitted frame, this is performed only once in the multipath\_init() function before the main loop.

When turning to consideration of the unknown channel, the equalizer must be different and utilize the pilots to estimate the multipath channels frequency response and use zero-forcing equalization[3] accordingly. The linear interpolation equation is used to obtain the channel values between two pilots. and for the special case of the last subcarriers received after the last pilot data point; an extrapolation from the last two pilots is used.

Hp = D(1,1:L:end)./Pv; % Estimate unknown channel at pilots

l = 1:L-1;

for m = 1:Np-1 % Interpolate all but the last bits

n = (m-1)\*L; % n is the OFDM index of the pilot

k = n+l+1; % ks are the OFDM indexs of the values

% H\_est(n+1) = Hp(m); % Add pilots in (not for main sim)

H\_est(k) = Hp(m) + l/L.\*(Hp(m+1)-Hp(m)); % add interpolated values

end

H\_est(L\*(Np-1)+2:N) = Hp(Np)+l/L.\*(Hp(Np)-Hp(Np-1)); % extrapolate last values



Figure : Diagrams of Linear Interpolation and Extrapolation

## Plotting Results

The results of a particular run of the simulation are drawn by using the function BERPlot(). In the function BERPlot(), the theoretical BER is calculated for comparison with the simulated results by following code for Multipath, and similar, yet simpler code for the AWGN only channel:

T\_BER\_16QAM\_in\_MP = 0;

for i = 1:N

T\_BER\_16QAM\_in\_MP = T\_BER\_16QAM\_in\_MP +

qfunc(sqrt(0.8\*snr\_linear\*(abs(H(i)^2))));

end

T\_BER\_16QAM\_in\_MP = 1/N \* T\_BER\_16QAM\_in\_MP;

T\_BER\_16QAM\_in\_MP = (1-((1-(1.5\*T\_BER\_16QAM\_in\_MP)).^2))/4;

The results of the simulation are plotted against the theoretical BER using the BER matrix generate by the simulation.

It can also be useful to plot the Mean-squared error of the channel estimation done by using pilots, a separate function, MSEPlot() is used to do this. Finally, for testing purposes it is useful to created constellation Plots of the frame prior to demodulation. The function ConstellationPlot() is used to do this, but by setting Breaking Points in the MATLAB debugging system so that execution of the script is paused after receiver equalization of a frame of data sent at a specified SNR. ConstellationPlot() is then run in the interpreter to generate the plots at the specific SNR value.

# Results

## BER for AWGN Channel

The simulated BER for the AWGN only channel is shown below, both with a 64 bit long cyclic prefix and without any cyclic prefix prepended. Note that the plot does not show the BER at 16dB, as this was 0 in both cases (i.e. no errors were produced with or without a cyclic prefix). This shows that, for our simulation, the cyclic prefix is not increasing the quality of the connection for basic AWGN noise.

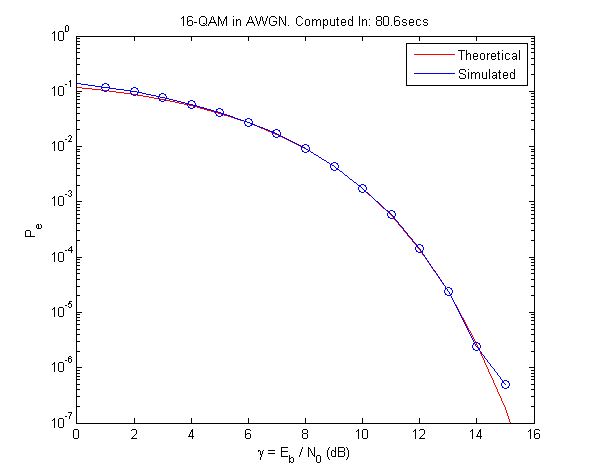


Figure : AWGN only with 64 bit CP

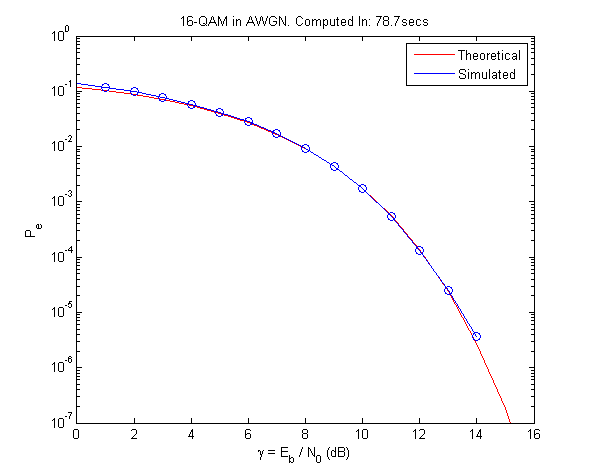


Figure : AWGN only without CP

From the two plots it can be seen that the simulated results match extremely closely with theory. They also introduce the basic principle that BER (bit error rate) decreases with increasing speed as the more signal ‘gets through’ the noise of the channel.

## BER for Multipath & AWGN Channel without Equalization

Without any equalization the multipath interference means that no matter the strength of the signal the BER remains high, such that this configuration would likely be unusable (or at least extremely inefficient) for any real world application. It can be seen that the performance does not follow the theoretical curve, although there is a small performance increase as the effects of AWGN are reduced. No visible difference was found between the use of a large cyclic Prefix compared to none, so only one plot (in this case with a cyclic prefix length of 128 bits) is shown:

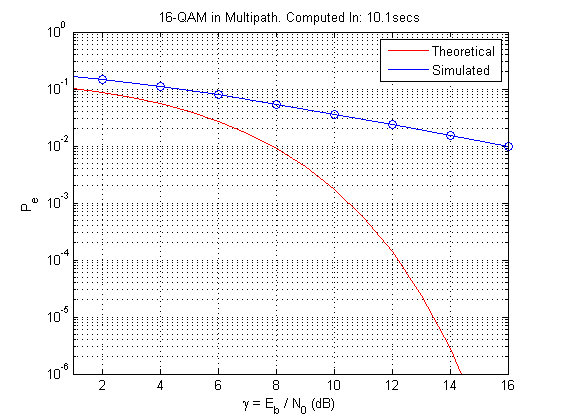


Figure : Multipath without Equalization, 128 bit CP

## BER for Multipath & AWGN Channel Using Known Channel Equalization

In order to mitigate the effect of multichannel interference, a one-tap equalizer is added in the receiver and the BER of the simulated result and the theoretical shows as following with various cyclic prefix lengths:

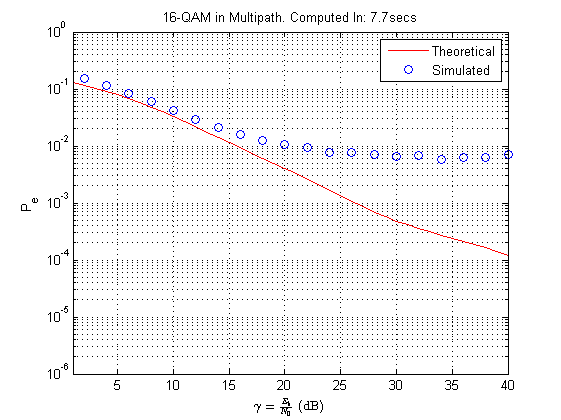


Figure : Known Multipath without CP

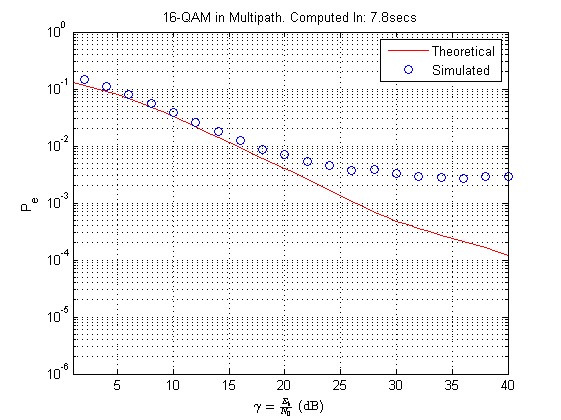


Figure : Known Multipath with 16 bit CP

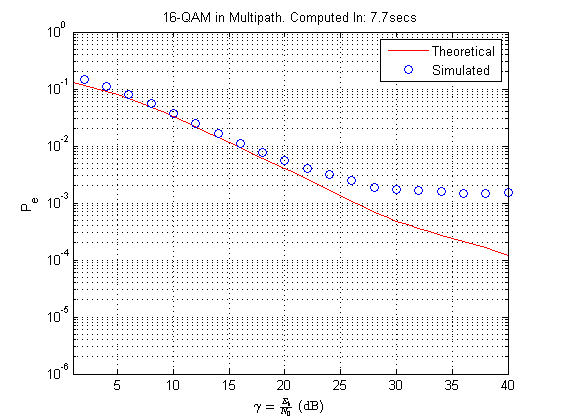


Figure : Known Multipath with 32 bit CP

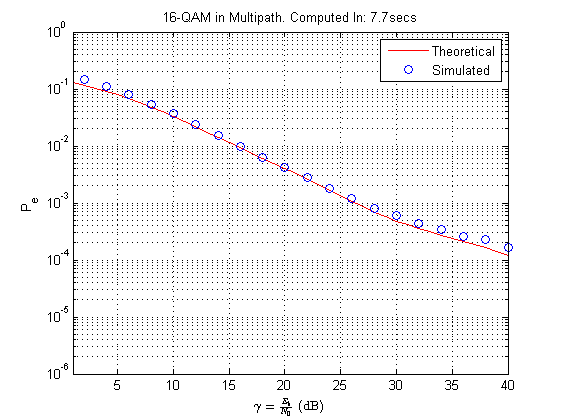


Figure : Known Multipath with 64 bit CP

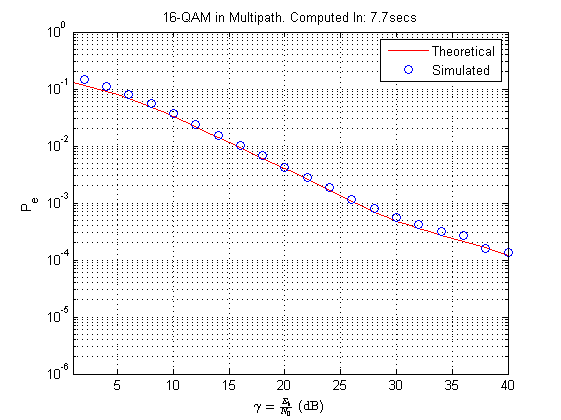


Figure : Known Multipath with 128 bit CP

While the significant performance improvement compared to that without an equalizer can be seen even without a cyclic prefix, it importance for getting as good as theoretically possible results when the signal is strong (greater than 20-25dB) can be verified in these plots. A significant difference in performance can be seen when using a cyclic prefix of 64 bits or above, compared to 32 bits or below.

## BER for Multipath &AWGN Channel Using Unknown Channel Equalization

Using pilots to estimate the frequency response of the multipath channel, rather than assume a known response, the BER is plotted as follows, for different cyclic prefixes:

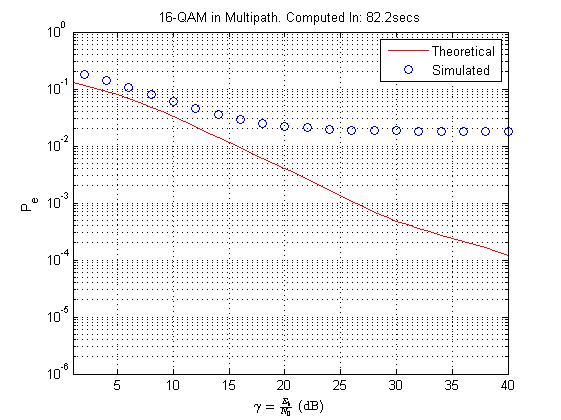


Figure : Unknown Multipath without CP

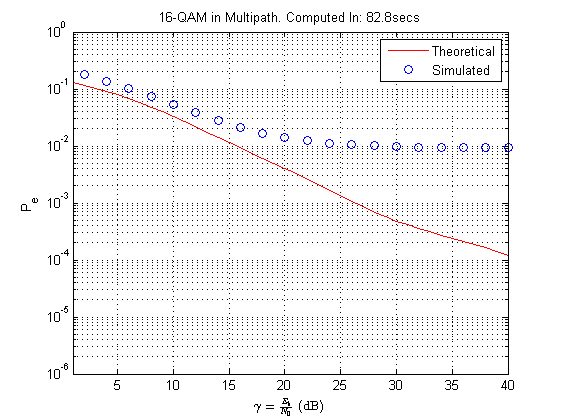


Figure : Unknown Multipath with 16 bit CP

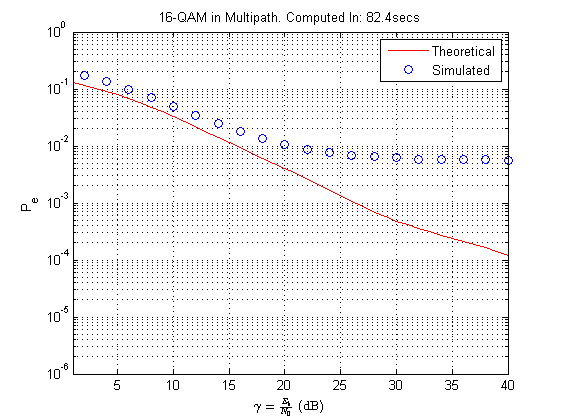


Figure : Unknown Multipath with 32 bit CP

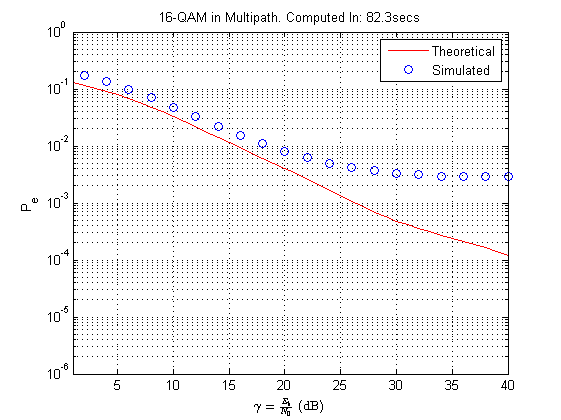


Figure : Unknown Multipath with 64 bit CP

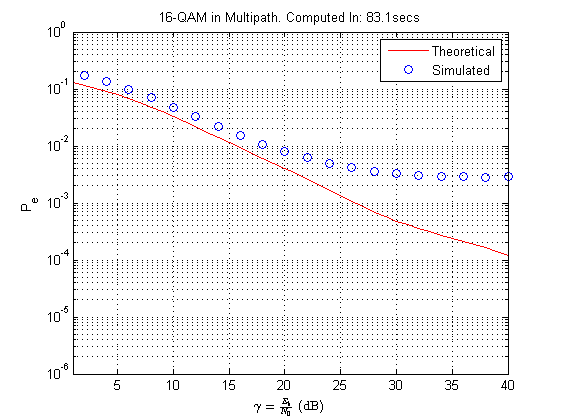


Figure : Unknown Multipath with 128 bit CP

In the case of an unknown Multipath channel the BER is increased at all points compared to that of the known channel, this happens even when ensuring to account for the pilot overhead in the BER calculation. The Multipath BER plots can be seen to all be significantly above the line of the theoretical model. This extra error rate must be accounted for in imprecision in the estimation of the frequency response of the multipath channel.

However the curve of the line the plots represent seems to follow a similar trend to that of that of the Known channel, with a fading off of the performance increases as the signal strength gets comparatively strong. However in this case, the CP length, while still helping to improve the overall performance, does not achieve the ability of cause that aligning of performance exactly to that of the theoretical model that was apparent in the Known channel simulations.

## Constellation Plots

For reference the symbols of the 16-QAM constellation are mapped as follows:

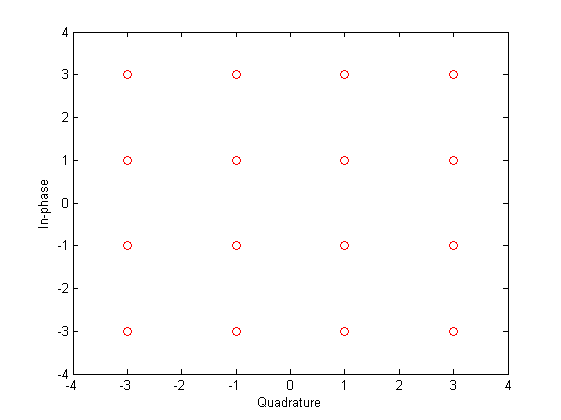


Figure : Constellation Points for 16-QAM

At the specific SNR points 5, 15 and 25dB, the constellation plot of the zero forcing equalizer’s output when the transmission is sent through an unknown Multipath with AWGN channel is plotted as follows:

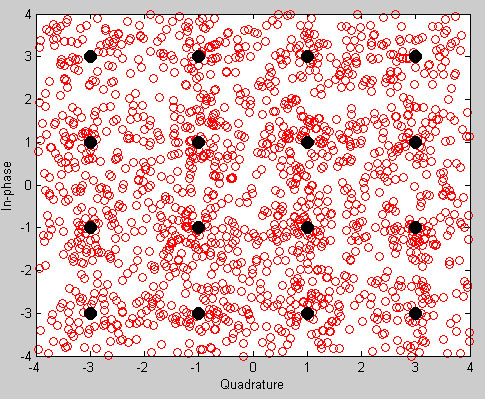


Figure : Constellation Plot; SNR = 5dB

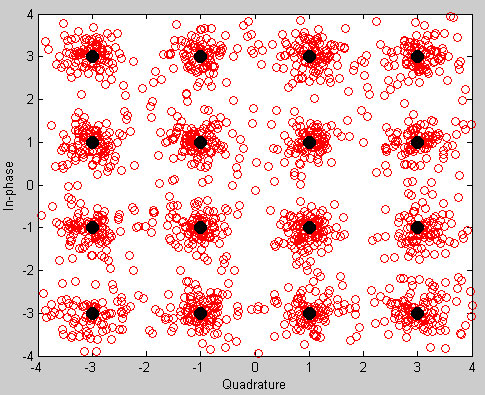


Figure : Constellation Plot; SNR = 15dB

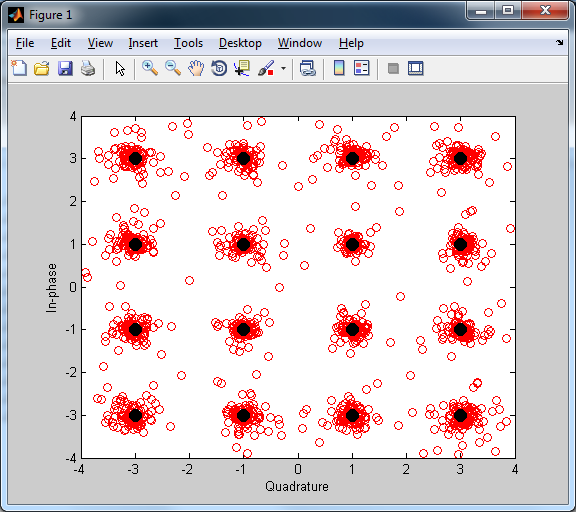


Figure : Constellation Plot; SNR=25dB

From the above three plots it is easy to see that when the SNR increases the data received in the signal gather increasingly close around the ideal 16-QAM constellation points. Conversely, when the SNR is low, the 16-QAM demodulator in the receiver which uses the minimum Euclidean distance approach will increasing assign the received points to the wrong symbol, hence the increase in the bit error rate.

## Mean-Squared Error Performance of Channel Estimation

The following shows the mean square error (MSE) of the unknown multipath channel estimation with different cyclic prefix lengths, the eMSE can be calculated by following formula eMSE=E{H(n)- Ĥ(n)}, Where the Ĥ(n) is estimated. The cyclic prefix lengths range from 0 to 128 and the results of MSE vs. SNR for three demonstrative lengths are shown below:

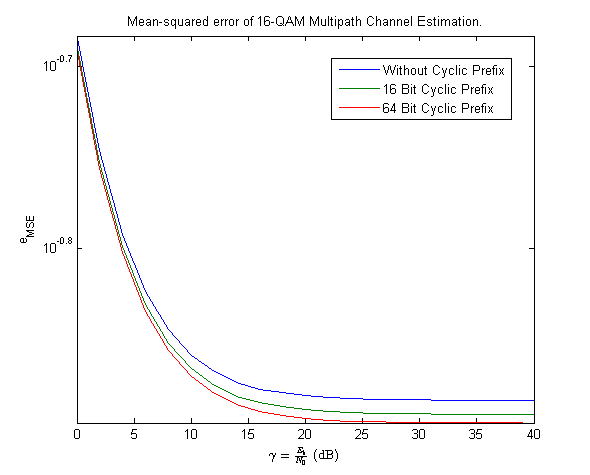


Figure : Mean-Squared-Errors for Three Cyclic Prefix Lengths

For each of the three cyclic prefix length the MSE vs. SNR relationship curve has an asymptotic decrease of from a maximum of around 0.2 down to a levelling off around 0. Between signal to noise level around 0dB and 10dB the Mean-Square errors falls very fast as signal strength improves in all cases. After the inflection point at around 10dB the improvements being to wear off as the errors approach 0. The Cyclic prefix length changes the error amount shown at strong signal strength, with a CP of 128 bits the MSE falls very close to zero at SNR over 30dB. With smaller lengths the Mean Squared Errors encountered seem to level off somewhat higher.

## Pilot Frequency

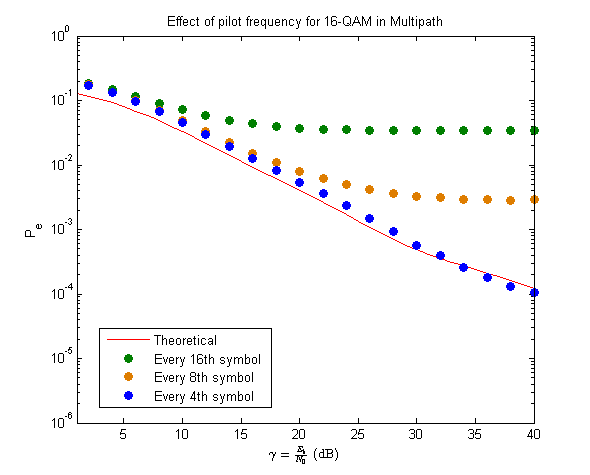


Figure : Effect of pilot frequency

In figure 27 the BER vs SNR noise for the unknown channel when uses a 16 bit CP are shown for three different pilot insertion schemes. By default the pilots are inserted every 8th symbol, if the pilots are inserted twice as frequently performance mirrors that of the theory quite well. If pilots are inserted half as frequently performance degrades significantly looking similar to figure 11 where no equalization is carried out.

# Discussion

Although the results seem to show something interesting, they are not of any value unless they can be shown to be correct. If any error was made in the setup of the simulation or in the assumptions of how and why a mathematical model fits as an approximation to real-world conditions and effects, then any conclusions made from them are in doubt. Thus it is important to both verify that each part conforms to its specification, and ensure that specification is a reasonable fit to requirements.

As the majority of the functionality of the transmitter and receiver are digital in design, then it can be expected that the simulations modelling of those systems will function similarly as both are of a like nature, being defined actions performed through computation running on a CPU. More difficult is the justify the design of the simulated channel effects, which seek to model natural phenomena, it needs to both be shown that the model approximates the more significant impairments to signal propagation in a wireless communications channel. Our model uses a combination of Gaussian Noise to simulate the unavoidable thermal effects within the electrical components of the transmitter and receiver and a mathematical model of multipath effects to simulate the self-interference effects of signal reflections that causes signal attenuation and data corruption in wireless propagation in real-world environments.

The noise due to the random effects of the thermal motion of the electrons flowing in conductors has long been recognized [4] to have a Gaussian probability distribution. To model this through the addition of pseudo-randomly generated, but normally distributed random numbers from the MATLAB randn() function applied to both the real and imaginary parts of the OFDM symbol frame can be seen from the results of channel noise in the constellation plots in figures 23-25, as the noise is increased the transmitted symbols are scattered wider from the original constellation points, but not equally, the distribution falls off at the edges, so that as can be seen best in figure 24 most still cluster not far from the mean, thus following a Standard/Gaussian distribution with a reasonably low standard deviation.

The Multipath channel effects of reflections are modelled mathematically using a Channel Impulse Response (CIR) model and are performed in the MATLAB code using the stock filter() function. This is possible because the simulated model is time independent, i.e. we can simulate so fast any change to the CIR appears very slowly and static multipath channel can be assumed. Thus we can treat the channel as time-invariant, hence the possible use of the filter() function. Otherwise much more complex models would be needed, given a variable multipath impulse response.

The results show that OFDM with 16-QAM modulation can be made to transmit data somewhat reliably through a reasonable approximation of real world noise and interference effects. Figure 11, which plot the results without an equalizer in the receiver show that good equalization is essential for the explored communication model to be useful, as without it the BER stays high no matter how high the SNR.

With a known multipath channel, performance that replicates the theoretical model is possible. However this occurs fairly suddenly when a high CP (above 64 bits) is used. This indicates the importance of a CP in avoiding intercarrier interference. It is interesting to note that 64 is equal to the constellation size multiplied by the symbol size in bits (64 = 16 x 4).

The most important results from a practical standpoint are shown primarily through figures 17-21 which shows the BER for a Multipath & AWGN Channel using unknown channel equalization. Distilling the data down, below is a table of BER for the different CP lengths when signal strength is good, i.e. where it flattens out above an SNR of around 20-25dB:

|  |  |
| --- | --- |
| CP Length | BER |
| 0 | 0.02 |
| 16 | 0.01 |
| 32 | 0.006 |
| 64 | 0.003 |
| 128 | 0.002 |

The table shows that if the simulation is correct OFDM with 16-QAM modulation on its own can be an effective model in transmitting data at the level of a coded speech signal when combined with a Cyclic Prefix as it achieves a BER in the 10-2-10-3 range.

The longer the Cyclic Prefix the better the BER, however the CP is an overhead that reduces the bandwidth of the communication system. It would seem intuitively that a CP of 64 bits gives the but compromise between error reduction and overhead, especially given the effects seen in figure 11 as discussed above. However, when calculated, a BER improvement of 0.006 to 0.003 from 32 but CP to 64 bit CP represents a reduction in errors by approximately 1 symbol per 2048 symbol frame. The extra 32 bit used (8 Symbols) per frame would seem therefore to be a waste considering the error reduction they allow. The correct compromise between Cyclic Prefix length and acceptable BER ultimately depend on the application of the system, but it would seem that a shorter length such as 16 or 32 would be most suitable in most applications. No Cyclic Prefix at all reduces the reliability of the system below that required by coded speech, but more concretely, it means more errors are encountered than the overhead of a 16 bit CP. This indicates the desirability of always utilizing a CP when using this communication model.

The most significant factor on the BER at reasonable SNR strengths (above 20dB) seems to be errors induced by poor Channel estimation. The increase in performance shown in figure 27, when using a pilot insertion scheme of one pilot every 4 symbols would seem to increase performance significantly. However, for a frame of 2048 symbols, the overhead of a quarter of all data being overhead (thus 1536 data symbols per frame), rather than an eighth (with 1792 data symbols per frame) means that an overhead of 256 extra symbols is introduced. This overhead is inefficient when compared by the small number of errors that are avoided by using so many pilots.

With additional software error detection such as the use of a hamming code, the BER could be significantly improved without too much significant bandwidth reduction. Allowing the model to be a reasonable choice for general purpose wireless communications.

# Conclusions

In our project we discovered that OFDM with 16-QAM modulation, coupled with a cyclic prefix of 16 or 32 bits and pilots injected every 8 symbols works reasonably well as a communication model in simulation.

In carrying out this simulation a great deal was learnt about the methods used in designing efficient digital communication models.

# References

[1] Blake R., “Basic Electronic Communications”, West Publishing

[2] Simon M., “COFDM Technology Basics”, (www.webe.org/papers05/rohde-cofdm.pdf)

[3] Mark J. & Weihua Z., “Wireless Communications and Networking”, Prentice Hall, (pg 139-144 - 3rd Edition)

[4] Shannon C.E., "Communication in the presence of noise", Proc. Institute of Radio Engineers 37 (1949)

[5] Mugala S., Butime J. & Okello D., “Crest Factor Reduction of an OFDM/WiMAX Network”, IST-Africa 2011 Conference Proceedings

[6] Proakis J. G., “Digital Communications” (3rd Edition) (various usages throughout text)

[7] Horowitz P. & Hill W., “The Art of Electronics” (2nd Edition) (pg 431)

[8] Geier J., “How to: Define Minimum SNR Values for Signal Coverage”, (wireless-nets.com)

[9] [Johnson](http://publish.aps.org/search/field/author/J.%20B.%20Johnson) J. B., “Thermal Agitation of Electricity in Conductors”, Phys. Rev. 32, 97–109 (1928)

[10] Dmitry Chizhik, Jonathan Ling, Peter W. Wolniansky, Reinaldo A. Valenzuela, Nelson Costa, and Kris Huber (April 2003). "Multiple-Input–Multiple-Output Measurements and Modeling in Manhattan". *IEEE Journal on Selected Areas in Communications* **21** (3): 321–331

[11] Weisstein, Eric W. "Gray Code." MathWorld (mathworld.wolfram.com/GrayCode.html)

[12] Metropolis, N. and Ulam, S. "The Monte Carlo Method." J. Amer. Stat. Assoc. 44, 335-341, 1949.

[13] “FFT” MATLAB Help Documentaion, Mathworks

[14] Jain R., “Channel Models A Tutorial, (www.cse.wustl.edu/~jain)

[15] Won Gi Jeon, Kyung Hi Chang & Yong Soo Cho, “An equalization technique for orthogonal frequency-division multiplexing systems in time-variant multipath channels”, Communications, IEEE Transactions on (Volume:47 , Issue: 1, 1999 pg 27-32)

# Appendix

Below is the final version of the code, it is currently prepared for the simulation of the unknown channel. But can be modified by commenting out certain lines to perform simulations of other configurations of the simulated system:

function plot\_16QAM\_BER()

initialise();

% Make important constants global in scope

global N M cp L Np Pv C P tau snr\_dB snr\_linear sv;

N = 2048; % Frame Size

M = 16; % Constellation Size (16-QAM has 16 symbols)

cp = 128; % Cyclic Prefix Length (could be 0, 16, 32, or 64)

L = 8; % Frequency of pilots in the bit frame

Np = N/L; % Number of pilots in a frame

Pv = 3+3j; % Pilot Value

max\_dB = 40; % Maximum Signal to Noise Ratio Level Generated

step\_dB = 2; % Gap between SNR levels simulated from 0 to max\_dB

numsims = 500; % Number of times to simulate each SNR value.

BER\_list= []; % matrix to store the various BERs for each SNR value

MSE\_list= [];

C=[ -3+3j, -3+1j, -3-3j, -3-1j, ... % 16-QAM Constellation matrix

-1+3j, -1+1j, -1-3j, -1-1j, ...

3+3j, 3+1j, 3-3j, 3-1j, ...

1+3j, 1+1j, 1-3j, 1-1j ];

P = [0 -0.9 -4.9 -8.0 -7.8 -23.9]; % ITU Pedestrian Multipath mags

tau = [0 5 18 27 52 84]; % Delay of multipath magnitudes

snr\_dB = [0:1:max\_dB]; % Generate a range of SNR ratios

snr\_linear = 10.^(snr\_dB/10); % convert Log dB scale to Linear scale

sv = SigmaVector();

zf = []; % Initialise filter state (no interference yet)

[H, h] = multipath\_init();

disp('Starting Simulation');

time=0;

%matlabpool open % start default parallel computing system

% we try SNR values from 0dB to 40dB stepping by 2dB each time

for m = 0:1:max\_dB/step\_dB % regular loop: swappable with parallel

%parfor m = 0:1:max\_dB/step\_dB % parallel loop: swappable with regular

snr = m \* step\_dB;

errors = 0; % reset error counter for new SNR values

mse\_acc = 0; % reset mean square error accumulator

b\_est = zeros(1,N);

tic;

for n = numsims:-1:1

[x, b] = Transmitter(); % Generate 16-QAM Signal

[z, zf]= Channel(x, sv(snr+1), h, zf); % Send thru noisy channel

[b\_est,D,mse] = Receiver(z, H); % Retrieve info at other end

b(1,1:L:end) = 0; % ignore pilot positions - should be 9 in both b and b\_est

e\_sym = biterr(b, b\_est); % Compare with original

errors = (e\_sym/log2(M)) + errors; % nb. 4 bits per symbol...

mse\_acc = mse + mse\_acc;

end

BER = (errors/numsims)/(N-Np);

BER\_list = [BER\_list BER];

MSE = mse\_acc/numsims;

MSE\_list = [MSE\_list MSE];

disp(sprintf('\tSNR=%02.0fdB BER=%0.4f T=%0.1fs', snr, BER, toc));

time = toc + time;

end

disp('Ending Simulation');

%matlabpool close; % Close the parallel system down

BERPlot(BER\_list, time, H);

MSEPlot(MSE\_list);

end

function [RN, RS] = initialise()

clear; clc; close all; % Clear workspace

RN = sum(100\*clock); % reset random number generator

RS = RandStream('mt19937ar','seed',RN);

RandStream.setGlobalStream(RS);

end

function sv = SigmaVector()

global M C snr\_linear; % use globals from main function

mK = log2(M); % mK is bits per QAM symbol

Es = 1/M \* sum(abs(C).^2); % Energy per symbol

Eb = Es/mK; % Eb is energy per bit

En = Eb./snr\_linear; % snr = Eb/En - rearranged

sv = sqrt(En/2);

end

function [H, h] = multipath\_init()

global N P tau; % use globals from main function

h\_i = sqrt(10.^(P/10)); % Channel magnitude

U = sqrt(sum(abs(h\_i.^2))); % normalisation to unity i.e. sum(all)=1

h\_i = h\_i/U;

h = zeros(1,85); % initialize frequency response matrix (most zero)

for i = 1:length(tau); % generate full frequency response

h(tau(i)+1) = h\_i(i); % add each given multipath delay magnitudes

end

H = fft(h, N);

end

function [x, b] = Transmitter()

global N M cp L Pv C; % use globals from main function

D = zeros(1, N); % Initialize Frame (column matrix)

b = randi([0 (M-1)], 1, N);% Generate random symbols (0-15 for 16-QAM)

D = C(b+1); % Use C as look-up to modulate frame

D(1,1:L:end) = Pv; % Insert Pilots

d = sqrt(N)\*ifft(D); % Perform Inverse FFT and Normalize Energy

x = [d(end-(cp-1) : end) d];% Insert Cyclic Prefix

end

function [z,zf] = Channel(x, stddev, h, zf)

global N cp sv; % use globals from main function

[y,zf] = filter (h, 1, x, zf); % Multipath noise added using fir filter

w = stddev\*(randn(1,N+cp) + (1j\*randn(1,N+cp))); % create AWGN of frame size

z = w+y; % add AWGN to the signal

end

function [b\_est D mse] = Receiver(y, H)

global N cp L Pv Np C; % use globals from main function

d = y(cp+1:end); % Remove the Cyclic Prefix

D = 1/sqrt(N)\*fft(d); % Perform FFT and Normalize Energy

Hp = D(1,1:L:end)./Pv; % Estimate unknown channel at pilots

l = 1:L-1;

for m = 1:Np-1 % Interpolate all but the last bits

n = (m-1)\*L; % n is the OFDM index of the pilot

k = n+l+1; % ks are the OFDM indexs of the values

% H\_est(n+1) = Hp(m); % Add pilots in (not for main sim)

H\_est(k) = Hp(m) + l/L.\*(Hp(m+1)-Hp(m)); % add interpolated values

end

%H\_est(((Np-1)\*L)+1) = Hp(Np); % Add final Pilot in (not for main sim)

H\_est(L\*(Np-1)+2:N) = Hp(Np)+l/L.\*(Hp(Np)-Hp(Np-1)); % extrapolate last values

D = D./H\_est; % 1 tap equalizer

for i = 1:N % for each point in frame, find the

[val, pos] = min(abs(C-D(:,i))); % symbol with min euclidean distance

b\_est(i) = pos - 1; % and return index ie. demodulate

end

mse = mean(abs(H - H\_est).^2); % compute the mean-square error

end

function BERPlot(BER, time, H)

global N snr\_dB snr\_linear;

% Calculate Theoretical Response

T\_BER\_16QAM\_in\_MP = 0;

for i = 1:N

T\_BER\_16QAM\_in\_MP = T\_BER\_16QAM\_in\_MP + qfunc(sqrt(0.8\*snr\_linear\*(abs(H(i)^2))));

end

T\_BER\_16QAM\_in\_MP = 1/N \* T\_BER\_16QAM\_in\_MP;

T\_BER\_16QAM\_in\_MP = (1-((1-(1.5\*T\_BER\_16QAM\_in\_MP)).^2))/4;

semilogy(snr\_dB, T\_BER\_16QAM\_in\_MP,'r-'); % plotted for comparison

hold on;

semilogy(snr\_dB(1:2:end), BER,'bo'); % Plot BER points

ylim([1e-06 1]); xlim([1 40]);

ylabel('P\_e'); xlabel('$\gamma = \frac{E\_b}{N\_0}$ (dB)','interpreter','latex');

legend('Theoretical', 'Simulated');grid on;

title(sprintf(strcat('16-QAM in Multipath. Computed In: %0.1fsecs'), time));

end

function MSEPlot(MSE)

global snr\_dB;

figure(2);

semilogy(snr\_dB(1:2:end), MSE);

ylabel('e\_{MSE}'); xlabel('$\gamma = \frac{E\_b}{N\_0}$ (dB)','interpreter','latex');

title('Mean-squared error of 16-QAM Multipath Channel Estimation.');

end

function ConstellationPlot(frame)

global C;

plot(frame,'ro');

hold

plot(C,'ko','MarkerFaceColor','k','MarkerSize',10); % Black dots added (like figure on page 25 of handout

axis([-4 4 -4 4]);

ylabel('In-phase');

xlabel('Quadrature');

end