

**CHANNEL ADAPTIVE PROCESS RESILIENT ULTRA LOW-POWER  
TRANSMITTER DESIGN WITH SIMULATED-ANNEALING BASED SELF-  
DISCOVERY**

A Thesis

Presented to

The Academic Faculty

By

Keertana Mutnuri

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science in E.C.E

Georgia Institute of Technology

May 2014

Copyright © 2014 by Keertana Mutnuri

CHANNEL ADAPTIVE PROCESS RESILIENT ULTRA LOW-POWER TRANSMITTER  
DESIGN WITH SIMULATED-ANNEALING BASED SELF-DISCOVERY

Approved by:

Dr. Abhijit Chatterjee, Advisor  
School of Electrical and Computer Engineering  
*Georgia Institute of Technology*

Dr. David Keezer  
School of Electrical and Computer Engineering  
*Georgia Institute of Technology*

Dr. Madhavan Swaminathan  
School of Electrical and Computer Engineering  
*Georgia Institute of Technology*

Date Approved: January 07, 2014

## ACKNOWLEDGEMENTS

Over the last two years I have received support and encouragement from a great number of individuals. Dr. Abhijit Chatterjee has been a great mentor and his guidance has made this a thoughtful and rewarding journey. I would like to thank my committee members, Dr. David Keezer and Dr. Madhavan Swaminathan for their support as I moved from an idea to a completed study.

During the formation of the idea, Debashis Banerjee spent countless hours explaining me concepts useful for my research. He provided the much needed encouragement and insights. I have learnt a lot through our conversations. Finally, thanks to Charan Langton for making a wonderful all-inclusive tutorial on OFDM (which got me started in my research) and for letting me use certain concepts from it in my thesis.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
LIST OF ABBREVIATIONS.....	ix
SUMMARY.....	x
CHAPTER 1: Introduction to Orthogonal Frequency Division Multiplexing.....	1
1.1 The Importance of being orthogonal.....	2
1.2 An example of OFDM using 4 sub-carriers.....	3
1.3 Using Inverse FFT to create OFDM symbol.....	8
1.4 Defining Fading.....	13
1.5 Properties of OFDM.....	21
CHAPTER 2 : PAPR Problem in OFDM and need for companding.....	26
2.1 Companding.....	26
2.2 A-Law.....	27
2.3 $\mu$ -Law.....	29
CHAPTER 3 : Role of Power Amplifier in an OFDM system.....	34
3.1 Non-linear effects of Power Amplifier on OFDM system.....	34
3.2 Process variations in a Power Amplifier.....	38
CHAPTER 4 : Need for Simulated Annealing.....	40
4.1 What is Simulated Annealing.....	41
4.2 Advantages of Simulated Annealing.....	41
4.3 Acceptance Function.....	42

4.4 Algorithm Overview.....	43
4.5 Temperature Initialisation.....	44
4.6 Example.....	44
CHAPTER 5 : Analysis of OFDM packet structure for protocol level implications.....	46
5.1 Frame Structure.....	46
5.2 Bit-rate Calculation.....	48
5.3 Coherence-time Considerations.....	49
5.4 Error Vector Magnitude Measurement.....	51
5.5 Relation between parameters.....	52
CHAPTER 6 : Implementation.....	54
6.1 Simulation of an OFDM system.....	54
6.2 Code to simulate OFDM.....	55
6.3 Companding.....	58
6.4 Adding a power amplifier and analysis.....	61
6.5 Simulated Annealing for our application.....	67
6.6 Results.....	70
6.7 Conclusions.....	76
REFERENCES.....	77

## LIST OF TABLES

Table 1.1	Serial to parallel conversion of data bits.....	4
Table 1.2	Parameters of real OFDM.....	25
Table 5.1	OFDM Parameters for bit-rate calculation.....	48
Table 5.2	Summary.....	51
Table 6.1	Basic definitions.....	55

## LIST OF FIGURES

Figure 1.1	Multicarrier FDM and Multi-code division multiplex.....	2
Figure 1.2	The area under a sine and cosine wave over one period is always zero.....	2
Figure 1.3	The area under a sine wave multiplied by its own harmonic is always zero.....	3
Figure 1.4	A bit stream that will be modulated using a 4-carrier OFDM.....	4
Figure 2.1	A-Law compressor characteristics.....	29
Figure 2.2	$\mu$ -Law compressor characteristics.....	31
Figure 2.3	Comparison between A-Law and $\mu$ -Law.....	32
Figure 3.1	OFDM signal before passing through high power amplifier.....	35
Figure 3.2	OFDM signal after passing through high power amplifier.....	36
Figure 4.1	Explaining SA.....	42
Figure 4.2	Algorithm.....	43
Figure 4.3	TSP.....	44
Figure 4.4	TSP Solution.....	45
Figure 5.1	One OFDM Symbol.....	47
Figure 6.1	OFDM diagram.....	54
Figure 6.2	Compression using Mu-Law.....	58
Figure 6.3	Positive axis for Mu-Law.....	59
Figure 6.4	Negative axis for Mu-Law.....	59
Figure 6.5	Expansion for Mu-Law.....	60
Figure 6.6	Positive axis for Mu-Law.....	60

Figure 6.7	Negative axis for Mu-Law.....	61
Figure 6.8	Mu vs EVM.....	62
Figure 6.9	Mu vs PL.....	64
Figure 6.10	Minimum power vs PL.....	65
Figure 6.11	Power convergence.....	71
Figure 6.12	EVM convergence.....	72
Figure 6.13	Power convergence across channels.....	73
Figure 6.14	EVM convergence across channels.....	74



## LIST OF ABBREVIATIONS

ACPR	Adjacent Channel Power Ratio
AM	Amplitude Modulation
BER	Bit-error rate
EVM	Error Vector Magnitude
FDM	Frequency Division Multiplexing
ISI	Inter-symbol interference
OFDM	Orthogonal Frequency Division Multiplexing
PA	Power Amplifier
PAPR	Peak to average power ratio
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature phase shift keying
RF	Radio Frequency
SA	Simulated Annealing

## SUMMARY

Modern day wireless communication systems are constantly facing increasing bandwidth demands due to a growing consumer base. To cope up with it, they are required to have a better power vs performance from the RF devices. The amount of data being exchanged over wireless links has tremendously increased and simultaneously, there is a need to switch to portable RF devices and this has in turn forced the issue of low-power RF system design. Therefore, what we need is an RF transceiver that operates at high data rates and over adverse channels with a low power consumption.

A major portion of the power is utilized by the RF front end of the wireless system. Many methods like controlled positive feedback, reutilizing bias current, etc have been employed to reduce the power consumption of the RF front end. The most modern wireless systems adapt to the channel quality by adjusting the data transmission rates and by adjusting the output power of the RF Power Amplifier. However, each of these methods concentrates on working for the worst case channel and giving the highest data rate. What needs to be known is that the channel conditions are not always worst. Even for a normal channel, the system is going to utilize a lot of power and give the highest possible data rate which may or may not be necessary. And thus, for the most part, the system is going to use up more power than necessary.

What we need instead, is a system which works nominally for a normal channel and exhaustively for a harsh channel condition. This requires the system to adapt to the channel conditions. Also another major factor causing fluctuations in the performance is the process

variations. This calls for a channel-dependent dynamic transceiver with adequate power management and tuning.

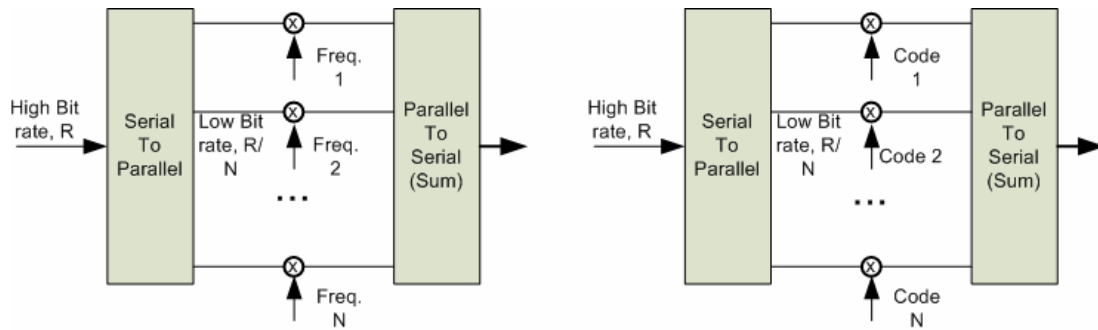
In our work, we try to devise a method to dynamically minimize the power considering the varying channel conditions and process variations. We first use companding to reduce the dynamic range of the signal so that it can be used on facilities with smaller dynamic range. This brings down the transmitted power. We also create multiple instances of the Power Amplifier to simulate process variations. After finding the optimum tuning knob settings for one instance of the PA, we try to use it to obtain the optimum settings for another instance. This requires the use of some heuristics and in our work, we have supplemented it with Simulated Annealing. Using SA, we can dynamically tune the power of a system for changing channel conditions and existing process variations. Towards the end, we have also proved that the slower the cooling rate of the experiment, the more elaborate the search space is and the more accurate the result is.

## CHAPTER 1

### Introduction to orthogonal frequency division multiplexing

OFDM is a combination of modulation and multiplexing. Multiplexing generally refers to independent signals, those produced by different sources. So it is a question of how to share the spectrum with these users. In OFDM, the question of multiplexing is applied to independent signals but these independent signals are a sub-set of the one main signal. In OFDM, the signal itself is first split into independent channels, modulated by data and then re-multiplexed to create the OFDM carrier. OFDM is essentially, a special case of Frequency Division Multiplex (FDM). OFDM has an advantage over FDM when it comes to its robust response to interference. In the presence of interference, the whole data in FDM is polluted while in OFDM,  $1/N$  of the data is corrupt while the rest is preserved.

Every individual subcarrier must be orthogonal to the other in OFDM. Subsequently, the independent sub-channels can be multiplexed by frequency division multiplexing (FDM), called multi-carrier transmission or it can be based on a code division multiplex (CDM), in this case it is called multi-code transmission.

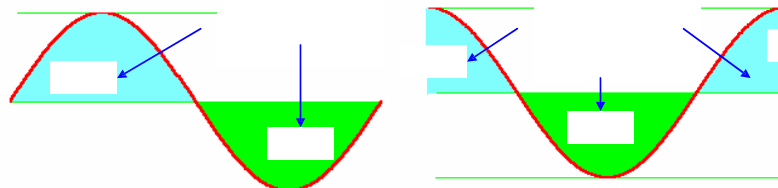


**Figure 1.1 Multi-carrier FDM and Multi-code division multiplex**

### 1.1 The importance of being orthogonal

The main concept in OFDM is orthogonality of the sub-carriers. Since the carriers are all sine/cosine wave, we know that area under one period of a sine or a cosine wave is zero. This is easily shown in figure 1.2.

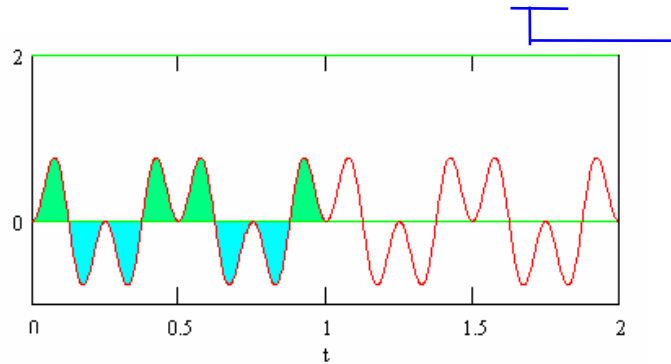
*Positive and negative area cancel.*



**Figure 1.2. The area under a sine and a cosine wave over one period is always zero.**

Let's take a sine wave of frequency  $m$  and multiply it by a sinusoid (sine or a cosine) of a frequency  $n$ , where both  $m$  and  $n$  are integers. The integral or the area under this product is given by

$$f(t) = \sin mwt \times \sin nwt$$



**Figure 1.3 The area under a sine wave multiplied by its own harmonic is always zero**

By the simple trigonometric relationship, this is equal to a sum of two sinusoids of frequencies  $(n-m)$  and  $(n+m)$

$$= \frac{1}{2} \cos(m - n) - \frac{1}{2} \cos(m + n)$$

These two components are each a sinusoid, so the integral is equal to zero over one period as shown in figure 1.3.

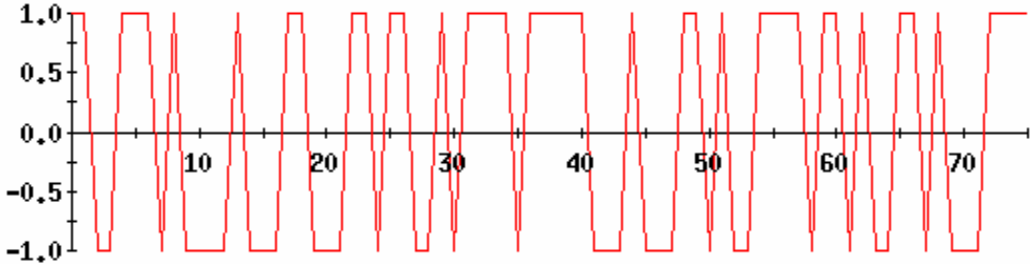
$$= \int_0^{2\pi} \left( \frac{1}{2} \cos(m - n) - \frac{1}{2} \cos(m + n) \right) \omega t$$

We conclude that when we multiply a sinusoid of frequency  $n$  by a sinusoid of frequency  $m/n$ , the area under the product is zero. In general, for all integers  $n$  and  $m$ ,  $\sin nx$ ,  $\sin mx$ ,  $\cos nx$ ,  $\cos mx$  are all orthogonal to each other. These frequencies are called harmonics. This idea is key to understanding OFDM. The orthogonality allows simultaneous transmission on a lot of sub-carriers in a tight frequency space without interference from each other. In essence this is similar to CDMA, where codes are used to make data sequences independent (also orthogonal) which allows many independent users to transmit in same space successfully.

## 1.2 An example of OFDM using 4 sub-carriers

In OFDM we have  $N$  carriers,  $N$  can be anywhere from 16 to 1024 in present technology

and depends on the environment in which the system will be used. Let's examine the following bit sequence shown in figure 1.4 that we wish to transmit and show the development of the OFDM signal using 4 sub-carriers. The signal has a symbol rate of 1 and sampling frequency is 1 sample per symbol, so each transition is a bit.



**Figure 1.4 A bit stream that will be modulated using a 4 carrier OFDM.**

Let's now write these bits in rows of fours, since this demonstration will use only four sub- carriers. We have effectively done a serial to parallel conversion.

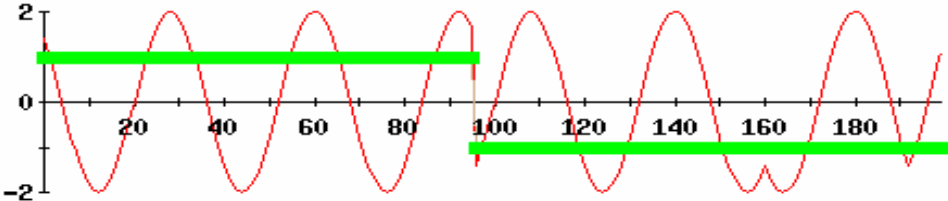
**Table 1.1 Serial to parallel conversion of data bits.**

C1	C2	C3	C4
1	1	-1	-1
1	1	1	-1
1	-1	-1	-1
-1	1	-1	-1
-1	1	1	-1
-1	-1	1	1

Each column in Table 1.1 represents the bits that will be carried by one sub-carrier. Let's start with the first carrier, c1. From the Nyquist sampling Theorem, we know that smallest frequency that can convey information has to be twice the information rate. In this case, the information rate per carrier will be 1/4 or 1 symbol per second total for all 4 carriers. So the smallest frequency that can carry a bit rate of 1/4 is 1/2 Hz. But we picked 1 Hz for convenience. Had we picked 1/2 Hz as our starting frequency, then the harmonics would have been 1, 3/2 and 2 Hz. We could have chosen 7/8 Hz to start with and in which the harmonics would be 7/4, 7/2, 21/2 Hz.

We pick BPSK as our modulation scheme for this example. (For QPSK, just imagine the same thing going on in the Q channel, and then double the bit rate while keeping the symbol rate the same.) Note that we can pick any other modulation method, QPSK, 8PSK 32-QAM or whatever. No limit here on what modulation to use. We can even use TCM which provides coding in addition to modulation.

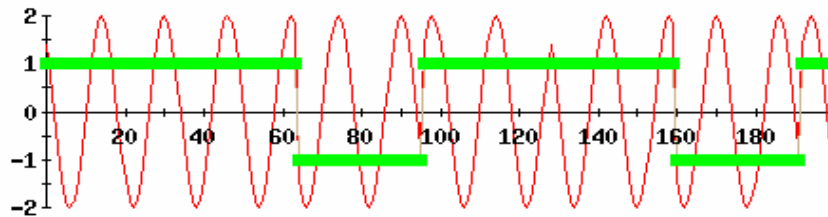
Carrier 1 - We need to transmit 1, 1, 1 -1, -1, -1 which we show below superimposed on the BPSK carrier of frequency 1 Hz. First three bits are 1 and last three -1. If we had shown the Q channel of this carrier (which would be a cosine) then this would be a QPSK modulation.



**Figure 1.5. Sub-carrier 1 and the bits it is modulating (the first column of Table 1.1)**

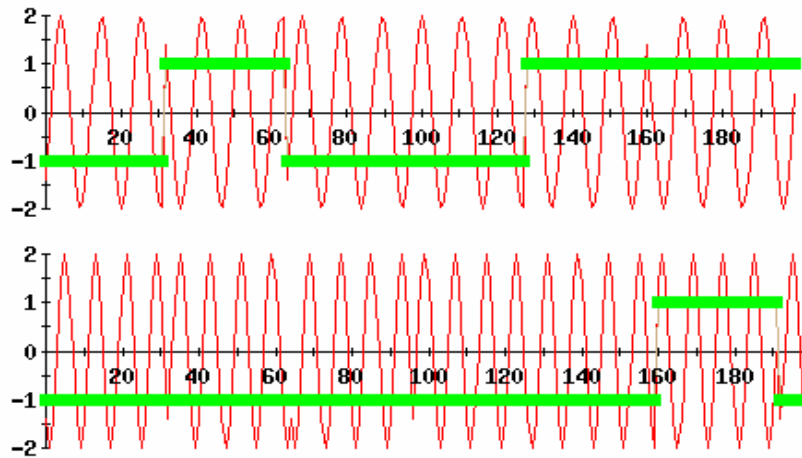


Carrier 2 - The next carrier is of frequency 2 Hz. It is the next orthogonal/harmonic to frequency of the first carrier of 1 Hz. Now we take the bits in the second column, marked c2, 1, -1, 1, 1, -1 and modulate this carrier with these bits as shown in Figure 1.6.



**Figure 1.6. Sub-carrier 2 and the bits that it is modulating (the 2nd column of Table 1.1)**

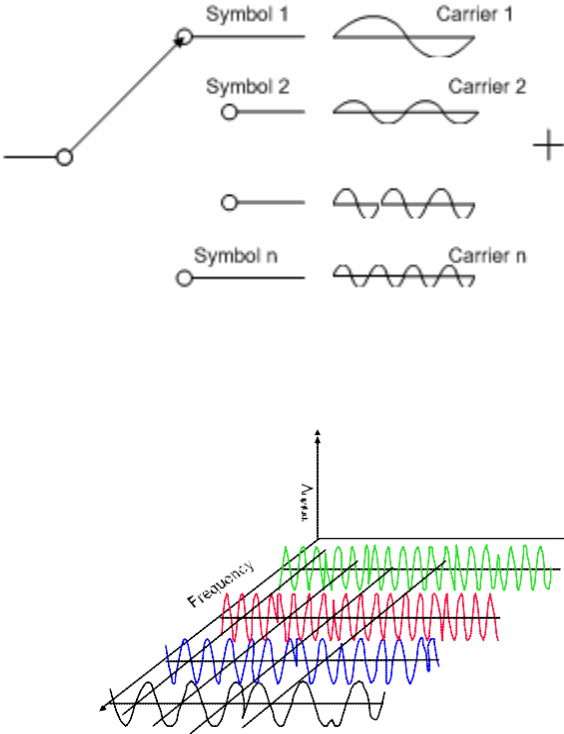
Carrier 3 – Carrier 3 frequency is equal to 3 Hz and fourth carrier has a frequency of 4 Hz. The third carrier is modulated with -1, 1, 1, -1, -1, 1 and the fourth with -1, -1, -1, -1, -1, -1, 1 from Table 1.1.



**Figure 1.7 Sub-carrier 3 and 4 and the bits that they modulating (the 3<sup>rd</sup> and 4th columns of Table 1.1)**

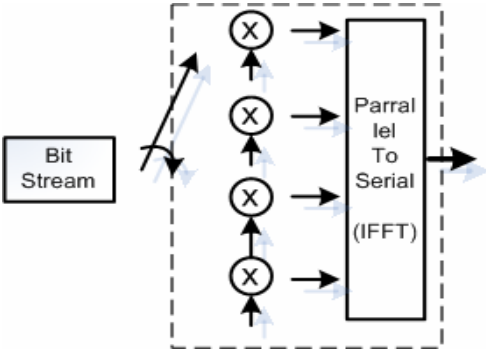
Now we have modulated all the bits using four independent carriers of orthogonal

frequencies 1 to 4 Hz. What we have done is taken the bit stream, distributed the bits, one bit at a time to the four sub-carriers as shown in figure 1.8.

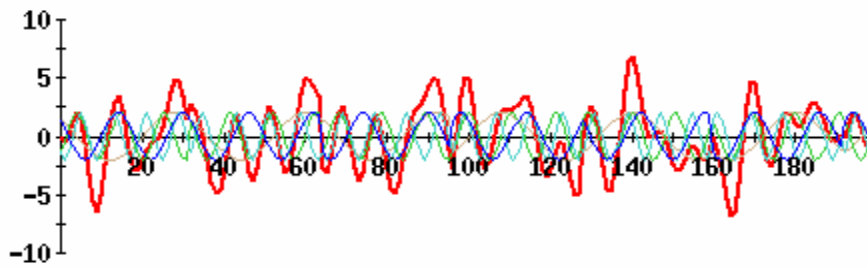


**Figure 1.8. OFDM signal in time and frequency domain.**

Now adding all four of these modulated carriers to create the OFDM signal, often produced by a block called the IFFT.



**Figure 1.9. Functional diagram of an OFDM signal creation.**



**Figure 1.10. The generated OFDM signal.**

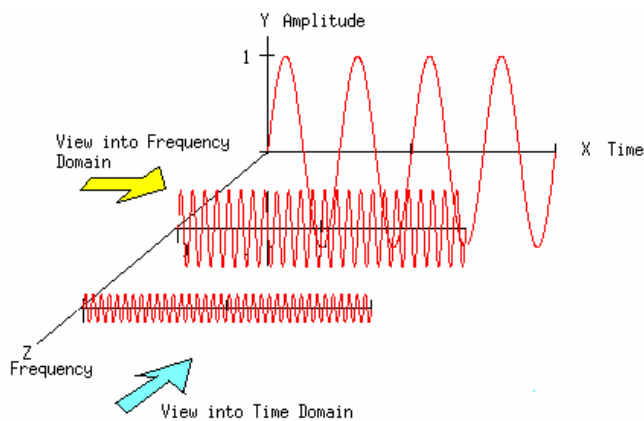
In short-hand, we can write the process above as

$$c(t) = \sum_{n=1}^N m(t)\sin(2\pi nt)$$

This is basically the equation for inverse FFT.

### 1.3 Using Inverse FFT to create the OFDM symbol

The equation above is essentially an inverse FFT. So let's examine briefly what an FFT/IFFT does. For that, observe figure 1.11 which gives the two views of the signal.



**Figure 1.11 Two views of a signal**

Forward FFT takes a random signal, multiplies it successively by complex exponentials over the range of frequencies, sums each product and plots the results as a coefficient of that frequency. The coefficients are called a spectrum and represent “how much” of that frequency is present in the input signal. The results of the FFT in common understanding is a frequency domain signal.

We can write FFT in sinusoids as

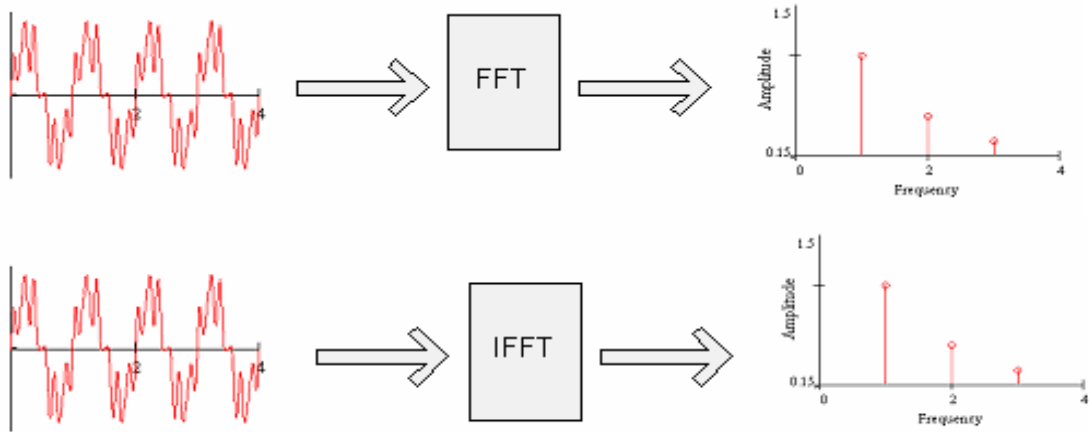
$$x(k) = \sum_{n=0}^{N-1} x(n) \sin\left(\frac{2\pi kn}{N}\right) + j \sum_{n=0}^{N-1} x(n) \cos\left(\frac{2\pi kn}{N}\right)$$

Here  $x(n)$  are the coefficients of the sines and cosines of frequency  $2\pi k / N$ , where  $k$  is the index of the frequencies over the  $N$  frequencies, and  $n$  is the time index.  $x(k)$  is the value of the spectrum for the  $k$ th frequency and  $x(n)$  is the value of the signal at time  $n$ . The inverse FFT takes this spectrum and converts the whole thing back to time domain signal by again successively multiplying it by a range of sinusoids.

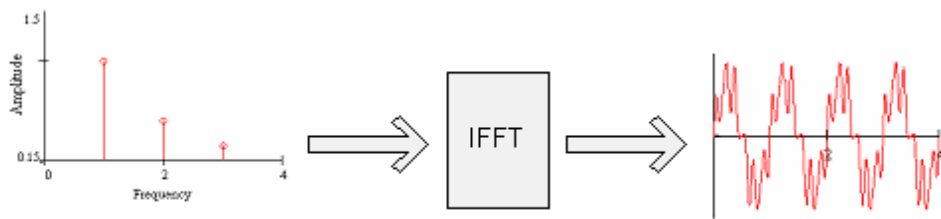
The equation for IFFT is

$$X(n) = \sum_{k=0}^{N-1} x(k) \sin\left(\frac{2\pi kn}{N}\right) - j \sum_{k=0}^{N-1} x(k) \cos\left(\frac{2\pi kn}{N}\right)$$

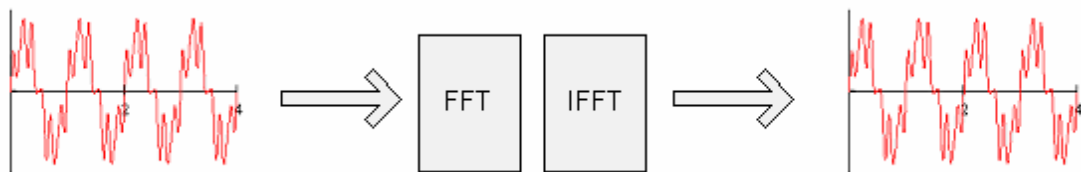
The difference between the above two equations is in the type of coefficients the sinusoids are taking, and the minus sign. The coefficients by convention are defined as time domain samples  $x(k)$  for the FFT and  $X(n)$  frequency bin values for the IFFT. The two processes are a linear pair. Using both in sequence will give the original result back.



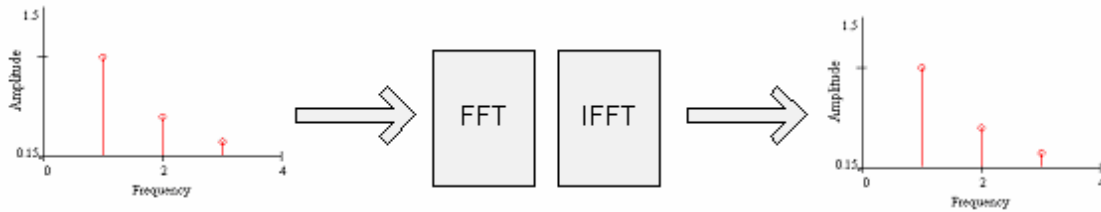
**Figure 1.12. A time domain signal comes out as a spectrum out of a FFT and IFFT.**



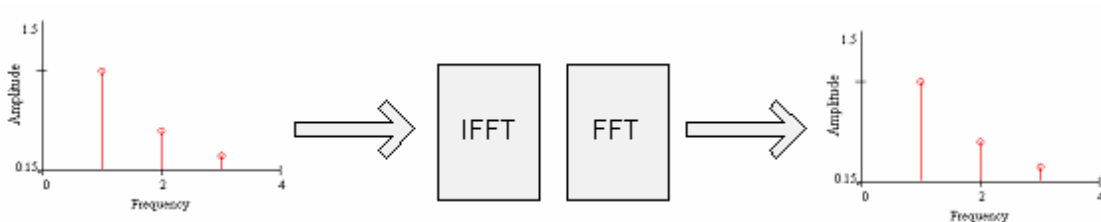
**Figure 1.13 A frequency domain signal comes out as a time domain signal out of a IFFT.**



**Figure 1.14 The pair return back the original input.**



**Figure 1.15 The pair return back the input no matter what it is.**



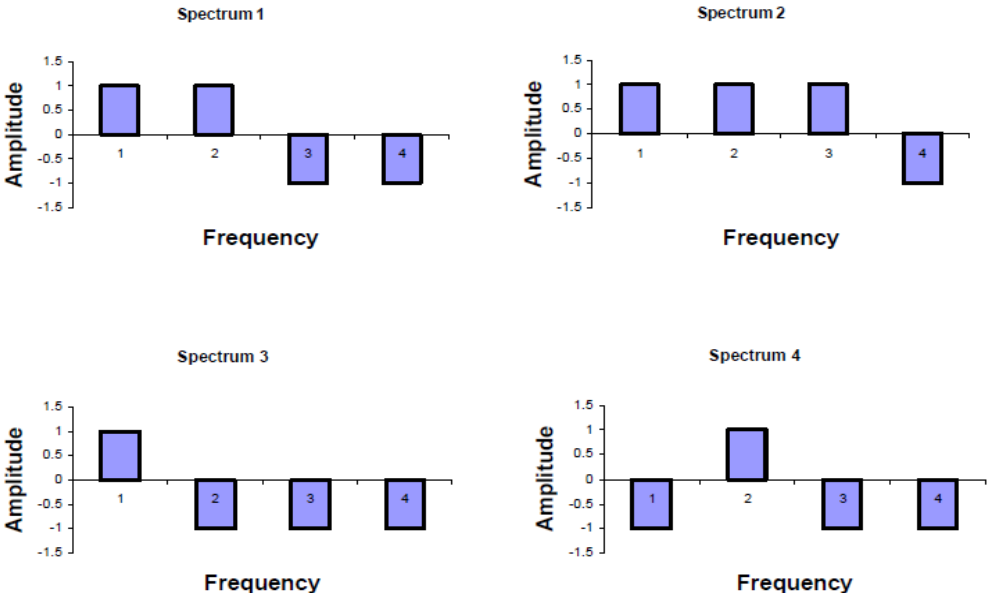
**Figure 1.16 The pair is commutable so they can be reversed and they will still return the original input.**

In Column 1 of Table 1.1, the signal bits can be considered to be of the amplitudes of a certain range of sinusoids. So we can use the IFFT to produce a time domain signal. They are already in time domain. Now how can we process a time domain signal to produce an another time domain signal? The answer is that we pretend that the input bits are not time domain representations but are frequency amplitudes. In this way, we can take these bits and by using the IFFT, we can create an output signal which is actually a time-domain OFDM signal.

The IFFT is a mathematical concept and does not really care what goes in and what goes out. As long as what goes in is amplitudes of some sinusoids, the IFFT will crunch these numbers to produce a correct time domain result. Both FFT and IFFT will produce identical results on the same input.

Each row of Table 1.1 can be considered a spectrum as plotted below. These rows aren't actually spectrums, but that does not matter. Each row spectrum has only 4 frequencies

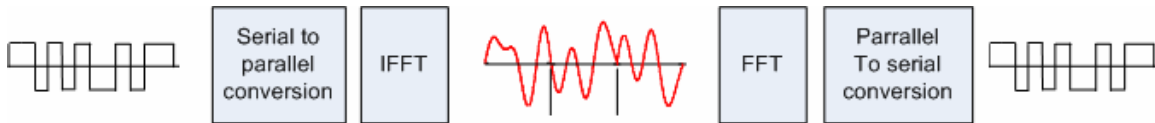
which are 1, 2, 3 and 4 Hz . Each of these spectrums can be converted to produce a time-domain signal which is exactly what an IFFT does. Only in this case, the input is really a time domain signal disguising as a spectrum.



**Figure 1.17 The incoming block of bits can be seen as a four bin spectrum**

IFFT quickly computes the time-domain signal instead of having to do it one carrier at a time and then adding. Calling this functionality IFFT may be more satisfying because we are producing a time domain signal. Because FFT and IFFT are linear processes and completely reversible, it should be called a FFT instead of a IFFT. The results are the same whether you do FFT or IFFT.

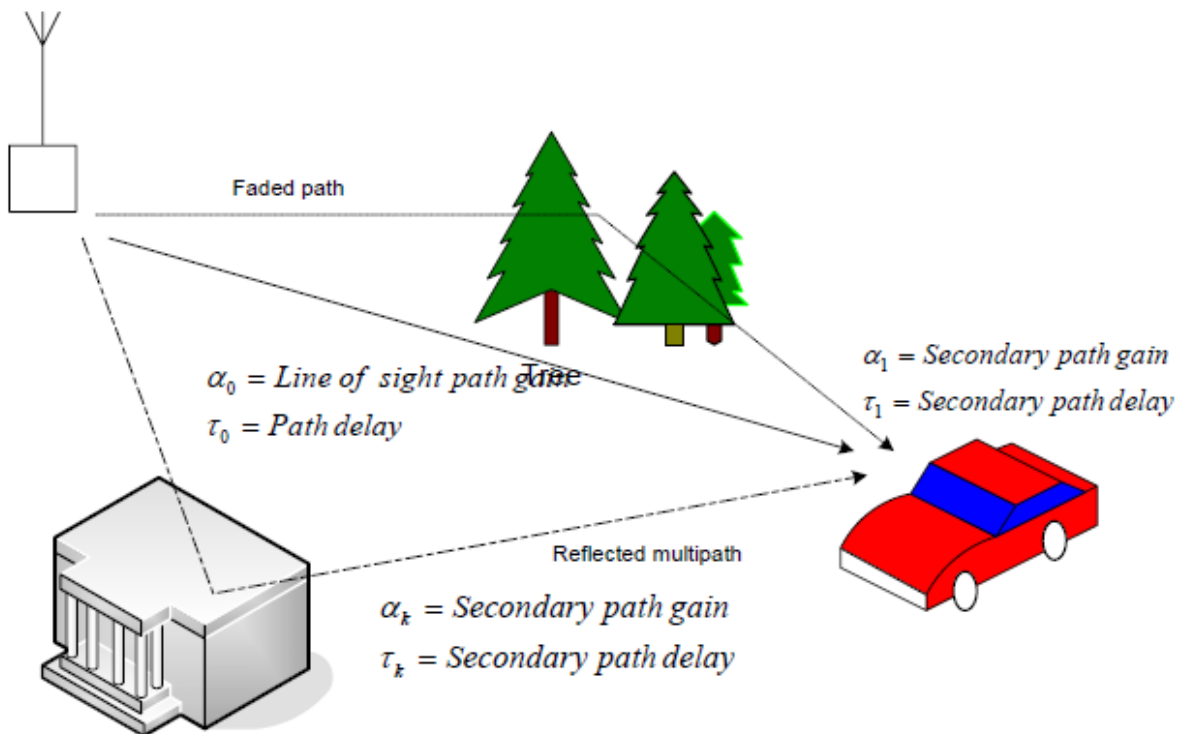
The functional block diagram of how the signal is modulated/demodulated is given next.



**Figure 1.18 The OFDM link functions**

### 1.4 Defining fading

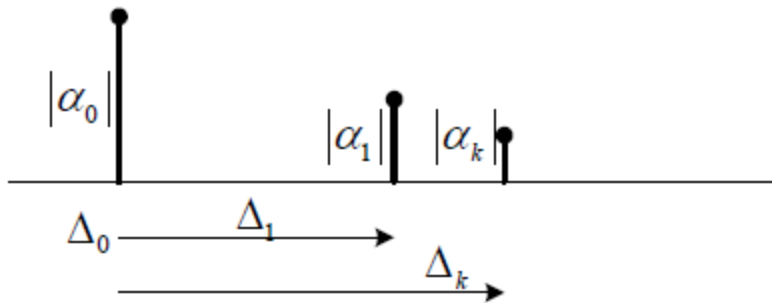
If the path from the transmitter to the receiver either has reflections or obstructions, we can get fading effects. In this case, the signal reaches the receiver from many different routes, each a copy of the original. Each of these rays has a slightly different delay and slightly different gain. The time delays result in phase shifts which added to main signal component (assuming there is one.) causes the signal to be degraded.



**Figure 1.19 Fading is big problem for signals**



If we draw the interferences as impulses, they look like this



**Figure 1.20 Reflected signals arrive at a delayed time period**

In fading, the reflected signals that are delayed add to the main signal and cause either gains in the signal strength or deep fades. And by deep fades, we mean that the signal is nearly wiped out. The signal level is so small that the receiver cannot decide what was there.

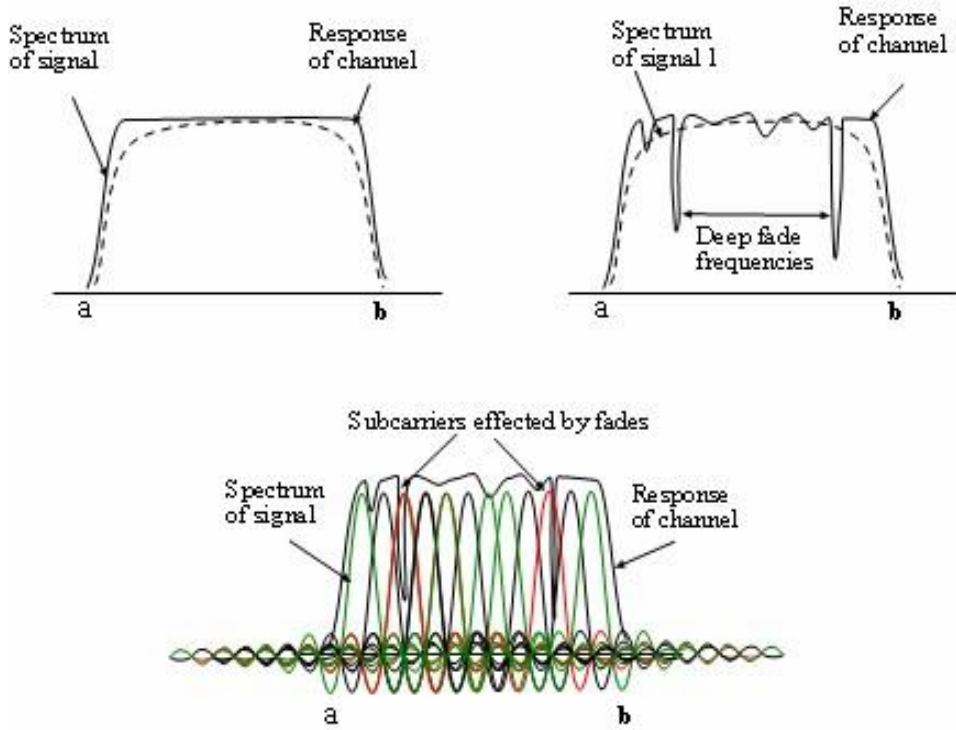
The maximum time delay that occurs is called the delay spread of the signal in that environment. This delay spread can be short so that it is less than symbol time or larger. Both cases cause different types of degradations to the signal. The delay spread of a signal changes as the environment is changing as all cell phone users know.

Figure 1.20 above shows the spectrum of the signal, the dark line shows the response we wish the channel to have. It is like a door through which the signal has to pass. The door is large enough that it allows the signal to go through without bending or distortion. A fading response of the channels is something like the one shown in Figure 1.21, we note that at some frequencies in the band, the channel does not allow any information to go through, so they are called deep fade frequencies. This form of channel frequency response is called frequency selective fading because it does not occur uniformly across the band. It occurs at selected

frequencies. And the environment selects these frequencies. If the environment is changing such as for a moving car, then this response is also changing and the receiver must have some way of dealing with it.

Rayleigh fading is a term used when there is no direct component and all signals reaching the receiver are reflected. This type of environment is called Rayleigh fading.

In general when the delay spread is less than one symbol, we get what is called flat fading. When delay spread is much larger than one symbol then it is called frequency-selective fading.



**Figure 1.21 Effect of fading**

An OFDM signal offers an advantage in a channel that has a frequency selective fading response. As we can see, when we lay an OFDM signal spectrum against the frequency-

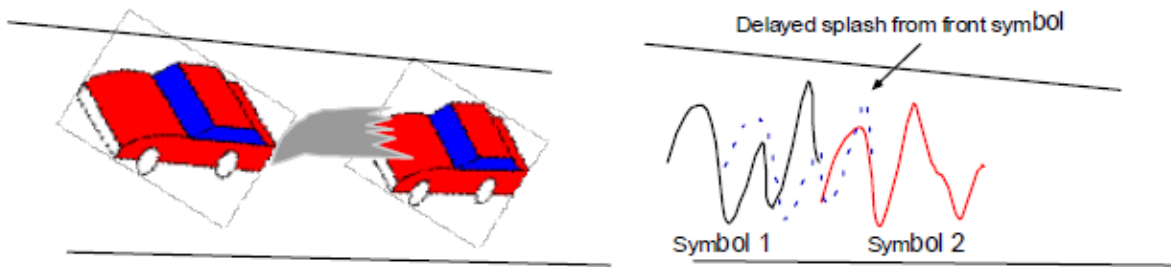
selective response of the channel, only two sub-carriers are affected, all the others are perfectly OK. Instead of the whole symbol being knocked out, we lose just a small subset of the  $(1/N)$  bits. With proper coding, this can be recovered.

The BER performance of an OFDM signal in a fading channel is much better than the performance of QPSK/FDM which is a single carrier wideband signal. Although the underlying BER of a OFDM signal is exactly the same as the underlying modulation, that is if 8PSK is used to modulate the sub-carriers, then the BER of the OFDM signal is same as the BER of 8PSK signal in Gaussian channel. But in channels that are fading, the OFDM offers far better BER than a wide band signal of exactly the same modulation. The advantage here is coming from the diversity of the multi-carrier such that the fading applies only to a small subset.

In FDM carriers, often the signal is shaped with a Root Raised Cosine shape to reduce its bandwidth, in OFDM since the spacing of the carriers is optimal, there is a natural bandwidth advantage and use of RRC does not buy us as much.

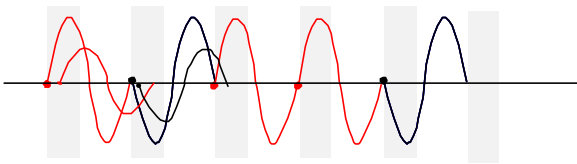
#### 1.4.1 Delay spread and the use of cyclic prefix to mitigate it

If we are driving in rain, and the car in front splashes water, we move further back far enough so that the splash won't reach us. If we equate the reach of splash to delay spread of a splashed signal then we have a better picture of the phenomena and how to avoid it.



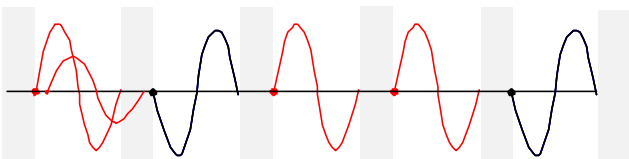
**Figure 1.22 Delay spread**

Delay spread is like the undesired splash we might get from the car ahead of us. In fading, the front symbol similarly throws a splash backwards which we wish to avoid.



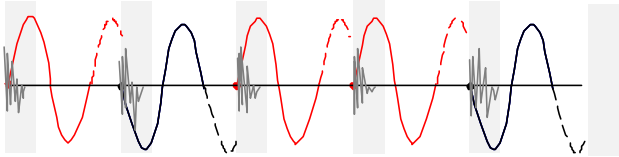
**Figure 1.23 Symbol and its splash**

In composite, these splashes become noise and affect the beginning of the next symbol. To mitigate this noise at the front of the symbol, we will move our symbol further away from the region of delay spread as shown below. A little bit of blank space has been added between symbols to catch the delay spread.



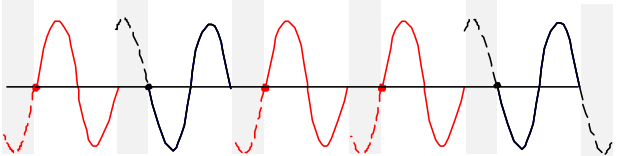
**Figure 1.24 No interference to the next symbol**

But we cannot have blank spaces in signals. This is won't work for the hardware which likes to crank out signals continuously. So it's clear we need to have something there. Let us just extend the symbol for now.



**Figure 1.25 Symbol extension**

If we just extend the symbol, then the front of the symbol which is important to us since it allows us to figure out the phase of the symbol, is now corrupted by the “splash”. Next, we extend the symbol into the empty space, so the actual symbol is more than one cycle. But now the start of the symbol is still in the danger zone, and this start is the most important thing about our symbol since the slicer needs it in order to make a decision about the bit. We do not want the start of the symbol to fall in this region, so lets just slide the symbol backwards, so that the start of the original symbol lands at the outside of this zone. And then fill this area with something.



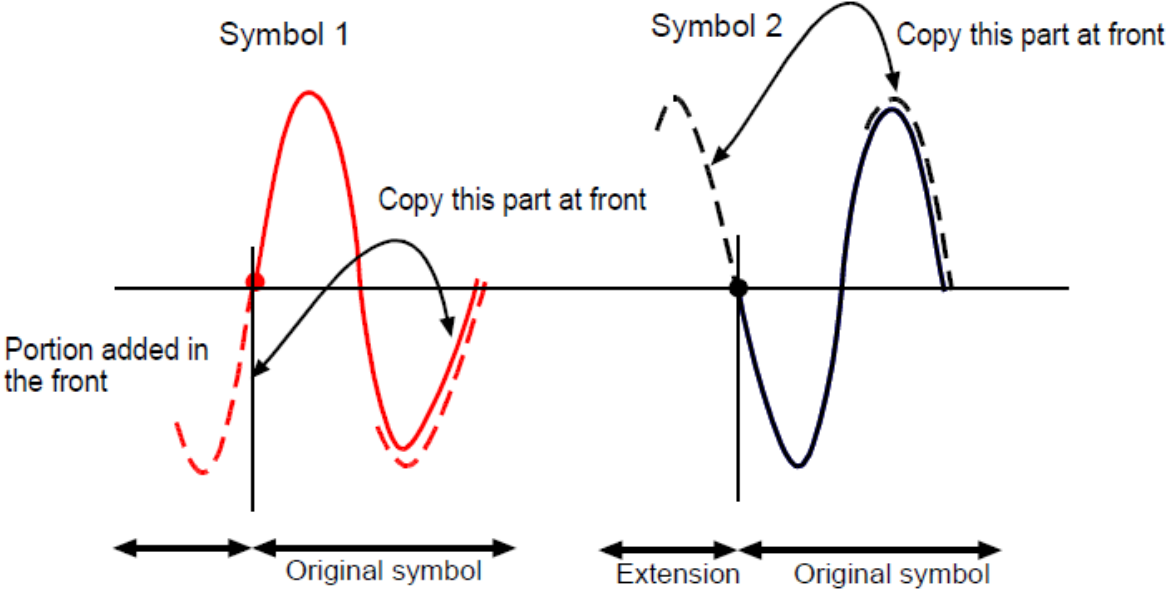
**Figure 1.26 Symbol Filler**

If we move the symbol back and just put in a convenient filler in this area, then not only do we have a continuous signal but one that can get corrupted and we don't care since we will

just cut it out anyway before demodulating. We slide the symbol to start at the edge of the delay spread time and then fill the guard space with a copy of what turns out to be tail end of the symbol.

1. We want the start of the symbol to be out of the delay spread zone so it is not corrupted and
2. We start the signal at the new boundary such that the actual symbol edge falls outside this zone.

We will be extending the symbol so it is 1.25 times as long, to do this, we copy the back of the symbol and glue it in the front. In reality, the symbol source is continuous, so all we are doing is adjusting the starting phase and making the symbol period longer.



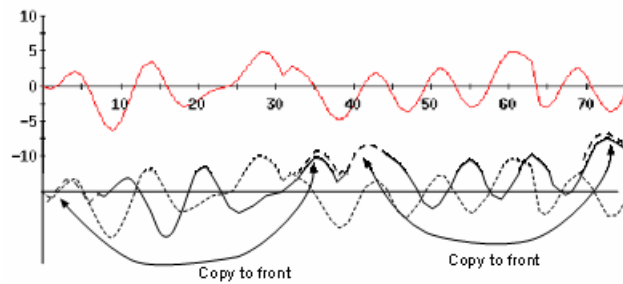
**Figure 1.27 Addition of Cyclic prefix**

This procedure is called adding a cyclic prefix. Since OFDM, has a lot of carriers, we would do this to each and every carrier. But that’s only in theory. In reality since the OFDM

signal is a linear combination, we can add cyclic prefix just once to the composite OFDM signal. The prefix is anywhere from 10% to 25% of the symbol time.

Here is an OFDM signal with period equal to 32 samples. We want to add a 25% cyclic shift to this signal.

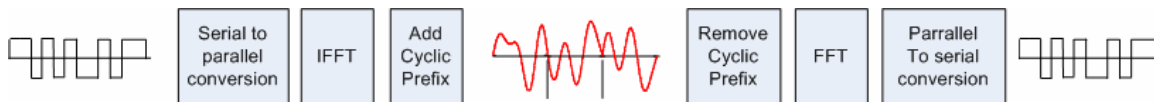
1. First we cut pieces that are 32 samples long.
2. Then we take the last  $.25 (32) = 8$  samples, copy and append them to the front as shown.



**Figure 1.28 Adding cyclic shift**

We add the prefix after doing the IFFT just once to the composite signal. After the signal has arrived at the receiver, first remove this prefix, to get back the perfectly periodic signal so it can be FFT'd to get back the symbols on each carrier.

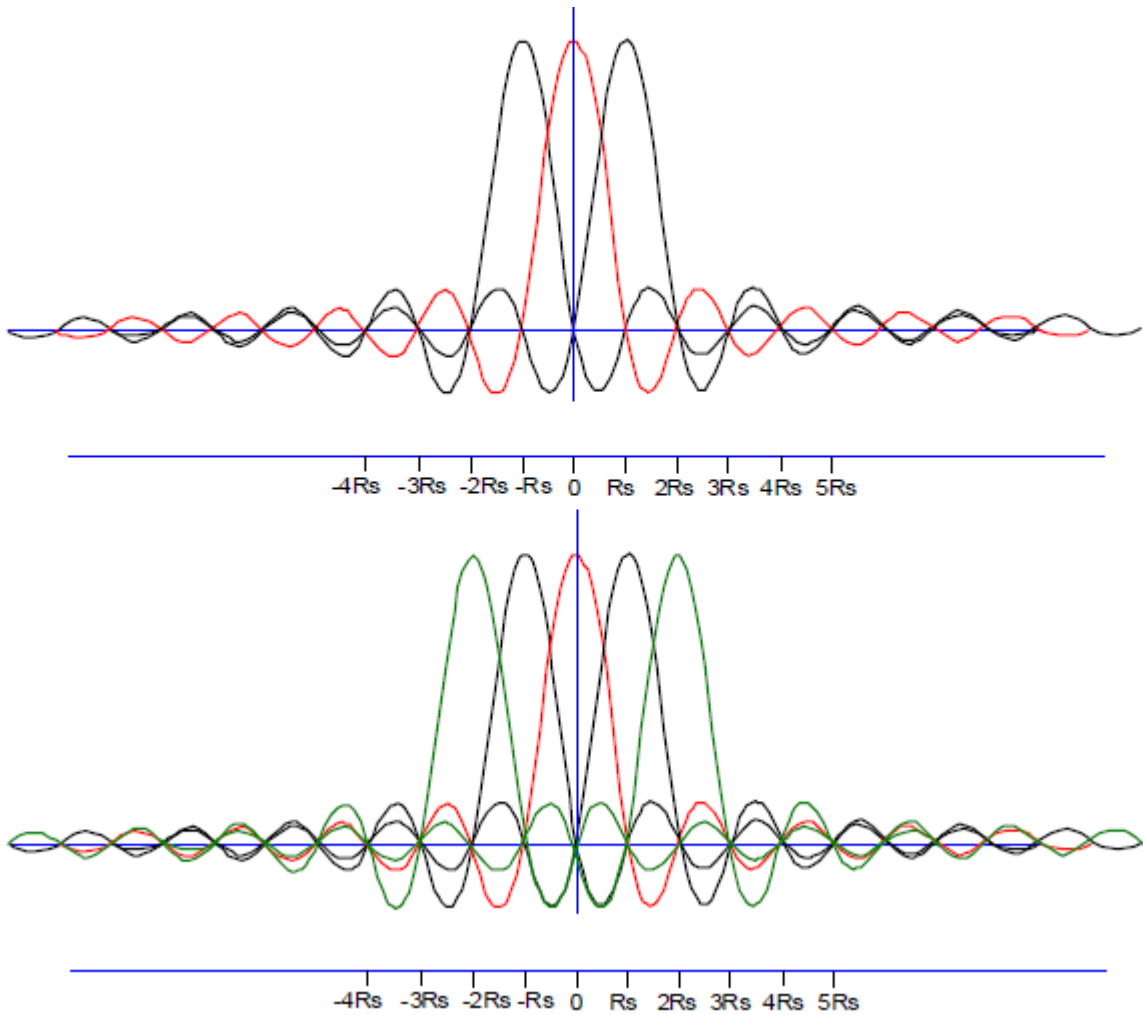
However, the addition of cyclic prefix which mitigates the effects of link fading and inter symbol interference, increases the bandwidth.



**Figure 1.29 Addition of Cyclic prefix to the OFDM signal**

## 1.5 Properties of OFDM

### 1.5.1 Spectrum and performance



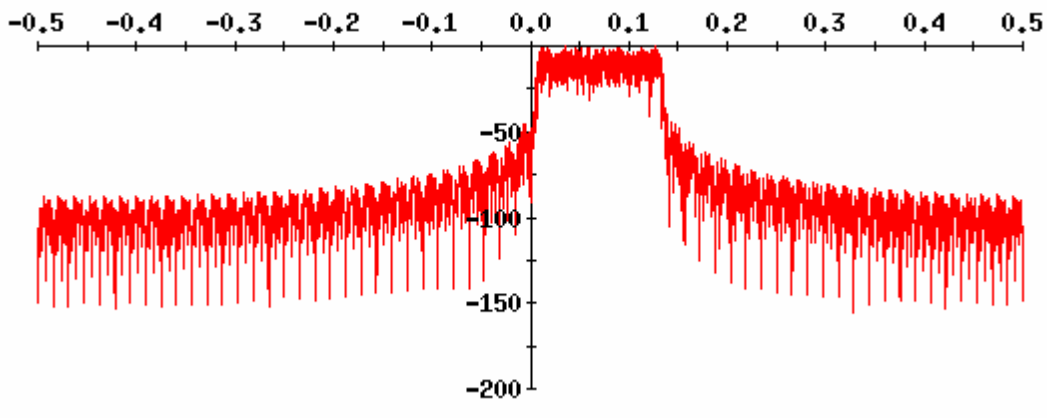
**Figure 1.30 Spectrum of an OFDM signal**

Unshaped QPSK signal produces a spectrum such that its bandwidth is equal to  $(1 + \alpha)R_s$ . In OFDM, the adjacent carriers can overlap in the manner shown here. The addition of two carriers, now allows transmitting  $3R_s$  over a bandwidth of  $-2R_s$  to  $2R_s$  or total of  $4R_s$ . This gives a bandwidth efficiency of  $4/3$  Hz per symbol for 3 carriers and  $6/5$  for 5 carriers.



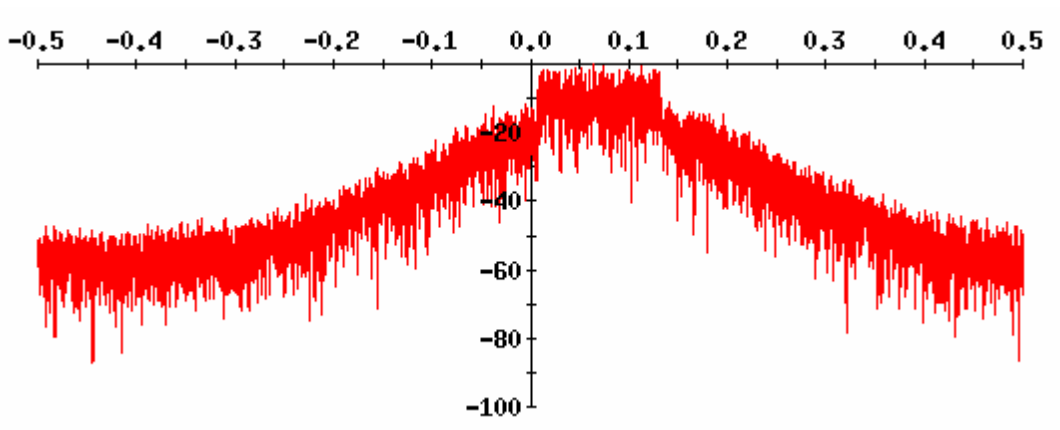
As more and more carriers are added, the bandwidth approaches,  
 $(N+1)/N$  bits per Hz.

So the larger the number of carriers, the better. Here is a spectrum of an OFDM signal.  
Note that the out of band signal is down by 50 dB without any pulse shaping.



**Figure 1.31** The spectrum of an OFDM signal with 1024 sub-carriers

Comparing this to the spectrum of a QPSK signal, we note that the sidebands are lower for OFDM and the variance is less too.



**Figure 1.32** The spectrum of a QPSK signal

### 1.5.2 Bit Error Rate performance

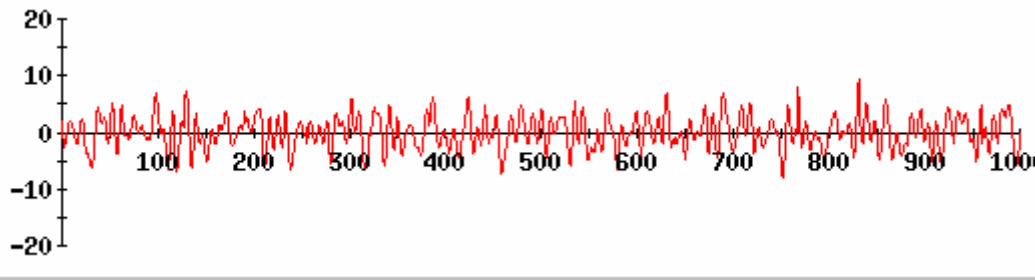
The BER of an OFDM is only exemplary in a fading environment. We would not use OFDM in a straight line of sight link such as a satellite link. OFDM signal due to its amplitude variation does not behave well in a non-linear channel such as created by high power amplifiers on board satellites. Using OFDM for a satellite would require a fairly large backoff, on the order of 3 dB, so there must be some other compelling reason for its use such as when the signal is to be used for a moving user.

### 1.5.3 Peak to average power ratio (PAPR)

If a signal is a sum of  $N$  signals each of max amplitude equal to 1 v, then it is conceivable that we could get a max amplitude of  $N$  that is all  $N$  signals add at a moment at their max points.

The PAPR is defined as

$$R = \frac{|x(t)|^2}{P_{avg}}$$



**Figure 1.33 OFDM signal**

For an OFDM signal that has 128 carriers, each with normalized power of 1 w, then the

max PAPR can be as large as  $\log(128)$  or 21 dB. This is at the instant when all 128 carriers combine at their maximum point, unlikely but possible. The RMS PAPR will be around half this number or 10-12 dB. This same PAPR is seen in CDMA signals as well.

The large amplitude variation increases in-band noise and increases the BER when the signal has to go through amplifier non-linearities. Large back off is required in such cases. This makes use of OFDM just as problematic as Multi-carrier FDM in high power amplifier applications such as satellite links. Several ideas are used to mitigate the large PAPR :

1. Clipping : We can just clip the signal at a desired power level. This reduces the PAPR but introduces other distortions and ICI.
  2. Selective Mapping : Multiply the data signal by a set of codes, do the IFFT on each and then pick the one with the least PAPR. This is essentially doing the process many times using a CDMA like code.
  3. Partial IFFT : Divide the signal in clusters, do IFFT on each and then combine these. So that if we subdivide 128 carrier in to a group of four 32 carriers, each, the max PAPR of each will be 12 dB instead of 21 for the full. Combine these four sequences to create the transmit signal.
- These are some of the things to keep the effect of the non-linearity manageable.

#### 1.5.4 Synchronization

The other problem is that tight synchronization is needed. Often pilot tones are served in the sub- carrier space. These are used to lock on phase and to equalize the channel.

#### 1.5.5 Coding

The sub-carriers are typically coded with Convolutional coding prior to going through

IFFT. The coded version of OFDM is called COFDM or Coded OFDM.

1.5.6 Parameters of real OFDM

The OFDM use has increased greatly in the last 10 years. It is now proposed for radio broadcasting such as in Eureka 147 standard and Digital Radio Mondiale (DRM). OFDM is used for modem/ADSL application where it coexists with phone line. For ADSL use, the channel, the phone line, is filtered to provide a high SNR. OFDM here is called Discrete Multi Tone (DMT.) (Remember the special filters on you phone line if you have cable modem.) OFDM is also in use in our wireless internet modem and this usage is called 802.11a. Let's take a look at some parameters of this application of OFDM. The summary of these are given below.

**Table 1.2 Parameters of real OFDM**

Data Rates	6 Mbps to 48 Mbps
Modulation	BPSK, QPSK, 16 QAM and 64 QAM
Coding	Convolutional concatenated with Reed Solomon
FFT Size	64 with 52 subcarriers used, 48 for data and 4 for pilots
Subcarrier Frequency Spacing	20 MHz divided by 64 carriers or 0.3125 MHz
FFT Period	Also called symbol period, $3.2 \mu s = 1/\Delta f$
Guard Duration	One quarter of symbol time, $0.8 \mu s$
Symbol Time	$4 \mu s$

## CHAPTER 2

### PAPR problem in OFDM and need for companding

PAPR is defined as the ratio between the maximum power and the average power for the envelope of a baseband complex signal, i.e,

$$PAPR(s(t)) = \frac{\max|s(t)|^2}{\text{avg}|s(t)|^2}$$

Linear power amplifiers are being used in the transmitter so that the Q-point is in the linear region. Due to the high PAPR, the Q-point moves to the saturation region hence, clipping the signal peaks that generates in-band and out-of-band distortion.

In order to keep the Q-point in the linear region, the dynamic range of the power amplifier should be increased which again reduces its efficiency and enhances the cost. Hence, a trade-off exists between nonlinearity and efficiency. And also with the increase of this dynamic range, the cost of power amplifier increases. Our objective is to reduce this PAPR.

#### 2.1 Companding

In telecommunication and signal processing, companding (occasionally called compansion) is a method of mitigating the detrimental effects of a channel with limited dynamic range. The name is a portmanteau of compressing and expanding. The use of companding allows signals with a large dynamic range to be transmitted over facilities that have a smaller dynamic range capability. Companding is employed in telephony and other audio applications such as professional wireless microphones and analog recording.

While the dynamic range compression used in audio recording and the like depends on

a variable-gain amplifier, and so is a locally linear process (linear for short regions, but not globally), companding is non-linear and takes place in the same way at all points in time. The dynamic range of a signal is compressed before transmission and is expanded to the original value at the receiver.

The electronic circuit that does this is called a compandor and works by compressing or expanding the dynamic range of an analog electronic signal such as sound recorded by a microphone. One variety is a triplet of amplifiers: a logarithmic amplifier, followed by a variable-gain linear amplifier and an exponential amplifier. Such a triplet has the property that its output voltage is proportional to the input voltage raised to an adjustable power. In practice, companders are designed to operate according to relatively simple dynamic range compressor functions that are designed to be suitable for implementation using simple analog electronic circuits. The two most popular compander functions used for telecommunications are the A-law and  $\mu$ -law functions.

## 2.2 A-Law

An A-law algorithm is a standard companding algorithm, used in European digital communications systems to optimize, i.e., modify the dynamic range of an analog signal for digitizing.

For a given input  $x$ , the equation for A-law encoding is given by,

$$\begin{aligned}
 F(x) &= \operatorname{sgn}(x) \frac{A|x|}{1+\ln(A)} && \text{for } |x| < \frac{1}{A} \\
 &= \operatorname{sgn}(x) \frac{1+\ln(A|x|)}{1+\ln(A)} && \text{for } 1/A \leq |x| \leq 1
 \end{aligned}$$

A-law expansion is given by the inverse function,

$$F^{-1}(y) = \operatorname{sgn}(y) \frac{|y|^{(1+\ln(A))}}{A} \quad \text{for } |y| < \frac{1}{1+\ln(A)}$$

$$= \operatorname{sgn}(y) \frac{\exp(|y|(1+\ln(A))-1)}{A} \quad \text{for } \frac{1}{1+\ln(A)} \leq |y| < 1$$

The reason for this encoding is that the wide dynamic range of speech does not lend itself well to efficient linear digital encoding. A-law encoding effectively reduces the dynamic range of the signal, thereby increasing the coding efficiency and resulting in a signal-to-distortion ratio that is superior to that obtained by linear encoding for a given number of bits.

The  $\mu$ -law algorithm provides a slightly larger dynamic range than the A-law at the cost of worse proportional distortion for small signals. By convention, A-law is used for an international connection if at least one country uses it.

In A-law companding method, the compressor characteristic is piecewise, made up of a linear segment for low level inputs and a logarithmic segment for high level inputs. Figure 2.1 shows the A-law compressor characteristics for different values of A.

Corresponding to A=1, we observe that the characteristic is linear (no compression) which corresponds to a uniform quantization. A-law has mid riser at the origin. Hence it contains non-zero value. The practically used value of "A" is 87.6. The A-law companding is used for PCM telephone systems.

The linear segment of the characteristic is for low level inputs whereas the logarithmic segment is for high level inputs.

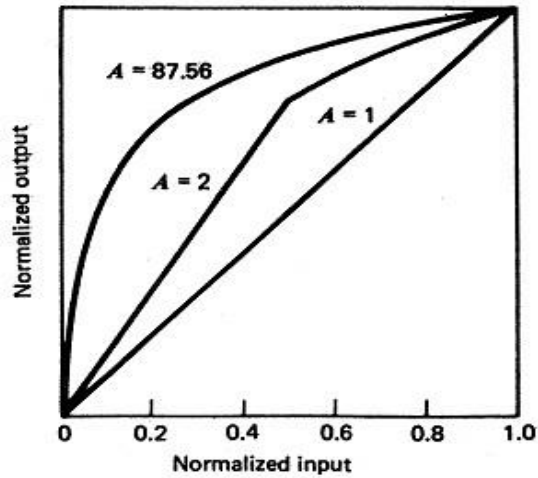


Figure 2.1 A-law compressor characteristics

### 2.3 $\mu$ -Law

The  $\mu$ -law algorithm (sometimes written "mu-law", or approximated as "u-law") is a companding algorithm, primarily used in the digital telecommunication systems of North America and Japan. Companding algorithms reduce the dynamic range of an audio signal. In analog systems, this can increase the signal-to-noise ratio (SNR) achieved during transmission, and in the digital domain, it can reduce the quantization error (hence increasing signal to quantization noise ratio). These SNR increases can be traded instead for reduced bandwidth for equivalent SNR.

It is similar to the A-law algorithm used in regions where digital telecommunication signals are carried on E-1 circuits, e.g. Europe.

For a given input  $x$ , the equation for  $\mu$ -law encoding is,

$$F(x) = \text{sgn}(x) \frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \quad \text{for } -1 \leq x \leq 1$$



where  $\mu = 255$  (8 bits) in the North American and Japanese standards. It is important to note that the range of this function is  $-1$  to  $1$ .

$\mu$ -law expansion is then given by the inverse equation:

$$F^{-1}(y) = \text{sgn}(y) \frac{1}{\mu} ((1 + \mu)^{|y|} - 1) \quad \text{for } -1 \leq y \leq 1$$

There are three ways of implementing a  $\mu$ -law algorithm:

### 1. Analog

Use an amplifier with non-linear gain to achieve companding entirely in the analog domain.

### 2. Non-linear ADC

Use an analog-to-digital converter with quantization levels which are unequally spaced to match the  $\mu$ -law algorithm.

### 3. Digital

Use the quantized digital version of the  $\mu$ -law algorithm to convert data once it is in the digital domain.

In the  $\mu$ -law companding, the compressor characteristic is piecewise, made up of a linear segment for low level inputs and a logarithmic segment for high level inputs. Figure 1.2 shows the  $\mu$ -law compressor characteristics for different values of  $\mu$ . Higher the value of " $\mu$ ", more is the compression. Corresponding to  $\mu=0$ , we observe that the characteristic is linear (no compression) which corresponds to a uniform quantization.  $\mu$ -law has mid tread at the origin. Hence, it contains zero value. The practically used value of " $\mu$ " is 255. The  $\mu$ -law companding is used for speech & music signals. This  $\mu$ -law companding technique is used in United States (U.S.), Canada, Japan, etc.

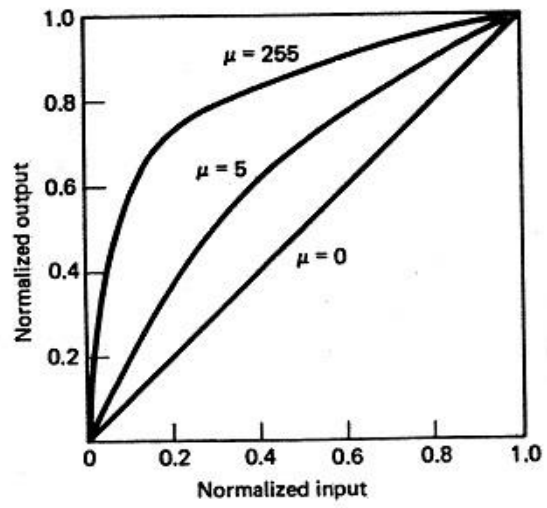
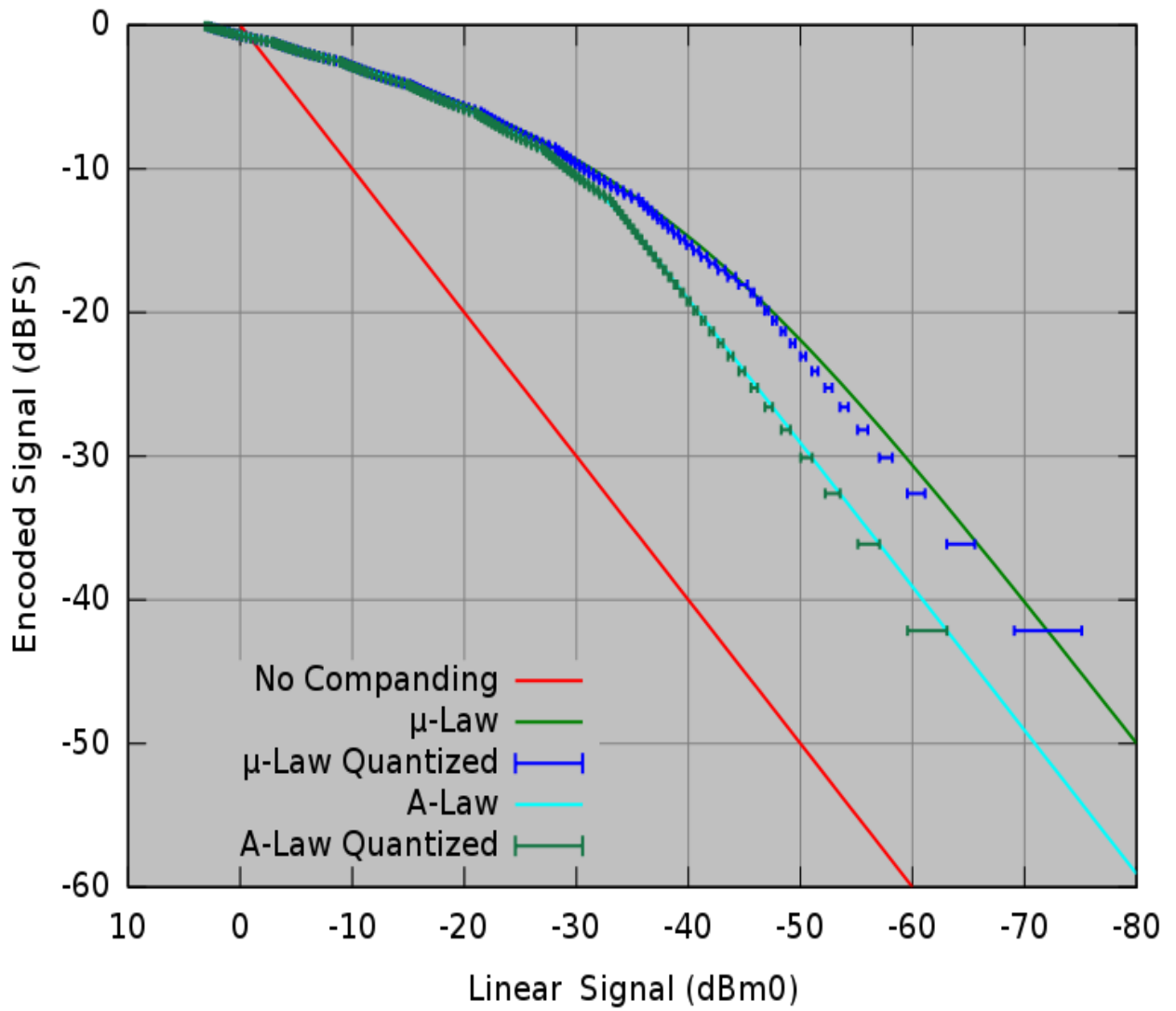


Figure 1.2  $\mu$ -law characteristics



**Figure 1.3 Comparison between A-law and  $\mu$ -law**

## CHAPTER 3

### Role of power amplifier in an OFDM system

The principal drawback of an OFDM system is the high peak-to-average power ratio (PAPR). This imposes strong requirements on the linearity of power amplifiers (PAs). Such linearity requirements translate into high back-off that results in low power efficiency. Real power amplifiers (PAs) have a nonlinear transfer function causing signal compression and clipping that result in signal waveform distortion and adjacent channel interference. Power back-off and PAPR reduction techniques reduce the nonlinear distortion effects but result in low power efficiency. Furthermore, broadband PAs introduce memory which gives rise to intersymbol interference (ISI).

From a performance point-of-view (considering both BER and power efficiency) it may be preferable to mitigate the nonlinear distortion at the transmitter, e.g., by applying a pre-distorter that includes PA memory effects. However, limited power budget and hardware complexity can motivate a receiver cancellation technique. In addition, receiver techniques can be used in combination with a pre-distorter. Lot of work has been done to study the trade-off between power and performance introduced by a power amplifier in an OFDM system.

High-efficiency RF power amplifiers (PAs) are critical in portable battery-operated wireless communication systems (such as cellular phones, personal digital assistants, and laptops) because they can dominate the power consumption. Power amplifiers demonstrate the highest efficiency when operated in the compression region (such as in Class-A, Class-AB and Class-B modes) or in the switching mode (such as Class D, E, F). However, these highly efficient nonlinear

PAs can only amplify constant envelope modulation signals (such as global system for mobile communications (GSM)) without nonlinear distortion. With modern wireless communication systems evolving to more spectrally efficient and higher data-rate modulation formats, highly linear PAs are required to avoid the out-of-channel interference (e.g., adjacent channel power ratio (ACPR)) and distortion (e.g., error vector magnitude (EVM)).

For example, the wireless local area network (WLAN) 802.11 g standard employs 64-QAM modulation and 52 orthogonal frequency division multiplexing (OFDM) carriers at a 54-Mb/s data rate. This modulation format has a high envelope peak-average ratio (PAR) of 8–10 dB.

The traditional approach to linearly amplify the non-constant envelope modulated signal is to back-off the linear Class-A or Class-AB power amplifier's output power until the distortion level is within acceptable limits. Unfortunately, this lowers efficiency significantly, especially for high PAR signals. Thus, there is an inherent tradeoff between linearity and efficiency in power amplifier design. This problem has been thoroughly investigated over many years and envelope elimination and restoration (EER), predistortion, feedback, feed-forward, envelope tracking, linear amplification with nonlinear control and gate dynamic biasing are just some of the techniques explored.

### **3.1 Non-linear effects of power amplifier on OFDM system**

Consider an input signal in polar coordinates as

$$x = \rho e^{j\phi}$$

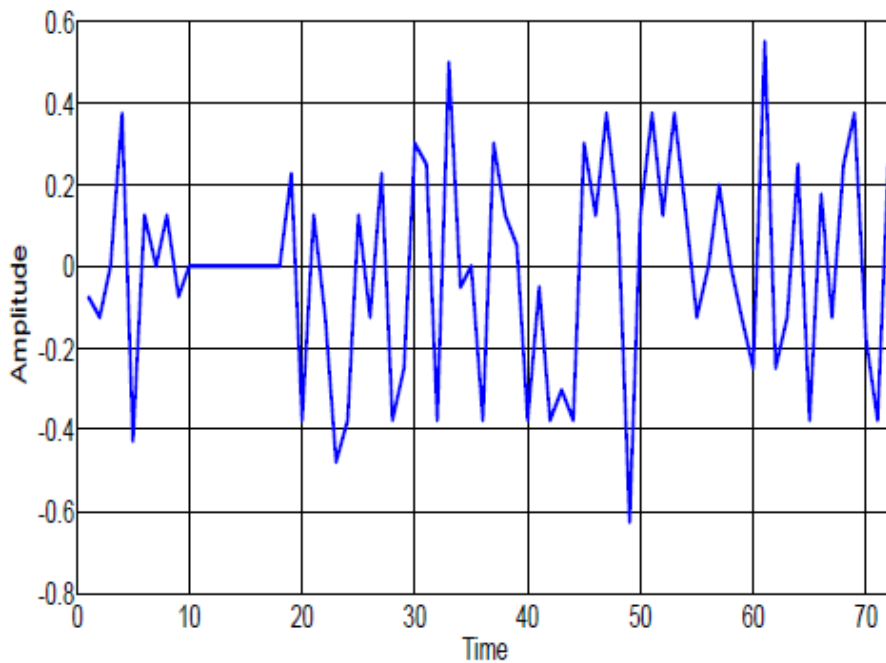
where,  $\rho$  is the amplitude and  $\phi$  is the phase of input signal.

The output of the power amplifier can be written as

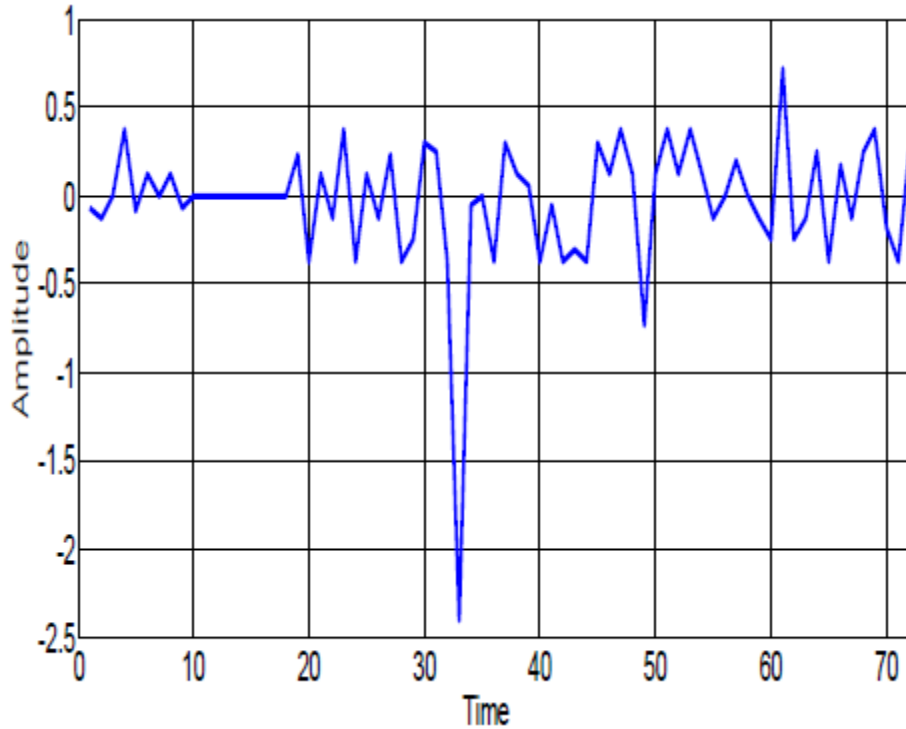
$$g(x) = M(\rho)e^{j(\phi+p(\rho))}$$

where,  $g(x)$  represents the output of the power amplifier,  $M(\rho)$  represents the AM/AM conversion and  $p(\rho)$  represents the AM/PM conversion characteristics of the power amplifier.

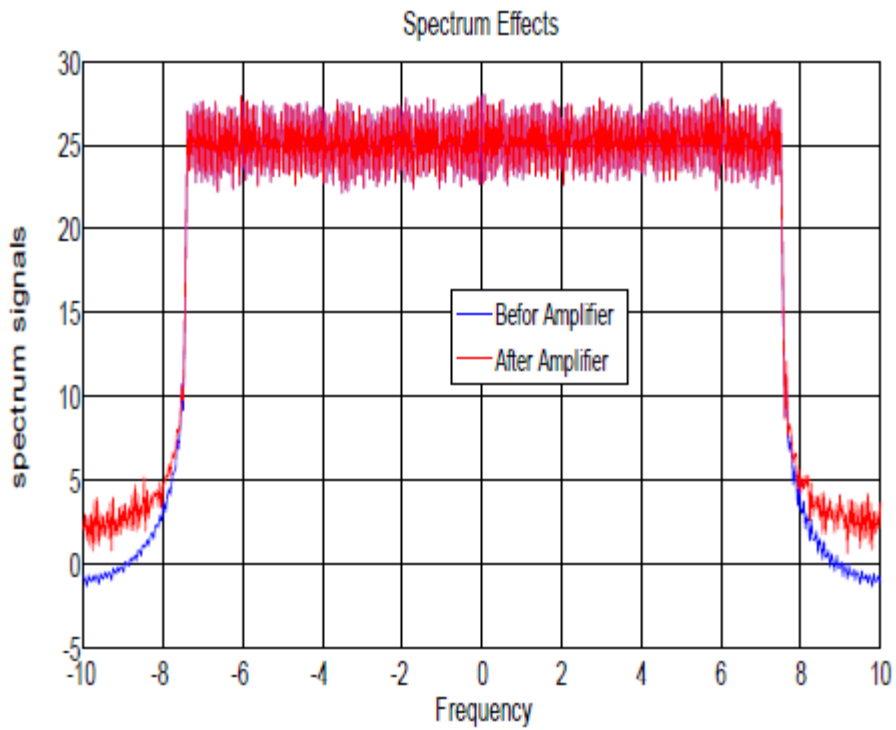
In this section, numerical simulation results are presented to show the nonlinear effects of a high power amplifier on an OFDM system. Power amplifier nonlinearity may have bad influence on OFDM signals mainly on two aspects: 1) Out-of-band distortion, which will cause the OFDM power spectrum distortion, i.e. the spectral spreading of the amplified signal and introduce adjacent channel interference (ACI). Requirements on adjacent channel interference for RF systems are very strict especially for large number of subscribers; therefore, it is of great importance to decrease the out-of-band distortion. 2) In-band distortion, which may disturb the OFDM constellations. PSD is utilized to evaluate the effects of AM/AM distortions on OFDM signals. Figure 3.1 shows the time domain waveform of a typical OFDM signal before passing through the high-power amplifier.



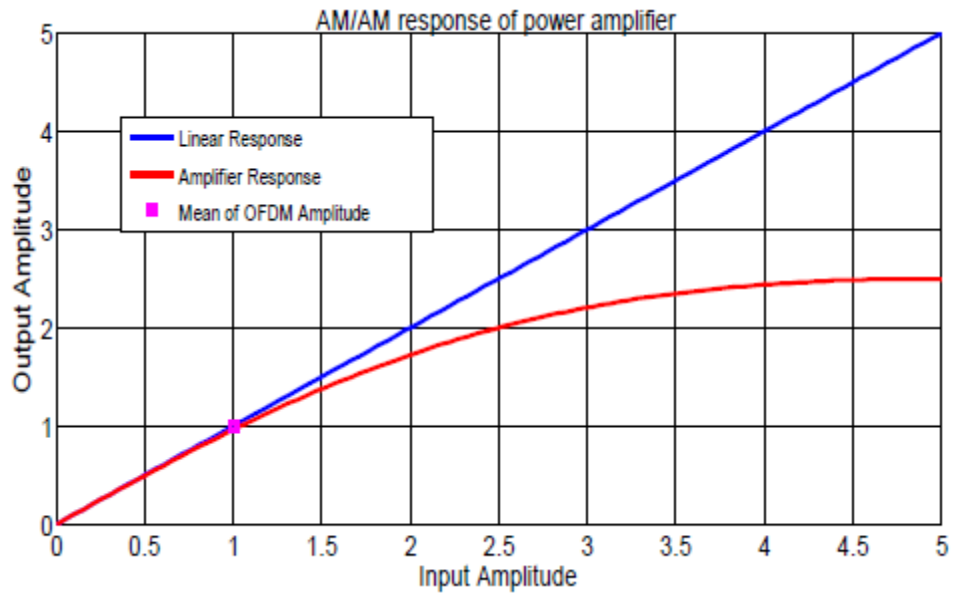
**Figure 3.1 OFDM signal before passing through high-power amplifier**



**Figure 3.2 OFDM signal after passing through high-power amplifier**



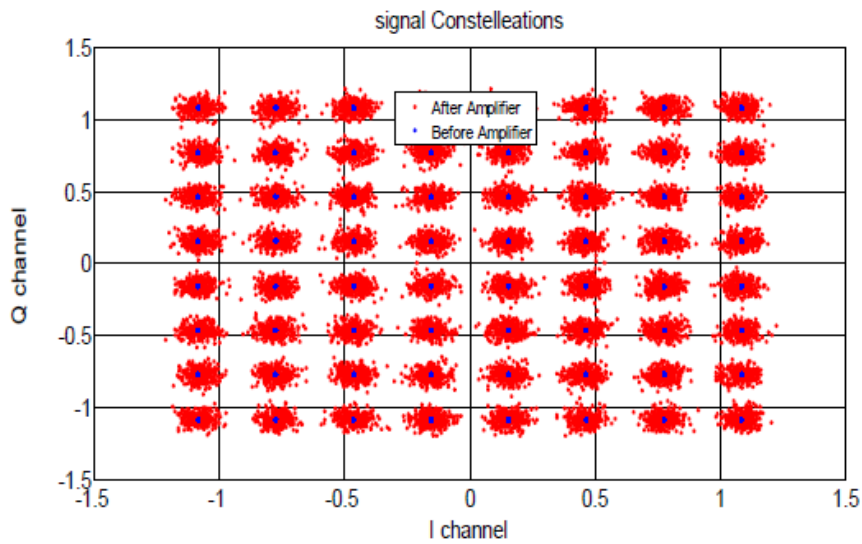
**Figure 3.3 Power spectrum of OFDM signal before and after amplifier shows spectral spreading**



**Figure 3.4 AM/AM response of OFDM signal**

Figure 3.4 shows the ideal and practical responses of the power amplifier. From input=1, non-linearity starts for the power amplifier.

From the figure 3.5 below, it is clear that the distortion causes not only dispersion but also rotation of the constellations.



**Figure 3.5 OFDM constellations**



Therefore, it is obvious from the graphs above that the performance of the OFDM system will be greatly degraded if the non-linear effects of the power amplifier are not minimized.

### **3.2 Process variations in a power amplifier**

Process variation is the naturally occurring variation in the attributes of transistors (length, widths, oxide thickness) when integrated circuits are fabricated. It becomes particularly important at smaller process nodes (<65 nm) as the variation becomes a larger percentage of the full length or width of the device and as feature sizes approach the fundamental dimensions such as the size of atoms and the wavelength of usable light for patterning lithography masks.

Process variation causes measurable and predictable variance in the output performance of all circuits but particularly analog circuits due to mismatch.<sup>[1]</sup> If the variance causes the measured or simulated performance of a particular output metric (bandwidth, gain, rise time, etc.) to fall below or rise above the specification for the particular circuit or device it reduces the overall yield for that set of devices.

Silicon mm-wave integrated circuits have enabled numerous new high frequency applications that previously were not economically feasible. However, high power mm-wave design in nanometer scale CMOS technologies is particularly susceptible to transistor variations, model accuracy, environmental changes, and aging. Reliable modeling is also a greater challenge in high frequency circuit design as the transistor parasitics contribute a larger fraction of the total relevant circuit capacitance. This is further exacerbated by the fact that the models are often primarily optimized for digital circuits. Finally, aging effects and environmental variations such as load mismatch and temperature can significantly degrade the performance of mm-wave power amplifiers (PAs), where both output power and efficiency are

strongly dependent on process and environmental parameters. Self healing can mitigate these issues by identifying any degradation and modifying the parameters of the circuit to improve its performance post fabrication with minimum overhead.

This thesis constitutes a self-healing power amplifier and therefore, a self-learning RF-frontend which adapts to channel conditions as well as process variability effects to minimize power consumption.

Standard circuit simulations are usually based on nominal values being assigned to the active and passive components. Still, the designer must be aware, that deviations in the process steps affecting all circuit elements simultaneously (process variations) as well as irregularities that impact only certain components (component mismatch) can occur during the chip production. Thus, the measured circuit performance can significantly differ from the simulation based on nominal values. The used design kit includes specifications for the variances of the different process steps that affect both active and passive devices.

The process deviations can occur with a certain probability for a specific chip production run. In order to assess the possible performance changes, Monte Carlo simulations can be applied. The approach is to generate a probability distribution by executing a large number of independent simulation runs. For each simulation run, a random number generator determines which of the process variations out of a set of given ones occur. The complete Monte Carlo simulation can then for example be used to state the probability, that a circuit performance figure remains inside a given range of values.

Therefore, using the Monte Carlo analysis, the designer can change the PA circuit components to minimize the variation for a specific performance metric. Trade-offs with other performance factors of the design are most likely.

## CHAPTER 4

### Need for simulated annealing

Since the power amplifier is susceptible to process variability and component mismatch, for the purpose of this thesis, we create multiple instances of the power amplifier using Monte Carlo simulations. Each instance mimicks a new process variation and gives elaborate tables with varying P1dB, Gain and dc power. The idea is to make a self-learning system where we do not need to characterize anything during the design phase. The system learns through process variations and adapts to give minimum power consumption.

Let us assume we know the nominal and optimum settings of tuning knobs for one instance of the power amplifier, say instance 1. Through our work, we prove that we can find out the optimum tuning knob settings for any other instance using the information from instance 1. Therefore, if  $V_{low}$  (instance1),  $V_{high}$  (instance2) and  $\mu$  (companding factor) for instance 1 is known, we devise a way to find out  $V_{low}$  (instance*i*),  $V_{high}$  (instance*i*) and  $\mu$  (companding factor) for instance *i*. So everytime there is a process change, there is a particular instance that corresponds to it and we have the optimum tuning knob configurations for that particular instance. This optimum tuning knob configuration gives us minimum power. Furthermore, the same concept can be extended to make the system adaptable to varying channel conditions too.

The above paragraph gives a gist of the work and necessitates the use of an optimization algorithm to find out the optimum tuning knob settings for an instance of the power amplifier if the same is known for another instance. For our work, we use an algorithm called Simulated

Annealing(SA). The following sections concentrate on explaining the concept of simulated annealing. The actual details of implementation of SA are given in Part II.

#### **4.1 What is simulated annealing**

The simulated annealing algorithm was originally inspired from the process of annealing in metal work. Annealing involves heating and cooling a material to alter its physical properties due to the changes in its internal structure. As the metal cools its new structure becomes fixed, consequently causing the metal to retain its newly obtained properties. In simulated annealing we keep a temperature variable to simulate this heating process. We initially set it high and then allow it to slowly 'cool' as the algorithm runs. While this temperature variable is high the algorithm will be allowed, with more frequency, to accept solutions that are worse than our current solution. This gives the algorithm the ability to jump out of any local optimums it finds itself in early on in execution. As the temperature is reduced so is the chance of accepting worse solutions, therefore allowing the algorithm to gradually focus in on a area of the search space in which hopefully, a close to optimum solution can be found. This gradual 'cooling' process is what makes the simulated annealing algorithm remarkably effective at finding a close to optimum solution when dealing with large problems which contain numerous local optimums. The nature of the traveling salesman problem makes it a perfect example.

#### **4.2 Advantages of simulated annealing**

The simulated annealing algorithm is excellent at avoiding the problem of getting stuck at local minima and is much better on average at finding an approximate global optimum.

Comparatively, a hill climber algorithm will simply accept neighbor solutions that are

better than the current solution. When the hill climber can't find any better neighbors, it stops.



**Figure 4.1 Explaining SA**

In the example above we start our hill climber off at the red arrow and it works its way up the hill until it reaches a point where it can't climb any higher without first descending. In this example we can clearly see that it's stuck in a local optimum. If this were a real world problem we wouldn't know how the search space looks so unfortunately we wouldn't be able to tell whether this solution is anywhere close to a global optimum. Simulated annealing works slightly differently than this and will occasionally accept worse solutions. This characteristic of simulated annealing helps it to jump out of any local optimums it might have otherwise got stuck in.

### **4.3 Acceptance function**

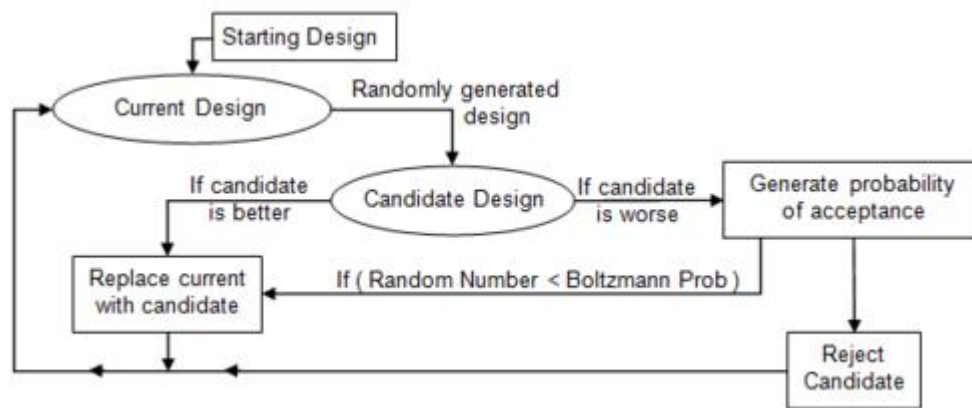
First we check if the neighbor solution is better than our current solution. If it is, we accept it unconditionally. If however, the neighbor solution isn't better we need to consider a

couple of factors. Firstly, how much worse the neighbor solution is; and secondly, how high the current 'temperature' of our system is. At high temperatures the system is more likely accept solutions that are worse.

$$probability = \exp\left(\frac{solution\ energy - neighbor\ energy}{Temperature}\right)$$

Basically, the smaller the change in energy (the quality of the solution), and the higher the temperature, the more likely it is for the algorithm to accept the solution.

#### 4.4 Algorithm overview



**Figure 4.2 Algorithm**

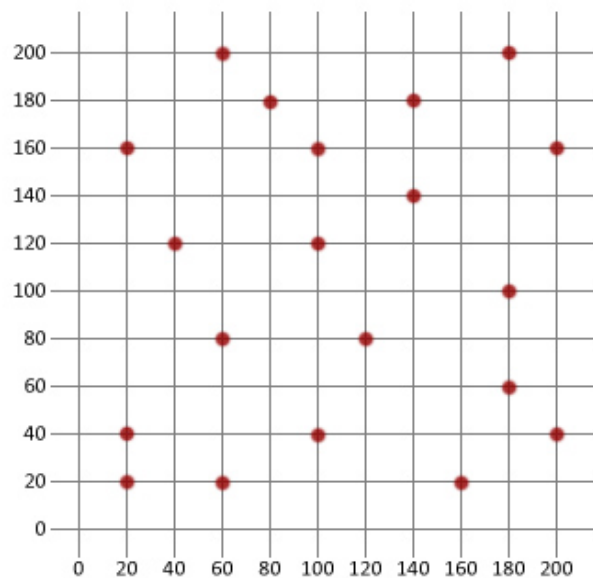
- 1) First we need set the initial temperature and create a random initial solution.
- 2) Then we begin looping until our stop condition is met. Usually either the system has sufficiently cooled, or a good-enough solution has been found.
- 3) From here we select a neighbor by making a small change to our current solution.
- 4) We then decide whether to move to that neighbor solution.
- 5) Finally, we decrease the temperature and continue looping.

#### 4.5 Temperature initialization

For better optimization, when initializing the temperature variable we should select a temperature that will initially allow for practically any move against the current solution. This gives the algorithm the ability to better explore the entire search space before cooling and settling in a more focused region.

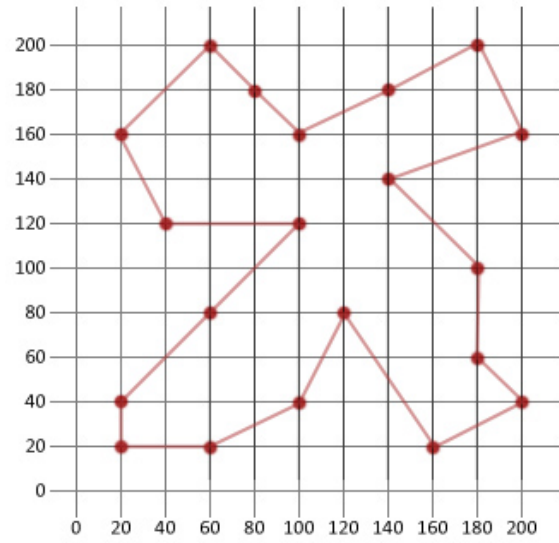
#### 4.6 Example

Let us consider the traveling salesman problem.



**Figure 4.3 TSP**

The solution to the traveling salesman problem is given by simulated annealing as shown -



**Figure 4.4 TSP solution**



## CHAPTER 5

### Analysis of OFDM packet structure for protocol level implications in online learning

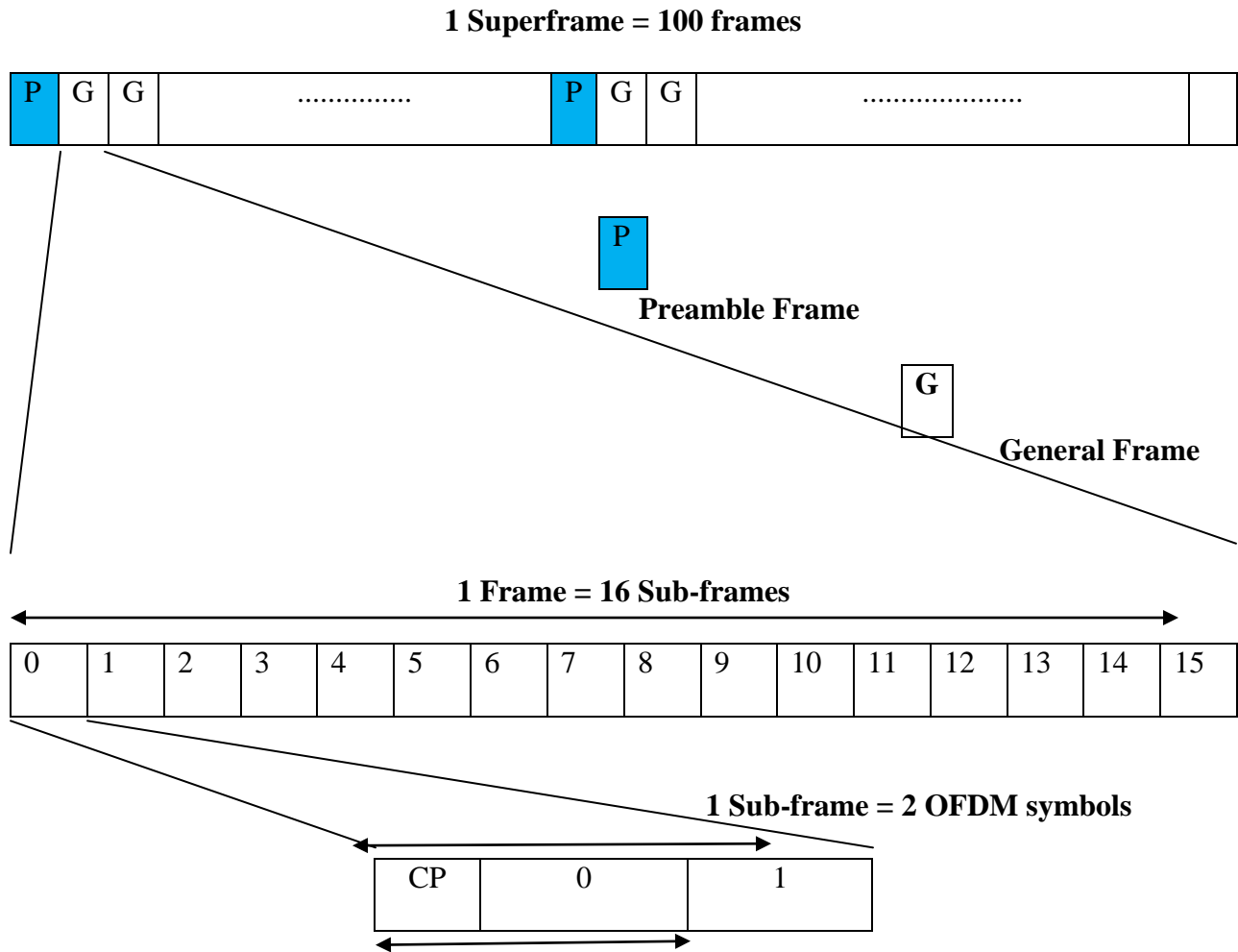
Orthogonal frequency division multiplexing is a method of encoding digital signals using a large number of parallel narrow-band subcarriers instead of a single wide-band carrier to transport information. This method has the advantage of dealing with multi-path easily and efficiently. Also, it is robust against narrow-band interference. Some of the drawbacks of the method include sensitivity to frequency offset and phase noise. A high peak-to-average power ratio (PAPR) reduces the power efficiency of the RF power amplifier at the transmitter. OFDM is used for various standards like DSL, 802.11a, DAB and DVB.

Let us study the feasibility of the thesis proposal before understanding the actual work. The entire proposal is based on the assumption that a system is feasible if EVM is within the threshold of 33%. The accuracy of EVM calculation improves with the frequency. An EVM calculation averaged over 10 symbols gives a better estimation compared to an EVM calculation that is done over just one symbol. According to our estimate, one in a 100 symbols can be used to perfectly calculate the EVM. But since a certain number of symbols are dedicated for this purpose, it may reduce the throughput. Our calculations below prove that using one in a 100 symbols for EVM calculations does not degrade the throughput.

#### 5.1 Frame structure

For the purpose of this proof, we will consider the OFDM numerology suggested in

below. This is purely a hypothesis. According to it, the downstream OFDM numerology is as illustrated in the diagram.



**Figure 5.1 One OFDM Symbol**

The diagram specifies a particular hierarchy that is -

1 super-frame = 100 frames ( $N_f$ )

1 frame = 16 sub-frames ( $N_{sf}$ )

1 sub-frame = 2 symbols (CP + 2 symbols) ( $N_{sym}$ )

=> 1 frame = 32 symbols

We will only pick up the hierarchy, not necessarily the values, so  $N_f$ ,  $N_{sf}$  and  $N_{sym}$  are defined.

Analogy -

$$1 \text{ frame} = N_{sf} \times N_{sym}$$

## 5.2 Bit-rate calculation

Now, since the paper assumes the use of WIMAX OFDM, the basic parameters used in this report are as follows -

**Table 5.1 OFDM parameters for bit-rate calculation**

Parameter	Fixed	Mobile			
Channel width MHz	3.5	1.25	5	10	20
FFT	256	128	512	1024	2048
Subcarriers spacing KHz	15.625	10.94			
Data Subcarriers	192	72	360	720	1440
Pilot Subcarriers	8	12	60	120	240
Null/Guard Subcarriers	56	44	92	184	368
OFDM symbol duration	72us	102.9us			
Cyclic prefix/Guard Time $T_g/T_b$		1/32, 1/6, 1/8, 1/4			
Guard Time assuming 1/8	8us	11.4us			
Useful Symbol Time	64us	91.4us			
OFDM Symbols in 5ms frame	69	48			

Calculation for bit-rate[2] -

$$\text{symbol duration} = 72 \text{ us}$$

data - carrying subcarriers = 192

pilot subcarriers = 8

bits/subchannel (assuming 64-QAM) = 6

data bits/OFDM symbol = 6 x 192 = 1152

pilot bits/OFDM symbol = 6 x 8 = 48

channel coding reduces data bits/symbol =  $3/4 \times 1152 = 3 \times 288 = 864$

channel coding reduces pilot bits/symbol =  $3/4 \times 48 = 36$

data bit rate for 802.16 =  $864 / (72 \times 10^{-6}) = 12$  Megabits/sec for 256 point **fixed** Wimax

pilot bit rate =  $36 / (72 \times 10^{-6}) = 0.5$  Megabits of pilot/sec for 256 point **fixed** Wimax

Similarly, for 512 point **mobile** Wimax,

symbol duration = 102.9 us

data bit rate = 15.74 Megabits/sec for 512 point **mobile** Wimax

pilot bit rate = 2.6 Megabits/sec for 512 point **mobile** Wimax

### 5.3 Coherence Time considerations

Coherent time is the time over which a propagating wave may be considered coherent. It is the time interval within which its phase is, on an average, predictable.

The proposed scheme must adhere to the coherent time requirements and here are some basic calculations to explain the same -

$T_{\text{coh}} = c / (v \times f_c)$ , where  $c$  is the speed of light =  $3 \times 10^8$  m/s

$v$  is the speed of MS (let us assume a highway speed of 100 mile/hr)

$f_c$  is the central frequency of the carrier

The frequency range for Wimax is 2-66 Ghz.

If the frequency is 2.5 Ghz, for a highway speed of 100 mile/hr, the  $T_{coh} > 3ms$  approx(actual = 2.6ms)

If the frequency is 60Ghz, for a highway speed of 100 mile/hr, the  $T_{coh} = 100us$  approx(actual = 111.8 us)

Assuming the channel width for Wimax (2-66 Ghz) is 5Mhz for a mobile Wimax system (at a highway speed of 100 miles/hr) that operates at a frequency of 50Ghz, the coherence time is

$$T_{coh} = (3 \times 10^8 \times 3600) / (100 \times 1609.34 \times 50 \times 10^9) = 134.22 \text{ us}$$

Therefore, for a mobile Wimax system operating at 50Ghz, if the channel width is 5Mhz and its a 512 point system, the coherence time = 134.22 us and the data bit rate and pilot bit rate (from part 2) are 15.74 Megabits/sec and 2.6 Megabits/sec, respectively.

But for our purpose of application, we assume a mobile Wimax system (3 m/s) operating between 2-2.5GHz. Repeating the calculations above,

$$T_{coh} = (3 \times 10^8) / (3 \times 2.4 \times 10^9) = 41.67 \text{ ms}$$

The number of data bits that can be transmitted over this coherence time for a 512 point system =  $(15.74 \times 10^6) \times (0.04167) = 655885 \text{ bits}$

The number of pilot bits that can be transmitted over the coherence time for a 512 point system =  $(2.6 \times 10^6) \times (0.04167) = 108342 \text{ bits}$

From part 1, data bits/symbol =  $360 \times 6 = 2160 \text{ bits}$

But the number of data bits that can be sent during the coherence time = 655885

Therefore the number of data symbols transmitted during the coherence time = 303

Similarly, pilot bits/symbol =  $60 \times 6 = 360 \text{ bits}$

But the number of pilot bits that can be sent during the coherence time = 108342

Therefore the number of pilot symbols transmitted during the coherence time = 300

## 5.4 Error Vector Magnitude Measurement

Error Vector Magnitude or EVM is a measure used to quantify the performance of a digital radio transmitter or receiver. For an ideal transmitter or receiver, the constellation points are precisely at their ideal locations, however due to carrier leakage, phase noise, etc the actual constellation points deviate from ideal locations.

As proposed by the work, 1 in every 100 symbols can be used to calculate EVM with minimal drop in throughput.

Summarizing all the calculations -

**Table 5.2 Summary**

	Bits/symbol	bits in $T_{coh}$ (41.67 ms)	Symbo ls in $T_{coh}$	Bits for 100 symbols
Data	2160	655885	303	216000
Pilot	360	108342	300	36000

According to the table, approximately 300 symbols can be transmitted during the coherence time.

Therefore, the number of symbols that can be used for EVM calculations is  $300/100 = 3$

Throughput without EVM calculation

$$= 300/0.04167 \text{ sym/sec} = 7199.42 \text{ sym/sec} = 15.55 \text{ Mbits/s}$$

Throughput with EVM calculation

$$= (300-3)/0.04167 \text{ sym/sec} = 7127.43 \text{ sym/sec} = 15.395 \text{ Mbits/s}$$

Reduction in throughput due to EVM calculation

$$= (15.55 - 15.395) / 15.55$$

$$= 0.155 / 15.55$$

$$= 0.99678 \%$$

$$= 1\% \text{ approx}$$

Therefore, **using 1 in every 100 symbols reduces the throughput by just 1%.**

## 5.5 Relation between parameters

For relation between all parameters, please see graphs from db\_paper.m

$T_{\text{coh}} \propto 1/\text{Velocity}$

But there is an upper limit to velocity. If the velocity increases too much,  $T_{\text{coh}}$  will reduce to the extent of not allowing even 100 symbols of transmission. And since we are calculating EVM every once in 100 symbols, let us assume that a 100 symbol transmission is a requirement.

To satisfy that,

$$\text{Symbol time} = 102.9 \text{ us}$$

$$\text{Time for 100 symbols} = 100 \times 102.9 \text{ us}$$

Therefore,

$$T_{\text{coh}} > 10.29 \text{ ms}$$

$$\text{Now, } T_{\text{coh}} = c/(v \times f_c)$$

$$\implies v = c/(T_{\text{coh}} \times f_c) = 300000000/(0.01029 \times 2.4 \times 1000000000) = 12.15 \text{ m/s}$$

Therefore, the upper limit on velocity is 12.15 m/s.

Lets see what happens after 12.15 m/s.

Assuming velocity = 15 m/s,

$$T_{\text{coh}} = 0.0083$$

Number of bits transmitted over  $T_{\text{coh}} = 15.74 \times 10^6 \times 0.0083 = 130642$  bits = 60 symbols

We need one in a hundred symbols for EVM calculation.

In this condition, throughput =  $59/60 = 98.33\%$  and reduction in throughput =  $1.67\%$ .

Moreover, our requirement of a 100 symbol transmission is not met. Therefore, for our assumptions, **velocity must be within 12.15 m/s.**



## CHAPTER 6 Implementation

### 6.1 Simulation of an OFDM system

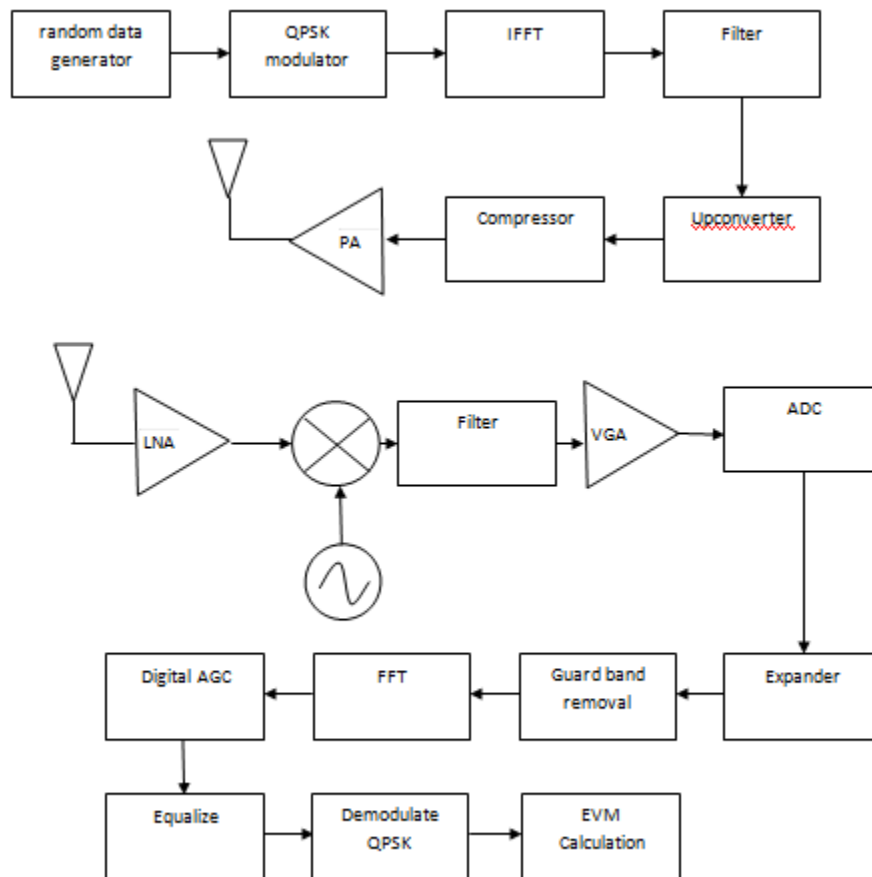


Figure 6.1 OFDM diagram

## 6.2 Code to simulate OFDM

Below are the details of the code-

### 6.2.1 Basic definitions

**Table 6.1 Basic definitions**

Tu (useful OFDM symbol period)	3.2e-6
T (baseband elementary period)	Tu/64
G (choice of 1/4, 1/8, 1/16, 1/32)	1/4
Delta (guard band duration)	G*Tu
Ts (total OFDM symbol period)	Tu+delta
Kmax (number of subcarriers)	64
Kmin (lowest carrier number)	0
N (number of FFT points)	128
q (carrier period to elementary period ratio)	120
fc (carrier frequency)	q*1/T
Rs (simulation period)	4*fc
gb (number of guard band points in base band)	round(G*N)
t (time vector)	0:1/Rs:Ts
Tt	0:T/2:Tu
Ts	1/Rs

### 6.2.2 Setting up useful objects

In this section, we set up objects for measuring Error Vector Magnitude(EVM), Adjacent Channel Power Ratio(ACPR) and modulator objects that can be used with "modulate" and "demodulate" for performing QPSK modulation and QPSK demodulation.

### 6.2.3 Random data generation

We use the function rand and round it off to obtain data for transmission

### 6.2.4 QPSK modulation

Modulates the data using "modulate" function.

### 6.2.5 IFFT

Performs IFFT.

### 6.2.6 Filter

It is a passband filter with a sampling frequency( $F_s$ ) of 320MHz, passband frequency( $F_{pass}$ ) of 20MHz and a stopband frequency( $F_{stop}$ ) of 25MHz. Further, the filter has a passband ripple( $A_{pass}$ ) of 1dB and stopband attenuation( $A_{stop}$ ) of 80dB. We construct a filter object that can be used to call the chebychev method.

### 6.2.7 Upconverter

In the upconverter, both the signal and noise are scaled down.

### 6.2.8 Compressor

We use a Mu-Law compressor with the value of  $\mu$  varying for different instances of power amplifier.

### 6.2.9 Power amplifier

The P1dB, IIP3 and Gain of the power amplifier are given in real-time depending upon the instance of the power amplifier used. For the first use, we randomly choose an instance and give the data accordingly.

### 6.2.10 Channel

The whole experiment is conducted for 5-8 types of channels. The variable PL that varies between 50-90 dB decides the quality and type of the channel. The AWGN channel has multipath effect, noise, interference and channel effect modeled into it.

### 6.2.11 LNA

The low-noise amplifier is the first component of the receiver front-end. The LNA gain is set as 15dB and IIP3 is -15dB. The noise figure for both LNA and PA is set as 1 for now.

### 6.2.12 Mixer

The mixer is initialized with a gain of 15dB and infinite IIP3. The noise figure of the mixer is

currently 0.

#### 6.2.13 Filter

It is similar to the one in the transmitter.

#### 6.2.14 VGA

It is a variable gain amplifier with a maximum gain of 70dB and a minimum gain of 70dB, infinite IIP3 and 0 noise figure.

#### 6.2.15 ADC

It is a 12-bit Sigma-delta ADC with uniform quantization and a  $V_{max}$  of 0.4V.

#### 6.2.16 Expander

It is a Mu-Law expander with a  $\mu$  varying with the power amplifier instances.

#### 6.2.17 Guardband removal

This section removes the guard-band for further equalization and demodulation.

#### 6.2.18 FFT AND Digital AGC

These sections perform FFT and automatic gain control for the VGA and the LNA, respectively.

#### 6.2.19 Equalization

Here, equalization of the signal takes place depending on the channel estimation details.

#### 6.2.20 Demodulate

Here, we use the object created with `modem.qamdemod`. The object is used to call the "demodulate" function. It is a QPSK demodulator.

#### 6.2.21 EVM Calculation

In this section, we use the object created in the beginning of the code for EVM calculation. The results are stored in an array for each combination of tuning knob settings.

### 6.3 Compinging

The Mu-Law compressor is placed between the upconverter block and the PA block. The Mu-Law expander is placed between the ADC and the guard-band removal block.

MATLAB has inbuilt functions for both Mu-Law compressor and Mu-Law expander -

`out = compand(in,Mu,v,'mu/compressor')` and `out = compand(in,Mu,v,'mu/expander')`

in - input vector

Mu - varies from 1 to 255

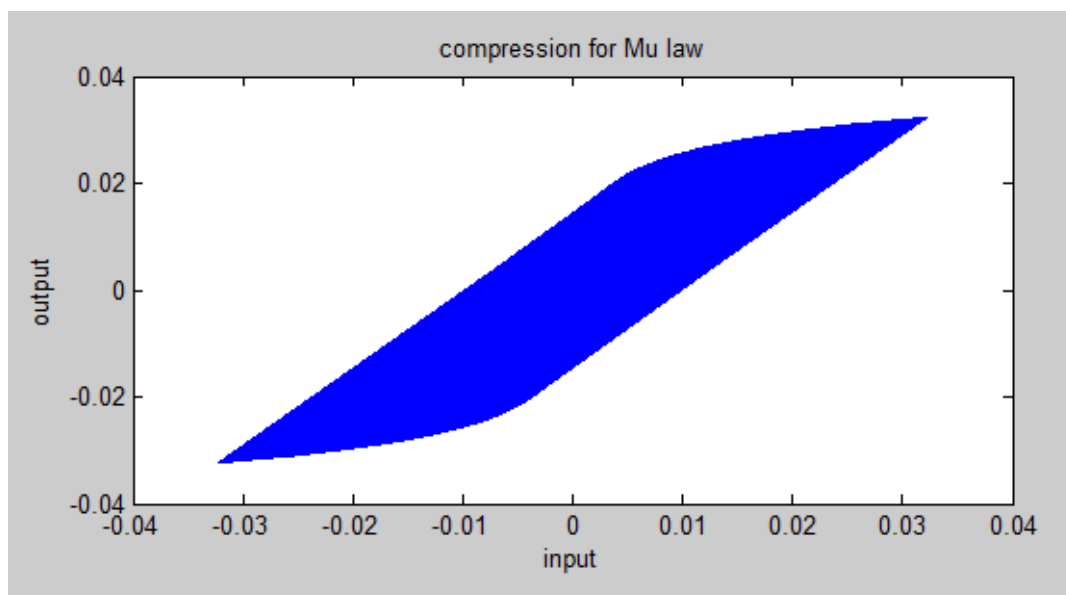
V - input signal's maximum magnitude

out - output vector, "out" has the same dimensions and magnitude as "in".

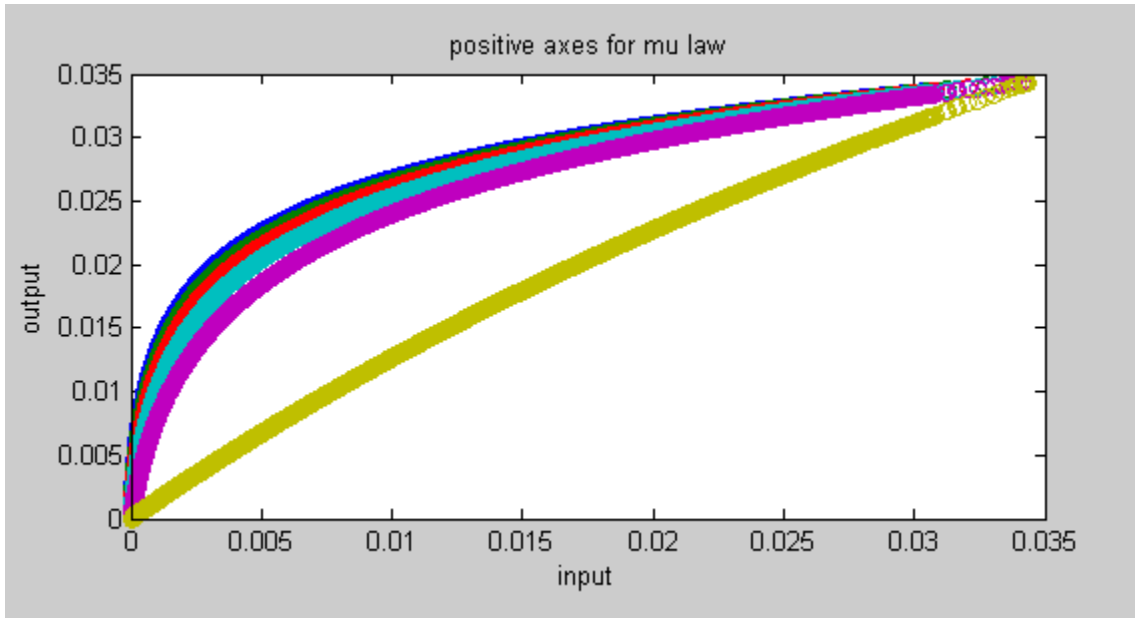
The following changes were noticed after adding a compressor and an expander -

1. The EVM increased, and to bring it back within 33%, the signal had to be toned down at various places.
2. The transmitted power reduced and also the PAPR.

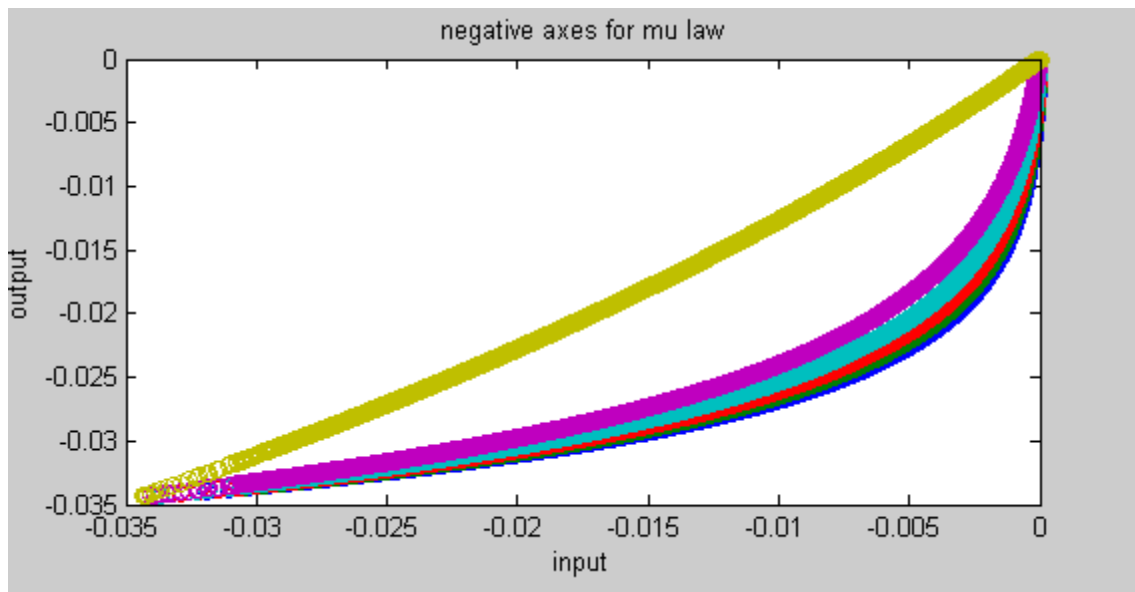
The following graphs explain some concepts -



**Figure 6.2 Compression using Mu-Law**



**Figure 6.3 Positive axis for Mu-Law**



**Figure 6.4 Negative axis for Mu-Law**

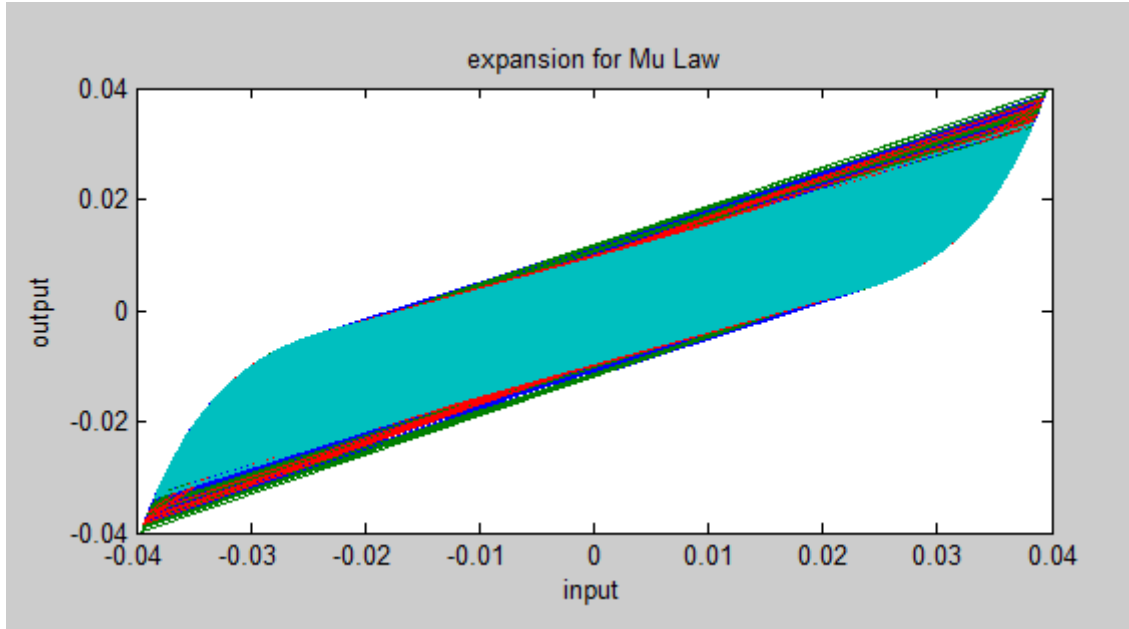
The first graph shows both the positive and negative y-axes merged for Mu-Law compression.

The next two graphs are plotted for values of Mu varying between 1 and 255.

Mu<sub>1</sub> = 255 Mu<sub>2</sub> = 200 Mu = 150 Mu = 100 Mu = 50 Mu = 1

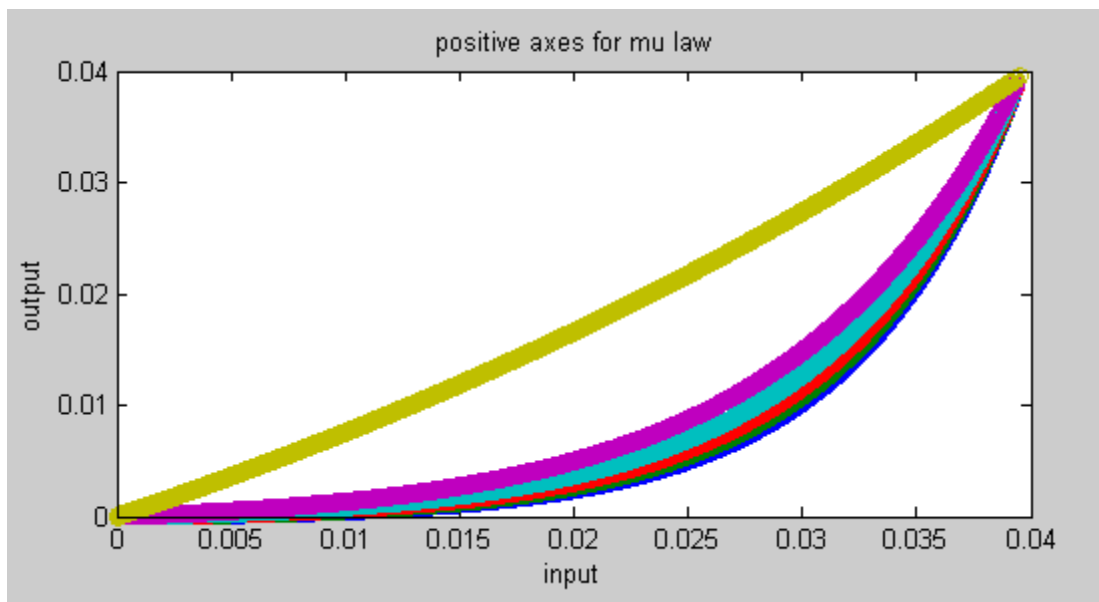
It is observed that as the value of Mu increases, from 1 to 255, the amount of compression

increases.

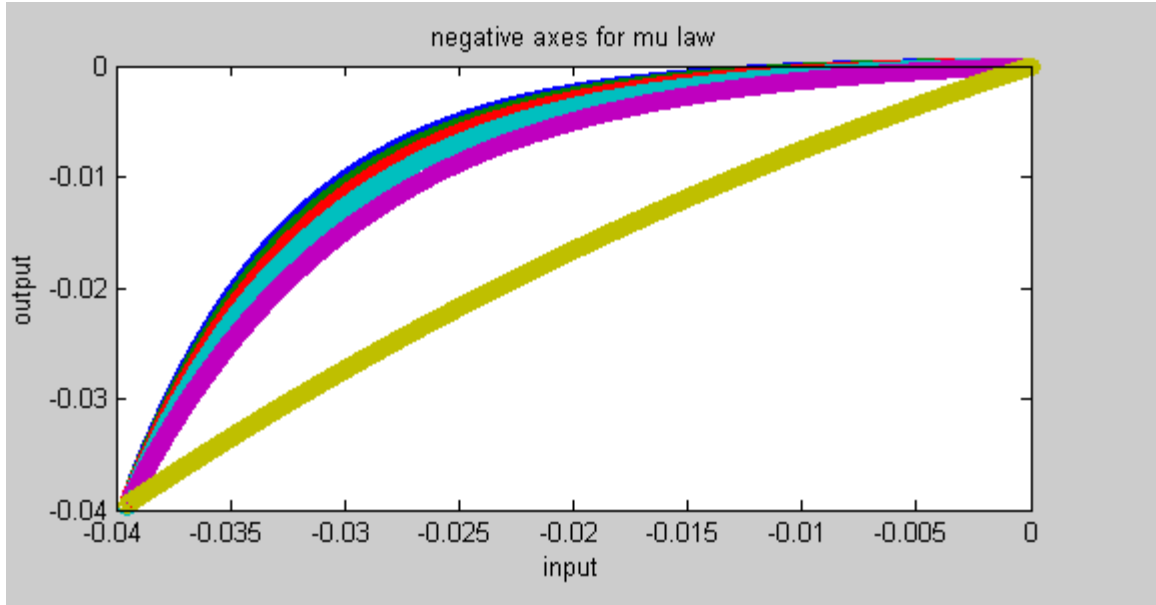


**Figure 6.5 Expansion for Mu-Law**

The graph above shows expansion for  $\mu=255$  with both the positive and the negative y axes merged.



**Figure 6.6 Positive axis for Mu-Law**



**Figure 6.7 Negative axis for Mu-Law**

The above two graphs show Mu-Law expansion with the positive y-axis and the negative y-axis separate. The different values of Mu used are

$$\text{Mu}_1 = 255 \quad \text{Mu}_2 = 200 \quad \text{Mu} = 150 \quad \text{Mu} = 100 \quad \text{Mu} = 50 \quad \text{Mu} = 1$$

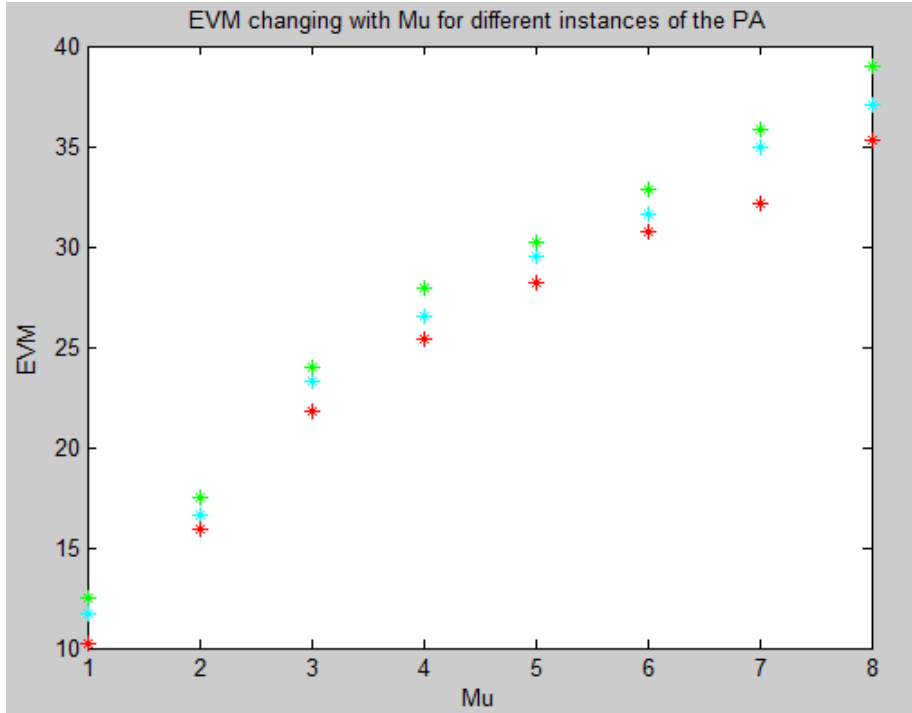
It is observed that as the value of Mu increases, from 1 to 255, the amount of expansion increases.

#### **6.4 Adding a power amplifier and analysis**

The power amplifier was characterized for 11 different instances. Each instance has 54 possible combinations of the tuning knobs Vlow and Vhigh. Vlow ranges from 0.7V to 1.2V (6 settings) and Vhigh ranges from 1.0V to 1.8V (9 settings). In all, every instance has 54 different values of P1dB, IIP3 (which is mostly P1dB+10), Gain and Pdc. The PA characterization gives a 6x9 matrix for each parameter - P1dB, Gain and Pdc.



### 6.4.1 Relation between Mu and EVM



**Figure 6.8 Mu vs EVM**

The graph above gives the relationship for EVM vs Mu for different three instances of the power amplifier for a constant channel condition of PL=50dB.

The various factors governing the EVM and Power are -

$\mu$  : 1 to 255

Vlow : 0.7 to 1.2 Volts

Vhigh : 1.0 to 1.8 Volts

PL : 50 to 90 dB

### 6.4.2 Optimal tuning knob settings

Optimal tuning knob settings are the combination of Vlow, Vhigh and Mu for which the power is minimum and the EVM<33%.

Now, let us pick an instance, say instance 11. For instance 11, let us try to find out Pdc,

Gain, P1dB and the optimum tuning knob settings.

For instance 11,

Pdc\_array =

0.2110	0.2400	0.2700	0.3010	0.3340	0.3690	0.4050	0.4420	0.4820
0.3290	0.3710	0.4150	0.4600	0.5070	0.5550	0.6050	0.6570	0.7110
0.4540	0.5100	0.5680	0.6280	0.6890	0.7520	0.8160	0.8820	0.9510
0.5840	0.6550	0.7270	0.8010	0.8770	0.9550	1.0340	1.1150	1.1980
0.7180	0.8040	0.8910	0.9790	1.0700	1.1620	1.2560	1.3530	1.4510
0.8540	0.9540	1.0560	1.1600	1.2660	1.3730	1.4820	1.5930	1.7070

Gain\_array =

17.8400	17.9650	18.0730	18.1680	18.2510	18.3240	18.3840	18.4300	18.4580
18.1100	18.2440	18.3530	18.4440	18.5210	18.5840	18.6330	18.6640	18.6730
18.1660	18.3290	18.4550	18.5560	18.6390	18.7050	18.7540	18.7830	18.7850
18.1130	18.3150	18.4670	18.5850	18.6790	18.7520	18.8060	18.8370	18.8400
17.9730	18.2230	18.4070	18.5480	18.6570	18.7410	18.8030	18.8400	18.8460
17.7560	18.0660	18.2890	18.4570	18.5850	18.6840	18.7570	18.8030	18.8160

P1dB\_array =

1.5000	2.2000	2.9000	3.4000	3.9000	4.4000	4.8000	5.2000	5.6000
1.7000	2.6000	3.4000	4.1000	4.7000	5.3000	5.8000	6.3000	6.8000
1.2000	2.2000	3.1000	3.8000	4.5000	5.1000	5.6000	6.0000	6.5000
0.6000	1.7000	2.6000	3.4000	4.0000	4.6000	5.0000	5.3000	5.6000
0.2000	1.2000	2.2000	3.0000	3.6000	4.2000	4.5000	4.8000	4.9000
-0.3000	0.8000	1.8000	2.6000	3.3000	3.8000	4.2000	4.4000	4.5000

Using a search algorithm, we found out what values of Vlow, Vhigh and Mu give minimum power and an EVM within 33%.

For instance 11, the optimum tuning knob settings for different channels are given as follows -

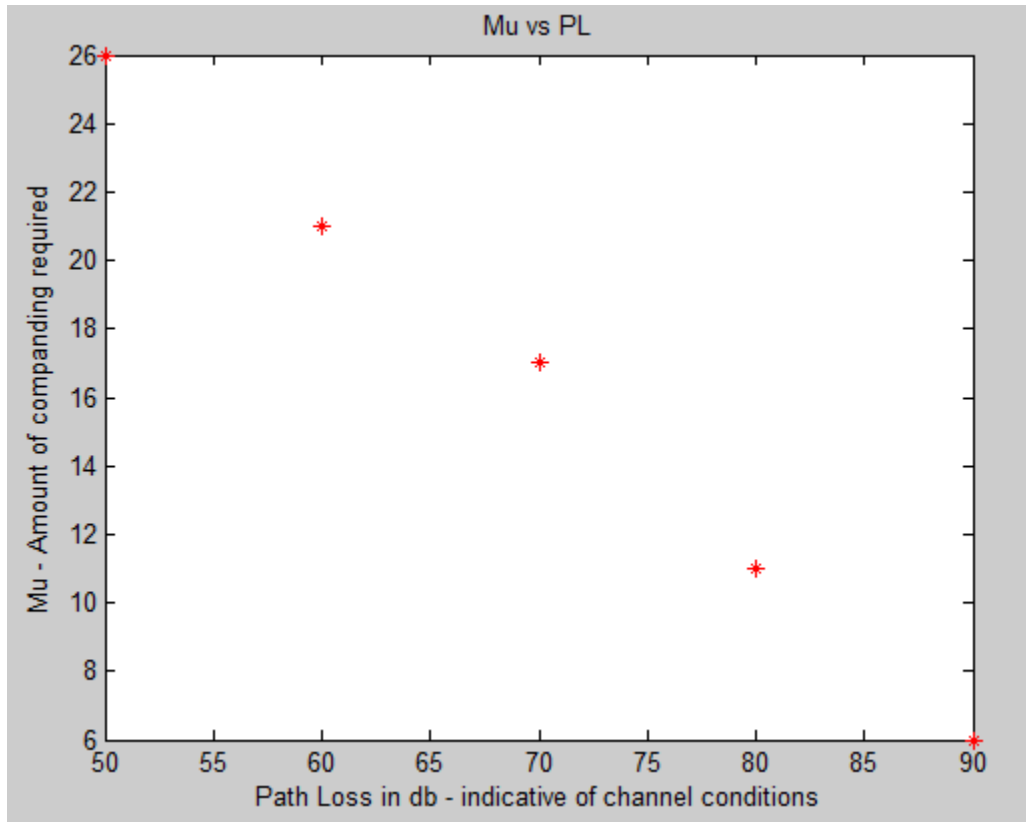
PL = 50 Vlow=0.8 Vhigh=1.0 Mu= 26 Power= 0.3290

PL = 60 Vlow=0.8 Vhigh=1.6 Mu= 21 Power= 0.4050

PL = 70 Vlow=0.9 Vhigh=1.3 Mu= 17 Power= 0.6280

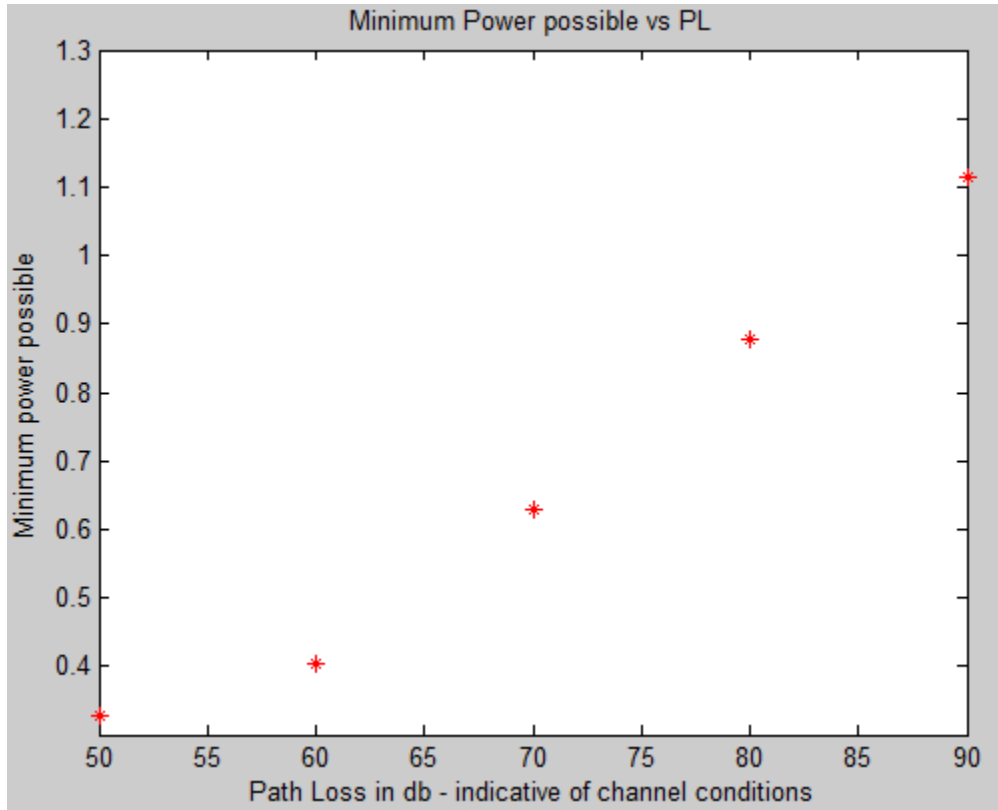
PL = 80 Vlow=1.0 Vhigh=1.4 Mu= 11 Power= 0.8770

PL = 90 Vlow=1.0 Vhigh=1.7 Mu=6 Power= 1.1150



**Figure 6.9 Mu vs PL**

The graph above proves that as the channel gets worse, the need for companding reduces, i.e we get minimum power within  $EVM < 33\%$  for lesser values of Mu.



**Figure 6.10 Minimum power vs PL**

The graph above shows that as the PL increases, i.e the channel gets worse, the minimum power possible increases.

Now let us find out the optimum tuning knob settings for instance 2.

For instance 2,

Pdc\_array =

0.2580	0.2930	0.3280	0.3660	0.4050	0.4450	0.4870	0.5310	0.5770
0.3870	0.4360	0.4860	0.5380	0.5920	0.6480	0.7050	0.7640	0.8260
0.5230	0.5870	0.6520	0.7200	0.7890	0.8600	0.9330	1.0080	1.0850
0.6630	0.7430	0.8240	0.9070	0.9920	1.0780	1.1670	1.2580	1.3510
0.8070	0.9020	0.9990	1.0980	1.1980	1.3010	1.4050	1.5120	1.6210
0.9510	1.0630	1.1760	1.2900	1.4070	1.5250	1.6460	1.7690	1.8940

Gain\_array =

18.0950	18.2160	18.3180	18.4070	18.4840	18.5510	18.6040	18.6420	18.6590
18.3150	18.4490	18.5570	18.6450	18.7190	18.7790	18.8240	18.8490	18.8510

18.3370	18.5010	18.6270	18.7260	18.8070	18.8690	18.9150	18.9380	18.9340
18.2380	18.4440	18.5970	18.7140	18.8060	18.8770	18.9280	18.9550	18.9510
18.0470	18.3050	18.4920	18.6330	18.7420	18.8250	18.8850	18.9190	18.9210
17.7760	18.0990	18.3290	18.5000	18.6300	18.7280	18.8000	18.8450	18.8550

P1dB\_array =

1.7000	2.5000	3.3000	3.9000	4.4000	5.0000	5.4000	5.9000	6.3000
1.5000	2.4000	3.3000	4.0000	4.6000	5.2000	5.7000	6.2000	6.7000
0.9000	1.9000	2.8000	3.6000	4.2000	4.8000	5.2000	5.6000	6.0000
0.4000	1.4000	2.4000	3.2000	3.8000	4.3000	4.7000	5.0000	5.2000
-0.1000	1.0000	1.9000	2.8000	3.4000	4.0000	4.3000	4.5000	4.6000
-0.5000	0.6000	1.6000	2.4000	3.1000	3.7000	4.0000	4.2000	4.2000

For instance 2, the optimum tuning knob settings for different channels are given as follows(found using a search algorithm) -

PL = 50 Vlow=0.7 Vhigh=1.0 Mu= 29 Power= 0.2580

PL = 60 Vlow=0.7 Vhigh=1.3 Mu= 24 Power= 0.3660

PL = 70 Vlow=0.7 Vhigh=1.7 Mu= 16 Power= 0.5310

PL = 80 Vlow=1.0 Vhigh=1.4 Mu= 8 Power= 0.9920

PL = 90 Vlow=1.1 Vhigh=1.5 Mu=5 Power= 1.3010

It can be clearly seen that if we try to use the optimum settings of one instance and one channel condition and try to apply it to another instance in the same channel condition, it may not necessarily give us minimum power. This proves that every instance has its own optimum tuning knob settings for different channel conditions which may or may not match another instance. We now need to figure out a way to use the optimum settings of one instance to find the optimum tuning knob configuration of another instance. That is how we will be able to simulate process variations and the system's ability to adapt to it and to the varying channel conditions.

This necessitated the use of some heuristics or some optimization algorithm. For the purpose of this work, we will use an algorithm called Simulated Annealing.

Let us review the flow of Simulated Annealing -

1. Select the initial configuration  $x$ .
2. Set the number of Monte Carlo steps,  $n_{MCS} = 0$ .
3. Set the initial temperature to some high value  $T_0$ .
4. Choose a transition  $\Delta x$  at random.
5. Calculate the function value before the transition,  $f_b = f(x)$ .
6. Do the trial transition as  $x = x + \Delta x$ .
7. Calculate the function value after the transition,  $f_a = f(x)$ .
8. Calculate  $\Delta f = f_a - f_b$ .

If  $\Delta f \leq 0$  accept the state

If  $\Delta f > 0$

Generate a random number  $u$  between 0 and 1

Accept the state only if  $u < e^{\frac{-\Delta f}{T}}$

9. If the state is rejected, return to the previous state,  $x = x - \Delta x$ .
10. Reduce the temperature by some small value,  $T = T - \rho_T$ .
11. Set  $n_{MCS} = n_{MCS} + 1$ .
12. If  $T > 0$ , return to step 1.

## 6.5 Simulated annealing for our application

For the purpose of our application, we divide the code into a few sections -

### 1. Temperature setting -

In SA, the perturbation for getting neighbor reduces with temperature. The time it takes to converge to the most optimum solution is directly proportional to the rate of cooling. This means, if the rate of cooling is slower, the solution takes more time to converge. But a slower rate of cooling ensures that the search space is wide enough for a thorough exploration. The best way to get a rate is to use an exponential decay. For the purpose of our work,

$$T = T_{\text{prev}} \times 0.95$$

### 2. Selecting a neighbor -

This function slightly perturbs the current configuration by a random amount that depends on the temperature of the system. As mentioned earlier, a higher temperature allows for a higher perturbation and a lower temperature allows for a lower perturbation. In our code, T is the current temperature and  $T_0$  is the temperature of the system at the beginning of the process. The function rand() is used to generate a random n-tuple between -1 and 1. In our work, the neighbor function is meant to implement in such a way that the n tuning knobs are perturbed in discrete steps, not all at once.

### 3. Energy calculation function -

The aim of the algorithm is to find the optimum configuration for a system, given a particular channel condition so that the system is maintained at a maximum allowable EVM(<33%) and gives minimum power consumption. The initial tuning knob setting( $s_0$ ) is obtained through offline characterization for a nominal device. The process variations decide how well or how worse the device performs compared to the nominal device. The EVM for a particular channel is therefore,

$$EVM_0 = f(s_0, \text{process})$$

Our energy function is given by

$$F = \text{abs}(\text{EVM}_{\text{th}} - \text{EVM})$$

for the condition that our device performs worse than the nominal.

However, if our device performs better than the nominal, i.e  $\text{EVM}_0 < \text{EVM}_{\text{th}}$ ,

$$F = \text{Power}$$

#### 4. Accept the new state

This function helps the algorithm decide whether or not to accept the new perturbed state depending on the current temperature.

For  $\text{EVM}_0 > \text{EVM}_{\text{th}}$

the logic for the accept function would comprise of

```
if(Fnew < F)
    accept = 1;
else
    change_F = Fnew - F;
    probability = exp(-(change_F)/T);
    if rand() < probability
        accept = 1;
    else
        accept = 0;
return accept
```

If  $\text{EVM}_0 < \text{EVM}_{\text{th}}$ , then the accept function looks like

```
if(EVM > EVMth)
    accept = 0;
else
    change_F = Fnew - F;
    probability = exp(-(change_F)/T);
    if rand() < probability
        accept = 1;
    else
        accept = 0;
return accept
```



## 6.6 Results

In this work, a set of 5 channels ranging from bad to good is simulated and the proposed idea is demonstrated on these channels. The EVM threshold for QPSK is set as 33%.

During the design and characterization phase of the transceiver, we find a nominal setting. The optimum tuning knob setting for minimum power for each channel can be obtained using Simulated Annealing as explained in the previous section. In SA, a slower cooling schedule tends to give a more accurate and optimized value. So we shift from  $T = T_{\text{prev}} * 0.95$  to  $T = T_{\text{prev}} * 0.98$ .

Let us get back to the optimum settings of instance 11. They are given as follows -

PL = 50 Vlow=0.8 Vhigh=1.0 Mu= 26 Power= 0.3290

PL = 60 Vlow=0.8 Vhigh=1.6 Mu= 21 Power= 0.4050

PL = 70 Vlow=0.9 Vhigh=1.3 Mu= 17 Power= 0.6280

PL = 80 Vlow=1.0 Vhigh=1.4 Mu= 11 Power= 0.8770

PL = 90 Vlow=1.0 Vhigh=1.7 Mu=6 Power= 1.1150

If we consider this data as the nominal data and use it to find the optimum tuning knob configurations for instance 2, we get

PL = 50 Vlow=0.7 Vhigh=1.0 Mu= 29 Power= 0.2580

PL = 60 Vlow=0.7 Vhigh=1.3 Mu= 24 Power= 0.3660

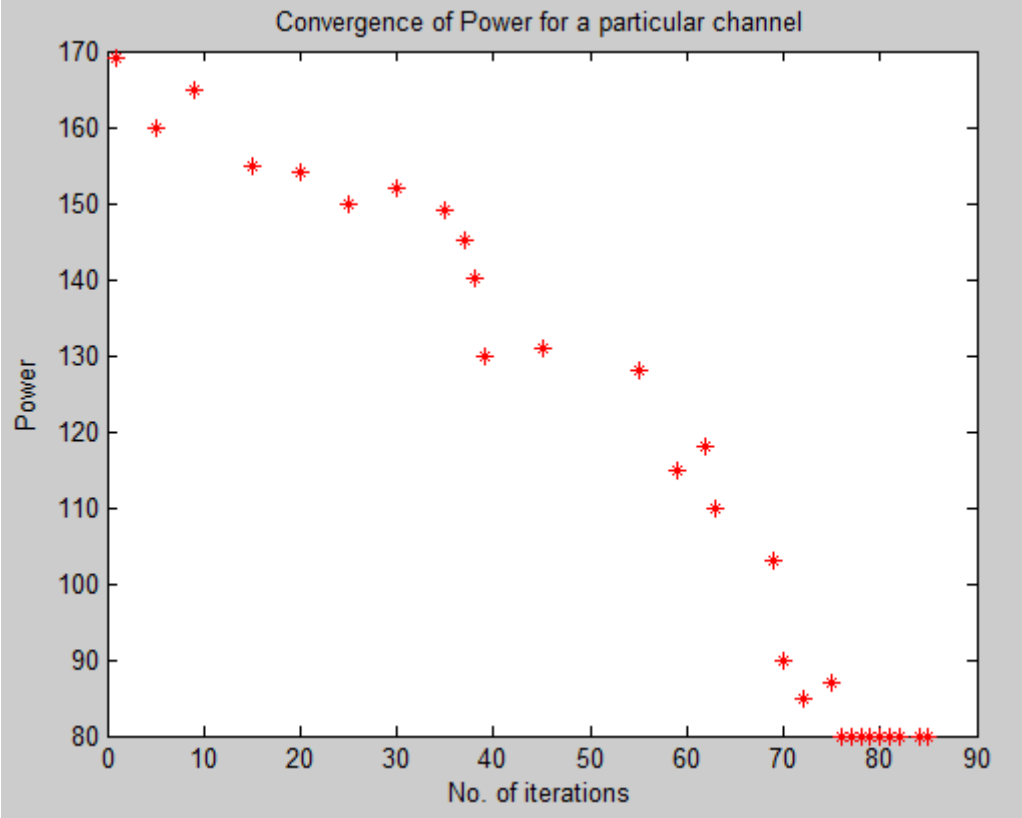
PL = 70 Vlow=0.7 Vhigh=1.7 Mu= 16 Power= 0.5310

PL = 80 Vlow=1.0 Vhigh=1.4 Mu= 8 Power= 0.9920

PL = 90 Vlow=1.1 Vhigh=1.5 Mu=5 Power= 1.3010

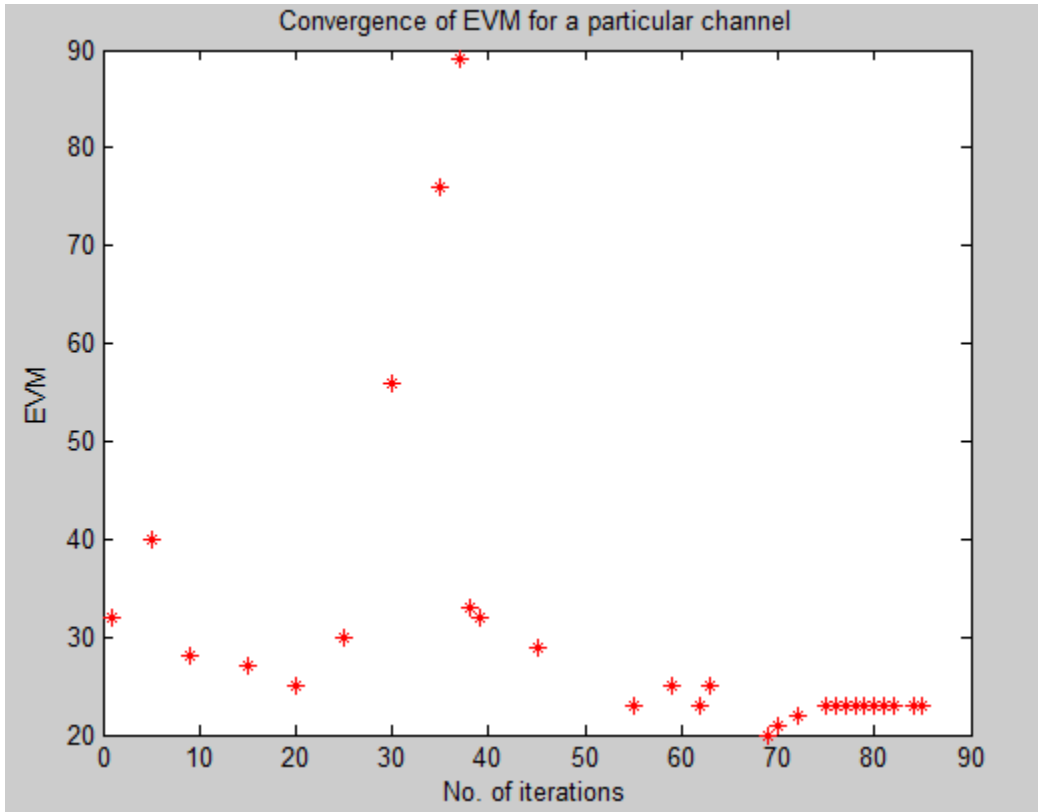
This data is the same as the data found using specific search algorithm. This proves the

Simulated Annealing algorithm correct. The functionality of the SA algorithm was also verified for instance 1,5,8 and 9. In all these cases, T was set to 50 initially and subsequent values of T were obtained by  $T = T_{prev} \times 0.98$ .



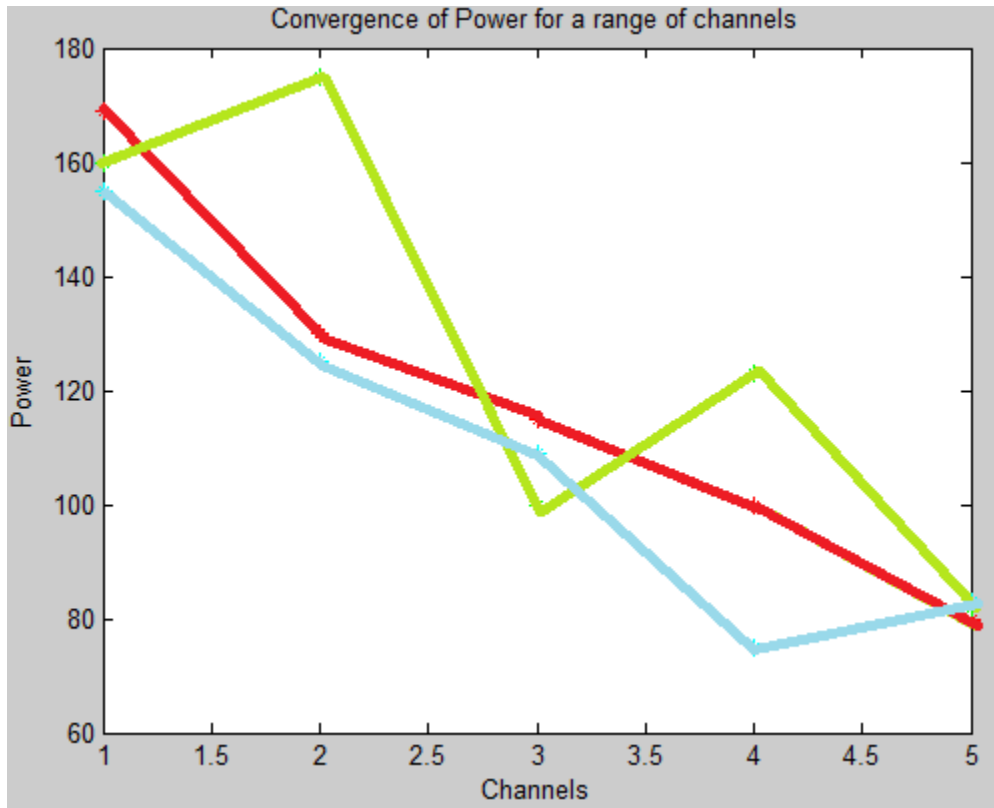
**Figure 6.10 Power Convergence**

The above graph explains how the power converges to its most accurate and lowest possible value as the number of iterations increase. In over 85 iterations, power reaches its actual value.



**Figure 6.11 EVM Convergence**

The above graph explains how the EVM converges to its most accurate value as the number of iterations increase. In over 85 iterations, the EVM reaches its actual value.



**Figure 6.12 Power Convergence across channels**

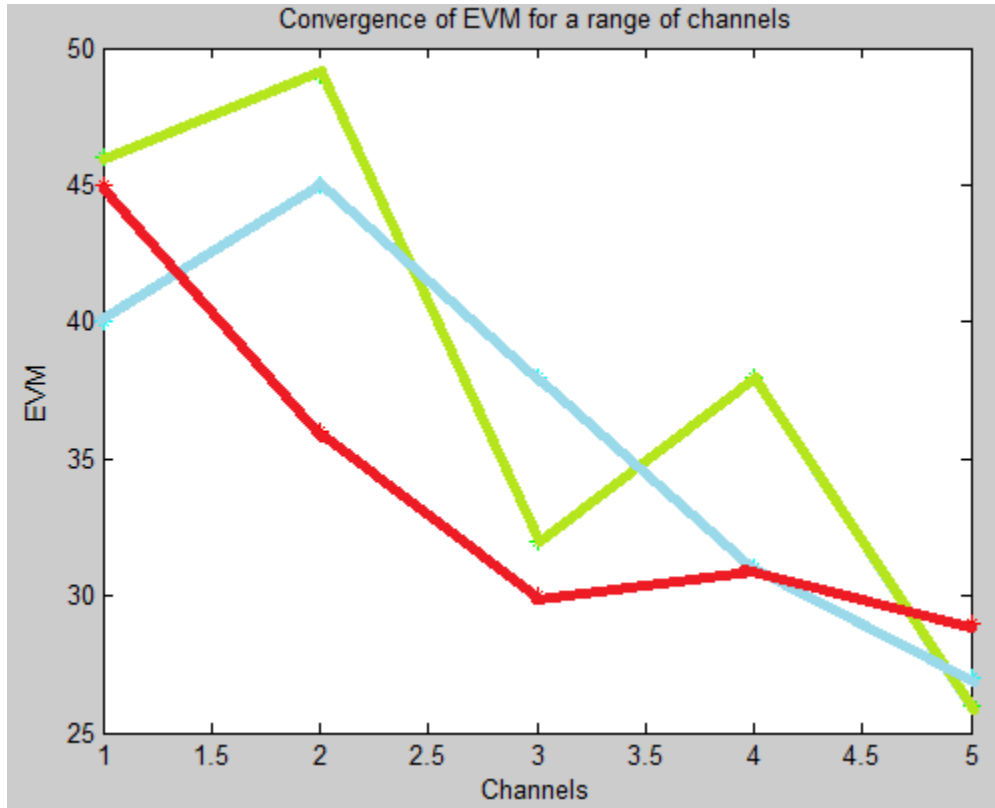
The above graph shows how power converges for a range of channels.

Green plot indicates 40 iterations

Blue plot indicates 75 iterations

Red plot indicates 90 iterations.

If the number of iterations is smaller, the graph is very abrupt and has huge variations. As the number of iterations increases, the graph turns smoother and converges to the best value.



**Figure 6.13 EVM Convergence across channels**

The above graph shows how EVM converges for a range of channels.

Green plot indicates 40 iterations

Blue plot indicates 75 iterations

Red plot indicates 90 iterations.

If the number of iterations is smaller, the graph is very abrupt and has huge variations. As the number of iterations increases, the graph turns smoother and converges to the best value.

We have hence, proved that using Simulated Annealing approach, it is possible to make a transmitter design that is adaptable to the process variations as well as changing channel conditions. Further, the rate of cooling the temperature determines how many iterations would be

performed and that in turn determines how accurate the results can be.

This way, knowing the nominal settings of one instance of a PA, we can find the optimum settings for another instance of PA. And each instance is nothing but a different object created due to the process variations. In real-time, sensors and actuators can be used to sense the changes and convert them to appropriate P1dB\_array, Gain\_array and Pdc\_array tables, and thus create new instances according to the process variations.

## 6.7 Conclusion

Our work demonstrates how the RF front end of a transmitter can be designed to learn by itself the best possible settings and configurations for giving minimum power with a fairly decent quality of transmission. The method is robust and simultaneously adapts to changes in the channel as well as process changes. There is a small reduction in the throughput in the initial phase but that is very low and only lasts for a very brief period.

On applying companding it was observed that the transmitted power reduced. It helped us reduce the dynamic range of the signal so it could be passed through a smaller range medium. A non-linear power amplifier was designed with 11 instances each of which simulated a different process variation. We used simulated annealing to find out the most optimum settings of the tuning knobs for low-power operation for one instance of the power amplifier for various channels using the information from the nominal power amplifier device.

It was observed that simulated annealing was more effective when the rate of cooling was really low. We could obtain the optimum settings and the minimum power. Both power and EVM converged over a large number of iterations of simulated annealing. The same experiment was repeated for multiple channel conditions.

The entire experiment was validated in hardware using actuators and sensors. Our design is now a self-learning system which adapts to both adverse channel conditions as well as unavoidable process changes.

## REFERENCES

Abhijit C., Debashis B. (2014) Self-learning RF Front Ends : Concurrent Adaptation to Channel Conditions and Process Variability Effects To Minimize Power Consumption

Monir M., Tushar R. (2013) High Power Amplifier Effects Analysis for OFDM System

Vishwajit S., Sanjay K. (2013) Comparative Analysis between A-law and  $\mu$ -law Companding Technique for PAPR Reduction in OFDM

Charan L. <http://complextoreal.com/tutorials/tutorial-22-orthogonal-frequency-division-multiplex-ofdm-dmt/#.UsokcPRDsuc>

<http://www.theprojectspot.com/tutorial-post/simulated-annealing-algorithm-for-beginners/6>

<http://apmonitor.com/me575/index.php/Main/SimulatedAnnealing>