ADAPTIVE DIGITAL PREDISTORTION FOR LINEARIZATION OF POWER AMPLIFIERS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Burak Şekerlisoy

June 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Abdullah Atalar(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. Tarık Reyhan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Cemal Yalabık

Approved for the Institute of Engineering and Science:

Director of the Institute

ABSTRACT

ADAPTIVE DIGITAL PREDISTORTION FOR LINEARIZATION OF POWER AMPLIFIERS

Burak Şekerlisoy
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Abdullah Atalar
June 2009

In most communication systems, power amplifiers are used to obtain high output power. The nonlinear characteristics of the power amplifier leads to the distortion of the output signal. This distortion affects the efficiency of the power amplifier. The way to reduce this effect is to linearize the power amplifier near the saturation region where it is nonlinear. The widely used technique for the linearization of power amplifiers is predistortion. The proposed technique for predistortion uses a LUT(look-up-table), a complex multiplier, an address calculator, delay elements and an adaptation logic. A new adaptation logic to update the LUT coefficients, is used. The predistorter is simulated in Matlab software using a baseband model for the power amplifier. 16-QAM baseband modulation is used to simulate the predistorter. In order to see the performance of the proposed predistorter, hardware logic is implemented in FPGA and experimental setup with RF circuits and RF power amplifier is used. For different LUT sizes, the algorithm is tested and for the LUT size of 64, nearly 15 dB improvement in power spectrum is observed. The LUT size of 64 is observed to be the optimal LUT size in the experiments.

Keywords: adaptive digital predistortion, look-up table predistortion, linearization, amplifier nonlinearity, FPGA.

ÖZET

UYARLANABİLİR SAYISAL ÖNBOZMA KULLANILARAK GÜÇ KUVVETLENDİRİCİLERİNİN DOĞRUSALLAŞTIRILMASI

Burak Şekerlisoy
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Abdullah Atalar
Haziran 2009

Günümüzde kullanılan iletişim sistemlerinde, güç yükselteçler yüksek güç elde etmek için kullanılır. Güç yükseltecin doğrusal olmayan bu özelliği güç yükselteç çıkışının bozulmasına neden olur. Bu bozulma güç yükseltecin verimini azaltır. Bu bozulmanın etkilerini azaltmanın yolu, güç yükselteci doğrusalsızlığının fazla olduğu doyma bölgesi yakınlarında doğrusallaştırmaktır. Güç yükselteçlerin doğrusallaştırılması için kullanılan yöntem önbozma yöntemidir. Bu tezde sunulan önbozma yöntemi, tablo, karmaşık sayı çarpıcı, adres hesaplayıcı, gecikme elemanları ve uyum devresinden oluşur. Tablo katsayılarını güncellemek için yeni bir yöntem geliştirilmiştir. Önbozucu Matlab yazılım ortamında simüle edilmiştir. Bu yazılımda güç yükseltecin temel bant modeli kullanılmıştır. Modülasyon türü olarak 16'lık dörtlük genlik modülasyonu seçilmiştir.Yazılımda simüle edilen önbozucu için, FPGA içinde donanım hazırlanmıştır. Bu donanımla beraber gerçek RF devreleri ve RF güç yükselteç kullanılarak farklı tablo boylarında testler yapılmıştır. Önbozucunun en iyi performansı 64'lük tablo boyutuyla sağlanmıştır ve bu tablo boyutunda güç izgesinde yaklaşık olarak 15 dB'lik iyileşme tespit edilmiştir.

Anahtar sözcükler: uyarlanabilir sayısal önbozma, taramalı tablo temelli ön bozma, doğrusallaştırma, güç yükselteç doğrusalsızlığı, spektrum verimi, FPGA..

Acknowledgement

I would like to express my special thanks to my supervisor Prof. Abdullah Atalar for his suggestions, supervision, and guidance throughout the development of this thesis.

I would like to thank the members of the thesis committee for reviewing the thesis.

I gratefully thank Oğuzhan Atak, OPTISIS and Makbule Pehlivan Aslan for their help and effort that they provided during the software and hardware implementation part of my study.

I would like to thank my beloved wife Sibel with deepest appreciation for her incessant encouragement and motivation.

I would also like to thank my parents and my brother for their love and support that they gave me all my life.

Finally, I would like to thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for the financial support provided during my M.S. study.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Background

Spectral efficiency is very important in today's mobile telecommunication systems. The power amplifier efficiency and linearity is a critical design issue. The nonlinear characteristics of power amplifier causes the power amplifier output to interfere with the adjacent channels and this interference has to be avoided in communication systems. The way to avoid this interference is either to reduce the power efficiency or to use linearization techniques.

Predistortion is a technique to compansate the nonlinear response of the power amplifier. The power amplifier distorts the signal and in order to have a linear replica of the input signal at the output of the power amplifier, the signal is distorted before being passed to the power amplifier. This distortion is done in such a way that it cancels with the distortion of the power amplifier. This is why the method is called predistortion. The predistorter models the inverse of the power amplifier. The power amplifier introduces both phase and amplitude nonlinearities and based on these phase and amplitude response of the power amplifier, a predistorter is constructed that has also nonlinear phase and amplitude responses. It uses these nonlinear responses to cancel the nonlinearity effects of the power amplifier.

The predistortion can be applied in baseband to the transmitted signal before the upconversion and it can be applied in digital domain. The digital predistortion is a flexible design and it is easy to adapt the predistortion to communication systems. FPGAs or DSPs can be used to implement digital predistortion. Another issue with the digital predistortion is that it can be made adaptive to the variations of the power amplifier characteristics over time. This method is called as the Adapative Digital Predistortion and throughout this thesis the studies are done to build an adaptive digital predistorter scheme.

1.2 Thesis Objective and Outline

Throughout this thesis, the aim is to develop a new adaptive digital predistortion scheme. The proposed scheme is easy to be implemented in hardware and software. The effects of design parameters will be observed to find the optimal scheme. The outline of the thesis is as follows: Chapter 2 is an introduction about the nonlinearity concept and the characteristics of the power amplifier showing the baseband model used in simulations. Chapter 3 describes the basic predistortion schemes in today's systems. Chapter 4 introduces a new look-up-table based adaptive predistortion scheme and the simulation results. Chapter 5 presents a new hardware implementable adaptive predistortion scheme with the hardware and software design. The design in this proposed scheme can be easily adapted to other systems. Chapter 6 gives the conclusion and presents possible future work.

Chapter 2

POWER AMPLIFIER AND NONLINEARITY

2.1 Concept of Nonlinearity

Let us characterize a system by a block box with input x(t) and output y(t) as shown in Figure 2.1. In general, the system is said to be nonlinear if the output power is a nonlinear function of the input power

$$y(t) = N[x(t)] \tag{2.1}$$

where N is the nonlinear transfer function. If we expand the nonlinear function in simple Taylor series, the result shows the dependence of the nonlinearity on $x(t)^2$. For periodic input signals, the nonlinearity results in the generation of new frequency components at the output, as shown in Figure 2.1. This distortion caused by nonlinear systems is in the form of crossmodulation and intermodulation at the output signal spectrum. As the amplitude arises, these modulation components grow faster than the signal itself and dominate the spectrum. This distortion cannot be prevented by using linear operations such as filtering since intermodulation causes new frequency components get bigger in the adjacent channels. As a result, the phase and the amplitude of the output signal will be affected depending on the amplitude and phase of the input signal. These effects

lead to distortion at the output signal.

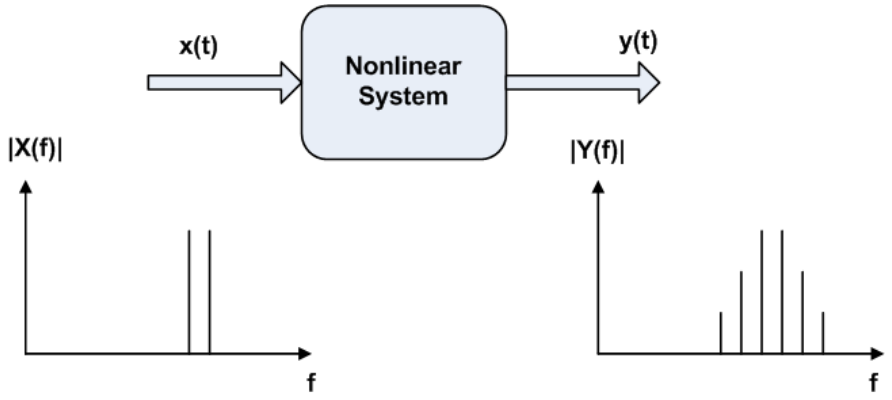

Figure 2.1: A Nonlinear System

2.2 Power Amplifier Characteristics

In telecommunications obtaining the output power high enough is the task of a power amplifier. After the modulation and upconversion, the input is multiplied with a gain by power amplifier to achieve the required power at the output. As it can be seen from the Figure 2.2(b) the gain is constant for low input powers and it reduces as approaching its saturation region. Saturation region is easily visible from the output power curve where inrease in input power brings no additional increase in output power. In Figure 2.2 (a) 1 dB compression point is also shown, which refers to the output power level at which the amplifier's transfer characteristics deviates from the ideal one by 1 dB. This is a widely used measure of amplifier linearity.

Power amplifiers are mainly characterized by the second and third-order intercept point, 1 dB gain compression point and input back-off. When the input is increased, the second harmonic will increase proportional to the square of the input signal and the third harmonic will increase proportional to the cube of the input signal which means that the second and the third harmonics will increase at a greater rate than the fundamental component. The signal level at which the second harmonic is equal to the fundamental is called the second order intercept

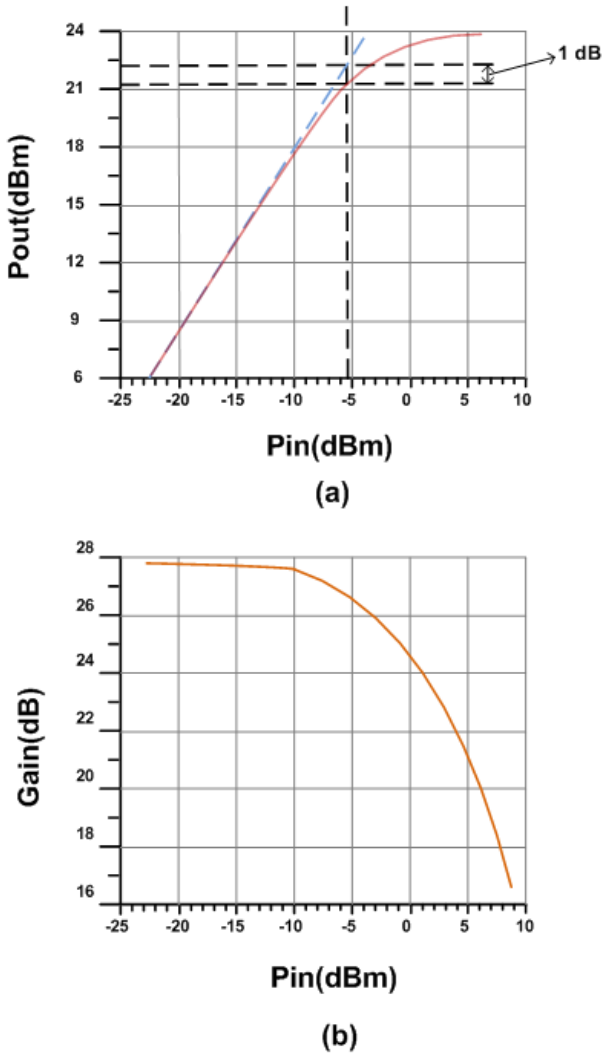

Figure 2.2: Power Amplifier Characteristics (a) Output Power (b) Gain

point and the point at which the third harmonic is equal to the fundamental is called the third order intercept point. The 1 dB compression point is defined as the point at which the output power level has dropped 1 dB below the ideal output power. The ideal output power is the power where the power amplifier is fully linear. Input back-off is defined as the ratio of the signal power measured at the input of the power amplifier to the input signal power that produces the maximum signal power at the amplifier's output.

2.3 AM/AM and AM/PM Nonlinearity of Power Amplifiers

Nonlinear relation between input and output power leads to AM/AM distortion. If the input signal includes amplitude modulation, power amplifier nonlinearity converts this modulation also into phase modulation and distorts the phase of the input signal. This is known as AM/PM conversion effect of the power amplifier. These two effects are shown in Figure 2.3.

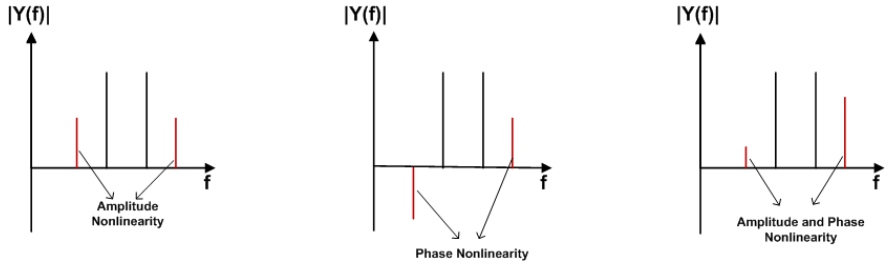

Figure 2.3: IMD products due to AM/AM and AM/PM Conversion

A nonlinear and memoryless power amplifier can be modeled as a power series transfer function and the input-output relation of the power amplifier can be written as in Equation 2.2.

$$V_o(t) = a_0 + a_1 V_{in}(t) + a_2 V_{in}(t)^2 + a_3 V_{in}(t)^3 + a_4 V_{in}(t)^4 + a_5 V_{in}(t)^5 + \cdots \quad (2.2)$$

A two-tone test signal can be applied to the input of the power amplifier whose relation is given by Equation 2.2. The two tone test signal is given in Eguation 2.3.

$$V_{in}(t) = vcos(w_1 t) + vcos(w_2 t) \quad (2.3)$$

Then the output of the power amplifier is given by Equation 2.4

$$\begin{aligned} V_o(t) = {} & a_1 v[cos(w_1 t) + cos(w_2 t)] + a_2 v^2[cos(w_1 t) + cos(w_2 t)]^2 + \\ & a_3 v^3[cos(w_1 t) + cos(w_2 t)]^3 + a_4 v^4[cos(w_1 t) + cos(w_2 t)]^4 + \\ & a_5 v^5[cos(w_1 t) + cos(w_2 t)]^5 + \cdots \end{aligned} \quad (2.4)$$

The terms, except the fundamental term, in Equation 2.4 generates distortion products . In general, even order terms generate products which can be accepted as out-of-band distortion. Out-of-band distortion products can be avoided by linear operations. However, odd order terms generate distortion products which are in-band and they are hard to get rid of. The adjacent band distortion is a very important problem in communications while out-of-band distortion can be suppressed by filtering. In Figure 2.4, the fundamental compo-

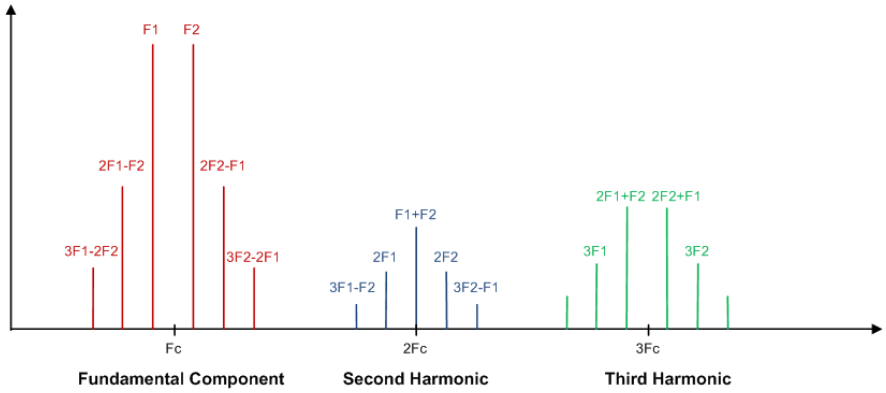

Figure 2.4: Harmonic Distortion for Two Tone Signal

nent,second and third harmonic products are shown. Second and third harmonic products are far from the fundamental component and can be avoided. However the distortion products which are adjacent to the fundamental component $(2F1 - F2, 2F2 - F1, 3F1 - 2F2, 3F2 - 2F1)$ are difficult to be avoid. And also their amplitudes grow faster than the fundamental component's amplitude and it becomes a problem if the power amplifier is to be operated at a high input power level.

2.4 Power Amplifier Modeling

The AM/AM and AM/PM distortion of the power amplifier can be modeled as a nonlinear complex gain. In the simulations throughout this thesis, the Saleh model for memoryless power amplifiers([1]) have been used. The model is constructed in baseband as a complex baseband gain. It simply generates a complex

gain whose phase and amplitude depend on the amplitude of the input signal. The distortion functions are given by Equations 2.5 and 2.6.

$$F_{AM/AM}(r) = \frac{\alpha}{1+\beta r^2} \tag{2.5}$$

$$F_{AM/PM}(r) = \frac{Ar^2}{1+Br^2} \tag{2.6}$$

where α, β, A, B are constants which can be used to define the nonlinear characteristics of the power amplifier. In the simulations throughout this thesis they are selected as

$$\begin{aligned} \alpha &= 3 \\ \beta &= 0.0552 \\ A &= 1 \\ B &= 9 \end{aligned} \tag{2.7}$$

Using the values in Eguation 2.7 the power amplifier is modeled in baseband. The AM/AM and AM/PM distortion is plotted in Figures 2.5 and 2.6.

Let us assume a signal $a + jb = r^{j\theta}$ in baseband is passing through the power amplifier model. Depending on the amplitude of the input signal($r = \sqrt{a^2 + b^2}$), a complex gain is found using Equations 2.5 and 2.6. The complex gain has amplitude $|G(r)| = F_{AM/AM}(r)$ and phase $\angle G(r) = F_{AM/PM}(r)$. Then the output of the power amplifier model is calculated as the multiplication of two complex numbers : $re^{j\theta}$ and $|G(r)|e^{j\angle G(r)}$. The output is $r|G(r)|e^{j(\theta+\angle G(r))}$ and it is not a linear function of the input signal.

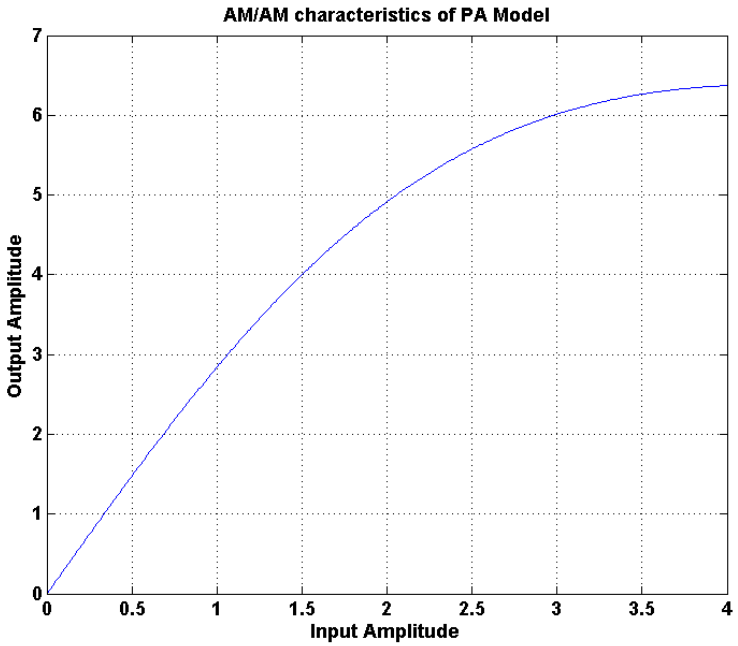

Figure 2.5: AM/AM Distortion of Power Amplifier using Saleh Model

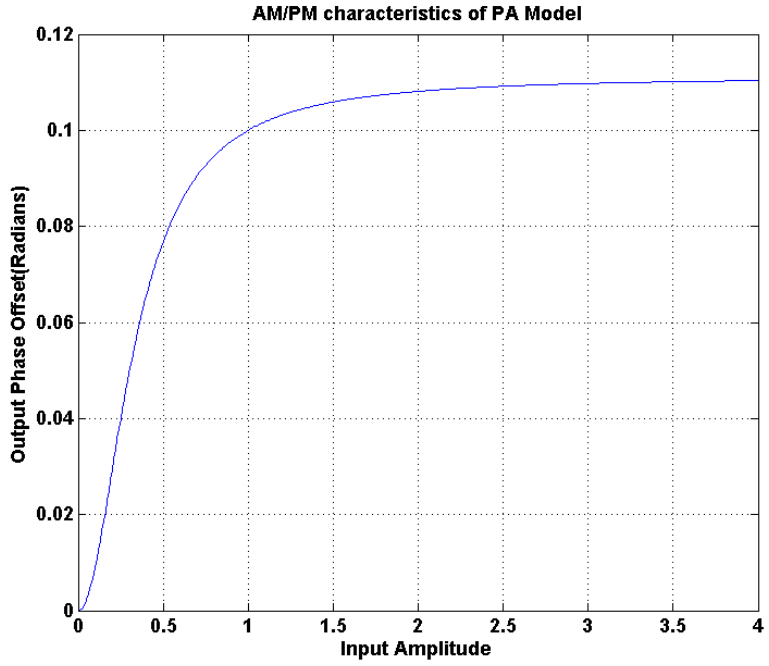

Figure 2.6: AM/PM Distortion of Power Amplifier using Saleh Model

Chapter 3

LINEARIZATION USING PREDISTORTION

This chapter focuses on linearization of power amplifiers using adaptive digital predistortion showing some of the schemes using today : LUT-based predistortion and polynomial based predistortion.

3.1 Basics of Predistortion

Predistortion is a technique to obtain linear amplification and high power efficiency. It allows the power amplifier to operate at higher power level by correcting the distortion, caused by nonlinearity, up to the saturation. Beyond the saturation level the distortion cannot be recovered since any increase in input power will generate no additional increase in output power.

The principle of the predistortion is to distort the signal in such a way that the cascade response of the predistorter and the power amplifier is linear. For this purpose, a nonlinear predistorter block is implemented in front of the power amplifier. The predistorter block then generates a distortion which compansates the distortion of the power amplifier and the output of the power amplifier has a

linear relation with the input of the predistorter.

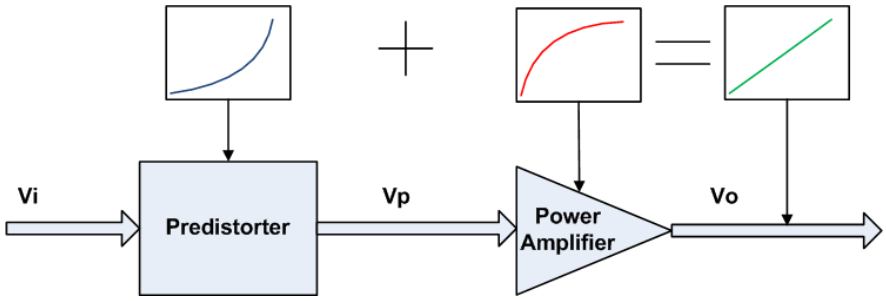

Figure 3.1: Basic Predistortion Scheme

Referring to Figure 3.1, power amplifier output,V_o, is a scaled replica of the predistorter input, V_i. Writing down the input-output relations for the blocks in Figure 3.1 we get the following equations:

$$\begin{aligned} V_p &= V_i F(|V_i|) \\ V_o &= V_p G(|V_p|) \end{aligned} \tag{3.1}$$

writing kV_i for V_o and $V_p = V_i F(|V_i|)$ in the above equations we get Equation 3.2

$$\begin{aligned} kV_i &= G(|V_i F(|V_i|)|)F(|V_i|)Vi \\ k &= G(|V_i F(|V_i|)|)F(|V_i|) \end{aligned} \tag{3.2}$$

The equation 3.2 shows the condition for the predistorter to linearize the power amplifier. This equation is important in the sense that it tells us that, in order to linearize the power amplifier, we first have to know the response of the power amplifier which is the $G(.)$ function in Equation 3.2. The performance of the linearization is directly depending on how precise we know or estimate the AM/AM and AM/PM response of the power amplifier. If the precision is poor, linearization will also be poor.

3.2 Digital Predistortion

Digital predistortion is implemented in baseband and this method is used in base-stations for mobile communication systems for linearity improvement, where adjacent channel power ratio is an issue. Digital predistortion is simple, stable and flexible. It can be adapted to the variations of the power amplifier and other circuit components. This adaptation capability is a powerful property in the expense of the hardware and software complexity. Figure 3.2 is the generic structure used for adaptive digital predistortion.

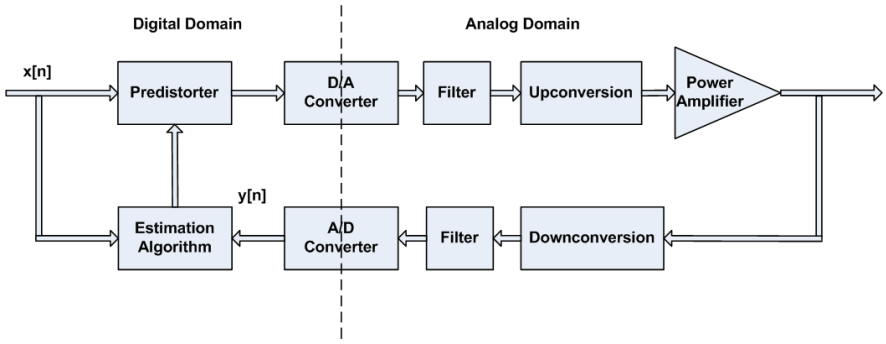

Figure 3.2: Basic Predistortion Architecture

Digital Predistorter techniques use the architecture in Figure 3.2. The samples in baseband pass through the predistorter before being transmitted to the power amplifier and the output of the predistorter is sent to the D/A converter. Upconversion is then applied to the output of the D/A converter before the power amplifier. Power amplifier output is divided into two parts : one for transmission to antenna and one for feedback to the predistorter. The latter part is downconverted and then sampled by the A/D converter. The carrier of the downconversion and upconversion has to be common. The output samples of the A/D converter is fed into the predistorter for the adaptation operation. In general predistorter learns the characteristics(AM/AM and AM/PM responses of the power amplifier) using the feedback path. Predistorter knows the data it sends to the power amplifier and the data it receives from the power amplifier. Using this information, it adapts itself and distorts the samples in baseband to compansate the power amplifier distortion. Feedback path makes the predistorter technique a powerful technique in terms of adaptation capability. However, it also brings

stability problem and hardware and software complexity. Stability problem has to be taken into account and predistorter has to be prevented to go into unstable operation.

3.3 Digital Predistortion Schemes

There are mainly two different digital predistortion types : the LUT based predistorter([2],[3]) and the polynomial predistorter([4],[5]). In LUT-based predistortion, predistorter is modeled as a look-up table whose coefficients are complex values. Incoming signal is multiplied with the complex gain from the LUT according to its amplitude. In polynomial predistortion, predistorter is modeled as a polynomial function and the coefficients of the polynomial are adapted so that the predistorter and the power amplifier is linear. In theory, LUT-based predistorter can linearize the power amplifier more precisely while the precision of the polynomial predistortion directly depends on the order of the polynomial. In general 5^{th} order polynomial is required to suppress the 3^{rd} and 5^{th} order IMD products in the adjacent frequency band ([4]). In LUT-based method, the adaptation is done by comparing the inphase and quadrature values of the incoming signal and the output of the power amplifier. The input of the predistorter is the original signal and adaptation is done to achieve the output of the power amplifier as a linear replica of the original signal. In polynomial predistortion, the adaptation is done to minimize the adjacent channel power at the output of the power amplifier. This power is taken as an error signal and is to be minimized by polynomial predistortion using direct search or gradient methods. Adaptation algorithm, in terms of complexity, is difficult using polynomial predistortion.

3.3.1 Look-up-table (LUT) Based Predistortion

LUT-based predistorter is simpler than the polynomial predistorter if we consider the complexity and calculations. Only six multiplications are required while calculating the adress of the LUT and complex output signal. However, in polynomial predistortion, more multiplication and addition operations are required.

LUT coefficients must be calculated according to a particular addressing. Addressing by the magnitude square is the easiest method and it can be done by simply adding the squares of the I and Q values of the input baseband signal. This is the power of the input signal. Addressing in this way gives more important to the power levels near saturation and it improves the efficiency and the linearization performance of the predistorter.

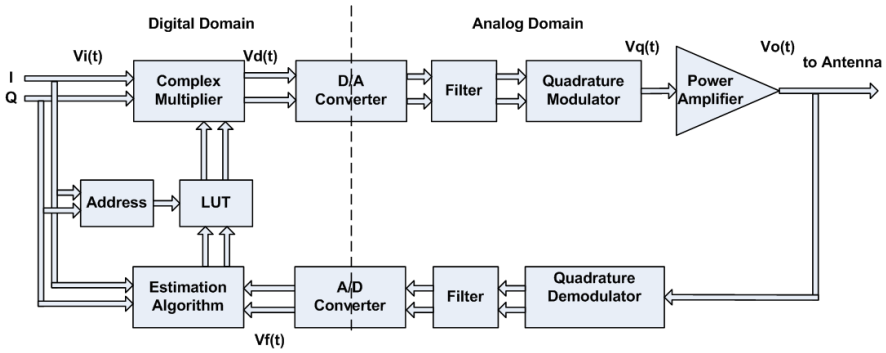

Figure 3.3: LUT-Based predistorter Scheme

A detailed block diagram of LUT-based predistorter is shown in Figure 3.3. The system including the predistorter consists of address calculation unit,LUT, complex multiplier, ADC, DAC, filters, modulator, demodulator, which has to have a common carrier with the modulator, and power amplifier. I and Q baseband signals are input to the system and they are multiplied by a complex gain from the LUT which models the inverse of the power amplifier characteristics. Once the complex signal $V_i(t)$ arrives at the input, the address is obtained by calculating the magnitude square and this address is sent to the LUT. Then the corresponding complex gain in the LUT is multiplied with the input signal and we get $V_d(t)$. $V_d(t)$ is then converted to the analog signal and after being filtered and modulated, it is sent to the power amplifier. The output of the power amplifier

$V_o(t)$ is used for adaptation to the variations of the characteristics of the power amplifier for reliable operation.

Feedback signal $V_f(t)$ and the original signal $V_i(t)$ is used the find the error signal and using the error signal the LUT coefficients are updated to have a linear relation between $V_o(t)$ and $V_i(t)$. In order to adapt the predistorter to power amplifier nonlinearity variations Equation 3.2 can be used to update the LUT coefficients. In LUT-based method this equation is used to generate an iterative method which is in general the well-known LMS algorithm.

Referring to Figure 3.3, Let the predistorter response be $F(.)$ and the power amplifier response be $G(.)$. Then using Equation 3.2 the relationship between $F(.)$ and $G(.)$ for a linear response can be written as in Equation 3.3

$$K = F(x_i)G(x_i|F(x_i)|) \tag{3.3}$$

where x_i is the amplitude of the incomig signal V_i. Then the adaptation is done according to the Equation 3.4 as follows

$$F(x_i, j+1) = \frac{K}{G(x_i|F(x_i, j)|)} \tag{3.4}$$

In Equation 3.4, j is the iteration index and $F(x_i, j)$ is the complex LUT value at iteration j. This equation can be solved by using LMS algorithm ([6]) and applying this equation in runtime makes the predistortion adaptive. The variations in $G(.)$ which are the variations of the characteristics of the power amplifier, will be adapted by changing the predistorter characteristics $F(.)$ so that Equation 3.4 is satisfied.

3.3.2 Polynomial Predistortion

Polynomial predistorter uses complex polynomial function to model the inverse of the power amplifier. Each sample coming into the predistorter is processed using the ploynomial function. The order of the polynomial function depends on the distortion of the power amplifier. For instance, if power amplifier generates 5^{th} order distortion products then a 5^{th} order polynomial can be used. ([4],[5])

Figure 3.4 shows the detailed block diagram of a polynomial predistorter. The adaptation is done by updating the complex-valued coefficients of the polynomial function to fit the predistorter response to the inverse of the power amplifier.

The polynomial predistorter is composed of a address calculation unit using amplitude square as in LUT-based method, a polynomial predistortion function, a complex multiplier, an optimization circuit for polynomial coefficients, DAC, ADC and the feedback path. The number of multiplication and addition operations is higher and it also arises with the order of the polynomial. The complex voltages $Vi(t)$, $Vd(t)$, $Vq(t)$ and $Vo(t)$ are as defined before in the LUT based predistorter case. Just the feedback signal $V_f(t)$ is no more a complex voltage but a scalar value proportional to the unwanted adjacent channel power emission. In the following the derivation of a polynomial predistortion system is given which is done according to magnitude square of the input.

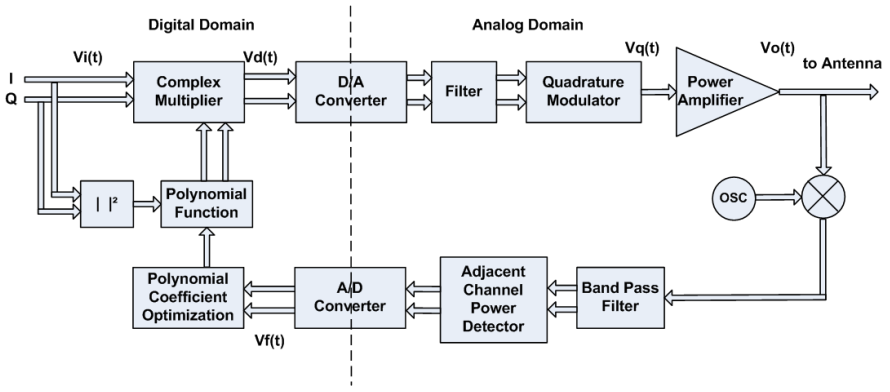

Figure 3.4: Polynomial predistorter Scheme

The following analysis of the polynomial predistorter assumes the power amplifier and predistorter response as a function of the magnitude square of the input signal. The odd powers of the magnitude are not used while modeling power amplifiers, because these terms do not generate in-band distortion products. The referring to Figure 3.4, we can write the response of the power amplifier as

$$G(|V_d(t)|^2) = \beta_1 + \beta_3|V_d(t)|^2 + \beta_5|V_d(t)|^4 \tag{3.5}$$

then the output of the power amplifier is written as

$$V_o(t) = V_d(t)G(|V_d(t)|^2) \tag{3.6}$$

combining equations 3.5 and 3.6

$$V_o(t) = \beta_1 V_d(t) + \beta_3 V_d(t)|V_d(t)|^2 + \beta_5 V_d(t)|V_d(t)|^4 \tag{3.7}$$

Like power amplifier, the predistorter is modeled as a function of the magnitude square of the input signal as

$$F(|V_i(t)|^2) = \alpha_1 + \alpha_3|V_i(t)|^2 + \alpha_5|V_i(t)|^4 \tag{3.8}$$

the output of the power amplifier is written as

$$V_d(t) = V_i(t)F(|V_i(t)|^2) \tag{3.9}$$

From Equations 3.6 and 3.9 we can write the relationship between $V_i(t)$ and $V_o(t)$ as

$$V_o(t) = V_i(t)F(|V_i(t)|^2)G(|V_i(t)F(|V_i(t)|^2)|^2) \tag{3.10}$$

and if the linear gain we are targeting to achieve is K then Equation 3.11 has to be satisfied.

$$V_o(t) = K(V_i(t))V_i(t) \tag{3.11}$$

Using Equations 3.11 and 3.10 we can write

$$K(V_i(t)) = F(|V_i(t)|^2)G(|V_i(t)F(|V_i(t)|^2)|^2) \tag{3.12}$$

then $K(V_i(t))$ is a function of the magnitude square of $V_i(t)$ and can be written as

$$K(V_i(t)) = \gamma_1 + \gamma_3|V_i(t)|^2 + \gamma_5|V_i(t)|^4 \tag{3.13}$$

In order to liearize the power amplifier γ_1 must be equal to the targeted gain and γ_3,γ_5 must be small. Polynomial predistortion algorithms tries to achieve Equation 3.13([4]

Chapter 4

A NEW DIGITAL ADAPTIVE PREDISTORTION METHOD

This chapter introduces a new algorithm for adaptive digital predistortion using look-up-table method. The following sections describe the new algorithm and show the simulation results. The algorithm works in baseband and throughout this chapter, power amplifier and predistorter will be modeled as they both work in baseband. For power amplifier baseband model, we will use the nonlinear memoryless Saleh model which is explained in Section 2.4

4.1 New predistorter Scheme

The scheme to be used in the adaptive digital predistorter is given in Figure 4.1. It consists of complex multiplier, delay elements, LUT, adaptation logic and power amplifier.

The LUT and complex multiplier models the predistorter response and the adaptation logic updates the coefficients of the LUT according to the power amplifier characteristics. The coefficients of the LUT are complex. These complex coefficients are calculated by the adaptation logic. The blocks in Figure 4.1 is

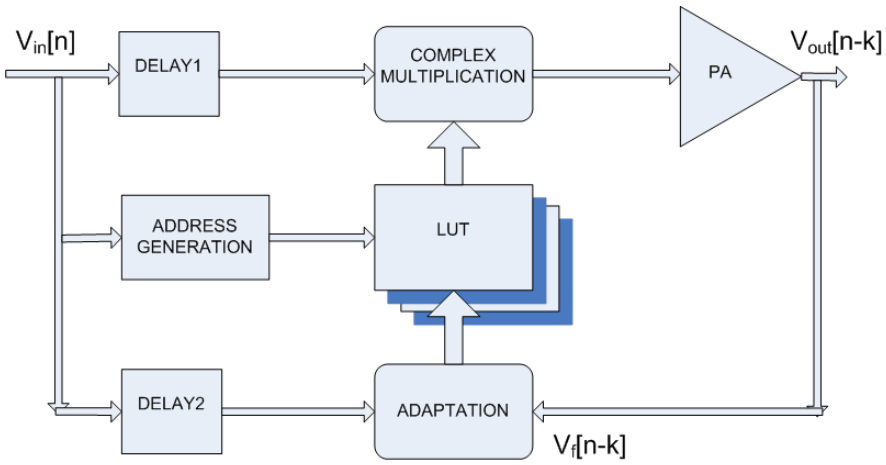

Figure 4.1: New Predistortion Structure

explained in detail in the following subsections.

4.1.1 Delay Elements

The delay elements in the adaptive predistorter scheme provide alignment. There are two delay elements in Figure 4.1. The first one, DELAY1, is used in order to align the data with its corresponding LUT coefficients. If the latency caused by address generation from the input signal is t_1 and the latency caused the loading of the corresponding complex coefficient to the complex multiplier is t_2, then in order to multiply the input signal with its corresponding complex LUT value, we have to have a delay of $t_1 + t_2$. This delay is DELAY1. DELAY2 is required to align the incoming data,$V_{in}[n]$ and the feedback data,$V_f[n-k]$. The adaptation logic has to know the feedback data corresponding to the incoming data and DELAY2 is used for this purpose. One of the key points in this design is that the DELAY2 resolution depend directly on the sampling rate. For instance, if the sampling clock period is 10ns(nanoseconds), and the delay between incoming data and the feedback data is 55ns, then the latency is assumed as 50ns or 60ns. This is expected since the alignment is done in discrete time domain and the resolution is limited with the sampling period.

4.1.2 Address Calculation

The LUT is addressed according to the magnitude square of the input baseband signal. The address calculation unit simply finds the amplitude square of a complex number. It maps the amplitude square of the input signal to the address of the LUT. This mapping with amplitude square enables giving more importance to the high amplitude signals where the power amplifier is nonlinear, and giving less importance to low amplitude signals where the power amplifier is linear. The

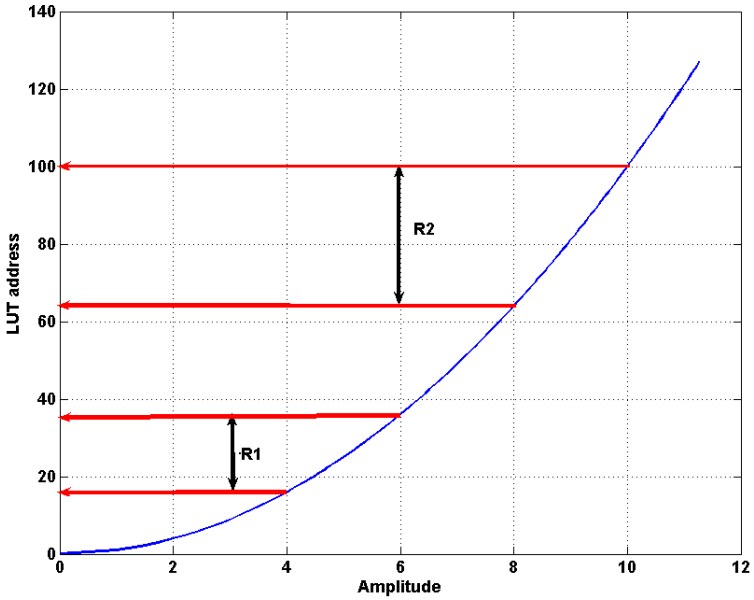

Figure 4.2: LUT Adressing with respect to the amplitude

Figure 4.2 shows the LUT address with respect to the amplitude if the LUT address is linear with amplitude square of the signal. With the same interval on the amplitude axis, we have more address space at high amplitudes. R_1 is the address range for input amplitudes between 4 and 6 and R_2 is the address range for input amplitudes between 8 and 10 and it is clear that $R_2 > R_1$ which means that we have more address space in the LUT for the signals with higher amplitudes.

4.1.3 Complex Multiplication

In order to compansate both AM/AM and AM/PM distortions of the power amplifier the LUT coeeficients are chosen to be complex. The complex multiplier multiplies the input sample with its corresponding complex gain. Complex multiplication means that we are changing the amplitude and the phase of the incoming sample.

4.1.4 Look-Up-Table

Look up table is one of the core blocks of the predistorter and it models the inverse of the power amplifier. The structure of the LUT is shown in Figure 4.3. The LUT which is used in this algorithm has two parts : real part and imaginary part. For every incoming sample, LUT is addressed and a complex gain $r_k + ji_k$

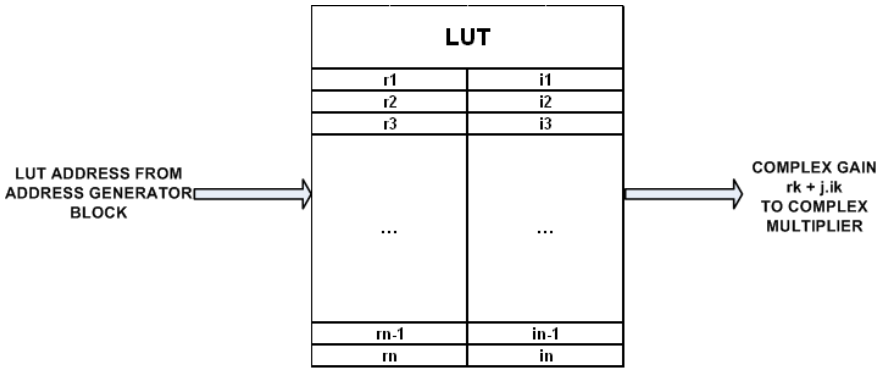

Figure 4.3: Look-Up-Table Structure

is loaded to the complex multiplier. This complex gain is multiplied with the incoming baseband signal and compansates it for phase and amplitude distortion of the power amplifier.

4.1.5 Adaptation

The adaptation logic is a block which inputs incoming samples and feedback samples and outputs LUT coefficients. The difference of the new algorithm from

other LUT-based methods is based on the difference of the adaptation of the LUT to the variations of the power amplifier. The new proposed method is summarized in Figure 4.4. The steps of the flowchart is explained in this section in detail.

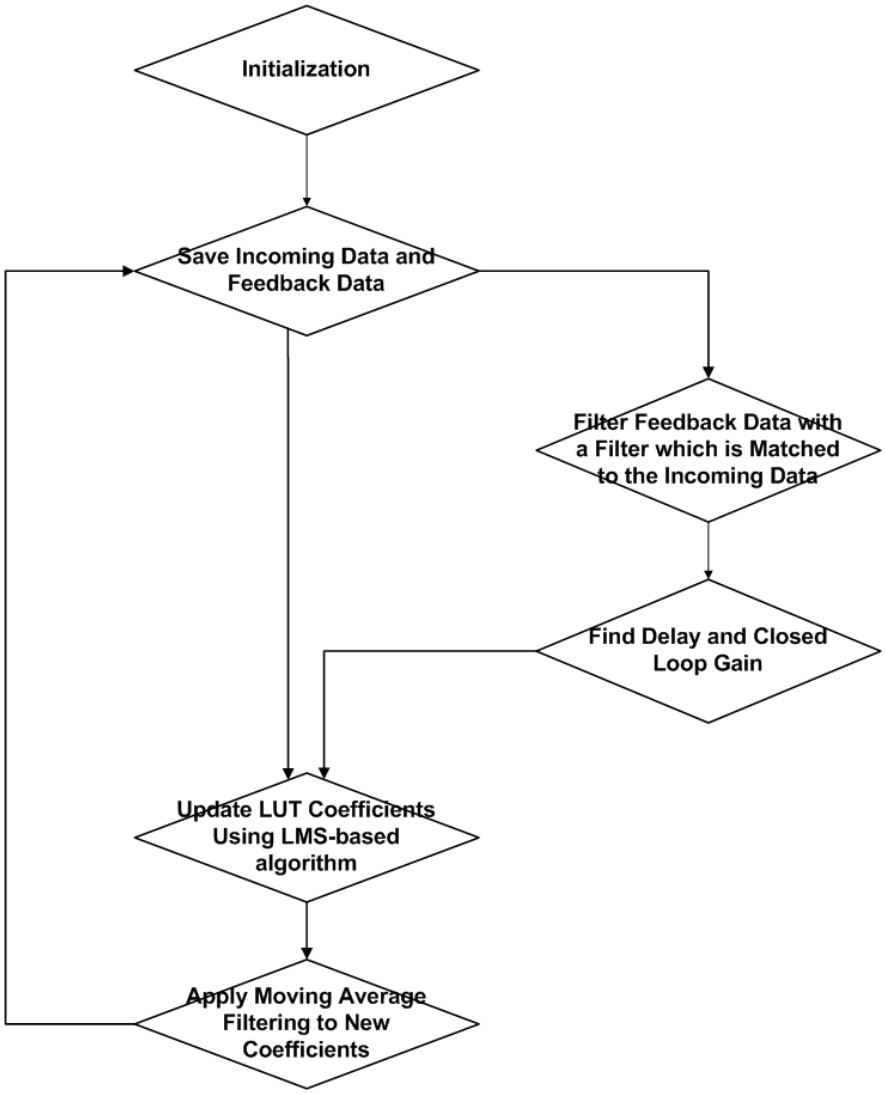

Figure 4.4: Flow-Diagram of the Algorithm

4.1.5.1 Initialization

In this step, we initialize all the LUT coefficients for unit delay. For this purpose we set all the real parts of the LUT coefficients to 1 and all the imaginary parts

of the LUT coefficients to 0.

4.1.5.2 Finding Delay and Closed-Loop Gain

The delay between incoming data and the feedback data has to be known for alignment. The gain of the closed loop also has to be known for the adaptation algorithm. In order to find these variables, a matched filter is used. This filter is matched filter of the incoming data and it is applied to the feedback data. The filtering operation is shown in the following Equations :

$$s'[n] = as[n-k] + v[n] \tag{4.1}$$

where $s'[n]$ is the feedback data, $s[n]$ is the incoming data, a is the closed-loop gain,k is the delay between incoming data and feedback data and $v[n]$ is the noise which is assumed to be independent of feedback data and incoming data. Then Using Equation 4.1 we can write Equation 4.2

$$\begin{aligned}\frac{< s'[n], s[n] >}{||s[n]||^2} &= \frac{< as[n-k] + v[n], s[n] >}{||s[n]||^2} \\ &= \frac{< as[n-k], s[n] >}{||s[n]||^2} + \frac{< v[n], s[n] >}{||s[n]||^2}\end{aligned} \tag{4.2}$$

The $< . >$ operation in Equation 4.2 is the inner product operation. Notice that the second term in this Equation is zero since the noise is independent of the incoming data. So the result is simply the inner product of the incoming data with itself at lag k multiplied by the closed-loop gain and divided by the norm-square of the incoming data as in Equation 4.3

$$\frac{< s'[n], s[n-l] >}{||s[n-l]||^2} = a\frac{< s[n-k], s[n-l] >}{||s[n-l]||^2} \tag{4.3}$$

Equation 4.3 shows us the way to find the delay and closed-loop gain. The term in this equation gives its maximum when $l = k$. The result at lag k is simply the closed-loop gain since $< s[n-k], s[n-k] >$ is equal to $||s[n-k]||^2$. So in order to find the delay we have to find at which lag $< s'[n], s[n-l] >$ is maximum and the a parameter at this lag gives us the closed-loop gain. The method to find gain and delay variables is shown in Equations 4.4 and 4.5.

$$gain = \max_l \frac{< s'[n], s[n-l] >}{||s[n]||^2} \tag{4.4}$$

$$delay = arg\max_{l} \frac{< s'[n], s[n-l] >}{||s[n]||^2} \tag{4.5}$$

The Equations 4.4 and 4.5 can be implemented using a filter which is matched to the incoming data,$s[n]$. So the impulse response of the filter is $s_{MF}[n] = s^*[-n]$ and $y[n]$,the output of the convolution of the feedback data with the matched filter $s_{MF}[n]$, is shown in Equation 4.6.

$$y[l] = \sum_{k=-\infty}^{\infty} s'[k]s_{MF}[l-k] \tag{4.6}$$

Substituting $s_{MF}[n] = s^*[-n]$ into Equation 4.6 we get Equation 4.7.

$$y[l] = \sum_{k=-\infty}^{\infty} s'[k]s^*[k-l] =< s'[n], s[n-l] > \tag{4.7}$$

We find a scheme to implement Equation 4.7 as in Figure 4.5. The normalization simply divides $y[n]$ to the norm-square term,$||s[n]||^2$

Figure 4.5: Matched-Filter Usage

4.1.5.3 Coefficient Update Algorithm

Before developping our own method, we used secant method for updating the LUT coefficients([6]). Referencing the Figure 4.6 the following steps were followed in order to apply the secant method.

Step1 : Find the index for the input signal $Vm(k)$. Indexing of the LUT was done with respect to the amplitude-square of the input samples. Let this index be i.

Step2 : Multiply the input sample with the corresponding gain of the predistorter,$Fi(k)$).

$$V_d(k) = F_i(k).V_m(k) \tag{4.8}$$

Step3 : Find the error signal

$$e(k) = PA_{Gain}.Vm(k) - Va(k) \tag{4.9}$$

Step4 : Update the gain at the index i of the LUT using secant method

$$F_i(k+1) = F_i(k) + \alpha.e(k)\frac{F_i(k) - F_i(k-1)}{e(k) - e(k-1)} \tag{4.10}$$

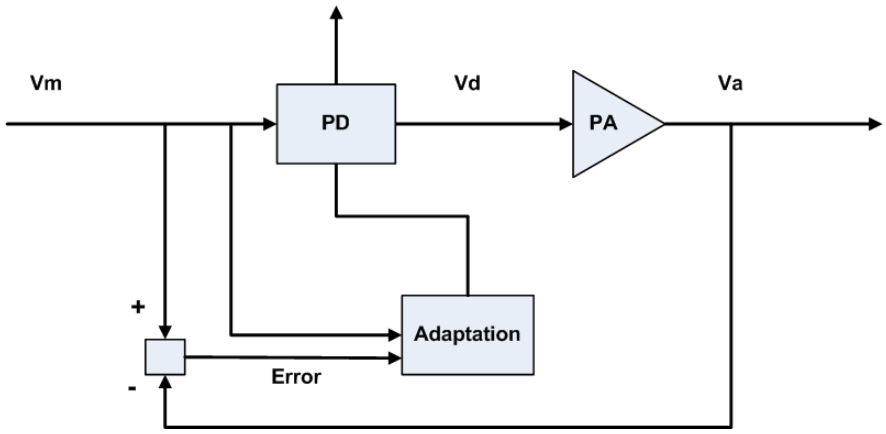

Figure 4.6: Adaptation Scheme

Following these steps, we try to find an easier way to update the LUT coefficients. In other words, we try to change this algorithm in *Step4* to have an easier and hardware implementable method. In real life while implementing this design on FPGA, the division operation in the algorithm takes too many resources and the design is not cost-effective. For that reason we had to add some improvement to the algorithm and in other words we have to find a term instead of the fractional term. The new term is found by observing Step4 in vector domain. The vector representation of the new method is shown in Figure 4.7-a. The vectoral representation shows the baseband signals in two dimensional vector domain. The error signal is shown in Figure 4.7-a. Our aim for linearization is to move the output of the power amplifier,$V_a[k]$, towards the input of the predistorter,$V_m[k]$, assuming the power amplifier has unity gain in linear region. In order to have undistorted sample at the output of the power amplifier, we have to rotate the incoming sample, in other words the gain of the predistorter must move $V_m(k)$ in the direction so that $V_a(k)$ moves towards $V_m(k)$. This means that we must multiply the error signal with a complex number and the multiplication must move

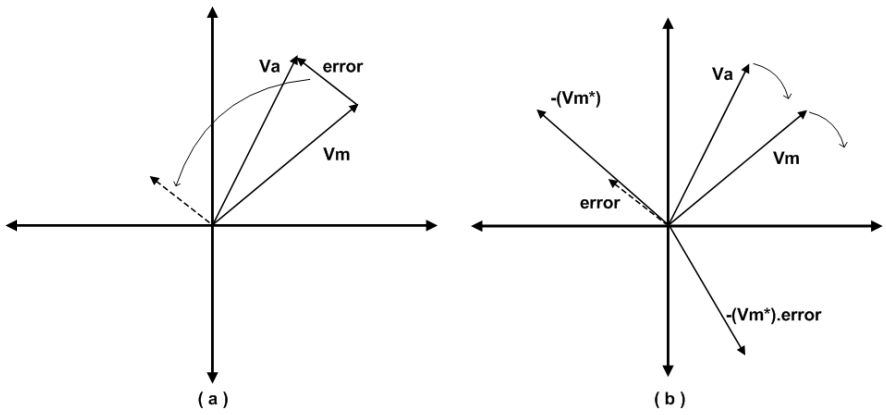

Figure 4.7: Vector Representation for Adaptation

$V_m(k)$ in the desired direction. A suitable term is $-(Vm^*)$. The explanation of multiplication with $-(Vm^*)$ is shown in Figure 4.7-b. We know that when we multiply two complex numbers we add their angles and multiply their amplitudes. And in order to move $V_a[k]$ towards $V_m[k]$, the predistorter must multiply $V_m[k]$ with a complex number such that $V_m[k]$ and $V_a[k]$ move in the direction shown in Figure 4.7-b. In this method, convergence is an issue and α parameter has to be chosen properly so that the update algorithm is stable and convergent. Choosing small α increases the convergence time while it makes the method more stable.

4.1.6 Moving-Average Filtering to LUT coefficients

After updating the LUT coefficients, MA filtering is applied. In simulations a 9-tap MA filter is used. The impulse response of the filter is given in Equation 4.11.

$$h[n] = \sum_{k=-4}^{4} \frac{1}{9}\delta[n-k] \tag{4.11}$$

This filtering is done to improve convergence time. It sets the neighboring LUT cells even if there has been no sample corresponding to those cells.

4.2 Simulations and Results

We begin simulations by first generating a 16-QAM signal in baseband as shown in Figure 4.8. In this Figure there are 3 blocks used to generate the simulation signal. The first block is Random Integer Generator generates integer from 0 to 15 and Rectangular QAM modulator maps this integer value to a complex number which is included in the constellation of the 16-QAM modulation. The output of the QAM modulator is passed through a pulse shaping filter and the output of the pulse shaping filter is the signal which is used in our simulations.

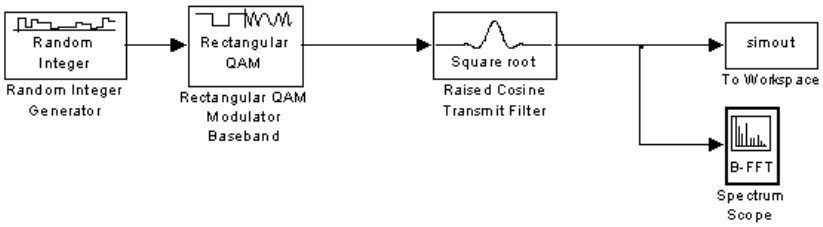

Figure 4.8: 16-QAM Signal Generation for Simulation

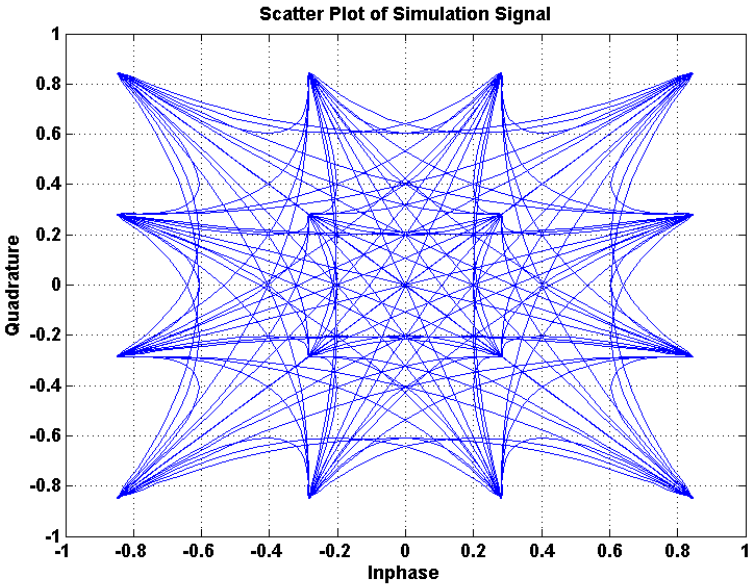

Figure 4.9: Scatter Plot of the Simulation Signal

The scatter plot of the generated signal is given in Figure 4.9 and the power

spectrum of the generated signal is given in Figure 4.10. The constellation of 16-QAM modulation is clearly seen in the scatter plot.

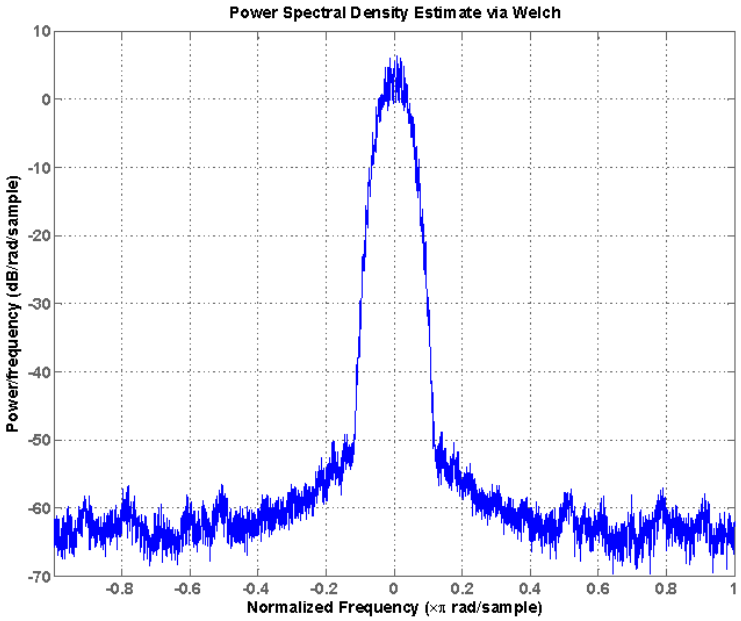

Figure 4.10: Power Spectrum of Incoming Signal,V_m

The generated signal is the input signal of the predistorter which corresponds to V_m in Figure 4.6. Before simulating the predistorter, the input signal is first passed through the power amplifier without predistortion to see the effects of the power amplifier nonlinearity on the scatter plot and the power spectrum of the output signal. The scatter plot is shown in Figure 4.11. This scatter plot shows the effect of the nonlinearity on the 16-QAM constellation. The constellation is rotated by the power amplifier. In addition, the power spectrum of the output signal is shown in Figure 4.12. The power amplifier amplifies the signal however, the distortion occurs in adjacent frequency bands. This distortion is caused by the third order nonlinearity of the power amplifier so that ACPR is 20dB worse than the incoming signal. The rotation of the constellation in the scatter plot and the distortion of the power spectrum is caused by the AM/AM and AM/PM nonlinearities of the power amplifier model used in the simulations.

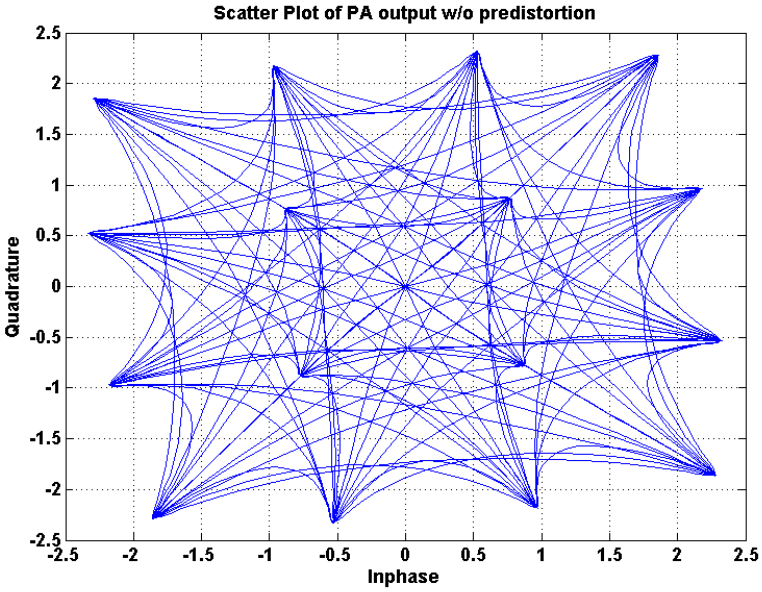

Figure 4.11: Scatter Plot of Power Amplifier Output without using Predistortion

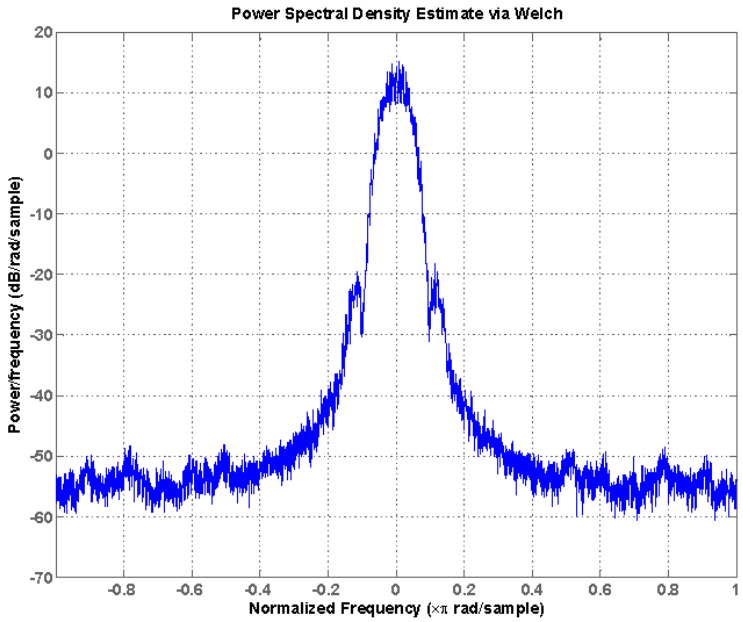

Figure 4.12: Power Spectrum of Power Amplifier Output without using Predistortion

After observing the effects of the power amplifier on the input signal, the predistorter is used to see if it compansates the nonlinearity caused by the power amplifier. In this case we used the scheme which is explained in Section 4.1.The input signal is passed through the predistorter before passing through the power amplifier. The scatter plot and power spectrum of the output signal is shown in Figure 4.14 and Figure 4.13 respectively. The scatter plot shows that predistorter compansates the rotation of the constellation and the output constellation seems to be an amplified replica of the input constellation. The power spectrum shows that the third order distortion is compansated so that ACPR is improved by nearly 20 dB. The power spectrum plot in Figure 4.15 shows the improvement in ACPR more clearly by plotting the output spectrum of the power amplifier with and without using predistortion. The 20dB improvement in ACPR is shown in this figure. The compensation of the constellation and power spectrum means that the predistorter linearizes the power amplifier as expected.

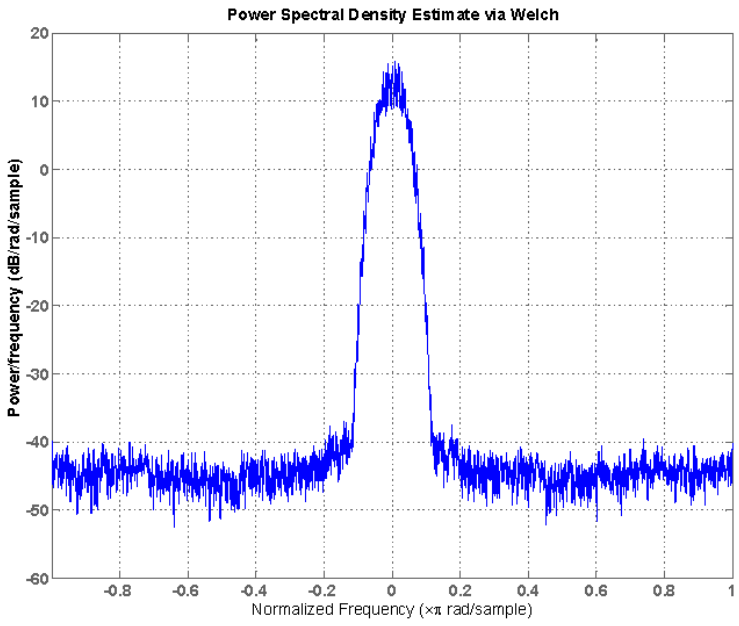

Figure 4.13: Power Spectrum of Power Amplifier Output using Predistortion,V_a

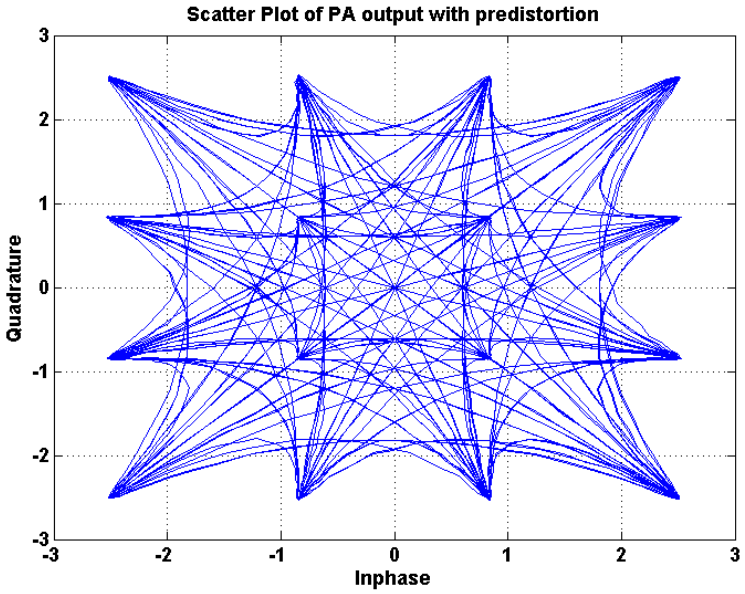

Figure 4.14: Scatter Plot of Power Amplifier Output using Predistortion,V_a

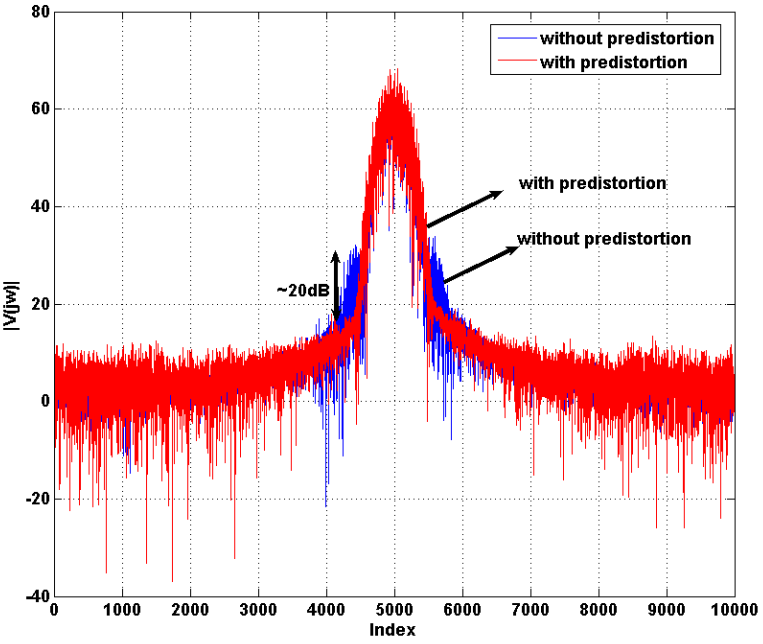

Figure 4.15: Power Spectrum of Power Amplifier Output with and without using Predistortion

In addition to observing the power spectrum and scatter plots for the predistorter signal, the LUT coefficients are observed to see their amplitudes and phase when the algorithm has converged. The amplitude of the LUT coeeficient with respect to LUT index is shown in Figure 4.16 and the phase is shown in Figure 4.17. The amplitude of the LUT coefficients is like a monotonically increasing function. The high amplitude input signals have to be amplified more by the predistorter since the gain of the power amplifier for high amplitude signals is less. So the resulting plot of the amplitude of the LUT coefficients is an expected plot. The Phase of the LUT coefficients is like a monotonically decreasing function with respect to LUT index. If we observe the AM/PM response of the power amplifier model which is given in Figure 2.6, in order to compansate the phase distortion we have to add the inverse of the hase of the power amplifier to the input signal and the plot of phase of LUT coefficients is an expected plot which compansates the phase gain and makes the cascaded system(predistorter+power amplifier) have linear phase.

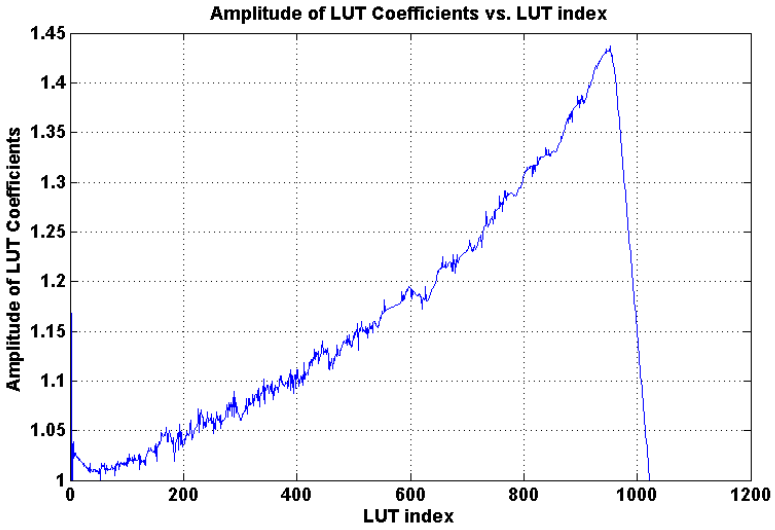

Figure 4.16: Amplitude of the LUT Coefficients

We notice that the very high indexes of the LUT is not calculated by the adaptation algorithm. The reason is the amplitude of the input signal is not so high to address the high indexes of the LUT.

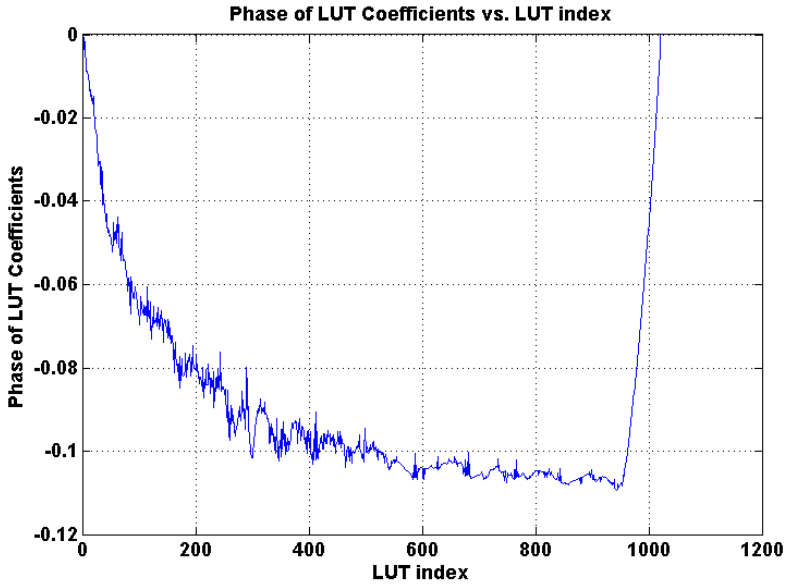

Figure 4.17: Phase of the LUT Coefficients

This New Digital Adaptive Predistortion Method has been simulated in MATLAB but it has not been verified in hardware. The simulations show us that the LUT coefficients converge after processing 1 Mega samples. It is not practical to use this type of algorithm in the software of a processor and this algorithm has to be done purely in hardware for convergence issue. Next chapter explains an easy hardware implementable adaptive predistortion scheme where a processor is used.

Chapter 5

HARDWARE IMPLEMENTABLE PREDISTORTER

In the previous chapter, we proposed a new LUT-based adaptive digital predistortion scheme. This new scheme can be used as a hardware block in communication systems. However, if we have a processor in our hardware design then the adaptation of the predistortion can be done by the software of the processor. Using this technique, no hardware resources will be used for adaptation.In addition, predistorter design with processor is more flexible and easily adaptable to other systems.

This chapter focuses on an easy hardware implementable adaptive predistortion scheme. For this purpose an IP Core in hdl(Hardware Description Language) is designed and implemented in hardware. The steps in the design of the predistorter are explained in detail.

5.1 Hardware Implementation of Predistorter

In order to realize the predistortion, the hardware implementation is necessary and we should find an effective way of implementation. The method we use is to design an IP Core in FPGA and connect it to the hard processor which will control the IP Core. The high level structure of the hardware design is shown in Figure 5.1. As we see in this Figure, the IP inputs the incoming baseband signal and the feedback IF signal and it outputs the baseband signal to be upconverted for transmission to power amplifier. The controls of the IP Core is done by the processor via the processor local bus. The IP Core has an interface with this local bus for the transaction of control and data signals between processor and the predistorter.([7])The idea to apply predistortion is to get a predetermined number of samples from incoming signal and feedback signal and then send this samples to processor via the local bus. The processor will run the predistortion algorithm and load the LUT coefficients back to the IP via the local bus. This method is an offline predistorion method but it is still adaptive. The predistortion is done in software of the processor and this makes the predistortion flexible and easily adaptable to other systems. Any predistortion scheme which uses LUT-based method can use this hardware. Since the only change in different LUT-based predistortion is the adaptation part and it is done in software for this design, the hardware for all these different predistorter methods will be the same.

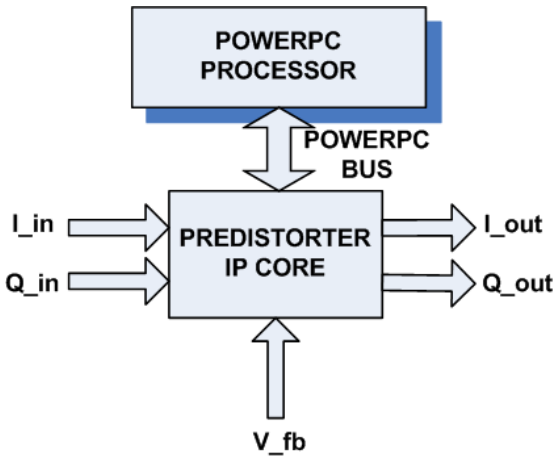

Figure 5.1: High Level Hardware Structure

The hardware system in Figure5.1 is designed as a generic IP Core for adaptation to all LUT-based memoryless predistortion systems. The only thing to be done to adapt this generic hardware is putting this IP Core before the upconversion. The baseband signals which were sent to the upconverter will be first passed through the IP Core and the outputs of the IP will be connected to the input of the upconverter. The Figure 5.2 shows the basic structure to connect the IP Core to any design which uses LUT-based method. If the system in Figure 5.2-a is to be adapted for using predistortion, the system in Figure 5.2-b can be used to adapt out IP Core to the system.

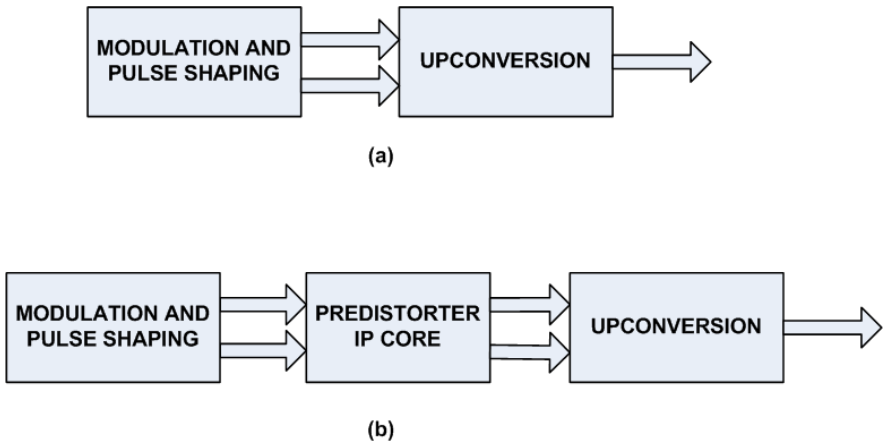

Figure 5.2: Adaptation of Predistorter IP Core - (a)Old design (b) New design

The low level hardware implementation of the predistorter and the internal hardware structure is shown in Figure 5.3. The hardware is implemented in FPGA(Virtex4-fx100) which has hard PowerPC processor core and the Figure 5.3 shows the controls of the IP Core by the PowerPC. The predistorter can be seperated into 3 subblocks and the function of each subblock is explained in the following sections.

5.1.1 Subblock1 : Predistortion

This subblock of the predistorter block includes the Address Calculation, LUT, Delay and Complex Multiplier. The Address Calculation is done by adding the

squares of the inphase and quadrature components of the incoming baseband signal. The address, generated by this block is sent to LUT to get the corresponding complex gain. Then the delayed baseband signal is multiplied by the LUT complex gain and this subblock outputs the predistorted signal. The LUT in this subblock can be updated by the PowerPC and for this reason the write port of the LUT is connected to local bus of the processor. The adaptation software will update the LUT via this write port. This subblock works real-time. For every incoming sample the Address calculation, LUT addressing, Delay element and Complex Multiplier are used to generate the corresponding baseband sample, in other words this subblock works continuously.

5.1.2 Subblock2 : Adaptation Hardware

Subblock2 in Figure 5.3, functions as a memory which samples the feedback baseband signal and the incoming baseband signal. The feedback IF signal is downconverted usind the Direct Digital Synthesizer block. The DDS (Direct Digital Synthesizer) is used to generate the sine and cosine for the IF downconversion. The output frequency of the DDS is software controllable and can be adjusted by the PowerPC according to the center frequency of the feedback IF signal.

There are two memories to save the baseband samples from the incoming signal and the feedback signal. These are Input Sample Memory and Feedback Sample Memory, respectively. The write ports of these memories are connected to the internal hardware and the read ports are connected to PowerPC. PowerPC can read the baseband samples from both memories and updates the LUT coefficients according to these samples. This subblock is used for Adaptation and it does not work continuously. The sampling operation starts with a command given by PowerPC. When the sampling command arrives, this subblock samples 4K(4096) baseband samples and waits for PowerPC to read the memories. After PowerPC reads these memories, it runs the predistortion algorithm and updates the LUT. The next command for sampling has to be started after a certain amount of time because the updated LUT will effect the output of the power amplifier after some time. So the software algorithm has to wait before sampling the baseband

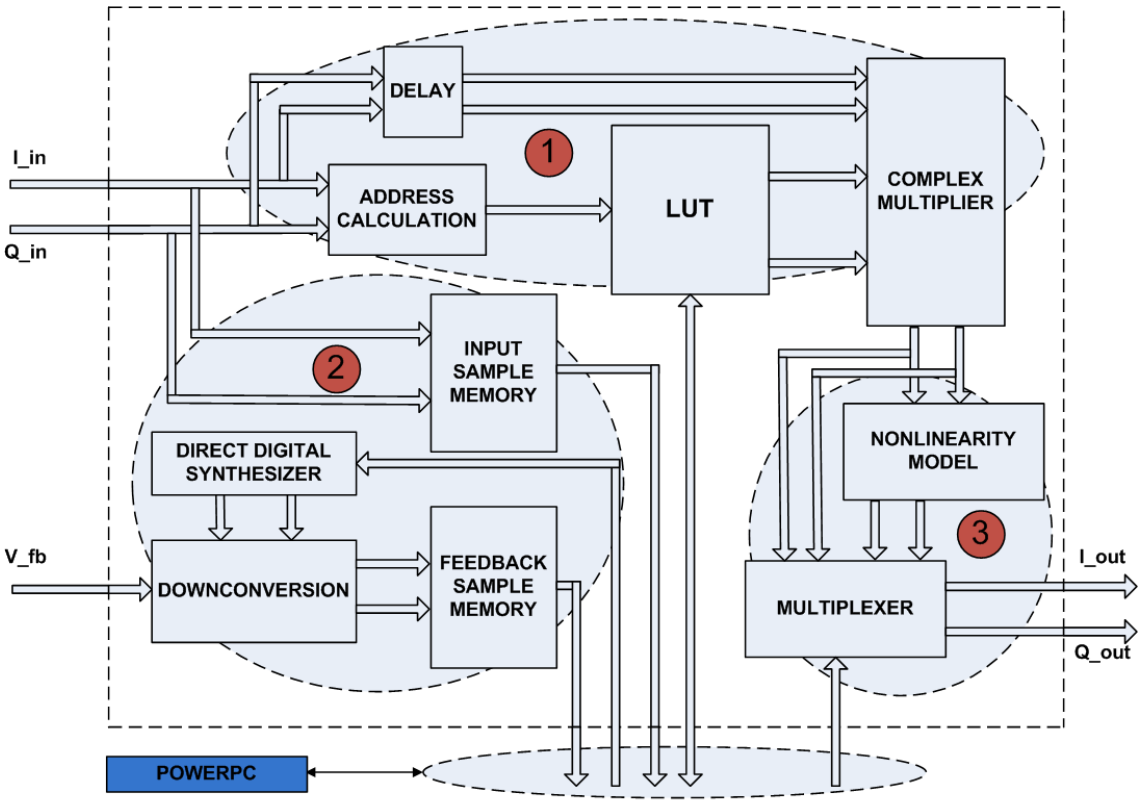

Figure 5.3: Low Level Hardware Structure

samples again.

5.1.3 Subblock3 : Hardware Simulation

Subblock3 in Figure 5.3, is used for simulation purpose. At the output stage of the predistorter, a nonlinearity model is designed and the nonlinear characteristics of this model can be determined by PowerPC. In our experiments we started our simulations by first generating a nonlinear model in hardware and connect the output of the DAC to input of the ADC. Using this loopback operation and generating the nonlinearity model in hardware we modeled the power amplifier characteristics without using the power amplifier and the RF circuits. This is

a way of simulating our predistortion algorithm in hardware. The nonlinearity model can be disabled or enabled by PowerPC by the control of the multiplexer in the output stage. For instance, the Nonlinearity model is adjusted to give third order distortion by setting its AM gain as a function of the square of index. Then the predistortion algorithm runs in software and the output of the DAC is observed to be linear replica of the input of the predistorter. The nonlinearity model consists of an Address Calculator, a LUT and a Complex Multiplier as in subblock2. The address of the nonlinear LUT is calculated by the address calculator and the output of the LUT is multiplied by the baseband signal. This is the same operation used in subblock2.

5.2 Software Development for Predistortion

After designing a generic hardware in Section 5.1, we have the opportunity of working with real data. The first thing we do is to get real samples from incoming baseband signal and process it in MATLAB using the power amplifier simulation model. The second step is to get real samples from incoming baseband signal and the feedback baseband signal and process it in MATLAB. The second step continues with writing the LUT in the hardware with tha data which is found in MATLAB software offline. The last step is to do all the software in PowerPC software. These three steps will be explained in detail in the following subsections.

5.2.1 Software Development Step1 : MATLAB Simulation with Real Data

The simulation of the experimental setup is very crucial step in our study. The setup is shown in Figure 5.4. The model consists of a predistorter, a low-pass filter and the power amplifier. The filter is modeling the band-pass filter infront of the power amplifier. Since we run our simulations in baseband, the band-pass filter is modeled as a low-pass filter considering the bandwidth of the band-pass filter.

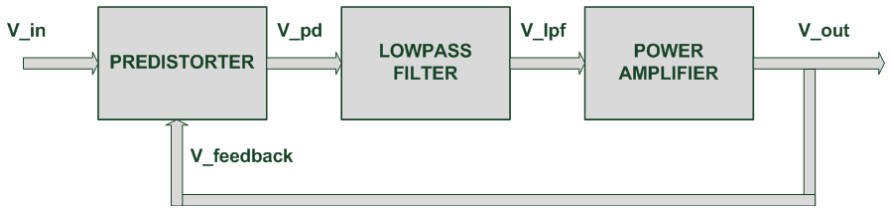

Figure 5.4: Simulation Model for Experimental Setup

The flow of the simulation is shown in flow chart in Figure 5.5.The states are numbered in the flowchart in Figure 5.5. The flow starts by State0 and in this state the LUT of the predistorter is initialized for unity gain. Initially, no nonlinearity is introduced by the predistorter. So after the flow starts, initial state sets the LUT of the predistorter to all ones. In State1, the data is passed through the predistorter but after initialization the predistorter input and output is the same. So for the first time that we arrive State1, the predistorter equates its output to its input. In State2, the output of the predistorter is filtered by the low pass filter. The filtered signal is then passed through the power amplifier, which is modeled using Saleh model, which we have used in all our simulations throughout this thesis. In State4 the LUT coefficients of the predistorter are calculated using the incoming and feedback baseband signals. In State5 the LUT of the predistorter is loaded with the calculated coefficients and then we return back to State1. For this turn, predistorter no longer has unity gain and it introduces nonlinearity to compansate the nonlinearity of the power amplifier. The states are then followed according to the flow in Figure 5.5. The key step in the flow is to calculate the coefficients of the LUT of the predistorter. The method that we used for the calculation is a new method and it is introduced in this subsection.

5.2.1.1 Calculation of the LUT Coefficients

Based on the Scheme in Figure 5.7, we derive an algorithm for calculation of the LUT coefficients. The flow of the algorithm is shown in Figure 5.7. The first step is the alignment of the incoming and the feedback data. The alignment is done by the method explained in section 4.1.5.2. In addition to the alignment in

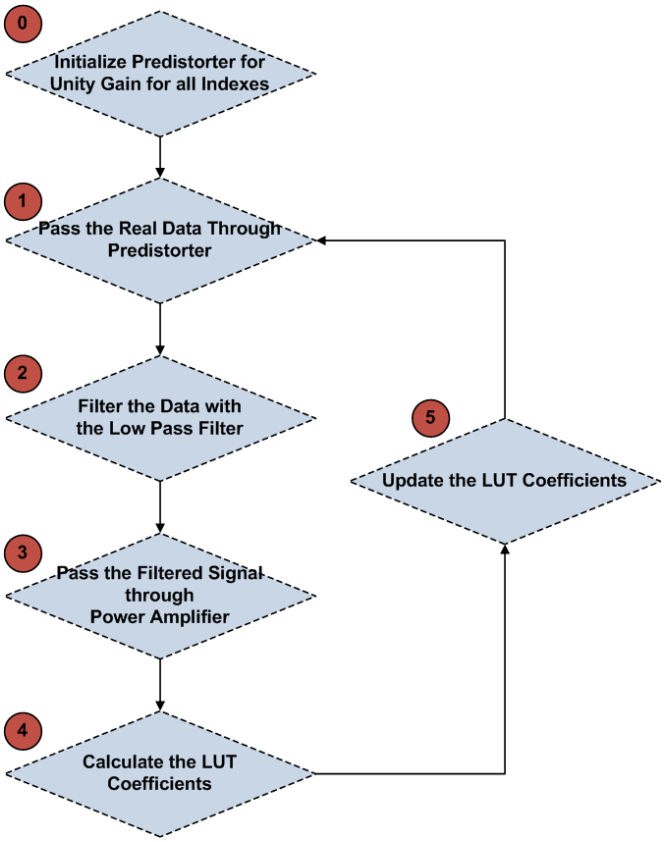

Figure 5.5: Flow for Simulation of the Experimantal Setup

this section, the analog delay alignment is also done by using the phase of the output of the matched filter and this phase information is used for the alignment of the inphase and quadrature components of the incoming and feedback signals. After aligning tha data, next step is to find the AM/AM and AM/PM responses of the power amplifier. This is done by simply dividing the baseband feedback signal to the incoming baseband signal. In other words we divide two complex numbers.The output of the complex division gives the response of the power amplifier in baseband. The equation 5.1 shows the division operation.

$$\frac{I_{in} + jQ_{in}}{I_{feedback} + jQ_{feedback}} = |A|e^{j\phi} \tag{5.1}$$

According to the Equation 5.1, $|A|$ is the amplitude of the division and ϕ is the phase of the division. The next step is to use this division result in finding the AM/AMand AM/PM responses of the power amplifier. As expected $|A|$ values of each division is arranged with respect to the amplitude-square of the incoming

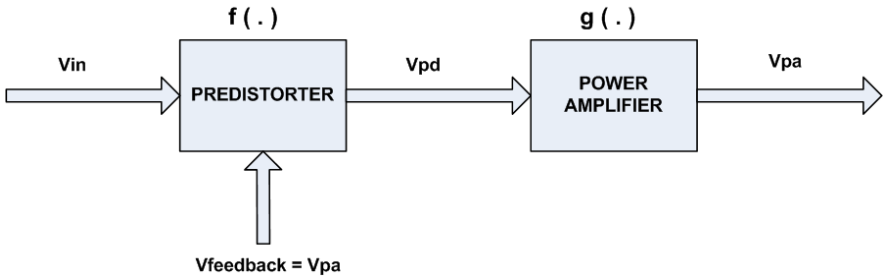

Figure 5.6: System for Calculation of LUT Coefficients

signal and for the same amplitude-square address the arranged $|A|$ values are averaged to get the AM/AM response of the power amplifier corresponding to that amplitude-squared address. The AM/PM response is also found by the same way and this time ϕ parameters of the divisions are used. The figure 5.7 shows the calculation operation step-by-step.

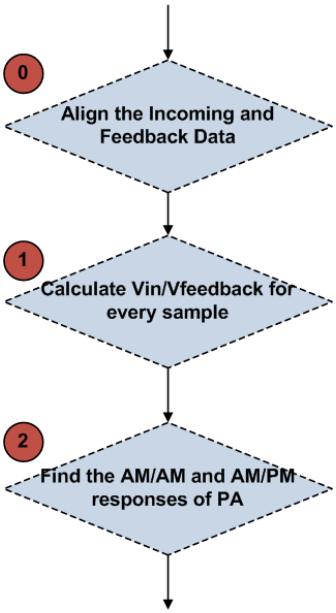

Figure 5.7: Flow for Finding Response of the Power Amplifier

After finding the response of the power amplifier, the next step is to find the coefficients for the predistorter. The following iterative steps shows the way which we used to find the LUT coefficients referring to Figure 5.6:

Step0 : Initialize $f(.)$ for unity gain for all inputs. To do this, set the complex values of the LUT of the predistorter to $1 + j0$. This operation equates V_{pd} to

V_{in}.

Then find $g_0(.)$ by the following equations :

$$V_{pa} = V_{pd}.g_0(|V_{pd}|^2) \tag{5.2}$$

$$g_0(|V_{in}|^2) = \frac{V_{pa}}{V_{pa}} \tag{5.3}$$

After finding $g_0(.)$, find $f_0(.)$ that makes $V_{pa} = V_{in}$ using the following equations :

$$V_{pd} = V_{in}.f_0(|V_{in}|^2) \tag{5.4}$$

$$V_{pa} = V_{pd}.g_0(|V_{pd}|^2) = V_{in}.f_0(|V_{in}|^2).g_0(|V_{in}|^2.|f_0(|V_{in}|^2)|^2) \tag{5.5}$$

$$f_0(|V_{in}|^2).g_0(|V_{in}|^2.|f_0(|V_{in}|^2)|^2) = 1 \tag{5.6}$$

Putting $k = |V_{in}|^2$ in Equation 5.6 we get :

$$f_0(k).g_0(k.|f_0(k)|^2) = 1 \tag{5.7}$$

And knowing $g_0(.)$ function in 5.7, we find the predistorter function $f_0(.)$. The problem of calculation of LUT coefficient reduces to finding the inverse of a function to find the predistorter LUT coefficients. The solution to this reduced problem is solved by an easy method which is found by Oguzhan Atak. He gives the name "pred" to this method and the method will be explained in detail.

Step1 : Put $f_0(.)$ to the LUT of the predistroter and collect new samples from the incoming and feedback signals.

Then find $g_1(.)$ by the following equations :

$$V_{pd} = V_{in}.f_0(|V_{in}|^2) \tag{5.8}$$

$$V_{pa} = V_{pd}.g_1(|V_{pd}|^2) \tag{5.9}$$

$$g_1(|V_{pd}|^2) = \frac{V_{pa}}{V_{pd}} \tag{5.10}$$

After finding $g_1(.)$, find $f_1(.)$ that makes $V_{pa} = V_{in}$ using the following equations :

$$f_1(k).g_1(k.|f_1(k)|^2) = 1 \tag{5.11}$$

$f_1(.)$ is found by the method in Step0. The pred method is used in all steps to find the LUT coefficients with known power amplifier response.

.

.

.

Stepi : Put $f_{i-1}(.)$ to the LUT of the predistroter and collect new samples from the incoming and feedback signals.

Then find $g_1(.)$ by the following equations :

$$V_{pd} = V_{in}.f_{i-1}(|V_{in}|^2) \tag{5.12}$$

$$V_{pa} = V_{pd}.g_i(|V_{pd}|^2) \tag{5.13}$$

$$g_i(|V_{pd}|^2) = \frac{V_{pa}}{V_{pd}} \tag{5.14}$$

After finding $g_i(.)$, find $f_i(.)$ that makes $V_{pa} = V_{in}$ using the following equations :

$$f_i(k).g_i(k.|f_i(k)|^2) = 1 \tag{5.15}$$

The "pred" method is explained in this part of this section. The "pred" method is an easy and practical method to solve the equation $f(k).g(k.|f(k)|^2) = 1$. It also uses the fact that the amplitude of $g(k)$ function is a monotonically decreasing function. The method can be shown by the following states :

State0 : Assign $f(k)$ to $\frac{1}{g(k)}$ for all indexes. Set the initial index to zero, i=0. go to State1

State1 : Equate a temporary variable j, to current index i. go to State2

State2 : Calculate $result = f(j).g(j.|f(j)|^2)$. if $result < 1$ then go to State3 else go to State4

State3 : Decrease j. Assign $f(j) =$ to $\frac{1}{g(j)}$. go to State2

State4 : if $i < (maximumLUTaddress)$ go to State1 else Finish

The method first assigns $f(k)$ to $\frac{1}{g(k)}$ and looks for $f(k).g(k.|f(k)|^2) = 1$. If it does not find the solution for that index,it decreases the index and tries the equation again. It tries this operation until finding the solution for all indexes. The states of the algorithm is shown above. The results of the studies will be shown in next section.

5.2.1.2 Results of the Simulations

This section shows the results of the studies in the previous section. The spectrum of the real data to the predistorter is shown in Figure 5.8. Referring to Figure 5.4, Figure 5.8 shown the spectrum of V_{in}.

The filtered signal V_{lpf} has a spectrum as shown in Figure 5.9. When the algorithm starts, the predistorter is adjusted for unity gain and V_{pd} is equal to V_{in}. So the output of the low-pass filter is the filtered version of V_{in}. This signal goes to the baseband power amplifier model and the output spectrum of the power amplifier is shown in Figure 5.10. As you see, the spectrum of the PA output shows the nonlinearity affects in the adjacent bands. The ACPR is reduced by more than 10 dB at the output of the power amplifier.

Applying our studies on the linearization of the power amplifier in Section 5.2.1.1, we find power amplifier response and the LUT coefficients for the predistorter. The AM/AM response of the power amplifier is shown in Figure 5.11 and the AM/PM response is shown in Figure 5.12. As we see the amplitude gain

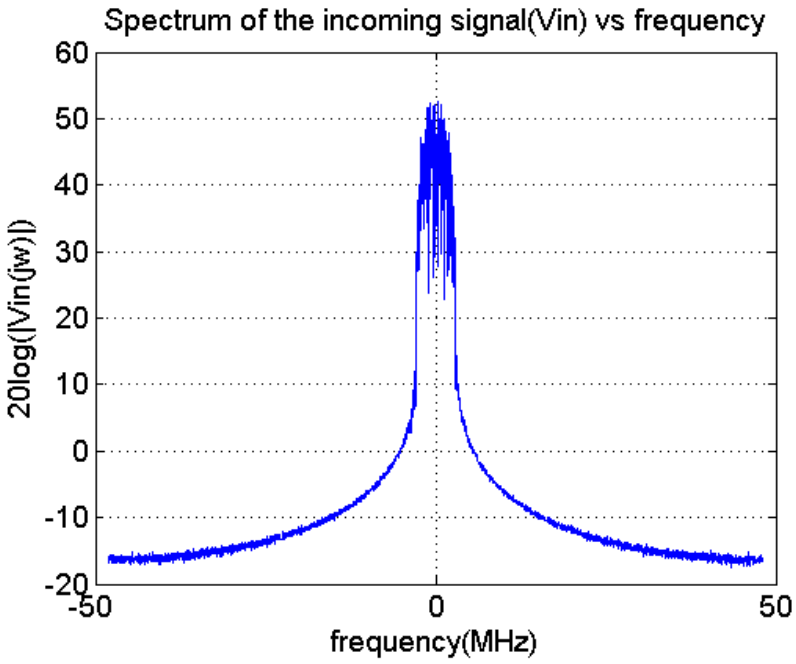

Figure 5.8: Power Spectrum of Incoming Signal

decreases as the input amplitude increases.

The phase of the power amplifier has a nonlinear behaviour. After finding the power amplifier response, our method finds the corresponding LUT coefficients for the predistorter. The predistorter has also AM/AM and AM/PM responses and they are shown in Figure 5.13 and Figure 5.14 respectively. As we see in Figure 5.13, the amplitude of the predistorter response is high at high amplitudes and low at low amplitudes. This is an expected result since the predistorter has to compansate the affects of the power amplifier for high amplitudes, where the power amplifier is nonlinear. The phase of the predistorter is also found and shown in Figure 5.13, and we see that the phase is nearly the inverse of the phase of the power amplifier to have a linear phase system in cascade. The predistorter adapts itself in a way that the cascade response of the predistorter and the power amplifier is linear. A low pass-filter is put between the predistorter and the power amplifier. The low-pass filter is put into this design to see the filtering effects and to model the hardware closer to the real circuit.

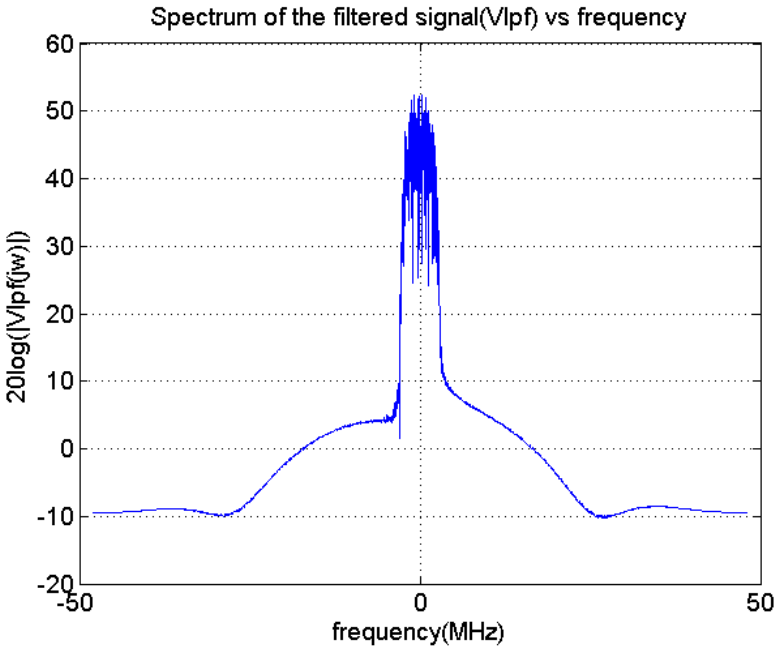

Figure 5.9: Power Spectrum of Filtered Signal

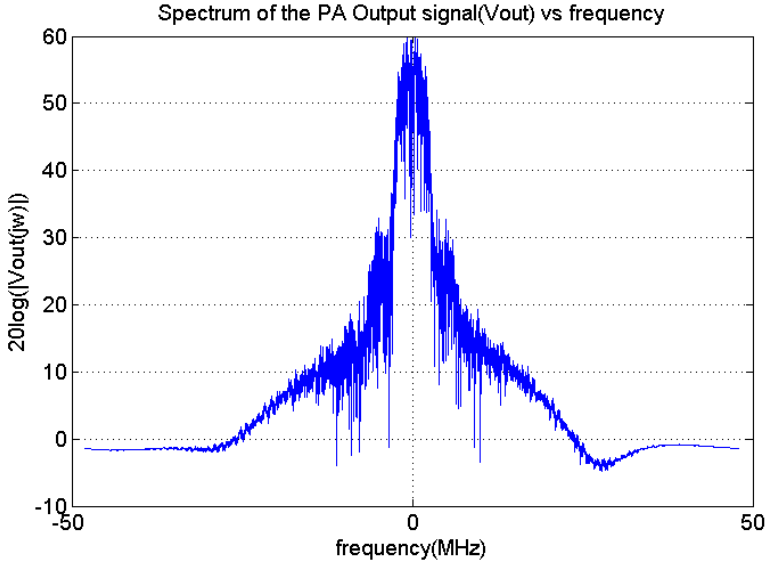

Figure 5.10: Power Spectrum of PA Output Without Using Predistortion

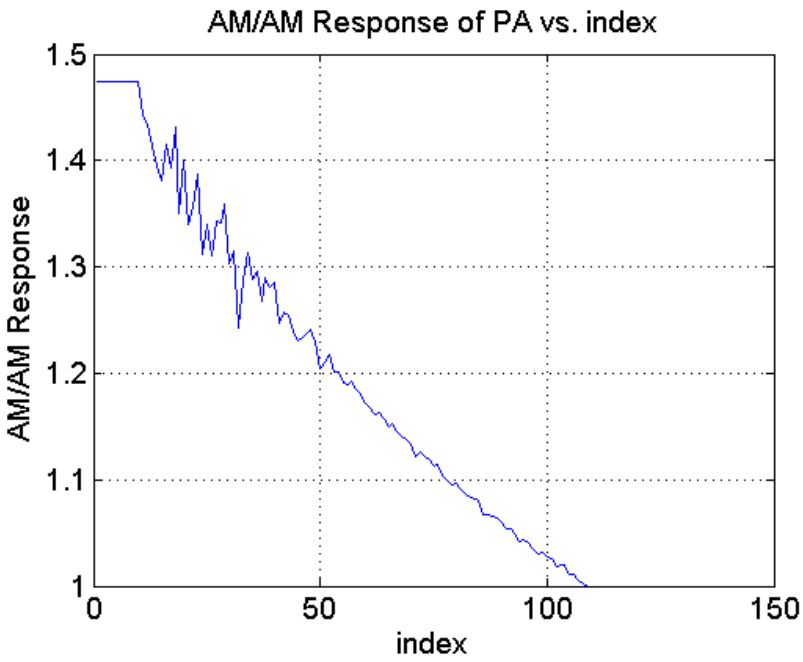

Figure 5.11: AM/AM response of the Power Amplifier

After finding the LUT coefficients, we put coefficients to the LUT of the predistorter and the observed if the power amplifier output is linearized or not. The plot of the power amplifier output spectrum when using predistortion is shown in Figure 5.15. Comparing this Figure with the spectrum in Figure 5.10, we see that the ACPR is improved by nearly 10 dB and that means the predistortion linearizes the power amplifier and the predistortion algorithm works.

5.2.2 Software Development Step2 : MATLAB-Hardware Cosimulation with Real Data

This section focuses on the software development after the completion of the MATLAB simulation with real data. In this step, the incoming data to the predistorter and the feedback data from the power amplifier is collected and the LUT coefficients are calculated in MATLAB and then the calculated coefficients are embedded in LUT hardware and the results have been observed.

The Figures shown in Figure 5.16 and Figure 5.17, are the screenshots of the

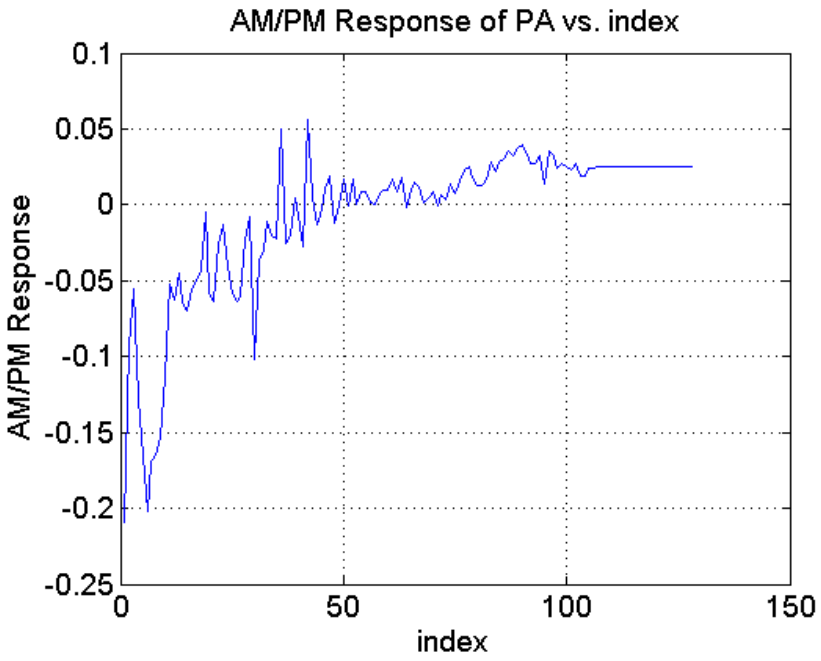

Figure 5.12: AM/AM response of the Power Amplifier

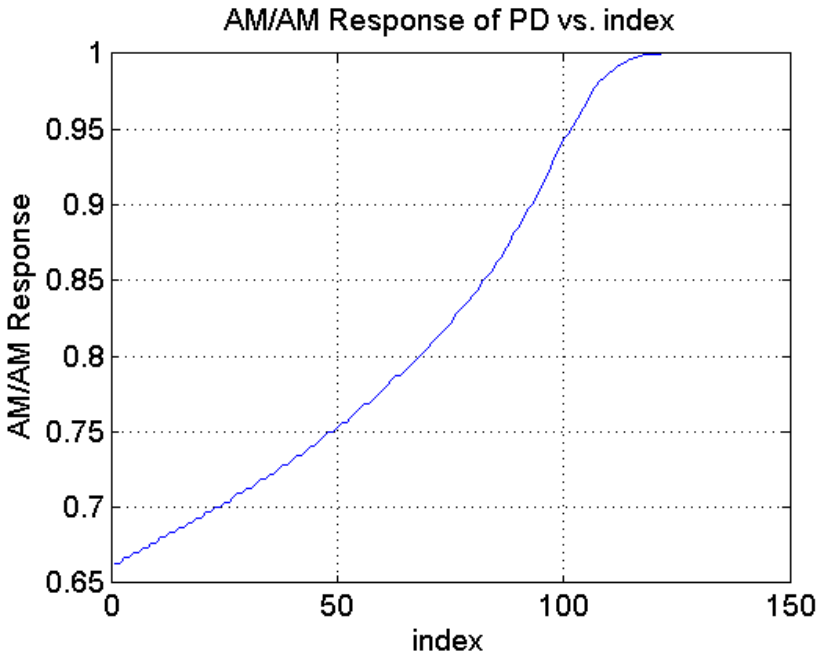

Figure 5.13: AM/AM response of the Predistorter

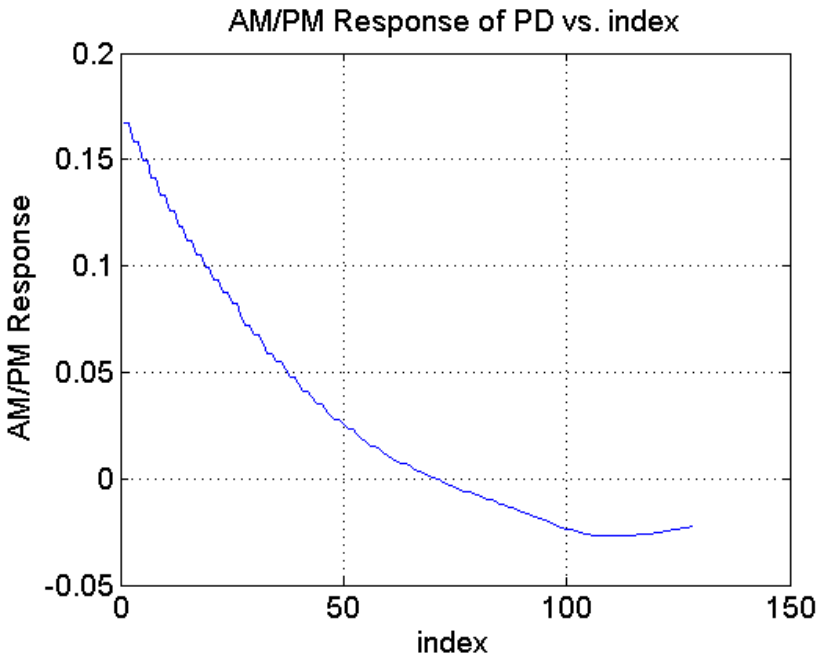

Figure 5.14: AM/AM response of the Predistorter

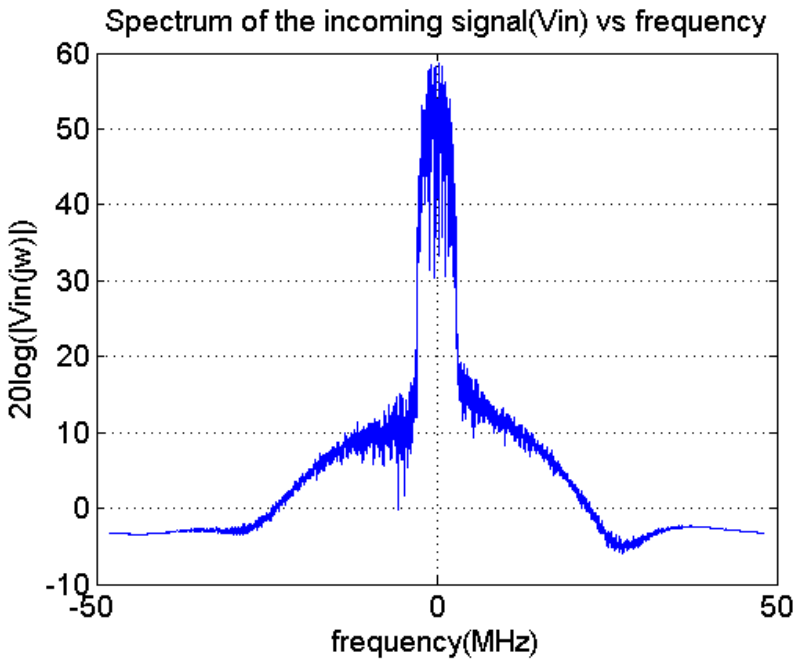

Figure 5.15: Power Spectrum of PA Output Using Predistortion

spectrum analyzer. The plot in Figure 5.16 shows the spectrum of the power amplifier output with and without using the predistortion at the same time when the input power is low. The nonlinear one, is the one which we get without predistortion and when predistortion is applied, nearly 10 dB ACPR improvement can be seen. The plot in Figure 5.17 shows the spectrum of the power amplifier output with and without using the predistortion at the same time when the input power is high. The high input amplitude leads to more distortion. By applying predistortion, we get more than 20 dB improvement in ACPR.

5.2.3 Software Development Step3 : POWERPC Software Development

In this step, the software in MATLAB is written in POWERPC software and predistortion algorithm and its results are observed for different LUT table sizes. The results are shown in Figure 5.18,5.19 and 5.20 for LUT sizes of 32,64 and 128 respectively. As we see from these figures, the optimum performance is achieved with a LUT size of 64 with ACPR improvement by nearly 15 dB. Using LUT sizes of 32 and 128 we get nearly 10 dB improvement in ACPR.

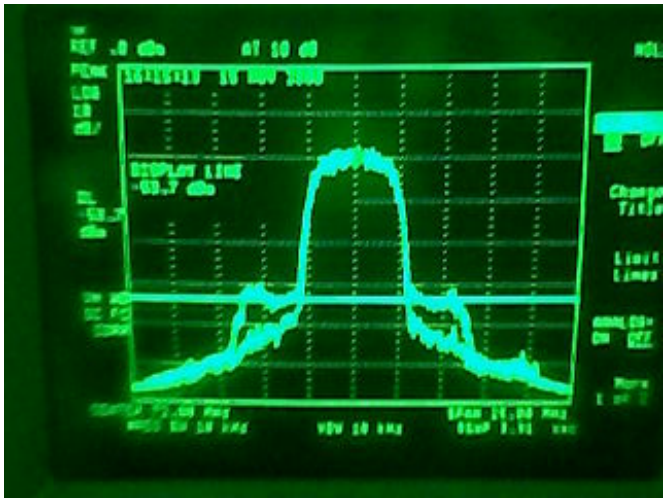

Figure 5.16: Spectrum of PA Output Using Predistortion at Low Input Power

Figure 5.17: Spectrum of PA Output Using Predistortion at High Input Power

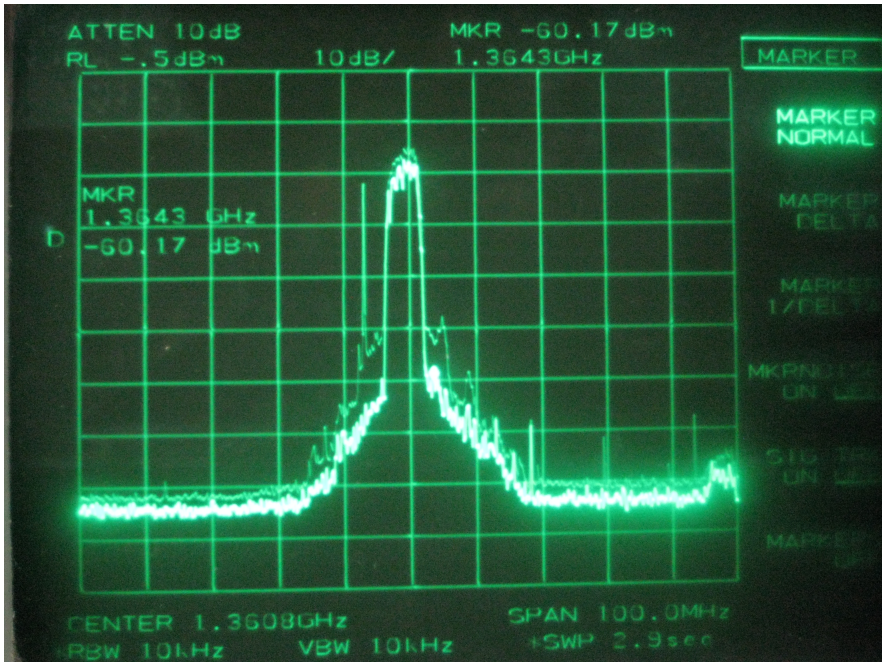

Figure 5.18: Predistortion with LUT size = 32

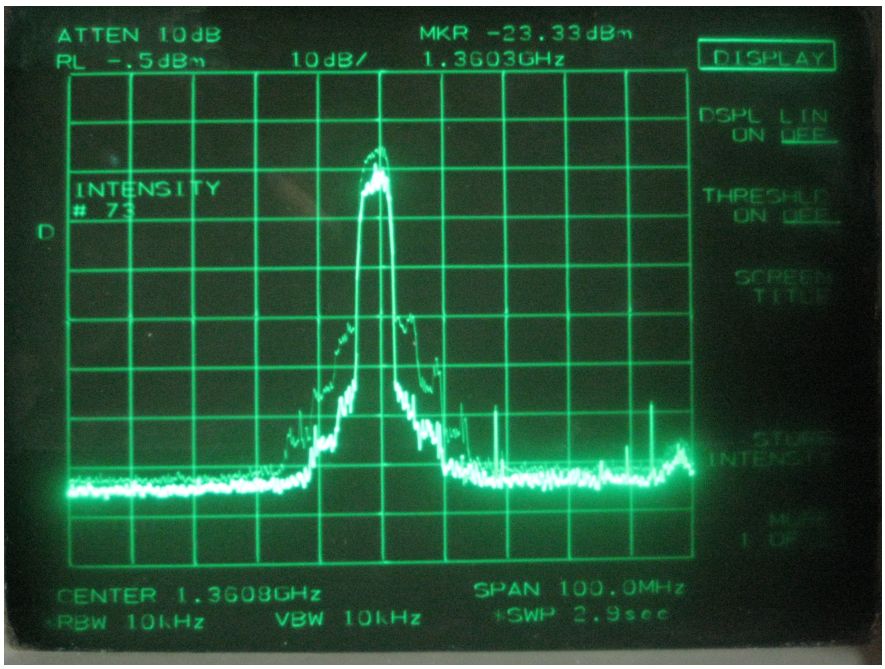

Figure 5.19: Predistortion with LUT size = 64

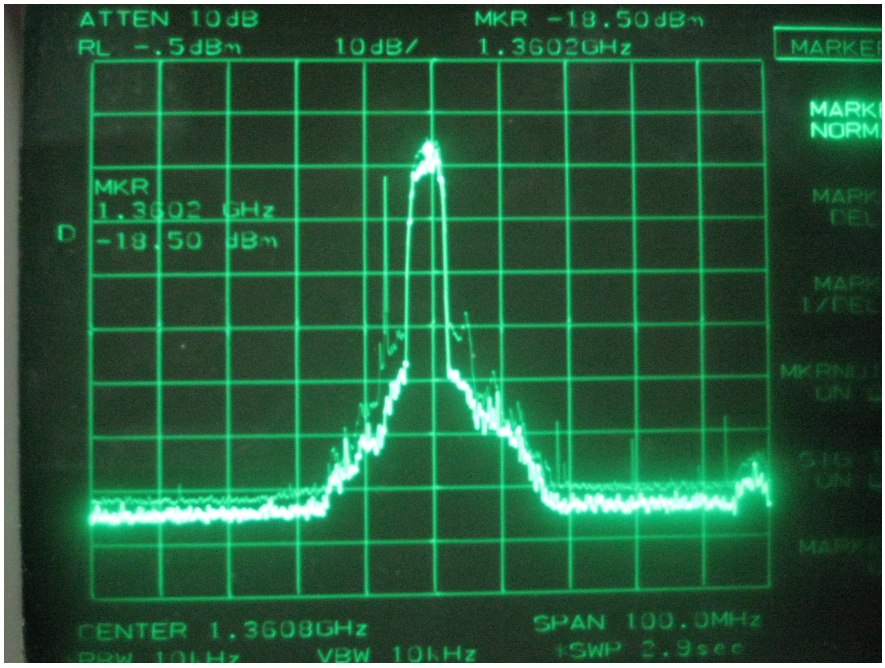

Figure 5.20: Predistortion with LUT size = 128

Chapter 6

CONCLUSION AND FUTURE WORK

The thesis begins with an introduction about the nonlinearity concept and power amplifier characteristics. The commonly used predistortion schemes are explained in detail. The invention of a new method for adaptation of the predistorter is shown with the block diagram. The Matlab simulation results are observed. The hardware implementation is then shown in detail and the invention of the new hardware implementable predistortion method is proposed. The experimental results shows the hardware design gives the best performance with a table size of 64 and using this table size we have nearly 15 dB improvement in ACPR.

In our experiments, one of the important points is the bandwidth. The bandwidth of the power amplifier input has to be high enough for the predistorted signal to be passed to the power amplifier. For instance, to linearize the power amplifier for an input signal of bandwidth 20 MHz, a 40 MHz predistorter signal may be needed. If there is a filter before the power amplifier, the filter has to be designed considering this issue.

As a future work, the hardware design can be developed to be adapted to systems which use power amplifiers with memory. For this purpose, a new hardware implementation is required such that its output depends on both the current

baseband sample and previous baseband sample. This can be done by adding one more stage to the hardware design in Figure 5.3. The stage to be added is a copy of the first stage with a different LUT coefficient set. The adaptation can still be done by the software in POWERPC. The software has to be modified to adapt the predistorter with memory.

In Addition, the algorithm described in Chapter 4 can be realized in hardware. For this purpose a hardware logic has to be implemented which is independent of a processor. This will require more resources but in designs where processors are not used, this hardware logic is indispensible.

Bibliography

[1] S. Meza, M. ODroma 2, Y. Lei, A. Goacher. Some New Memoryless Behavioural Models of Wireless Transmitter Solid State Power Amplifiers

[2] J. K. Cavers. Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements, IEEE Transactions on Vehicular Technology, vol. 39, pp. 374382, Nov. 1990.

[3] K. J. Muhonen, R. Krishnamoorthy, and M. Kavehrad. Look-up table techniques for adaptive digital predistortion, A development and comparison, IEEE Transactions on Vehicular Technology, vol. 49, pp. 19952002, Feb. 2000.

[4] S. P. Stapleton, F. C. Costescu. An adaptive predistorter for a power amplifier based on adjacent channel emission, IEEE Transactions on Vehicular Technology, vol. 41, pp. 4956, Feb. 1992.

[5] G. Baudoin, P. Jardin. Adaptive polynomial pre-distortion for linearization of power amplifiers in wireless communications and WLAN, in Proc. of the International Conference on Trends in Communications, Jul. 2001, pp. 157160.

[6] J. K. Cavers. Amplifier linearization by Adaptive Predistortion, United States Patent, Patent Number : 5049832, 1991.

[7] PowerPC 405 Processor Block Reference Guide, Embedded Development Kit. http://www.xilinx.com/support/documentation/userguides/ug018.pdf.