POLITECNICO DI TORINO

SCUOLA DI DOTTORATO
Dottorato in Dispositivi elettronici – XXVIII

Tesi di Dottorato

Behavioral modeling and FPGA implementation of digital predistortion for RF and microwave power amplifiers

Jiang Tao

Tutore
Prof. Marco Pirola
Prof. Vittorio Camarchia

Coordinatore

Prof. Giovanni Ghione

March 2016

Abstract

With the high interest in digital modulation techniques which are very sensitive to the PA nonlinearity, modern wireless communication systems require the usage of linearization techniques to improve the linear behavior of the RF power amplifier. The powerful and cheap digital processing technology makes the digital predistortion (DPD) a competitive candidate for the linearization of the PA. This thesis introduces the basic principle of DPD, its implementation on FPGA and the adaptive DPD system.

The linearization of 4 PAs with DPD technique has been introduced: for the hybrid class AB PA operating at 2.6 GHz with a WiMAX testing signal, 33.7 dBm average power, 29.6 % drain efficiency, 13 dB ACPR and 9 dB NMSE improvement have been obtained; for the hybrid Doherty PA operating at 3.4 GHz with an I/Q testing signal, 35.0 dBm average power, 36.8 % drain efficiency, 12 dB ACPR and 13 dB NMSE improvement have been obtained; for the MMIC class AB PA operating at 7 GHz with an I/Q testing signal, 29.4 dBm average power, 25.7 % drain efficiency, 12 dB ACPR and 12 dB NMSE improvement have been obtained; for the two-stage PA operating at 24 GHz with an I/Q testing signal, 23.5 dBm average power, more than 14.0 % drain efficiency, 11 dB ACPR and 11 dB NMSE improvement have been obtained.

The DPD algorithm has been implemented on FPGA with two methods based on LUT and a direct structure with only adders and multipliers. The block RAM on the FPGA board is chosen as the table in the LUT methods. The linearization performance for these three methods is similar. The test PA is the hybrid Doherty PA mentioned above and the test signal is the I/Q signal with 7.4 dB PAPR. 35.1 dBm average power, 36.8 % efficiency, 11 dB ACPR and 11 dB NMSE improvement have been obtained. The cost of logic resources for the direct structure method is the largest with 1,172 flip-flops, while the number of flip-flops for the two LUT methods is 263 and 583.

A new adaptive algorithm has been proposed in this thesis for the adaptive DPD

system. This new algorithm improves the performance in extracting the model parameters in complex number domain. With the experimental data from a combined class AB PA, the final accuracy of the model extracted by the new algorithm has been improved from -20 dB to about -40 dB and the converge speed is faster.

Acknowledgments

I want to thank Prof. Marco Pirola and Prof. Vittorio Camarchia, my tutors, for all their help and guidance throughout my PhD carrier. It is a great pleasure working with such knowledgeable and experienced professors. I am grateful to have an opportunity to learn from them.

Roberto Quaglia, a research assistant in this lab group, provided me huge assistance during my work. I am very grateful for his willingness to always give me a help whenever I need.

To all my friends at Politecnico di Torino, thanks for making here a great place to study.

I would like to thank my parents for their love and support. Without you I would not be where I am today.

Abbreviation

ACPR	Adjancent Channel Power Ratio
ADC	Analog to Digital
CLB	Configurable Logic Block
DAC	Digital to Analog
DSP	Digital Signal Processing
DUT	Device Under Test
ETSI	European Telecommunications Standards Institute
FPGA	Field Programmable Gate Array
GaN	Gallium Nitride
GPIB	General Purpose Interface Bus
HEMT	High Electron Mobility Transistor
IMD	Intermodulation Distortion
LMS	Least Mean Square
LUT	Look Up Table
LS	Least Square
MMIC	Monolithic Microwave Integrated Circuit
NMSE	Normalized Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
PA	Power Amplifier
PAE	Power Added Efficiency
PAPR	Peak to Average Power Ratio
PCI	Peripheral Component Interconnect
QAM	Quadrature Amplitude Modulation
RFPA	Radio Frequency Power Amplifier
RLS	Recursive Least Square
RTL	Register Transfer Level

Table of contents

Chapter 1

Power Amplifier Overview

In a typical wireless transmitter system, the radio frequency power amplifier (RF PA) is one of the most important components. A RF PA is a power device which can convert DC supply into RF power. It is used to amplify the signal to a significant power before transmitting it to the antenna. In the early days, RF power was generated by spark, arc and alternator techniques. From the late 1920s, vacuum tube transmitters were dominant for several decades. At the end of the 1960s, discrete solid state RF power devices began to appear. Since then, variety of new solid state RF power devices, such as MOSFET, HEMT, HFET, HBT, became mature and were used in industrial applications. These new devices have extended the working bandwidth and have made high output power level available.

Linearity, power efficiency and bandwidth are the three figures of merit that need to be considered when we design the PA for the modern wireless transmission applications. The original intention of PAs is to generate higher power, therefore, the output power level, as well as the power gain, are the PA's primary performance. Moreover, to limit the power consumption, efficiency is also essential to be high. Power efficiency is directly associated with the cost of the communication infrastructure.

On the other hand, high linearity is required to minimize the distortions caused by the PA. Less distortion means higher integrity of the signal during the transmission process. However, the linearity has an inherent conflict with the efficiency. PA with high linearity, like Class A PA, presents poor efficiency performance. Moreover, efficient PA architectures, such as Doherty, envelope tracking, have severe linearity problem. Therefore, linearity improvement techniques are needed for these high efficient PAs to guarantee the linearity.

1.1   Power Gain and Efficiency

In general, a power amplifier is defined as a device that increases the power of an arbitrary input signal. The added power is transferred to the output signal and is taken from a DC supply [2]. Ideally, all supplied power should be converted into RF power. However, it is not the case in reality and only part of the supplied power can be converted into the output power. Power efficiency is introduced to indicated how much of the supplied power is converted. The efficiency can be described in two ways: drain efficiency and power added efficiency [3]. Drain efficiency is more general and is defined as

$$\eta_d = \frac{P_{out}}{P_{DC}} \tag{1.1}$$

where P_{out} is the power delivered to the load and P_{DC} is the power taken from the supply. The drain efficiency does not take the input power into consideration. On the other hand, the power added efficiency (PAE) regards the input power as a power loss which should be subtracted from the output power. Therefor, the PAE is defined as

$$PAE = \frac{P_{out} - P_{in}}{P_{DC}} \tag{1.2}$$

where P_{in} is the input power. Power added efficiency describes the performance of the PA more accurately by including also the input power. It is also a function of the power gain as show in equation (1.3). PAE is a reasonable description of PA's performance when the power gain is high. However, it could be negative for a PA with a low power gain.

$$PAE = \frac{P_{out} - P_{out}/G}{P_{DC}} \tag{1.3}$$

where $G = P_{out}/P_{in}$ is the power gain of the power amplifier. The power gain of a two-port device, for instance a power amplifier, is the ratio of the output power to the input power. It is usually represented in decibel as

$$G_{dB} = 10 \; log_{10}(\frac{P_{out}}{P_{in}}) \tag{1.4}$$

In the meanwhile, the power itself can also be represented in decibel form, called dBm, which is ratio between the power and 1 mW:

$$P_{dBm} = 10 \; log_{10}(\frac{P}{1mW}) \tag{1.5}$$

Typically, the efficiency of a PA, both drain efficicy and PAE, reaches its maximum

value at saturated power and decreases rapidly as the power decreases. However, at high power region, the PA faces a linear problem. The nonlinearity will be discussed in next section.

1.2 Power Amplifier Nonlinearity

In common sense, the word linearity describes a mathematical abstraction or relationship, $f(x)$, which should satisfy the following two properties: homogeneity and superposition.

$$\textit{Homogeneity}: \ f(\alpha x) = \alpha f(x)$$
$$\textit{Superposition}: \ f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

where α, β are two constant numbers. If two quantities have a linear relation, they will be proportional to each other, called homogeneity. Therefore, if a power amplifier is said to be linear, the output power level should have a constant gain over each input power level and the characteristic of the PA should be a straight line. However, it is not a straight line for a real PA and there is compression at high input power level, as shown in Figure 1.1(b) the solid blue line. The nonlinear distortion of a power amplifier is caused mainly due to the output power limitation. Therefore, the nonlinear behavior is more obvious when the output power tends to the saturation.
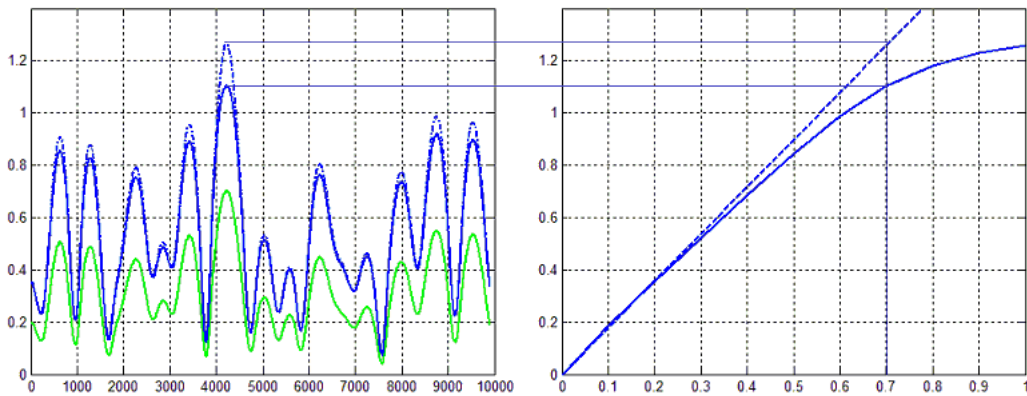

Figure 1.1.   (a) Amplitude as a function of time, the green line is the input amplitude, the solid blue line is the real output amplitude and the dashed blue line is the ideal linear output amplitude. (b) AM/AM characteristic of a power amplifier, dashed line is ideal linear case and the solid line is the real case

3

Figure 1.1 depicts clearly the nonlinearity of a PA in time domain. In Figure 1.1(a), the green line is the input signal amplitude in time domain. If this signal passes through a linear PA with constant power gain, the output will be the dashed blue line. However, in real case, the output is the solid blue line. We can see that at lower input power level the dashed and solid blue lines coincide, meaning that the PA works with linear behavior. In the meanwhile, at some top peaks of the wave (the highest point in Figure 1.1(a) as an example), where the input power level is high, the solid line is lower than the dashed line, meaning that the PA is with compression and is not linear with high input power.

1.2.1 Power Amplifier Classification

The class of operation of a PA has a significant effect on the linearity of the PA. Class A amplifiers are the most linear while Class AB amplifiers are mildly nonlinear and Class C, D ,E amplifiers have highly nonlinear behavior.

A Class A PA is defined as a PA in which output current flows for the full-cycle (360°) of the input signal. In other words, the transistor remains forward biased during the whole input cycle. The class A amplifier is the most common and simplest form of PA that uses the switching transistor in the standard common emitter circuit configuration. The transistor stays always in "ON" state so that the current flows throughout one complete cycle of the input signal, producing minimum distortion and maximum amplitude to the output.

A class B amplifier has zero DC bias so that the amplifier is not in its active region when the input signal is in the range near to 0. In this way, class B amplifier has a significant advantage over class A amplifier in that no current flows through the transistor when there is no input signal at Base. Therefore, no power is dissipated in the transistor when there is no signal present while class A amplifier stage dissipates lots of heat even without input signal. However, there is a critical drawback of class B amplifiers which is that they suffer from an effect commonly known as *Crossover Distortion* [3].

The class AB amplifier is a compromise between the class A amplifier and the class B amplifier. It uses two diodes between Base and Emitter to build a voltage difference or bias voltage. The class AB amplifier attempts to solve the distortion of the crossover region that the class B amplifier exhibits at a cost of efficiency decrease. Any input signal will make the transistor to operate normally in its active region, minimizing or even eliminating the crossover distortion which is present in class B

4

configurations. But the efficiency of class AB amplifier is smaller than that of class B, with a value between that of class A and class B amplifiers.

In a class C amplifier configuration, the transistor being biased with negative voltage, the working point is far beyond the cut-off point. So it operates for less than half of the input cycle and the resulting output signal is strongly non-linear. The efficiency of a class C amplifier is very high, reaching ideally 100% in best cases, however it generates strong crossover distortions too.
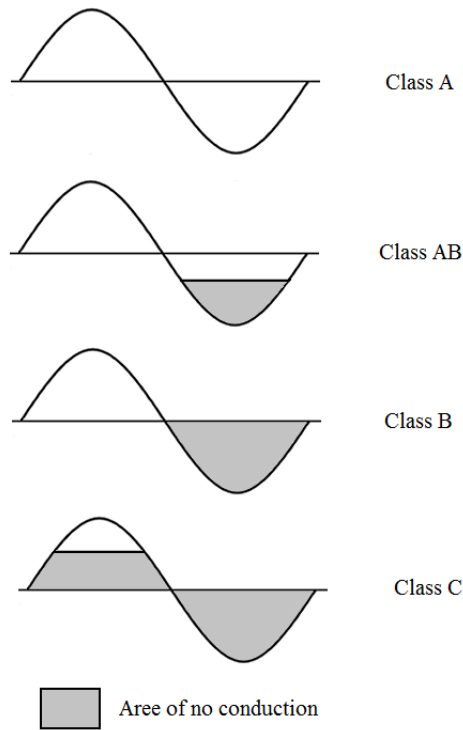
Figure 1.2.   Comparison of output signals of different amplifier classes of operation

Figure 1.2 clearly illustrate the output signals for different classes of operation. The gray area is where no signal is conducted.

Class D amplifiers [1] are switching mode amplifiers. They have at least two transistors operating as switches to create a pulse width modulation (PWM) signal at the output of the transistors. The generated signal is then low pass filtered passing the first harmonic to the load. As the current is flowing just through the switched-on active elements, the theoretical efficiency is 100% if the active devices are ideal.

Nevertheless, ideal transistors are not existed and there is also power loss due to saturation, switching speed and junction capacitance. The real efficiency is less than 100%.

Class of operation	Efficiency(%)	Linearity
A	50	Good
AB	Between Class A and Class B	Between Class A and Class B
B	78.5	Moderate
C	100	Poor
D	100	Poor

Table 1.1. Comparison of different class of operation in terms of efficiency and linearity

Table 1.1 summarizes the linearity situation of different classes of operation for a PA. The linearity varies with different amplifier classed.

1.2.2 Distortion due to nonlinearity

Amplitude distortion

The nonlinear behavior of a PA can be modeled as a power series:

$$v_o = a_1 v_i + a_2 v_i^2 + a_3 v_i^3 \cdots \tag{1.6}$$

where a_1, a_2, a_3 are coefficients and v_i, v_o are the input and output of the power amplifier, respectively. When the input signal has one single frequency, $v_i = V cos(\omega t)$, the v_o will be

$$
\begin{aligned}
v_o &= a_1 V cos(\omega t) + a_2 V^2 cos^2(\omega t) + a_3 V^3 cos^3(\omega t) + \cdots \\
&= \frac{1}{2} a_2 V^2 + (a_1 V + \frac{3}{4} a_3 V^3) cos(\omega t) \\
&+ \frac{1}{2} a_2 V^2 cos(2\omega t) + \frac{1}{4} a_3 V^3 cos(3\omega t) + \cdots
\end{aligned}
\tag{1.7}
$$

In (7.2), the output signal consists not only of the signal with the input signal frequency but also of a DC component and the harmonic components. Even with a ideal filter that can cancel the DC and harmonic parts, the amplifier gain is no longer the linear gain a_1, but is $a_1 + \frac{3}{4} a_3 V^2$. This new gain is no longer constant but is a function of input signal magnitude V. If $a_3 > 0$, the amplifier will be with

gain expansion. In general, power amplifier has the feature of gain compression with $a_3 < 0$.

Intermodulation distortion

The intermodulation distortion is explained with a two-tone signal which is represented by:

$$v_i = V_1 cos(\omega_1 t) + V_2 cos(\omega_2 t) \tag{1.8}$$

where V_1, V_2 are the magnitudes of the two tones, and ω_1, ω_2 are their angle frequencies. The PA's nonlinearity is still modeled by a third order power series, expressed as:

$$v_o = a_1 v_i + a_2 v_i^2 + a_3 v_i^3 \cdots \tag{1.9}$$

Therefore, the output of the PA as a function of input signal can be obtained by substituting (1.8) in (1.9):

$$
\begin{aligned}
v_o = {} & \frac{1}{2} a_2 V_1^2 + \frac{1}{2} a_2 V_2^2 \\
& + \left[\left(a_1 + \frac{3}{4} a_3 V_1^2 + \frac{3}{4} a_3 V_2^2 \right) V_1 cos(\omega_1 t) + \left(a_1 + \frac{3}{4} a_3 V_2^2 + \frac{3}{4} a_3 V_1^2 \right) V_2 cos(\omega_2 t) \right] \\
& + \left[\frac{1}{2} a_2 V_1^2 cos(2\omega_1 t) + \frac{1}{2} a_2 V_2^2 cos(2\omega_2 t) \right] + \left[\frac{1}{4} a_3 V_1^3 cos(3\omega_1 t) + \frac{1}{4} a_3 V_2^3 cos(3\omega_2 t) \right] \\
& + \left[a_2 V_1 V_2 cos((\omega_2 - \omega_1)t) + a_2 V_1 V_2 cos((\omega_2 + \omega_1)t) \right] \\
& + \left[\frac{3}{4} a_3 V_1^2 V_2 cos((2\omega_1 - \omega_2)t) + \frac{3}{4} a_3 V_1 V_2^2 cos((2\omega_2 - \omega_1)t) \right] \\
& + \left[\frac{3}{4} a_3 V_1^2 V_2 cos((2\omega_1 + \omega_2)t) + \frac{3}{4} a_3 V_1 V_2^2 cos((2\omega_2 + \omega_1)t) \right]
\end{aligned}
\tag{1.10}
$$

Equation (7.6) can be illustrated more clearly in Figure 1.3 which is the frequency domain representation of the output signal. The output signal consists of the useful signal, the DC component, harmonics and intermodulation products [6]. The useful signal is the signal at the fundamental frequency. The harmonics and the second-order intermodulation components can be canceled by filters. However, there are third-order intermodulation components which are close in frequency to the fundamental frequency. These intermodulation distortions thus cannot be removed simply by filters.
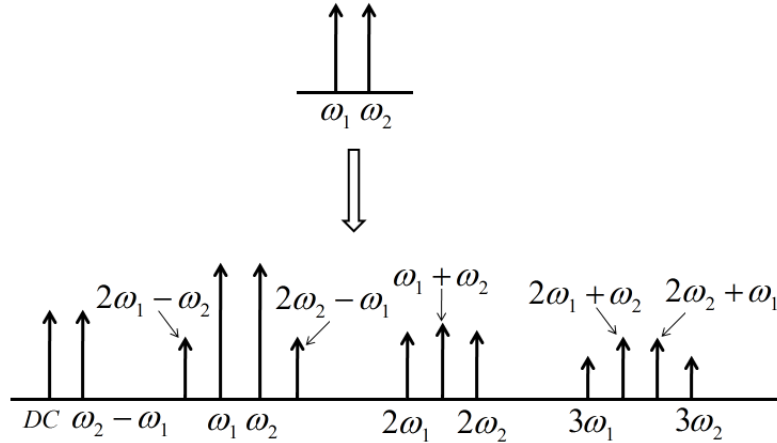
Figure 1.3.   Frequency domain output of a nonlinear PA driven by a two-tone signal

1.3   AM/AM and AM/PM Characteristics

AM/AM and AM/PM characteristics [7] are the basic curves to characterize the linearity of the PA. They get their name from the fact that amplitude modulation (AM) within the input signal results in amplitude and phase modulation (AM and PM) in the power amplifier gain. In a more general case, one PA is well described by four characteristics: amplitude to amplitude modulation (AM/AM) characteristic, the amplitude to phase modulation (AM/PM) characteristic, the phase to phase modulation (PM/PM) characteristic, and the phase to amplitude modulation (PM/AM) characteristic. Since phase modulated signals are immune to the distortions, only AM/AM and AM/PM characteristics are considered.

Let's consider a PA with a modulated signal whose baseband input and output signal x and y are

$$x = x_I + jx_Q$$
$$y = y_I + jy_Q$$

(1.11)

where x_I, y_I are the in-phase components and x_Q, y_Q are the quadrature components. Therefore, the magnitude and phase of the instantaneous signal gain can be expressed as

$$|G| = \frac{|y|}{|x|} = \frac{\sqrt{y_I^2 + y_Q^2}}{\sqrt{x_I^2 + x_Q^2}}$$

(1.12)

$$\angle G = tan^{-1}\left(\frac{y_Q}{y_I}\right) - tan^{-1}\left(\frac{x_Q}{x_I}\right)$$

(1.13)

8

Figure 1.4.   Sample AM/AM characteristic of a power amplifier

Figure 1.5.   Sample AM/PM characteristic of a power amplifier

The AM/AM characteristic of PA can be obtained by plotting the amplitude of the output signal as a function of the corresponding input signal. Similarly, the AM/PM

characteristic of the PA is the phase of the power gain as a function of the instantaneous input signal. Figure 1.4 and Figure 1.5 are an example of the AM/AM and AM/PM characteristics, respectively. These two figures provide a very clear view of the linearity of the PA.

1.4 Memory Effects

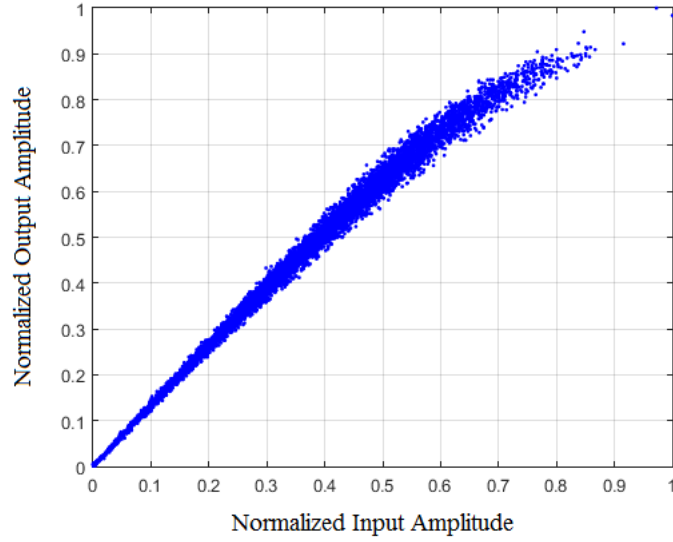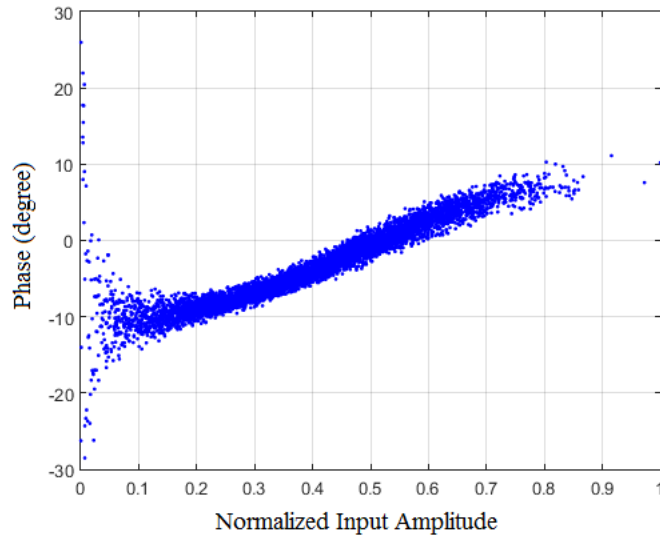Memory effects [8–11] in a PA can be explained as that the output of the PA at time t_o is a function not only of the input at time t_o but also of the inputs at some other previous time instants. In other words, the past input signals also contribute to the current output value. The influence of the past input signals is not infinite and the memory depth describes how long of the past input signals that contribute. The relation between the input and the output of a PA with memory can be expressed as

$$y(t_o) = f\left(x(t_o),\, x(t_o - 1),\, \cdots x(t_o - \tau)\right) \tag{1.14}$$

where the memory depth τ is a positive constant since every physical system is causal. From (1.14), we can see that the outputs corresponding to the same input value may be different since the past input signals may be different. Therefore, the characteristics of the output signal as a function of input signal, both in magnitude and phase, for a system with memory are not a single line as shown in Figure 1.4, Figure 1.5.

Memory effect is mainly a result of energy storage in the device. A simple example that exhibits memory is the RC circuit in which the current flowing in the circuit depends on how much charge accumulated on the capacitor. The sources of memory effects [8, 12–14] are not unique and they can be classified into two groups according to the timescale of the occurrence: short-term memory effects and long-term memory effects.

1.4.1 Short-Term Memory Effects

Short-term memory effects are those that occur with a timescale same with the RF carrier frequency. The main sources of short-term memory effects are matching network and device capacitance. The matching networks at both input and output of a PA are essential to keep the maximum power transfer. However, these matching networks contain capacitors, inductors and transmission lines which present dynamic

properties. Therefore, the matching network should be carefully designed to reduce the memory impact.

The transistors themselves have parasitic capacitance and inductance which will contribute to the memory effect. These parasitic reactive components are associated with the charge transfer delay in the channel of the transistors. Therefore, the transistor is usually modeled with reactive components in the small-signal model. Generally, we should try to minimize the parasitic capacitance and inductance to reduce the delay.

1.4.2   Long-Term Memory Effects

Similar to the short-term memory effects definition, the long-term memory effects are those that occur with a timescale much longer. These kinds of sources include thermal effects [15] and charge trapping. The PA is not ideal 100% energy transferred and some of the energy will be wasted as heat which will change the temperature of the PA. This temperature change is relatively slow compared to the carrier frequency. However, this change can still affect some temperature dependent parameters, such as channel mobility, and the behavior of the PA will be different even with the same input signal.

Charge trapping is mainly due to the imperfect semiconductor materials and processing and these imperfections happen mainly at the interfaces of two dissimilar semiconductors. Then the local potential will be disrupted. As a result, the charge carriers may be trapped and released later, causing a change of current flow, if the potential conditions are different. This will change the behavior of the PA, contributing to the memory effects.

Chapter 2

Power Amplifier Linearization Techniques

RF power amplifiers are the main cause of the nonlinear behavior in the modern wireless transmission system. Nowadays, spectrum efficient modulation techniques, such as orthogonal frequency division multiplexing (OFDM), are required in modern communication systems. However, these non-constant envelope modulation are very sensitive to the nonlinearity. The signal with non-constant envelope generates intermodulation distortion (IMD) when it passes the PA. The IMD power will interfere between adjacent channels which is an unwanted phenomenon.

Moreover,these efficient modulation techniques will lead to modulated signals with high peak-to-average power ratio (PAPR). The PA should operate with power back-off which will result with a drastically decreased efficiency. To ensure the high efficient operation, many techniques have been proposed in literature. Among them, the Doherty amplifier [16] is one of the most promising techniques for high efficiency operation with non-constant envelope modulated signal. It has several advantages in terms of relative simplicity of implementations and cost effectiveness. However, the linearity and efficiency of PAs usually present opposite performances. The Doherty amplifier operates with high efficiency often at the cost of poor linearity. Therefore, linearization techniques have to be used to compensate the nonlinear distortions generated by PAs.

Various techniques are available for this purpose, such as feedforward, feedback and analog predistortion. However these solutions suffer from complex circuitry and stability problems. The digital predistortion is one of the most useful and cost-effective linearization techniques, due to the already existing digital signal processing devices.

2.1   Power Back-off

Figure 2.1 shows the typical performance of a power amplifier, including the magnitude response, the power gain and the efficiency. From the figure, we know that the PA is linear at the lower input power range while the linearity becomes worse with higher input power. Therefore, the simplest way to improve the linearity of the PA is to back-off the input power to the linear region. This is called the power back-off technique [1].
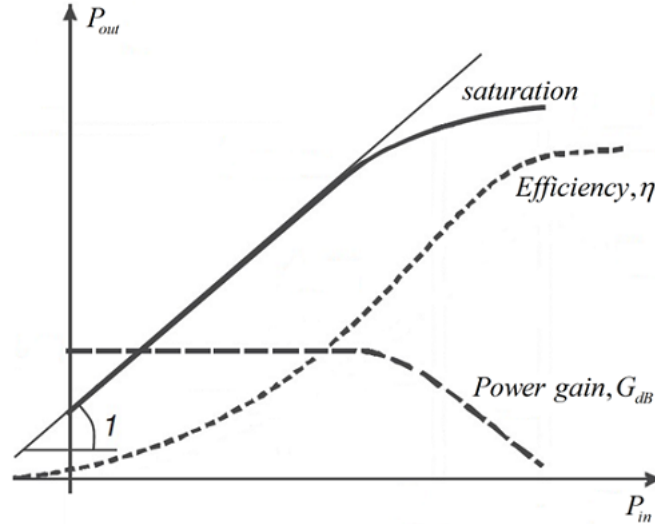
Figure 2.1.   Basic AM/AM characteristic of a power amplifier

Another explanation is more quantitative with mathematical equations. Since every device with nonlinear characteristic can be expanded with Taylor series, we can model the PA with the following expression:

$$v_{out} = V_0 + a_1 v_{in} + a_2 v_{in}^2 + \cdots \tag{2.1}$$

where V_0 is the DC part which can be blocked in RF power amplifiers, v_{in} and v_{out} are the input and output, respectively. Therefore, if the value of v_{in} reduces to a real low level, the linear term $a_1 v_{in}$ will dominate over the other power terms, leading to a more linear characteristic.

The power back-off method is still widely used for low-power, small size PAs. However, when back-off to the linear region, the efficiency of the PA is degraded, especially for signals with high peak to average power ratio (PAPR). Therefore, the power back-off solution has no effect to overcome the trade-off between linearity and

efficiency. It improves the linearity by sacrificing the efficiency since the efficiency is best at saturation (as shown in Figure 2.1). To obtain both linearity and efficiency, other linearization techniques are needed.

2.2 LINC

LINC refers to Linear amplification using Nonlinear Components [17–20], which can realize a linear amplification with high efficient nonlinear amplifiers. As illustrated in Figure 2.2, the basic idea of LINC is to separate the amplitude modulated input signal into two phase modulated signals. In this way, the non-constant envelope modulation, which are very sensitive to the PA's nonlinearity, will be replaced by the constant envelope modulation. Therefore, the resulted modulated signals can be amplified separately by two efficient PAs without any concern of distortion. These two efficient PAs should be ideally identical, with same characteristics and the delay in the two channels should also be matched. Finally, the amplified signals from the two channels should be combined to produce the amplified replica of the input signal.
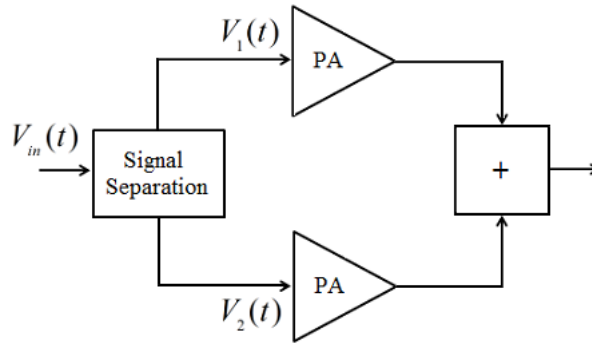
Figure 2.2.   The basic principle of LINC

Consider first a amplitude modulation signal:

$$V_{in}(t) = a(t)cos(2\pi f_c t + \varphi(t)) \tag{2.2}$$

where $a(t)$ is the envelope of the signal. When passing this signal into the signal separation component, it will be divided into two constant-amplitude signals $V_1(t)$

14

and $V_2(t)$, expressed as:

$$V_1(t) = 0.5V\cos(2\pi f_c t + \varphi(t) + \theta(t)) \tag{2.3}$$

$$V_2(t) = 0.5V\cos(2\pi f_c t + \varphi(t) - \theta(t)) \tag{2.4}$$

where $\theta(t) = cos^{-1}(a(t)/V)$, V is the maximum amplitude of $a(t)$.

The concept of LINC has appeared since 1930s. However, at that time, the separated signals in LINC technique were not accurate since it was difficult to generate the $\theta(t)$ with the analog circuits. The modern digital signal processing technique can solve this problem and can make a significant improvement for the accuracy. Nevertheless, there are still problems. For instance, the two channels should be ideally matched.

2.3 Feedforward

Feedforward [21–25] is an old linearization technique which applies the feedback correction at the output of the PA. It was first introduced by Howard Black in 1923 at Bell Telephone Laboratories. Figure 2.3 shows the basic principle of feedforward technique.

Figure 2.3. Basic schematic of the feedforward technique

The input signal is first divided into two signals, one being amplified by the main

amplifier as required, the other one being used as a reference with the same delay with the main amplifier. The delayed reference signal is then subtracted from the coupled signal from the main amplifier. Therefore, if there is no amplitude and phase distortion, the subtraction procedure will have a zero value. Otherwise, there will be a error signal after the subtraction. The error signal is then fed to the error amplifier to be amplified back to the original level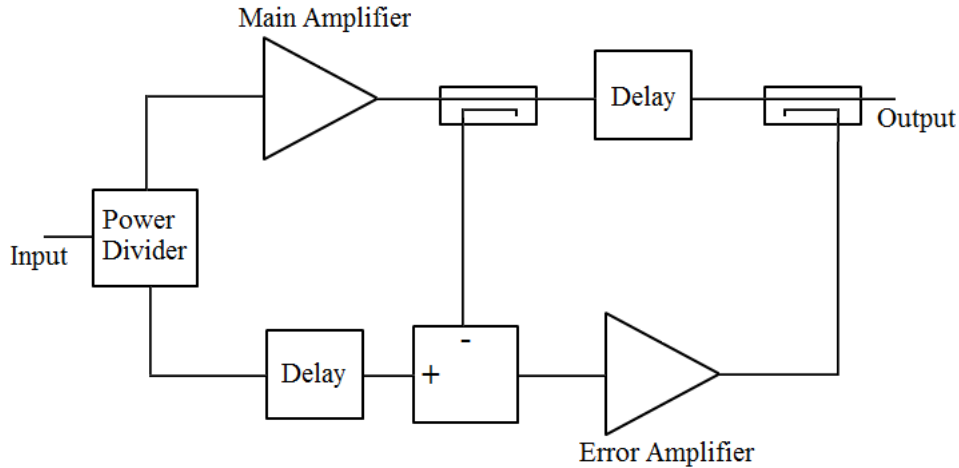. The amplified error signal is the compensation to the output signal from the main amplifier to get the linear and amplified signal at the output of the feedforward system.

The feedforward technique can remove both the amplitude and phase distortions of the PA and is immune to memory effects. Compared to the feedback technique, the feedforward method can provide the same benefits of feedback technique without the problem of bandwidth limitation. However, all theses come with costs. Firstly, the error amplifier in Figure 2.3 should be linear since the gain errors introduced by the error amplifier can not be compensated. Moreover, it should also be powerful enough to amplify the error signal to the same level of the main signal. Secondly, the delay lines must be carefully designed to produce the accurate delay and to match the signals in the two channels. The linearization performance is highly dependent on the precision of the signals matching in the two channels. Another aspect, the feedforward technique works with poor efficiency and has high demand on the hardware. In addition, feedforward can not linearize the PA operated at its saturation power.

2.4   Feedback

Feedback technique [1,26], which is based on the knowledge of dynamic control system, was also mentioned by Howard Black in 1927 after the failing of the feedforward idea. It has been widely used in many systems since it has been mentioned. The basic idea of feedback technique is that a part of output signal is fed back to the input with a close loop. Therefore, the difference between feedforward and feedback techniques is that the former compares the error signal at the output of the PA, whereas the latter compares at the input. A general direct feedback linearization is illustrated in Figure 2.4.

In Figure 2.4, a negative feedback loop is used to linearize the PA. The loop consists of the PA with power gain G, a divider of gain $1/K$ and a comparator for signal subtraction.
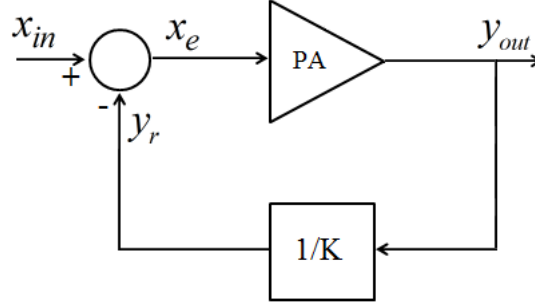
Figure 2.4.   General direct feedback system

A part of the output signal y_{out} from the PA is fed back to the input passing through the divider. The signal after the divider is

$$y_r = \frac{1}{K}y_{out} \tag{2.5}$$

The error signal, x_e, after the comparator is defined as

$$x_e = x_{in} - y_r \tag{2.6}$$

where x_{in} is the input signal. Therefore, the output signal after the amplifier is

$$y_{out} = Gx_e \tag{2.7}$$

Finally, combining all the equations, we will get a linear gain. Therefore, this close loop can force the output signal to be a replica of the input which makes the PA linear.

$$y_{out} = \frac{KG}{K+G}x_{in} \tag{2.8}$$

This simple technique is widely used in applications with low frequencies, in which, the time delay of the close loop is short enough and can be neglected. However, at RF, this delay will be comparable to the signal frequency and cannot ignored. In addition, at high frequencies, the phase of the PA gain varies rapidly with the frequency and usually this can lead to oscillation. Therefore, the direct feedback technique has a bandwidth limitation and works well only at low frequencies. To overcome this restriction, an indirect feedback technique [27] is applied, as shown in Figure 2.5.

In this technique, there are two peak detectors at both the input and output of the RF PA. Both the input and output signals are captured and down-converted to the IF frequency. Then the detected baseband signals are fed to a video differential amplifier where an error correction signal is generated. Then this error signal, as a
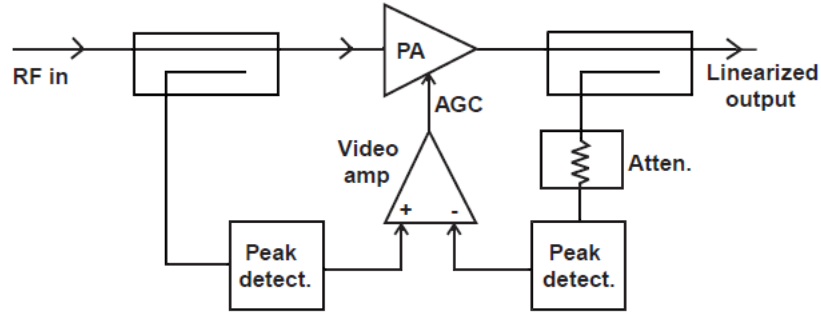
Figure 2.5.   General indirect feedback system

driver, passes to a gain control of the amplifier. As a result, this loop will force the output envelope to be identical to the input envelope.

This technique still has a problem: the high order intermodulation products can not be corrected. This problem is inevitable because of the delay limitation of the feedback systems. Despite their good linearization performance, feedback techniques are still found a limited usage for the linearization of RF PAs.

2.5   Baseband Digital Predistortion

In these years, baseband digital predistortion (DPD) [28, 29] becomes the preferred choice for the linearization of RF power amplifier because of its relative simplicity and good performance. Different from feedforward and feedback techniques, the DPD technique is achieved by placing a nonlinear block, called *Predistorter*, just in front of the PA as shown in Figure 2.6. The objective of this predistorter is to produce an inverse behavior of the PA.
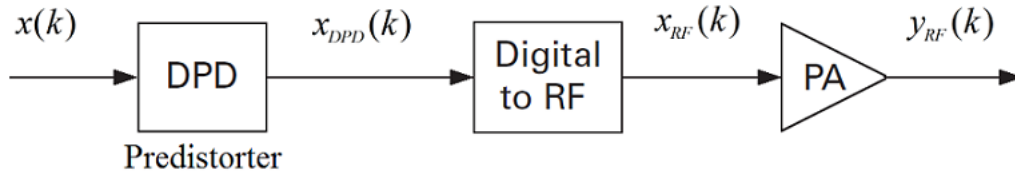
Figure 2.6.   Digital transmitter with digital predistortion

Digital predistortion is now gaining worldwide popularity. It has smaller size and less cost compared to other linearization techniques. Digital predistortion can also be a standalone device which makes it more convenient in industrial applications. More details will be introduced in next chapter for digital predistortion.

Chapter 3

Baseband Digital Predistortion

The aim of studying distortions [8], caused by nonlinearity of the power amplifier, is to design a linearization architecture to minimize the distortions. There are many techniques mentioned in the previous chapter, among which the digital predistortion (DPD) is one of the most popular solution. The popularity of baseband DPD is mainly associated with its flexible implementation and good accuracy due to the implementation of digital signal processing techniques. Another attractive aspect is that the entire DPD system can be encapsulate into one standalone block. This will avoid redesigning the analog circuits when the DPD system is used.

FPGA is a good choice for implementing the DPD technique in wireless applications. It has many advantages in digital signal processing, including high speed processing, flexible implementation, high reliability and parallelism computation. Lookup Table (LUT) method is an efficient way to implement the predistorion function. The proposed DPD technique is not solely limited to FPGA and it can also be implemented based on other commercial products.

This chapter introduces the basic principle of the digital predistortion technique and the implementation issues of DPD for the RF wireless communication system. Two DPD architectures, direct and indirect learning architecture, are compared. With DPD, the PA has a significant improvement in linearity which can satisfy the highly linearity requirement of high efficient power amplifiers. However, digital predistortion also suffers from some problems, including the bandwidth expansion of the signal and additional energy consumption.

3.1 Basic Principle

The basic idea of digital predistortion [5, 28–31] is to introduce a nonlinear component, called *Predistorter*, just in front of the PA. The objective of this nonlinear block is to produces the nonlinear behavior which is the reverse of the PA's nonlinear behavior in both magnitude and phase. In this way, the predistorter will counteract the nonlinearity of the PA and the final behavior will be linear, as shown in Figure 3.1.
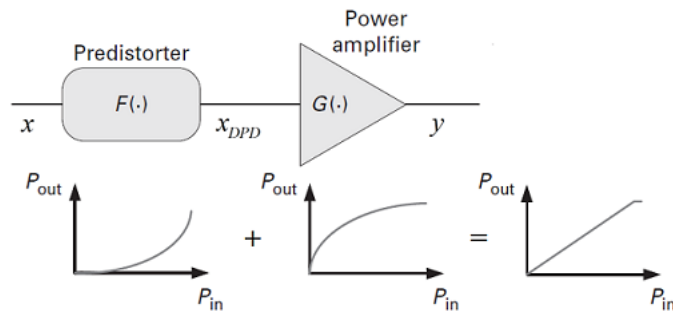
Figure 3.1.   Basic principle of digital predistortion

Another interpretation of digital predistortion in frequency domain is to view the predistorter as a generator of intermodulation distortion (IMD) products. Since the PA is usually nonlinear, it will create IMD products. If the IMD products of the PA and the predistorter have the equal amplitude and 180 degree out of phase, the distortion will be cancelled, as shown in Figure 3.2 in which the downward arrow means the anti-phase.
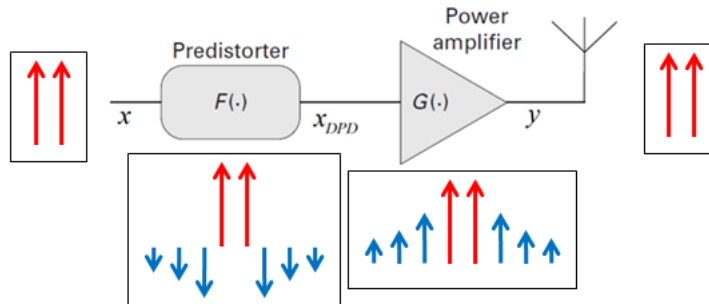
Figure 3.2.   Frequency domain interpretation of digital predistortion

21

In the following, the principle of DPD will be explained with mathematical equations. The predistorted signal, denoted as x_{DPD}, is

$$x_{DPD} = F(x), \tag{3.1}$$

where x is the input signal. The predistorted signal can be expressed in another form:

$$x_{DPD} = G_{DPD}(x) \cdot x, \tag{3.2}$$

where $G_{DPD}(x)$ can be seen as the nonlinear gain of the predistorter at point x, or it is the slope of the transfer function between x_{DPD} and x. The predistored signal will be fed to the PA and the output signal will be

$$y = G(x_{DPD}) = G(F(x)) \tag{3.3}$$

which can also be expressed as

$$y = G_{PA}(x_{DPD}) \cdot x_{DPD}, \tag{3.4}$$

where $G_{PA}(x_{DPD})$ is the nonlinear gain of the PA and also can be seen as the slope of the transfer function between y and x_{DPD}. Since the PA and the predistorter have the opposite nonlinear behavior, under the assumption that the output signal y is normalized by the power gain of the PA, we will have

$$G_{DPD} = \frac{1}{G_{PA}} \tag{3.5}$$

The gain of the entire system, consisting of both predistorter and PA, can be derived by

$$\begin{aligned} G = \frac{dy}{dx} = \frac{dy}{dx_{DPD}} \cdot \frac{dx_{DPD}}{dx} \\ = G_{PA} \cdot G_{DPD} = 1 \end{aligned} \tag{3.6}$$

Therefore, the gain of the system is 1, a normalized constant, which means that the system is linear.

We will offer an example with data measured from a real PA to demonstrate the principle of the DPD. Figure 3.3 and Figure 3.4 illustrate the AM/AM and AM/PM characteristics for the sample PA with memory (blue), a corresponding predistorter (green) and the entire system with both PA and predistorter (red). In Figure 3.3, the horizontal axis is the magnitude of the input signal to predistorter x and the input to PA x_{DPD}, while the vertical axis is the magnitude of the output signal from predistorter x_{DPD} and the normalized output from PA y/G_o. As shown in Figure 3.3, the magnitude response of the PA is not a straight and the power gain is compressed at high power region. The predistorter, on the contrary, produces the gain expansion to compensate the compression. As a result, the resulting magnitude is
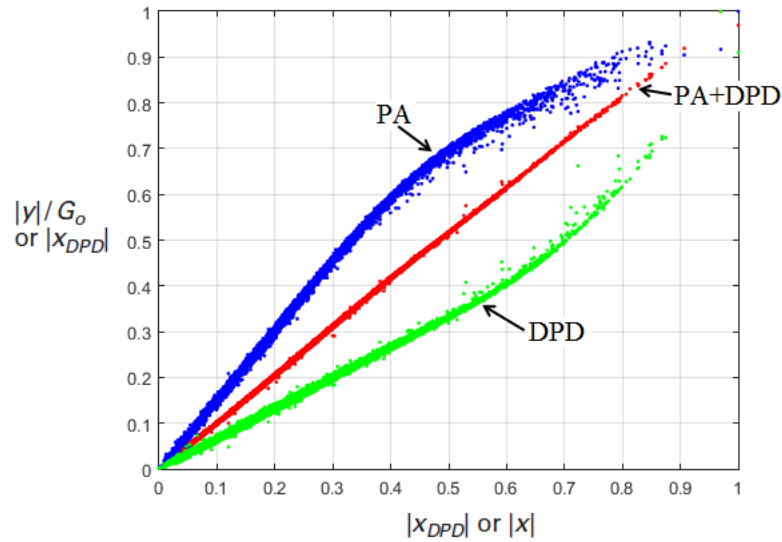
Figure 3.3. Transfer functions for a sample PA with memory (blue), predistorter suitable to obtain linear response (green), PA and predistorter(red)
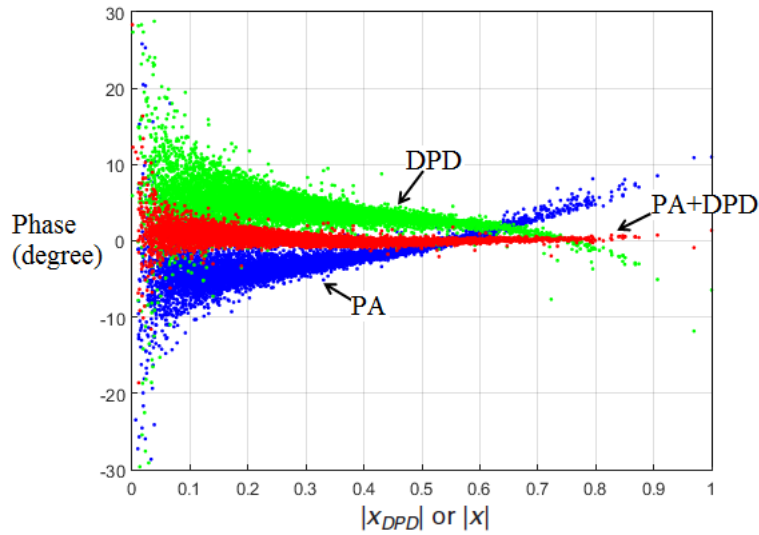
Figure 3.4. Phase curves for a sample PA with memory (blue), predistorter suitable to obtain linear response (green), PA and predistorter(red)

a straight line and the gain is flat, meaning that the PA is linear.

Meanwhile, in Figure 3.3 of the AM/PM characteristics, the horizontal axis is still the magnitude of the input to predistorter x and the input to PA x_{DPD}, while the

vertical axis is the phase of the PA, the predistorter and the entire system. The phase here means the phase difference between the output and input of the device. Therefore, same to the theoretical principle, the phase of the PA and the predistorter is opposite to each other. Moreover, with the DPD, the resulting phase of the whole system is nearly zero, leading to a linear response.

The basic principle of DPD can be explained in another more intuitive way with Figure 3.5. Suppose that at the input, there is a signal with power at point x_1. The response at the output of the PA is with power at point y_1. However, with an ideal linear behavior, the output should be at point y_2. Therefore, to obtain the power at point y_2, the input signal fed to the PA should have the power at point x_2. It is the job for the predistorter to convert the power from point x_1 to the point x_2 in order to get the linear result from the PA.
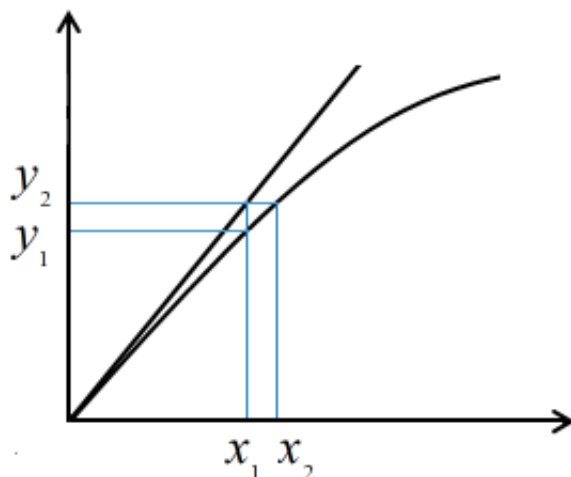
Figure 3.5.   Power amplifier output power vs input power

The DPD technique is implemented in digital domain, where digital signal processing technique is applied to simplify the computation. Analog-to-digital (ADC) devices are needed to convert the analog signal into digital domain. Then a model is required to describe the predistorter and to generate the predistorted signal. The detail on the modeling will be introduced in next chapter. The digital predistorted signal should pass through a digital-to-analog (DAC) to be converted back to analog signal. At last, the signal will be up-converted to the required frequency and is fed to the PA.

3.2   DPD implementation

Due to the fast improvements of digital signal processing techniques, the baseband DPD has became the most adopted linearization technique. Figure 3.6 illustrates the simplified implementation of the DPD system. The signal is firstly digitized and then fed to the predistorter. The predistorter function is usually applied on a digital signal processing unit (FPGA or DSP) where the input signal is processed with a predefined algorithm. The predistorted signal is then converted back to analog signal which will be modulated to the required frequency and fed to the power amplifier.
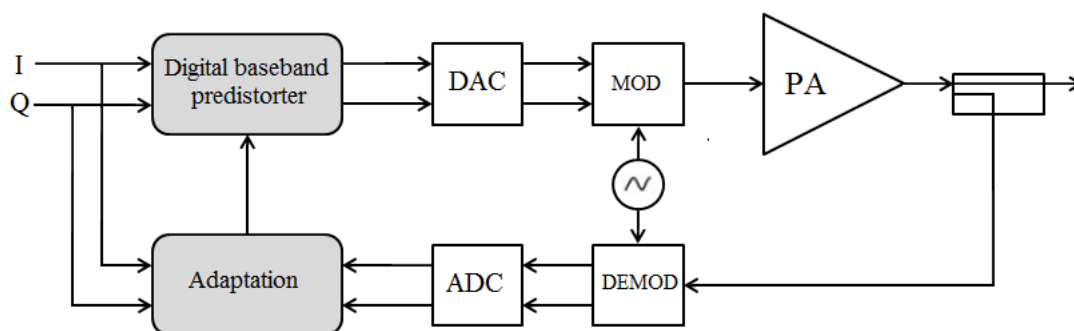
Figure 3.6.   Simplified block diagram of a DPD implementation in a transmitter

In Figure 3.6, there is a closed loop feedback which is for adaptive behavior of DPD. Since the characteristic of a PA varies in many cases, including different bias supply, changes in temperature, aging, it is essential to continuously update the predistorter function. In an adaptive DPD system, a fraction of the output signal from the PA is fed back by using a coupler. The feedback signal is demodulated to baseband and digitized for extracting the DPD function. The detail of the adaptive algorithms used in DPD will be discussed in chapter 7.

The input and output signals used to extract the predistoter function are the baseband signal to PA and the baseband signal output from PA, respectively. Therefore, the nonlinear behavior we get consists of all the nonlinear components in the chain and the DPD can minimize the nonlinearity of all these nonlinear components.

3.3 DPD Architecture

The basic principle of digital predistortion technique [5, 32] is to introduce a non-linear component which has the inverse characteristic of the PA. Therefore, it is essential to derive the inverse model with the input and output data of the PA. One method employed in the DPD system is the model inverse.

The model inverse structure is to find an inverse function to be used as the model of predistorter. There are two commonly used architectures: direct learning and indirect learning.

3.3.1 Direct Learning Architecture

The direct learning architecture, as its name states, directly adjusts the predistorter model parameters by using the feedback error $e(n)$ with an adaption algorithm, as shown in Figure 3.7. The error is the difference between the input signal $x(n)$ and the normalized output signal $z(n)$, which are time aligned. Several adaption algorithms [30–32] have been proposed for direct learning architecture. These algorithms tune the parameters of the predistorter according to the feedback error in order to minimize the feedback error. Since the error is zero for linear system, we will obtain an approximately linear behavior at last with minimum feedback error.
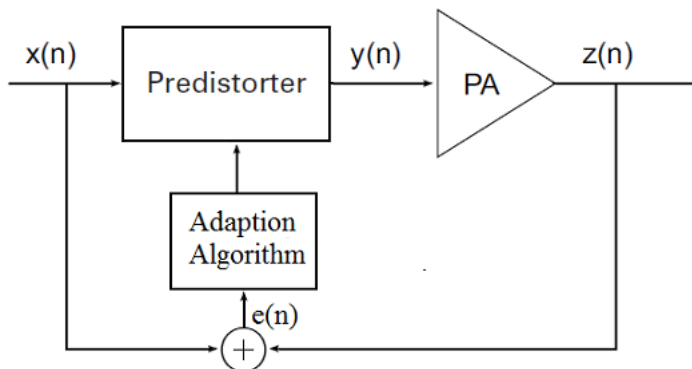
Figure 3.7.   Block diagram of the direct learning architecture

3.3.2   Indirect Learning Architecture

Indirect learning architecture [5, 33–36], which introduces a post predistorter, is another commonly used method in the model parameters extraction for predistorter. Its basic schematic is depicted in Figure 3.8, in which $x(n)$ is the input to PD, $y(n)$ is the output from PD and also the input to PA, $z(n)$ is the normalized output from PA, $\hat{y}(n)$ is the output from the post predistorter block, the error signal $e(n)$ is given by $e(n) = y(n) - \hat{y}(n)$. The post-inverse estimation block and the predistorter have the identical nonlinear transfer function. The post-inverse estimation block generates the parameters of PD by minimizing the error signal $e(n)$. Since that when the PA is linear, $x(n) = z(n)$ and thus $y(n) = \hat{y}(n)$. Finally, the estimated parameters are copied to the predistorter.
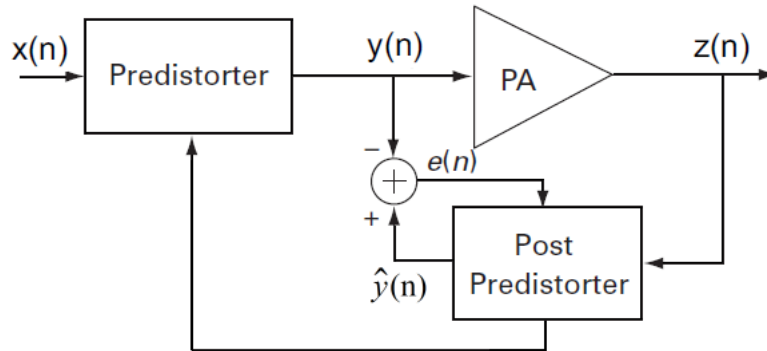
Figure 3.8.   Block diagram of the indirect learning architecture

3.4   DPD problems

The major problem is the bandwidth expansion of the predistorted signal x_{DPD}. The predistorted signal x_{DPD} usually has a wider bandwidth than the input signal x since the predistorter introduces the nonlinear distortion compensation. As a result, the sampling rate of x_{DPD} should be several times higher than the sample rate associated with input signal x. This throws a higher requirement on digital-to-analog converters (DACs). In return, the DAC speed limitation will restrict the maximum bandwidth of the input signal.

Another concern is focused on energy consumption of the predistorter circuitry which is usually several watt. The efficiency of a PA with DPD should also take

the energy of the predistorter into consideration. Therefore, the DPD technique is suitable for PAs that exceeds 10 W. The energy consumption is related to the size of the hardware used to implement the predistorter. Therefore, the complexity of the predistorter should not be too high in order to reduce the energy consumed in predistorter.

The value of PAPR, which is defined as the ratio between the peak power and the average power, will become larger after the predistorter since the predistorter has a gain expansion. Therefore, the input power level should back-off by more than the original PAPR from the saturation level.

Chapter 4

Power Amplifier Behavioral models

The basic principle of DPD is to add a nonlinear block just in front of the PA to compensate the PA gain compression. This block produces the exact inverse non-linearity of the PA. Hence, a model that can predict accurately the behavior of the PA and the predistorter is required. Since RF PAs usually exhibit memory effects, the model should be able to describe not only dynamic nonlinear behavior but also memory effects. Behavioral model is the best choice, since it is built simply from the observation of the input-output behavior of the device instead of the knowledge of the device's internal constitution. There are various behavioral models proposed in the literature, such as the most comprehensive, full Volterra model and its simplified versions, like memory polynomial model or generalized memory polynomial model. Other polynomial function based models include rational function model, Hammerstein model, Wiener model and so on. We should choose one among all these models which is a tough work. The model selection is mainly based on model accuracy, model computational complexity and model extraction technique. The memory polynomial model is chosen at last due to its low complexity and satisfactory accuracy.

The model extraction technique is another important aspect that need to be taken into consideration. The most common solution is the least square (LS) technique which requires that the model should be linear in its parameters. The LS method can be adopted for memory polynomial model. Other extraction techniques, such as least mean square (LMS), recursive least square (RLS), are also popular for DPD systems. Different from LS method, the LMS and RLS techniques generate the model parameters sample by sample with the input and output data.

4.1 Behavioral models

One prerequisite to apply digital predistortion is to model the PA and predistorter accurately which is determined by the model selected. There are mainly three basic and different strategies for the device modeling: physical modeling, circuit modeling and behavioral modeling. Each of them has its advantage and drawback and is applied in different fields.

Physical model is based on the device physics. Using a microwave metal-semiconductor field-effect transistor (MESFET) as an example, the physical properties including oxide thickness, doping density, carrier mobility and so on, are the basic knowledge to build the physical model. They should solve the transport and Poisson equations at a microscopic level. These kind of models are complex to extract that make them inadequate in the real simulation.
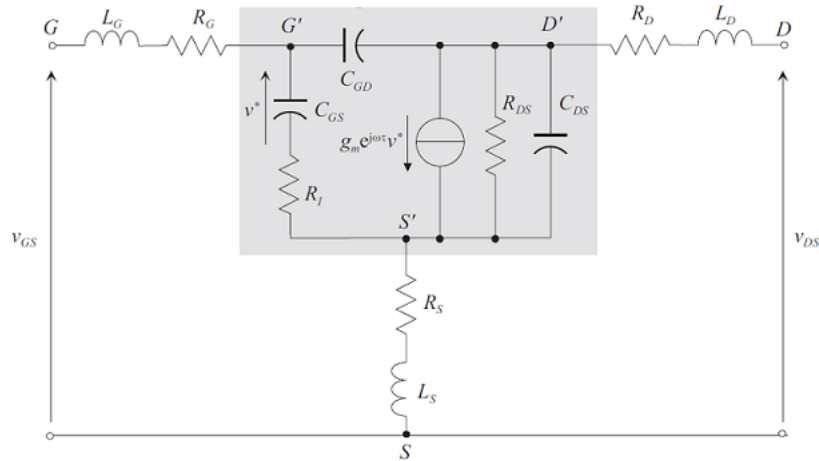
Figure 4.1.   A sample FET small-signal equivalent circuit model

Circuit modeling uses active passive components to model the behavior of the transistors or diodes. Figure 4.1 shows a typical small-signal equivalent circuit model of a FET which consists both the intrinsic and extrinsic parts. These models can be constructed from S-parameter or I-V characteristics. Circuit model can be used to predict the time domain current or voltage of a PA, although they are never directly measured.

On the contrary, behavioral modeling [37,38], also called black-box modeling (Figure 4.2), does not require the knowledge of the internal constitution of the PA and
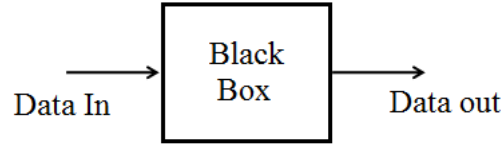
Figure 4.2.   Black-box modeling

relies only on the observation of input and output signal of the PA. The procedure is to build and tune a mathematical expression in order to make the output of the model coincided with the measured output of the PA with the same input signal. Therefore, behavioral modeling is a more computationally efficient solution and it is easier to implement the model with computational devices. Only the input and output signals of the PA are required, instead of looking deep inside the PA's internal structure. The selection of a behavioral model is mainly based on the criteria of the model accuracy, the computational complexity and the model extraction technique. It is the ideal tool for the system level simulation and it can find a widespread use in wireless communication systems.

Behavioral models can generally be divided into two groups according to the memory effects: memoryless models and models with memory. Each group has a variant number of models with different complexity and performance which will be discussed in the following sections.

4.2   Figure of Merit

One of the most critical requirements in DPD technique is to model the predistorter accurately. It is essential to clearly evaluate the performance of the model. Therefore, model performance evaluation criteria should be adopted to choose the proper model. The most commonly used criteria are NMSE in time domain and ACPR in frequency domain.

In the following sections, the test signal for the models is the LTE signal with 9.8 dB PAPR and a channel of 28 MHz. The device under test (DUT) is a wideband class AB amplifier whose characteristic is shown in Figure 4.3. It can be seen clearly that this DUT exhibits memory effects since the characteristic behavior is not a single line.
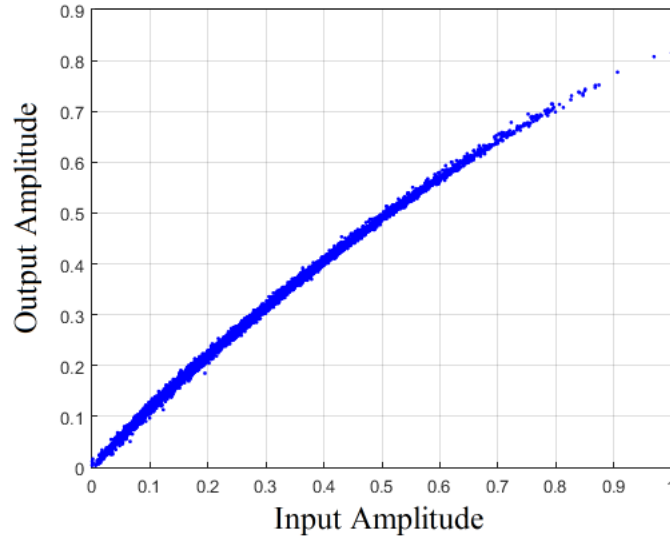
Figure 4.3.   AM/AM characteristic of the sample DUT.

4.2.1   NMSE

NMSE refers to Normalized Mean Square Error, which is an estimator of the overall deviations between the predicted and measured values in time domain. Therefore, it is the most straightforward approach and it is often expressed in decibels as following.

$$\text{NMSE} = 10 \; log_{10} \left(\frac{\sum_k |y_k - x_k|^2}{\sum_k |x_k|^2} \right) \tag{4.1}$$

where x_k is the experimental output instant of the DUT and y_k is the output instant obtained from the model. Therefore, the accuracy of the model is inversely proportional to NMSE. In other words, the lower the NMSE is, the higher the model accuracy is.

The NMSE can also represent the linearity of a PA with x_k, y_k being the input instant and normalized output instant of the PA, respectively. In this case, -35 dB of NMSE will mean that the PA has a very good linear behavior.

4.2.2 ACPR

The adjacent channel power ratio (ACPR) is a frequency domain evaluation criterion, which is defined as the power ratio between the total power of adjacent channels and main channel. The main objective is to check the power distribution between in in-band channel and adjacent channels. It is possible to quantify the error in different frequency domain ranges. Its expression is given as

$$ACPR = 10 \, log_{10} \left(\frac{\int_{f_1}^{f_2} |E(f)|^2 \, df + \int_{f_3}^{f_4} |E(f)|^2 \, df}{\int_{f_2}^{f_3} |E_{desired}(f)|^2 \, df} \right) \tag{4.2}$$

where $|E(f)|^2$ is the power at one frequency f and frequency range $(f_1 \ f_2)$ is the lower adjacent channel and $(f_3 \ f_4)$ is the upper adjacent channel and $(f_2 \ f_3)$ is the in-band channel. Therefore, the first and the second integral in numerator calculate the lower and upper adjacent channel power, respectively while the denominator is the in-band channel power. Figure 4.4 is an example of ACPR figure of an output signal where the in-band frequency range is -14 MHz to 14 MHz.

Figure 4.4.   ACPR figure of a sample PA output

4.3 Even-order Terms in the Polynomial

In this chapter, all the models with power polynomials include only the odd-order terms. Although, in [77], it is proved that it is beneficial to include both the odd- and even-order terms for the models. However, most other published papers (e.g. [55,78]) consider only the odd-order terms. In this section, we will explain why only the odd-order terms are considered for the modeling in our DPD with one example of a real test power amplifier.
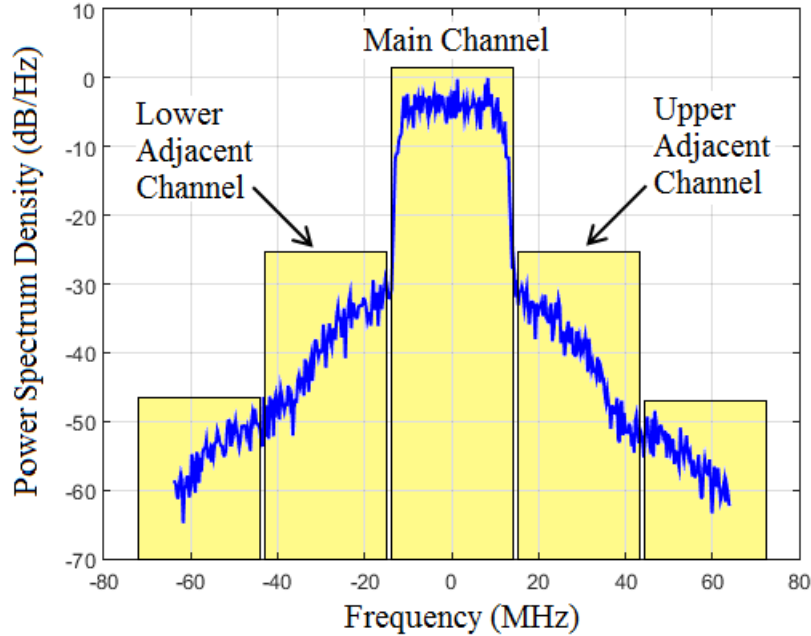
It would seem that all the terms in the polynomial should be used to deliver the best representation of the PA magnitude and phase responses. However, when we acquire the output signal from a PA in a real measurement, the signal is limited to the frequency band of interest. Only the carrier and odd-order intermodulation signals can be detected, while the even-order intermodulation signals will be filtered. For this reason, the even-order terms would appear to be redundant.

We do the same test which has been done in [77] with the memory polynomial model and our test PA data. We use the same three sets of polynomial orders for comparison: $K_1 = \{1, 3, 5, 7, 9\}$, $K_2 = \{1, 3, 5\}$ and $K_3 = \{1, 2, 3, 4, 5\}$. The three sets are compared quantitatively with the NMSE between the output signal of the experimental data and the model which are shown in Table 4.1. K_2 and K_3 have the same largest polynomial order, 5. Therefore, the performance of the model is better with both even- and odd-order terms if the largest polynomial order is the same. However, if the number of the terms in a polynomial model is the same, K_1 contains only the odd-order terms and has a better fit to the experimental data.

	K_1	K_2	K_3
NMSE (dB)	-37.6	-35.2	-36.9

Table 4.1.   Comparison of NMSE between 3 sets of polynomial orders

The same result can be seen in Figure 4.5. The x-axis is the input amplitude and the y-axis is the output amplitude error between the experimental data and the model. The magenta points stands for polynomial order of $K_1 = \{1, 3, 5, 7, 9\}$, green for $K_2 = \{1, 3, 5\}$ and blue for $K_3 = \{1, 2, 3, 4, 5\}$. K_1 yields the best result with slightly smaller error than K_3 while K_2 shows the worst fit to the experimental data. We can conclude that the odd-order terms are enough for the modeling with polynomial terms. Therefore, we only consider the odd-order terms in the memory polynomial model and other models with power terms.
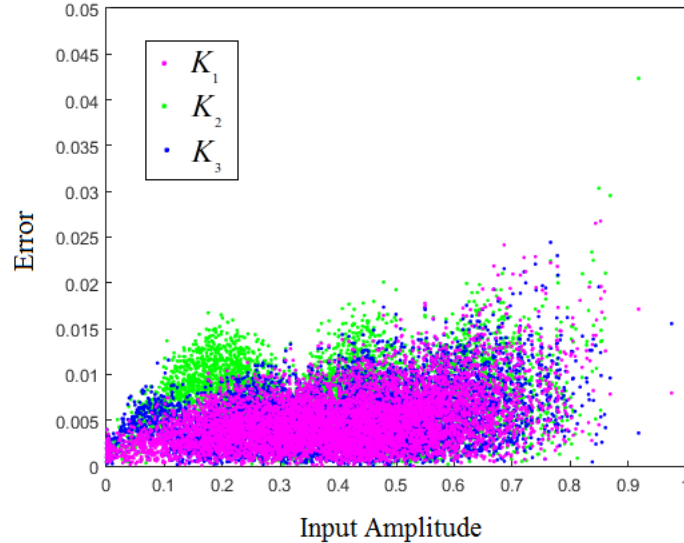
Figure 4.5. Comparison between 3 sets of polynomial orders. The x-axis is the input amplitude and the y-axis is the output amplitude error between the experimental data and the model. $K_1 = \{1, 3, 5, 7, 9\}$, $K_2 = \{1, 3, 5\}$ and $K_3 = \{1, 2, 3, 4, 5\}$.

4.4 Memoryless models

Memoryless behavioural modelling has many advantages, including easier computational implementation, relative efficiency in system simulations and its acceptable level of accuracy in many situations. Therefore, it has been used for many years in system level simulation.

Memoryless means that the output is only dependent on the current input and the previous samples have on effect on the output. There is a one-to-one mapping between present input and output signals. Therefore, if the PA is strictly memoryless, it will only introduce amplitude distortion.

The most used memoryless model is polynomial function [39]. Others, including Rapp model [42] and Ghorbani model [41] which are similar of the Saleh model [10], are also introduced in the literatures. Look-up table (LUT) method is another structure to model the memoryless systems, where the data are stored in a table in advance.

4.4.1 Memoryless polynomial model

The most natural and used memoryless model is the polynomial function [39], whose equation is

$$y = \sum_{p=0}^{P} a_p x^p \tag{4.3}$$

where x and y are the input and output signals, respectively, P is the polynomial order and a_p is the corresponding coefficients. The comparison between the experimental data and the model is shown in Figure 4.6. The polynomial order is 4 with only even order. As a result. the NMSE for this memoryless polynomial model is -32.2 dB.
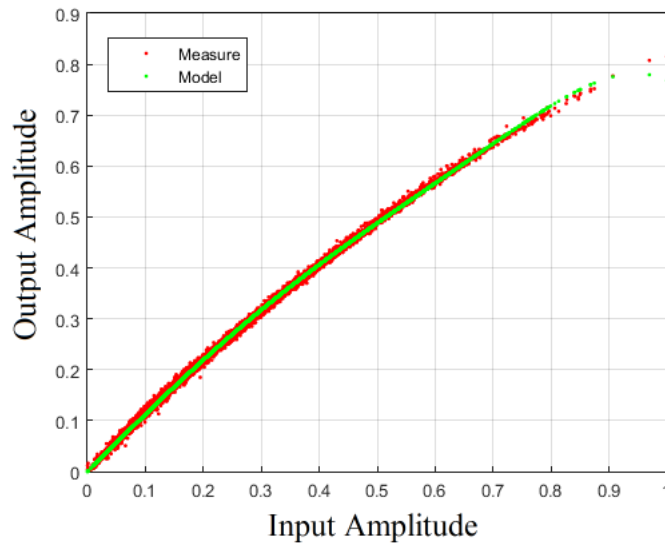
Figure 4.6.   Measured and modeled AM-AM characteristics comparison for memoryless polynomial model. The red represents the measured data and the green is for the model.

There are other memoryless or quasi-memoryless models including Rapp model [42] and Ghorbani model [41] which are similar of the Saleh model [10]. There are also several other similar models based on power series, including Bessel Function Based Model [43], Gegenbauer Polynomials Based Model [44] and Zernike Polynomials Based Model [45].

4.4.2   LUT model

The LUT model [28] is another basic behavioral model with momoryless nonlinearity, which is relatively simple and easily implement. Different from the analytical function-based model, the LUT model stores all possible values in a table in advance and retrieve them later. It is a memory-cost model and the memory is used as the table. In other words, the LUT model saves the computational complexity with memory storage.

The simplest solution for LUT model is the one-to-one mapping, saving each possible output signal in the table indexed by its corresponding input signal. This solution is able to compensate any memoryless distortion. However, since every possible signal has to be stored, the memory size required is huge. Another propose is to store only the complex gain indexed by the magnitude of the input signal. There is a direct relation between the input and output, expressed as:

$$y(n) = G(|x(n)|) \cdot x(n), \tag{4.4}$$

where $G(|x(n)|)$ is the instantaneous complex gain, $x(n)$ and $y(n)$ are the input and output signals, respectively. The complex gain $G(|x(n)|)$ is calculate and stored in a table. Figure 4.7 illustrates the operation of the LUT model based on complex gain method, where $|.|$ means the magnitude. Each input corresponds to a unique complex gain and the output is the multiplication of the input and the corresponding gain. Therefore, this model is suitable for real-time applications.
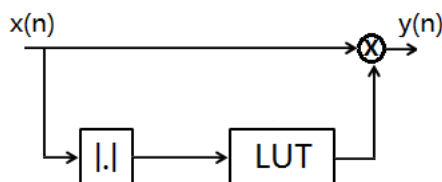
Figure 4.7.   Look-up-table model

However, in the meantime, LUT model also suffers some problems. For instance, extra memory resource is required to be as the tables, to store the complex gain. And only a limit number of LUT entries can be implemented which will lead to the quantization error.

LUT model can also used for PAs with memory effects. In memory LUT model [40],

the complex gain is a function of the present and several preceding inputs instead of just present input. Therefore, the LUT size is K^{M+1}, where K is the size of memoryless LUT and M is the memory depth. It faces a more severe problem in memory size.

4.5 Models with memory

On the contrary, a model with memory is not only dependent on the current input but also on a certain depth of previous samples, which makes the system dynamic. Moreover, to accurately model the PA, some models with memory also depend on previous output values. Most of the RF power amplifiers used in the modern wireless communication systems exhibit the memory effects. Therefore, behavioral models with memory are more and more important.

The most comprehensive memory model is the full Volterra model. Volterra model is a polynomial expansion which can represent nonlinear systems with memory. It is typically truncated to simplified structures, giving rise to several other memory models with less complexity. Among them, the memory polynomial model is widely used in the behavioral modeling.

4.5.1 Full Volterra model

The Volterra model [46,47] provides a general way to model a nonlinear system with memory which is a combination of linear convolution and nonlinear power series. It is considered as an extension of the Taylor series. In this model, the relationship between the input and output signals is:

$$y(n) = \sum_{p=1}^{P} \sum_{i_1=0}^{M} \cdots \sum_{i_p=0}^{M} h_p(i_1, \cdots, i_p) \prod_{j=1}^{p} x(n - i_j) \qquad (4.5)$$

where $x(n)$ and $y(n)$ are the input and output signals, respectively, $h_p(i_1, \cdots, i_p)$ are the coefficients of the Volterra model, often called Volterra kernels, P is the nonlinearity order of the model, and M is the memory depth.

We can improve accuracy of the model by increasing K and M. Unfortunately, this high accuracy is obtained at the cost of unnecessary computational complexity

since the number of parameters will grow exponentially. Nevertheless, the Volterra model is well suitable for modeling dynamic nonlinear behavior.

4.5.2 Memory polynomial model

The memory polynomial model [36, 48–50] has been widely used as a behavioral modeling in digital predistortion of PAs with memory effects for many years. It corresponds to a reduction of the Volterra series in which only diagonal terms are kept. It has a reasonable compromise between computational complexity and model accuracy. The output waveform of the model is

$$y(n) = \sum_{m=0}^{M} \sum_{p=0}^{P} a_{mp} x(n-m)|x(n-m)|^p \tag{4.6}$$

where $x(n)$ and $y(n)$ are the input and output signals, respectively, P is the nonlinear order, M is the memory depth and a_{mp} are the model coefficients. The comparison between the experimental data and the model is shown in Figure 4.8. The polynomial order is 7 with only even order and the memory depth is 2. As a result, the NMSE for this memory polynomial model is -35.6 dB.

Figure 4.8. Measured and modeled AM-AM characteristics comparison for memory polynomial model. The red represents the measured data and the green is for the model.

A variant of other memory polynomial models have been proposed in literature. They are Orthogonal Memory Polynomial model [51], Non-Uniform Memory Polynomial model [52, 53], Envelope Memory Polynomial model [54].

4.5.3   Generalized Memory Polynomial Model

As mentioned above, the memory polynomial model is a simplified case of Volterra model which eliminates all the cross memory parts. However, as the the signal bandwidth has significantly increased, the memory polynomial model is not enough and the introduction of the cross terms in the full Volterra model is needed. A generalized form of the pth memory polynomial component in (4.6) can be written as:

$$\sum_{m=0}^{M}\sum_{p=0}^{P} a_{mp}x(n)|x(n-m)|^p, \tag{4.7}$$

where we have inserted a local delay of m samples between the signal and its exponential part. The delay in (4.7) could be both positive and negative. If we add both these positive and negative local memory that close to the current memory to the typical memory polynomial model in (4.6), it will come out the generalized memory polynomial model [55].

$$\begin{aligned} y(n) &= \sum_{p=0}^{P_a}\sum_{l=0}^{L_a} a_{pl}x(n-l)|x(n-l)|^p \\ &+ \sum_{p=1}^{P_b}\sum_{l=0}^{L_b}\sum_{m=1}^{M_b} b_{plm}x(n-l)|x(n-l-m)|^p \\ &+ \sum_{p=1}^{P_c}\sum_{l=0}^{L_c}\sum_{m=1}^{M_c} c_{plm}x(n-l)|x(n-l+m)|^p \end{aligned} \tag{4.8}$$

where the first part is the same as memory polynomial model, while the last two terms are the cross memory with both positive and negative time shifts. P_a and L_a, P_b and L_b, P_c and L_c are the polynomial order and memory depth for the current, positive and negative parts, respectively, M_b and M_c are the local memory that shift from the current memory.

The generalized memory polynomial model can be seen as the advanced version of the memory polynomial model. It has a better performance, in terms of reducing

spectral regrowth, than the memory polynomial model. However, it suffers a problem of higher computational complexity.

P_a	L_a	P_b	L_b	M_b	P_b	L_b	M_b
7	2	2	2	1	2	2	1

Table 4.2.   Memory depth and polynomial order chosen

The comparison between the experimental data and the model is shown in Figure 4.9. The memory depth and the polynomial order are chosen as listed in Table 4.2. As a result, the NMSE for the generalized memory polynomial model is -35.6 dB.
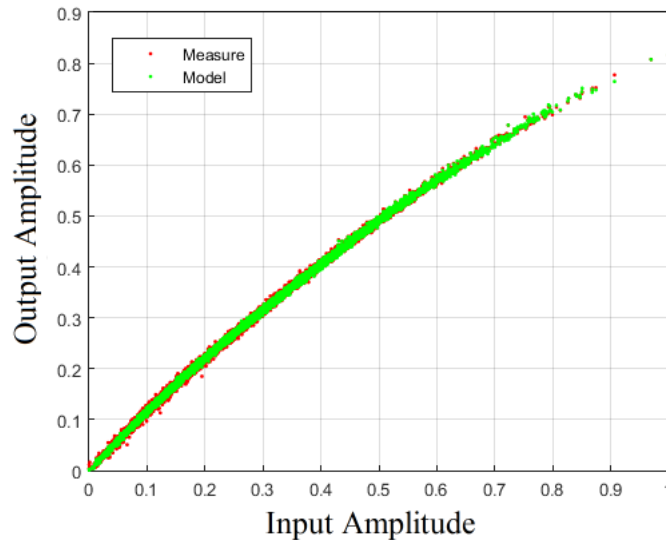

Figure 4.9.   Measured and modeled AM-AM characteristics comparison for generalized memory polynomial model. The red represents the measured data and the green is for the model.

4.5.4   Dynamic Deviation Reduction-based Volterra Model

Dynamic deviation reduction-based Volterra model [56–58] is another modified and simplified version of the Volterra model. In this model, the input data are organized according to the order of dynamics involved. We remove the high order dynamics, because the nonlinear effects can be ignored with high order dynamics for most PAs. The property of linearity in the parameters has been reserved in this model.

As mentioned above, the number of parameters increased exponentially with non-linearity order and memory depth for Volterra model. However, for this model, the situation is different and the number of parameters increases almost linearly. Therefore, it is suitable to accurately model a PA with static strong nonlinearities and with long-term linear and low-order nonlinear memory effects.

The expression for this model is

$$
\begin{aligned}
y(n) = {} & \sum_{p=1}^{P} h_{p,0} x^{p}(n) \\
& + \sum_{p=1}^{P} \left[x^{p-1}(n) \sum_{i=1}^{M} h_{p,1}(i) x(n-i) \right] \\
& + \sum_{p=2}^{P} \left[x^{p-2}(n) \sum_{i_1=1}^{M} \sum_{i_2=i_1}^{M} h_{p,2}(i_1,i_2) x(n-i_1) x(n-i_2) \right]
\end{aligned}
\tag{4.9}
$$

where P is chosen as 7 and M is 2. The comparison between the experimental data and the model is shown in Figure 4.10. As a result, the NMSE for the dynamic deviation reduction-based model is -33.1 dB.

Figure 4.10. Measured and modeled AM-AM characteristics comparison for dynamic deviation reduction-based Volterra model. The red represents the measured data and the green is for the model.

4.5.5 Rational function

A rational function is mathematically defined as the ratio of two polynomial functions, expressed as:

$$y(n) = \frac{a_0 + a_1 x(n) + \cdots a_I x^I(n)}{b_0 + b_1 x(n) + \cdots b_J x^J(n)} \tag{4.10}$$

where $x(n)$ is the input and $y(n)$ is the output of the rational function at instance n, and the highest polynomial order, I and J, are not required to be equal.

It is expected that the rational function should be able to model the PA accurately [59, 60]. However, (4.10) can not compensate the memory effects. In order to include the memory effects, a new model based on the rational function was proposed [61], where memory effects were included both in the numerator and denominator, whose expression is:

$$y(n) = \frac{\sum_{p=0}^{P_n} \sum_{m_n=0}^{M_n} a_{p,m_n} x(n - m_n)|x(n - m_n)|^p}{1 + \sum_{p=0}^{P_d} \sum_{m_d=0}^{M_d} b_{p,m_d}|x(n - m_d)|^p} \tag{4.11}$$

where $x(n)$ and $y(n)$ are the input and output signals, respectively, P_n and M_n, P_d and M_d are the polynomial order and memory depth for the numerator and denominator, respectively. The memory effects are described with absolute terms in the denominator and with complex-envelope terms in the numerator.

The model expressed in (4.11) is more complex and less accurate than memory polynomial model. Therefore, another version of rational function was proposed [62], which replaces the absolute terms memory with the complex-envelop terms without memory. The proposed model is given by

$$y(n) = \frac{\sum_{p=0}^{P_n} \sum_{m=0}^{M} a_{p,m} x(n - m)|x(n - m)|^p}{1 + \sum_{p=0}^{P_d} b_p x(n)|x(n)|^p} \tag{4.12}$$

where $x(n)$ and $y(n)$ are still the input and output signals, respectively, M is the memory depth, P_n and P_d are the polynomial orders in the numerator and denominator, respectively, and $a_{p,m}$ and b_p are the complex coefficients for the numerator and denominator, respectively.

M, P_n, P_d are chosen as 2, 6, 2, respectively. The comparison between the experimental data and the model is shown in Figure 4.11. As a result, the NMSE for the rational function based model is -35.8 dB.
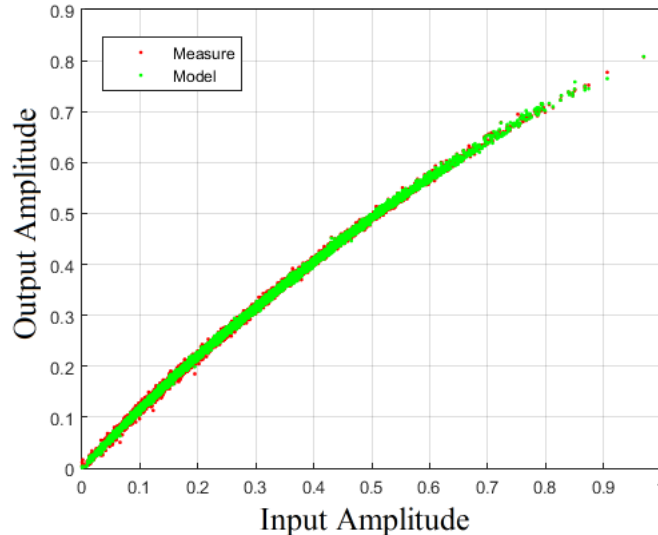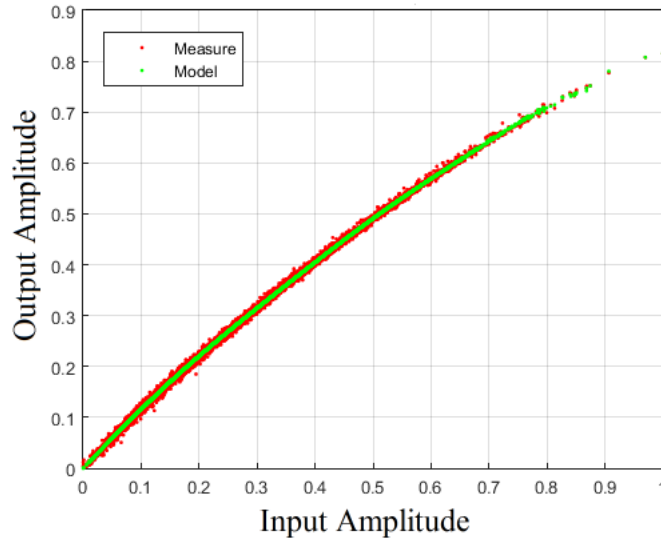
Figure 4.11.   Measured and modeled AM-AM characteristics comparison for rational function model. The red represents the measured data and the green is for the model.

4.5.6   Hammerstein Model

Hammerstein model [63,64] consists of two cascade stages, as shown in Figure 4.12. The input signal first goes through the nonlinear block then passes the linear memory block. For the nonlinear block, a general power series are used while for the linear block we consider an FIR filter.
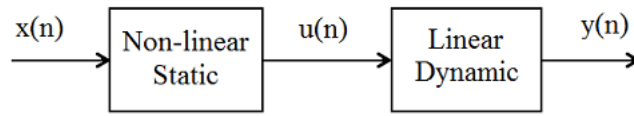
Figure 4.12.   Hammerstein's nonlinear model

The Hammerstein model is usually defined as:

$$u(n) = \sum_{p=0}^{P} a_p x(n)|x(n)|^p \tag{4.13}$$

$$y(n) = \sum_{m=0}^{M} b_m u(n - m) \tag{4.14}$$

44

where $y(n)$ is the output of the PA and $u(n)$ is the intermediate output of the non-linear block. P is the order of the memoryless polynomial.

If we combine the two equations, we obtain the following expression

$$y(n) = \sum_{m=0}^{M} b_m \left(\sum_{p=0}^{P} a_p x(n-m)|x(n-m)|^p \right) \tag{4.15}$$

From equation (4.15), we can find that the final expression of the Hammerstein model is the same with the memory polynomial model if we combine each pair of a_p and b_m into only one coefficient. Therefore, comparing to memory polynomial model, the Hammerstein model has the same accuracy, but with more coefficients to be extracted.

4.5.7 Wiener Model

On the contrary to Hammerstein model, the Wiener model [65, 66] passes the linear memory block firstly then goes through the nonlinear stage, depicted in Figure 4.13. Wiener model is more complicated to estimate than Hammerstein model.
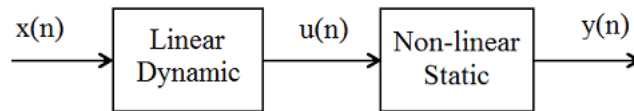
Figure 4.13.   Wiener's nonlinear model

For the nonlinearity we have considered power series as well as in the Hammerstein model. Therefore, the expressions for Wiener model is described as

$$u(n) = \sum_{m=0}^{M} b_m x(n-m) \tag{4.16}$$

$$y(n) = \sum_{p=0}^{P} a_p u(n)|u(n)|^p \tag{4.17}$$

If we combine these two equations, we obtain

$$y(n) = \sum_{p=0}^{P} a_p \left(\sum_{m=0}^{M} b_m x(n-m) \right) \mid \sum_{m=0}^{M} b_m x(n-m)\mid^p \qquad (4.18)$$

From (4.18), the coefficients are not linear since the filter coefficients are integrated in the power series. This makes the extraction of the coefficients more difficult. P, M are chosen as 2, 6,respectively. The comparison between the experimental data and the model is shown in Figure 4.14. As a result, the NMSE for the Wiener model is -33.1 dB.
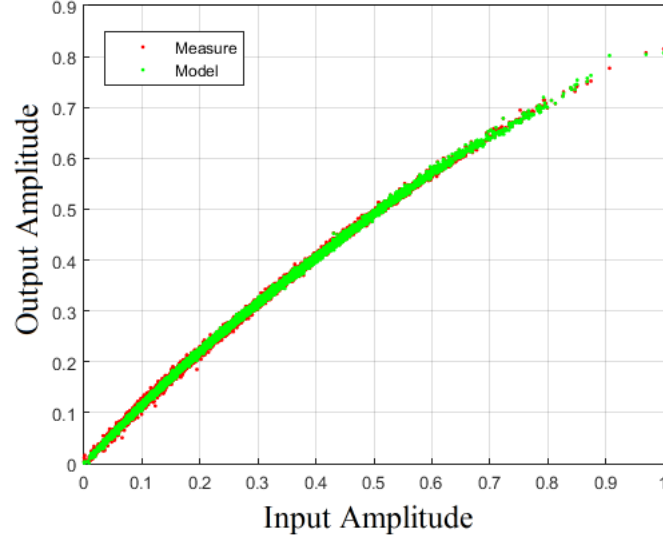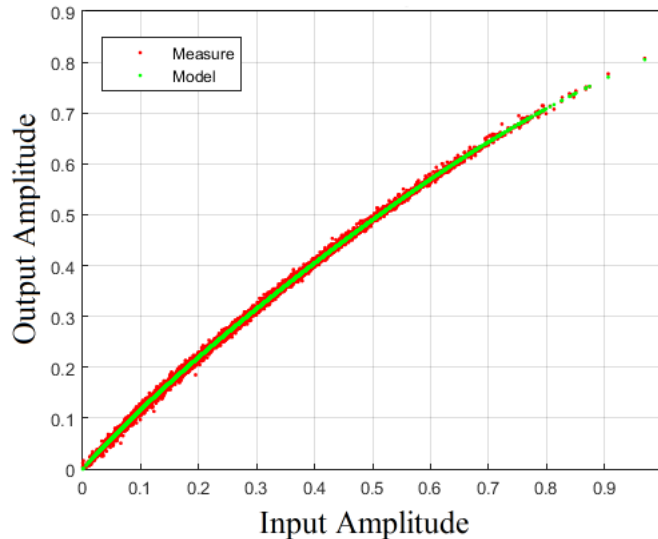

Figure 4.14. Measured and modeled AM-AM characteristics comparison for Wiener model. The red represents the measured data and the green is for the model.

4.6   Model Parameters Identification technique

In the previous sections, we have introduced some behavioral models for the modeling and predistortion of power amplifiers. All these models are just mathematical equations with unknown coefficients. These coefficients can be extracted, using identification techniques, if we have the input and output data of the PA. The accuracy of the model depends on the model type chosen. For example, if the PA exhibits memory effects during the measurements, the memory model should be chosen to make sure that the model can include these memory effects, while a memoryless

model cannot predict accurately the memory effects of the PA.

Therefore, choosing the adequate model structure is certainly important. After choosing a model, the model parameters have to be identified to fit the experimental data of the PA. The most common way is by solving a least square problem for models with linear property in their coefficients. Other linear adaptive techniques, including the recursive least square and least mean square algorithms, could also be adopted to estimate the coefficients.

4.6.1 Least Square Method

Least square technique [4] is a general estimation method introduced by A. Legendre in the early 1800's. This method has been most used in data fitting. The basic idea is to minimize the sum of the square residuals which is the difference between the measured data and data obtained from the model. Therefore, the best fit means the minimal residual.

The procedure is to adjust the parameters of a model to determine the best fit for the data set. A simple data set consists of n points (x,y). The model function has the form of $f(x,\beta)$ where β contains the parameters need to be identified. The final objective of this method is to find the parameters of the model which "best" fit the data.

The linear model with several explanatory variables is given by the equation

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_{k-1} x_{k-1} + \varepsilon \tag{4.19}$$

where ε can be seen as the error between the data from experimental measurement and model. If we have n pairs of data (x,y), (4.19) can be written in the matrix form as:

$$
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}
=
\begin{bmatrix}
1 & x_{11} & \cdots & x_{1k-1} \\
1 & x_{21} & \cdots & x_{2k-1} \\
\vdots & \vdots & & \vdots \\
1 & x_{n1} & \cdots & x_{nk-1}
\end{bmatrix}
\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{k-1} \end{bmatrix}
+
\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}
\tag{4.20}
$$

where the subscript n represents the observation number (in rows) while k refers to the variable number (in columns). (4.20) can be represented in summary vector notation by

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{4.21}$$

where $\boldsymbol{y}$ is an $n \times 1$ vector of output data, $\boldsymbol{X}$ is an $n \times k$ matrix of state values derived from input data, $\boldsymbol{\beta}$ is a $k \times 1$ vector of unknown parameters, $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector of unobserved disturbances.

Given a estimated parameter $\boldsymbol{\beta}$, the difference between the measured output and the model output is

$$\boldsymbol{\varepsilon} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \tag{4.22}$$

The criterion for the best model is to minimize the sum of the square of residuals which is written as

$$\begin{aligned} \boldsymbol{J}(\boldsymbol{\beta}) =& \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} \\ =& (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \\ =& \boldsymbol{y}'\boldsymbol{y} - \boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{y} + \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} \end{aligned} \tag{4.23}$$

Due to the fact that the transpose of a scalar is the scalar itself, $\boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta}$ and $\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{y}$ are identical. Therefore, the sum of the squares of residuals is

$$\boldsymbol{J}(\boldsymbol{\beta}) = \boldsymbol{y}'\boldsymbol{y} - 2\boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} \tag{4.24}$$

The minimum $\boldsymbol{J}(\boldsymbol{\beta})$ is obtained by setting the first-order derivation of $\boldsymbol{J}(\boldsymbol{\beta})$ equal to zero. According to the rules of matrix differentiation, the derivative is

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\beta}} = -2\boldsymbol{y}'\boldsymbol{X} + 2\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X} = 0 \tag{4.25}$$

which is transposed to provide the so-called normal equations

$$\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}'\boldsymbol{y} \tag{4.26}$$

Solving this for $\boldsymbol{\beta}$, we obtain

$$\boldsymbol{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{4.27}$$

where we assume that the inverse matrix of $\boldsymbol{X}'\boldsymbol{X}$ exists, which means the number of measurements is larger than the number of parameters, and these measurements are linearly independent.

4.6.2 Model Identification Examples using LS method

Most of the models introduced above are linear in their parameters which can be identified using the least square technique. Here we choose two models as examples

to implement the least square method.

The first model is the memory polynomial model whose expression is

$$y(n) = \sum_{m=0}^{M} \sum_{p=0}^{P} a_{mp} x(n-m) |x(n-m)|^p \tag{4.28}$$

If we rewrite it as a vector equation

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} \tag{4.29}$$

therefore, $\boldsymbol{y} = [y(0)\ y(1)\ \cdots\ y(n-1)]^T$, $\boldsymbol{\beta} = [\beta_{10} \cdots \beta_{P0} \cdots \beta_{1M} \cdots \beta_{PM}]$, $\boldsymbol{X} = [\boldsymbol{x_{10}} \cdots \boldsymbol{x_{P0}} \cdots \boldsymbol{x_{1M}} \cdots \boldsymbol{x_{PM}}]$, $\boldsymbol{x_{PM}} = [x_{PM}(0) \cdots x_{PM}(n-1)]^T$. The least square solution will be

$$\boldsymbol{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{4.30}$$

Another model is the rational function model whose expression is

$$y(n) = \frac{\sum_{j=0}^{P_n} \sum_{m=0}^{M} a_{j,m} x(n-m) |x(n-m)|^j}{1 + \sum_{i=0}^{P_d} b_i x(n) |x(n)|^i} \tag{4.31}$$

The model can also be written as

$$\begin{aligned} y(n) = & - y(n) \sum_{i=0}^{P_d} b_i x(n) |x(n)|^i \\ & + \sum_{j=0}^{P_n} \sum_{m=0}^{M} a_{j,m} x(n-m) |x(n-m)|^j \end{aligned} \tag{4.32}$$

It is clear that the model equation is linear in its parameters. Equation (4.32) can be written in a matrix form as:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} \tag{4.33}$$

where $\boldsymbol{y}$ is the output with n points, $\boldsymbol{\beta}$ is the coefficient vector defined as

$$\boldsymbol{\beta} = [b_0, \cdots, b_{P_d}, a_{00}, \cdots, a_{P_n 0}, a_{01}, \cdots, a_{P_n 1}, \cdots, a_{P_n M}]^T \tag{4.34}$$

To easily express matrix $\boldsymbol{X}$, we separate it into 2 parts as $\boldsymbol{X} = [\boldsymbol{X_1}\ \boldsymbol{X_2}]$, where

$$\boldsymbol{X}_1(n) = \begin{bmatrix} -y(n)x(n) & \cdots & -y(n)x(n)|x(n)|^{P_d} \\ -y(n-1)x(n-1) & \cdots & -y(n-1)x(n-1)|x(n-1)|^{P_d} \\ \vdots & \vdots & \vdots \\ -y(n-N)x(n-N) & \cdots & -y(n-N)x(n-N)|x(n-N)|^{P_d} \end{bmatrix}$$

$$\boldsymbol{X}_2(n) = \begin{bmatrix} x(n) & \cdots & x(n)|x(n)|^{P_n} & \cdots & x(n-M) & \cdots & x(n-M)|x(n-M)|^{P_n} \\ x(n-1) & \cdots & x(n-1)|x(n-1)|^{P_n} & \cdots & x(n-1-M) & \cdots & x(n-1-M)|x(n-1-M)|^{P_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x(n-N) & \cdots & x(n-N)|x(n-N)|^{P_n} & \cdots & x(n-N-M) & \cdots & x(n-N-M)|x(n-N-M)|^{P_n} \end{bmatrix}$$

where N is the number of data for the input or output. Therefore, using least square solution, the parameters are

$$\boldsymbol{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{4.35}$$

4.7 Model Comparison and Selection

There are various behavioral models that can be used in DPD system to simulate the behavior of the PA or the predistorter. An evaluation metric should be adopted to choose the proper model among a variety of behavioral models. Model performance, model computational complexity and model parameter extraction technique are the three main considerations for the model selection.

Model performance is straightforward and is based on the accuracy of the model. Several criteria have been proposed for evaluating the accuracy. The NMSE is the most common used metric in time domain while ACPR is the method in frequency domain.

Model complexity is also essential since the model should be implemented in hardware device in real applications. It is usually faster and more robust to use a model with less complexity. In our case, since the models are simulated in the Matlab software, we measure the time need to extract the model to indicate the complexity of each model. Figure 4.15 shows the performance versus complexity relation of each model mentioned in this chapter. In the figure, we can see that the memoryless polynomial model is the simplest but with the lowest performance. The generalized memory polynomial model has the best performance but with high complexity. The memory polynomial model, in the contrary, is with mild complexity and high performance.
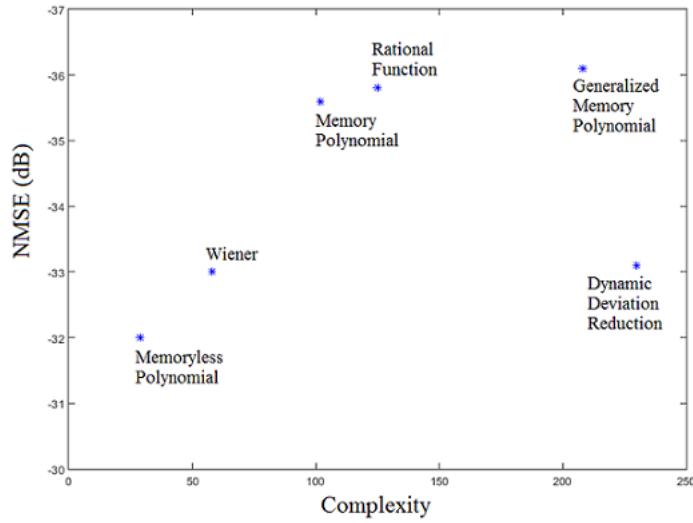
Figure 4.15.   Comparison of behavioral models for PA.

As for model parameter extraction, the least square method is the simplest and it requires that the model is linear for parameters. Among the models introduced in this chapter, only the Wiener model can not implement the least square method. These kinds of models, suffering from the problem of nonlinear parameter, requires iterative process which will increase the complexity.
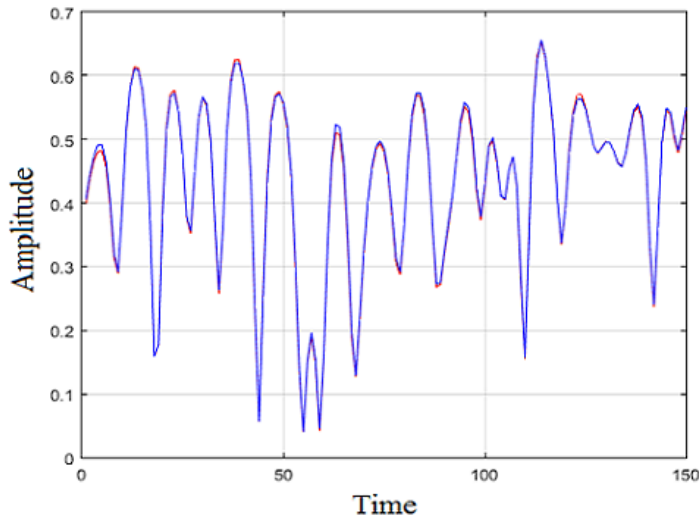
Figure 4.16.   Comparison of measured and model output signal in time domain.
Red is the measured data and blue is the modeled data.

Therefore, by considering the model performance, model complexity and the model parameters extraction technique, we select the memory polynomial model as the behavioral model to simulate the PA and the predistorter. Figure 4.16 is a comparison of the measured and model output signal in time domain for memory polynomial model.

Chapter 5

Software Simulation based on Matlab

The basic idea of digital predistortion is to introduce a nonlinear component, called *Predistorter*, in front of the PA to counteract the nonlinearity of the PA. One solution of the predistorter is to reproduce the inverse characteristic of the PA. Therefore, in basic, DPD is a mathematical algorithm and must be run on a device with computational capability. This device can be a processor controlled by a software or hardware logic resources like FPGA. A software testing environment based on Matlab is recommended at the initial develop stage since Matlab can be recognized as an ideal calculator with high computational accuracy during the computational procedure of the algorithm. If the DPD algorithm can not achieve a proper linearization performance in the ideal simulation, it will certainly not be able to linearize the real system. The input and corresponding output data of a PA are needed for the software simulation. After a successful validation in the ideal software environment, we can implement the DPD algorithm to linearize a real PA and test it with real instruments.

In this chapter, we will discuss some issues on how to simulate the digital predistortion technique in software testing environment (Matlab software). The entire system will be described, including the instruments used, the testbench and the simulation procedures. To validate the wide capability of the digital predistortion technique, several different PA architectures will be shown: a hybrid class AB PA, a hybrid Doherty PA, a MMIC class AB PA and a two-stages PA. Promising results have been achieved, indicating that our DPD algorithm works well in a wide range of applications.

5.1 Testbench Description

Figure 5.1 shows a block diagram of the experimental test bench for software simulation. The components, along the data flowing direction, are PC, signal generator, PA under test, attenuators and vector signal analyzer. All the components are introduced in detail in the following.
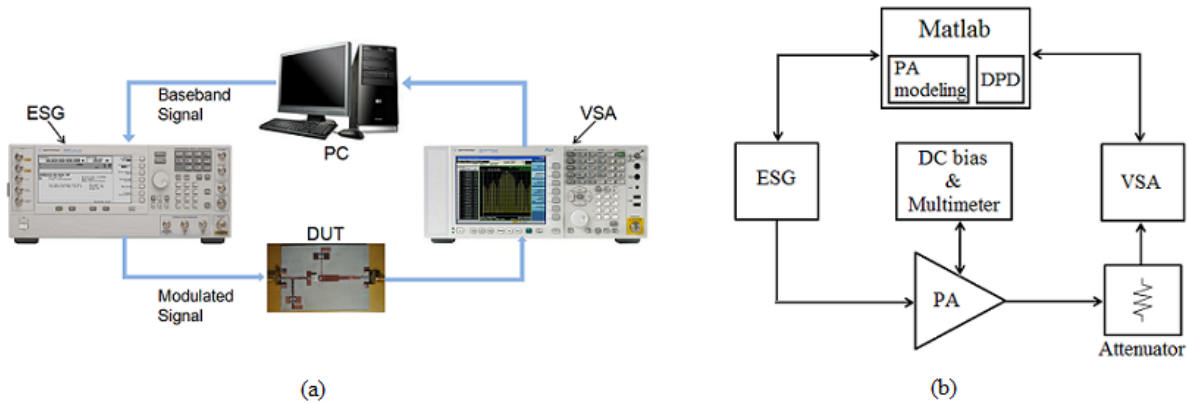
Figure 5.1.   Test bench with the most critical components for software simulation

1). PC
The PC acts as a processor to do the computational job which is controlled by the software Matlab. Both the PA model and the predistorter are estimated in Matlab. In other words, the entire DPD system can be ideally simulated in Matlab. Furthermore, the signal generator and the vector signal analyzer are both controlled by Matlab, including downloading original input signal to the signal generator and reading output data of the PA from the vector signal analyzer. At last, the performance of the DPD algorithm is evaluated in Matlab concerning the AM/AM, AM/PM characteristics, the NMSE between the output and the input signals and the ACPR of the output signal.

2). Signal Generator (ESG)
The signal generator used in our lab is the E8267D from Agilent Technologies, as shown in Figure 5.1(a). The ESG contains both the RF signal generator and the modulator. The modulation frequency can reach as high as 44 GHz. It supports various modulation types, including AM, PM, PSK, QAM and custom IQ modulation. The detail data sheet can be found in [67].

The ESG can be controlled by Matlab remotely via General Purpose Interface Bus (GPIB) and the programming scripts used can be found in [68]. An arbitrary IQ signal in baseband is firstly generated in Matlab and then we download it to the ESG with a unique GPIB address recognized by the Matlab scripts. The sampling rate, power level, modulation frequency and filter type are all preset in the Matlab scripts.

3). Vector Signal Analyzer (VSA)
The vector signal analyzer we use is the N9030A PXA Signal Analyzer from Agilent Technologies, as shown in Figure 5.1(a). The PXA signal analyzer is the highest-performance member of the X-Series, covering frequency range up to 50 GHz. The maximum input power level is 30 dBm which should be treated very carefully and attenuators should be added before the VSA for protection. The detailed data sheet can be found in [69].

The VSA can also be controlled by Matlab remotely with GPIB connection and the programming scripts can be found in [70]. It is helpful to read and store the output data into a PC since it is easier to deal with the data with Matlab software. The VSA is recognized by Matlab with its unique GPIB address. By selecting the basic IQ mode of the signal analyzer and the corresponding center frequency, the baseband IQ signal can be acquired in Matlab, together with other parameters, such as sample rate.

4). Others
The DC supply is the 6624A from Keysight Technologies. It has four channels, among which 2 of them are with range up to 20 V and the other 2 are with range up to 50 V. The multimeters are the 34401A Digital Multimeter from Agilent Technologies. They have 10 measurement functions among which only DC voltage and current channels are used in our measurements. The maximum input voltage is 1000 V and the maximum input current is 3 A which are enough for our measurements. The attenuators are from Weinschel Corp, operating between DC and 18 GHz frequency and with 10 dB of attenuation.

5.2 Simulation Procedure

To start the simulation, all the instruments have to be connected together according to Figure 5.1 where the ESG and the VSA are connected to the PC via GPIB. Furthermore, the ESG and the VSA are synchronized by the 10 MHz reference

frequency by connecting the 10 MHz OUT of ESG to the 10 MHz IN of VSA. Power attenuators are inserted before the VSA to avoid damage to the VSA. There are also DC bias circuit and the multimeters to measure the voltage and current at gate and drain of the PA which are used for the efficiency calculation. The testbench has been successfully prepared and the remaining test can follow the procedures which are summarized in Figure 5.2.
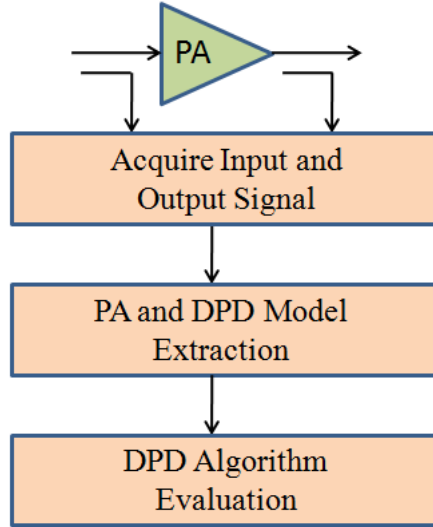
Figure 5.2.   Key procedures of DPD simulation

5.2.1   Attenuation Measurement

The attenuators should be selected according to the output power level of the PA and the maximum input power level of the VSA. It is not accurate to directly read the attenuation on the surface of the attenuator since the attenuation is frequency dependent. Therefore, the value of the total attenuation should be measured at the precise operation frequency of the PA. Suppose the power levels in decibel before and after the attenuators are P_1 and P_2, respectively. The attenuation P_{att} will be calculated as

$$P_{att} = P_1 - P_2 \tag{5.1}$$

To measure the power before the attenuators, a coupler is used after the PA and before the attenuators as illustrated in Figure 5.3. P_1 and P_2 can be read by the VSA directly.
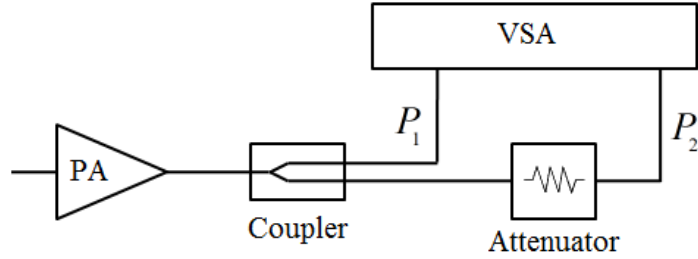
Figure 5.3.   Solution for attenuation measurement

5.2.2   Output Data Acquirement

To do the predistortion, the characteristics of the PA should be obtained first. There-fore, we should acquire the output data of the PA with a test input signal. The adopted excitation can be any kind of signals. In our simulation, it is an IQ signal with 256 quadrature amplitude modulation (QAM), 7.4 dB PAPR and a channel of 7 MHz and 28 MHz, or a WiMAX signal with 9.2 dB PAPR. The signal is created in Matlab and downloaded into the signal generator. In the signal generator, the baseband complex envelope signal will be up-converted to the required RF frequency and fed to the DUT as shown in Figure 5.4. The initial power level of the modulated signal from the ESG is recommended to be low to avoid any damage to the VSA. We can roughly know the compression power by checking the output power when it does not linearly increase with respect to the increase of the input power.
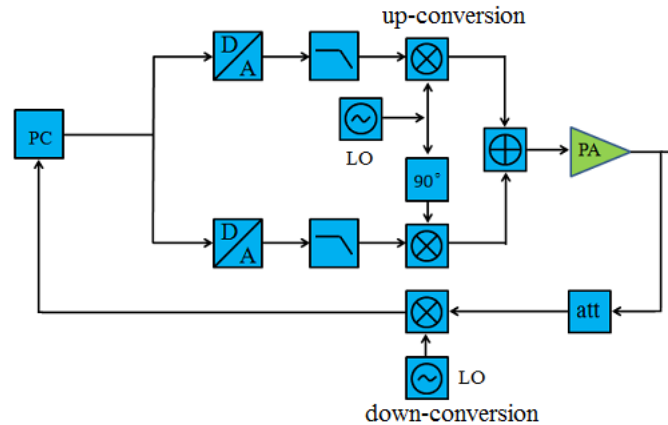
Figure 5.4.   A simplified data path of a wireless system

After selecting a proper power level and setting the operation frequency and the

correct bias voltages, the output data are ready to be measured. The RF output signal will be down-converted to baseband and stored in the memory of the VSA. The matlab scripts help us to move the output signal data from the VSA to the PC. Due to the transmission path, there is some latency between the original input signal and the obtained output signal. Therefore, time-alignment is required and Matlab can do this perfectly thanks to a built-in function *xcorr(x,y)*. This function returns the cross-correlation of two discrete-time sequences x and y, determining how long y is in late or in advance of x. And then y is time shifted to match x according to the result of function *xcorr(x,y)*. To increase the accuracy of the time delay, the two signal data could be upsampled by 200 before applying *xcorr()*, and then downsampled by 200.

5.2.3 DPD Builder

The captured output data of the PA are required for the model extraction. The model chosen is the memory polynomial model, described in Chapter 4. The model can be optimized by tuning the memory depth and the polynomial order. As my experimental experience, the more accuracy of the model, the better performance the DPD will be. After the extraction of the coefficients for the PA model, the model can be validated by comparing it with the experimental data. Moreover, we can check the AM/AM, AM/PM characteristics of the PA with the output and input data.

According to the *pth*-order post-inverse theory [71], the coefficients of the DPD can be directly estimated from the measured input and output of the PA. The performance of the DPD can also be optimized by adapting different memory depth and polynomial order. The DPD model has been successfully built with memory polynomial model. With the built PA and DPD model, we can test the performance of the DPD in the software Matlab by checking the AM/AM, AM/PM characteristics and the ACPR value of the PA with the DPD. Finally, the DPD algorithm will be implemented to the real PA. The input signal will first go through the DPD model where the signal has been predistorted before passing it to the ESG.

5.2.4 DPD Performance

The predistorted signal will be fed to the ESG as an excitation. Then the output data of the PA is re-captured following the same output data acquisition procedures

earlier described with the predistorted signal. The linear performance can be evaluated qualitatively from the AM/AM, AM/PM figures. The AM/AM characteristic is a straight line and the AM/PM characteristic is a flat line when the behavior of the PA is linear. Furthermore, the linear performance can be described quantitatively by the NMSE value between the normalized output and input signals. Generally, a value less than -30 dB for the NMSE indicates a very good linearity. ACPR, introduced in Chapter 4, is another important criterion in frequency domain to describe the linear behavior of the PA. The less power in adjacent channels, the better linear behavior the PA is.

5.3   Ideal DPD Simulated in Matlab

Every circuit design starts from the software simulation. If the ideal simulation can not satisfy the requirements, the design will certainly not be able to work properly for real physical applications. Figure 5.5 is an example of a PA characteristic generated by the memory polynomial model, the magenta line being the PA magnitude behavior and the blue line being the ideal linear behavior.
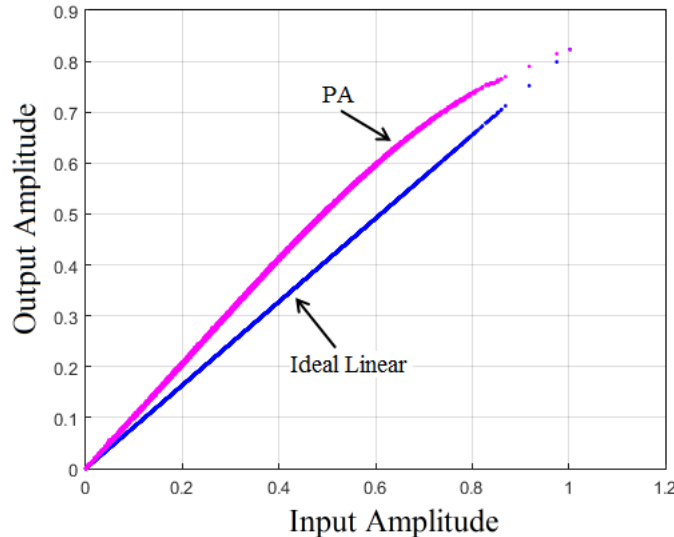
Figure 5.5.   A normalized PA magnitude characteristic generated by the memory polynomial model. The magenta line is the modeled PA magnitude behavior and the blue line is the ideal linear behavior.
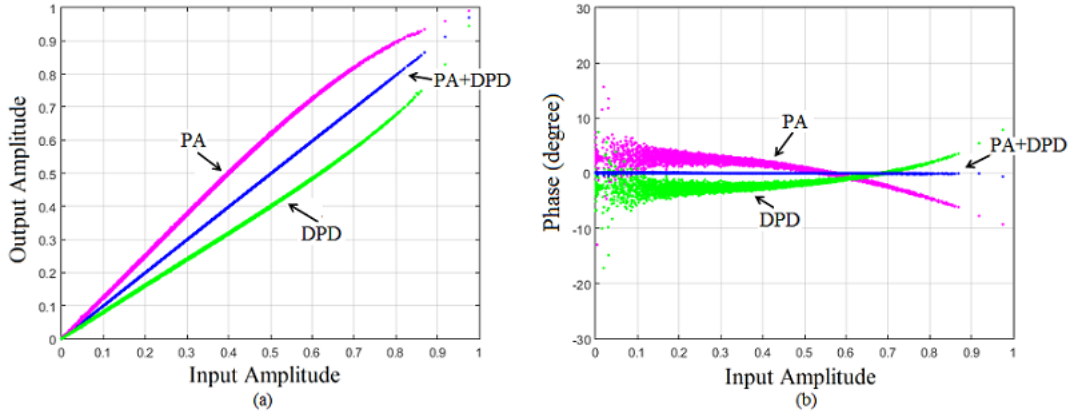
Figure 5.6.   Performances of DPD algorithm. (a) AM/AM performance: the magenta, green and blue lines are the PA, DPD and PA with DPD magnitude behavior, respectively. (b) AM/PM performance: the magenta, green and blue lines are the PA, DPD and PA with DPD phase behavior, respectively.
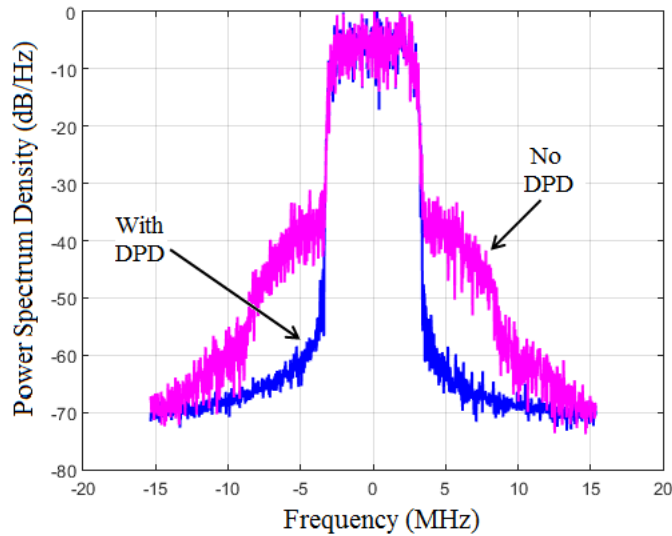
Figure 5.7.   Normalized power spectrum density: the magenta line is without DPD and the blue line is with DPD.

The DPD algorithm, based on a memory polynomial model with polynomial order of 6 and memory depth of 1, is employed to linearize the modeled PA characteristic in an ideal software environment. Figure 5.6 illustrates the AM/AM, AM/PM responses for the PA, the DPD and the PA with the DPD. Figure 5.6(a) is the magnitude characteristic in which the magenta, green and blue lines are the PA, DPD and PA with DPD magnitude behavior, respectively. And Figure 5.6(b) is the

phase characteristic in which the magenta, green and blue lines are the PA, DPD and PA with DPD phase behavior, respectively. From the figures of the magnitude and phase behaviors, the DPD algorithm has successfully linearized the PA, resulting a straight magnitude response and a flat phase response.

Figure 5.7 compares the output power spectrum with and without predistortion in which the magenta line is without DPD and the blue line is with DPD. The spectrum are normalized to the 0 dB power level. It is clear that the power of the adjacent channels has been reduced dramatically. The upper and lower ACPR has decreased from -37.1 dB to -59.6 dB and from -36.7 dB to -60.9 dB, respectively. Therefore, from the frequency domain criterion, the DPD algorithm works as well with an improvement of around 23 dB for ACPR.

5.4   Experimental Results

In this section, various PA examples, both in hybrid and MMIC features will be described. Figure 5.8 shows the realized experimental test bench for hybrid amplifiers. However, due to the tiny size of MMIC PAs, a specific test bench with needle probes to connect the PA is required for testing the MMIC PAs. The specific test bench is shown in Figure 5.9.
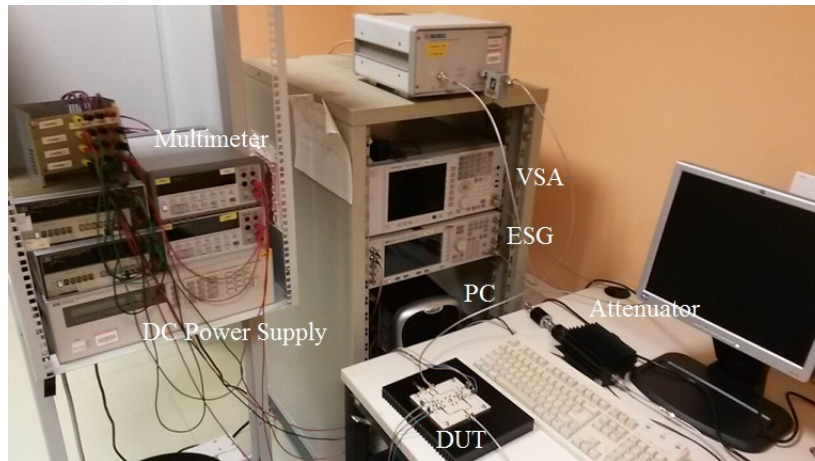
Figure 5.8.   Photograph of the DPD experimental test bench for hybrid power amplifiers.
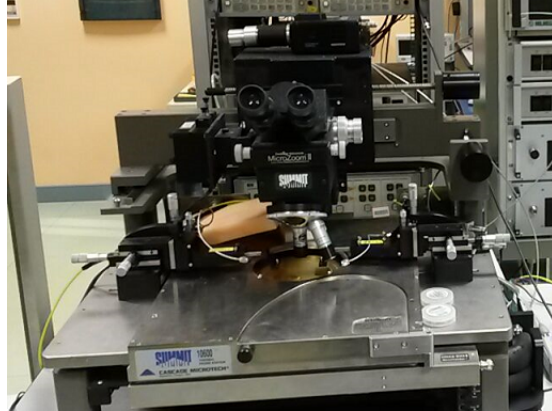
Figure 5.9.   Photograph of the test bench for MMIC power amplifiers.

5.4.1   Hybrid Class AB Amplifier

The device under test is a wideband class AB PA, exploiting N-section transformers at the output matching network. The key aspect of the proposed technique is its simplicity, making it a viable solution for wideband design. The amplifier CW characterization shows Power Added Efficiency (PAE) from 46% to 75% and more than 10 W of output power in a 145.5% fractional bandwidth with the central frequency at 2.2 GHz. The power amplifier has been realized exploiting a 10W GaN HEMT active device mounted on a microstrip hybrid passive structure. The entire circuit of the deigned PA is shown in Figure 5.10.
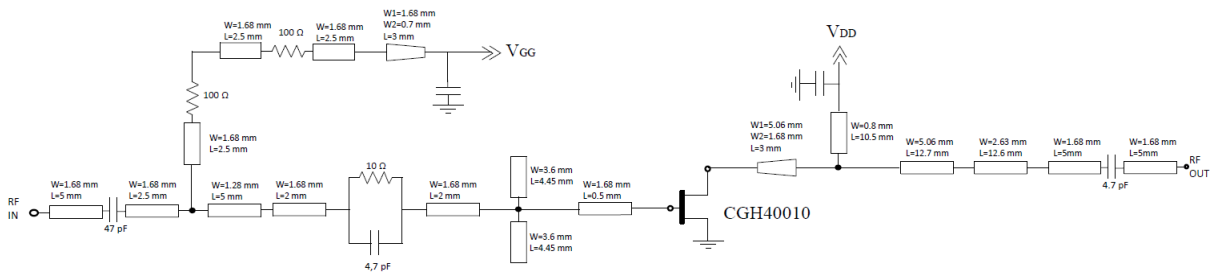
Figure 5.10.   Schematic of the wideband Class AB PA

Figure 5.11 is the picture of the realized wideband PA. The device CGH40010 from Cree has been selected in order to implement the power amplifier and verify the established approach. The amplifier is fabricated on a Taconic substrate with copper metallization (RF35 with $\epsilon_r = 3.5$, substrate height H = 0.76 mm, and metal
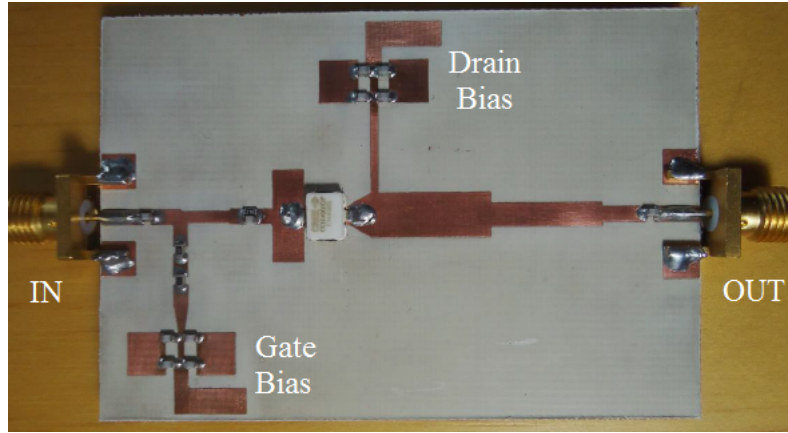
Figure 5.11.   Picture of the realized wideband Class AB PA

thickness t = 0.035 mm), mounted on a brass carrier.

The linearity of this wideband PA has been tested using a WiMAX signal with both 7 MHz and 28 MHz channel. The WiMAX signal has a 9.2-dB of peak-to-average power ratio (PAPR). The test operation frequency is chosen as 2.6 GHz. The spectrum mask, chosen according to the European Telecommunications Standards Institute (ETSI), is class 6LA for both cases. These spectrum masks limits are necessary for inter-system regulatory and performance requirements. The amplifier has been biased with: drain bias voltage of 28 V, gate bias voltage of -2.1 V, and drain current of 200 mA. However, the linearity requirements are not satisfied, since the mask compliance has been failed, see Figure 5.14 and Figure 5.15.

	7 MHz			28 MHz		
	P_{av} (dBm)	P_{max} (dBm)	Efficiency	P_{av} (dBm)	P_{max} (dBm)	Efficiency
No DPD	33.7	41.1	29.6%	33.7	41.9	29.5%
With DPD	33.7	40.8	29.6%	33.6	41.5	29.5%

Table 5.1.   Summary of the average power, maximum power and efficiency of the PA with and without DPD.

To improve the linearity, a DPD algorithm based on a static odd-polynomial model (5th order) has been adopted. Low memory effects have been manifested by the PA, thanks to careful bias network design and accurate mounting for thermal dissipation. Table 5.1 summarizes the average power, maximum power and efficiency of

the PA with and without DPD. We can conclude from the table that the DPD do not degrade the output power and efficiency performance of the PA since the output average and maximum power and the drain efficiency remain almost the same.
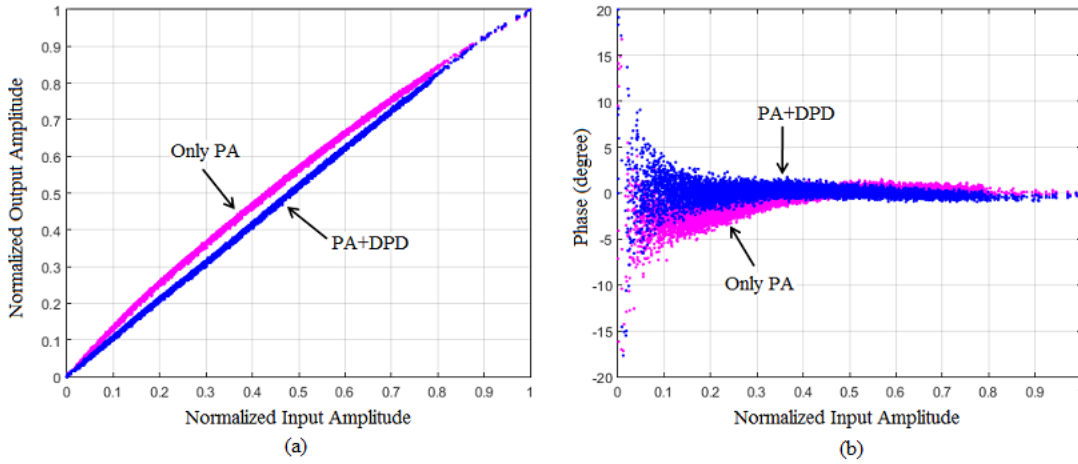
Figure 5.12.   Instantaneous magnitude and phase response for the 7 MHz channel WiMAX signal: (a) AM/AM characteristic, (b) AM/PM characteristic.
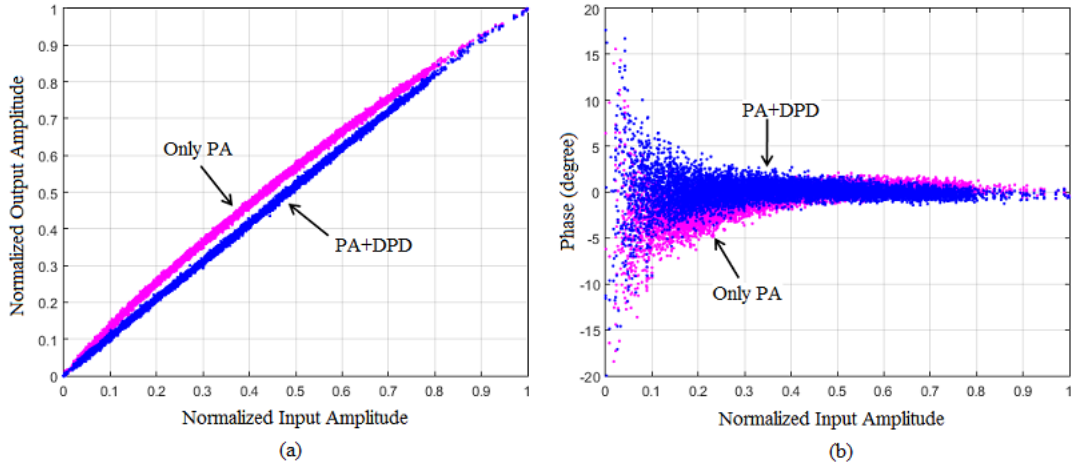
Figure 5.13.   Instantaneous magnitude and phase response for the 28 MHz channel WiMAX signal: (a) AM/AM characteristic, (b) AM/PM characteristic.

Figure 5.12, Figure 5.13 are the instantaneous magnitude and phase responses with
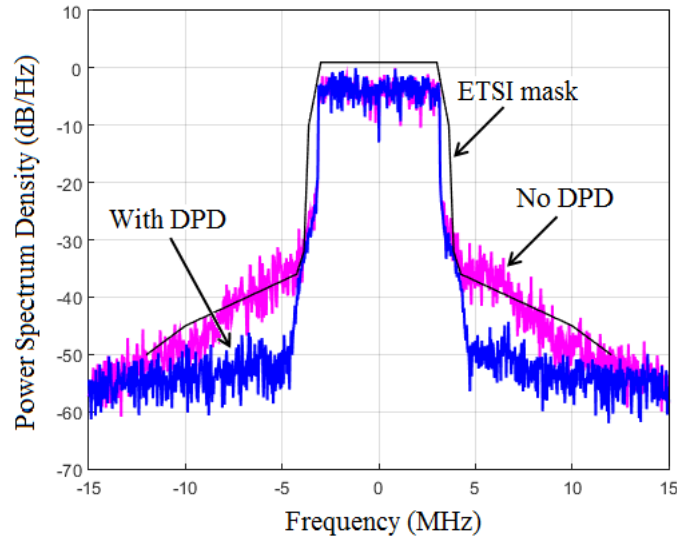
Figure 5.14. Normalized power spectrum density of the designed wideband PA with a 7 MHz WiMAX test signal with 9.2 dB PAPR. The black line is the ETSI emission mask.
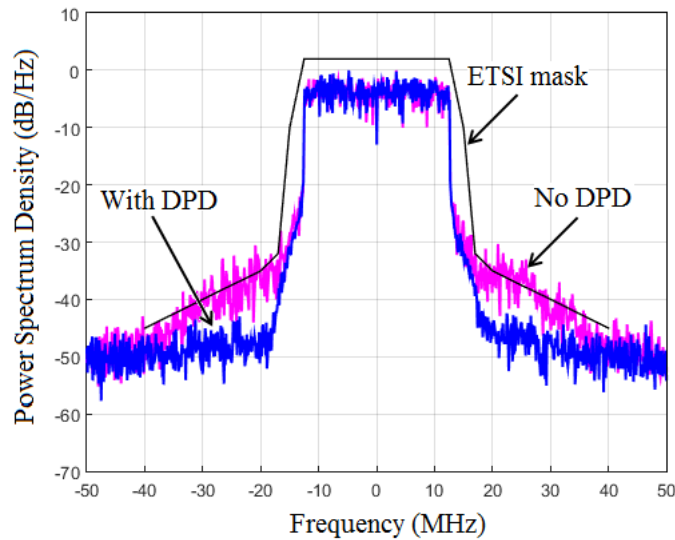
Figure 5.15. Normalized power spectrum density of the designed wideband PA with a 28 MHz WiMAX test signal with 9.2 dB PAPR. The black line is the ETSI emission mask.

7 MHz and 28 MHz channel WiMAX signals, respectively. In both figures, the magenta points represent the response of the PA and the blue points are for the PA

with DPD. The output of the PA excited by the 28 MHz channel WiMAX signal displays more memory behavior than that of 7 MHz signal. However, in both cases, the PA has been linearized with a more straight magnitude response and more flat phase response as shown in the figures.

The resulting measured output power spectrum density with DPD is shown in Figure 5.14 and Figure 5.15. An ACPR of around 30 dBc is obtained without applying digital predistortion for both cases. However, after employing the DPD algorithm, an ACPR of 40 dBc is obtained. Moreover, the mask compliance with the ETSI standards could be achieved for most adjacent channel.

Table 5.2 summarizes the linearization performance of the DPD concerning the ACPR and NMSE, where the NMSE is evaluated between the normalized input and output data of the PA. 13 dB improvement of both the lower and upper sideband ACPR has been obtained for the WiMAX signal with 7 MHz channel while the improvement is of 10 dB for the 28 MHz channel signal. The NMSE improvement is around 9 dB.

	7 MHz			28 MHz		
	ACPR (dBc)		NMSE (dB)	ACPR (dBc)		NMSE (dB)
	Lower Sideband	Upper Sideband		Lower Sideband	Upper Sideband	
No DPD	-34.7	-35.1	-25.1	-34.5	-34.7	-24.1
With DPD	-47.6	-48.6	-34.1	-43.9	-44.6	-32.9
Improvement	12.9	13.5	9.0	9.4	9.9	8.8

Table 5.2.   Summary of performance improvements of the DPD algorithm in terms of the ACPR and NMSE values.

If we back-off the operation power to the level where the linearity performance of the PA without DPD is similar to that with DPD. The results are shown in Table 5.3. When the linearity performance with power back-off is similar to the DPD technique in terms of the NMSE value, the power level and the efficiency of the PA are severely lowered. The average output power, maximum power and efficiency with 7 MHz channel WiMAX signal are 23.5 dBm, 32.3 dBm and 4.5%, respectively. In the mean while, for the 28 MHz channel signal, the average output power, maximum power and efficiency are 23.6 dBm, 33.9 dBm and 4.5%, respectively.

7 MHz					
	1	2	3	4	5
P_{av} (dBm)	30.5	28.9	27.1	25.6	23.5
P_{max} (dBm)	39.2	37.6	35.4	33.8	32.3
Efficiency	18.2%	13.5%	9.8%	6.9%	4.5%
NMSE (dB)	-25.5	-26.5	-28.1	-30.7	-32.0
28 MHz					
	1	2	3	4	5
P_{av} (dBm)	30.7	28.9	27.4	25.4	23.6
P_{max} (dBm)	40.1	38.6	37.2	35.4	33.9
Efficiency	18.9%	13.5%	10.2%	6.7%	4.5%
NMSE (dB)	-25.2	-25.8	-27.0	-28.2	-30.0

Table 5.3.   The power back-off performance: average power, maximum power and efficiency.

5.4.2   Hybrid Doherty Amplifier

Another example is a hybrid power amplifier, more specifically a wideband Doherty PA [16] designed for the 3 - 3.6 GHz frequency range (18% bandwidth) [72]. A simple technique based on wideband compensators inserted at the output of the peak and main cells is adopted in this PA design. Moreover, second-harmonic tuning of the main amplifier has been implemented at the upper bandwidth limit to help gain equalization versus frequency. The active device exploited in the microstrip hybrid circuit implementation is a packaged GaN HEMT on SiC (CGH40010 from Cree inc.), with typical output power of 10 W in the selected band.

Figure 5.16 shows the complete schematic of the realized amplifier, also including the wideband source input matching networks at the fundamental, that was designed to minimize the input mismatch under large-signal conditions. The input splitter was implemented as a branch line featuring a small imbalance between the main and peak ports; as for the $\lambda/4$ transformer, the bandwidth achieved with this simple solution was adequate.

The amplifier is fabricated on a Taconic substrate with copper metallization (RF35 with $\epsilon_r = 3.5$, substrate height H = 0.76 mm, and metal thickness t = 0.035 mm), and mounted on a brass carrier (see Figure 5.17). The PA is biased at $V_{DS} = 28$ V and $V_{GS} = -2.7$ V for the main, and $V_{DS} = 28$ V and $V_{GS} = -7$ V for the peak.

The linearity of this hybrid Doherty PA has been tested using a 7 MHz channel

Figure 5.16.　Complete scheme of the Doherty amplifier.

Figure 5.17.　Picture of the realized wideband Doherty PA.

I/Q signal with raised-cosine filter, roll-off of 0.2, 256-QAM modulation and PAPR of 7.4 dB. The test operation frequencies is chosen as 3.4 GHz. The spectrum mask, chosen according to the ETSI, is class 6LA.

To improve the linearity, a DPD based on a memory polynomial model with polynomial order of 6 and memory depth of 1 has been adopted. 0.8 dB power back-off has been introduced to keep the spectrum within the mask limits. Table 5.4 summarizes the average power, maximum power and efficiency of the PA with and without DPD.

	P_{av} (dBm)	P_{max} (dBm)	Efficiency
No DPD	36.1	42.4	39.1%
With DPD	35.9	42.8	39.2%
With DPD (0.8 dB OBO)	35.0	42.0	36.8%

Table 5.4.   Summary of the average power, maximum power and efficiency of the PA with and without DPD.

Figure 5.18 is the instantaneous magnitude and phase response with a 7 MHz channel I/Q signal. In the figure, the magenta points represent the response of the PA and the blue points are for the PA with DPD. The blue points form a straight line in Figure 5.18(a) and the blue points are flat around 0 degree in Figure 5.18(b), indicating that the PA has been linearized by the DPD algorithm.
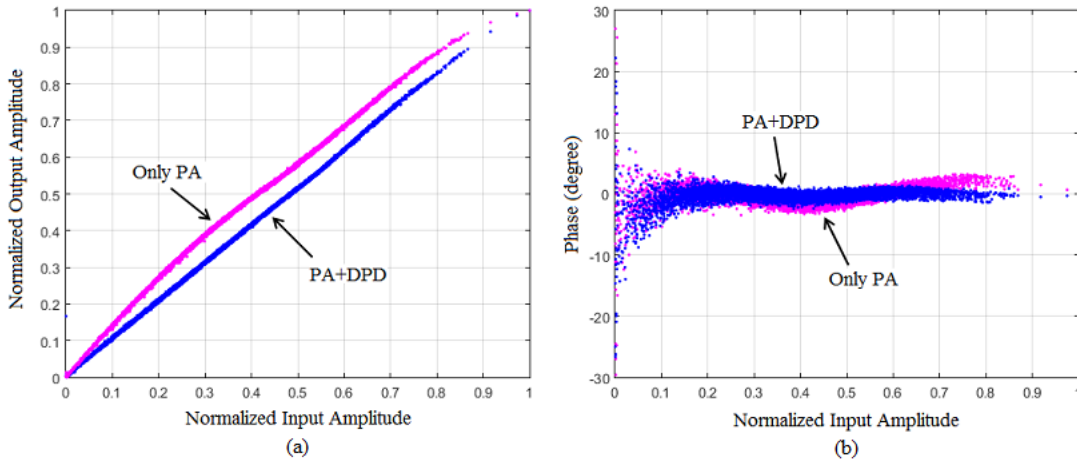
Figure 5.18.   Instantaneous magnitude and phase response for the 7 MHz channel I/Q signal: (a) AM/AM characteristic, (b) AM/PM characteristic.

The resulting measured output power spectrum density with DPD is shown in Figure 5.19. An ACPR of around 33 dBc is obtained without applying digital predistortion. In order to keep the spectrum within the mask limits, 0.8 dB of power back-off is

introduced. After employing the DPD algorithm, an ACPR of 45 dBc is obtained. Moreover, the mask compliance with the ETSI standards could be achieved by sacrificing a little output power.

Figure 5.19.   Normalized power spectrum density of the designed PA with a 7 MHz I/Q test signal with 7.4 dB PAPR. The black line is the ETSI emission mask.

	ACPR (dBc)		NMSE (dB)
	Upper Sideband	Lower Sideband	
No DPD	-33.7	-33.2	-20.5
With DPD	-40.2	-41.0	-31.5
With DPD (0.8 dB OBO)	-45.2	-47.6	-33.5

Table 5.5.   Summary of the performance improvements of the DPD algorithm in terms of the ACPR and NMSE values.

Table 5.5 summarizes the linearization performance of the DPD concerning the ACPR and NMSE, where the NMSE is evaluated between the normalized input and output data of the PA. 12 dB improvement of the upper sideband ACPR and 14 dB of the lower sideband ACPR have been obtained for the I/Q signal with 7 MHz channel. The NMSE improvement is around 13 dB.

5.4.3   MMIC Class AB Amplifier

The power amplifier under test is a combined class AB power amplifier [73] based on Monolithic Microwave Integrated Circuit (MMIC) technology. It is realized on TriQuint GaN HEMT on SiC MMIC for 7-GHz microwave backhaul applications. This PA is designed to maximize the back-off efficiency while limiting amplitude and phase distortion. A second-harmonic tuning strategy has been adopted in this class AB PA, to relax the constraints on the resonant behavior of the output combiner. Indeed, the back-off efficiency has been maximized, however linearity degradation has been observed as well. The complete schematic is shown in Figure 5.20 and the microscope photograph of the MMIC class AB amplifier is given in Figure 5.21.
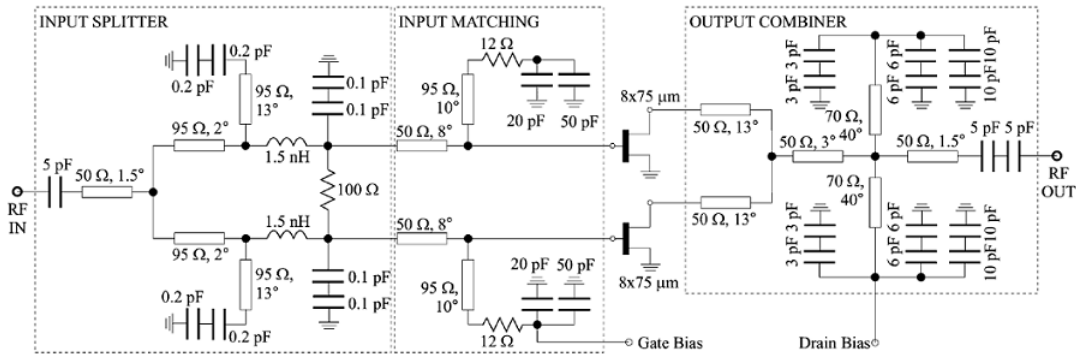
Figure 5.20.   Complete scheme of the harmonic tuned class AB amplifier.
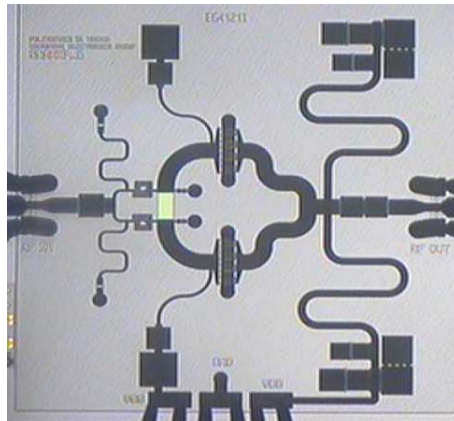
Figure 5.21.   Microscope photograph of the harmonic tuned MMIC amplifier.

The MMIC power amplifier is biased with 30 V drain voltage and 60 mA drain

current. The linearity of this MMIC has been tested using a 7 MHz channel I/Q signal with raised-cosine filter, roll-off of 0.2, 256-QAM modulation and PAPR of 7.4 dB. The test operation frequencies is chosen as 7 GHz. The spectrum mask, chosen according to the ETSI, is class 6LA.

The base-band system is integrated in a real-time power measurement bench (Figure 5.9) and the bench is calibrated through standard scattering calibration, and with a C.W. power calibration method [76]. Then, with a through under measurement, the VSA is connected at the bench output, and a single tone test signal is applied at the input.

To improve the linearity, a DPD based on memory polynomial model with polynomial order of 4 and memory depth of 2 has been adopted. Table 5.6 summarizes the average power, maximum power and efficiency of the PA with and without DPD. We can conclude that the DPD do not degrade the output power and efficiency performance since average and maximum output power and drain efficiency remain almost constant with and without DPD.

	P_{av} (dBm)	P_{max} (dBm)	Efficiency
No DPD	29.3	35.7	25.3%
With DPD	29.4	36.4	25.7%

Table 5.6. Summary of the average power, maximum power and efficiency of the PA with and without DPD.

Figure 5.22 is the instantaneous magnitude and phase response of the PA with a 7 MHz channel I/Q signal. In the figure, the magenta points represent the response of the PA and the blue points are for the PA with DPD. The blue points form a straight line in Figure 5.22(a) and the blue points are flat around 0 degree in Figure 5.22(b), indicating that the PA has been linearized by the DPD algorithm.

The resulting measured output power spectrum density with DPD is shown in Figure 5.23. An ACPR of around 30 dBc is obtained without applying digital predistortion. However, after employing the DPD algorithm, an ACPR of 45 dBc is achieved. Moreover, the mask compliance with the ETSI standards is achieved.

Table 5.7 summarizes the linearization performance of the DPD concerning the ACPR and NMSE, where the NMSE is evaluated between the normalized input and output data of the PA. 12 dB improvement of both the upper sideband ACPR and the lower sideband ACPR has been obtained for the I/Q signal with 7 MHz channel.

Figure 5.22. Instantaneous magnitude and phase response for the 7 MHz channel I/Q signal: (a) AM/AM characteristic, (b) AM/PM characteristic.
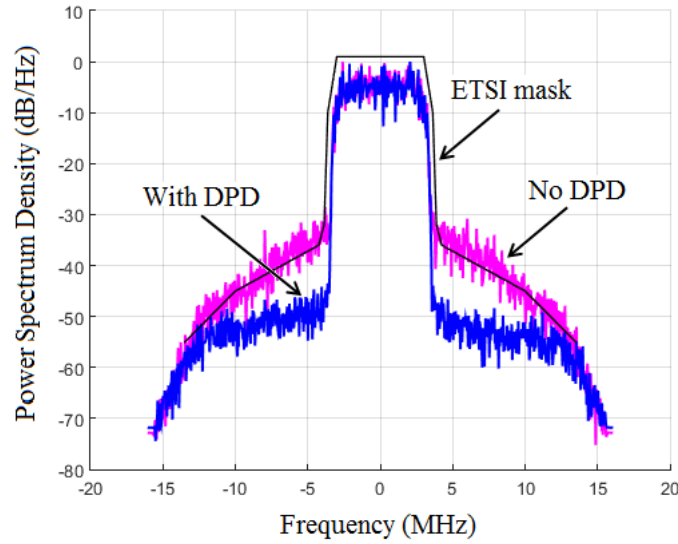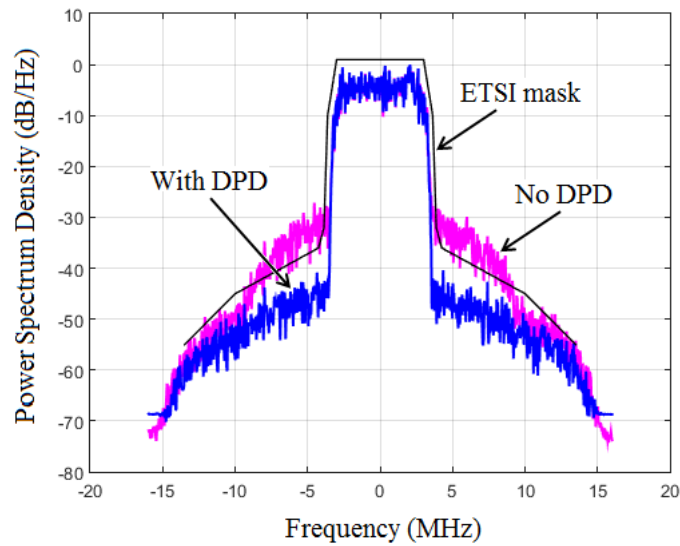

Figure 5.23. Normalized power spectrum density of the designed PA with a 7 MHz I/Q test signal with 7.4 dB PAPR. The black line is the ETSI emission mask.

The NMSE improvement is around 13 dB.

	ACPR (dBc)		NMSE (dB)
	Upper Sideband	Lower Sideband	
No DPD	-31.7	-31.8	-22.3
With DPD	-43.6	-44.1	-34.3
Improvement	11.9	12.3	12

Table 5.7.   Summary of the performance improvements of the DPD algorithm in terms of the ACPR and NMSE values.

5.4.4   Two-stage Amplifier

The power amplifier under test is a MMIC Doherty power amplifier [74, 75] with highly efficient driver stages on both the main and auxiliary branches, as shown in Figure 5.24. The optimized driver stages are designed to boost gain with minimal impact on power-added efficiency. Therefore, the output signal is the response of both the driver and the Doherty PA, which can be seen as PAs with multi-stages. The design of the Doherty PA is based on the 0.15 μm PWR pHEMT MMIC process of TriQuint Semiconductors and Figure 5.25 is the microscope image of the manufactured Doherty PA.
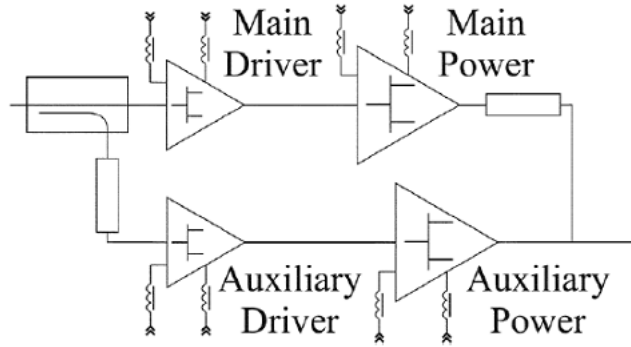
Figure 5.24.   A Doherty PA with drivers for both main and auxiliary branches.

The devices are biased at 6 V drain voltage: in the main branch, the bias current is 20 mA for the driver and 100 mA for the power device; in the auxiliary branch, the power device is biased with 30 mA while the driver is in class C, with -1.9 V gate voltage. The test operation frequency is 24 GHz.

The linearity of this Doherty PA has been tested using an I/Q signal with both 7 MHz and 28 MHz channels. The I/Q signal is with raised-cosine filter, roll-off of
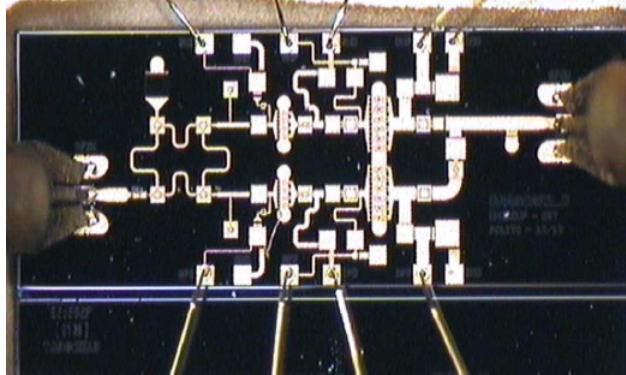
Figure 5.25.   Microscope image of the manufactured DPA.

0.2, 256-QAM modulation and PAPR of 7.4 dB. The spectrum mask, chosen according to the ETSI, is class 6LA for both cases.

To improve the linearity, a DPD algorithm based on memory polynomial model has been adopted. Since the test device exhibits a large memory effect, the memory depth is chosen as 1 for 7 MHz case and 2 for 28 MHz case with the same polynomial order of 6.

Figure 5.26 and Figure 5.27 are the instantaneous magnitude and phase responses with 7 MHz and 28 MHz channel I/Q signals, respectively. In both figures, the magenta points represent the response of the PA and the blue points are for the PA with DPD. The memory effect is more noteworthy than in the previous three test PAs, due to the complex stage PA architecture. Moreover, the output of the PA excited by the 28 MHz channle I/Q signal displays more memory behavior than that of 7 MHz signal. However, in both cases, the PA has been linearized with a more straight magnitude response and more flat phase response as shown in the figures, comparing to the characteristic with only PA.

The resulting measured output power spectrum density with DPD is shown in Figure 5.28 and Figure 5.29. An ACPR of around 30 dBc is obtained without applying digital predistortion for both cases. However, after employing the DPD algorithm, an ACPR of 40 dBc is obtained. Moreover, the mask compliance with the ETSI standards could be achieved for most adjacent channel. The average power of this PA with predistortion is 23.5 dBm and the drain efficiency is higher than 14% for the input signals of both 7 MHz and 28 MHz bandwidth.

Table 5.8 summarizes the linearization performance of the DPD concerning the
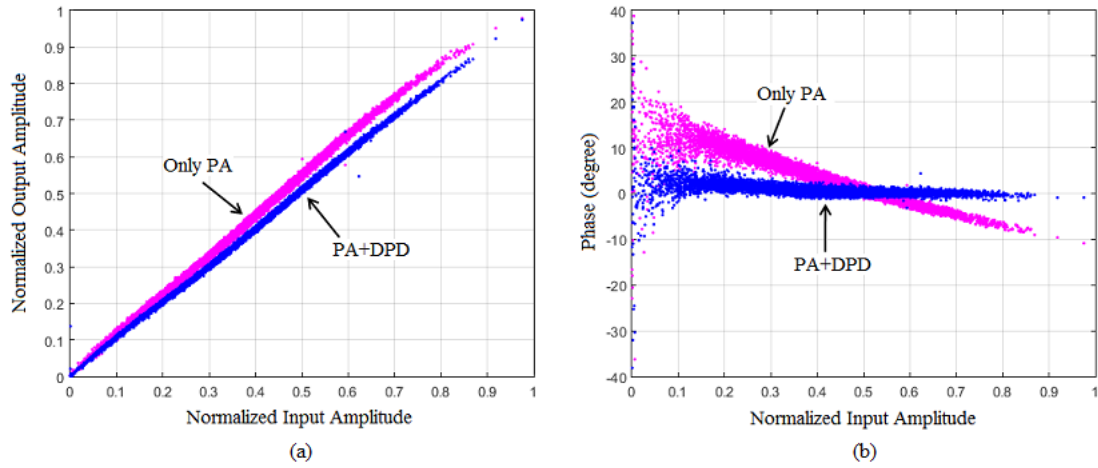
75
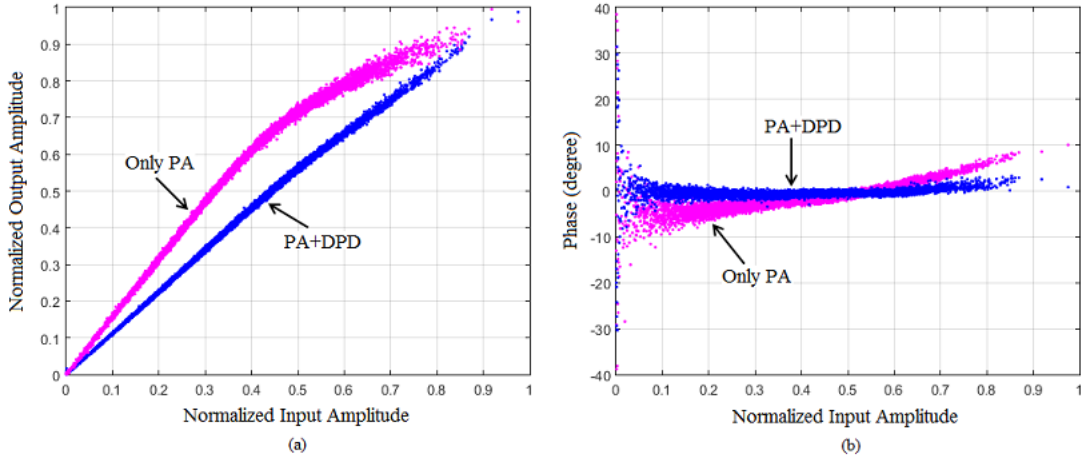
Figure 5.26.   Instantaneous magnitude and phase response for the 7 MHz channel I/Q signal: (a) AM/AM characteristic, (b) AM/PM characteristic.
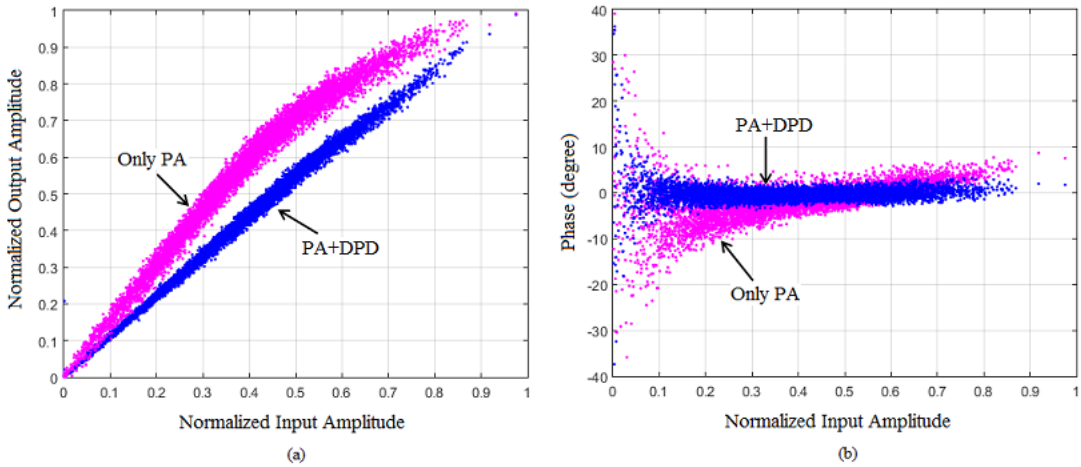
Figure 5.27.   Instantaneous magnitude and phase response for the 28 MHz channel I/Q signal: (a) AM/AM characteristic, (b) AM/PM characteristic.

ACPR and NMSE, where the NMSE is evaluated between the normalized input and output data of the PA. 11 dB improvement of both the lower and upper sideband ACPR has been obtained for the I/Q signal with 7 MHz channel while the improvement is 9 dB for the 28 MHz channel signal. The NMSE improvement is around 11 dB and 9 dB for the 7 MHz and 28 MHz cases, respectively.
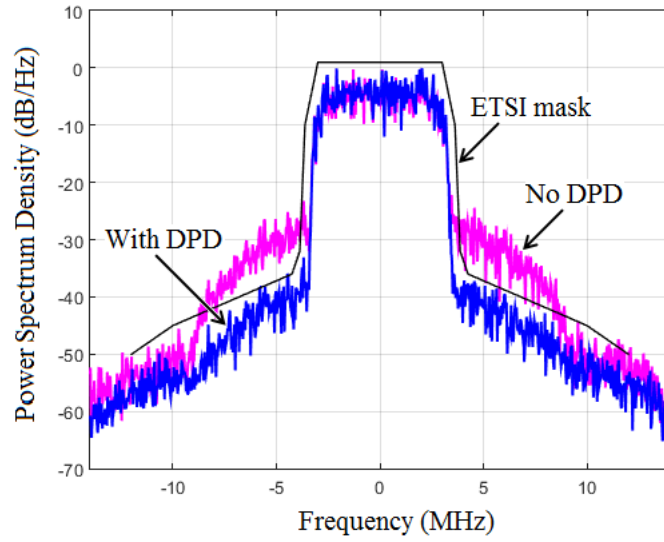
Figure 5.28.   Normalized Power Spectrum Density of the designed wideband PA with a 7 MHz I/Q test signal with 7.4 dB PAPR. The black line is the ETSI emission mask.
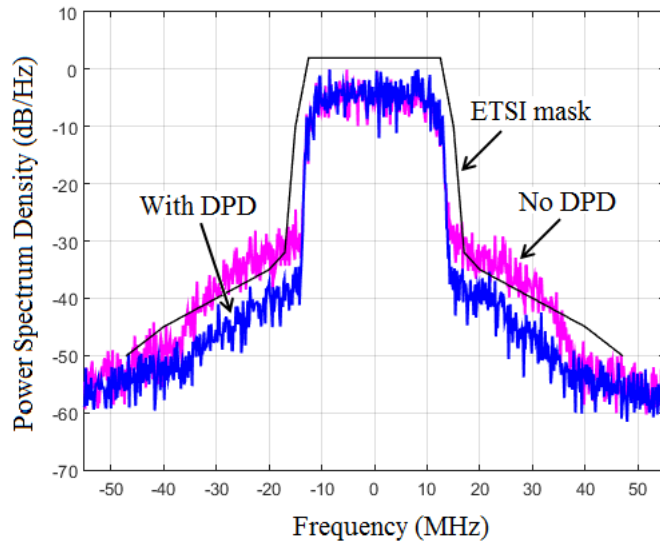
Figure 5.29.   Normalized power spectrum density of the designed wideband PA with a 28 MHz I/Q test signal with 7.4 dB PAPR. The black line is the ETSI emission mask.

77

	7 MHz			28 MHz		
	ACPR (dBc)		NMSE (dB)	ACPR (dBc)		NMSE (dB)
	Lower Sideband	Upper Sideband		Lower Sideband	Upper Sideband	
No DPD	-29.5	-29.4	-19.6	-31.2	-30.7	-20.1
With DPD	-40.4	-40.4	-30.8	-40.0	-39.3	-28.9
Improvement	10.9	11.0	11.2	8.8	8.6	8.8

Table 5.8.   Summary of performance improvements of the DPD algorithm in terms of the ACPR and NMSE values.

5.5   Conclusion

In this chapter, the simulation of a digital predistortion system based on memory polynomial model has been introduced and validated. The DPD algorithm has successfully linearized some PAs without degrading the output power and efficiency performance. Different test bench setups were described for both hybrid and MMIC PAs, including the instruments, the calibration. However, the procedures of the DPD implementation are similar for hybrid and MMIC PAs.

The simulation results prove that the DPD algorithm works well for various types of PAs excited by different input signals: different architectures, such as class AB PAs and Doherty PAs, different technologies, such as hybrid PAs and MMIC PAs, different operation frequencies, such as 3.6 GHz, 7 GHz, 24 GHz. The DPD algorithm can also linearize PAs with several stages. The test signals are WiMAX signal with 9.2 dB PAPR and I/Q signal with 7.4 dB PAPR, with both 7 MHz and 28 MHz signal bandwidth. For all the tests, the improvement of the ACPR and NMSE can reach 10 dB, with a best case being as high as 14 dB.

Chapter 6

Hardware Implementation based on FPGA

The solution introduced in the previews chapter, based on the software environment (Matlab), is suitable for laboratory usage. However, it cannot be used for real-time applications. In addition, another disadvantage of the software solution is the difficulty in implementing the adaptive algorithm. Therefore, to address the industrial requirements, commercial products should be used to replace the software in the DPD test system. field-programmable gate array (FPGA) is a good choice for implementing the DPD technique. It has many advantages in digital signal processing, including high speed processing, flexible implementation, high reliability and parallelism computation. The proposed DPD technique is not solely limited to FPGA and it can also be implemented based on other commercial products.

There are two critical points during the translation from software to hardware that need to be taken into consideration: accuracy and speed. Both these two points can be improved by the parallel operation mechanisms. The pipelined architecture used in FPGA can reduce the propagation time and synchronize the I/Q signals in different paths. There are two commonly used techniques to implement the DPD algorithm into FPGA, namely, the lookup table (LUT) method and the direct structure with multipliers and adders. The LUT method stores the contents of a relation in the memory in advance and retrieve it according to the input. This method saves the logic resources at the expense of on-chip memory. In other hand, the direct structure with multipliers and adders usually takes the advantage of the pipeline architecture of the FPGA to increase the throughout. This method needs more logic resources than the LUT method.

6.1   FPGA Introduction

The FPGA concept starts from programmable read-only memory (PROM) and programmable logic devices (PLDs) with programmable capability. The first industrial re-programmable logic device was manufactured by Altera in 1984. Another company, Xilinx, invented the first commercially viable field-programmable gate array in 1985. Since then, the FPGA industry accelerates with an explosive development and the potential benefit of using FPGA in telecommunication applications has been discovered. Nowadays, a more recent product from Xilinx, Virtex-7, has round 2 million logic cells with a transceiver speed as high as 28.05 Gb/s. The high performance and relative low price make FPGA competitive in various field applications.

An FPGA is an integrated circuit which is re-programmable by designers. It consists of an array of configurable logic blocks (CLB), switch matrices for programmable interconnections and on-chip memory. The FPGA board used in our laboratory is an Xtreme DSP Virtex-4 Development kit made by Nallatech, as shown in Figure 6.1. The kit has two independent ADC channels and two DAC channels with 14-bit resolution. A 105 MHz oscillator is the source of the clock or an external clock can be brought to the board. The board is connected to the PC via Peripheral Component Interconnect (PCI) in our experiments. Some important components on the FPGA will be introduced in detail in the following sections.
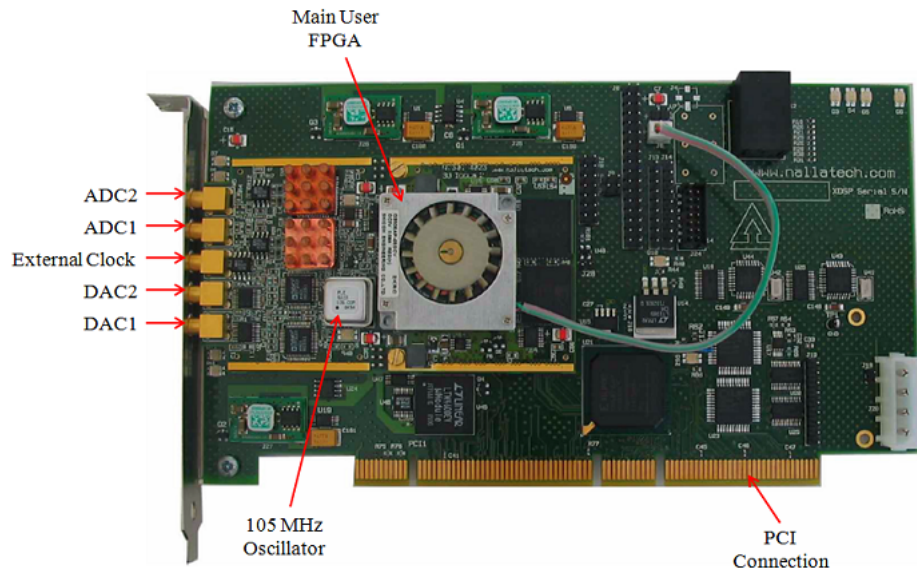
Figure 6.1.   Front view of virtex 4 board physical layout

6.1.1 Configurable Logic Blocks

Configurable Logic Blocks (CLBs), sometimes referred to logic cells, are the fundamental logic components in FPGA for implementing the logic circuits, both sequential and combinatorial [79]. They are connected by the switch matrix which can be programmable, as shown in Figure 6.2. All of the CLBs are identical and each CLB contains 4 interconnected slices. Furthermore, each slice is made of 2 LUTs as logic function generator, 2 storage elements, multiplexers, carry logic and arithmetic gates. For instance, the board used in our laboratory has in total 15,360 slices, 30,720 LUTs and 30,720 flip-flops.

In Virtex-4 FPGA, the LUT has 4 independent inputs, therefore, it is able to generate any four-input Boolean function. In addition, with the multiplexers, the LUTs can be combined together to implement any function with five, six, seven or even eight inputs in one CLB. There are two type of digital circuits that can be configured as storage elements: D flip-flops and latches. In Virtex-4 FPGA, both of these two circuits are used for storing bits.
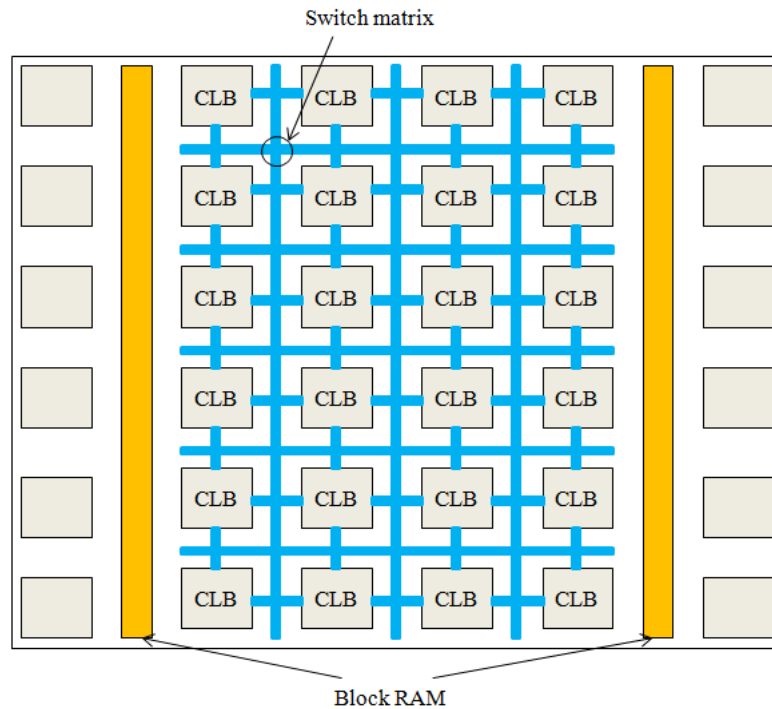
Figure 6.2. Schematic of a FPGA components distribution.

6.1.2   ADC and DAC

The Virtex-4 FPGA boad has 2 independent ADC channels and 2 DAC channels [80] as shown in Figure 6.1. The ADC channel has 14-bit resolution with 2's complement format and the data sampling rate is 105 MSample/s. The full scale range is -1.1 V to 1.1 V and the recommended maximum input signal magnitude is +/- 1 V to guarantee the ADC performance. Several clock sources can be used for the ADC sampling, including the 105 MHz crystal oscillator on the board and the external input clock source.

In the other hand, the board has also two independent DAC channels with the same 14-bit resolution and the same full scale range as the ADC channels. The original input signal is digitized by the ADC and then predistorted in FPGA. Since the final predistorted signal should be fed to the ESG, the digital signal has to pass through the DAC to be converted back to be as analog signal. Both the 2 ADC channels and the 2 DAC channels are connected with MCX connectors.

The DACs hardware operates for an input in offset binary where all '0' means the lowest value and all '1' is the highest value. Moreover, there is an inverting op-amp at the output of the DACs, converting all '0' to the highest value and all '1' to the lowest value. However, the digital signal processed in FPGA is in 2's complement format. Fortunately, it is possible to convert the 2's complement format to the offset binary format directly with the following VHDL scripts, where ADC1, ADC1 are the digital signals in 2's complement and DAC1, DAC2 are in offset binary format for the two channels.

DAC1 <= not (not ADC1(13) & ADC1(12 downto 0));
DAC2 <= not (not ADC2(13) & ADC2(12 downto 0));

6.1.3   Block RAM

The block RAM is a dedicated memory with single-port or dual-port, supporting synchronous read and write. The Virtex-4 FPGA provides a large number of block RAMs with each block RAM storing 18 Kbits data. In addition, more block RAMs can be combined to a larger size without losing the speed. The board used in our laboratory has 192 18-Kbits block RAMs. In Virtex-4 board, the block RAMs are placed in columns, as shown in Figure 6.2.

Figure 6.3 is the dual-port block RAM data flow of Virtex-4 board [79]. DIA and

DIB are the data input bus and ADDRA and ADDRB are the address bus. The write operation is controlled by the write enable WEA, WEB. The write and read operations are synchronous, therefore, clock CLKA, CLKB are required. At a clock edge, the data will be stored at the address determined by the address bus. The read operates in the same manner, reading data from the memory according to the address of the address bus. Data can be written to one or both ports and read from one or both ports.
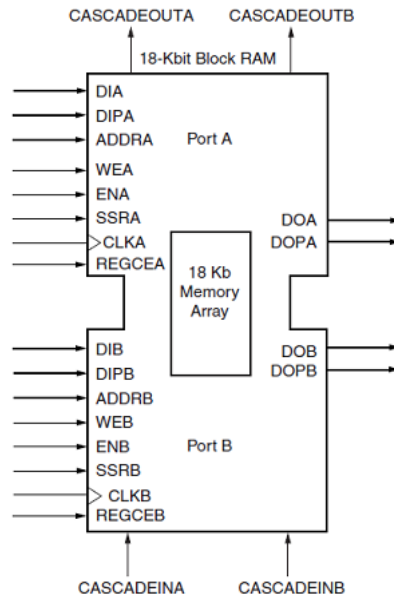
Figure 6.3.   Dual-port Block RAM Data Flow.

WRITE_FIRST, READ_FIRST, and NO_CHANGE are the three different operation modes. However, in our work, the block RAM can be seen as a read-only-memory (ROM) to implement a function like LUT. The write operation is not necessary and the content of the block RAMs can be initialized by a *coe* file. Why block RAM is used here to implement the LUT function instead of the distributed memory? It is because that the memory size need is large. The distribute RAM is configured with LUTs of the logic blocks and distributed throughout the FPGA. Therefore, when comes to large size memory, a huge amount of distribute RAMs are combined which may cause larger wiring delay. Fortunately, block RAMs have no such delay when more than one block are combined.

6.2 FPGA Design Procedures

The software we use for FPGA design is the Xilinx ISE design suit. As illustrated in Figure 6.4, the FPGA design procedures with Xilinx ISE include design entry, design synthesis, design implementation and device programming. Each of the procedures can be verified by behavioral simulation, functional simulation and in-circuit verification, respectively.
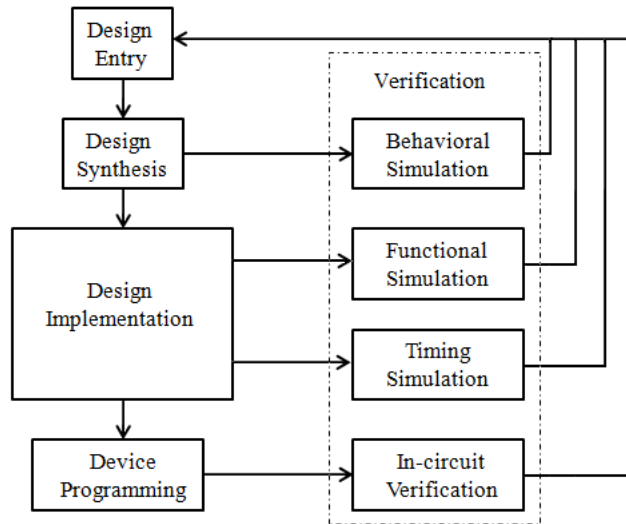
Figure 6.4.   FPGA design flow chat.

In design entry stage, the required circuit is described by schematic or hardware description language (HDL). We choose the VHDL, which is more flexible and widely used, to describe the design functionality. After creating the VHDL file, we can turn to next procedure and synthesize the design. This procedure is to check the VHDL code syntax and analyze the hierarchy of the design. The abstract VHDL code will be turned into a netlist of logic circuits. The resulting circuit has to be verified with the behavioral simulation to check if it works as intended. The behavioral simulation is performed to verify the RTL code. After synthesis, we can run the implementation which includes translate, map, place and route. Translate is to merge all the netlists and constrains into a Xilinx design file, map is to locate the design circuit into the resources of the FPGA board and place and route is to optimize the design to the time constrain. Finally, a bitstream file will be generated in the device programming procedure. The bitstream file can be downloaded to the FPGA board via software FUSE. The in-circuit verification can be done by software ChipScope Pro.

6.3 Implementation Methods

6.3.1 LUT method

LUT method is an efficient solution to implement the polynomial function in FPGA. Although the input signal is generated randomly, the value of the signal is usually limited in a range. Therefore, all the power terms with different orders can be calculated in advance and stored in the memory at the address associated to the magnitude of the complex-envelope input signal. In other words, the LUT entries are indexed by the input signal magnitude. It is easy and fast to retrieve the value in the LUT later with the utilization of few logic resources. Figure 6.5 depicts the basic cell of LUT method for polynomial functions, where $|\cdot|$ means the magnitude of the signal. The corresponding value of the polynomial is retrieved according the address which is the square of the input signal magnitude.
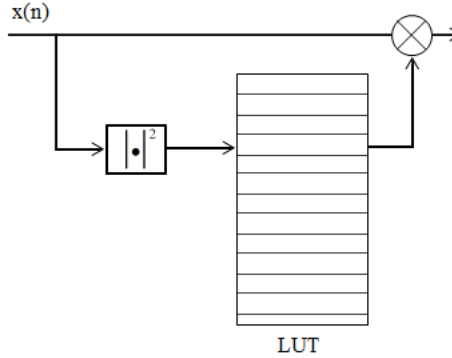
Figure 6.5.   Basic cell of LUT method for polynomial

A memory polynomial model is given as

$$
\begin{aligned}
y(n) =\, & x(n) \left[a_{00} + a_{01}|x(n)|^2 + \cdots + a_{0p}|x(n)|^{2(p-1)} \right] + \\
& x(n-1) \left[a_{10} + a_{11}|x(n-1)|^2 + \cdots + a_{1p}|x(n-1)|^{2(p-1)} \right] + \\
& \vdots \\
& x(n-m) \left[a_{m0} + a_{m1}|x(n-m)|^2 + \cdots + a_{mp}|x(n-m)|^{2(p-1)} \right]
\end{aligned}
\tag{6.1}
$$

One conventional solution is to build LUTs for each of the polynomial terms, e.g.

85

$|\cdot|^2$, $|\cdot|^4, \cdots$, as the case in [81–83]. This method is more flexible since only the coefficients are need to be changed and the contents of the LUTs keep the same when the test condition changes. However, more memory is required for implementing the LUTs and extra multipliers and adders are needed to combine all the polynomial terms into the final result. It is suitable for adaptive digital predistortion applications in which the model coefficients have to been updated repeatedly. Figure 6.6 is an example architecture of this LUT method to compute the polynomial function, where the complex signal x is left outside the LUT, since the LUT address is generated from magnitude and with the same magnitude x may have various values. The coefficients in front of each polynomial term are ignored in the figure.
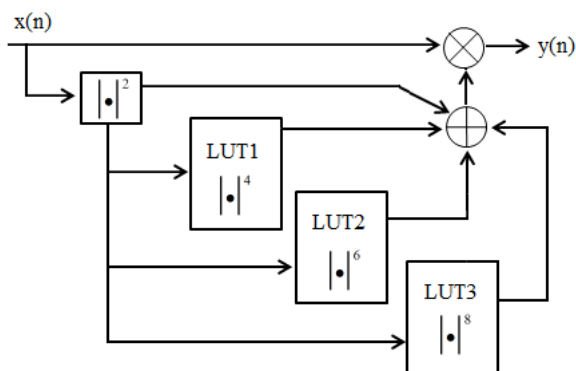
Figure 6.6.   Architecture of LUT method to calculate polynomial

In the other hand, we can store even more terms in one LUT [84, 85]. In (6.1), all the terms underline in each memory branch are associated to one input signal with a certain memory depth. Therefore, all terms with the same memory depth can be calculated and saved in one LUT in advance. With this approach, only $M + 1$ LUTs are needed, where M is the memory depth, and the number of multipliers and adders will also be reduced. Consequently, the memory size required is less and the complexity of the circuit is reduced. It is suitable for models with deep memory. However, it will be specific for only one system and any change happening to the system will cause the requirement of updating all the LUTs with new contents. Figure 6.7 is an example architecture of this LUT method to compute the polynomial function.

As mentioned above, the block RAMs are employed to implement the LUTs. LUT entries spacing has to be taken into consideration which can affect the performance of the digital predistortion. Both uniform and non-uniform spacing have been discussed in the literature [88–90]. Non-uniform spacing with a so-called companding
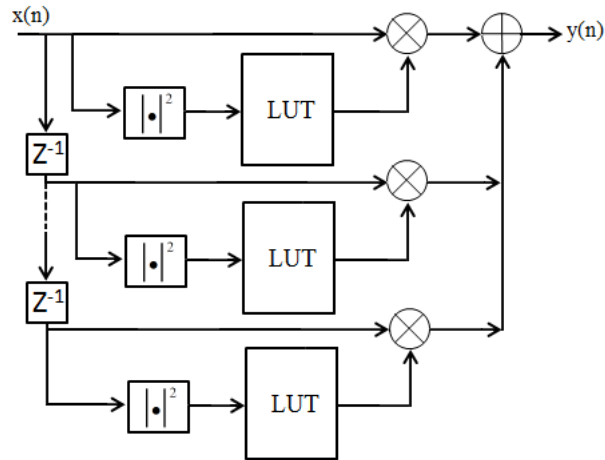
Figure 6.7.   Architecture of LUT method to calculate polynomial

function has been recognized to deliver the best linearization performance. The most common companding functions are Cavers companding function [89], and a more simplified sub-optimum companding function presented in [90].

However, the non-uniform spacing with companding function needs the information of the probability density function of the signal and is usually with high computational complexity. Therefore, we choose the uniform spacing, due to its relatively low complexity and good enough results in comparison to the non-uniform spacing. The range of the test signal in our experiments is between -1 and 1 and we divide it into 1024 intervals. The resolution of the Virtex-4 FPGA board is 14 bits and therefore the total size of one LUT is 14 Kbits. Fortunately, the size of one block RAM is 18 Kbits, enough for employing one LUT.

The CORE Generator software from Xilinx helps us to build the block RAM easily. Figure 6.8 is the window of the CORE Generator for building single port block RAM in Xilinx ISE. We should select 'Read Only' and fill the Width blank with 14 and Depth with 1024. The contents for each LUT is initialized with a *coe* file. We can generate the corresponding LUT contents in Matlab and save it to a *txt* file. Then we put the following two sentences at the beginning of the *txt* file and change the extension name from *txt* to *coe*.

memory_initialization_radix = 10;
memory_initialization_vector =

87
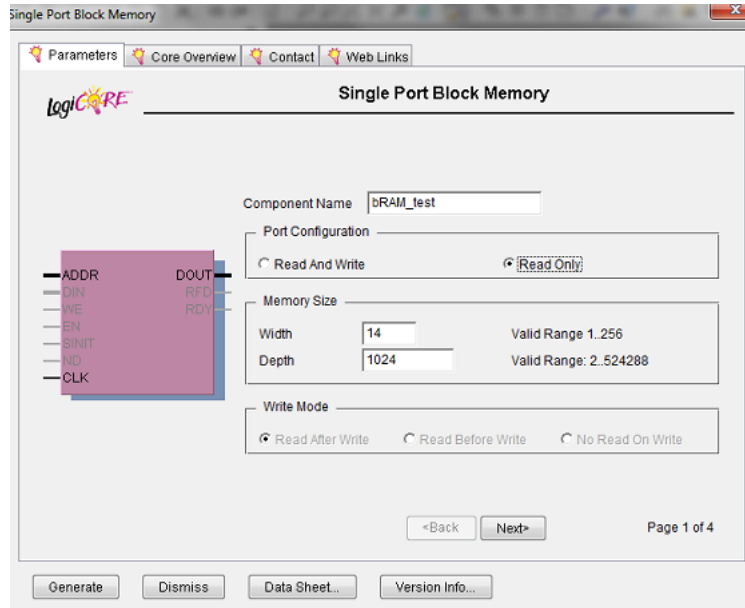
Figure 6.8.   Window of CORE Generator for building single port block RAM

Finally, in CORE Generator, we should select the generated *coe* file and load it. The LUT is successfully built with the block RAM. The VHDL code describing the block RAM will be automatically generated in Xilin ISE software.

6.3.2   Direct Multiple and Add Method

Apart from LUT implementation, another method [86] to implement the memory polynomial model is a direct structure with multipliers and adders based on Horner's method [87]. This method maintains the high processing speed thanks to the pipelined architecture of the FPGA. The Horner's method is an algorithm to calculate the polynomial in an efficient way. Given an polynomial

$$y(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots a_n x^n . \tag{6.2}$$

It can be re-ranged as

$$y(x) = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + a_n x))). \tag{6.3}$$

The new form of the function divides the polynomial into n stages, each of which has the same functionality $a + bx$. Pipeline structure is best suitable for the calculation

of the multi-stage polynomial.

If we extend to the memory polynomial model with odd-order terms, the Horner's method works the same. For instance, a memory polynomial model with polynomial order of 5 and memory depth of 0 is written as

$$y = a_1 x + a_2 x |x|^2 + a_3 x |x|^4 + a_4 x |x|^6 + a_5 x |x|^8, \tag{6.4}$$

which can be transformed as

$$y = x(a_1 + |x|^2(a_2 + |x|^2(a_3 + |x|^2(a_4 + a_5|x|^2)))). \tag{6.5}$$

It can be seen that the polynomial in (6.5) consists of several stages of the same block function $a + b|x|^2$. This block just deals with multiplication and summation algorithms and it is simple to implement in VHDL code. Therefore, we can evaluate a polynomial equation in the order implied by (6.5) using n pipelined stages. The first stage computes $a_4 + a_5|x|^2$ and passes the result and the value of x to the next stage. The second stage uses the previous result as the b coefficient, adding a_3 and passing the result and x value to the third stage. The remaining stages continue in the same manner. The data flow of the polynomial applying Horner's method is shown in Figure 6.9.
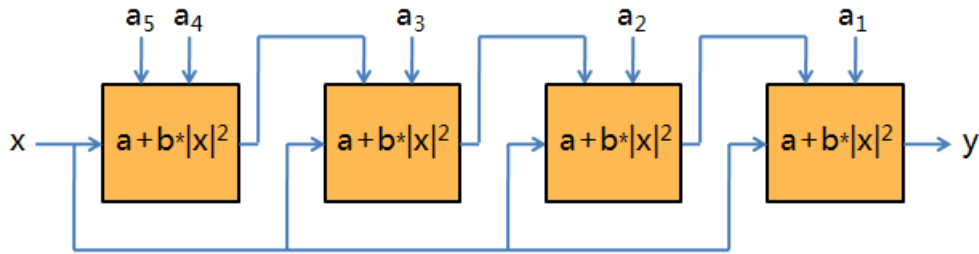
Figure 6.9.   Basic idea of the direct multiple and add method

The proposed Horner's method to calculate the polynomial can save the number of multipliers which are one of the most complex and expensive components in FPGA. Moreover, the throughout is the same with the data input, that is one input data sampling with one calculated output data in one clock with some latency. In conclusion, with Horner's method and the pipelined architecture, the memory polynomial model can be calculated with faster speed and less logic resources of the FPGA.

6.4 Testbench Description and Experimental Procedures

The test bench for FPGA based digital predistortion, as shown in Figure 6.10, is slightly different from that of software environment. In the software based simulation, the predistorted signal is directly generated in Matlab and then downloaded to the ESG. However, for FPGA implementation, the predistorted signal will be generated by FPGA. Although it is still possible to produce the predistorted signal in Matlab and directly store it in the memory of FPGA by bitstream file, to make it more general, we follow the procedures of ADC and DAC. The ESG outputs the original baseband signal to the FPGA via ADC channels and the digitized signal enters the main part of the FPGA where it will be predistorted. The predistorted signal coming out from FPGA via DAC will be converted back to be analog and is fed back to the ESG where it will be modulated. Finally, the modulated predistorted signal is sent to the PA as the excitation.
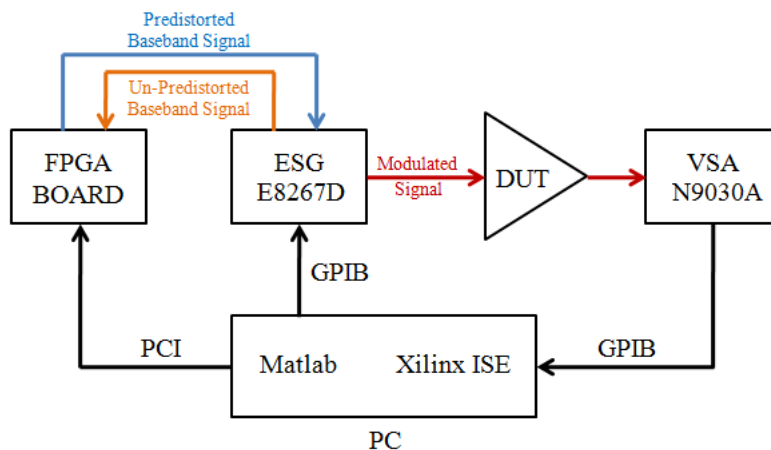
Figure 6.10.   Real-time FPGA based digital predistortion setup

The preparatory procedures are the same with the software simulation. The coefficients of the chosen memory polynomial model and the contents of the LUTs are calculated offline in Matlab. Executable bitfile was generated with the Xilinx ISE. FUSE System Software then downloaded the generated bitfile into FPGA via the PCI interface to control the functionality of the FPGA.

The performance of the applied DPD algorithm will be evaluated with the same criteria used in software simulation. Both time domain AM/AM, AM/PM characteristics and frequency domain power spectrum density have been involved to check

the linearization performance.

6.5 Experimental Results

The experimental setup for testing the FPGA based digital predistortion is shown in Figure 6.11. As seen in Figure 6.11(b), the FPGA board is connected inside the PC with PCI connector. There are two ports, *I output* and *Q output*, on the rear panel of the ESG [91] from where comes out the original baseband signal. The signal enters FPGA through the ADC channels and comes out from the DAC channels. The predistorted signal returns back to the ESG via the two ports at the front panel as shown in Figure 6.11(a).

The hybrid Doherty power amplifier [72], mentioned in the software simulation, was chosen to test the FPGA implementation of the digital predistortion. The PA is biased at $V_{DS} = 28$ V and $V_{GS} = -2.7$ V for the main, and $V_{DS} = 28$ V and $V_{GS} = -7$ V for the peak at 3.4 GHz operation frequency. The test signal is a 7 MHz channel I/Q signal with raised-cosine filter, roll-off of 0.2, 256-QAM modulation and PAPR of 7.4 dB.

The polynomial order and memory depth of the adopted memory polynomial model were chosen as 6 and 1, respectively, for both the LUT methods and the direct multiple and add method. The standard emission mask for power spectrum is still the class 6LA from ETSI.

6.5.1 Direct Multiple and Add Method

The RTL schematic, which is the register transfer level representation of the design of the DPD system, can be viewed in Xilinx ISE software, as shown in Figure 6.12. In the figure, A1 area is the input correction area since there is some DC offset at the input of the ADC channels. A2 area consists of 4 D flip-flops to realize the FIR filter with 1 delay for 2 channels. Furthermore, the A3 area is the main computational part consists of 5 blocks with the same functionality to calculate $a+b|x|^2$ as outlined in Horner's method. Therefore, 5 blocks in this design means polynomial order of 6 of the memory polynomial model. In addition, every block computes all the signals with different memory delay. The following A4 and A5 areas combine all the values to be the final I and Q signals. The last A6 converts the 2's complement format
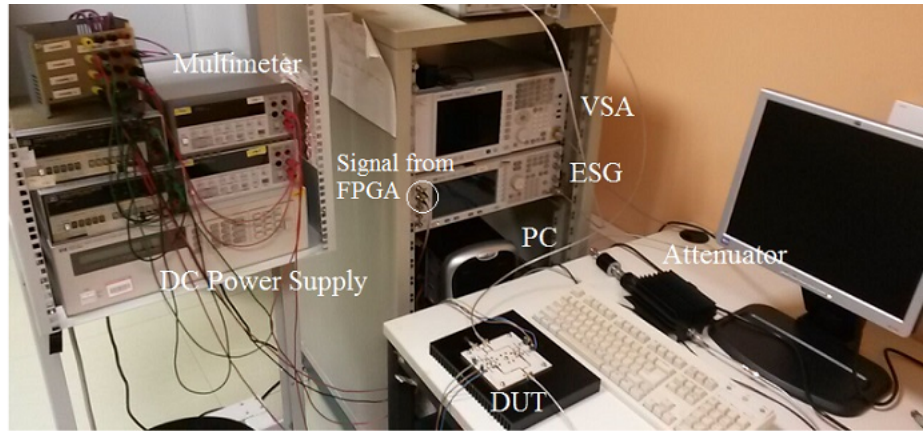
(a)
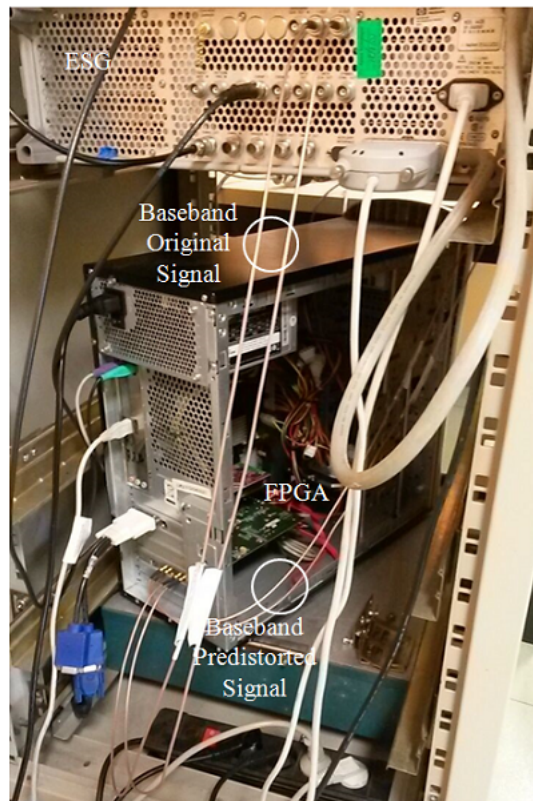
(b)

Figure 6.11.   Experimental setup of FPGA based DPD: (a) front, (b) back.

binary to offset binary with the reason mentioned above. The FPGA resources used
for this method are listed in Table 6.1.
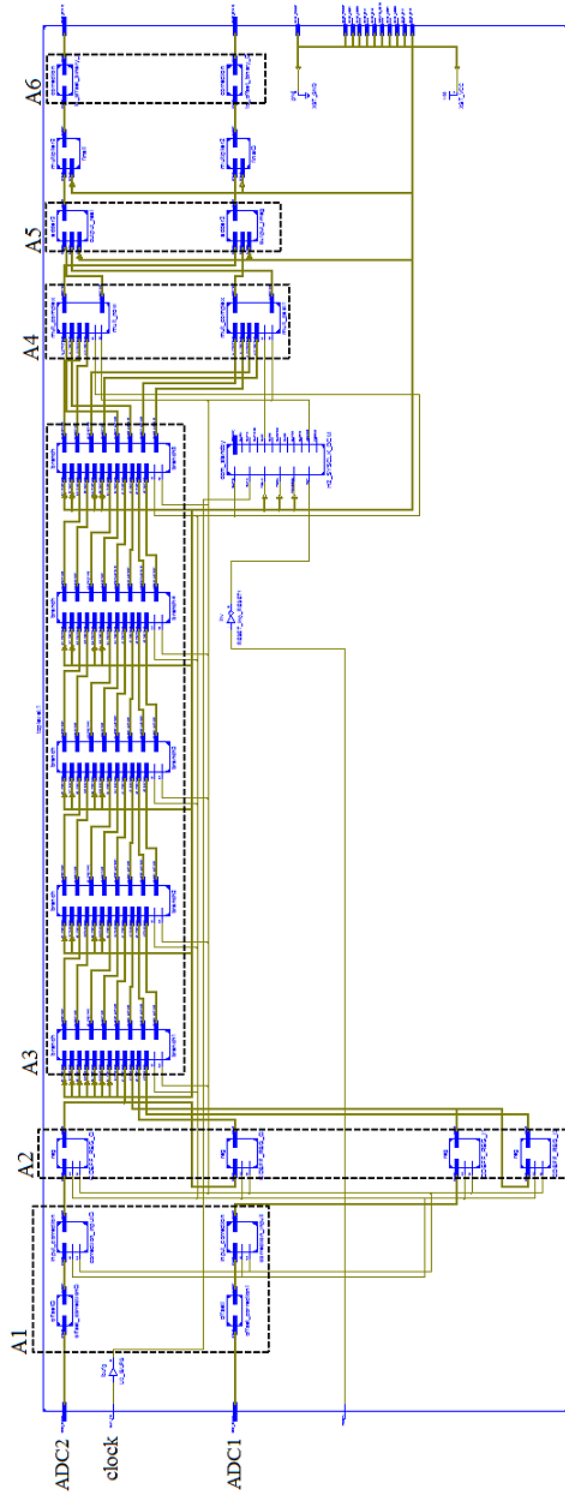
Figure 6.12: RTL schematic of the designed DPD system with direct multiple and add method.

	P = 6, M = 1	P = 6, M = 2
Flip Flops	1172	1680
Slices	753	1081
4 Input LUTs	752	998
DSP48	50	72

Table 6.1. FPGA resources utilization summary for direct multiple and add method

The extra power consumption of DPD system is one of the biggest concerns of applying DPD algorithm. The power in watts can be estimated by the Xilinx Power Analyzer, which is showed in detail in Table 6.2. It is clear in the table, the dominant power consumption is the leakage power, in other words the static power, with 76% of the total power. Moreover, the static power remains almost the same for the two memory polynomial models with different complexity. As a result, the power consumption increases slightly with the increase of the model complexity.

	P = 6, M = 1	P = 6, M = 2
Clock	0.05412	0.06571
Logic	0.00019	0.00019
Signals	0.00276	0.00367
DSPs	0.00000	0.00000
DCMs	0.08176	0.08176
IOs	0.00050	0.00050
Leakage	0.44300	0.44400
Total	0.582	0.595

Table 6.2. Power summary of the direct multiple and add method. All the power values are in watts.

The hybrid Doherty PA has been linearized with the DPD system implemented in FPGA with direct multiple and adder method. Figure 6.13 illustrates the AM/AM and AM/PM characteristics of both the original PA and linearized PA responses, where the magenta constellation represents the response of the PA solo and blue constellation is for the PA with DPD system. In Figure 6.13(a), the blue points form a more straight line and in the meanwhile, in Figure 6.13(b) the blue points are more flat. Both of two blue lines indicate a more linearized performance of PA with DPD algorithm in comparison with only PA. Quantitatively, the NMSE value between the normalized output and input of the PA has improved from -20.5 dB

without DPD to -31.3 dB with DPD.
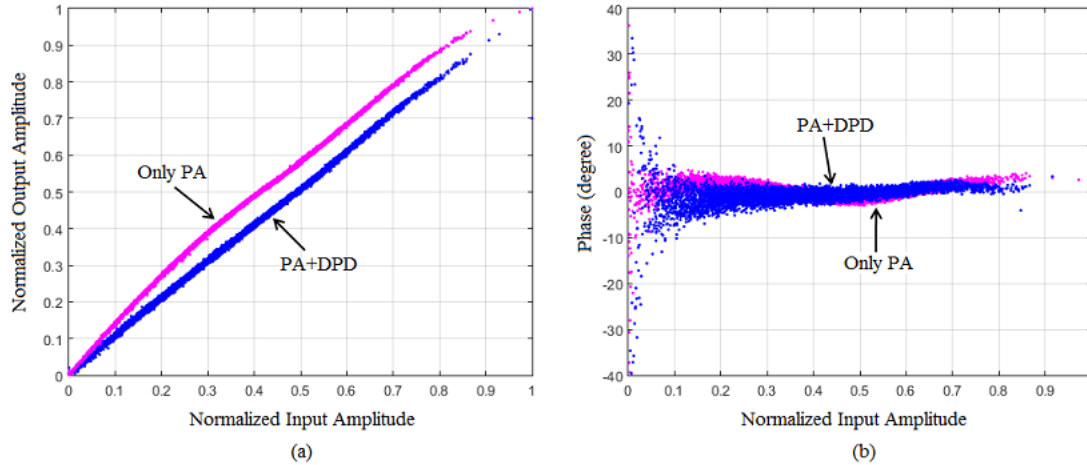
Figure 6.13. Instantaneous magnitude and phase response for the 7 MHz channel I/Q signal: (a) AM/AM characteristic, (b) AM/PM characteristic.

Figure 6.14. Normalized power spectrum density of the designed Doherty PA with a 7 MHz I/Q signal. The black line is the ETSI spectrum emission mask.

In the frequency domain, the linearization performance is evaluated with the ACPR, as shown in Figure 6.14. The out-band frequency power has been reduced and the

power spectrum has achieved the compliance with the ETSI emission mask. The ACPR improvements for upper and lower side band are from -33.7 dBc to -44.1 dBc and from -33.2 dBc to -44.2 dBc, respectively.

Consequently, the DPD algorithm implemented in FPGA with direct multiple and add method has successfully linearized the hybrid Doherty PA. 11 dB improvement of NMSE and 11 dB of ACPR have been obtained. The average output power of the linearized PA is 35.3 dBm with efficiency of 36.6%.

6.5.2   LUT Method

The results of the two methods with LUT are very close, therefore we introduce only the results for one method, the one with fewer LUTs. The RTL schematic is shown in Figure 6.15. In the figure, A1 area is the input correction area, functioning in the same way with the previous implementation. A2 and A3 are the main LUTs, where A2 is for the current signal and A3 is for the signal with one delay. Both A2 and A3 have two independent LUTs in each area for representing separately the I and Q signals. The blocks located just before A2 and A3 are the components to compute the magnitude of the signal which will be the address to the block RAMs for data retrieval. The following A4 adds all the values to be the final I and Q signals. The last A5 converts the 2's complement format binary to offset binary with the reason mentioned above. The remaining small blocks are D flip-flops for storing data or providing delay.

	P = 6, M = 1		P = 6, M = 2	
	M1	M2	M1	M2
Flip Flops	263	583	311	737
Slices	265	531	333	666
4 Input LUTs	360	663	425	792
DSP48	14	32	20	44
16RAMs	4	8	6	12

Table 6.3.   FPGA resources utilization summary for the two LUT methods

In spite of the similar linearization performance obtained, the FPGA resources used by the two LUT methods are different, which are listed in Table 6.3, where M1 stands for the method with one LUT representing all the polynomial terms with the same time state while M2 is the method with one LUT storing one polynomial term.
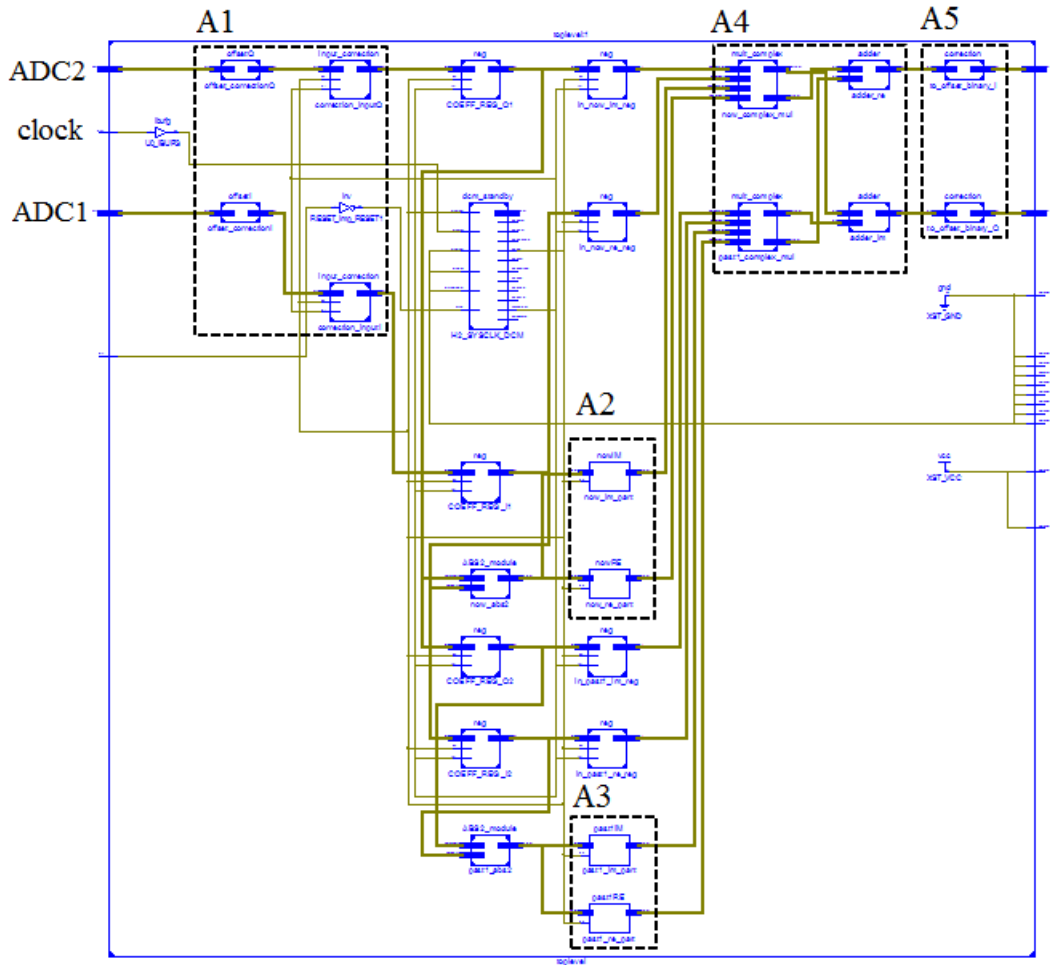
Figure 6.15.   RTL schematic of the designed DPD system with LUT method.

Two memory polynomial models with different memory depth and same polynomial order have been compared. Obviously, M1 demands both less logic resources and block RAMs, comparing to the M2 method. Moreover, the increment of FPGA resources requirement is with a very small amount when the memory depth of the model increases.

Table 6.4 records the power consumption in watts of each parts in FPGA for the two LUT methods with two different model complexity. We can see that the leakage power is still the largest power consumption, consuming nearly 80% of the total power.

	P = 6, M = 1		P = 6, M = 2	
	M1	M2	M1	M2
Clock	0.03544	0.04646	0.03876	0.04978
Logic	0.00019	0.00020	0.00019	0.00020
Signal	0.00105	0.00160	0.00103	0.00179
BRAM	0.00000	0.00000	0.00000	0.00000
DSP	0.00000	0.00000	0.00000	0.00000
DCM	0.08176	0.08176	0.08176	0.08176
IO	0.00050	0.00050	0.00050	0.00050
leakage	0.44300	0.44300	0.44300	0.44300
Total	0.562	0.573	0.565	0.577

Table 6.4. Power summary of the two LUT methods. All the power values are in watts.

The hybrid Doherty PA has been linearized with the DPD system implemented in FPGA with LUT methods. Figure 6.16 illustrates the AM/AM and AM/PM characteristics of both the original PA and linearized PA responses, where the magenta constellation represents the response of PA solo and blue constellation is for PA with DPD system. In Figure 6.16(a), the blue points form a more straight line and in the meanwhile, in Figure 6.16(b) the blue points are more flat near zero degree. Both of two blue lines indicate a more linearized performance of PA with DPD algorithm in comparison with only PA. Quantitatively, the NMSE value between the normalized output and input of the PA has improved from -20.5 dB without DPD to -31.0 dB with DPD.

In the frequency domain, the linearization performance is evaluated with the ACPR, as shown in Figure 6.17. The out-band frequency power has been reduced and the power spectrum has achieved the compliance with the ETSI standards. The ACPR improvements for upper and lower side band are from -33.7 dBc to -43.8 dBc and from -33.2 dBc to -43.6 dBc, respectively.

In conclusion, the results with the LUT method are very similar to the direct multiple and add method with almost the same improvements in NMSE and ACPR values. The average output power of the linearized PA with LUT method is 35.3 dBm with efficiency of 36.1%.
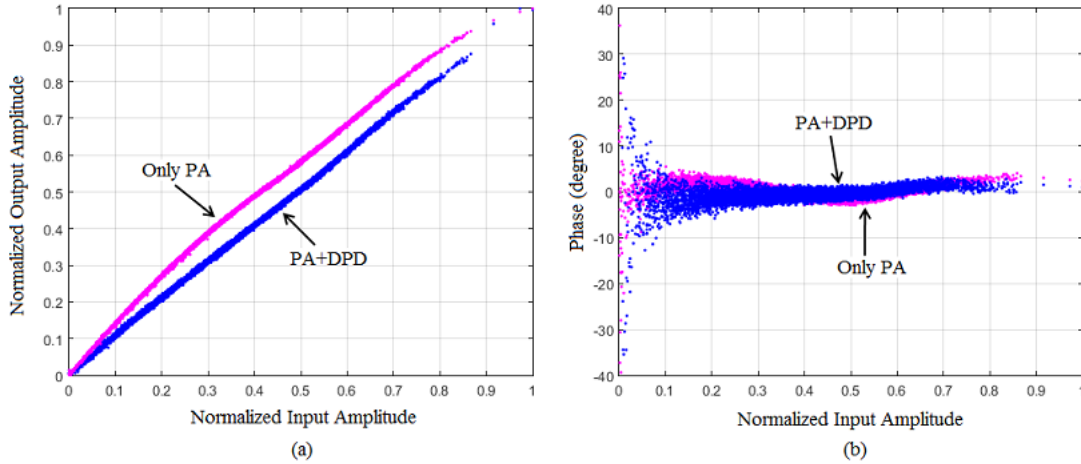
Figure 6.16.   Instantaneous magnitude and phase response for the 7 MHz channel
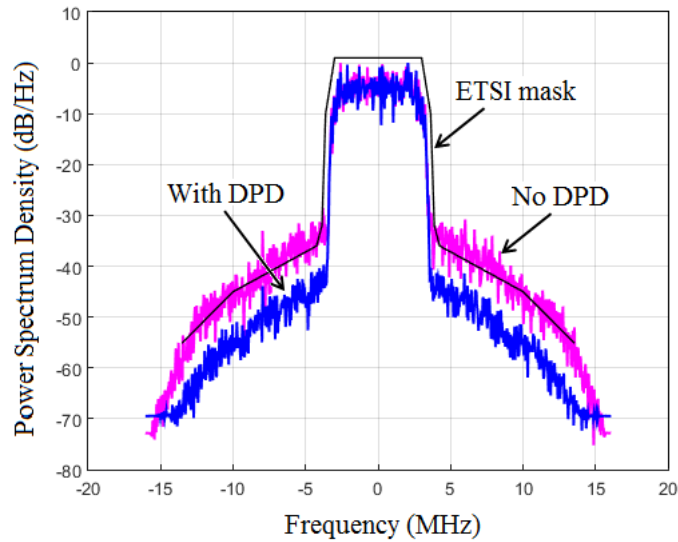I/Q signal: (a) AM/AM characteristic, (b) AM/PM characteristic.
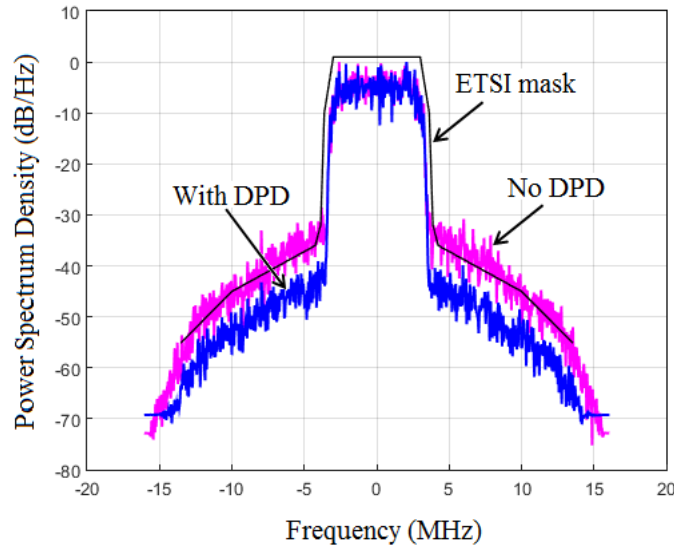
Figure 6.17.   Normalized power spectrum density of the designed Doherty PA with
a 7 MHz I/Q signal. The black line is the ETSI spectrum emission mask.

6.6   Comparison and Conclusion

The FPGA implementation of the digital predistortion has been tested with both
the LUT methods and the direct multiple and add method. The linearization of
a hybrid Doherty power amplifier has been successfully realized for both the two
methods with good results. The applied methods delivered similar linearization

performance with utilization of different logic resources and block RAMs on the FPGA. The resources used for different implementation methods are summarized in Table 6.5, where M1 is the direct multiple and add method, M2 is the LUT method with fewer LUTs and M3 is the LUT method with each LUT representing one polynomial term.

	P = 6, M = 1			P = 6, M = 2		
	M1	M2	M3	M1	M2	M3
Flip Flops	1172	263	583	1680	311	737
Slices	753	265	531	1081	333	666
4 Input LUTs	752	360	663	998	425	792
DSP48	50	14	32	72	20	44
Block RAMs	0	4	8	0	6	12
Power Consumption	0.582	0.562	0.573	0.595	0.565	0.577

Table 6.5.   Summary of the FPGA resources utilization.

From the table , we can conclude that the direct multiple and add method uses only the logic circuits resources of the FPGA and no memory resource. The LUT methods save a large amount of logic resources at the expense of block RAMs as look-up-tables. Moreover, M2 employs even less elements of the FPGA than M3 and the resource usage increases less when the memory depth of the model increases. In other hand, the circuit of M3 is more complex than that of M2, resulting more logic resources and block RAMs requirements. However, the logic components used in M3 is still much less than M1, more than half less. In other aspects, M2 has a drawback of flexibility, it is very specific and all the LUT contents should be updated with new data with any change of the model. The power consumption is not a critical factor since the value is almost the same for all three methods.

Furthermore, the linearization performance should also be compared with the Matlab based results, as shown in Table 6.6. The average power and the efficiency are nearly the same for all the methods. However, the linearization performance degrades slightly for the FPGA implementation methods, especially for LUT methods. The main reason may be the DC offset at the input of the ADC channels of the FPGA board in our laboratory which can not be compensated ideally. Another possible reason may be the resolution of the 14 bits representation of the signal which presents less accuracy compared to Matlab. Fortunately, in the frequency domain, the power spectrum for all the DPD systems with different methods can fulfill the requirement of the ETSI standard emission mask, as shown in Figure 6.18. In the figure, magenta stands the PA without DPD, red is with Matlab DPD, green

is with direct structure DPD, the blue is with LUT DPD and the ETSI standard is the black line.

	No DPD	Matlab	Direct Method	LUT method
Average power (dBm)	35.1	35.0	35.2	35.2
Efficiency	36.8%	36.8%	36.6%	36.5%
NMSE (dB)	-20.5	-33.5	-31.3	-31.0
Upper ACPR (dBc)	-33.7	-45.2	-44.1	-43.8
Lower ACPR (dBc)	-33.2	-47.6	-44.2	-43.6

Table 6.6. Comparisons of linearization performance.
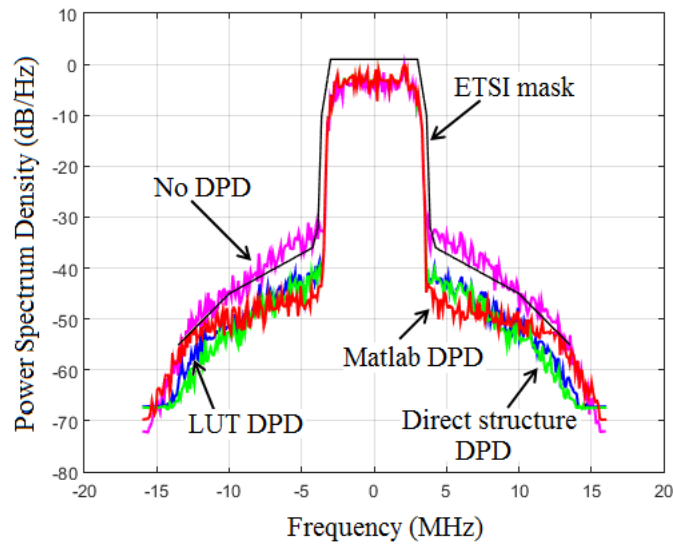
Figure 6.18. Normalized power spectrum density comparison: magenta stands the PA without DPD, red is with Matlab DPD, green is with direct structure DPD, the blue is with LUT DPD and the ETSI emission mask is the black line.

Chapter 7

Adaptive Predistorter Design

Digital predistortion has been verified, both in literature and in industrial applications, to be a sufficient technique for linearizing the power amplifier. It guarantees a good linear behavioral of a PA operating at a relatively high power region, maintaining high efficiency. In most papers and in the previous chapters on simulation of the digital predistortion of this thesis, the PA under test is assumed to be a time-invariant system, that is the PA characteristics do not change with the same operation conditions. Therefore, the model extracted for the PA keeps the same for the entire simulation procedures. However, in the real world, the PA behavioral may change over time due to some reasons, for instance, the change of ambient temperature, bias drifting or aging. Generally, the change of all these factors is irregular and cannot be predicted in advance. We should adjust the model during the entire operation time of the PA. In other words, an adaptive system is needed where the parameters of the model should be extracted and updated continually in the system.

Due to the fact that the change of the PA behavior is very slow compared to the model extraction time, the least square method can still be used for the extraction of the model parameters in the adaptive system. A group of data should be collected repeatedly for extracting the model with least square method. An extra memory is required for storing the collected data. In other hand, adaptive algorithm, like Least Mean Square (LMS) or Recursive Least Square (RLS), is preferred in the adaptive digital predistortion system. The LMS and RLS techniques can be seen as the adaptive filter to minimize the cost function with iteration. The RLS method presents better performance, in terms of converge speed and the final accuracy, at a cost of higher computational complexity than the LMS method. These two methods can continually update the model parameters sample by sample with the input and output data. The accuracy of the model with RLS is comparable with the least square method, making it a proper choice for adaptive digital predistortion.

7.1 Adaptive Algorithm

7.1.1 LMS

The LMS algorithm is one of the most widely used adaptive algorithms, mainly due to its low computational complexity. It can be derived as an application of steepest descent method [4]. Figure 7.1 is a general adaptive filter system, where $x(n)$, $y(n)$ and $d(n)$ are the input signal, the adaptive output signal and the desired signal, respectively. Therefore, $e(n)$ is the error between the desired signal and the obtained adaptive signal. The adaptive algorithm makes use of this error to adjust the coefficients of the adaptive filter to minimize the error. The adaptive filter is assumed to consist of a linear combination of basis functions constructed with the input signal, that is
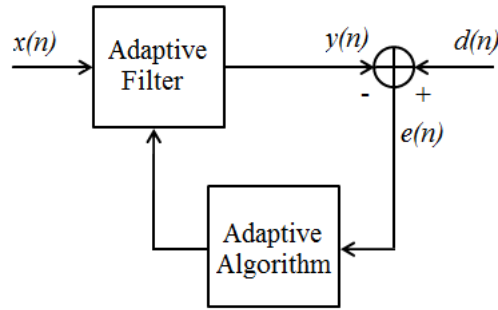
Figure 7.1. General adaptive technique scheme.

$$y(n) = \sum_{i=1}^{N-1} \omega_i x_i(n) = \boldsymbol{\omega}^T \boldsymbol{x} \ , \tag{7.1}$$

where $\boldsymbol{x} = [x_0(n)\ x_1(n)\ \cdots\ x_{N-1}(n)]^T$ is the vector of input signal basis with total number of N and $\boldsymbol{\omega} = [\omega_0\ \omega_1\ \cdots\ \omega_{N-1}]^T$ is the adaptive filter coefficients. Generally, $\boldsymbol{x}$ can be chosen as $x_0(n) = x(n)$, $x_1(n) = x(n-1)$, $\cdots$, $x_{N-1}(n) = x(n-N+1)$. The error signal will be

$$e(n) = d(n) - y(n) = d(n) - \sum_{i=0}^{N-1} \omega_i * x(n-i) \ . \tag{7.2}$$

The best filter is that the resulted error signal is with the minimum value. The mean square value of $e(n)$ is chosen as the cost function to be minimized:

$$J = E(|e(n)|^2) = E(e(n)e^*(n)) \ , \tag{7.3}$$

103

where E is the statistical expectation operator and the asterisk denotes *complex conjugation*. In our case, the input data are complex numbers, thus the coefficient $\boldsymbol{\omega}$ will be also complex which, with the ith coefficient as an example, can be expressed with real and imaginary parts:

$$\omega_i = a_i + jb_i \qquad i = 0,1,\cdots \tag{7.4}$$

Typically, the first order derivative is used to search for the optimal value which will result the minimum error. Therefore, by applying the first order derivative to the cost function J with respect to the coefficients vector, $\boldsymbol{\omega}$, of the adaptive filter, a multi-dimension complex gradient vector ∇J is obtained, in which the ith element is calculated as

$$
\begin{aligned}
\nabla_i J &= \frac{\partial J}{\partial a_i} + j\frac{\partial J}{\partial b_i} \\
&= E\left[\frac{\partial e(n)}{\partial a_i}e^*(n) + \frac{\partial e^*(n)}{\partial a_i}e(n) + \frac{\partial e(n)}{\partial b_i}je^*(n) + \frac{\partial e^*(n)}{\partial b_i}je(n)\right] \\
&= -2E[x(n-i)e^*(n)] \qquad i = 0,1,\cdots
\end{aligned}
\tag{7.5}
$$

where i is the input signal sample instance. The optimal value located at the moment when $\nabla J = 0$, that is for each sample instance during the iteration, the following equation should be satisfied:

$$E[x(n-i)e^*(n)] = 0 \qquad i = 0,1,\cdots \tag{7.6}$$

Substitute (7.2) into (7.6),

$$E\left[x(n-i)\left(d^*(n) - \sum_{k=0}^{N-1}\omega_{0k}x^*(n-k)\right)\right] = 0 \qquad i = 0,1,\cdots \tag{7.7}$$

where ω_{0k} is the kth coefficient of the optimal coefficient vector of the adaptive filter. Since, in general, the coefficients of the adaptive filter are determined numbers, ω_{0k} can be carried out of the expectation operator. (7.7) can be expanded and rearranged as

$$\sum_{k=0}^{N-1}\omega_{0k}E[x(n-i)x^*(n-k)] = E[x(n-i)d^*(n)] \qquad i = 0,1,\cdots \tag{7.8}$$

(7.8) should be valid for each $x(n-i)$ with different i and we can rewrite (7.8) with the matrix form as

$$\boldsymbol{R}\boldsymbol{\omega} = \boldsymbol{p} \tag{7.9}$$

where $\boldsymbol{R} = E[\boldsymbol{x}(n)\boldsymbol{x}^H(n)]$ and $\boldsymbol{p} = E[\boldsymbol{x}(n)d^*(n)]$ with $\boldsymbol{x}(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^T$ and the superscript H denotes *Hermitian transposition*. Therefore, $\boldsymbol{R}$ is a matrix given by

$$\boldsymbol{R} = \begin{bmatrix} E[x(n)x^*(n)] & E[x(n)x^*(n-1)] & \cdots & E[x(n)x^*(n-N+1)] \\ E[x(n-1)x^*(n)] & E[x(n-1)x^*(n-1)] & \cdots & E[x(n-1)x^*(n-N+1)] \\ \vdots & \vdots & \ddots & \vdots \\ E[x(n-N+1)x^*(n)] & E[x(n-N+1)x^*(n-1)] & \cdots & E[x(n-N+1)x^*(n-N+1)] \end{bmatrix}$$

and $\boldsymbol{p}$ is

$$\boldsymbol{p} = [E[x(n)d^*(n)],\ E[x(n-1)d^*(n)],\ \cdots,\ E[x(n-N+1)d^*(n)]]^T$$

The equation (7.9) can be solved for $\boldsymbol{\omega}$ by assuming that $\boldsymbol{R}$ is nonsingular, with the result as

$$\boldsymbol{\omega} = \boldsymbol{R}^{-1}\boldsymbol{p} \ , \tag{7.10}$$

where $\boldsymbol{R}^{-1}$ is the inverse matrix of $\boldsymbol{R}$. If matrix $\boldsymbol{R}$ and $\boldsymbol{p}$ are captured well, we can replace them with their determined samples that means $\boldsymbol{R} = \boldsymbol{x}(n)\boldsymbol{x}^H(n)$ and $\boldsymbol{p} = \boldsymbol{x}(n)d^*(n)$. Hence, a steepest descent based algorithm can be used to extract the solution of (7.10) as

$$\begin{aligned} \boldsymbol{\omega}(k+1) &= \boldsymbol{\omega}(k) - \mu \frac{\partial J}{\partial \boldsymbol{\omega}} \\ &= \boldsymbol{\omega}(k) + 2\mu(\boldsymbol{p} - \boldsymbol{R}\boldsymbol{\omega}) \\ &= \boldsymbol{\omega}(k) + 2\mu(d(k)\boldsymbol{x}(k) - \boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{\omega}(k)) \\ &= \boldsymbol{\omega}(k) + 2\mu\boldsymbol{x}(k)(d(k) - \boldsymbol{x}^T(k)\boldsymbol{\omega}(k)) \ , \end{aligned} \tag{7.11}$$

where μ is the step-size parameter. Therefore, the final result known as the least mean square algorithm is

$$\boldsymbol{\omega}(k+1) = \boldsymbol{\omega}(k) + 2\mu e(k)\boldsymbol{x}(k) \tag{7.12}$$

where μ is usually chosen with small real constant and k is the iteration number. The error signal $e(k)$ is the difference between the current desired value and the value from the adaptive filter with the current coefficient $\omega(k)$, that is $e(k) = d(k) - \boldsymbol{x}^T(k)\boldsymbol{\omega}(k)$. Therefore, at each iteration, $x(k), d(k), \omega(k)$ are required to obtain the new $\omega(k+1)$. In general, when we iterate the LMS algorithm, the initialization of

adaptive filter coefficient vector $\boldsymbol{\omega}$ is chosen with a all-zero vector, that is $\boldsymbol{\omega} = \boldsymbol{0}$. The LMS algorithm is summarized in Table 7.1.

Initialization
$\boldsymbol{\omega} = \boldsymbol{0}$
For the iteration, compute
$e(k) = d(k) - \boldsymbol{x}^T(k)\boldsymbol{\omega}(k)$
$\boldsymbol{\omega}(k+1) = \boldsymbol{\omega}(k) + 2\mu e(k)\boldsymbol{x}(k)$

Table 7.1.   Summary of the conventional LMS algorithm.

7.1.2   RLS

The RLS algorithm is an extension of least square method where the least square problem is computed in recursive form with new data samples. The architecture of RLS algorithm can also be represented by Figure 7.1, in which the adaptive algorithm is RLS method. Comparing with the LMS algorithm, RLS can achieve higher accuracy with faster converge speed. However, the computational complexity of RLS is also much higher than the LMS algorithm.

Similar to the LMS algorithm, the objective of RLS is also to adjust the adaptive filter coefficients such that the output signal from the filter will coincide as much as possible with the desired signal. The objective function that we choose to minimize is given by

$$
\begin{aligned}
J(n) &= \sum_{i=0}^{n} \lambda^{n-i} e^2(i) \\
&= \sum_{i=0}^{n} \lambda^{n-i} \left[d(i) - \boldsymbol{x}^T(i)\boldsymbol{\omega}(n) \right]^2,
\end{aligned}
\tag{7.13}
$$

where $\boldsymbol{x}$ is still chosen as the basis functions of a generic FIR adaptive filter with $\boldsymbol{x}(n) = [x(n), x(n-1), \cdots, x(n-N)]^T$ and $e(i)$ is the error between the desired signal and adaptive filter output signal at instant i. λ is called forgetting factor which is chosen in the range between 0 and 1. Therefore, the error signals with the instants far from the current input sample have reduced effect on the objective function.

To obtain the optimal value of the adaptive filter coefficients $\boldsymbol{\omega}(n)$ which gives the minimum objective function, we could differentiate the objective function $J(n)$ with respect to $\boldsymbol{\omega}(n)$:

$$\frac{\partial J(n)}{\partial \boldsymbol{\omega}(n)} = -2 \sum_{i=0}^{n} \lambda^{n-i} \boldsymbol{x}(i)[d(i) - \boldsymbol{x}^T(i)\boldsymbol{\omega}(n)] . \tag{7.14}$$

The optimal vector $\boldsymbol{\omega}(n)$ is found by forcing the first order derivative to be 0 in (7.14). With some algebra computation, the final expression for the optimal $\boldsymbol{\omega}(n)$ is given by

$$\begin{aligned}
\boldsymbol{\omega}(n) &= \left[\sum_{i=0}^{n} \lambda^{n-i} \boldsymbol{x}(i)\boldsymbol{x}^T(i) \right]^{-1} \sum_{i=0}^{n} \lambda^{n-i} \boldsymbol{x}(i)d(i) \\
&= \boldsymbol{R}^{-1}(n)\boldsymbol{p}(n) ,
\end{aligned} \tag{7.15}$$

where

$$\begin{aligned}
\boldsymbol{R}(n) &= \sum_{i=0}^{n} \lambda^{n-i} \boldsymbol{x}(i)\boldsymbol{x}^T(i) \\
&= \lambda \left(\sum_{i=0}^{n-1} \lambda^{n-1-i} \boldsymbol{x}(i)\boldsymbol{x}^T(i) + \lambda^{-1} \boldsymbol{x}(n)\boldsymbol{x}^T(n) \right) \\
&= \lambda \boldsymbol{R}(n-1) + \boldsymbol{x}(n)\boldsymbol{x}^T(n) .
\end{aligned} \tag{7.16}$$

To avoid the matrix inversion computation of $\boldsymbol{R}^{-1}(n)$ in (7.15), the matrix inversion lemma [92], which is described as follow, has been used. Suppose that $\mathbf{A}$ and $\mathbf{B}$ are two positive definite N-by-N matrices and have a relationship:

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{\mathbf{H}} , \tag{7.17}$$

where $\mathbf{C}$ is N-by-M matrix and $\mathbf{D}$ is M-by-N matrix. Then the inversion matrix of $\mathbf{A}$ can be expressed as

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^{\mathbf{H}}\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^{\mathbf{H}}\mathbf{B} . \tag{7.18}$$

(7.16) and (7.17) have the same relation if we make the following map between the elements in the two equations:

$$\mathbf{A} = \boldsymbol{R}(n)$$
$$\mathbf{B}^{-1} = \lambda \boldsymbol{R}(n-1)$$
$$\mathbf{C} = \boldsymbol{x}(n)$$
$$\mathbf{D} = 1$$

Then, according the matrix inversion lemma in (7.18), we will get

$$\boldsymbol{R}^{-1}(n) = \lambda^{-1}\boldsymbol{R}^{-1}(n-1) - \frac{\lambda^{-2}\boldsymbol{R}^{-1}(n-1)\boldsymbol{x}(n)\boldsymbol{x}^{H}(n)\boldsymbol{R}^{-1}(n-1)}{1 + \lambda^{-1}\boldsymbol{x}^{H}(n)\boldsymbol{R}^{-1}(n-1)\boldsymbol{x}(n)} \tag{7.19}$$

For simplicity, let

$$\boldsymbol{P}(n) = \boldsymbol{R}^{-1}(n) \tag{7.20}$$

and

$$\boldsymbol{k}(n) = \frac{\lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{x}(n)}{1 + \lambda^{-1}\boldsymbol{x}^{H}(n)\boldsymbol{P}(n-1)\boldsymbol{x}(n)} \tag{7.21}$$

Then, (7.19) can be rewritten as

$$\boldsymbol{P}(n) = \lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}\boldsymbol{k}(n)\boldsymbol{x}^{H}(n)\boldsymbol{P}(n-1) \tag{7.22}$$

In the next, we should conclude the expression for $\boldsymbol{\omega}(n)$. From (7.15), we know

$$\left[\sum_{i=0}^{n} \lambda^{n-i}\boldsymbol{x}(i)\boldsymbol{x}^{T}(i) \right] \boldsymbol{\omega}(n) = \lambda \left[\sum_{i=0}^{n-1} \lambda^{n-1-i}\boldsymbol{x}(i)d(i) \right] + \boldsymbol{x}(n)d(n) \tag{7.23}$$

Using the definition of $\boldsymbol{R}(n)$ and $\boldsymbol{p}(n)$ in (7.15), we can get

$$\begin{aligned} \boldsymbol{R}(n)\boldsymbol{\omega}(n) &= \lambda \boldsymbol{p}(n-1) + \boldsymbol{x}(n)d(n) \\ &= \lambda \boldsymbol{R}(n-1)\boldsymbol{\omega}(n-1) + \boldsymbol{x}(n)d(n) \\ &= \left[\sum_{i=0}^{n} \lambda^{n-i}\boldsymbol{x}(i)\boldsymbol{x}^{T}(i) - \boldsymbol{x}(n)\boldsymbol{x}^{T}(n) \right] \boldsymbol{\omega}(n-1) + \boldsymbol{x}(n)d(n) \\ &= \boldsymbol{R}(n)\boldsymbol{\omega}(n-1) + \boldsymbol{x}(n)(d(n) - \boldsymbol{x}^{T}(n)\boldsymbol{\omega}(n-1)) \end{aligned} \tag{7.24}$$

As defined above, $\boldsymbol{P}(n) = \boldsymbol{R}^{-1}(n)$, the final expression for $\boldsymbol{\omega}(n)$ will be

$$\boldsymbol{\omega}(n) = \boldsymbol{\omega}(n-1) + e(n)\boldsymbol{P}(n)\boldsymbol{x}(n) \tag{7.25}$$

where $e(n) = d(n) - \boldsymbol{x}^T(n)\boldsymbol{\omega}(n-1)$. Therefore, the RLS algorithm can be summarized as

---

Initialization
$\boldsymbol{P}(0) = \delta\boldsymbol{I}$
where δ is a constant and $\boldsymbol{I}$ is the identity matrix.
$\boldsymbol{\omega}(0) = \boldsymbol{0}$

For the iteration, compute
$\boldsymbol{k}(n) = \frac{\lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{x}(n)}{1+\lambda^{-1}\boldsymbol{x}^H(n)\boldsymbol{P}(n-1)\boldsymbol{x}(n)}$
$e(n) = d(n) - \boldsymbol{x}^T(n)\boldsymbol{\omega}(n-1)$
$\boldsymbol{\omega}(n) = \boldsymbol{\omega}(n-1) + e(n)\boldsymbol{P}(n)x(n)$
$\boldsymbol{P}(n) = \lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}\boldsymbol{k}(n)\boldsymbol{x}^H(n)\boldsymbol{P}(n-1)$

Table 7.2.   Summary of the conventional RLS algorithm

7.2   Adaptive Digital Predistortion Based on RLS

Figure 7.2 illustrates a general adaptive DPD system. According to the idea of adaptive algorithm, the output data from the power amplifier should be sent back continually with the same sample rate of the input signal. Hence, a coupler is applied at the end of the PA and a small amount of power will be fed back. The feedback signal should be down-converted to baseband by passing through the demodulator. Moreover, an ADC will lead the feedback baseband signal into digital domain. The coefficients of the model will be updated with the adaptive algorithm by comparing the input and output signal.

The model used in our adaptive DPD system is the memory polynomial model with odd-order terms. Therefore, the $\boldsymbol{x}(n)$ vector in (7.12) and (7.25) will be changed as following

$$\boldsymbol{x}(n) = [x(n),\ x(n)|x(n)|^2,\ \cdots,\ x(n)|x(n)|^{2(P-1)},\ x(n-1),\ \cdots\ x(n-1)|x(n-1)|^{2(P-1)},$$
$$\cdots,\ x(n-M),\ \cdots\ x(n-M)|x(n-M)|^{2(P-1)}]^T$$
$$(7.26)$$

where P and M are the polynomial order and memory depth of the memory polynomial model, respectively. The $\boldsymbol{\omega}$ vector in (7.12) and (7.25) will be the coefficients
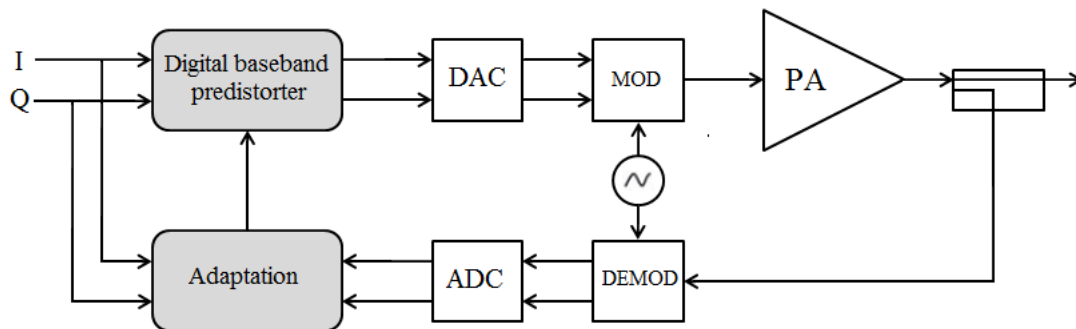
Figure 7.2.   General adaptive DPD architecture.

of the memory polynomial model. The objective of the adaptive algorithm is to update the memory polynomial model coefficients during each iteration according to the error signal.

7.3   Simulation results with RLS algorithm

The adaptive algorithms, LMS and RLS, have been tested with a combined class AB power amplifier [73]. A standard tuned load approach has been adopted in this PA and it is fabricated with monolithic microwave integrated circuits (MMICs). The related input and output signal of the PA have been acquired with the help of Matlab and VSA. The data will be used to extract the model coefficient by the adaptive algorithms.

Although the conventional adaptive algorithms, both LMS and RLS, work well for real number signals, the situation becomes worse for complex-envelope signals, which is the case in our experiment. Figure 7.3 shows the simulation results for the LMS and RLS algorithms. The y-axis is the NMSE value which describes the difference between the desired signal and predicted signal from the model. During each iteration, the model coefficients will be replaced with new values and the NMSE value will be calculated with the new model output data. In Figure 7.3, both the convergence and the final accuracy are not acceptable. For conventional RLS algorithm, it is even not converged and the NMSE is higher than -20 dB which is much higher than the least square result. Although, for LMS algorithm, it can converge, the NMSE is too high which is around -10 dB. The results are not usable in the real applications. The adaptive algorithm should be modified for the complex signals which will be discussed in next section.
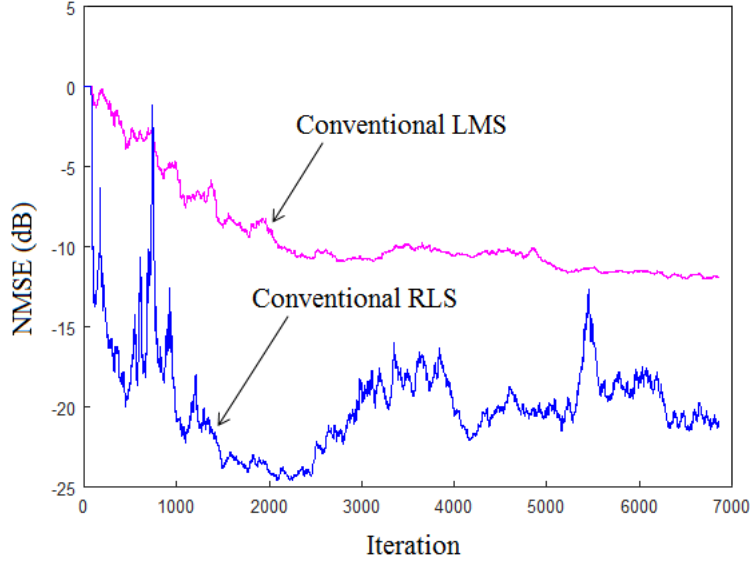
Figure 7.3.   The performance of the conventional LMS and RLS algorithm.

7.3.1   Algorithm Modification for Complex Number

Indicated by Figure 7.3, the conventional adaptive algorithms are not suitable for extracting the memory polynomial model in our case because of the bad performance in complex number domain. We will modify the adaptive algorithms for complex signals.

The memory polynomial model with polynomial order of 2 and no memory, for simplicity, is given by

$$y(n) = a_0 x(n) + a_1 x(n)|x(n)|^2 \ , \tag{7.27}$$

where both $x(n)$ and $y(n)$ are complex-envelope signals and a_0, a_1 are complex coefficients. We can separate the complex signal into real and imaginary parts:

$$\begin{aligned}
y_I + jy_Q &= (a_{0I} + ja_{0Q})(x_I + jx_Q) + (a_{1I} + ja_{1Q})(x_I + jx_Q)|x|^2 \\
&= (a_{0I}x_I - a_{0Q}x_Q) + j(a_{0I}x_Q + a_{0Q}x_I) \\
&\quad + ((a_{1I}x_I - a_{1Q}x_Q) + j(a_{1I}x_Q + a_{1Q}x_I))|x|^2
\end{aligned}$$

If two complex numbers are equal, both the real and imaginary parts of the numbers

should be equal. Therefore,

$$
\begin{aligned}
y_I &= (a_{0I}x_I - a_{0Q}x_Q) + (a_{1I}x_I - a_{1Q}x_Q)|x|^2 \\
y_Q &= (a_{0I}x_Q + a_{0Q}x_I) + (a_{1I}x_Q + a_{1Q}x_I)|x|^2 \ .
\end{aligned}
\tag{7.28}
$$

The complex model has been separated into two models with real signals. Hence, during each iteration in the LMS and RLS algorithms, the conventional computation will be processed 2 times for both real and imaginary signal samples. Moreover, the coefficients $\boldsymbol{\omega}$ and the basis functions $\boldsymbol{x}(n)$ in (7.12) and (7.25) should be rearranged as

$$
\boldsymbol{\omega} = [Re(\omega_0),\ Im(\omega_0),\ Re(\omega_1),\ Im(\omega_1),\ \cdots,\ Re(\omega_N),\ Im(\omega_N)]
$$

$$
\begin{aligned}
\boldsymbol{x}1(n) =&[Re(x(n)),\ -Im(x(n)),\ Re(x(n)|x(n)|^2),\ -Im(x(n)|x(n)|^2),\ \cdots, \\
&Re(x(n)|x(n)|^{2(P-1)}),\ -Im(x(n)|x(n)|^{2(P-1)}),\ \cdots, \\
&Re(x(n-M)),\ -Im(x(n-M)),\ \cdots, \\
&Re(x(n-M)|x(n-M)|^{2(P-1)}),\ -Im(x(n-M)|x(n-M)|^{2(P-1)})]^T
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{x}2(n) =&[Im(x(n)),\ Re(x(n)),\ Im(x(n)|x(n)|^2),\ Re(x(n)|x(n)|^2),\ \cdots, \\
&Im(x(n)|x(n)|^{2(P-1)}),\ Re(x(n)|x(n)|^{2(P-1)}),\ \cdots, \\
&Im(x(n-M)),\ Re(x(n-M)),\ \cdots, \\
&Im(x(n-M)|x(n-M)|^{2(P-1)}),\ -Re(x(n-M)|x(n-M)|^{2(P-1)})]^T
\end{aligned}
$$

where $Re(\cdot)$, $Im(\cdot)$ are the real part and imaginary part of the complex number, M and P are the memory depth and polynomial order of the model, respectively. By applying this modification, the complex number model has been divided into two real number models, for which the conventional adaptive algorithms have good enough accuracy.

Figure 7.4 compares the performance of the conventional adaptive algorithms and the proposed modified algorithms on the extraction of the memory polynomial model. The data are still obtained from the combined class AB power amplifier. In the figure, the proposed adaptive algorithms, both LMS and RLS, show a better performance than the conventional algorithms. The proposed RLS algorithm has the best results, converging with the fastest speed and obtaining the best accuracy. The final NMSE value of the RLS algorithm can reach as high as -40 dB, which is very low for the error between the desired signal and the modeled signal. Therefore, the proposed RLS algorithm with two separated conventional RLS structures is the
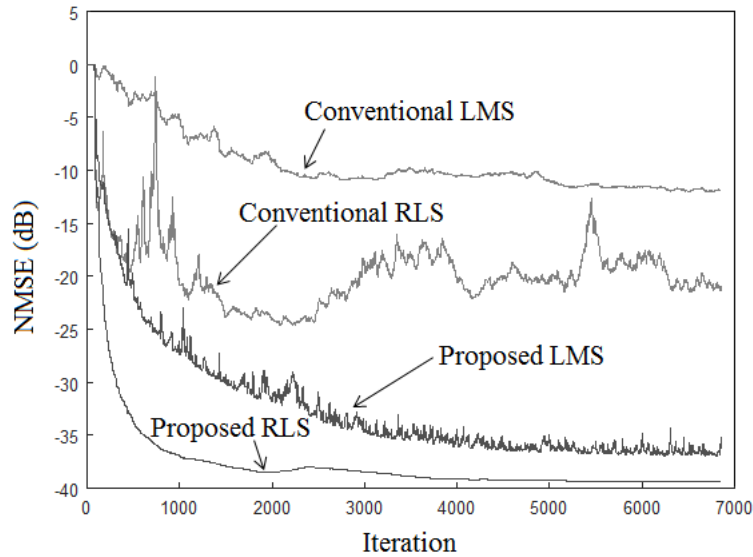
Figure 7.4.   Comparison between the conventional LMS and RLS algorithm and the modified LMS and RLS algorithm.

best choice for the adaptive DPD applications.

In conclusion, the conventional LMS and RLS algorithms have limited performance in dealing with the complex signals. The model extracted by the conventional adaptive algorithms has poor accuracy and sometimes not converged. Modification has been made to improve the performance of the conventional adaptive algorithm. The complex model could divide into two real signal models by rearranging the input signal vector and the coefficients of the adaptive filter. Better accuracy and faster converge speed have been obtained for the proposed adaptive algorithms, especially for the modified RLS algorithm. The final accuracy of the converged model has a comparable NMSE value with that obtained from least square method.

Chapter 8

Conclusion and Future Work

In modern wireless communication systems, as the demand for data continues to increase in a limited frequency range, complex digital modulation techniques are increasingly being used. However, these kinds of techniques are non-constant envelope modulation which are very sensitive to PA's nonlinearity. In addition, the modulated signal usually presents high PAPR. All these impose strict linearity requirements for the RF power amplifier. For a typical PA, obviously, the simplest solution is to back off the input power level to the linear region. This will degrade the efficiency performance of the PA, especially for signals with high PAPR. Therefore, in order to maintain both linearity and high efficiency, linearization technique is needed.

Apart from power back-off solution, there are other linearization techniques, such as feedforward, feedback method. Among all these techniques, the digital predistortion has become a preferred choice due to its relative simplicity and good accuracy. The basic idea of digital predistortion is to introduce a nonlinear block just in front of the PA. This block, called *Predistorter*, produces the inverse nonlinear response of the PA, both in magnitude and phase. In this way, the predistorter can counteract the PA's nonlinearity and the final behavior will be linear. Therefore, a model that can predict accurately the behavior of the PA and the predistorter is required. Various behavioral models were built and compared, including full Voterra model, memory polynomial model, generalized memory polynomial model, Wiener model and rational function model. Among these behavioral models, the memory polynomial model was chosen to model the PA and the predistorter since it has a good compromise between model accuracy and computational complexity. Moreover, since the memory polynomial model is linear in its parameters, least square method was applied for the model parameters extraction.

The model has been determined and in next part of this thesis, the evaluation of

114

DPD system was demonstrated with four PAs with different architectures: a hybrid class AB PA, a hybrid Doherty PA, a MMIC class AB PA and a two-stages PA. In the four tests, the linearity of the PAs had a significant improvement with the DPD system and the output power and efficiency of the PAs remained the same. In frequency domain, an emission mask (classes 6L) chosen according to the ETSI was introduced to define the linearity requirement for the output power spectrum density. For all the four PAs, the mask compliance could be achieved by the DPD system.

The following chapter of this thesis was to extent the DPD system for real applications. Therefore, to address the industrial requirements, commercial products should be used to replace the software in the DPD system. FPGA was selected which is an integrated circuit containing a large amount of logic blocks and reconfigurable interconnections. Two types of methods were discussed to implement the DPD system on FPGA: one was based on LUT method, the other one was a direct structure with only adders and multipliers. The block RAM on the FPGA was chosen to be as the LUT due to its large memory size and flexible implementation. Although the implementation methods were different, the final linearized performance was similar. The FPGA based DPD systems could also linearize the PA and fulfill the linear requirements according to the ETSI spectrum emission mask.

The last part of the thesis was on the adaptive DPD system. Since the PA's behavior may change over time due to some reasons, such as change of ambient temperature or aging, an adaptive system was required to keep the model accuracy where model parameters should be calculated and updated continually. Besides the least square method, least mean square (LMS) and recursive least square (RLS) techniques could also adopted for the adaptive system. The objective of these two algorithms was to update the model parameters sample by sample with the input and desired output signals. Although the conventional LMS and RLS algorithms has a good performance in real number applications, they presents a limited capability for models with complex number. In our test with a hybrid Doherty PA, the conventional LMS algorithm had a very poor accuracy and the RLS algorithm could not even converge. Our propose was to separate the complex memory polynomial model into 2 models with only real number. In this way, we could process the conventional LMS or RLS algorithm two times for the two separated models during each iteration. As the result, the propose solution demonstrated a much better performance, especially for the proposed RLS algorithm, which presented a fast converge speed and a high accuracy.

The linearization of wideband and multi-band PAs are the two main future challenges for DPD technique. Nowadays, as the user number and the demand for data

in wireless communication systems continue to increase, the signal bandwidth should also increase to fulfill the requirements and the digital modulation will become more spectral efficient. The resulting modulated signal usually presents wide bandwidth and large PAPR. Moreover, the sampling rate and the digital system bandwidth will increase correspondingly with the signal bandwidth. The digital predistortion system will become more complex, with more complex model and more coefficients. It will take more clock cycles to extract the model coefficients. In addition, the high sampling rate will also pose requirements for ADC and DAC systems. As a result, the high clock rate and high complexity will lead to more power consumption and more expensive components.

Another trend to transfer signal in wireless communication system is to accommodate multi-band signal which means that the PA should be able to amplify signals in multiple bands at the same time. The components of the distortions due to the PA nonlinearity are more complex, including not only harmonics, but also intermodulation distortions. Conventional behavioral models fail to predict the distortions accurately. New architecture should be developed to compensate the nonlinearities of multi-band transmitter.

Another challenge is on how to implement DPD system for low-power, small size devices. In the literature, the DPD solutions are usually designed for medium or high power amplifiers. Since the DPD system itself will consume some energy, the efficiency of the whole transmitter will be reduced, making the DPD technique improper for low-power PAs. Therefore, another open issue is on the implementation of DPD technique for low-power PAs.

Bibliography

[1] S. C. Cripps, *RF Power Amplifiers for Wireless Communications.* Norwood, MA: Artech House, 1999.

[2] R. W. Erickson, and D. Maksimovic, *Fundamentals of Power Electronics,* 2nd ed. Norwell, MA: Kliwer, 2000.

[3] H. Krauss, C. Bostian, and F. Raab, *Solid State Radio Engineering.* New York: Wiley, 1980, pp. 432-467.

[4] S. Haykin, *Adaptive Filter Theory.* Prentice Hall, Englewood Cliffs, New Jersey, 2013.

[5] Fa-Long Luo, *Digital Front-End in Wireless Communications and Broadcasting: Circuits and Signal Processing.* Cambridge: Cambridge University Press, 2011.

[6] N. Boulejfen, A. Harguem, and F. M. Ghannouchi, "New closedform expressions for the prediction of multitone intermodulation distortion in fifth-order nonlinear RF circuits/systems," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 1, pp. 121-132, Jan. 2004.

[7] A. A. Moulthrop, C. J. Clark, C. P. Silva, and M. S. Muha, "A dynamic AM/AM and AM/PM measurement technique," in *IEEE MTT-S Int. Microwave Symp. Dig.*, vol. 3, June 1997, pp. 1455-1458.

[8] J.H.K. Vuolevi and T. Rahkonen, *Distortion in RF Power Amplifiers*, Artech House, 2002.

[9] J. Vuolevi, J. Manninen, T. Rahkonen, "Memory effects compensation in RF power amplifiers by using envelope injection technique," *IEEE Radio and Wireless Conference,* Aug. 2001, p. 257 - 260.

[10] A. A. M. Saleh, "Frequency independent and frequency dependent nonlinear model of TWT amplifiers," *IEEE Trans. Commun.*, vol. COM-29, pp. 1715-1720, Nov. 1981.

[11] R. Raich, G. T. Zhou, "On the modeling of memory nonlinear effects of power amplifiers for communication applications", *Proc. 10th IEEE Digital Signal Processing Workshop*, pp.1-6

[12] Y. Zhu, J. K. Twynam, M. Yagura, M. Hasegawa, T. Hasegawa, Y. Eguchi, A. Yamada, E. Suematsu, K. Sakuno, H. Sato, and N. Hashizume, "Analytical model for electrical and thermal transients of self-heating semiconductor devices," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 2258-2263, 1998.

[13] F. H. Raab, P. Asbeck, S. Cripps, P. B. Kenington, Z. B. Popovic, N. Pothecary, J. F. Sevic, N. O. Sokal, "Power amplifiers and transmitters for RF and microwave," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no.3, pp.814-826, Mar., 2002

[14] C. J. Clark, G. Chrisikos, M. S. Muha, A. A. Moulthrop, and C. P. Silva, "Time-domain envelope measurement technique with application to wideband power amplifier modeling," *IEEE Trans. Microw. Theory Tech.*, vol. 46, no. 12, pp. 2531-40 , 1998

[15] S. Boumaiza and F. M. Ghannouchi, "Thermal memory effects modeling and compensation in RF power amplifiers and predistortion linearizers," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 12, pp. 2427-2433, Dec. 2003.

[16] A. Grebennikov, and S. Bulja, "High-efficiency Doherty power amplifiers: historical aspect and modern trends," *Proceedings of the IEEE*, vol. 100, no. 12, pp. 3190-3219, Dec. 2012.

[17] D. Cox, "Linear amplification with nonlinear components," *IEEE Trans. Commun.*, vol. COM-22, no. 12, pp. 1942-1945, 1974

[18] P. Garcia, J. de Mingo, A. Valdovinos and A. Ortega, "An adaptive digital method of imbalances cancellation in LINC transmitters," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 879-888, 2005

[19] B. Stengel and W. R. Eisenstadt, "LINC power amplifier combiner method efficiency optimization," *IEEE Trans. Veh. Technol.*, vol. 49, no. 1, pp.

229-234, 2000

[20] X. Zhang and L. E. Larson, "Gain and phase error-free LINC transmitter," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 1986-1994, 2000

[21] H. S. Black, "Translating System," U.S. Patent 1,686,792, issued October 29, 1928, and U.S. Patent 2,102,671, issued December 1937.

[22] A. S. H. Ghadam, S. Burglechner, A. H. Gokceoglu, M. Valkama, and A. Springer, "Implementation and performance of DSP-oriented feedforward power amplifier linearizer," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 2, pp. 409-425, Feb. 2012.

[23] E. Eid, F. M. Ghannouchi, and F. Beauregard, "Optimal feedforward linearization system design," Microw. J., pp. 78-86, Nov. 1995.

[24] H. Seidel, "A Microwave Feedforward Experiment," *Bell Syst. Tech. J.*, Vol. 50, November 1971, pp. 2879-2916.

[25] J. K. Cavers, "Adaption Behavior of a Feedforward Amplifier Linearizer," *IEEE Trans. Veh. Tech.*, Vol. 44, No. 1, 1995, pp. 31-40.

[26] P. B. Kenington, *High Linearity RF Amplifier Design*, Norwood, MA: Artech, 2000.

[27] T. Arthanayake and H. B. Wood, "Linear amplification using envelope feedback," *Electronics Letters*, vol. 7, no. 7, pp. 145-146, April 8, 1971.

[28] J. K. Cavers, "Amplifier linearisation using a digital predistorter with fast adaptation and low memory requirements," *IEEE Trans. Veh. Tech.*, vol. 39, no. 4, pp. 374-382, Nov. 1990.

[29] Y. Nagata, "Linear amplification techniques for digital mobile communications," *Proc. IEEE Veh. Tech. Conf. (VTC '89)*, San Francisco, pp. 159-164, May 1-3, 1989.

[30] Y. Y. Woo, J. Kim, J. Yi, S. Hong, I. Kim, J. Moon, and B. Kim, "Adaptive digital feedback predistortion technique for linearizing power amplifiers," *IEEE Trans. Microw. Theory Tech.*, vol. 55, no. 5, pp. 932-940, May 2007.

[31] M. K. Nezami, "Fundamentals of power amplifier linearization using digital

pre-distortion," *High Frequency Electronics*, pp. 54-59, Sep. 2004.

[32] D. Zhou and V. DeBrunner, "Novel adaptive nonlinear predistorter based on the direct learning algorithm," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 120-133, Jan. 2007.

[33] R. Marsalek, P. Jardin, and G. Baudoin, "From post-distortion to predistortion for power amplifiers linearization," *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 308-310, Jul. 2003.

[34] K. J. Cho, W. J. Kim, J. H. Kim, and S. P. Stapleton, "Linearity Optimization of a High Power Doherty Amplifier Based on Post-Distortion Compensation," *IEEE Commun. Lett.*, vol. 15, no. 11, pp. 748-750, Nov. 2005.

[35] C. Eun and E. J. Powers, "A new Volterra predistorter based on the indirect learning architecture," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 223-227, Jan. 1997.

[36] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "Memory polynomial predistorter based on the indirect learning architecture," in *IEEE GLOBECOM,* Nov. 2002, vol. 1, pp. 967-971.

[37] F. M. Ghannouchi and O. Hammi, "Behavioral modeling and predistortion," *IEEE Microw. Mag.*, vol. 10, no. 7, pp. 52-64, Dec. 2009.

[38] L. Guan and A. Zhu, "Green Communications–Digital Predistortion for Wideband RF Power Amplifiers," *IEEE Microw. Mag.*, vol. 15, no. 7, pp. 84-99, Dec. 2014.

[39] P. J. Lunsford, II, G. W. Rhyne, and M. B. Steer, "Frequencydomain bivariate generalized power series analysis of nonlinear analog circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 38, no. 6, pp. 815-818, 1990.

[40] O. Hammi, F. M. Ghannouchi, S. Boumaiza, and B. Vassilakis, "A data-based nested LUT model for RF power amplifiers exhibiting memory effects," *IEEE Microw. Wireless Compon. Lett.*, vol. 17, no. 10, pp. 712-714, Oct. 2007.

[41] A. Ghorbani, and M. Sheikhan, "The effect of solid state power amplifiers (SSPAs) nonlinearities on MPSK and M-QAM signal transmission," *Proceedings Sixth International Conference on Digital Processing of Signals in Communications*, Loughborough, UK, September 1991, pp. 193-197, 1991.

[42] C. Rapp, "Effects of HPA-nonlinearity on a 4-DPSK/OFDM signal for a digital sound broadcasting system," *Proceedings Second European Conference on Satellite Communications*, Liege, Belgium, pp. 179-184, Oct. 1991.

[43] M. S. O'Droma, "Dynamic range and other fundamentals of the complex Bessel function series approximation model for memoryless nonlinear devices," *IEEE Trans. Commun.*, vol. 37, pp. 397-398, Apr. 1989.

[44] A. Harguem, N. Boulejfen, F. M. Ghannouchi, and A. Gharsallah,"Robust behavioral modeling of dynamic nonlinearities using Gegenbauer polynomials with application to RF power amplifiers," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 24, n. 2, pp. 268-279, 2014.

[45] C. Jebali, N. Boulejfen, M. Rawat, et al, "Modeling of wideband radio frequency power amplifiers using Zernike polynomials," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 22, n. 3, pp. 289-296, 2012.

[46] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems.* Malabar, FL: Reprint Krieger, 2006.

[47] S. Benedetto, E. Biglieri, and R. Daffara, "Modeling and performance evaluation of nonlinear satellite links - a Volterra series approach," *IEEE Trans. Aero. Electronic Syst.*, vol. 15, no. 4, pp. 494-506, April 1979.

[48] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *IET Electron. Lett.*, vol. 37, no. 23, pp. 1417-1418, Nov. 2001.

[49] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159-165, Jan. 2004.

[50] R. N. Braithwaite, "Wide bandwidth adaptive digital predistortion of power amplifiers using reduced order memory correction," in *IEEE MTT-S Int. Microwave Symp. Dig.*, June 2008, pp. 1517-1520.

[51] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," *IEEE Trans. Veh. Technol.*, vol. 53, pp. 1468-1479, Sept. 2004.

[52] O. Hammi, A. M. Kedir, and F. M. Ghannouchi, "Nonuniform memory polynomial behavioral model for wireless transmitters and power amplifiers," *Proceedings of the 2012 IEEE Asia Pacific Microwave Conference (APMC)*, Kaohsiung, Taiwan, pp. 836-838, Dec. 2012.

[53] N. Messaoudi, M. C. Fares, S. Boumaiza, and J. Wood, "Complexity reduced odd-order memory polynomial pre-distorter for 400-watt multi-carrier Doherty amplifier linearization," *Digest 2008 IEEE MTT-S International Microwave Symposium (IMS)*, Atlanta, GA, pp. 419-422, June 2008.

[54] O. Hammi, F. M. Ghannouchi, and B. Vassilakis, "A compact envelope-memory polynomial for RF transmitters modeling with application to baseband and RF-digital predistortion," *IEEE Microw. Wireless Compon. Lett.*, vol. 18, n. 5, pp. 359-361, 2008.

[55] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Processing*, vol. 54, no. 10, pp. 3852-3860, 2006.

[56] A. Zhu, P. J. Draxler, J. J. Yan, T. J. Brazil, D. F. Kimball, and P. M. Asbeck, "Open-loop digital predistorter for RF power amplifiers using dynamic deviation reduction-based volterra series," *IEEE Trans. Microwave Theory Tech.*, vol. 56, no. 7, pp. 1524-1534, July 2008.

[57] A. Zhu, J. C. Pedro, and T. J. Brazil, "Dynamic deviation reductionbased Volterra behavioral modeling of RF power amplifiers," *IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 12, pp. 4323-4332, 2006.

[58] L. Guan and A. Zhu, "Simplified dynamic deviation reduction based volterra model for doherty power amplifiers," in *Proc. Workshop Integrated Nonlinear Microwave Millimetre-Wave Circuits*, 2011, pp. 1-4.

[59] H. D. Wasaff and J. S. Alvarez, "Rational characterization for memoryless adaptive pre-distortion," in *Proc. Eur. Signal Process. Conf.*, Sep. 2002, pp. 1-4.

[60] K. Yao, W. Niu, and M. Wang, "Adaptive RLS function for rational function predistorter," in *Proc. APMC*, Dec. 2005, pp. 1-3.

[61] T. M. Cunha, P. M. Lavrador, E. G. Lima, and J. C. Pedro, "Rational function-based model with memory for power amplifier behavioral modeling,"

in *Proc. Workshop INMMIC*, Apr. 2011, pp. 1-4.

[62] M. Rawat, K. Rawat, F. M. Ghannouchi, S. Bhattacharjee, and H. Leung, "Generalized rational functions for reduced-complexity behavioral modeling and digital predistortion of broadband wireless transmitters," *IEEE Trans. Instrum. Measur.*, vol. 63, no. 2, pp. 485-498, Feb. 2014.

[63] P. L. Gilabert, G. Montoro, and E. Bertran, "On the Wiener and Hammerstein models for power amplifier predistortion," in *Proc. Asia-Pacific Microwave Conf.*, 2005, vol. 2, pp. 1-3.

[64] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Augmented Hammerstein predistorter for linearization of broad-band wireless transmitters," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 4, pp. 1340-1349, Apr. 2006.

[65] S. H. W. Kang, Y. S. Cho, and D. H. Youn, "Adaptive precompensation of Wiener systems," *IEEE Trans. Signal Processing*, vol. 46, no. 10, pp. 2825-2829, 1998.

[66] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Deembedding static nonlinearities and accurately identifying and modeling memory effects in wide-band RF transmitters," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 11, pp. 3578-3587, Nov. 2005.

[67] http://literature.cdn.keysight.com/litweb/pdf/5989-0697EN.pdf?id=473817

[68] http://cp.literature.agilent.com/litweb/pdf/N5180-90005.pdf

[69] http://literature.cdn.keysight.com/litweb/pdf/5990-3952EN.pdf?id=1759326

[70] http://cp.literature.agilent.com/litweb/pdf/N9020-90112.pdf

[71] M. Schetzen, "Theory of pth-order inverses of nonlinear systems," *IEEE Trans. Circuits Syst.*, vol. CAS-23, no. 5, pp. 285-291, May 1976.

[72] J. Moreno Rubio, J. Fang, V. Camarchia, R. Quaglia, M. Pirola, and G. Ghione, "3-3.6 GHz wideband GaN Doherty power amplifier exploiting output compensation stages," *IEEE Trans. Microw. Theory Tech.*, vol. 60, no. 8, pp. 2543-2548, Aug. 2012.

[73] R. Quaglia, V. Camarchia, M. Pirola, J. Moreno Rubio, G. Ghione, "Linear

GaN MMIC Combined Power Amplifiers for 7-GHz Microwave Backhaul," *IEEE Trans. Microw. Theory Tech.*, vol.62, no.11, pp. 2700-2710, Nov. 2014.

[74] R. Quaglia, V. Camarchia, T. Jiang, M. Pirola, S. Donati Guerrieri, and B. Loran, "K-Band GaAs MMIC Doherty Power Amplifier for Microwave Radio With Optimized Driver," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 11, pp. 2518-2525, Nov 2014.

[75] V. Camarchia, S. Guerrieri, G. Ghione, M. Pirola, R. Quaglia, J. Moreno Rubio, B. Loran, F. Palomba, and G. Sivverini, "A K-band GaAs MMIC Doherty power amplifier for point-to-point microwave backhaul applications," in *Int. Integr. NonlinearMicrow. Millim. Wave Circuits Workshop*, Apr. 2014, pp. 1-3.

[76] V. Camarchia, V. Teppati, S. Corbellini, and M. Pirola, "Microwave Measurements, Part II Non-linear Measurements," *IEEE Instrumentation and Measurement Magazine*, vol. 10, no. 3, pp. 34-39, Jun. 2007.

[77] L. Ding and G. T. Zhou, "Effects of even-order nonlinear terms on power amplifier modeling and predistortion linearization," *IEEE Trans. Veh. Technol.*, vol. 53, pp. 156-162, Jan. 2004.

[78] H. Ku and J. S. Kenney, "Behavioral modeling of nonlinear RF power amplifiers considering memory effects," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 12, pp. 2495-2504, Dec. 2003.

[79] http://www.xilinx.com/support/documentation/user_guides/ug070.pdf

[80] http://www.xilinx.com/support/documentation/boards_and_kits /ug_xtremedsp_devkitIV.pdf

[81] P. Jardin and G. Baudoin, "Filter lookup table method for power amplifier linearization," *IEEE Trans. Veh. Technol.*, vol. 56, no. 3, pp. 1076-1087, May 2007.

[82] P. L. Gilabert, A. Cesari, G. Montoro, E. Bertran, and J.-M. Dilhac, "Multi-lookup table FPGA implementation of an adaptive digital predistorter for linearizing RF power amplifiers with memory effects," *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 2, pp. 372-384, Feb. 2008.

[83] L. Rexberg, "Power amplifier pre-distortion," U.S. Patent 20060133536, Jun. 22, 2006.

[84] L. Guan, and A. Zhu, "Low-cost FPGA implementation of Volterra series-based digital predistorter for RF power amplifiers," *IEEE Trans. Microw. Theory Tech.*, vol. 58, no. 4, pp. 866-872, Apr. 2010.

[85] A. Cesari, P. L. Gilabert, E. Bertran, G. Montoro, and J. M. Dilhac, "A FPGA based digital predistorter for RF Power amplifiers with memory effects," in *Proc. Int. Eur. Microw. Circuits Conf.*, Munich, Germany, Oct. 2007, pp. 135-138.

[86] Tao Jiang, Roberto Quaglia, Vittorio Camarchia, Marco Pirola, "FPGA-based digital predistortion of a 3.5 GHz GaN Doherty power amplifier ", In: *10th International Conference on Wireless Communications, Networking and Mobile computing*, Beijing, September 2014.

[87] https://en.wikipedia.org/wiki/Horner%27s_method#cite_note-HornerRule-2

[88] J. K. Cavers, "Optimum indexing in predistorting amplifier linearizers," in *Proc. IEEE Veh. Tech. Conf.*, Phoenix, AZ, May 1997, vol. 2, pp. 676-680.

[89] J. K. Cavers, "Optimum table spacing in predistorting amplifier linearizers," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1699-1705, Sep. 1999.

[90] J. K. Muhonen, M. Kavehrad, and R. Krishnamoorthy, "Adaptive baseband predistortion techniques for amplifier linearization," in *33rd Asilomar Signals Syst. Comput. Conf.*, Monterey, CA, Oct. 1999, vol. 2, pp. 888-892.

[91] http://anlage.umd.edu/Microwave%20Measurements%20for%20Personal%20Web%20Site/E8251-90353.pdf

[92] G.C. Goodwin, R.L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*, Academic, New York, 1977